Property Testing and Probability Distributions: New Techniques, New Models, and New Goals

Clément L. Canonne

Submitted in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy
in the Graduate School of Arts and Sciences

COLUMBIA UNIVERSITY

2017

ABSTRACT

Property Testing and Probability Distributions: New Techniques, New Models, and New Goals

Clément L. Canonne

In order to study the real world, scientists (and computer scientists) develop simplified models that attempt to capture the essential features of the observed system. Understanding the power and limitations of these models, when they apply or fail to fully capture the situation at hand, is therefore of uttermost importance.

In this thesis, we investigate the role of some of these models in property testing of probability distributions (*distribution testing*), as well as in related areas. We introduce natural extensions of the standard model (which only allows access to independent draws from the underlying distribution), in order to circumvent some of its limitations or draw new insights about the problems they aim at capturing. Our results are organized in three main directions:

(i) We provide systematic approaches to tackle distribution testing questions. Specifically, we provide two general algorithmic frameworks that apply to a wide range of properties, and yield efficient and near-optimal results for many of them. We complement these by introducing two methodologies to prove information-theoretic lower bounds in distribution testing, which enable us to derive hardness results in a clean and unified way.

(ii) We introduce and investigate two new models of access to the unknown distributions, which both generalize the standard sampling model in different ways and allow testing algorithms to achieve significantly better efficiency. Our study of the power and limitations of algorithms in these models shows how these could lead to faster algorithms in practical situations, and yields a better understanding of the underlying bottlenecks in the standard sampling setting.

(iii) We then leave the field of distribution testing to explore areas adjacent to property testing. We define a new algorithmic primitive of *sampling correction*, which in some sense lies in between distribution learning and testing and aims to capture settings where data originates from imperfect or noisy sources. Our work sets out to model these situations in a rigorous and abstracted way, in order to enable the development of systematic methods to address these issues.

Contents

List of Figures

List of Tables

Acknowledgments

> The Butcher would gladly have talked till next day,
> But he felt that the Lesson must end,
> And he wept with delight in attempting to say
> He considered the Beaver his friend.
>
> Lewis Carroll, *The Hunting of the Snark*

As Isaac Newton famously did not say, *"We are but dwarfs standing on the shoulders of somewhat bigger dwarfs."* Looking back at these past five years, I cannot help but let this non-quote resonate in my heart: and indeed, I may not have seen much further; indeed, I may not really have stood that much – but for whatever I did, I feel so grateful, and to so many.

Of course, it is common practice to thank one's advisor, in my case Rocco Servedio. *Common practice be damned:* I thank Rocco wholeheartedly, as the best advisor one could hope for and fathom, as the finest human being one could possibly envision. During these five years, he never made me feel dumb – I was; he never complained about my bothering him – I did; he was always thoughtful, smiling, knowledgeable, insightful; always available. I cannot possibly thank him enough – here's only a poor attempt. Thank you, Rocco.

This set the bar quite high; yet, I was lucky enough to interact and be mentored by people above this bar, and way beyond. I specifically aim these thanks at Dana Ron, Ronitt Rubinfeld, and Madhu Sudan; who not only taught me so much, but also left me convinced that every single person in our field was wonderful, approachable, and intrinsically awesome. I hope I can live to their standard. I also am incredibly grateful to all my coauthors and collaborators: Jayadev Acharya, Tuğkan Batu, Eric Blais, Omri Ben-Eliezer, Anindya De, Ilias Diakonikolas, Themis Gouleakis, Elena Grigorescu, Talya Eden, Tom Gur, Venkat Guruswami, Daniel Kane, Gautam Kamath, Akash Kumar, Amit Levi, Reut Levi, Raghu Meka, Igor Carboni Oliveira, Alistair Stewart, Li-Yang Tan, and Karl Wimmer. I cannot promise I've learned from all of you – I *can* promise I tried, and wish I did.

I must also thank everybody that made my graduate life at Columbia so enjoyable and fulfilling, or, more pragmatically, even possible: Alex Andoni, Xi Chen, Fernando Krell, Igor Carboni Oliveira, Tal Malkin, Dimitris Paparas, Jessica Rosa, Erik Waingarten and – of course! – the coffee machine on the 5^{th} floor. Well, you know – thank you. (Incidentally, Jessica, I sure hope I didn't drive you insane.) I must also thank the whole department at MIT; Gautam, again, and foremost; Ilya, Jerry, and everyone in CSAIL. You welcomed me, and all I had to offer was bad puns. Which brings me to my flatmates and friends in New York City and beyond. Thank you Rémi, Joschi, Yumi, Laurent, Fay, and Juba; thank you, Vicky, and Narges. Thank you,

Alaa. You put up with me for all these years, through all these jokes I'm pretty sure were not all funny: I don't know why, but I sure am grateful. Also, please, get a better sense of humor.

To my family – Mom, Dad, Thomas, Cécile, Marion, Nicolas, Quentin, Margot, Sacha, and now (!) Lola. I will not thank you, the point is kind of moot – you know what you are to me, what you have been all this time; and how could I possibly ever acknowledge so much? (Sorry, there is no joke here. I'm damn serious.) Finally, to Nandini. For everything, and more.

To Nandini, who told me what a wombat was.

Introduction

> "The thing can be done," said the Butcher, "I think.
> The thing must be done, I am sure.
> The thing shall be done! Bring me paper and ink,
> The best there is time to procure."
>
> Lewis Carroll, *The Hunting of the Snark*

This dissertation revolves around discrete probability distributions: the wild and empirical ones, found in the "real world" wherever data can be found; or the familiar and abstract ones, which underly our (idealized) models of that very same world and let us reason about it. The practical details of the situations in which these distributions show up will not be of too much concern for us: instead, we will take their presence as a given, seeing them as an abstract source of data, values – "samples."

And indeed, inferring *information* from the probability distribution underlying available data is a fundamental problem in Statistics and data analysis, with applications and ramifications in countless other fields. One may want to approximate that distribution in its entirety; or, less ambitiously, to check whether it is consistent with a prespecified model; one may even only want to approximate some simple parameters such as its mean or first few moments. But this decades-old inference question, regardless of its specific variant, has undergone a significant shift these past few years: the amount of data to analyze has grown huge, and our distributions now are often over a *very large* domain. So huge and so large, in fact, that the seasoned and well-studied methods from Statistics and learning theory are no longer practical; and one has to look for faster, more sample-efficient techniques and algorithms.

We may not be able to obtain these in general. But in many situations, we are only interested in figuring out some very specific information about our probability distribution: we made an assumption or formulated a hypothesis, and want to check whether we were right. To get this *one bit* of information, and this bit only, it may just be possible to overcome the formidable complexity of the task. Understanding when it is, and how, is precisely what the field of distribution testing is about.

Distribution testing, as first explicitly introduced in [22], is a branch of property testing [156, 103]: in the latter, access to an unknown "huge object" is presented to an algorithm *via* the ability to perform local "inspections." By making only a small number of such queries to the object, the algorithm must determine with high probability whether the object exhibits some prespecified property of interest, or is *far* from every object with the property. (For a more detailed presentation and overview of the field of property testing, the reader is referred to [95, 150, 149, 101, 100, 28].)

In distribution testing, this "huge object" is an unknown probability distribution (or a collection thereof) over some known (usually discrete) domain Ω; and the type of access granted to this distribution is (usually) access to independent samples drawn from that distribution. The question now becomes to bound the number of samples required to test a given statistical property – as a function of the domain size and the "farness" parameter:

> Given a property of distributions $\mathcal{P}$ and access to an *arbitrary* distribution $\mathbf{p}$, distinguish between the case that (a) $\mathbf{p} \in \mathcal{P}$, versus (b) $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{p}') > \varepsilon$ for all $\mathbf{p}' \in \mathcal{P}$.

Here, d_{TV} denotes the total variation distance, also known as statistical distance. (We note that the focus is explicitly on the sample complexity: the running time of the algorithm is usually only a secondary concern, even though obtaining time-efficient testers is an explicit goal in many works.) Distribution testing has been a very active area over the past fifteen years, with a flurry[1] of variants and exciting developments: starting with [104, 20, 21], this includes the testing of symmetric properties [146, 174, 171, 172], of structured families [19, 117, 2, 44, 3, 51, 43], as well as testing under some assumption on the unknown instance [155, 74, 82, 81]. Tight upper and lower bounds on the sample complexity have been obtained for a vast number of properties such as uniformity, identity to a specified distribution, monotonicity, independence, and many more. We refer the reader of this thesis to the surveys [154, 42], and the book [100], for a more complete picture; and will focus afterwards on our narrow contribution to this field.

Our contributions

Before delving into the specific and technical details, we provide a high-level overview of our contributions. As we shall see, they are organized in three main axes:

Beyond the standard distribution testing *techniques*

As aforementioned, distribution testing has been the focus of a significant body of works over recent years, culminating in a full understanding of the complexity for a large number of testing questions. However, while many of these questions have seen their sample complexity fully resolved, these advances have for a large part been the result of distinct, *ad hoc* techniques tailored to the specific problems they were meant to solve. Overall, we still lack *general* tools to tackle distribution testing questions – both to establish (algorithmic) upper and (information-theoretic) lower bounds.

The first contribution of this thesis is to establish both general algorithmic frameworks ("Swiss Army knives") and lower bound techniques ("easy reductions") to attack these questions, in Chapters 2 and 3 respectively. Our results are widely applicable, and yield optimal or near-optimal bounds for a variety of (arguably) fundamental testing questions. In this sense, our work can be viewed as building up a user-friendly toolbox for distribution testing, which should come in handy to anyone in the field.

[1] I immensely enjoy the word "flurry."

Beyond the *standard* distribution testing techniques

One of the takeaway messages of the aforementioned recent flurry of results in distribution testing is that achieving a sublinear sample complexity with regard to the domain size *is* possible for most properties of interest. Another takeaway, however, is that this sublinear sample complexity has to be *polynomial* in this domain size n, i.e. of the form $n^{\Omega(1)}$ – which, in many real-world settings, turns out to still be prohibitively high. Thus, it is reasonable to consider natural extensions of the standard "sample-only" model, where algorithms now get to have a stronger type of *access* to the unknown probability distribution – and see if this additional power allows them to achieve a significant better sample complexity.

In Chapter 4 of this thesis, we introduce and study two such generalizations (along with some of their variants), which we argue can be implemented in practical situations. The main message is that, whenever these new models are applicable, one can perform much better than in the standard sampling model – sometimes with a sample complexity *independent* of the domain size. Moreover, such stronger models can also help us in understanding what exactly makes these questions "hard" in the standard sampling model in the first place, and therefore hopefully guide implementations even of "standard" testing algorithms by adapting them on a case-by-case basis.

Beyond the standard distribution *testing* techniques

So far, we stayed within the realm of distribution testing: focusing on a specific property of distributions, how to decide whether the unknown one we have access to indeed satisfies this property. This, however, may not be the end goal: for instance, what if after running such a test, we knew the distribution is *close* to having that property – yet are not guaranteed it does *exactly*? What if a subsequent algorithm, or application, requires such a guarantee? To handle such questions, we introduce in Chapter 5 the notion of a *sampling corrector*, which (broadly speaking) acts as a filter between a source of imperfect samples and an algorithm to provide access to "corrected samples" – whose distribution is close to the original one, but now does satisfy the property of interest. We further explore this new paradigm of simple correction (and its weaker variant of sampling *improvement*), and study its connections to distribution testing and learning – showing two-way implications that may prove fruitful in establishing new upper and lower bound in either direction.

Organization of the dissertation

In Chapter 1, we lay down the necessary notation and definitions that will be used throughout this thesis, and state some results from the literature that we shall need afterwards. We will also prove there several simple results that will be relied upon in the other chapters, and more generally set up the board and pieces. Chapter 2 then will be concerned with general strategies to play the game; or, put differently, with unified frameworks to obtain algorithmic *upper bounds* on distribution testing questions. In more detail, Section 2.1 describes a unified approach for testing membership in classes of distributions, particularly relevant for classes of

shape-restricted distributions; while Section 2.2 contains a different approach for this question, well-suited for those classes of distributions which enjoy "nice" Fourier spectra. The first is based on joint work with Ilias Diakonikolas, Themis Gouleakis, and Ronitt Rubinfeld [51], and the second on the paper [45] with Ilias Diakonikolas and Alistair Stewart.

In Chapter 3, we complement these algorithmic frameworks by describing new general approaches to obtaining information-theoretic *lower bounds* in distribution testing. Section 3.1, based on [51], describes a reduction technique which allows us to lift hardness of testing a sub-property $\mathcal{P}' \subseteq \mathcal{P}$ to that of testing $\mathcal{P}$ itself, modulo a mild learnability condition on the latter. As a corollary, we obtain new (as well as previously known) lower bounds for many distribution classes, in a clean and unified way. Section 3.2 (based on the paper [34] with Eric Blais and Tom Gur) then provides another framework to easily establish distribution testing lower bounds, this time by carrying over lower bounds from *communication complexity*. We show how this reduction from communication complexity, besides enabling us to easily derive lower bounds for a variety of distribution testing questions, can also shed light on existing results, leading to an unexpected connection between distribution testing and the seemingly unrelated field of interpolation theory.

In these two chapters, we were concerned with the "standard" setting of distribution testing, which only assumes access to independent samples; and developed general methods to tackle questions in this setting. In Chapter 4, we take a different path: instead of finding new strategies to play the game, we change the *rules* themselves – granting the testing algorithms a more powerful type of access to the unknown distribution. Based on a work with Dana Ron and Rocco Servedio [49], Section 4.1 introduces and studies the *conditional sampling model*, in which the algorithm can get samples from the underlying probability distribution conditioned on subsets of events of its choosing. In Section 4.2, we define and study two different settings, the *dual access* and *cumulative dual access* models, in which one can both draw independent samples from the distribution and query the value on any point of the domain of either its probability mass function or cumulative distribution function. (This is based on the paper [50], with Ronitt Rubinfeld.) Both sections thus consider testing algorithms that are at least as powerful as those from the standard sampling setting; the question is to quantify *how much* more powerful these algorithms can be, and what limitations remain.

Finally, in Chapter 5 we venture out of property testing to explore a different – albeit related – paradigm: that of distribution *correcting*. Changing now the *goal* of the game, we introduce the notion of sampling corrector: granted access to independent samples from a probability distribution only *close* to having some property $\mathcal{P}$ of interest, one must provide access to samples from a "corrected" distribution which, while still being close to the original distribution, does satisfy $\mathcal{P}$. We prove general results on this new algorithmic primitive, and study its relation to both distribution learning and testing; before focusing specifically on correction of a well-studied property of distributions, monotonicity. This last chapter contains material from [46], joint work with Themis Gouleakis and Ronitt Rubinfeld.

Chapter 1

Set up and Preliminaries

"Skip all that!" cried the Bellman in haste.
If it once becomes dark, there's no chance of a Snark–
We have hardly a minute to waste!"

Lewis Carroll, *The Hunting of the Snark*

1.1 Notation.

All throughout this thesis, we denote by $[n]$ the set $\{1, \dots, n\}$, and by $[\![n]\!]$ the set $\{0, \dots, n-1\}$. We will write $\log$ (resp. $\ln$) for the binary logarithm (resp. the natural logarithm). Besides the standard asymptotic conventions, we use the notations $\tilde{O}(f), \tilde{\Omega}(f)$ to hide polylogarithmic dependencies on the argument, and will sometimes write $O_\varepsilon(f)$ to signify that the hidden constant depends on the parameter ε (while f does not).

We now formally introduce the main actor of this dissertation. A *probability distribution* over a (countable) domain[1] Ω is a non-negative function $\mathbf{p}\colon \Omega \to [0,1]$ such that $\sum_{x\in\Omega} \mathbf{p}(x) = 1$. We denote by $\Delta(\Omega)$ the (convex) polytope of all such distributions, and by $\mathbf{u}(\Omega)$ the uniform distribution on Ω (when well-defined); when clear from context, we may sometimes omit the domain and simply write $\mathbf{u}$. Given a distribution $\mathbf{p}$ over Ω and a set $S \subseteq \Omega$, we write $\mathbf{p}(S)$ for the total probability weight $\sum_{x\in S} \mathbf{p}(x)$ assigned to S by $\mathbf{p}$. Moreover, for $S \subseteq \Omega$ such that $\mathbf{p}(S) > 0$, we denote by $\mathbf{p}_S$ the conditional distribution of $\mathbf{p}$ restricted to S, that is $\mathbf{p}_S(x) = \frac{\mathbf{p}(x)}{\mathbf{p}(S)}$ for $x \in S$ and $\mathbf{p}_S(x) = 0$ otherwise. We also let $\mathrm{supp}(\mathbf{p}) \stackrel{\text{def}}{=} \{\, x \in \Omega :\ \mathbf{p}(x) > 0 \,\}$ be the *(effective) support* of the distribution, i.e. the subset of the domain to which $\mathbf{p}$ assigns non-zero probability weight . Finally, for a probability distribution $\mathbf{p} \in \Delta(\Omega)$ and integer m, we write $\mathbf{p}^{\otimes m} \in \Delta(\Omega^m)$ for the m-fold product distribution obtained by drawing m independent samples $s_1, \dots, s_m \sim \mathbf{p}$ and outputting $(s_1, \dots, s_m)$.

When the domain is a subset of the natural numbers $\mathbb{N}$, we shall often abuse notation and identify a distribution $\mathbf{p} \in \Delta(\Omega)$ with the sequence $(\mathbf{p}_i)_{i\in\Omega} \in \ell_1$ corresponding to its probability mass function (pmf).

[1] For the sake of this thesis, all distributions will be supported on a finite or at least discrete domain; thus, we do not consider the fully general definitions from measure theory.

1.2 Property testing, distributions, and metrics.

As is usual in property testing of distributions, throughout this dissertation the distance between two distributions $\mathbf{p}_1, \mathbf{p}_2 \in \Delta(\Omega)$ will be the *total variation distance*:

$$\mathrm{d}_{\mathrm{TV}}(\mathbf{p}_1, \mathbf{p}_2) \stackrel{\text{def}}{=} \max_{S \subseteq \Omega} (\mathbf{p}_1(S) - \mathbf{p}_2(S)) = \frac{1}{2} \sum_{x \in \Omega} |\mathbf{p}_1(x) - \mathbf{p}_2(x)| = \frac{1}{2} \|\mathbf{p}_1 - \mathbf{p}_2\|_1 \tag{1.1}$$

which takes value in $[0, 1]$. (Due to the equivalence between total variation and ℓ_1 distances, we will sometimes phrase our results in terms of the latter, and ask the reader for their forgiveness.) In some cases, it is useful to consider – either as a proxy towards total variation, or for the sake of the analysis – different metrics, such as ℓ_2, Kolmogorov, or Hellinger distances. More on these can be found in Section 1.4.

A property $\mathcal{P}$ of distributions over Ω is then simply a subset of $\Delta(\Omega)$, consisting of all distributions that have the property. The distance from $\mathbf{p}$ to a property $\mathcal{P}$, denoted $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathcal{P})$, is then defined as $\inf_{\mathbf{p}' \in \mathcal{P}} \mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{p}')$. Given a distribution $\mathbf{p}$ and a property $\mathcal{P}$, we say that $\mathbf{p}$ is ε-*close* to $\mathcal{P}$ if $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathcal{P}) \leq \varepsilon$; otherwise, $\mathbf{p}$ is ε-*far* from $\mathcal{P}$. We shall oftentimes refer to some properties as "classes" of distribution, trading $\mathcal{P}$ for the symbol $\mathcal{C}$; specifically, this will be the case for *structured* properties of distributions, in keeping with the existing literature.

We recall the standard definition of testing algorithms for properties of distributions over Ω, where n is the relevant parameter for Ω (i.e., in most cases, its size $|\Omega|$). To be consistent with the rest of this dissertation, we chose to phrase it in the most general setting possible, with regard to how the unknown distribution is "queried": and will specify this aspect further in the relevant chapters (sampling access, conditional access, etc.).

Definition 1.2.1. Let $\mathcal{P}$ be a property of distributions over Ω. Let $\mathsf{ORACLE}_{\mathbf{p}}$ be an oracle providing some type of access to $\mathbf{p}$. A *q-query testing algorithm for $\mathcal{P}$* (for this type of oracle) is a randomized algorithm $\mathcal{T}$ which takes as input $n \in \mathbb{N}$, $\varepsilon \in (0, 1)$, as well as access to $\mathsf{ORACLE}_{\mathbf{p}}$. After making at most $q(\varepsilon, n)$ calls to the oracle, $\mathcal{T}$ either outputs accept or reject, such that the following holds:

- if $\mathbf{p} \in \mathcal{P}$, then with probability at least $2/3$, $\mathcal{T}$ outputs accept;
- if $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathcal{P}) > \varepsilon$, then with probability at least $2/3$, $\mathcal{T}$ outputs reject;

where the probability is taken over the algorithm's randomness and (if any) the randomness from the oracle's answers.

The most common type of oracle is the "sampling oracle," which provides access to independent samples drawn from $\mathbf{p}$. Besides this standard definition of testing algorithms, we will also be interested in a generalization, that of *tolerant* testers – roughly, algorithms robust to a relaxation of the first item above:

Definition 1.2.2. Let $\mathcal{P}$ and $\mathsf{ORACLE}_{\mathbf{p}}$ be as above. A *q-query tolerant testing algorithm for $\mathcal{P}$* is a randomized algorithm $\mathcal{T}$ which takes as input $n \in \mathbb{N}$, $0 \leq \varepsilon_1 < \varepsilon_2 \leq 1$, as well as access to ORACLE_D. After

making at most $q(\varepsilon_1, \varepsilon_2, n)$ calls to the oracle, $\mathcal{T}$ outputs either accept or reject, such that the following holds:

- if $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathcal{P}) \leq \varepsilon_1$, then with probability at least $2/3$, $\mathcal{T}$ outputs accept;
- if $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathcal{P}) \geq \varepsilon_2$, then with probability at least $2/3$, $\mathcal{T}$ outputs reject;

where the probability is taken over the algorithm's randomness and (if any) the randomness from the oracle's answers.

Note that these definitions in particular do not specify the behavior of the algorithms when $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathcal{P}) \in (0, \varepsilon)$ (resp. $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathcal{P}) \in (\varepsilon_1, \varepsilon_2)$): in this case, any answer from the tester is considered valid. Furthermore, we stress that the two definitions above only deal with the query complexity, and not the running time. Almost every lower bound will however apply to computationally unbounded algorithms, while most upper bounds we will cover are achieved by testing algorithms whose running time is polynomial in the number of queries they make.

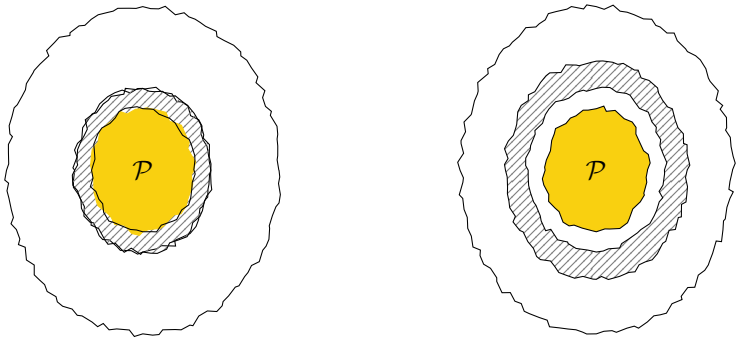

Figure 1.1: Testing vs. tolerant testing: the algorithm is off the hook whenever the unknown distribution belongs to the gray area. It looks like eggs, really.

A related notion is that of *distance estimators*; that is, of algorithms which compute an approximation of the distance of the unknown distribution to a property.

Definition 1.2.3. Let $\mathcal{P}$ and $\mathsf{ORACLE}_{\mathbf{p}}$ be as above. A *q-query distance estimation algorithm for $\mathcal{P}$* is a randomized algorithm $\mathcal{A}$ which takes as input $n \in \mathbb{N}$, $\varepsilon \in (0, 1]$, as well as access to ORACLE_D. After making at most $q(\varepsilon, n)$ calls to the oracle, $\mathcal{T}$ outputs a value $\gamma \in [0, 1]$ such that, with probability at least $2/3$, it holds that $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathcal{P}) \in [\gamma - \varepsilon, \gamma + \varepsilon]$.

Remark 1.2.4 (Tolerant testing and distance approximation). Parnas, Ron, and Rubinfeld define and formalize in [140] the notion of tolerant testing, and show that distance approximation and (fully)[2] tolerant testing are equivalent, up to a logarithmic factor in $1/\varepsilon$ in the sample complexity (Claims 1 and 2, Section 3.1).

[1] Note that, as standard in property testing, the threshold $2/3$ is arbitrary: any $1 - \delta$ confidence can be achieved at the cost of a multiplicative factor $\log(1/\delta)$ in the query complexity, by repeating the test and outputting the majority vote.

[2] I.e., tolerant testing algorithms as above that allow *any* inputs $0 \leq \varepsilon_1 < \varepsilon_2 \leq 1$, without further restriction on the range of authorized values.

The last notion we shall require is that of distribution *learning* (also referred to as *density estimation*). The exact formalization of what *learning a probability distribution* means has been considered in Kearns et al. [120]. We note that in their language, the variant of learning this thesis considers is *learning to generate*.[3] We give the precise definition below:

Definition 1.2.5. Let $\mathcal{C} \subseteq \Delta(\Omega)$ be a class of probability distributions and $\mathbf{p} \in \mathcal{C}$ be an unknown distribution. Let also $\mathcal{H}$ be a hypothesis class of distributions. A *q-sample learning algorithm for $\mathcal{C}$* is a randomized algorithm $\mathcal{L}$ which, given sample access to $\mathbf{p}$ and parameters $\varepsilon, \delta \in (0,1)$, outputs the description of a distribution $\hat{\mathbf{p}} \in \mathcal{H}$ such that with probability at least $1-\delta$ one has $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \hat{\mathbf{p}}) \leq \varepsilon$.
If in addition $\mathcal{H} \subseteq \mathcal{C}$, then we say $\mathcal{L}$ is a *proper* learning algorithm.

The above definition assumes that the probability distribution to be approximated belongs to a known class $\mathcal{C}$. However, in many cases and applications this assumption may not be exactly satisfied – this is often referred to as "model misspecification." In that case, one may still ask for a learning algorithm which would approximate $\mathbf{p}$ "as well as the best distribution from $\mathcal{C}$." This generalization of the above notion of distribution learning is known as *agnostic learning*:

Definition 1.2.6. Let $\mathcal{C}$ and $\mathcal{H}$ be as above. A *(semi-)agnostic learning algorithm for $\mathcal{C}$* (using hypothesis class $\mathcal{H}$) is an algorithm $\mathcal{A}$ which, given sample access to an arbitrary distribution $\mathbf{p}$ and parameters $\varepsilon, \delta \in (0,1)$, outputs a hypothesis $\hat{\mathbf{p}} \in \mathcal{H}$ such that, with probability at least $1-\delta$,

$$\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \hat{\mathbf{p}}) \leq c \cdot \textsc{opt}_{\mathcal{C},\mathbf{p}} + \varepsilon$$

where $\textsc{opt}_{\mathcal{C},\mathbf{p}} \stackrel{\mathrm{def}}{=} \inf_{\mathbf{p}' \in \mathcal{C}} \mathrm{d}_{\mathrm{TV}}(\mathbf{p}', \mathbf{p})$ and $c \geq 1$ is an absolute constant (if $c = 1$, the learner is said to be *agnostic*).

Generalization. These definitions can easily be extended to cover situations in which there are two (or more) "unknown" distributions $\mathbf{p}_1, \mathbf{p}_2$ that are accessible respectively via $\mathsf{ORACLE}_{\mathbf{p}_1}$ and $\mathsf{ORACLE}_{\mathbf{p}_2}$ oracles. For instance, we shall consider algorithms for testing whether $\mathbf{p}_1 = \mathbf{p}_2$ versus $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}_1, \mathbf{p}_2) > \varepsilon$ in such a setting, the property now being formally a subset of $\Delta(\Omega) \times \Delta(\Omega)$.

On adaptivity and one-sidedness. As usual in property testing, it is possible to specialize these definitions for some classes of algorithms. In particular, a tester which never errs when $\mathbf{p} \in \mathcal{P}$ (but is only allowed to be wrong with probability $1/3$ when $\mathbf{p}$ is far from $\mathcal{P}$) is said to be *one-sided*; as defined above, testers are *two-sided*.[4] More important in this thesis is the notion of *adaptive* testers: if an algorithm's queries do not depend on the previous answers made to the oracle(s), it is said to be *non-adaptive*. However, if the i-th

[3] We remark that the notion of distance Kearns et al. considered was that of Kullback–Leibler (KL) divergence; while here – as in most of the recent distribution learning literature – we focus on total variation distance.

[4] Most of the algorithms we shall be concerned with will be two-sided: this follows from the simple observation that, for almost any property of interest, in the standard sampling model one-sided testers are information-theoretically impossible (see e.g. [100, Chapter 11]).

query can be a function of the j-th answer for $j < i$, then it is *adaptive*. (Roughly speaking, a non-adaptive algorithm is one that can write down all the queries it is going to make "in advance," only after tossing its own random coins).

1.3 Classes of distributions

We give here the formal descriptions of the classes of distributions that shall appear in this dissertation, starting with that of monotone distributions.

Definition 1.3.1 (monotone)**.** A distribution $\mathbf{p}$ over $[n]$ is *monotone* (non-increasing) if its probability mass function (pmf) satisfies $\mathbf{p}(1) \geq \mathbf{p}(2) \geq \dots \mathbf{p}(n)$.

A natural generalization of the class $\mathcal{M}_n$ of monotone distributions is the set of t-modal distributions, i.e. distributions whose pmf can go "up and down" or "down and up" up to t times:[5]

Definition 1.3.2 (t-modal)**.** Fix any distribution $\mathbf{p}$ over $[n]$, and integer t. $\mathbf{p}$ is said to have t *modes* if there exists a sequence $i_0 < \dots < i_{t+1}$ such that either $(-1)^j \mathbf{p}(i_j) < (-1)^j \mathbf{p}(i_{j+1})$ for all $0 \leq j \leq t$, or $(-1)^j \mathbf{p}(i_j) > (-1)^j \mathbf{p}(i_{j+1})$ for all $0 \leq j \leq t$. We call $\mathbf{p}$ t-*modal* if it has at most t modes, and write $\mathcal{M}_{n,t}$ for the class of all t-modal distributions. The particular case of $t = 1$ corresponds to the set $\mathcal{M}_{n,1}$ of *unimodal* distributions.

Definition 1.3.3 (Log-concave)**.** A distribution $\mathbf{p}$ over $[n]$ is said to be *log-concave* if it satisfies the following conditions: (i) for any $1 \leq i < j < k \leq n$ such that $\mathbf{p}(i)\mathbf{p}(k) > 0$, $\mathbf{p}(j) > 0$; and (ii) for all $1 < k < n$, $\mathbf{p}(k)^2 \geq \mathbf{p}(k-1)\mathbf{p}(k+1)$. We write $\mathcal{LCV}_n$ for the class of all log-concave distributions.

Definition 1.3.4 (Concave and Convex)**.** A distribution $\mathbf{p}$ over $[n]$ is said to be *concave* if it satisfies the following conditions: (i) for any $1 \leq i < j < k \leq n$ such that $\mathbf{p}(i)\mathbf{p}(k) > 0$, $\mathbf{p}(j) > 0$; and (ii) for all $1 < k < n$ such that $\mathbf{p}(k-1)\mathbf{p}(k+1) > 0$, $2\mathbf{p}(k) \geq \mathbf{p}(k-1) + \mathbf{p}(k+1)$; it is *convex* if the reverse inequality holds in (ii). We write $\mathcal{K}_n^-$ (resp. $\mathcal{K}_n^+$) for the class of all concave (resp. convex) distributions.

It is easy to see that convex and concave distributions are unimodal; moreover, every concave distribution is also log-concave, i.e. $\mathcal{K}_n^- \subseteq \mathcal{LCV}_n$. Note that in both Definition 1.3.3 and Definition 1.3.4, condition (i) is equivalent to enforcing that the distribution be supported on an interval.

Definition 1.3.5 (Monotone Hazard Rate)**.** A distribution $\mathbf{p}$ over $[n]$ is said to have *monotone hazard rate* (MHR) if its *hazard rate* $H(i) \stackrel{\text{def}}{=} \frac{\mathbf{p}(i)}{\sum_{j=i}^{n} \mathbf{p}(j)}$ is a non-decreasing function. We write $\mathcal{MHR}_n$ for the class of all MHR distributions.

It is known that every log-concave distribution is both unimodal and MHR (see e.g. [9, Proposition 10]), and that monotone distributions are MHR. Two other classes of distributions have elicited significant interest in

[5] Note that this slightly deviates from the Statistics literature, where only the peaks are counted as modes (so that what is usually referred to as a bimodal distribution is, according to our definition, 3-modal).

the context of density estimation, those of *histograms* (piecewise constant) and *piecewise polynomial densities*:

Definition 1.3.6 (Piecewise Polynomials [55])**.** A distribution $\mathbf{p}$ over $[n]$ is said to be a *t-piecewise degree-d distribution* if there is a partition of $[n]$ into t disjoint intervals $I_1, \ldots, I_t$ such that $\mathbf{p}(i) = p_j(i)$ for all $i \in I_j$, where each $p_1, \ldots p_t$ is a univariate polynomial of degree at most d. We write $\mathcal{P}_{n,t,d}$ for the class of all t-piecewise degree-d distributions. (We note that t-piecewise degree-0 distributions are also commonly referred to as *t-histograms*, and write $\mathcal{H}_{n,t}$ for $\mathcal{P}_{n,t,0}$.)

Finally, we recall the definition of the two following classes, which both extend the family of Binomial distributions $\mathcal{BIN}_n$: the first, by removing the need for each of the independent Bernoulli summands to share the same bias parameter.

Definition 1.3.7. A random variable X is said to follow a *Poisson Binomial Distribution* (with parameter $n \in \mathbb{N}$) if it can be written as $X = \sum_{k=1}^{n} X_k$, where $X_1 \ldots, X_n$ are independent, non-necessarily identically distributed Bernoulli random variables. We denote by $\mathcal{PBD}_n$ the class of all such Poisson Binomial Distributions.

It is not hard to show that Poisson Binomial Distributions are in particular log-concave. One can generalize even further, by allowing each random variable of the summation to be integer-valued:

Definition 1.3.8. Fix any $k \geq 0$. We say a random variable X is a *k-Sum of Independent Integer Random Variables* with parameter $n \in \mathbb{N}$ ((n,k)-SIIRV) if it can be written as $X = \sum_{j=1}^{n} X_j$, where $X_1 \ldots, X_n$ are independent, non-necessarily identically distributed random variables taking value in $[\![k]\!]$. We denote by $\mathcal{SIIRV}_{n,k}$ the class of all such (n,k)-SIIRVs.

(The class of Poisson Binomial Distributions thus corresponds to the case $k = 2$, that is $(n,2)$-SIIRVS.) A different type of generalization is that of Poisson Multinomial Distributions, where each summand is a random variable supported on the k vectors of the standard basis of $\mathbb{R}^k$, instead of $[k]$:

Definition 1.3.9. Fix any $k \geq 0$. We say a random variable X is a *(n,k)-Poisson Multinomial Distribution ((n,k)-PMD)* with parameter $n \in \mathbb{N}$ if it can be written as $X = \sum_{j=1}^{n} X_j$, where $X_1 \ldots, X_n$ are independent, non-necessarily identically distributed random variables taking value in $\{e_1, \ldots, e_k\}$ (where $(e_i)_{i \in [k]}$ is the canonical basis of $\mathbb{R}^k$). We denote by $\mathcal{PMD}_{n,k}$ the class of all such (n,k)-PMDs.

1.4 Previous tools from the literature.

As previously mentioned, in this thesis we will be concerned with the total variation distance between distributions. Of interest for the analysis of some of our algorithms, and assuming Ω is totally ordered (in our case, $\Omega = [n]$), one can also define the *Kolmogorov distance* between $\mathbf{p}_1$ and $\mathbf{p}_2$ as

$$\mathrm{d}_{\mathrm{K}}(\mathbf{p}_1, \mathbf{p}_2) \stackrel{\text{def}}{=} \max_{x \in \Omega} |F_1(x) - F_2(x)| \tag{1.2}$$

where F_1 and F_2 are the respective cumulative distribution functions (cdf) of $\mathbf{p}_1$ and $\mathbf{p}_2$. Thus, the Kolmogorov distance is the ℓ_∞ distance between the cdf's; and $\mathrm{d_K}(\mathbf{p}_1, \mathbf{p}_2) \leq \mathrm{d_{TV}}(\mathbf{p}_1, \mathbf{p}_2) \in [0, 1]$.

We will also occasionally rely on the *Hellinger distance*, a third metric on $\Delta(\Omega)$ defined as

$$\mathrm{d_H}(\mathbf{p}_1, \mathbf{p}_2) = \sqrt{\frac{1}{2}\sum_{x\in\Omega}\left(\sqrt{\mathbf{p}_1(x)} - \sqrt{\mathbf{p}_2(x)}\right)^2} = \sqrt{1 - \sum_{x\in\Omega}\sqrt{\mathbf{p}_1(x)}\sqrt{\mathbf{p}_2(x)}} = \frac{1}{\sqrt{2}}\|\sqrt{\mathbf{p}_1} - \sqrt{\mathbf{p}_2}\|_2$$

which also takes values in $[0, 1]$. One particularly useful feature of the Hellinger distance is its close relation to total variation:

Fact 1.4.1 ([14, Corollary 2.39])**.** *For any probability distributions $\mathbf{p}_1$, $\mathbf{p}_2$ as above,*

$$\mathrm{d_{TV}}(\mathbf{p}_1, \mathbf{p}_2)^2 \leq \mathrm{d_H}(\mathbf{p}_1, \mathbf{p}_2)^2 \leq \mathrm{d_{TV}}(\mathbf{p}_1, \mathbf{p}_2) \tag{1.3}$$

For more on the Kolmogorov and Hellinger distances and their relation to total variation, we refer the reader to [42, Appendix C].

On several occasions we will use the *data processing inequality for variation distance*. This intuitive yet fundamental result says that for any two distributions $\mathbf{p}$, $\mathbf{p}'$, applying any (possibly randomized) function to both $\mathbf{p}$ and $\mathbf{p}'$ can never increase their statistical distance; see e.g. part (iv) of [147, Lemma 2] for a proof of this lemma.

Fact 1.4.2 (Data Processing Inequality for Total Variation Distance)**.** *Let $\mathbf{p}_1$, $\mathbf{p}_2$ be two distributions over a domain Ω. Fix any randomized function*[6] *F on Ω, and let $F(\mathbf{p}_1)$ be the distribution such that a draw from $F(\mathbf{p}_1)$ is obtained by drawing independently x from $\mathbf{p}_1$ and f from F and then outputting $f(x)$ (likewise for $F(\mathbf{p}_2)$). Then we have*

$$\mathrm{d_{TV}}(F(\mathbf{p}_1), F(\mathbf{p}_2)) \leq \mathrm{d_{TV}}(\mathbf{p}_1, \mathbf{p}_2).$$

Finally, we recall below a fundamental fact from probability theory that will be useful to us, the *Dvoretzky–Kiefer–Wolfowitz (DKW) inequality*. Informally, this result says that one can learn the cumulative distribution function of a distribution up to an additive error ε in ℓ_∞ distance, by taking only $O\big(1/\varepsilon^2\big)$ samples from it.

Theorem 1.4.3 ([91, 131])**.** *Let $\mathbf{p}$ be a distribution over $[n]$. Given m independent samples $x_1, \ldots, x_m$ from $\mathbf{p}$, define the empirical distribution $\hat{\mathbf{p}}$ as follows:*

$$\hat{\mathbf{p}}(i) \stackrel{\text{def}}{=} \frac{|\{\, j \in [m] \,:\, x_j = i \,\}|}{m}, \quad i \in [n].$$

Then, for all $\varepsilon > 0$, $\Pr[\,\mathrm{d_K}(\mathbf{p}, \hat{\mathbf{p}}) > \varepsilon\,] \leq 2e^{-2m\varepsilon^2}$, where the probability is taken over the samples.

In particular, setting $m = \Theta\Big(\frac{\log(1/\delta)}{\varepsilon^2}\Big)$ we get that $\mathrm{d_K}(\mathbf{p}, \hat{\mathbf{p}}) \leq \varepsilon$ with probability at least $1 - \delta$.

[6] Which can be seen as a distribution over functions over Ω.

Flattenings. For a distribution $\mathbf{p}$ and a partition of $[n]$ into intervals $\mathcal{I} = (I_1, \ldots, I_\ell)$, we define the *flattening of* $\mathbf{p}$ *with relation to* $\mathcal{I}$ as the distribution $\Psi_{\mathcal{I}}(\mathbf{p})$, where $\Psi_{\mathcal{I}}(\mathbf{p})(i) = \mathbf{p}(I_k)/|I_k|$ for all $k \in [\ell]$ and $i \in I_k$. A straightforward computation shows that such flattening cannot increase the distance between two distributions, i.e.,

$$\mathrm{d}_{\mathrm{TV}}(\Psi_{\mathcal{I}}(\mathbf{p}_1), \Psi_{\mathcal{I}}(\mathbf{p}_2)) \leq \mathrm{d}_{\mathrm{TV}}(\mathbf{p}_1, \mathbf{p}_2). \tag{1.4}$$

Proof of Eq. (1.4). Fix a partition $\mathcal{I}$ of $[n]$ into ℓ intervals $I_1, \ldots, I_\ell$, and let $\mathbf{p}_1$, $\mathbf{p}_2$ be two arbitrary distributions on $[n]$. Recall that $\Psi_{\mathcal{I}}(\mathbf{p})$ is the flattening of distribution $\mathbf{p}_j$ (with relation to the partition $\mathcal{I}$).

$$\begin{aligned}
2\mathrm{d}_{\mathrm{TV}}(\Psi_{\mathcal{I}}(\mathbf{p}_1), \Psi_{\mathcal{I}}(\mathbf{p}_2)) &= \sum_{i=1}^{n} |\Psi_{\mathcal{I}}(\mathbf{p}_1)(i) - \Psi_{\mathcal{I}}(\mathbf{p}_2)(i)| = \sum_{k=1}^{\ell} \sum_{i \in I_k} \left| \frac{\mathbf{p}_1(I_k)}{|I_k|} - \frac{\mathbf{p}_2(I_k)}{|I_k|} \right| \\
&= \sum_{k=1}^{\ell} |\mathbf{p}_1(I_k) - \mathbf{p}_2(I_k)| = \sum_{k=1}^{\ell} \left| \sum_{i \in I_k} (\mathbf{p}_1(i) - \mathbf{p}_2(i)) \right| \\
&\leq \sum_{k=1}^{\ell} \sum_{i \in I_k} |\mathbf{p}_1(i) - \mathbf{p}_2(i)| = \sum_{i=1}^{n} |\mathbf{p}_1(i) - \mathbf{p}_2(i)| = 2\mathrm{d}_{\mathrm{TV}}(\mathbf{p}_1, \mathbf{p}_2).
\end{aligned}$$

(we remark that Eq. (1.4) could also be obtained directly by applying the data processing inequality for total variation distance (Fact 1.4.2) to $\mathbf{p}_1$, $\mathbf{p}_2$, for the transformation $\Psi_{\mathcal{I}}(\cdot)$.) □

We state here a few facts about monotone distributions, namely that they admit a *succinct* approximation, itself monotone, close in total variation distance. This theorem, originally from [32], has recently been pivotal in several results on learning and testing k-modal distributions [63, 74].

Definition 1.4.4 (Birgé decomposition)**.** Given a parameter $\alpha > 0$, the corresponding (oblivious) *Birgé decomposition of* $[n]$ is the partition $\mathcal{I}_\alpha = (I_1, \ldots, I_\ell)$, where $\ell = O\left(\frac{\ln(\alpha n+1)}{\alpha}\right) = O\left(\frac{\log n}{\alpha}\right)$ and $|I_k| = \lfloor (1+\alpha)^k \rfloor$, $1 \leq k \leq \ell$.

Note that this partition consists of logarithmically many intervals *and crucially only depends on* n *and* ε (and not on any specific distribution $\mathbf{p}$): for this reason, we will often refer to it as the "oblivious" decomposition.

For a distribution $\mathbf{p}$ and parameter α, define $\Phi_\alpha(\mathbf{p})$ to be the "flattened" distribution with relation to the oblivious decomposition $\mathcal{I}_\alpha$, that is $\Phi_\alpha(\mathbf{p}) = \Psi_{\mathcal{I}_\alpha}(\mathbf{p})$. The next theorem states that every monotone distribution can be well-approximated by its flattening on the Birgé decomposition's intervals:

Theorem 1.4.5 ([32, 74])**.** *If* $\mathbf{p}$ *is monotone, then* $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \Phi_\alpha(\mathbf{p})) \leq \alpha$.

As a corollary, one can extend the theorem to distributions only promised to be *close* to monotone:

Corollary 1.4.6. *Suppose* $\mathbf{p}$ *is* ε*-close to monotone, and let* $\alpha > 0$*. Then* $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \Phi_\alpha(\mathbf{p})) \leq 2\varepsilon + \alpha$*. Furthermore,* $\Phi_\alpha(\mathbf{p})$ *is also* ε*-close to monotone.*

Proof of Corollary 1.4.6. Let $\mathbf{p}$ be ε-close to monotone, and $\mathbf{p}'$ be a monotone distribution such that $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{p}') =$

$\eta \leq \varepsilon$. By Eq. (1.4), we have

$$\mathrm{d}_{\mathrm{TV}}(\Phi_\alpha(\mathbf{p}), \Phi_\alpha(\mathbf{p}')) \leq \mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{p}') = \eta \tag{1.5}$$

proving the last part of the claim (since $\Phi_\alpha(\mathbf{p}')$ is easily seen to be monotone).
Now, by the triangle inequality,

$$\begin{aligned}\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \Phi_\alpha(\mathbf{p}')) &\leq \mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{p}') + \mathrm{d}_{\mathrm{TV}}(\mathbf{p}', \Phi_\alpha(\mathbf{p}')) + \mathrm{d}_{\mathrm{TV}}(\Phi_\alpha(\mathbf{p}'), \Phi_\alpha(\mathbf{p})) \\ &\leq \eta + \alpha + \eta \\ &\leq 2\varepsilon + \alpha\end{aligned}$$

where the last inequality uses the assumption on $\mathbf{p}'$ and Theorem 1.4.5 applied to it. □

We now restate a result of Batu et al. relating closeness to uniformity in ℓ_2 and ℓ_1 norms to "overall flatness" of the probability mass function, and which will be one of the ingredients of the proof of Theorem 2.1.1:

Lemma 1.4.7 ([22, 21])**.** *Let $\mathbf{p}$ be a distribution on a domain S. (a) If $\max_{i \in S} \mathbf{p}(i) \leq (1+\varepsilon) \min_{i \in S} \mathbf{p}(i)$, then $\|\mathbf{p}\|_2^2 \leq (1+\varepsilon^2)/|S|$. (b) If $\|\mathbf{p}\|_2^2 \leq (1+\varepsilon^2)/|S|$, then $\|\mathbf{p} - \mathbf{u}_S\|_1 \leq \varepsilon$.*

Some of our algorithms will need to check that condition (b) above holds. To do so, they rely on the following, which one can derive from the techniques in [82] and whose proof we defer to the appendix (p. 306):

Lemma 1.4.8 (Adapted from [82, Theorem 11])**.** *There exists an algorithm* CHECK-SMALL-ℓ_2 *which, given parameters $\varepsilon, \delta \in (0,1)$ and $c \cdot \sqrt{|I|}/\varepsilon^2 \log(1/\delta)$ independent samples from a distribution $\mathbf{p}$ over I (for some absolute constant $c > 0$), outputs either yes or no, and satisfies the following.*

- *If $\|\mathbf{p} - \mathbf{u}_I\|_2 > \varepsilon/\sqrt{|I|}$, then the algorithm outputs no with probability at least $1 - \delta$;*
- *If $\|\mathbf{p} - \mathbf{u}_I\|_2 \leq \varepsilon/2\sqrt{|I|}$, then the algorithm outputs yes with probability at least $1 - \delta$.*

Finally, recall the following well-known result on distinguishing biased coins (which can for instance be derived from Eq. (2.15) and (2.16) of [6]), that shall come in handy in proving our lower bounds:

Fact 1.4.9. *Let $p \in [\eta, 1-\eta]$ for some fixed constant $\eta > 0$, and suppose $m \leq \frac{c_\eta}{\varepsilon^2}$, with c_η a sufficiently small constant and $\varepsilon < \eta$. Then,*

$$\mathrm{d}_{\mathrm{TV}}(\mathrm{Bin}(m,p), \mathrm{Bin}(m,p+\varepsilon)) < \frac{1}{3}.$$

1.4.1 Tools from Analysis and Probability

We first give several variants of the Chernoff bounds (see e.g. [134, Chapter 4]), which we will use extensively in this thesis.

Theorem 1.4.10. *Let $Y_1, \ldots, Y_m$ be m independent random variables that take on values in $[0,1]$, where*

$\mathbb{E}[Y_i] = p_i$, *and* $\sum_{i=1}^m p_i = P$. *For any* $\gamma \in (0,1]$ *we have*

$$\textit{(additive bound)} \quad \Pr\left[\sum_{i=1}^m Y_i > P + \gamma m\right], \Pr\left[\sum_{i=1}^m Y_i < P - \gamma m\right] \leq \exp(-2\gamma^2 m) \tag{1.6}$$

$$\textit{(multiplicative bound)} \quad \Pr\left[\sum_{i=1}^m Y_i > (1+\gamma)P\right] < \exp(-\gamma^2 P/3) \tag{1.7}$$

and

$$\textit{(multiplicative bound)} \quad \Pr\left[\sum_{i=1}^m Y_i < (1-\gamma)P\right] < \exp(-\gamma^2 P/2). \tag{1.8}$$

The bound in Eq. (1.7) is derived from the following more general bound, which holds from any $\gamma > 0$*:*

$$\Pr\left[\sum_{i=1}^m Y_i > (1+\gamma)P\right] \leq \left(\frac{e^\gamma}{(1+\gamma)^{1+\gamma}}\right)^P , \tag{1.9}$$

and which also implies that for any $B > 2eP$*,*

$$\Pr\left[\sum_{i=1}^m Y_i > B\right] \leq 2^{-B} . \tag{1.10}$$

The following extension of the multiplicative bound is useful when we only have upper and/or lower bounds on P (see e.g. [89, Exercise 1.1]):

Claim 1.4.11. *In the setting of Theorem 1.4.10 suppose that* $P_L \leq P \leq P_H$*. Then for any* $\gamma \in (0,1]$*, we have*

$$\Pr\left[\sum_{i=1}^m Y_i > (1+\gamma)P_H\right] < \exp(-\gamma^2 P_H/3) \tag{1.11}$$

$$\Pr\left[\sum_{i=1}^m Y_i < (1-\gamma)P_L\right] < \exp(-\gamma^2 P_L/2) \tag{1.12}$$

We will also rely on the following corollary of Theorem 1.4.10:

Corollary 1.4.12. *Let* $0 \leq w_1, \ldots, w_m \in \mathbb{R}$ *be such that* $w_i \leq \kappa$ *for all* $i \in [m]$*, where* $\kappa \in (0,1]$*. Let* $X_1, \ldots, X_m$ *be i.i.d. Bernoulli random variables with* $\Pr[X_i = 1] = 1/2$ *for all* i*, and let* $X = \sum_{i=1}^m w_i X_i$ *and* $W = \sum_{i=1}^m w_i$*. For any* $\gamma \in (0,1]$*,*

$$\Pr\left[X > (1+\gamma)\frac{W}{2}\right] < \exp\left(-\gamma^2 \frac{W}{6\kappa}\right) \quad \textit{and} \quad \Pr\left[X < (1-\gamma)\frac{W}{2}\right] < \exp\left(-\gamma^2 \frac{W}{4\kappa}\right) ,$$

and for any $B > e \cdot W$*,*

$$\Pr[X > B] < 2^{-B/\kappa} .$$

Proof. Let $w'_i \stackrel{\text{def}}{=} w_i/\kappa$ (so that $w'_i \in [0,1]$), $W' \stackrel{\text{def}}{=} \sum_{i=1}^m w'_i = W/\kappa$, and for each $i \in [m]$ let $Y_i \stackrel{\text{def}}{=} w'_i X_i$, so that Y_i takes on values in $[0,1]$ and $\mathbb{E}[Y_i] = w'_i/2$. Let $X' = \sum_{i=1}^m w'_i X_i = \sum_{i=1}^m Y_i$, so that $\mathbb{E}[X'] =$

$W'/2$. By the definitions of W' and X' and by Eq. (1.7), for any $\gamma \in (0,1]$,

$$\Pr\left[X > (1+\gamma)\frac{W}{2} \right] = \Pr\left[X' > (1+\gamma)\frac{W'}{2} \right] < \exp\left(-\gamma^2 \frac{W'}{6}\right) = \exp\left(-\gamma^2 \frac{W}{6\kappa}\right),$$

and similarly by Eq. (1.8)

$$\Pr\left[X < (1-\gamma)\frac{W}{2} \right] < \exp\left(-\gamma^2 \frac{W}{4\kappa}\right) .$$

For $B > e \cdot W = 2e \cdot W/2$ we apply Eq. (1.10) and get

$$\Pr[\, X > B \,] = \Pr[\, X' > B/\kappa \,] < 2^{-B/\kappa},$$

as claimed. □

Next, we state a standard probabilistic result that some of our proofs will rely on, the Paley–Zygmund anticoncentration inequality:

Theorem 1.4.13 (Paley–Zygmund inequality)**.** *Let X be a non-negative random variable with finite variance. Then, for any $\theta \in [0,1]$,*

$$\Pr[\, X > \theta \mathbb{E}[X] \,] \geq (1-\theta)^2 \frac{\mathbb{E}[X]^2}{\mathbb{E}[X^2]}.$$

We also recall a classical inequality for sums of independent random variables, due to Bennett [38, Chapter 2]:

Theorem 1.4.14 (Bennett's inequality)**.** *Let $X = \sum_{i=1}^{n} X_i$, where $X_1, \ldots, X_n$ are independent random variables such that (i) $\mathbb{E}[X_i] = 0$ and (ii) $|X_i| \leq \alpha$ almost surely for all $1 \leq i \leq n$. Letting $\sigma^2 = \mathrm{Var}[X]$, we have, for every $t \geq 0$,*

$$\Pr[X > t] \leq \exp\left(-\frac{\mathrm{Var}[X]}{\alpha^2} \vartheta\left(\frac{\alpha t}{\mathrm{Var}[X]} \right) \right)$$

where $\vartheta(x) = (1+x)\ln(1+x) - x$.

We will also require the following version of the rearrangement inequality, due to Hardy and Littlewood (cf. for instance [25, Theorem 2.2]):

Theorem 1.4.15 (Hardy–Littlewood Inequality)**.** *Fix any $f, g \colon \mathbb{R} \to [0,\infty)$ such that $\lim_{\pm\infty} f = \lim_{\pm\infty} g = 0$. Then,*

$$\int_{\mathbb{R}} fg \leq \int_{\mathbb{R}} f^* g^*$$

where f^*, g^* denote the symmetric decreasing rearrangements of f, g respectively.

1.4.2 Discrete Fourier transform

For our SIIRV testing algorithm, we will need the following definition of the Fourier transform.

Definition 1.4.16 (Discrete Fourier Transform)**.** For $x \in \mathbb{R}$, we let $e(x) \stackrel{\text{def}}{=} \exp(-2i\pi x)$. The *Discrete Fourier Transform (DFT) modulo* M of a function $F\colon [\![n]\!] \to \mathbb{C}$ is the function $\widehat{F}\colon [\![M]\!] \to \mathbb{C}$ defined as

$$\widehat{F}(\xi) = \sum_{j=0}^{n-1} e\left(\frac{\xi j}{M}\right) F(j)$$

for $\xi \in [\![M]\!]$. The DFT modulo M of a distribution $\mathbf{p}$, $\widehat{\mathbf{p}}$, is then the DFT modulo M of its probability mass function (note that one can then equivalently see $\widehat{\mathbf{p}}(\xi)$ as the expectation $\widehat{\mathbf{p}}(\xi) = \mathbb{E}_{X\sim F}[e\left(\frac{\xi X}{M}\right)]$, for $\xi \in [\![M]\!]$).

The *inverse DFT modulo* M onto the range $[m, m+M-1]$ of $\widehat{F}\colon [\![M]\!] \to \mathbb{C}$, is the function $F\colon [m, m+M-1] \cap \mathbb{Z} \to \mathbb{C}$ defined by

$$F(j) = \frac{1}{M} \sum_{\xi=0}^{M-1} e\left(-\frac{\xi j}{M}\right) \widehat{F}(\xi),$$

for $j \in [m, m+M-1] \cap \mathbb{Z}$.

Note that the DFT (modulo M) is a linear operator; moreover, we recall the standard fact relating the norms of a function and of its Fourier transform, that we will use extensively:

Theorem 1.4.17 (Plancherel's Theorem)**.** *For $M \geq 1$ and $F, G\colon [\![n]\!] \to \mathbb{C}$, we have (i) $\sum_{j=0}^{n-1} F(j)\overline{G(j)} = \frac{1}{M}\sum_{\xi=0}^{M-1} \widehat{F}(\xi)\overline{\widehat{G}(\xi)}$; and (ii) $\|F\|_2 = \frac{1}{\sqrt{M}}\|\widehat{F}\|_2$, where $\widehat{F}, \widehat{G}$ are the DFT modulo M of F, G, respectively.*

(The latter equality is sometimes referred to as Parseval's theorem.) We also note that later, for our PMD testing result, we shall need the appropriate generalization of the Fourier transform to the multivariate setting. We leave this generalization to the corresponding section, Section 2.2.6.

1.5 Error-Correcting Codes.

For an alphabet Σ, we denote the projection of $x \in \Sigma^n$ to a subset of coordinates $I \subseteq [n]$ by $x|_I$. For $i \in [n]$, we write $x_i = x|_{\{i\}}$ to denote the projection to a singleton. We denote the *relative Hamming distance*, over alphabet Σ, between two strings $x \in \Sigma^n$ and $y \in \Sigma^n$ by $\mathrm{dist}(x, y) \stackrel{\text{def}}{=} |\{\, x_i \neq y_i \,:\, i \in [n] \,\}|/n$. Analogously to the distribution case, we say that x is *ε-close* to y if $\mathrm{dist}(x, y) \leq \varepsilon$, and otherwise we say that x is *ε-far* from y. Similarly, we denote the *relative Hamming distance* of x from a non-empty set $S \subseteq \Sigma^n$ by $\mathrm{dist}(x, S) \stackrel{\text{def}}{=} \min_{y\in S} \mathrm{dist}(x, y))$. If $\mathrm{dist}(x, S) \leq \varepsilon$, we say that x is *ε-close* to S, and otherwise we say that x is *ε-far* from S.

Let $k, n \in \mathbb{N}$, and let Σ be a finite alphabet. A *code* is a one-to-one function $C\colon \Sigma^k \to \Sigma^n$ that maps *messages* to *codewords*, where k and n are called the code's *dimension* and *block length*, respectively. The *rate* of the code, measuring the redundancy of the encoding, is defined to be $\rho \stackrel{\text{def}}{=} k/n$. We will sometime identify the code C with its image $C(\Sigma^k)$. In particular, we shall write $c \in C$ to indicate that there exists $x \in \{0,1\}^k$ such that $c = C(x)$, and say that c is a codeword of C. The *relative distance* of a code is the minimal relative distance between two codewords of C, and is denoted by $\delta \stackrel{\text{def}}{=} \min_{c\neq c'\in C}\{\mathrm{dist}(c, c')\}$.

We say that C is an *asymptotically good code* if it has constant rate and constant relative distance. We shall make an extensive use of asymptotically good codes that are *balanced*, that is, codes in which each codeword consists of the same number of 0's and 1's

Proposition 1.5.1 (Good Balanced Codes)**.** *For any constant $\delta \in [0, 1/3)$, there exists a good balanced code $C\colon \{0,1\}^k \to \{0,1\}^n$ with relative distance δ and constant rate. Namely, there exists a constant $\rho > 0$ such that the following holds.*

(i) Balance: $|C(x)| = \frac{n}{2}$ for all $x \in \{0,1\}^k$;

(ii) Relative distance: $\mathrm{dist}(C(x), C(y)) > \delta$ for all distinct $x, y \in \{0,1\}^k$;

(iii) Constant rate: $\frac{k}{n} \geq \rho$.

Proof. Fix any code C' with linear distance δ and constant rate (denoted ρ'). We transform $C'\colon \{0,1\}^k \to \{0,1\}^{n'}$ to a balanced code $C\colon \{0,1\}^k \to \{0,1\}^{2n'}$ by representing 0 and 1 as the balanced strings 01 and 10 (respectively). More accurately, we let $C(x) \stackrel{\text{def}}{=} C'(x) \odot \overline{C'(x)} \in \{0,1\}^n$ for all $x \in \{0,1\}^k$, where $\odot$ denotes the concatenation and $\bar{z}$ is the bitwise negation of z. It is immediate to check that this transformation preserves the distance, and that C is a balanced code with rate $\rho \stackrel{\text{def}}{=} 2\rho'$. □

On uniformity. For the sake of notation and clarity, throughout this dissertation we define all algorithms and objects non-uniformly. Namely, we fix the relevant parameter (typically $n \in \mathbb{N}$), and restrict ourselves to inputs or domains of size n (for instance, probability distributions over domain $[n]$). However, we still view it as a generic parameter and allow ourselves to write asymptotic expressions such as $O(n)$. Moreover, although our results are stated in terms of non-uniform algorithms, they can be extended to the uniform setting in a straightforward manner.

On the domain and parameters. Unless specified otherwise, Ω will hereafter by default be the n-element set $[n]$. When stating the results, the accuracy parameter $\varepsilon \in (0, 1]$ is to be understood as taking small values, either a fixed (small) constant or a quantity tending to 0 as $n \to \infty$; however, the actual parameter of interest will always be n, viewed as "going to infinity." Hence any dependence on n, no matter how mild, shall be considered as more expensive than any function of ε only.

Chapter 2

Testing Classes of Distributions: Upper Bounds from Swiss Army Knives

"Should we meet with a Jubjub, that desperate bird,
We shall need all our strength for the job!"

Lewis Carroll, *The Hunting of the Snark*

In this chapter, we focus on obtaining algorithmic upper bounds in distribution testing. Our goal, however, departs from most of the previous literature, in that it is not to solve a specific testing problem by coming up with a tailor-made algorithm for that task. We take instead a different path, and set out to provide *general* algorithms applicable across-the-board to a variety of problems *at once*. Marginally more formally, here is the objective we seek to address:

Problem 2.0.1. *Design general-purpose testing algorithms, that when applied to a property $\mathcal{P}$ would have (tight, or near-tight, or not absolutely laughable) sample complexity $q(\varepsilon, \tau)$ as long as $\mathcal{P}$ satisfies some "structural assumption" $\mathcal{S}_\tau$ parameterized by τ.*

For instance, one could think of $\mathcal{S}_\tau$ as "every $\mathbf{p} \in \mathcal{P}$ is close to some piecewise-constant distribution on $1/\tau$ intervals" (as is e.g. the case for monotone distributions, with $\tau = \frac{1}{\mathrm{poly}(\log n, 1/\varepsilon)}$ by Theorem 1.4.5); or, in another vein, "all $\mathbf{p} \in \mathcal{P}$ have $\|\mathbf{p}\|_{17/4} \leq \tau$" (technically, one *could* think of that one).

We make significant progress in this direction by providing two unified frameworks for the question of testing various properties of probability distributions. First, we describe in Section 2.1 a meta-algorithm to test membership in any distribution class, particularly well-suited to any class (including monotone, log-concave, t-modal, piecewise-polynomial, and Poisson Binomial distributions) which satisfies a "shape constraint." (Broadly speaking, whenever any distribution in the class is well-approximated, in a strong ℓ_2-type sense, by a piecewise-constant distribution on relatively few pieces.)

In Section 2.2, we detail our second general technique, based on an entirely different type of structural assumption. Namely, this approach now leverages purported properties of the *Fourier transform* of the distributions, and performs particularly well for those classes containing distributions with sparse Fourier transform – such as, for instance, the classes of Poisson Binomial distributions and SIIRVs.

Our two frameworks yield near-sample-optimal and computationally efficient testers for a wide range of distribution families; for most of these, we provide the first non-trivial tester in the literature.

2.1 The Shape Restrictions Knife

2.1.1 Introduction

In many situations, it is natural to assume that the data exhibits some simple structure because of known properties of the origin of the data, and in fact these assumptions are crucial in making the problem tractable. Such assumptions translate as constraints on the probability distribution – e.g., it is supposed to be Gaussian, or to meet a smoothness or "fat tail" condition (see e.g., [130, 114, 165]).

As a result, the problem of deciding whether a distribution possesses such a structural property has been widely investigated both in theory and practice, in the context of *shape restricted inference* [17, 164] and *model selection* [132]. Here, it is guaranteed or thought that the unknown distribution satisfies a shape constraint, such as having a monotone or log-concave probability density function [162, 13, 175, 79].

In this work, we consider this decision question from the Theoretical Computer Science viewpoint, namely in the context of distribution testing. We provide a unified framework for the question of testing various "shape restricted" properties of probability distributions – more specifically, we describe a generic technique to obtain upper bounds on the sample complexity of this question, which applies to a broad range of structured classes. Our technique yields sample near-optimal and computationally efficient testers for a wide range of distribution families. Conversely, we also develop a general approach to prove lower bounds on these sample complexities, and use it to derive tight or nearly tight bounds for many of these classes. (This lower bound approach will be covered in the next chapter, Section 3.1.)

Related work Batu et al. [19] initiated the study of efficient property testers for monotonicity and obtained (nearly) matching upper and lower bounds for this problem; while [2] later considered testing the class of Poisson Binomial Distributions, and settled the sample complexity of this problem (up to the precise dependence on ε). Indyk, Levi, and Rubinfeld [117], focusing on distributions that are piecewise constant on t intervals ("t-histograms") described a $\tilde{O}(\sqrt{tn}/\varepsilon^5)$-sample algorithm for testing membership in this class. Another body of work by [23], [19], and [74] shows how assumptions on the shape of the distributions can lead to significantly more efficient algorithms. They describe such improvements in the case of identity and closeness testing as well as for entropy estimation, under monotonicity or k-modality constraints. Specifically, Batu et al. show in [19] how to obtain a $O\big(\log^3 n/\varepsilon^3\big)$-sample tester for closeness in this setting, in stark contrast to the $\Omega\big(n^{2/3}\big)$ general lower bound. Daskalakis et al. [74] later gave $O(\sqrt{\log n})$ and $O(\log^{2/3} n)$-sample testing algorithms for testing respectively identity and closeness of monotone distributions, and obtained similar results for k-modal distributions. Finally, we briefly mention two related results, due respectively to [23] and [63]. The first one states that for the task of getting a multiplicative *estimate* of the entropy of a distribution, assuming monotonicity enables exponential savings in sample complexity – $O\big(\log^6 n\big)$, instead of $\Omega(n^c)$ for the general case. The second describes how to test if an unknown k-modal distribution is in fact monotone, using only $O\big(k/\varepsilon^2\big)$ samples. Note that the latter line of work differs from ours in that it *presupposes* the distributions satisfy some structural property, and uses this knowledge to test

something else about the distribution; while we are given *a priori* arbitrary distributions, and must *check* whether the structural property holds. Except for the properties of monotonicity and being a PBD, nothing was previously known on testing the shape restricted properties that we study.

Moreover, for the specific problems of identity and closeness testing,[1] recent results of [82, 81] describe a general algorithm which applies to a large range of shape or structural constraints, and yields optimal identity testers for classes of distributions that satisfy them. We observe that while the question they answer can be cast as a specialized instance of membership testing, our results are incomparable to theirs, both because of the distinction above (testing *with* versus testing *for* structure) and as the structural assumptions they rely on are fundamentally different from ours.

Concurrent and followup work Independently and concurrently to the initial conference publication of this work, Acharya, Daskalakis, and Kamath [3] obtained a sample near-optimal efficient algorithm for testing log-concavity, as well as sample-optimal algorithms for testing the classes of monotone, unimodal, and monotone hazard rate distributions (along with matching lower bounds on the sample complexity of these tasks). Their work builds on ideas from [2] and their techniques are orthogonal to ours: namely, while at some level both works follow a "testing-by-learning" paradigm, theirs rely on first learning in the (more stringent) χ^2 distance, then applying a testing algorithm which is robust to some amount of noise (i.e., tolerant testing) in this χ^2 sense (as opposed to noise in an ℓ_1 sense, which is known to be impossible without a near-linear number of samples [167]).

Subsequent to the publication of the conference version of this work, [43] improved on both [117] and our results for the specific class of t-histograms, providing nearly tight upper and lower bounds on testing membership in this class. Specifically, it obtains an upper bound of $\tilde{O}(\sqrt{n}/\varepsilon^2 + t/\varepsilon^3)$, complemented with an $\Omega(\sqrt{n}/\varepsilon^2 + t/(\varepsilon \log t))$ lower bound on the sample complexity.

Building on our work, Fischer, Lachish, and Vasudev recently generalized in [96] our approach and algorithm to the *conditional sampling model* of [54, 49], obtaining analogues of our testing results in this different setting of distribution testing where the algorithm is allowed to condition the samples it receives on subsets of the domain of its choosing. In the "standard" sampling setting, [96] additionally provides an alternative to the first subroutine of our testing algorithm: this yields a simpler and non-recursive algorithm, with a factor $\log n$ shaved off at the price of a worse dependency on the distance parameter ε. (Namely, their sample complexity is dominated by $O(\sqrt{nL}\log^2(1/\varepsilon)/\varepsilon^5)$, to be compared to the $O(\sqrt{nL}\log n/\varepsilon^3)$ term of Theorem 2.1.15.)

2.1.1.1 Results and Techniques

A natural way to tackle our membership testing problem would be to first learn the unknown distribution $\mathbf{p}$ *as if* it satisfied the property, before checking if the hypothesis obtained is indeed both close to the original

[1] Recall that the identity testing problem asks, given the explicit description of a distribution $\mathbf{p}^*$ and sample access to an unknown distribution $\mathbf{p}$, to decide whether $\mathbf{p}$ is equal to $\mathbf{p}^*$ or far from it; while in closeness testing both distributions to compare are unknown.

distribution and to the property. Taking advantage of the purported structure, the first step could presumably be conducted with a small number of samples; things break down, however, in the second step. Indeed, most approximation results leading to the improved learning algorithms one would apply in the first stage only provide very weak guarantees, that is in the ℓ_1 sense only. For this reason, they lack the robustness that would be required for the second part, where it becomes necessary to perform *tolerant* testing between the hypothesis and $\mathbf{p}$ – a task that would then entail a number of samples almost linear in the domain size. To overcome this difficulty, we need to move away from these global ℓ_1 closeness results and instead work with stronger requirements, this time in ℓ_2 norm.

At the core of our approach is an idea of Batu et al. [19], which show that monotone distributions can be well-approximated (in a certain technical sense) by piecewise constant densities on a suitable interval partition of the domain; and leverage this fact to reduce monotonicity testing to uniformity testing on each interval of this partition. While the argument of [19] is tailored specifically for the setting of monotonicity testing, we are able to abstract the key ingredients, and obtain a generic membership tester that applies to a wide range of distribution families. In more detail, we provide a testing algorithm which applies to any class of distributions which admit succinct approximate decompositions – that is, each distribution in the class can be well-approximated (in a strong ℓ_2 sense) by piecewise constant densities on a small number of intervals (we hereafter refer to this approximation property, formally defined in Definition 2.1.13, as (Succinctness); and extend the notation to apply to any *class* $\mathcal{C}$ of distributions for which all $\mathbf{p} \in \mathcal{C}$ satisfy (Succinctness)). Crucially, the algorithm does not care about *how* these decompositions can be obtained: for the purpose of testing these structural properties we only need to establish their *existence*. Specific examples are given in the corollaries below. Informally, our main algorithmic result, informally stated (see Theorem 2.1.15 for a detailed formal statement), is as follows:

Theorem 2.1.1 (Main Theorem)**.** *There exists an algorithm* TESTSPLITTABLE *which, given sampling access to an unknown distribution* $\mathbf{p}$ *over* $[n]$ *and parameter* $\varepsilon \in (0, 1]$*, can distinguish with probability* $2/3$ *between (a)* $\mathbf{p} \in \mathcal{P}$ *versus (b)* $\ell_1(\mathbf{p}, \mathcal{P}) > \varepsilon$*, for* any *property* $\mathcal{P}$ *that satisfies the above natural structural criterion* (Succinctness).

Moreover, we remark that for many such properties this algorithm is computationally efficient, and its sample complexity is optimal (up to logarithmic factors and the exact dependence on ε). We instantiate this result to obtain "out-of-the-box" *computationally efficient* testers for several classes of distributions, by showing that they satisfy the premise of our theorem (the definition of these classes is given in Section 1.3):

Corollary 2.1.2. *The algorithm* TESTSPLITTABLE *can test the classes of monotone, unimodal, log-concave, concave, convex, and monotone hazard rate (MHR) distributions, with* $\tilde{O}\left(\sqrt{n}/\varepsilon^{7/2}\right)$ *samples.*

Corollary 2.1.3. *The algorithm* TESTSPLITTABLE *can test the class of* t*-modal distributions, with* $\tilde{O}\left(\sqrt{tn}/\varepsilon^{7/2}\right)$ *samples.*

Corollary 2.1.4. *The algorithm* TESTSPLITTABLE *can test the classes of t-histograms and t-piecewise degree-d distributions, with $\tilde{O}\left(\sqrt{tn}/\varepsilon^3\right)$ and $\tilde{O}\left(\sqrt{t(d+1)n}/\varepsilon^{7/2} + t(d+1)/\varepsilon^3\right)$ samples respectively.*

Corollary 2.1.5. *The algorithm* TESTSPLITTABLE *can test the classes of Binomial and Poisson Binomial Distributions, with $\tilde{O}\left(n^{1/4}/\varepsilon^{7/2}\right)$ samples.*

Class	Upperbound	Lowerbound
Monotone	$\tilde{O}\left(\frac{\sqrt{n}}{\varepsilon^6}\right)$ [19], $\tilde{O}\left(\frac{\sqrt{n}}{\varepsilon^{7/2}}\right)$ (Corollary 2.1.2), $O\left(\frac{\sqrt{n}}{\varepsilon^2}\right)$ [3]$^{(\ddagger)}$	$\Omega\left(\frac{\sqrt{n}}{\varepsilon^2}\right)$ [19], $\Omega\left(\frac{\sqrt{n}}{\varepsilon^2}\right)$ (Corollary 2.1.6)
Unimodal	$\tilde{O}\left(\frac{\sqrt{n}}{\varepsilon^{7/2}}\right)$ (Corollary 2.1.2), $O\left(\frac{\sqrt{n}}{\varepsilon^2}\right)$ [3]$^{(\ddagger)}$	$\Omega\left(\frac{\sqrt{n}}{\varepsilon^2}\right)$ (Corollary 2.1.6)
t-modal	$\tilde{O}\left(\frac{\sqrt{tn}}{\varepsilon^{7/2}}\right)$ (Corollary 2.1.3)	$\Omega\left(\frac{\sqrt{n}}{\varepsilon^2}\right)$ (Corollary 2.1.6)
Concave, convex	$\tilde{O}\left(\frac{\sqrt{n}}{\varepsilon^{7/2}}\right)$ (Corollary 2.1.2)	$\Omega\left(\frac{\sqrt{n}}{\varepsilon^2}\right)$ (Corollary 2.1.6)
Log-concave	$\tilde{O}\left(\frac{\sqrt{n}}{\varepsilon^{7/2}}\right)$ (Corollary 2.1.2), $O\left(\frac{\sqrt{n}}{\varepsilon^2}\right)$ [3]$^{(\ddagger)}$	$\Omega\left(\frac{\sqrt{n}}{\varepsilon^2}\right)$ (Corollary 2.1.6)
Monotone Hazard Rate (MHR)	$\tilde{O}\left(\frac{\sqrt{n}}{\varepsilon^{7/2}}\right)$ (Corollary 2.1.2), $O\left(\frac{\sqrt{n}}{\varepsilon^2}\right)$ [3]$^{(\ddagger)}$	$\Omega\left(\frac{\sqrt{n}}{\varepsilon^2}\right)$ (Corollary 2.1.6)
Binomial, Poisson Binomial (PBD)	$\tilde{O}\left(\frac{n^{1/4}}{\varepsilon^2} + \frac{1}{\varepsilon^6}\right)$ [2], $\tilde{O}\left(\frac{n^{1/4}}{\varepsilon^{7/2}}\right)$ (Corollary 2.1.5)	$\Omega\left(\frac{n^{1/4}}{\varepsilon^2}\right)$ ([2], Corollary 2.1.7)
t-histograms	$\tilde{O}\left(\frac{\sqrt{tn}}{\varepsilon^5}\right)$ [117], $\tilde{O}\left(\frac{\sqrt{n}}{\varepsilon^2} + \frac{t}{\varepsilon^3}\right)$ [43]$^{(\ddagger)}$, $\tilde{O}\left(\frac{\sqrt{tn}}{\varepsilon^3}\right)$ (Corollary 2.1.4)	$\Omega\left(\sqrt{tn}\right)$ (for $t \leq \frac{1}{\varepsilon}$) [117], $\Omega\left(\frac{\sqrt{n}}{\varepsilon^2} + \frac{t}{\varepsilon}\right)$ [43]$^{(\ddagger)}$, $\Omega\left(\frac{\sqrt{n}}{\varepsilon^2}\right)$ (Corollary 2.1.6)
t-piecewise degree-d	$\tilde{O}\left(\frac{\sqrt{t(d+1)n}}{\varepsilon^{7/2}} + \frac{t(d+1)}{\varepsilon^3}\right)$ (Corollary 2.1.4)	$\Omega\left(\frac{\sqrt{n}}{\varepsilon^2}\right)$ (Corollary 2.1.6)
(n,k)-SIIRV	$O\left(\frac{k^{1/2}n^{1/4}}{\varepsilon^2}\log^{1/4}\frac{1}{\varepsilon}\right) + \tilde{O}\left(\frac{k^2}{\varepsilon^2}\right)$ (Theorem 2.2.1, [45])$^{(\ddagger)}$	$\Omega\left(\frac{k^{1/2}n^{1/4}}{\varepsilon^2}\right)$ (Corollary 2.1.8)

Table 2.1: Summary of results obtained *via* our first general class testing framework (Theorem 2.1.1). The corresponding lower bounds stated in this table originate from the technique covered in the next chapter (specifically, Section 3.1); while the symbol $(\ddagger)$ indicates a result independent of or subsequent to our work.

We remark that the aforementioned sample upper bounds are information-theoretically near-optimal in the domain size n (up to logarithmic factors). See Table 2.1 and the following chapter for the corresponding lower bounds. We did not attempt to optimize the dependence on the parameter ε, though a more careful analysis might lead to such improvements.

We stress that prior to our work, no non-trivial testing bound was known for most of these classes – specifically, our nearly-tight bounds for t-modal with $t > 1$, log-concave, concave, convex, MHR, and piecewise polynomial distributions are new. Moreover, although a few of our applications were known in the literature (the $\tilde{O}\left(\sqrt{n}/\varepsilon^6\right)$ upper and $\Omega\left(\sqrt{n}/\varepsilon^2\right)$ lower bounds on testing monotonicity can be found in [19], while the $\Theta\left(n^{1/4}\right)$ sample complexity of testing PBDs was recently given[2] in [2], and the task of testing t-histograms is considered in [117]), the crux here is that we are able to derive them in a *unified* way, by applying the same generic algorithm to all these different distribution families. We note that our upper bound for t-histograms (Corollary 2.1.4) also significantly improves on the previous $\tilde{O}\left(\sqrt{tn}/\varepsilon^5\right)$-sample tester with regard to the dependence on the proximity parameter ε. In addition to its generality, our framework yields

much cleaner and conceptually simpler proofs of the upper and lower bounds from [2].

Lower Bounds To complement our upper bounds, we also give a generic framework for proving lower bounds against testing classes of distributions. While this framework will be the focus of the next chapter, we state here some of the results it enables us to derive for specific structured distribution families; the reader is referred to Section 3.1 (and specifically Theorem 3.1.1) for the details and formal statement of our reduction-based lower bound theorem.

Corollary 2.1.6. *Testing log-concavity, convexity, concavity, MHR, unimodality, t-modality, t-histograms, and t-piecewise degree-d distributions each require $\Omega\left(\sqrt{n}/\varepsilon^2\right)$ samples (the last three for $t = o(\sqrt{n})$ and $t(d+1) = o(\sqrt{n})$, respectively), for any $\varepsilon \geq 1/n^{O(1)}$.*[3]

Corollary 2.1.7. *Testing the classes of Binomial and Poisson Binomial Distributions each require $\Omega\left(n^{1/4}/\varepsilon^2\right)$ samples, for any $\varepsilon \geq 1/n^{O(1)}$.*

Corollary 2.1.8. *There exist absolute constants $c > 0$ and $\varepsilon_0 > 0$ such that testing the class of (n,k)-SIIRV distributions requires $\Omega\left(k^{1/2}n^{1/4}/\varepsilon^2\right)$ samples, for any $k = o(n^c)$ and $1/n^{O(1)} \leq \varepsilon \leq \varepsilon_0$.*

Tolerant Testing Using our techniques, we also establish nearly–tight upper and lower bounds on tolerant testing for shape restrictions. Similarly, our upper and lower bounds are matching as a function of the domain size. More specifically, we give a simple generic upper bound approach (namely, a learning followed by tolerant testing algorithm). Our tolerant testing lower bounds follow the same reduction-based approach as in the non-tolerant case, and will be covered in the next chapter, Chapter 3. In more detail, our results are as follows (see Section 2.1.5 for the upper bounds, and further down the road Section 3.1 for the lower bounds):

Corollary 2.1.9. *Tolerant testing of log-concavity, convexity, concavity, MHR, unimodality, and t-modality can be performed with $O\left(\frac{1}{(\varepsilon_2-\varepsilon_1)^2}\frac{n}{\log n}\right)$ samples, for $\varepsilon_2 \geq C\varepsilon_1$ (where $C > 2$ is an absolute constant).*

Corollary 2.1.10. *Tolerant testing of the classes of Binomial and Poisson Binomial Distributions can be performed with $O\left(\frac{1}{(\varepsilon_2-\varepsilon_1)^2}\frac{\sqrt{n\log(1/\varepsilon_1)}}{\log n}\right)$ samples, for $\varepsilon_2 \geq C\varepsilon_1$ (where $C > 2$ is an absolute constant).*

Corollary 2.1.11. *Tolerant testing of log-concavity, convexity, concavity, MHR, unimodality, and t-modality each require $\Omega\left(\frac{1}{(\varepsilon_2-\varepsilon_1)}\frac{n}{\log n}\right)$ samples (the latter for $t = o(n)$).*

---

[2]For the sample complexity of testing monotonicity, [19] originally states an $\tilde{O}\left(\sqrt{n}/\varepsilon^4\right)$ upper bound, but the proof seems to only result in an $\tilde{O}\left(\sqrt{n}/\varepsilon^6\right)$ bound. Regarding the class of PBDs, Acharya and Daskalakis [2] obtain an $n^{1/4}\cdot\tilde{O}\left(1/\varepsilon^2\right)+\tilde{O}\left(1/\varepsilon^6\right)$ sample complexity, to be compared with our $\tilde{O}\left(n^{1/4}/\varepsilon^{7/2}\right)+O\left(\log^4 n/\varepsilon^4\right)$ upper bound; and this is complemented (also in [2]) by an $\Omega\left(n^{1/4}/\varepsilon^2\right)$ lower bound.

[3]Here, the restriction on ε should be read as "for each of these distribution classes, there exists an absolute constant $c > 0$ (which may depend on the corresponding class) such that the result applies for every $\varepsilon \geq \frac{1}{n^c}$."

Corollary 2.1.12. *Tolerant testing of the classes of Binomial and Poisson Binomial Distributions each require* $\Omega\left(\frac{1}{(\varepsilon_2-\varepsilon_1)}\frac{\sqrt{n}}{\log n}\right)$ *samples.*

On the scope of our results We point out that our main theorem is likely to apply to many other classes of structured distributions, due to the mild structural assumptions it requires. However, we did not attempt here to be comprehensive; but rather to illustrate the generality of our approach. Moreover, for all properties considered in this paper the generic upper and lower bounds we derive through our methods turn out to be optimal up to at most polylogarithmic factors (with regard to the support size). The reader is referred to Table 2.1 for a summary of our results and related work.

2.1.1.2 Organization of the Section

We begin by establishing our main result, the proof of Theorem 2.1.1 (our general testing algorithm), in Section 2.1.2. In Section 2.1.3, we establish the necessary structural theorems for each class of distributions considered, enabling us to derive the upper bounds of Table 2.1. Section 2.1.4 introduces a slight modification of our algorithm which yields stronger testing results for classes of distributions with small effective support, and use it to derive Corollary 2.1.5, our upper bound for Poisson Binomial distributions. (The details of our lower bound methodology and of its applications to the classes of Table 2.1, however, are deferred to Chapter 3.) Finally, Section 3.1.1 is concerned with the extension of this methodology to *tolerant* testing, of which Section 2.1.5 describes a generic upper bound counterpart.

2.1.2 The General Algorithm

In this section, we obtain our main result, restated below:

Theorem 2.1.1 (Main Theorem)**.** *There exists an algorithm* TESTSPLITTABLE *which, given sampling access to an unknown distribution* $\mathbf{p}$ *over* $[n]$ *and parameter* $\varepsilon \in (0,1]$*, can distinguish with probability* $2/3$ *between (a)* $\mathbf{p} \in \mathcal{P}$ *versus (b)* $\ell_1(\mathbf{p},\mathcal{P}) > \varepsilon$*, for* any *property* $\mathcal{P}$ *that satisfies the above natural structural criterion* (Succinctness).

Intuition Before diving into the proof of this theorem, we first provide a high-level description of the argument. The algorithm proceeds in 3 stages: the first, the *decomposition step*, attempts to recursively construct a partition of the domain in a small number of intervals, with a very strong guarantee. If the decomposition succeeds, then the unknown distribution $\mathbf{p}$ will be close (in ℓ_1 distance) to its "flattening" on the partition; while if it fails (too many intervals have to be created), this serves as evidence that $\mathbf{p}$ does not belong to the class and we can reject. The second stage, the *approximation step*, then learns this flattening of the distribution – which can be done with few samples since by construction we do not have many intervals. The last stage is purely computational, the *projection step*: where we verify that the flattening we have learned is indeed close to the class $\mathcal{C}$. If all three stages succeed, then by the triangle inequality it must be the case

that $\mathbf{p}$ is close to $\mathcal{C}$; and by the structural assumption on the class, if $\mathbf{p} \in \mathcal{C}$ then it will admit succinct enough partitions, and all three stages will go through.

Turning to the proof, we start by defining formally the "structural criterion" we shall rely on, before describing the algorithm at the heart of our result in Section 2.1.2.1. (We note that a modification of this algorithm will be described in Section 2.1.4, and will allow us to derive Corollary 2.1.5.)

Definition 2.1.13 (Decompositions)**.** Let $\gamma, \zeta > 0$ and $L = L(\gamma, \zeta, n) \geq 1$. A class of distributions $\mathcal{C}$ on $[n]$ is said to be (γ, ζ, L)-*decomposable* if for every $\mathbf{p} \in \mathcal{C}$ there exists $\ell \leq L$ and a partition $\mathcal{I}(\gamma, \zeta, \mathbf{p}) = (I_1, \ldots, I_\ell)$ of the interval $[1, n]$ such that, for all $j \in [\ell]$, one of the following holds:

(i) $\mathbf{p}(I_j) \leq \frac{\zeta}{L}$; or

(ii) $\max_{i \in I_j} \mathbf{p}(i) \leq (1+\gamma) \cdot \min_{i \in I_j} \mathbf{p}(i)$.

Further, if $\mathcal{I}(\gamma, \zeta, \mathbf{p})$ is *dyadic* (i.e., each I_k is of the form $[j \cdot 2^i + 1, (j+1) \cdot 2^i]$ for some integers i, j, corresponding to the leaves of a recursive bisection of $[n]$), then $\mathcal{C}$ is said to be (γ, ζ, L)-*splittable*.

Lemma 2.1.14. *If $\mathcal{C}$ is (γ, ζ, L)-decomposable, then it is (γ, ζ, L')-splittable for $L'(\gamma, \zeta, n) = O(\log n) \cdot L(\gamma, \frac{\zeta}{2(\log n+1)}, n)$.*

Proof. We will begin by proving a claim that for every partition $\mathcal{I} = \{I_1, I_2, ..., I_L\}$ of the interval $[1, n]$ into L intervals, there exists a refinement of that partition which consists of at most $L \cdot O(\log n)$ dyadic intervals. So, it suffices to prove that every interval $[a, b] \subseteq [1, n]$ can be partitioned in at most $O(\log n)$ dyadic intervals. Indeed, let ℓ be the largest integer such that $2^\ell \leq \frac{b-a}{2}$ and let m be the smallest integer such that $m \cdot 2^\ell \geq a$. If follows that $m \cdot 2^\ell \leq a + \frac{b-a}{2} = \frac{a+b}{2}$ and $(m+1) \cdot 2^\ell \leq b$. So, the interval $I = [m \cdot 2^\ell + 1, (m+1) \cdot 2^\ell]$ is fully contained in $[a, b]$ and has size at least $\frac{b-a}{4}$.

We will also use the fact that, for every $\ell' \leq \ell$,

$$m \cdot 2^\ell = m \cdot 2^{\ell - \ell'} \cdot 2^{\ell'} = m' \cdot 2^{\ell'} \tag{2.1}$$

Now consider the following procedure: Starting from right (resp. left) side of the interval I, we add the largest interval which is adjacent to it and fully contained in $[a, b]$ and recurse until we cover the whole interval $[(m+1) \cdot 2^\ell + 1, b]$ (resp. $[a, m \cdot 2^\ell]$). Clearly, at the end of this procedure, the whole interval $[a, b]$ is covered by dyadic intervals. It remains to show that the procedure takes $O(\log n)$ steps. Indeed, using Eq. (2.1), we can see that at least half of the remaining left or right interval is covered in each step (except maybe for the first 2 steps where it is at least a quarter). Thus, the procedure will take at most $2 \log n + 2 = O(\log n)$ steps in total. From the above, we can see that each of the L intervals of the partition $\mathcal{I}$ can be covered with $O(\log n)$ dyadic intervals, which completes the proof of the claim.

In order to complete the proof of the lemma, notice that the two conditions in Definition 2.1.13 are closed under taking subsets: so that the second is immediately verified, while for the first we have that for any of the "new" intervals I that $\mathbf{p}(I) \leq \frac{\zeta}{L} \leq \frac{\zeta \cdot (2 \log n + 2)}{L'}$. □

2.1.2.1 The algorithm

Theorem 2.1.1, and with it Corollary 2.1.2, Corollary 2.1.3, and Corollary 2.1.4 will follow from the theorem below, combined with the structural theorems from Section 2.1.3:

Theorem 2.1.15. *Let $\mathcal{C}$ be a class of distributions over $[n]$ for which the following holds.*

1. *$\mathcal{C}$ is $(\gamma, \zeta, L(\gamma, \zeta, n))$-splittable;*
2. *there exists a procedure* PROJECTIONDIST$_{\mathcal{C}}$ *which, given as input a parameter $\alpha \in (0, 1)$ and the explicit description of a distribution $\mathbf{p}$ over $[n]$, returns* yes *if the distance $\ell_1(\mathbf{p}, \mathcal{C})$ to $\mathcal{C}$ is at most $\alpha/10$, and* no *if $\ell_1(\mathbf{p}, \mathcal{C}) \geq 9\alpha/10$ (and either* yes *or* no *otherwise).*

Then the algorithm TESTSPLITTABLE *(Algorithm 1) is a $O\left(\max\left(\sqrt{nL}\log n/\varepsilon^3, L/\varepsilon^2\right)\right)$-sample tester for $\mathcal{C}$, for $L = L(O(\varepsilon), O(\varepsilon), n)$. (Moreover, if* PROJECTIONDIST$_{\mathcal{C}}$ *is computationally efficient, then so is* TESTSPLITTABLE.*)*

Algorithm 1 TESTSPLITTABLE

Require: Domain I (interval), sample access to $\mathbf{p}$ over I; subroutine PROJECTIONDIST$_{\mathcal{C}}$
Input: Parameters ε and "splittable" function $L_{\mathcal{C}}(\cdot, \cdot, \cdot)$.

SETTING UP
  Define $\gamma \stackrel{\text{def}}{=} \frac{\varepsilon}{80}$, $L \stackrel{\text{def}}{=} L_{\mathcal{C}}(\gamma, \gamma, |I|)$, $\kappa \stackrel{\text{def}}{=} \frac{\varepsilon}{160L}$, $\delta \stackrel{\text{def}}{=} \frac{1}{10L}$; and $c > 0$ be as in Lemma 1.4.8.
  Set $m \stackrel{\text{def}}{=} C \cdot \max\left(\frac{1}{\kappa}, \frac{\sqrt{L|I|}}{\varepsilon^3}\right) \cdot \log|I| = \tilde{O}\left(\frac{\sqrt{L|I|}}{\varepsilon^3} + \frac{L}{\varepsilon}\right)$ ▷ C is an absolute constant.
  Obtain a sequence $\mathbf{s}$ of m independent samples from $\mathbf{p}$. ▷ For any $J \subseteq I$, let m_J be the number of samples falling in J.

DECOMPOSITION
  while $m_I \geq \max\left(c \cdot \frac{\sqrt{|I|}}{\varepsilon^2}\log\frac{1}{\delta}, \kappa m\right)$ and at most L splits have been performed **do**
    Run CHECK-SMALL-ℓ_2 (from Lemma 1.4.8) with parameters $\frac{\varepsilon}{40}$ and δ, using the samples of $\mathbf{s}$ belonging to I.
    if CHECK-SMALL-ℓ_2 outputs no **then**
      Bisect I, and recurse on both halves (using the same samples).
    end if
  end while
  if more than L splits have been performed **then**
    return reject
  else
    Let $\mathcal{I} \stackrel{\text{def}}{=} (I_1, \ldots, I_\ell)$ be the partition of $[n]$ from the leaves of the recursion. ▷ $\ell \leq L$.
  end if

APPROXIMATION
  Learn the flattening $\Phi(\mathbf{p}, \mathcal{I})$ of $\mathbf{p}$ to ℓ_1 error $\frac{\varepsilon}{20}$ (with probability $1/10$), using $O\left(\ell/\varepsilon^2\right)$ new samples. Let $\tilde{\mathbf{p}}$ be the resulting hypothesis. ▷ $\tilde{\mathbf{p}}$ is an ℓ-histogram.

OFFLINE CHECK
  return accept if and only if PROJECTIONDIST$_{\mathcal{C}}(\varepsilon, \tilde{\mathbf{p}})$ returns yes. ▷ No samples needed.

2.1.2.2 Proof of Theorem 2.1.15

We now give the proof of our main result (Theorem 2.1.15), first analyzing the sample complexity of Algorithm 1 before arguing its correctness. For the latter, we will need the following simple fact from [117], restated below:

Fact 2.1.16 ([117, Fact 1])**.** *Let* $\mathbf{p}$ *be a distribution over* $[n]$*, and* $\delta \in (0,1]$*. Given* $m \geq C \cdot \frac{\log \frac{n}{\delta}}{\eta}$ *independent samples from* $\mathbf{p}$ *(for some absolute constant* $C > 0$*), with probability at least* $1-\delta$ *we have that, for every interval* $I \subseteq [n]$*:*

(i) if $\mathbf{p}(I) \geq \frac{\eta}{4}$*, then* $\frac{\mathbf{p}(I)}{2} \leq \frac{m_I}{m} \leq \frac{3\mathbf{p}(I)}{2}$*;*

(ii) if $\frac{m_I}{m} \geq \frac{\eta}{2}$*, then* $\mathbf{p}(I) > \frac{\eta}{4}$*;*

(iii) if $\frac{m_I}{m} < \frac{\eta}{2}$*, then* $\mathbf{p}(I) < \eta$*;*

where $m_I \stackrel{\text{def}}{=} |\{\, j \in [m] : \, x_j \in I \,\}|$ *is the number of the samples falling into* I*.*

2.1.2.3 Sample complexity.

The sample complexity is immediate, and comes from Steps 4 and 20. The total number of samples is

$$m + O\left(\frac{\ell}{\varepsilon^2}\right) = O\left(\frac{\sqrt{|I| \cdot L}}{\varepsilon^3} \log |I| + \frac{L}{\varepsilon} \log |I| + \frac{L}{\varepsilon^2}\right) = O\left(\frac{\sqrt{|I| \cdot L}}{\varepsilon^3} \log |I| + \frac{L}{\varepsilon^2}\right).$$

2.1.2.4 Correctness.

Say an interval I considered during the execution of the "Decomposition" step is *heavy* if m_I is big enough on Step 7, and *light* otherwise; and let $\mathcal{H}$ and $\mathcal{L}$ denote the sets of heavy and light intervals respectively. By choice of m, we can assume that with probability at least $9/10$ the guarantees of Fact 2.1.16 hold simultaneously for all intervals considered. We hereafter condition on this event.

We first argue that if the algorithm does not reject in Step 13, then with probability at least $9/10$ we have $\|\mathbf{p} - \Phi(\mathbf{p}, \mathcal{I})\|_1 \leq \varepsilon/20$ (where $\Phi(\mathbf{p}, \mathcal{I})$ denotes the flattening of $\mathbf{p}$ over the partition $\mathcal{I}$). Indeed, we can write

$$\begin{aligned}\|\mathbf{p} - \Phi(\mathbf{p}, \mathcal{I})\|_1 &= \sum_{k\colon I_k \in \mathcal{L}} \mathbf{p}(I_k) \cdot \|\mathbf{p}_{I_k} - \mathbf{u}_{I_k}\|_1 + \sum_{k\colon I_k \in \mathcal{H}} \mathbf{p}(I_k) \cdot \|\mathbf{p}_{I_k} - \mathbf{u}_{I_k}\|_1 \\ &\leq 2 \sum_{k\colon I_k \in \mathcal{L}} \mathbf{p}(I_k) + \sum_{k\colon I_k \in \mathcal{H}} \mathbf{p}(I_k) \cdot \|\mathbf{p}_{I_k} - \mathbf{u}_{I_k}\|_1 .\end{aligned}$$

Let us bound the two terms separately.

- If $I' \in \mathcal{H}$, then by our choice of threshold we can apply Lemma 1.4.8 with $\delta = \frac{1}{10L}$; conditioning on all of the (at most L) events happening, which overall fails with probability at most $1/10$ by a union bound, we get

$$\|\mathbf{p}_{I'}\|_2^2 = \|\mathbf{p}_{I'} - \mathbf{u}_{I'}\|_2^2 + \frac{1}{|I'|} \leq \left(1 + \frac{\varepsilon^2}{1600}\right) \frac{1}{|I'|}$$

as CHECK-SMALL-ℓ_2 returned yes; and by Lemma 1.4.7 this implies $\|\mathbf{p}_{I'} - \mathbf{u}_{I'}\|_1 \le \varepsilon/40$.

- If $I' \in \mathcal{L}$, then we claim that $\mathbf{p}(I') \le \max(\kappa, 2c \cdot \frac{\sqrt{|I'|}}{m\varepsilon^2} \log \frac{1}{\delta})$. Clearly, this is true if $\mathbf{p}(I') \le \kappa$, so it only remains to show that $\mathbf{p}(I') \le 2c \cdot \frac{\sqrt{|I'|}}{m\varepsilon^2} \log \frac{1}{\delta}$. But this follows from Fact 2.1.16 (i), as if we had $\mathbf{p}(I') > 2c \cdot \frac{\sqrt{|I'|}}{m\varepsilon^2} \log \frac{1}{\delta}$ then $m_{I'}$ would have been big enough, and $I' \notin \mathcal{L}$. Overall,

$$\sum_{I' \in \mathcal{L}} \mathbf{p}(I') \le \sum_{I' \in \mathcal{L}} \left(\kappa + 2c \cdot \frac{\sqrt{|I'|}}{m\varepsilon^2} \log \frac{1}{\delta} \right) \le L\kappa + 2 \sum_{I' \in \mathcal{L}} c \cdot \frac{\sqrt{|I'|}}{m\varepsilon^2} \log \frac{1}{\delta} \le \frac{\varepsilon}{160} \left(1 + \sum_{I' \in \mathcal{L}} \sqrt{\frac{|I'|}{|I|\, L}} \right) \le \frac{\varepsilon}{80}$$

for a sufficiently big choice of constant $C > 0$ in the definition of m; where we first used that $|\mathcal{L}| \le L$, and then that $\sum_{I' \in \mathcal{L}} \sqrt{\frac{|I'|}{|I|}} \le \sqrt{L}$ by Jensen's inequality.

Putting it together, this yields

$$\|\mathbf{p} - \Phi(\mathbf{p}, \mathcal{I})\|_1 \le 2 \cdot \frac{\varepsilon}{80} + \frac{\varepsilon}{40} \sum_{I' \in \mathcal{H}} \mathbf{p}(I_k) \le \varepsilon/40 + \varepsilon/40 = \varepsilon/20.$$

Soundness. By contrapositive, we argue that if the test returns accept, then (with probability at least $2/3$) $\mathbf{p}$ is ε-close to $\mathcal{C}$. Indeed, conditioning on $\tilde{\mathbf{p}}$ being $\varepsilon/20$-close to $\Phi(\mathbf{p}, \mathcal{I})$, we get by the triangle inequality that

$$\begin{aligned} \|\mathbf{p} - \mathcal{C}\|_1 &\le \|\mathbf{p} - \Phi(\mathbf{p}, \mathcal{I})\|_1 + \|\Phi(\mathbf{p}, \mathcal{I}) - \tilde{\mathbf{p}}\|_1 + \operatorname{dist}(\tilde{\mathbf{p}}, \mathcal{C}) \\ &\le \frac{\varepsilon}{20} + \frac{\varepsilon}{20} + \frac{9\varepsilon}{10} = \varepsilon. \end{aligned}$$

Overall, this happens except with probability at most $1/10 + 1/10 + 1/10 < 1/3$.

Completeness. Assume $\mathbf{p} \in \mathcal{C}$. Then the choice of of γ and L ensures the existence of a good dyadic partition $\mathcal{I}(\gamma, \gamma, \mathbf{p})$ in the sense of Definition 2.1.13. For any I in this partition for which (i) holds ($\mathbf{p}(I) \le \frac{\gamma}{L} < \frac{\kappa}{2}$), I will have $\frac{m_I}{m} < \kappa$ and be kept as a "light leaf" (this by contrapositive of Fact 2.1.16 (ii)). For the other ones, (ii) holds: let I be one of these (at most L) intervals.

- If m_I is too small on Step 7, then I is kept as "light leaf."
- Otherwise, then by our choice of constants we can use Lemma 1.4.7 and apply Lemma 1.4.8 with $\delta = \frac{1}{10L}$; conditioning on all of the (at most L) events happening, which overall fails with probability at most $1/10$ by a union bound, CHECK-SMALL-ℓ_2 will output yes, as

$$\|\mathbf{p}_I - \mathbf{u}_I\|_2^2 = \|\mathbf{p}_I\|_2^2 - \frac{1}{|I|} \le \left(1 + \frac{\varepsilon^2}{6400} \right) \frac{1}{|I|} - \frac{1}{|I|} = \frac{\varepsilon^2}{6400\,|I|}$$

  and I is kept as "flat leaf."

Therefore, as $\mathcal{I}(\gamma, \gamma, \mathbf{p})$ is dyadic the DECOMPOSITION stage is guaranteed to stop within at most L splits (in the worst case, it goes on until $\mathcal{I}(\gamma, \gamma, \mathbf{p})$ is considered, at which point it succeeds).[4] Thus Step 13 passes, and the algorithm reaches the APPROXIMATION stage. By the foregoing discussion, this

implies $\Phi(\mathbf{p}, \mathcal{I})$ is $\varepsilon/20$-close to $\mathbf{p}$ (and hence to $\mathcal{C}$); $\tilde{\mathbf{p}}$ is then (except with probability at most $1/10$) $(\frac{\varepsilon}{20} + \frac{\varepsilon}{20} = \frac{\varepsilon}{10})$-close to $\mathcal{C}$, and the algorithm returns accept.

2.1.3 Structural Theorems

In this section, we show that a wide range of natural distribution families are succinctly decomposable, and provide efficient projection algorithms for each class.

2.1.3.1 Existence of Structural Decompositions

Theorem 2.1.17 (Monotonicity)**.** *For all $\gamma, \zeta > 0$, the class $\mathcal{M}_n$ of monotone distributions on $[n]$ is (γ, ζ, L)-splittable for $L \stackrel{\text{def}}{=} O\left(\frac{\log^2 \frac{n}{\zeta}}{\gamma}\right)$.*

Note that this proof can already be found in [19, Theorem 10], interwoven with the analysis of their algorithm. For the sake of being self-contained, we reproduce the structural part of their argument, removing its algorithmic aspects:

Proof of Theorem 2.1.17. We define the $\mathcal{I}$ recursively as follows: $\mathcal{I}^{(0)} = ([1, n])$, and for $j \geq 0$ the partition $\mathcal{I}^{(j+1)}$ is obtained from $\mathcal{I}^{(j)} = (I_1^{(j)}, \ldots, I_{\ell_j}^{(j)})$ by going over the $I_i^{(j)} = [a_i^{(j)}, b_i^{(j)}]$ in order, and:

(a) if $\mathbf{p}(I_i^{(j)}) \leq \frac{\zeta}{L}$, then $I_i^{(j)}$ is added as element of $\mathcal{I}^{(j+1)}$ ("marked as leaf");

(b) else, if $\mathbf{p}(a_i^{(j)}) \leq (1+\gamma)\mathbf{p}(b_i^{(j)})$, then $I_i^{(j)}$ is added as element of $\mathcal{I}^{(j+1)}$ ("marked as leaf");

(c) otherwise, bisect $I^{(j)}$ in $I_{\mathrm{L}}^{(j)}$, $I_{\mathrm{R}}^{(j)}$ (with $\left|I_{\mathrm{L}}^{(j)}\right| = \lceil |I^{(j)}|/2 \rceil$) and add both $I_{\mathrm{L}}^{(j)}$ and $I_{\mathrm{R}}^{(j)}$ as elements of $\mathcal{I}^{(j+1)}$.

and repeat until convergence (that is, whenever the last item is not applied for any of the intervals). Clearly, this process is well-defined, and will eventually terminate (as $(\ell_j)_j$ is a non-decreasing sequence of natural numbers, upper bounded by n). Let $\mathcal{I} = (I_1, \ldots, I_\ell)$ (with $I_i = [a_i, a_{i+1})$) be its outcome, so that the I_i's are consecutive intervals all satisfying either (a) or (b). As (b) clearly implies (ii), we only need to show that $\ell \leq L$; for this purpose, we shall leverage as in [19] the fact that $\mathbf{p}$ is monotone to bound the number of recursion steps.

The recursion above defines a complete binary tree (with the leaves being the intervals satisfying (a) or (b), and the internal nodes the other ones). Let t be the number of recursion steps the process goes through before converging to $\mathcal{I}$ (height of the tree); as mentioned above, we have $t \leq \log n$ (as we start with an interval of size n, and the length is halved at each step.). Observe further that if at any point an interval $I_i^{(j)} = [a_i^{(j)}, b_i^{(j)}]$

[4] In more detail, we want to argue that if $\mathbf{p}$ is in the class, then a decomposition with at most L pieces is found by the algorithm. Since there *is* a dyadic decomposition with at most L pieces (namely, $\mathcal{I}(\gamma, \gamma, \mathbf{p}) = (I_1, \ldots, I_t)$), it suffices to argue that the algorithm will never split one of the I_j's (as every single I_j will eventually be considered by the recursive binary splitting, unless the algorithm stopped recursing in this "path" before even considering I_j, which is even better). But this is the case by the above argument, which ensures each such I_j will be recognized as satisfying one of the two conditions for "good decomposition" (being either close to uniform in ℓ_2 distance, or having very little mass).

has $\mathbf{p}(a_i^{(j)}) \leq \frac{\zeta}{nL}$, then it immediately (as well as all the $I_k^{(j)}$'s for $k \geq i$ by monotonicity) satisfies (a) and is no longer split ("becomes a leaf"). So at any $j \leq t$, the number of intervals i_j for which neither (a) nor (b) holds must satisfy

$$1 \geq \mathbf{p}(a_1^{(j)}) > (1+\gamma)\mathbf{p}(a_2^{(j)}) > (1+\gamma)^2\mathbf{p}(a_3^{(j)}) > \cdots > (1+\gamma)^{i_j-1}\mathbf{p}(a_{i_j}^{(j)}) \geq (1+\gamma)^{i_j-1}\frac{\zeta}{nL}$$

where a_k denotes the beginning of the k-th interval (again we use monotonicity to argue that the extrema were reached at the ends of each interval), so that $i_j \leq 1 + \frac{\log \frac{nL}{\zeta}}{\log(1+\gamma)}$. In particular, the total number of internal nodes is then

$$\sum_{i=1}^{t} i_j \leq t \cdot \left(1 + \frac{\log \frac{nL}{\zeta}}{\log(1+\gamma)}\right) \leq \frac{2\log^2 \frac{n}{\zeta}}{\log(1+\gamma)} \leq L\,.$$

This implies the same bound on the number of leaves ℓ. □

Corollary 2.1.18 (Unimodality). *For all $\gamma, \zeta > 0$, the class $\mathcal{M}_{n,1}$ of unimodal distributions on $[n]$ is (γ, ζ, L)-decomposable for $L \stackrel{\text{def}}{=} O\left(\frac{\log^2 \frac{n}{\zeta}}{\gamma}\right)$.*

Proof. For any $\mathbf{p} \in \mathcal{M}_{n,1}$, $[n]$ can be partitioned in two intervals I, J such that $\mathbf{p}_I$, $\mathbf{p}_J$ are either monotone non-increasing or non-decreasing. Applying Theorem 2.1.17 to $\mathbf{p}_I$ and $\mathbf{p}_J$ and taking the union of both partitions yields a (no longer necessarily dyadic) partition of $[n]$. □

The same argument yields an analogous statement for t-modal distributions:

Corollary 2.1.19 (t-modality). *For any $t \geq 1$ and all $\gamma, \zeta > 0$, the class $\mathcal{M}_{n,t}$ of t-modal distributions on $[n]$ is (γ, ζ, L)-decomposable for $L \stackrel{\text{def}}{=} O\left(\frac{t\log^2 \frac{n}{\zeta}}{\gamma}\right)$.*

Corollary 2.1.20 (Log-concavity, concavity and convexity). *For all $\gamma, \zeta > 0$, the classes $\mathcal{LCV}_n$, $\mathcal{K}_n^-$ and $\mathcal{K}_n^+$ of log-concave, concave and convex distributions on $[n]$ are (γ, ζ, L)-decomposable for $L \stackrel{\text{def}}{=} O\left(\frac{\log^2 \frac{n}{\zeta}}{\gamma}\right)$.*

Proof. This is directly implied by Corollary 2.1.18, recalling that log-concave, concave and convex distributions are unimodal. □

Theorem 2.1.21 (Monotone Hazard Rate). *For all $\gamma, \zeta > 0$, the class $\mathcal{MHR}_n$ of MHR distributions on $[n]$ is (γ, ζ, L)-decomposable for $L \stackrel{\text{def}}{=} O\left(\frac{\log \frac{n}{\zeta}}{\gamma}\right)$.*

Proof. This follows from adapting the proof of [56], which establishes that every MHR distribution can be approximated in ℓ_1 distance by a $O(\log(n/\varepsilon)/\varepsilon)$-histogram. For completeness, we reproduce their argument, suitably modified to our purposes, in Section 2.1.6. □

Theorem 2.1.22 (Piecewise Polynomials). *For all $\gamma, \zeta > 0$, $t, d \geq 0$, the class $\mathcal{P}_{n,t,d}$ of t-piecewise degree-d distributions on $[n]$ is (γ, ζ, L)-decomposable for $L \stackrel{\text{def}}{=} O\left(\frac{t(d+1)}{\gamma}\log^2 \frac{n}{\zeta}\right)$. (Moreover, for the class of t-histograms $\mathcal{H}_{n,t}$ ($d = 0$) one can take $L = t$.)*

Proof. The last part of the statement is obvious, so we focus on the first claim. Observing that each of the t pieces of a distribution $\mathbf{p} \in \mathcal{P}_{n,t,d}$ can be subdivided in at most $d+1$ intervals on which $\mathbf{p}$ is monotone (being degree-d polynomial on each such piece), we obtain a partition of $[n]$ into at most $t(d+1)$ intervals. $\mathbf{p}$ being monotone on each of them, we can apply an argument almost identical to that of Theorem 2.1.17 to argue that each interval can be further split into $O(\log^2 n/\gamma)$ subintervals, yielding a good decomposition with $O(t(d+1)\log^2(n/\zeta)/\gamma)$ pieces. □

2.1.3.2 Projection Step: computing the distances

This section contains details of the distance estimation procedures for these classes, required in the last stage of Algorithm 1. (Note that some of these results are phrased in terms of distance approximation, as estimating the distance $\ell_1(\mathbf{p}, \mathcal{C})$ to sufficient accuracy in particular yields an algorithm for this stage.)

We focus in this section on achieving the sample complexities stated in Corollary 2.1.2, Corollary 2.1.3, and Corollary 2.1.4 – that is, our procedures will not require any additional sample from the distribution. While almost all the distance estimation procedures we give in this section are efficient, running in time polynomial in all the parameters or even with only a polylogarithmic dependence on n, there are two exceptions – namely, the procedures for monotone hazard rate (Lemma 2.1.25) and log-concave (Lemma 2.1.26) distributions. We *do* describe computationally efficient procedures for these two cases as well in Section 2.1.3.2, at a modest additive cost in the sample complexity (that is, these more efficient procedures *will* require some additional samples from the distribution).

Lemma 2.1.23 (Monotonicity [19, Lemma 8])**.** *There exists a procedure* PROJECTIONDIST$_{\mathcal{M}_n}$ *that, on input n as well as the full (succinct) specification of an ℓ-histogram $\mathbf{p}$ on $[n]$, computes the (exact) distance $\ell_1(\mathbf{p}, \mathcal{M}_n)$ in time* $\mathrm{poly}(\ell)$.

A straightforward modification of the algorithm above (e.g., by adapting the underlying linear program to take as input the location $m \in [\ell]$ of the mode of the distribution; then trying all ℓ possibilities, running the subroutine ℓ times and picking the minimum value) results in a similar claim for unimodal distributions:

Lemma 2.1.24 (Unimodality)**.** *There exists a procedure* PROJECTIONDIST$_{\mathcal{M}_{n,1}}$ *that, on input n as well as the full (succinct) specification of an ℓ-histogram $\mathbf{p}$ on $[n]$, computes the (exact) distance $\ell_1(\mathbf{p}, \mathcal{M}_{n,1})$ in time* $\mathrm{poly}(\ell)$.

A similar result can easily be obtained for the class of t-modal distributions as well, with a $\mathrm{poly}(\ell, t)$-time algorithm based on a combination of dynamic and linear programming. Analogous statements hold for the classes of concave and convex distributions $\mathcal{K}_n^+, \mathcal{K}_n^-$, also based on linear programming (specifically, on running $O(n^2)$ different linear programs – one for each possible support $[a, b] \subseteq [n]$ – and taking the minimum over them).

Lemma 2.1.25 (MHR)**.** *There exists a (non-efficient) procedure* PROJECTIONDIST$_{\mathcal{MHR}_n}$ *that, on input n,*

ε, as well as the full specification of a distribution $\mathbf{p}$ on $[n]$, distinguishes between $\ell_1(\mathbf{p}, \mathcal{MHR}_n) \leq \varepsilon$ and $\ell_1(\mathbf{p}, \mathcal{MHR}_n) > 2\varepsilon$ in time $2^{\tilde{O}_\varepsilon(n)}$.

Lemma 2.1.26 (Log-concavity)**.** *There exists a (non-efficient) procedure* PROJECTIONDIST$_{\mathcal{LCV}_n}$ *that, on input n, ε, as well as the full specification of a distribution $\mathbf{p}$ on $[n]$, distinguishes between $\ell_1(\mathbf{p}, \mathcal{LCV}_n) \leq \varepsilon$ and $\ell_1(\mathbf{p}, \mathcal{LCV}_n) > 2\varepsilon$ in time $2^{\tilde{O}_\varepsilon(n)}$.*

Proof of Lemma 2.1.25 and Lemma 2.1.26. We here give a naive algorithm for these two problems, based on an exhaustive search over a (huge) ε-cover $\mathcal{S}$ of distributions over $[n]$. Essentially, $\mathcal{S}$ contains all possible distributions whose probabilities $p_1, \ldots, p_n$ are of the form $j\varepsilon/n$, for $j \in \{0, \ldots, n/\varepsilon\}$ (so that $|\mathcal{S}| = O((n/\varepsilon)^n)$). It is not hard to see that this indeed defines an ε-cover of the set of all distributions, and moreover that it can be computed in time $\mathrm{poly}(|\mathcal{S}|)$. To approximate the distance from an explicit distribution $\mathbf{p}$ to the class $\mathcal{C}$ (either $\mathcal{MHR}_n$ or $\mathcal{LCV}_n$), it is enough to go over every element S of $\mathcal{S}$, checking (this time, efficiently) if $\|S - \mathbf{p}\|_1 \leq \varepsilon$ and if there is a distribution $P \in \mathcal{C}$ close to S (this time, pointwise, that is $|P(i) - S(i)| \leq \varepsilon/n$ for all i) – which also implies $\|S - P\|_1 \leq \varepsilon$ and thus $\|P - \mathbf{p}\|_1 \leq 2\varepsilon$. The test for pointwise closeness can be done by checking feasibility of a linear program with variables corresponding to the logarithm of probabilities, i.e. $x_i \equiv \ln P(i)$. Indeed, this formulation allows to rephrase the log-concave and MHR constraints as linear constraints, and pointwise approximation is simply enforcing that $\ln(S(i) - \varepsilon/n) \leq x_i \leq \ln(S(i) + \varepsilon/n)$ for all i. At the end of this enumeration, the procedure accepts if and only if for some S both $\|S - \mathbf{p}\|_1 \leq \varepsilon$ and the corresponding linear program was feasible. □

Lemma 2.1.27 (Piecewise Polynomials)**.** *There exists a procedure* PROJECTIONDIST$_{\mathcal{P}_{n,t,d}}$ *that, on input n as well as the full specification of an ℓ-histogram $\mathbf{p}$ on $[n]$, computes an approximation Δ of the distance $\ell_1(\mathbf{p}, \mathcal{P}_{n,t,d})$ such that $\ell_1(\mathbf{p}, \mathcal{P}_{n,t,d}) \leq \Delta \leq 3\ell_1(\mathbf{p}, \mathcal{P}_{n,t,d}) + \varepsilon$, and runs in time $O\big(n^3\big) \cdot \mathrm{poly}(\ell, t, d, \frac{1}{\varepsilon})$.*

Moreover, for the special case of t-histograms ($d = 0$) there exists a procedure PROJECTIONDIST$_{\mathcal{H}_{n,t}}$*, which, given inputs as above, computes an approximation Δ of the distance $\ell_1(\mathbf{p}, \mathcal{H}_{n,t})$ such that $\ell_1(\mathbf{p}, \mathcal{H}_{n,t}) \leq \Delta \leq 4\ell_1(\mathbf{p}, \mathcal{H}_{n,t}) + \varepsilon$, and runs in time $\mathrm{poly}(\ell, t, \frac{1}{\varepsilon})$, independent of n.*

Proof. We begin with PROJECTIONDIST$_{\mathcal{H}_{n,t}}$. Fix any distribution $\mathbf{p}$ on $[n]$. Given any explicit partition of $[n]$ into intervals $\mathcal{I} = (I_1, \ldots, I_t)$, one can easily show that $\|\mathbf{p} - \Phi(\mathbf{p}, \mathcal{I})\|_1 \leq 2\mathrm{OPT}_{\mathcal{I}}$, where $\mathrm{OPT}_{\mathcal{I}}$ is the optimal distance of $\mathbf{p}$ to any histogram on $\mathcal{I}$ (recall that we write $\Phi(\mathbf{p}, \mathcal{I})$ for the flattening of $\mathbf{p}$ over the partition $\mathcal{I}$). To get a 2-approximation of $\ell_1(\mathbf{p}, \mathcal{H}_{n,t})$, it thus suffices to find the minimum, over all possible partitionings $\mathcal{I}$ of $[n]$ into t intervals, of the quantity $\|\mathbf{p} - \Phi(\mathbf{p}, \mathcal{I})\|_1$ (which itself can be computed in time $T = O(\min(t\ell, n))$). By a simple dynamic programming approach, this can be performed in time $O\big(tn^2 \cdot T\big)$. The quadratic dependence on n, which follows from allowing the endpoints of the t intervals to be at any point of the domain, is however far from optimal and can be reduced to $(t/\varepsilon)^2$, as we show below.

For $\eta > 0$, define an *η-granular decomposition* of a distribution $\mathbf{p}$ over $[n]$ to be a partition of $[n]$ into $s = O(1/\eta)$ intervals $J_1, \ldots, J_s$ such that each interval J_i is either a singleton or satisfies $\mathbf{p}(J_i) \leq \eta$. (Note that if $\mathbf{p}$ is a known ℓ-histogram, one can compute an η-granular decomposition of $\mathbf{p}$ in time $O(\ell/\eta)$ in a

greedy fashion.)

Claim 2.1.28. *Let $\mathbf{p}$ be a distribution over $[n]$, and $\mathcal{J} = (J_1, \ldots, J_s)$ be an η-granular decomposition of $\mathbf{p}$ (with $s \geq t$). Then, there exists a partition of $[n]$ into t intervals $\mathcal{I} = (I_1, \ldots, I_t)$ and a t-histogram H on $\mathcal{I}$ such that $\|\mathbf{p} - H\|_1 \leq 2\ell_1(\mathbf{p}, \mathcal{H}_{n,t}) + 2t\eta$, and $\mathcal{I}$ is a coarsening of $\mathcal{J}$.*

Before proving it, we describe how this will enable us to get the desired time complexity for PROJECTIONDIST$_{\mathcal{H}_{n,t}}$. Phrased differently, the claim above allows us to run our dynamic program using the $O(1/\eta)$ endpoints of the $O(1/\eta)$ instead of the n points of the domain, paying only an additive error $O(t\eta)$. Setting $\eta = \frac{\varepsilon}{4t}$, the guarantee for PROJECTIONDIST$_{\mathcal{H}_{n,t}}$ follows.

Proof of Claim 2.1.28. Let $\mathcal{J} = (J_1, \ldots, J_s)$ be an η-granular decomposition of $\mathbf{p}$, and $H^* \in \mathcal{H}_{n,t}$ be a histogram achieving OPT $= \ell_1(\mathbf{p}, \mathcal{H}_{n,t})$. Denote further by $\mathcal{I}^* = (I_1^*, \ldots, I_t^*)$ the partition of $[n]$ corresponding to H^*. Consider now the $r \leq t$ endpoints of the I_i^*'s that do not fall on one of the endpoints of the J_i's: let $J_{i_1}, \ldots, J_{i_r}$ be the respective intervals in which they fall (in particular, these cannot be singleton intervals), and $S = \cup_{j=1}^r J_{i_j}$ their union. By definition of η-granularity, $\mathbf{p}(S) \leq t\eta$, and it follows that $H^*(S) \leq t\eta + \frac{1}{2}$OPT. We define H from H^* in two stages: first, we obtain a (sub)distribution H' by modifying H^* on S, setting for each $x \in J_{i_j}$ the value of H to be the minimum value (among the two options) that H^* takes on J_{i_j}. H' is thus a t-histogram, and the endpoints of its intervals are endpoints of $\mathcal{J}$ as wished; but it may not sum to one. However, by construction we have that $H'([n]) \geq 1 - H^*(S) \geq 1 - t\eta - \frac{1}{2}$OPT. Using this, we can finally define our t-histogram distribution H as the renormalization of H'. It is easy to check that H is a valid t-histogram on a coarsening of $\mathcal{J}$, and

$$\|\mathbf{p} - H\|_1 \leq \|\mathbf{p} - H'\|_1 + (1 - H'([n])) \leq \|\mathbf{p} - H^*\|_1 + \|H^* - H'\|_1 + t\eta + \frac{1}{2}\text{OPT} \leq 2\text{OPT} + 2t\eta$$

as stated. □

Turning now to PROJECTIONDIST$_{\mathcal{P}_{n,t,d}}$, we apply the same initial dynamic programming approach, which will result on a running time of $O\big(n^2 t \cdot T\big)$, where T is the time required to estimate (to sufficient accuracy) the distance of a given (sub)distribution over an interval I onto the space $\mathcal{P}_{n,d}$ of degree-d polynomials. Specifically, we will invoke the following result, adapted from [55] to our setting:

Theorem 2.1.29. *Let p be an ℓ-histogram over $[-1, 1)$. There is an algorithm* PROJECTSINGLEPOLY(d, η) *which runs in time* $\mathrm{poly}(\ell, d + 1, 1/\eta)$*, and outputs a degree-$d$ polynomial q which defines a pdf over $[-1, 1)$ such that $\|p - q\|_1 \leq 3\ell_1(p, \mathcal{P}_{n,d}) + O(\eta)$.*

The proof of this modification of [55, Theorem 9] is deferred to Section 2.1.7. Applying it as a blackbox with η set to $O(\varepsilon/t)$ and noting that computing the ℓ_1 distance to our explicit distribution on a given interval of the degree-d polynomial returned incurs an additional $O(n)$ factor, we obtain the claimed guarantee and running time. □

Computationally Efficient Procedures for Log-concave and MHR Distributions We now describe how to obtain *efficient* testing for the classes $\mathcal{LCV}_n$ and $\mathcal{MHR}_n$ – that is, how to obtain polynomial-time distance estimation procedures for these two classes, unlike the ones described in the previous section. At a very high-level, the idea is in both cases to write down a linear program on variables related *logarithmically* to the probabilities we are searching, as enforcing the log-concave and MHR constraints on these new variables can be done linearly. The catch now becomes the ℓ_1 objective function (and, to a lesser extent, the fact that the probabilities must sum to one), now highly non-linear.

The first insight is to leverage the structure of log-concave (resp. monotone hazard rate) distributions to express this objective as slightly stronger constraints, specifically pointwise $(1 \pm \varepsilon)$-multiplicative closeness, much easier to enforce in our "logarithmic formulation." Even so, doing this naively fails, essentially because of a too weak distance guarantee between our explicit histogram $\hat{\mathbf{p}}$ and the unknown distribution we are trying to find: in the completeness case, we are only promised ε-closeness in ℓ_1, while we would also require good additive pointwise closeness of the order ε^2 or ε^3.

The second insight is thus to observe that we "almost" have this for free: indeed, if we do not reject in the first stage of the testing algorithm, we do obtain an explicit k-histogram $\hat{\mathbf{p}}$ with the guarantee that $\mathbf{p}$ is ε-close to the distribution P to test. However, we *also* implicitly have another distribution $\hat{\mathbf{p}}'$ that is $\sqrt{\varepsilon/k}$-close to P *in Kolmogorov distance*: as in the recursive descent we take enough samples to use the DKW inequality (Theorem 1.4.3) with this parameter, i.e. an additive overhead of $O(k/\varepsilon)$ samples (on top of the $\tilde{O}(\sqrt{kn}/\varepsilon^{7/2})$). If we are willing to increase this overhead by just a small amount, that is to take $\tilde{O}\big(\max(k/\varepsilon, 1/\varepsilon^4)\big)$, we can guarantee that $\hat{\mathbf{p}}'$ be also $\tilde{O}\big(\varepsilon^2\big)$-close to P in Kolmogorov distance.

Combining these ideas yield the following distance estimation lemmas:

Lemma 2.1.30 (Monotone Hazard Rate)**.** *There exists a procedure* $\textsc{ProjectionDist}^*_{\mathcal{MHR}_n}$ *that, on input n as well as the full specification of a k-histogram distribution $\mathbf{p}$ on $[n]$ and of an ℓ-histogram distribution $\mathbf{p}'$ on $[n]$, runs in time $\mathrm{poly}(n, 1/\varepsilon)$, and satisfies the following.*

- *If there is $P \in \mathcal{MHR}_n$ such that $\|\mathbf{p} - P\|_1 \leq \varepsilon$ and $\mathrm{d_K}(\mathbf{p}', P) \leq \varepsilon^3$, then the procedure returns* yes*;*
- *If $\ell_1(\mathbf{p}, \mathcal{MHR}_n) > 100\varepsilon$, then the procedure returns* no*.*

Lemma 2.1.31 (Log-concavity)**.** *There exists a procedure* $\textsc{ProjectionDist}^*_{\mathcal{LCV}_n}$ *that, on input n as well as the full specifications of a k-histogram distribution $\mathbf{p}$ on $[n]$ and an ℓ-histogram distribution $\mathbf{p}'$ on $[n]$, runs in time $\mathrm{poly}(n, k, \ell, 1/\varepsilon)$, and satisfies the following.*

- *If there is $P \in \mathcal{LCV}_n$ such that $\|\mathbf{p} - P\|_1 \leq \varepsilon$ and $\mathrm{d_K}(\mathbf{p}', P) \leq \frac{\varepsilon^2}{\log^2(1/\varepsilon)}$, then the procedure returns* yes*;*
- *If $\ell_1(\mathbf{p}, \mathcal{LCV}_n) \geq 100\varepsilon$, then the procedure returns* no*.*

The proofs of these two lemmas are quite technical and deferred to Section 2.1.7. With these in hand, a simple modification of our main algorithm (specifically, setting $m = \tilde{O}(\max(\sqrt{L\,|I|}/\varepsilon^3, L/\varepsilon^2, 1/\varepsilon^c))$ for c

either 4 or 6 instead of $\tilde{O}(\max(\sqrt{L\,|I|}/\varepsilon^3, L/\varepsilon^2))$, to get the desired Kolmogorov distance guarantee; and providing the empirical histogram defined by these m samples along to the distance estimation procedure) suffices to obtain the following counterpart to Corollary 2.1.2:

Corollary 2.1.32. *The algorithm* TESTSPLITTABLE*, after this modification, can* efficiently *test the classes of log-concave and monotone hazard rate (MHR) distributions, with respectively* $\tilde{O}\big(\sqrt{n}/\varepsilon^{7/2} + 1/\varepsilon^4\big)$ *and* $\tilde{O}\big(\sqrt{n}/\varepsilon^{7/2} + 1/\varepsilon^6\big)$ *samples.*

We observe that Lemma 2.1.30 and Lemma 2.1.31 actually imply efficient *proper* learning algorithms for the classes of respectively MHR and log-concave distributions, with sample complexity $O\big(1/\varepsilon^4\big)$ and $O\big(1/\varepsilon^6\big)$. Along with analogous subroutines of [3], these were the first proper learning algorithms (albeit with suboptimal sample complexity) for these classes. (Subsequent work of Diakonikolas, Kane, and Steward [83] recently obtained, through a completely different approach, a sample-optimal and efficient learning algorithm for the class of log-concave distributions which is both *proper* and *agnostic*.)

2.1.4 Going Further: Reducing the Support Size

The general approach we have been following so far gives, out-of-the-box, an efficient testing algorithm with sample complexity $\tilde{O}(\sqrt{n})$ for a large range of properties. However, this sample complexity can for some classes $\mathcal{P}$ be brought down a lot more, by taking advantage in a preprocessing step of good concentration guarantees of distributions in $\mathcal{P}$.

As a motivating example, consider the class of Poisson Binomial Distributions (PBD). It is well-known (see e.g. [121, Section 2]) that PBDs are unimodal, and more specifically that $\mathcal{PBD}_n \subseteq \mathcal{LCV}_n \subseteq \mathcal{M}_{n,1}$. Therefore, using our generic framework we can test Poisson Binomial Distributions with $\tilde{O}(\sqrt{n})$ samples. This is, however, far from optimal: as shown in [2], a sample complexity of $\Theta\big(n^{1/4}\big)$ is both necessary and sufficient. The reason our general algorithm ends up making quadratically too many queries can be explained as follows. PBDs are tightly concentrated around their expectation, so that they "morally" live on a support of size $m = O(\sqrt{n})$. Yet, instead of testing them on this very small support, in the above we still consider the entire range $[n]$, and thus end up paying a dependence $\sqrt{n}$ – instead of $\sqrt{m}$.

If we could use that observation to first reduce the domain to the *effective support* of the distribution, then we could call our testing algorithm on this reduced domain of size $O(\sqrt{n})$. In the rest of this section, we formalize and develop this idea, and in Section 2.1.4.2 will obtain as a direct application a $\tilde{O}\big(n^{1/4}\big)$-query testing algorithm for $\mathcal{PBD}_n$.

Definition 2.1.33. Given $\varepsilon > 0$, the *ε-effective support* of a distribution $\mathbf{p}$ is the smallest interval I such that $\mathbf{p}(I) \geq 1 - \varepsilon$.

The last definition we shall require is that of the *conditioned distributions* of a class $\mathcal{C}$:

Definition 2.1.34. For any class of distributions $\mathcal{C}$ over $[n]$, define the set of *conditioned distributions of $\mathcal{C}$*

(with respect to $\varepsilon > 0$ and interval $I \subseteq [n]$) as $\mathcal{C}^{\varepsilon,I} \stackrel{\text{def}}{=} \{ \mathbf{p}_I : \mathbf{p} \in \mathcal{C}, \mathbf{p}(I) \geq 1 - \varepsilon \}$.

Finally, we will require the following simple result:

Lemma 2.1.35. *Let $\mathbf{p}$ be a distribution over $[n]$, and $I \subseteq [n]$ an interval such that $\mathbf{p}(I) \geq 1 - \frac{\varepsilon}{10}$. Then,*

- *If $\mathbf{p} \in \mathcal{C}$, then $\mathbf{p}_I \in \mathcal{C}^{\frac{\varepsilon}{10},I}$;*
- *If $\ell_1(\mathbf{p}, \mathcal{C}) > \varepsilon$, then $\ell_1(\mathbf{p}_I, \mathcal{C}^{\frac{\varepsilon}{10},I}) > \frac{7\varepsilon}{10}$.*

Proof. The first item is obvious. As for the second, let $P \in \mathcal{C}$ be any distribution with $P(I) \geq 1 - \frac{\varepsilon}{10}$. By assumption, $\|\mathbf{p} - P\|_1 > \varepsilon$: but we have, writing $\alpha = 1/10$,

$$\begin{aligned}
\|\mathbf{p}_I - P_I\|_1 &= \sum_{i \in I} \left| \frac{\mathbf{p}(i)}{\mathbf{p}(I)} - \frac{P(i)}{P(I)} \right| = \frac{1}{\mathbf{p}(I)} \sum_{i \in I} \left| \mathbf{p}(i) - P(i) + P(i)\left(1 - \frac{\mathbf{p}(I)}{P(I)}\right) \right| \\
&\geq \frac{1}{\mathbf{p}(I)} \Big(\sum_{i \in I} |\mathbf{p}(i) - P(i)| - \left| 1 - \frac{\mathbf{p}(I)}{P(I)} \right| \sum_{i \in I} P(i) \Big) \\
&= \frac{1}{\mathbf{p}(I)} \Big(\sum_{i \in I} |\mathbf{p}(i) - P(i)| - |P(I) - \mathbf{p}(I)| \Big) \geq \frac{1}{\mathbf{p}(I)} \Big(\sum_{i \in I} |\mathbf{p}(i) - P(i)| - \alpha\varepsilon \Big) \\
&\geq \frac{1}{\mathbf{p}(I)} \big(\|\mathbf{p} - P\|_1 - \sum_{i \notin I} |\mathbf{p}(i) - P(i)| - \alpha\varepsilon \big) \geq \frac{1}{\mathbf{p}(I)} \big(\|\mathbf{p} - P\|_1 - 3\alpha\varepsilon \big) \\
&> (1 - 3\alpha)\varepsilon = \frac{7}{10}\varepsilon.
\end{aligned}$$

□

We now proceed to state and prove our result – namely, efficient testing of *structured* classes of distributions with nice *concentration properties*.

Theorem 2.1.36. *Let $\mathcal{C}$ be a class of distributions over $[n]$ for which the following holds.*

1. *there is a function $M(\cdot, \cdot)$ such that each $\mathbf{p} \in \mathcal{C}$ has ε-effective support of size at most $M(n, \varepsilon)$;*
2. *for every $\varepsilon \in [0, 1]$ and interval $I \subseteq [n]$, $\mathcal{C}^{\varepsilon,I}$ is (γ, ζ, L)-splittable;*
3. *there exists an efficient procedure* PROJECTIONDIST$_{\mathcal{C}^{\varepsilon,I}}$ *which, given as input the explicit description of a distribution $\mathbf{p}$ over $[n]$ and interval $I \subseteq [n]$, computes the distance $\ell_1(\mathbf{p}_I, \mathcal{C}^{\varepsilon,I})$.*

Then, the algorithm TESTEFFECTIVESPLITTABLE *(Algorithm 2) is a $O\Big(\max\Big(\frac{1}{\varepsilon^3}\sqrt{m\ell}\log m, \frac{\ell}{\varepsilon^2}\Big)\Big)$-sample tester for $\mathcal{C}$, where $m = M(n, \frac{\varepsilon}{60})$ and $\ell = L(\frac{\varepsilon}{1200}, \frac{\varepsilon}{1200}, m)$.*

Algorithm 2 TESTEFFECTIVESPLITTABLE

Require: Domain Ω (interval of size n), sample access to $\mathbf{p}$ over Ω; subroutine PROJECTIONDIST$_{\mathcal{C}^{\varepsilon,I}}$

Input: Parameters $\varepsilon \in (0,1]$, function $L(\cdot,\cdot,\cdot)$, and upper bound function $M(\cdot,\cdot)$ for the effective support of the class $\mathcal{C}$.

1: Set $m \stackrel{\text{def}}{=} O(1/\varepsilon^2)$, $\tau \stackrel{\text{def}}{=} M(n, \frac{\varepsilon}{60})$.
2: EFFECTIVE SUPPORT
3: Compute $\hat{\mathbf{p}}$, an empirical estimate of $\mathbf{p}$, by drawing m independent samples from $\mathbf{p}$.
4: Let J be the largest interval of the form $\{1,\dots,j\}$ such that $\hat{\mathbf{p}}(J) \le \frac{\varepsilon}{30}$.
5: Let K be the largest interval of the form $\{k,\dots,n\}$ such that $\hat{\mathbf{p}}(K) \le \frac{\varepsilon}{30}$.
6: Set $I \leftarrow [n] \setminus (J \cup K)$.
7: **if** $|I| > \tau$ **then return** reject
8: **end if**
9:
10: TESTING
11: Call TESTSPLITTABLE with I (providing simulated access to $\mathbf{p}_I$ by rejection sampling, returning fail if the number of samples q from $\mathbf{p}_I$ required by the subroutine is not obtained after $O(q)$ samples from $\mathbf{p}$), PROJECTIONDIST$_{\mathcal{C}^{\varepsilon,I}}$, parameters $\varepsilon' \stackrel{\text{def}}{=} \frac{7\varepsilon}{10}$ and $L(\cdot,\cdot,\cdot)$.
12: **return** accept if TESTSPLITTABLE accepts, reject otherwise.
13:

2.1.4.1 Proof of Theorem 2.1.36

By the choice of m and the DKW inequality, with probability at least $23/24$ the estimate $\hat{\mathbf{p}}$ satisfies $\mathrm{d_K}(\mathbf{p},\hat{\mathbf{p}}) \le \frac{\varepsilon}{60}$. Conditioning on that from now on, we get that $\mathbf{p}(I) \ge \hat{\mathbf{p}}(I) - \frac{\varepsilon}{30} \ge 1 - \frac{\varepsilon}{10}$. Furthermore, denoting by j and k the two inner endpoints of J and K in Steps 4 and 5, we have $\mathbf{p}(J\cup\{j+1\}) \ge \hat{\mathbf{p}}(J\cup\{j+1\}) - \frac{\varepsilon}{60} > \frac{\varepsilon}{60}$ (similarly for $\mathbf{p}(K \cup \{k-1\})$), so that I has size at most $\sigma + 1$, where σ is the $\frac{\varepsilon}{60}$-effective support size of $\mathbf{p}$.

Finally, note that since $\mathbf{p}(I) = \Omega(1)$ by our conditioning, the simulation of samples by rejection sampling will succeed with probability at least $23/24$ and the algorithm will not output fail.

Sample complexity The sample complexity is the sum of the $O(1/\varepsilon^2)$ in Step 3 and the $O(q)$ in Step 11. From Theorem 2.1.1 and the choice of I, this latter quantity is $O\left(\max\left(\frac{1}{\varepsilon^3}\sqrt{m\ell}\log m, \frac{\ell}{\varepsilon^2}\right)\right)$ where $m = M(n, \frac{\varepsilon}{60})$ and $\ell = L(\frac{\varepsilon}{1200}, \frac{\varepsilon}{1200}, M(n, \frac{\varepsilon}{60}))$.

Correctness If $\mathbf{p} \in \mathcal{C}$, then by the setting of τ (set to be an upper bound on the $\frac{\varepsilon}{60}$-effective support size of any distribution in $\mathcal{C}$) the algorithm will go beyond Step 6. The call to TESTSPLITTABLE will then end up in the algorithm returning accept in Step 12, with probability at least $2/3$ by Lemma 2.1.35, Theorem 2.1.1 and our choice of parameters.

Similarly, if $\mathbf{p}$ is ε-far from $\mathcal{C}$, then either its effective support is too large (and then the test on Step 6 fails), or the main tester will detect that its conditional distribution on I is $\frac{7\varepsilon}{10}$-far from $\mathcal{C}$ and output reject in Step 12.

Overall, in either case the algorithm is correct except with probability at most $1/24 + 1/24 + 1/3 = 5/12$ (by a union bound). Repeating constantly many times and outputting the majority vote brings the probability of failure down to $1/3$. □

2.1.4.2 Application: Testing Poisson Binomial Distributions

In this section, we illustrate the use of our generic two-stage approach to test the class of Poisson Binomial Distributions. Specifically, we prove the following result:

Corollary 2.1.37. *The class of Poisson Binomial Distributions can be tested with* $\tilde{O}\big(n^{1/4}/\varepsilon^{7/2}\big)+\tilde{O}\big(\log^2 n/\varepsilon^3\big)$ *samples, using Algorithm 2.*

This is a direct consequence of Theorem 2.1.36 and the lemmas below. The first one states that, indeed, PBDs have small effective support:

Fact 2.1.38. *For any* $\varepsilon > 0$*, a PBD has* ε*-effective support of size* $O\Big(\sqrt{n\log(1/\varepsilon)}\Big)$.

Proof. By an additive Chernoff Bound, any random variable X following a Poisson Binomial Distribution has $\Pr[\,|X-\mathbb{E}X|>\gamma n\,]\leq 2e^{-2\gamma^2 n}$. Taking $\gamma \stackrel{\text{def}}{=} \sqrt{\frac{1}{2n}\ln\frac{2}{\varepsilon}}$, we get that $\Pr[\,X\in I\,]\geq 1-\varepsilon$, where $I \stackrel{\text{def}}{=} [\mathbb{E}X-\sqrt{\frac{n}{2}\ln\frac{2}{\varepsilon}},\mathbb{E}X+\sqrt{\frac{n}{2}\ln\frac{2}{\varepsilon}}]$. □

It is clear that if $\mathbf{p}\in\mathcal{PBD}_n$ (and therefore is unimodal), then for any interval $I\subseteq[n]$ the conditional distribution $\mathbf{p}_I$ is still unimodal, and thus the class of *conditioned PBDs* $\mathcal{PBD}_n^{\varepsilon,I} \stackrel{\text{def}}{=} \{\,\mathbf{p}_I\,:\,\mathbf{p}\in\mathcal{PBD}_n, \mathbf{p}(I)\geq 1-\varepsilon\,\}$ falls under Corollary 2.1.18. The last piece we need to apply our generic testing framework is the existence of an algorithm to compute the distance between an (explicit) distribution and the class of conditioned PBDs. This is provided by our next lemma:

Claim 2.1.39. *There exists a procedure* $\textsc{ProjectionDist}_{\mathcal{PBD}_n^{\varepsilon,I}}$ *that, on input* n *and* $\varepsilon\in[0,1]$, $I\subseteq[n]$ *as well as the full specification of a distribution* $\mathbf{p}$ *on* $[n]$*, computes a value* τ *such that* $\tau\in[1\pm 2\varepsilon]\cdot\ell_1(\mathbf{p},\mathcal{PBD}_n^{\varepsilon,I})\pm\frac{\varepsilon}{100}$*, in time* $n^2\,(1/\varepsilon)^{O(\log 1/\varepsilon)}$.

Proof. The goal is to find a $\gamma=\Theta(\varepsilon)$-approximation of the minimum value of $\sum_{i\in I}\left|\frac{P(i)}{P(I)}-\frac{\mathbf{p}(i)}{\mathbf{p}(I)}\right|$, subject to $P(I)=\sum_{i\in I}P(i)\geq 1-\varepsilon$ and $P\in\mathcal{PBD}_n$. We first note that, given the parameters $n\in\mathbb{N}$ and $p_1,\ldots,p_n\in[0,1]$ of a PBD P, the vector of $(n+1)$ probabilities $P(0),\ldots,P(n)$ can be obtained in time $O\big(n^2\big)$ by dynamic programming. Therefore, computing the ℓ_1 distance between $\mathbf{p}$ and any PBD with known parameters can be done efficiently. To conclude, we invoke a result of Diakonikolas, Kane, and Stewart, that guarantees the existence of a succinct (proper) cover of $\mathcal{PBD}_n$:

Theorem 2.1.40 ([85, Theorem 4] (rephrased))**.** *For all* $n,\gamma>0$*, there exists a set* $\mathcal{S}_\gamma\subseteq\mathcal{PBD}_n$ *such that:*

(i) $\mathcal{S}_\gamma$ *is a* γ*-cover of* $\mathcal{PBD}_n$*; that is, for all* $\mathbf{p}\in\mathcal{PBD}_n$ *there exists some* $\mathbf{p}'\in\mathcal{S}_\gamma$ *such that* $\|\mathbf{p}-\mathbf{p}'\|_1\leq\gamma$

(ii) $|\mathcal{S}_\gamma|\leq n\,(1/\gamma)^{O(\log 1/\gamma)}$

(iii) $\mathcal{S}_\gamma$ *can be computed in time* $n\,(1/\gamma)^{O(\log 1/\gamma)}$

and each $\mathbf{p}\in\mathcal{S}_\gamma$ *is explicitly described by its set of parameters.*

We further observe that the factor n in both the size of the cover and running time can be easily removed in our case, as we know a good approximation of the support size of the candidate PBDs. (That is, we only need

to enumerate over a subset of the cover of [85], that of the PBDs with effective support compatible with our distribution $\mathbf{p}$.)

Set $\gamma \stackrel{\text{def}}{=} \frac{\varepsilon}{250}$. Fix $P \in \mathcal{PBD}_n$ such that $P(I) \geq 1-\varepsilon$, and $Q \in \mathcal{S}_\gamma$ such that $\|P-Q\|_1 \leq \gamma$. In particular, it is easy to see via the correspondence between ℓ_1 and total variation distance that $|P(I)-Q(I)| \leq \gamma/2$. By a calculation similar to that of Lemma 2.1.35, we have

$$\begin{aligned}
\|P_I - Q_I\|_1 &= \sum_{i\in I}\left|\frac{P(i)}{P(I)} - \frac{Q(i)}{Q(I)}\right| = \sum_{i\in I}\left|\frac{P(i)}{P(I)} - \frac{Q(i)}{P(I)} + Q(i)\left(\frac{1}{P(I)} - \frac{1}{Q(I)}\right)\right| \\
&= \sum_{i\in I}\left|\frac{P(i)}{P(I)} - \frac{Q(i)}{P(I)}\right| \pm \sum_{i\in I} Q(i)\left|\frac{1}{P(I)} - \frac{1}{Q(I)}\right| = \frac{1}{P(I)}\left(\sum_{i\in I}|P(i)-Q(i)| \pm |P(I)-Q(I)|\right) \\
&= \frac{1}{P(I)}\left(\sum_{i\in I}|P(i)-Q(i)| \pm \frac{\gamma}{2}\right) = \frac{1}{P(I)}\left(\|P-Q\|_1 \pm \frac{5\gamma}{2}\right) \\
&\in \left[\|P-Q\|_1 - 5\gamma/2, (1+2\varepsilon)\left(\|P-Q\|_1 + 5\gamma/2\right)\right]
\end{aligned}$$

where we used the fact that $\sum_{i\notin I}|P(i)-Q(i)| = 2\left(\sum_{i\notin I:\ P(i)>Q(i)}(P(i)-Q(i))\right) + Q(I) - P(I) \in [-2\gamma, 2\gamma]$. By the triangle inequality, this implies that the minimum of $\|P_I - \mathbf{p}_I\|_1$ over the distributions P of $\mathcal{S}_\varepsilon$ with $P(I) \geq 1-(\varepsilon+\gamma/2)$ will be within an additive $O(\varepsilon)$ of $\ell_1(\mathbf{p}, \mathcal{PBD}_n^{\varepsilon,I})$. The fact that the former can be found (by enumerating over the cover of size $(1/\varepsilon)^{O(\log 1/\varepsilon)}$ by the above discussion, and for each distribution in the cover computing the vector of probabilities and the distance to $\mathbf{p}$) in time $O(n^2)\cdot|S_\varepsilon| = n^2\cdot(1/\varepsilon)^{O(\log 1/\varepsilon)}$ concludes the proof. □

As previously mentioned, this approximation guarantee for $\ell_1(\mathbf{p}, \mathcal{PBD}_n^{\varepsilon,I})$ is sufficient for the purpose of Algorithm 1.

Proof of Corollary 2.1.37. Combining the above, we invoke Theorem 2.1.36 with $M(n,\varepsilon) = O(\sqrt{n\log(1/\varepsilon)})$ (Fact 2.1.38) and $L(\gamma,\zeta,m) = O\big(\frac{1}{\gamma}\log^2\frac{m}{\zeta}\big)$ (Corollary 2.1.18). This yields the claimed sample complexity; finally, the efficiency is a direct consequence of Claim 2.1.39. □

2.1.5 A Generic Tolerant Testing Upper Bound

To conclude this work, we address the question of tolerant testing of distribution classes. In the same spirit as before, we focus on describing a generic approach to obtain such bounds, in a clean conceptual manner. The most general statement of the result we prove in this section is stated below, which we then instantiate to match the lower bounds from Section 3.1.1:

Theorem 2.1.41. *Let $\mathcal{C}$ be a class of distributions over $[n]$ for which the following holds:*

(i) there exists a semi-agnostic learner $\mathcal{L}$ for $\mathcal{C}$, with sample complexity $q_L(n,\varepsilon,\delta)$ and "agnostic constant" c;

(ii) for any $\eta \in [0,1]$, every distribution in $\mathcal{C}$ has η-effective support of size at most $M(n,\eta)$.

Then, there exists an algorithm that, for any fixed $\kappa > 1$ and on input $\varepsilon_1, \varepsilon_2 \in (0,1)$ such that $\varepsilon_2 \geq C\varepsilon_1$, has the following guarantee (where $C > 2$ depends on c and κ only). The algorithm takes $O\Big(\frac{1}{(\varepsilon_2-\varepsilon_1)^2}\frac{m}{\log m}\Big) + q_L(n, \frac{\varepsilon_2-\varepsilon_1}{\kappa}, \frac{1}{10})$ samples (where $m = M(n,\varepsilon_1)$), and with probability at least $2/3$ distinguishes between (a) $\ell_1(\mathbf{p},\mathcal{C}) \leq \varepsilon_1$ and (b) $\ell_1(\mathbf{p},\mathcal{C}) > \varepsilon_2$. (Moreover, one can take $C = (1 + (5c+6)\frac{\kappa}{\kappa-1})$.)

Corollary 2.1.9. *Tolerant testing of log-concavity, convexity, concavity, MHR, unimodality, and t-modality can be performed with $O\big(\frac{1}{(\varepsilon_2-\varepsilon_1)^2}\frac{n}{\log n}\big)$ samples, for $\varepsilon_2 \geq C\varepsilon_1$ (where $C > 2$ is an absolute constant).*

Applying now the theorem with $M(n,\varepsilon) = \sqrt{n\log(1/\varepsilon)}$ (as per Corollary 2.1.37), we obtain an improved upper bound for Binomial and Poisson Binomial distributions:

Corollary 2.1.10. *Tolerant testing of the classes of Binomial and Poisson Binomial Distributions can be performed with $O\big(\frac{1}{(\varepsilon_2-\varepsilon_1)^2}\frac{\sqrt{n\log(1/\varepsilon_1)}}{\log n}\big)$ samples, for $\varepsilon_2 \geq C\varepsilon_1$ (where $C > 2$ is an absolute constant).*

High-level idea Somewhat similar to the lower bound framework described later in Section 2.1.5, the gist of the approach is to reduce the problem of tolerant testing membership of $\mathbf{p}$ to the *class* $\mathcal{C}$ to that of tolerant testing identity to a known *distribution* – namely, the distribution $\hat{\mathbf{p}}$ obtained after trying to agnostically learn $\mathbf{p}$. Intuitively, an agnostic learner for $\mathcal{C}$ should result in a good enough hypothesis $\hat{\mathbf{p}}$ (i.e., $\hat{\mathbf{p}}$ close enough to both $\mathbf{p}$ and $\mathcal{C}$) when $\mathbf{p}$ is ε_1-close to $\mathcal{C}$; but output a $\hat{\mathbf{p}}$ that is significantly far from either $\mathbf{p}$ or $\mathcal{C}$ when $\mathbf{p}$ is ε_2-far from $\mathcal{C}$ – sufficiently for us to be able to tell. Besides the many technical details one has to control for the parameters to work out, one key element is the use of a tolerant testing algorithm for closeness of two distributions due to [172], whose (tight) sample complexity scales as $n/\log n$ for a domain of size n. In order to get the right dependence on the effective support (required in particular for Corollary 2.1.10), we have to perform a first test to identify the effective support of the distribution and check its size, in order to only call this tolerant closeness testing algorithm on this much smaller subset. (This additional preprocessing step itself has to be carefully done, and comes at the price of a slightly worse constant $C = C(c,\kappa)$ in the statement of the theorem.)

2.1.5.1 Proof of Theorem 2.1.41

As described in the preceding section, the algorithm will rely on the ability to perform tolerant testing of equivalence between two unknown distributions (over some known domain of size m). This is ensured by an algorithm of Valiant and Valiant, restated below:

Theorem 2.1.42 ([172, Theorem 3 and 4])**.** *There exists an algorithm $\mathcal{E}$ which, given sampling access to two unknown distributions $\mathbf{p}_1, \mathbf{p}_2$ over $[m]$, satisfies the following. On input $\varepsilon \in (0,1]$, it takes $O(\frac{1}{\varepsilon^2}\frac{m}{\log m})$ samples from $\mathbf{p}_1$ and $\mathbf{p}_2$, and outputs a value Δ such that $|\|\mathbf{p}_1 - \mathbf{p}_2\|_1 - \Delta| \leq \varepsilon$ with probability $1 - 1/\operatorname{poly}(m)$. (Furthermore, $\mathcal{E}$ runs in time $\operatorname{poly}(m)$.)*

For the proof, we will also need this fact, similar to Lemma 2.1.35, which relates the distance of two distributions to that of their conditional distributions on a subset of the domain:

Fact 2.1.43. *Let $\mathbf{p}$ and P be distributions over $[n]$, and $I \subseteq [n]$ an interval such that $\mathbf{p}(I) \geq 1-\alpha$ and $P(I) \geq 1-\beta$. Then,*

- $\|\mathbf{p}_I - P_I\|_1 \leq \frac{3}{2}\frac{\|\mathbf{p}-P\|_1}{\mathbf{p}(I)} \leq 3\|\mathbf{p}-P\|_1$ *(the last inequality for $\alpha \leq \frac{1}{2}$); and*
- $\|\mathbf{p}_I - P_I\|_1 \geq \|\mathbf{p}-P\|_1 - 2(\alpha+\beta)$.

Proof. To establish the first item, write:

$$\begin{aligned}
\|\mathbf{p}_I - P_I\|_1 &= \sum_{i\in I}\left|\frac{\mathbf{p}(i)}{\mathbf{p}(I)} - \frac{P(i)}{P(I)}\right| = \frac{1}{\mathbf{p}(I)}\sum_{i\in I}\left|\mathbf{p}(i) - P(i) + P(i)\big(1-\frac{\mathbf{p}(I)}{P(I)}\big)\right| \\
&\leq \frac{1}{\mathbf{p}(I)}\Big(\sum_{i\in I}|\mathbf{p}(i)-P(i)| + \left|1-\frac{\mathbf{p}(I)}{P(I)}\right|\sum_{i\in I}P(i)\Big) \\
&= \frac{1}{\mathbf{p}(I)}\Big(\sum_{i\in I}|\mathbf{p}(i)-P(i)| + |P(I)-\mathbf{p}(I)|\Big) \leq \frac{1}{\mathbf{p}(I)}\Big(\sum_{i\in I}|\mathbf{p}(i)-P(i)| + \frac{1}{2}\|\mathbf{p}-P\|_1\Big) \\
&\leq \frac{1}{\mathbf{p}(I)}\cdot\frac{3}{2}\|\mathbf{p}-P\|_1
\end{aligned}$$

where we used the fact that $|P(I)-\mathbf{p}(I)| \leq \mathrm{d}_{\mathrm{TV}}(\mathbf{p},P) = \frac{1}{2}\|\mathbf{p}-P\|_1$. Turning now to the second item, we have:

$$\begin{aligned}
\|\mathbf{p}_I - P_I\|_1 &= \frac{1}{\mathbf{p}(I)}\sum_{i\in I}\left|\mathbf{p}(i) - P(i) + P(i)\left(1-\frac{\mathbf{p}(I)}{P(I)}\right)\right| \geq \frac{1}{\mathbf{p}(I)}\Big(\sum_{i\in I}|\mathbf{p}(i)-P(i)| - \left|1-\frac{\mathbf{p}(I)}{P(I)}\right|\sum_{i\in I}P(i)\Big) \\
&= \frac{1}{\mathbf{p}(I)}\Big(\sum_{i\in I}|\mathbf{p}(i)-P(i)| - |P(I)-\mathbf{p}(I)|\Big) \geq \frac{1}{\mathbf{p}(I)}\Big(\sum_{i\in I}|\mathbf{p}(i)-P(i)| - (\alpha+\beta)\Big) \\
&\geq \frac{1}{\mathbf{p}(I)}\big(\|\mathbf{p}-P\|_1 - \sum_{i\notin I}|\mathbf{p}(i)-P(i)| - (\alpha+\beta)\big) \geq \frac{1}{\mathbf{p}(I)}\big(\|\mathbf{p}-P\|_1 - 2(\alpha+\beta)\big) \\
&\geq \|\mathbf{p}-P\|_1 - 2(\alpha+\beta).
\end{aligned}$$

□

With these two ingredients, we are in position to establish our theorem:

Proof of Theorem 2.1.41. The algorithm proceeds as follows, where we set $\varepsilon \stackrel{\text{def}}{=} \frac{\varepsilon_2-\varepsilon_1}{17\kappa}$, $\theta \stackrel{\text{def}}{=} \varepsilon_2 - ((6+c)\varepsilon_1 + 11\varepsilon)$, and $\tau \stackrel{\text{def}}{=} 2\frac{(3+c)\varepsilon_1+5\varepsilon}{2}$:

(1) using $O(\frac{1}{\varepsilon^2})$ samples, get (with probability at least $1-1/10$, by Theorem 1.4.3) a distribution $\tilde{\mathbf{p}}$ $\frac{\varepsilon}{2}$-close to $\mathbf{p}$ in Kolmogorov distance; and let $I \subseteq [n]$ be the smallest interval such that $\tilde{\mathbf{p}}(I) > 1 - \frac{3}{2}\varepsilon_1 - \varepsilon$. Output reject if $|I| > M(n,\varepsilon_1)$.

(2) invoke $\mathcal{L}$ on $\mathbf{p}$ with parameters ε and failure probability $\frac{1}{10}$, to obtain a hypothesis $\hat{\mathbf{p}}$;

(3) call $\mathcal{E}$ (from Theorem 2.1.42) on $\mathbf{p}_I$, $\hat{\mathbf{p}}_I$ with parameter $\frac{\varepsilon}{6}$ to get an estimate $\hat{\Delta}$ of $\|\mathbf{p}_I - \hat{\mathbf{p}}_I\|_1$;

(4) output reject if $\hat{\mathbf{p}}(I) < 1 - \tau$;

(5) compute "offline" (an estimate accurate within ε of) $\ell_1(\hat{\mathbf{p}}, \mathcal{C})$, denoted Δ;

(6) output reject is $\Delta + \hat{\Delta} > \theta$, and output accept otherwise.

The claimed sample complexity is immediate from Steps (2) and (3), along with Theorem 2.1.42. Turning to correctness, we condition on both subroutines meeting their guarantee (i.e., $\|\mathbf{p} - \hat{\mathbf{p}}\|_1 \leq c \cdot \mathrm{OPT} + \varepsilon$ and $\|\mathbf{p} - \hat{\mathbf{p}}\|_1 \in [\hat{\Delta} - \varepsilon, \hat{\Delta} + \varepsilon]$), which happens with probability at least $8/10 - 1/\operatorname{poly}(n) \geq 3/4$ by a union bound.

Completeness If $\ell_1(\mathbf{p}, \mathcal{C}) \leq \varepsilon_1$, then $\mathbf{p}$ is ε_1-close to some $P \in \mathcal{C}$, for which there exists an interval $J \subseteq [n]$ of size at most $M(n, \varepsilon_1)$ such that $P(J) \geq 1 - \varepsilon_1$. It follows that $\mathbf{p}(J) \geq 1 - \frac{3}{2}\varepsilon_1$ (since $|\mathbf{p}(J) - P(J)| \leq \frac{\varepsilon_1}{2}$) and $\tilde{\mathbf{p}}(J) \geq 1 - \frac{3}{2}\varepsilon_1 - 2 \cdot \frac{\varepsilon}{2}\varepsilon$; establishing existence of a good interval I to be found (and Step (1) does not end with reject). Additionally, $\|\mathbf{p} - \hat{\mathbf{p}}\|_1 \leq c \cdot \varepsilon_1 + \varepsilon$ and by the triangle inequality this implies $\ell_1(\hat{\mathbf{p}}, \mathcal{C}) \leq (1 + c)\varepsilon_1 + \varepsilon$.

Moreover, as $\mathbf{p}(I) \geq \tilde{\mathbf{p}}(I) - 2 \cdot \frac{\varepsilon}{2} \geq 1 - \frac{3}{2}\varepsilon_1 - 2\varepsilon$ and $|\hat{\mathbf{p}}(I) - \mathbf{p}(I)| \leq \frac{1}{2}\|\mathbf{p} - \hat{\mathbf{p}}\|_1$, we do have

$$\hat{\mathbf{p}}(I) \geq 1 - \frac{3}{2}\varepsilon_1 - 2\varepsilon - \frac{c\varepsilon_1}{2} - \frac{\varepsilon}{2} = 1 - \tau$$

and the algorithm does not reject in Step (4). To conclude, one has by Fact 2.1.43 that

$$\|\mathbf{p}_I - \hat{\mathbf{p}}_I\|_1 \leq \frac{3}{2}\frac{\|\mathbf{p} - \hat{\mathbf{p}}\|_1}{\mathbf{p}(I)} \leq \frac{3}{2}\frac{(c\varepsilon_1 + \varepsilon)}{1 - \frac{3}{2}\varepsilon_1 - 2\varepsilon} \leq 3(c\varepsilon_1 + \varepsilon) \qquad \text{(for } \varepsilon_1 < 1/4, \text{ as } \varepsilon < 1/17\text{)}$$

Therefore, $\Delta + \hat{\Delta} \leq \ell_1(\hat{\mathbf{p}}, \mathcal{C}) + \varepsilon + \|\mathbf{p}_I - \hat{\mathbf{p}}_I\|_1 + \varepsilon \leq (4c + 1)\varepsilon_1 + 6\varepsilon \leq \varepsilon_2 - ((6 + c)\varepsilon_1 + 11\varepsilon) = \theta$ (the last inequality by the assumption on $\varepsilon_2, \varepsilon_1$), and the tester accepts.

Soundness If $\ell_1(\mathbf{p}, \mathcal{C}) > \varepsilon_2$, then we must have $\|\mathbf{p} - \hat{\mathbf{p}}\|_1 + \ell_1(\hat{\mathbf{p}}, \mathcal{C}) > \varepsilon_2$. If the algorithm does not already reject in Step (4), then $\hat{\mathbf{p}}(I) \geq 1 - \tau$. But, by Fact 2.1.43,

$$\begin{aligned}\|\mathbf{p}_I - \hat{\mathbf{p}}_I\|_1 &\geq \|\mathbf{p} - \hat{\mathbf{p}}\|_1 - 2(\mathbf{p}(I^c) + \hat{\mathbf{p}}(I^c)) \geq \|\mathbf{p}_I - \hat{\mathbf{p}}_I\|_1 - 2\Big(\frac{3}{2}\varepsilon_1 + 2\varepsilon + \tau\Big) \\ &= \|\mathbf{p} - \hat{\mathbf{p}}\|_1 - ((6 + c)\varepsilon_1 + 9\varepsilon)\end{aligned}$$

we then have $\|\mathbf{p}_I - \hat{\mathbf{p}}_I\|_1 + \ell_1(\hat{\mathbf{p}}, \mathcal{C}) > \varepsilon_2 - ((6+c)\varepsilon_1 + 9\varepsilon)$. This implies $\Delta + \hat{\Delta} > \varepsilon_2 - ((6+c)\varepsilon_1 + 9\varepsilon) - 2\varepsilon = \varepsilon_2 - ((6+c)\varepsilon_1 + 11\varepsilon) = \theta$, and the tester rejects. Finally, the testing algorithm defined above is computationally efficient as long as both the learning algorithm (Step (2)) and the estimation procedure (Step (5)) are.

□

2.1.6 Proof of Theorem 2.1.21

In this section, we prove our structural result for MHR distributions, Theorem 2.1.21:

Theorem 2.1.21 (Monotone Hazard Rate). *For all $\gamma, \zeta > 0$, the class $\mathcal{MHR}_n$ of MHR distributions on $[n]$ is (γ, ζ, L)-decomposable for $L \stackrel{\text{def}}{=} O\left(\frac{\log \frac{n}{\zeta}}{\gamma}\right)$.*

Proof. We reproduce and adapt the argument of [56, Section 5.1] to meet our definition of decomposability (which, albeit related, is incomparable to theirs). First, we modify the algorithm at the core of their constructive proof, in Algorithm 4: note that the only two changes are in Steps 2 and 3, where we use parameters respectively $\frac{\zeta}{n}$ and $\frac{\zeta}{n^2}$. Following the structure of their proof, we write $\mathcal{Q} = \{I_1, \ldots, I_{|\mathcal{Q}|}\}$ with $I_i = [a_i, b_i]$, and define

Algorithm 3 RIGHT-INTERVAL$(\mathbf{p}, J, \tau)$

Require: explicit description of distribution $\mathbf{p}$ over $[n]$; interval $J = [a, b] \subseteq [n]$; threshold $\tau > 0$

1: **if** $\mathbf{p}(b) > \tau$ **then**
2: Set $i' \leftarrow b$
3: **else**
4: Set $i' \leftarrow \min\{\, a \le i \le b : \ \mathbf{p}([i, b]) \le \tau \,\}$
5: **end if**
6: **return** $[i', b]$

Algorithm 4 DECOMPOSE-MHR$'(\mathbf{p}, \gamma)$

Require: explicit description of MHR distribution $\mathbf{p}$ over $[n]$; accuracy parameter $\gamma > 0$

1: Set $J \leftarrow [n]$ and $\mathcal{Q} \leftarrow \emptyset$.
2: Let $I \leftarrow$ RIGHT-INTERVAL$(\mathbf{p}, J, \frac{\zeta}{n})$ and $I' \leftarrow$ RIGHT-INTERVAL$(\mathbf{p}, J \setminus I, \frac{\zeta}{n})$. Set $J \leftarrow J \setminus (I \cup I')$.
3: Set $i \in J$ to be the smallest integer such that $\mathbf{p}(i) \ge \frac{\zeta}{n^2}$. If no such i exists, let $I'' \leftarrow J$ and go to Step 9. Otherwise, let $I'' \leftarrow \{1, \ldots, i-1\}$ and $J \leftarrow J \setminus I''$.
4: **while** $J \neq \emptyset$ **do**
5: Let $j \in J$ bet the smallest integer such that $\mathbf{p}(j) \notin [\frac{1}{1+\gamma}, 1+\gamma]\mathbf{p}(i)$. If no such j exists, let $I''' \leftarrow J$; otherwise let $I''' \leftarrow \{i, \ldots, j-1\}$.
6: Add I''' to $\mathcal{Q}$ and set $J \leftarrow J \setminus I'''$.
7: Let $i \leftarrow j$.
8: **end while**
9: **return** $\mathcal{Q} \cup \{I, I', I''\}$

.

$\mathcal{Q}' = \{\, I_i \in \mathcal{Q} : \ \mathbf{p}(a_i) > \mathbf{p}(a_{i+1}) \,\}$, $\mathcal{Q}'' = \{\, I_i \in \mathcal{Q} : \ \mathbf{p}(a_i) \le \mathbf{p}(a_{i+1}) \,\}$.

We immediately obtain the analogues of their Lemmas 5.2 and 5.3:

Lemma 2.1.44. *We have $\prod_{I_i \in \mathcal{Q}'} \frac{\mathbf{p}(a_i)}{\mathbf{p}(a_{i+1})} \le \frac{n}{\zeta}$.*

Lemma 2.1.45. *Step 4 of Algorithm 4 adds at most $O\left(\frac{1}{\gamma} \log \frac{n}{\zeta}\right)$ intervals to $\mathcal{Q}$.*

Sketch. This derives from observing that now $\mathbf{p}(I \cup I') \ge \zeta/n$, which as in [56, Lemma 5.3] in turn implies

$$1 \ge \frac{\zeta}{n}(1+\gamma)^{|\mathcal{Q}'|-1}$$

so that $|\mathcal{Q}'| = O\left(\frac{1}{\gamma} \log \frac{n}{\zeta}\right)$.

Again following their argument, we also get

$$\frac{\mathbf{p}(a_{|\mathcal{Q}|+1})}{\mathbf{p}(a_1)} = \prod_{I_i \in \mathcal{Q}''} \frac{\mathbf{p}(a_{i+1})}{\mathbf{p}(a_i)} \cdot \prod_{I_i \in \mathcal{Q}'} \frac{\mathbf{p}(a_{i+1})}{\mathbf{p}(a_i)}$$

by combining Lemma 2.1.44 with the fact that $\mathbf{p}(a_{|\mathcal{Q}|+1} \leq 1$ and that by construction $\mathbf{p}(a_i) \geq \zeta/n^2$, we get

$$\prod_{I_i \in \mathcal{Q}''} \frac{\mathbf{p}(a_{i+1})}{\mathbf{p}(a_i)} \leq \frac{n}{\zeta} \cdot \frac{n^2}{\zeta} = \frac{n^3}{\zeta^2} \,.$$

But since each term in the product is at least $(1+\gamma)$ (by construction of $\mathcal{Q}$ and the definition of $\mathcal{Q}''$), this leads to

$$(1+\gamma)^{|\mathcal{Q}''|} \leq \frac{n^3}{\zeta^2}$$

and thus $|\mathcal{Q}''| = O\left(\frac{1}{\gamma} \log \frac{n}{\zeta}\right)$ as well. □

It remains to show that $\mathcal{Q} \cup \{I, I', I''\}$ is indeed a good decomposition of $[n]$ for $\mathbf{p}$, as per Definition 2.1.13. Since by construction every interval in $\mathcal{Q}$ satisfies Item (ii), we only are left with the case of I, I' and I''. For the first two, as they were returned by RIGHT-INTERVAL either (a) they are singletons, in which case Item (ii) trivially holds; or (b) they have at least two elements, in which case they have probability mass at most $\frac{\zeta}{n}$ (by the choice of parameters for RIGHT-INTERVAL) and thus Item (i) is satisfied. Finally, it is immediate to see that by construction $\mathbf{p}(I'') \leq n \cdot \zeta/n^2 = \zeta/n$, and Item (i) holds in this case as well. □

2.1.7 Proofs from Section 2.1.3

This section contains the proofs omitted from Section 2.1.3, namely the distance estimation procedures for t-piecewise degree-d (Theorem 2.1.29), monotone hazard rate (Lemma 2.1.30), and log-concave distributions (Lemma 2.1.31).

2.1.7.1 Proof of Theorem 2.1.29

In this section, we prove the following:

Theorem 2.1.46 (Theorem 2.1.29, restated)**.** *Let p be an ℓ-histogram over $[-1,1)$. There is an algorithm* PROJECTSINGLEPOLY(d,ε) *which runs in time* $\mathrm{poly}(\ell, d+1, 1/\varepsilon)$*, and outputs a degree-$d$ polynomial q which defines a pdf over $[-1,1)$ such that $\|p-q\|_1 \leq 3\ell_1(p, \mathcal{P}_{n,d}) + O(\varepsilon)$.*

As mentioned in Section 2.1.3, the proof of this statement is a rather straightforward adaptation of the proof of [55, Theorem 9], with two differences: first, in our setting there is no uncertainty or probabilistic argument due to sampling, as we are provided with an explicit description of the histogram p. Second, Chan et al. require some "well-behavedness" assumption on the distribution p (for technical reasons essentially due to the sampling access), that we remove here. Besides these two points, the proof is almost identical to theirs,

and we only reproduce (our modification of) it here for the sake of completeness. (Any error introduced in the process, however, is solely our responsibility.)

Proof. Some preliminary definitions will be helpful:

Definition 2.1.47 (Uniform partition)**.** Let p be a subdistribution on an interval $I \subseteq [-1,1)$. A partition $\mathcal{I} = \{I_1, \ldots, I_\ell\}$ of I is (p,η)-*uniform* if $p(I_j) \leq \eta$ for all $1 \leq j \leq \ell$.

We will also use the following notation: For this subsection, let $I = [-1,1)$ (I will denote a subinterval of $[-1,1)$ when the results are applied in the next subsection). We write $\|f\|_1^{(I)}$ to denote $\int_I |f(x)|dx$, and we write $\mathrm{d}_{\mathrm{TV}}^{(I)}(p,q)$ to denote $\|p-q\|_1^{(I)}/2$. We write $\mathrm{OPT}_{1,d}^{(I)}$ to denote the infimum of the distance $\|p-g\|_1^{(I)}$ between p and any degree-d subdistribution g on I that satisfies $g(I) = p(I)$.

The key step of ProjectSinglePoly is Step 2 where it calls the FindSinglePoly procedure. In this procedure $T_i(x)$ denotes the degree-i Chebychev polynomial of the first kind. The function FindSinglePoly should be thought of as the CDF of a "quasi-distribution" f; we say that $f = F'$ is a "quasi-distribution" and not a *bona fide* probability distribution because it is not guaranteed to be non-negative everywhere on $[-1,1)$. Step 2 of FindSinglePoly processes f slightly to obtain a polynomial q which is an actual distribution over $[-1,1)$.

Algorithm 5 ProjectSinglePoly

Require: parameters d, ε; and the full description of an ℓ-histogram p over $[-1,1)$.
Ensure: a degree-d distribution q such that $\mathrm{d}_{\mathrm{TV}}(p,q) \leq 3 \cdot \mathrm{OPT}_{1,d} + O(\varepsilon)$

1: Partition $[-1,1)$ into $z = \Theta((d+1)/\varepsilon)$ intervals $I_0 = [i_0, i_1), \ldots, I_{z-1} = [i_{z-1}, i_z)$, where $i_0 = -1$ and $i_z = 1$, such that for each $j \in \{1, \ldots, z\}$ we have $p(I_j) = \Theta(\varepsilon/(d+1))$ or ($|I_j| = 1$ and $p(I_j) = \Omega(\varepsilon/(d+1))$).
2: Call FindSinglePoly(d, ε, $\eta := \Theta(\varepsilon/(d+1))$, $\{I_0, \ldots, I_{z-1}\}$, p and output the hypothesis q that it returns.

The rest of this subsection gives the proof of Theorem 2.1.29. The claimed running time bound is obvious (the computation is dominated by solving the $\mathrm{poly}(d, 1/\varepsilon)$-size LP in ProjectSinglePoly, with an additional term linear in ℓ when partitioning $[-1,1)$ in the initial first step), so it suffices to prove correctness.

Before launching into the proof we give some intuition for the linear program. Intuitively $F(x)$ represents the cdf of a degree-d polynomial distribution f where $f = F'$. Constraint (a) captures the endpoint constraints that any cdf must obey if it has the same total weight as p. Intuitively, constraint (b) ensures that for each interval $[i_j, i_k)$, the value $F(i_k) - F(i_j)$ (which we may alternately write as $f([i_j, i_k))$) is close to the weight $p([i_j, i_k))$ that the distribution puts on the interval. Recall that by assumption p is $\mathrm{OPT}_{1,d}$-close to some degree-d polynomial r. Intuitively the variable w_ℓ represents $\int_{[i_\ell, i_{\ell+1})}(r-p)$ (note that these values sum to zero by constraint (c)(2.3), and y_ℓ represents the absolute value of w_ℓ (see constraint (c)(2.4)). The value τ, which by constraint (c)(2.5) is at least the sum of the y_ℓ's, represents a lower bound on $\mathrm{OPT}_{1,d}$. The constraints in (d) and (e) reflect the fact that as a cdf, F should be bounded between 0 and 1 (more on this below), and the (f) constraints reflect the fact that the pdf $f = F'$ should be everywhere nonnegative (again more on this below).

Algorithm 6 FINDSINGLEPOLY

Require: degree parameter d; error parameter ε; parameter η; (p,η)-uniform partition $\mathcal{I}_I = \{I_1, \ldots, I_z\}$ of interval I into z intervals such that $\sqrt{\varepsilon z} \cdot \eta \le \varepsilon/2$; a subdistribution p on I

Ensure: a number τ and a degree-d subdistribution q on I such that $q(I) = p(I)$,

$$\textsc{opt}_{1,d}^{(I)} \le \|p - q\|_1^{(I)} \le 3\textsc{opt}_{1,d}^{(I)} + \sqrt{\varepsilon z(d+1)} \cdot \eta + \text{error},$$

$0 \le \tau \le \textsc{opt}_{1,d}^{(I)}$ and error $= O((d+1)\eta)$.

1: Let τ be the solution to the following LP:

minimize τ subject to the following constraints:

(Below $F(x) = \sum_{i=0}^{d+1} c_i T_i(x)$ where $T_i(x)$ is the degree-i Chebychev polynomial of the first kind, and $f(x) = F'(x) = \sum_{i=0}^{d+1} c_i T_i'(x)$.)

(a) $F(-1) = 0$ and $F(1) = p(I)$;

(b) For each $0 \le j < k \le z$,

$$\left| \left(p([i_j, i_k)) + \sum_{j \le \ell < k} w_\ell \right) - (F(i_k) - F(i_j)) \right| \le \sqrt{\varepsilon \cdot (k-j)} \cdot \eta; \tag{2.2}$$

(c)

$$\sum_{0 \le \ell < z} w_\ell = 0, \tag{2.3}$$

$$-y_\ell \le w_\ell \le y_\ell \qquad \text{for all } 0 \le \ell < z, \tag{2.4}$$

$$\sum_{0 \le \ell < z} y_\ell \le \tau; \tag{2.5}$$

(d) The constraints $|c_i| \le \sqrt{2}$ for $i = 0, \ldots, d+1$;

(e) The constraints

$$0 \le F(z) \le 1 \quad \text{for all } z \in J,$$

where J is a set of $O\big((d+1)^6\big)$ equally spaced points across $[-1, 1)$;

(f) The constraints

$$\sum_{i=0}^{d} c_i T_i'(x) \ge 0 \quad \text{for all } x \in K,$$

where K is a set of $O((d+1)^2/\varepsilon)$ equally spaced points across $[-1, 1)$.

2: Define $q(x) = \varepsilon f(I)/|I| + (1-\varepsilon) f(x)$. Output q as the hypothesis pdf.

We begin by observing that PROJECTSINGLEPOLY calls FINDSINGLEPOLY with input parameters that satisfy FINDSINGLEPOLY's input requirements:

(I) the non-singleton intervals $I_0, \ldots, I_{z-1}$ are (p, η)-uniform; and

(II) the singleton intervals each have weight at least $\frac{\eta}{10}$.

We then proceed to show that, from there, FINDSINGLEPOLY's LP is feasible and has a high-quality optimal solution.

Lemma 2.1.48. *Suppose p is an ℓ-histogram over $[-1, 1)$, so that conditions (I) and (II) above hold; then the LP defined in Step 1 of* FINDSINGLEPOLY *is feasible; and the optimal solution τ is at most* $\mathrm{OPT}_{1,d}$.

Proof. As above, let r be a degree-d polynomial pdf such that $\mathrm{OPT}_{1,d} = \|p - r\|_1$ and $r(I) = p(I)$.We exhibit a feasible solution as follows: take F to be the cdf of r (a degree d polynomial). Take w_ℓ to be $\int_{[i_\ell, i_{\ell+1})}(r - p)$, and take y_ℓ to be $|w_\ell|$. Finally, take τ to be $\sum_{0 \le \ell < z} y_\ell$.

We first argue feasibility of the above solution. We first take care of the easy constraints: since F is the cdf of a subdistribution over I it is clear that constraints (a) and (e) are satisfied, and since both r and p are pdfs with the same total weight it is clear that constraints (c)(2.3) and (f) are both satisfied. Constraints (c)(2.4) and (c)(2.5) also hold. So it remains to argue constraints (b) and (d).

Note that constraint (b) is equivalent to $p + (r - p) = r$ and r satisfying $(\mathcal{I}, \varepsilon/(d+1), \varepsilon)$-inequalities, therefore this constraint is satisfied.

To see that constraint (d) is satisfied we recall some of the analysis of Arora and Khot [10, Section 3]. This analysis shows that since F is a cumulative distribution function (and in particular a function bounded between 0 and 1 on I) each of its Chebychev coefficients is at most $\sqrt{2}$ in magnitude.

To conclude the proof of the lemma we need to argue that $\tau \le \mathrm{OPT}_{1,d}$. Since $w_\ell = \int_{[i_\ell, i_{\ell+1})}(r - p)$ it is easy to see that $\tau = \sum_{0 \le \ell < z} y_\ell = \sum_{0 \le \ell < z} |w_\ell| \le \|p - r\|_1$, and hence indeed $\tau \le \mathrm{OPT}_{1,d}$ as required. □

Having established that with high probability the LP is indeed feasible, henceforth we let τ denote the optimal solution to the LP and F, f, w_ℓ, c_i, y_ℓ denote the values in the optimal solution. A simple argument (see e.g. the proof of [10, Theorem 8]) gives that $\|F\|_\infty \le 2$. Given this bound on $\|F\|_\infty$, the Bernstein–Markov inequality implies that $\|f\|_\infty = \|F'\|_\infty \le O((d+1)^2)$. Together with (f) this implies that $f(z) \ge -\varepsilon/2$ for all $z \in [-1, 1)$. Consequently $q(z) \ge 0$ for all $z \in [-1, 1)$, and

$$\int_{-1}^{1} q(x)dx = \varepsilon + (1 - \varepsilon)\int_{-1}^{1} f(x)dx = \varepsilon + (1 - \varepsilon)(F(1) - F(-1)) = 1.$$

So $q(x)$ is indeed a degree-d pdf. To prove Theorem 2.1.29 it remains to show that $\|p - q\|_1 \le 3\mathrm{OPT}_{1,d} + O(\varepsilon)$.

We sketch the argument that we shall use to bound $\|p - q\|_1$. A key step in achieving this bound is to bound the $\|\cdot\|_{\mathcal{A}}$ distance between f and $\widehat{p}_m + w$ where $\mathcal{A} = \mathcal{A}_{d+1}$ is the class of all unions of $d+1$ intervals and w is a function based on the w_ℓ values (see (2.8) below). If we can bound $\|(p + w) - f\|_{\mathcal{A}} \le O(\varepsilon)$ then it will not be difficult to show that $\|r - f\|_{\mathcal{A}} \le \mathrm{OPT}_{1,d} + O(\varepsilon)$.. Since r and f are both degree-d polynomials we

have $\|r-f\|_1 = 2\|r-f\|_{\mathcal{A}} \le 2\textsc{opt}_{1,d} + O(\varepsilon)$, so the triangle inequality (recalling that $\|p-r\|_1 = \textsc{opt}_{1,d}$) gives $\|p-f\|_1 \le 3\textsc{opt}_{1,d} + O(\varepsilon)$. From this point a simple argument (Proposition 2.1.50) gives that $\|p-q\|_1 \le \|p-f\|_1 + O(\varepsilon)$, which gives the theorem.

We will use the following lemma that translates $(\mathcal{I},\eta,\varepsilon)$-inequalities into a bound on $\mathcal{A}_{d+1}$ distance.

Lemma 2.1.49. *Let $\mathcal{I} = \{I_0 = [i_0, i_1), \dots, I_{z-1} = [i_{z-1}, i_z)\}$ be a (p,η)-uniform partition of I, possibly augmented with singleton intervals. If $h\colon I \to \mathbb{R}$ and p satisfy the $(\mathcal{I},\eta,\varepsilon)$-inequalities, then*

$$\|p-h\|^{(I)}_{\mathcal{A}_{d+1}} \le \sqrt{\varepsilon z(d+1)}\cdot\eta + \text{error},$$

where $\text{error} = O((d+1)\eta)$.

Proof. To analyze $\|p-h\|_{\mathcal{A}_{d+1}}$, consider any union of $d+1$ disjoint non-overlapping intervals $S = J_1 \cup \dots \cup J_{d+1}$. We will bound $\|p-h\|_{\mathcal{A}_{d+1}}$ by bounding $|p(S)-h(S)|$.

We lengthen intervals in S slightly to obtain $T = J_1' \cup \dots \cup J_{d+1}'$ so that each J_j' is a union of intervals of the form $[i_\ell, i_{\ell+1})$. Formally, if $J_j = [a,b)$, then $J_j' = [a',b')$, where $a' = \max_\ell \{\, i_\ell : i_\ell \le a \,\}$ and $b' = \min_\ell \{\, i_\ell : i_\ell \ge b \,\}$. We claim that

$$|p(S)-h(S)| \le O((d+1)\eta) + |p(T)-f(T)|\,. \tag{2.6}$$

Indeed, consider any interval of the form $J = [i_\ell, i_{\ell+1})$ such that $J \cap S \ne J \cap T$ (in particular, such an interval cannot be one of the singletons). We have

$$|p(J\cap S) - p(J\cap T)| \le p(J) \le O(\eta), \tag{2.7}$$

where the first inequality uses non-negativity of p and the second inequality follows from the bound $p([i_\ell, i_{\ell+1})) \le \eta$. The $(\mathcal{I},\eta,\varepsilon)$-inequalities (between h and p) implies that the inequalities in (2.7) also hold with h in place of p. Now (2.6) follows by adding (2.7) across all $J = [i_\ell, i_{\ell+1})$ such that $J\cap S \ne J\cap T$ (there are at most $2(d+1)$ such intervals J), since each interval J_j in S can change at most two such J's when lengthened.

Now rewrite T as a disjoint union of $s \le d+1$ intervals $[i_{L_1}, i_{R_1}) \cup \dots \cup [i_{L_s}, i_{R_s})$. We have

$$|p(T)-h(T)| \le \sum_{j=1}^{s} \sqrt{R_j - L_j}\cdot\sqrt{\varepsilon}\eta$$

by $(\mathcal{I},\eta,\varepsilon)$-inequalities between p and h. Now observing that that $0 \le L_1 \le R_1 \cdots \le L_s \le R_s \le t = O((d+1)/\varepsilon)$, we get that the largest possible value of $\sum_{j=1}^{s}\sqrt{R_j-L_j}$ is $\sqrt{sz} \le \sqrt{(d+1)z}$, so the RHS of (2.6) is at most $O((d+1)\eta) + \sqrt{(d+1)z\varepsilon}\eta$, as desired. ∎

Recall from above that F, f, w_ℓ, c_i, y_ℓ, τ denote the values in the optimal solution. We claim that

$$\|(p+w)-f\|_{\mathcal{A}} = O(\varepsilon), \tag{2.8}$$

where w is the subdistribution which is constant on each $[i_\ell, i_{\ell+1})$ and has weight w_ℓ there, so in particular $\|w\|_1 \le \tau \le \mathrm{OPT}_{1,d}$. Indeed, this equality follows by applying Lemma 2.1.49 with $h = f - w$. The lemma requires h and p to satisfy $(\mathcal{I},\eta,\varepsilon)$-inequalities, which follows from constraint (b) ($(\mathcal{I},\eta,\varepsilon)$-inequalities between $p+w$ and f) and observing that $(p+w)-f = p-(f-w)$. We have also used $\eta = \Theta(\varepsilon/(d+1))$ to bound the error term of the lemma by $O(\varepsilon)$.

Next, by the triangle inequality we have (writing $\mathcal{A}$ for $\mathcal{A}_{d+1}$)

$$\|r-f\|_{\mathcal{A}} \le \|r-(p+w)\|_{\mathcal{A}} + \|(p+w)-f\|_{\mathcal{A}}.$$

The last term on the RHS has just been shown to be $O(\varepsilon)$. The first term is bounded by

$$\|r-(p+w)\|_{\mathcal{A}} \le \frac{1}{2}\|r-(p+w)\|_1 \le \frac{1}{2}(\|r-p\|_1 + \|w\|_1) \le \mathrm{OPT}_{1,d}.$$

Altogether, we get that $\|r-f\|_{\mathcal{A}} \le \mathrm{OPT}_{1,d} + O(\varepsilon)$.

Since r and f are degree d polynomials, $\|r-f\|_1 = 2\|r-f\|_{\mathcal{A}} \le 2\mathrm{OPT}_{1,d} + O(\varepsilon)$. This implies $\|p-f\|_1 \le \|p-r\|_1 + \|r-f\|_1 \le 3\mathrm{OPT}_{1,d} + O(\varepsilon)$. Finally, we turn our quasidistribution f which has value $\ge -\varepsilon/2$ everywhere into a distribution q (which is nonnegative), by redistributing the weight. The following simple proposition bounds the error incurred.

Proposition 2.1.50. *Let f and p be any sub-quasidistribution on I. If $q = \varepsilon f(I)/|I| + (1-\varepsilon)f$, then $\|q-p\|_1 \le \|f-p\|_1 + \varepsilon(f(I)+p(I))$.*

Proof. We have

$$q - p = \varepsilon(f(I)/|I| - p) + (1-\varepsilon)(f-p).$$

Therefore

$$\|q-p\|_1 \le \varepsilon\|f(I)/|I| - p\|_1 + (1-\varepsilon)\|f-p\|_1 \le \varepsilon(f(I)+p(I)) + \|f-p\|_1. \qquad \square$$

We now have $\|p-q\|_1 \le \|p-f\|_1 + O(\varepsilon)$ by Proposition 2.1.50, concluding the proof of Theorem 2.1.29. $\square$

2.1.7.2 Proof of Lemma 2.1.30

Lemma 2.1.30 (Monotone Hazard Rate)**.** *There exists a procedure* $\textsc{ProjectionDist}^*_{\mathcal{MHR}_n}$ *that, on input n as well as the full specification of a k-histogram distribution $\mathbf{p}$ on $[n]$ and of an ℓ-histogram distribution $\mathbf{p}'$ on $[n]$, runs in time $\mathrm{poly}(n, 1/\varepsilon)$, and satisfies the following.*

- *If there is $P \in \mathcal{MHR}_n$ such that $\|\mathbf{p} - P\|_1 \leq \varepsilon$ and $\mathrm{d_K}(\mathbf{p}', P) \leq \varepsilon^3$, then the procedure returns* yes*;*
- *If $\ell_1(\mathbf{p}, \mathcal{MHR}_n) > 100\varepsilon$, then the procedure returns* no*.*

Proof. For convenience, let $\alpha \stackrel{\text{def}}{=} \varepsilon^3$; we also write $[i, j]$ instead of $\{i, \ldots, j\}$.

First, we note that it is easy to reduce our problem to the case where, in the completeness case, we have $P \in \mathcal{MHR}_n$ such that $\|\mathbf{p} - P\|_1 \leq 2\varepsilon$ and $\mathrm{d_K}(\mathbf{p}, P) \leq 2\alpha$; while in the soundness case $\ell_1(\mathbf{p}, \mathcal{MHR}_n) \geq 99\varepsilon$. Indeed, this can be done with a linear program on $\mathrm{poly}(k, \ell)$ variables, asking to find a $(k + \ell)$-histogram $\mathbf{p}''$ on a refinement of $\mathbf{p}$ and $\mathbf{p}'$ minimizing the ℓ_1 distance to $\mathbf{p}$, under the constraint that the Kolmogorov distance to $\mathbf{p}'$ be bounded by ε. (In the completeness case, clearly a feasible solution exists, as P is one.) We therefore follow with this new formulation: either

(a) $\mathbf{p}$ is ε-close to a monotone hazard rate distribution P (in ℓ_1 distance) *and* $\mathbf{p}$ is α-close to P (in Kolmogorov distance); and

(b) $\mathbf{p}$ is 32ε-far from monotone hazard rate

where $\mathbf{p}$ is a $(k + \ell)$-histogram.

We then proceed by observing the following easy fact: suppose P is a MHR distribution on $[n]$, i.e. such that the quantity $h_i \stackrel{\text{def}}{=} \frac{P(i)}{\sum_{j=i}^{n} P(i)}$, $i \in [n]$ is non-increasing. Then, we have

$$P(i) = h_i \prod_{j=1}^{i-1} (1 - h_j), \qquad i \in [n]. \tag{2.9}$$

and there is a bijective correspondence between P and $(h_i)_{i \in [n]}$.

We will write a linear program with variables $y_1, \ldots, y_n$, with the correspondence $y_i \stackrel{\text{def}}{=} \ln(1 - h_i)$. Note that with this parameterization, we get that if the $(y_i)_{i \in [n]}$ correspond to a MHR distribution P, then for $i \in [n]$

$$P([i, n]) = \prod_{j=1}^{i-1} e^{y_j} = e^{\sum_{j=1}^{i-1} y_j}$$

and asking that $\ln(1 - \varepsilon) \leq \sum_{j=1}^{i-1} y_j - \ln \mathbf{p}([i, n]) \leq \ln(1 + \varepsilon)$ amounts to requiring

$$P([i, n]) \in [1 \pm \varepsilon]\mathbf{p}([i, n]).$$

We focus first on the completeness case, to provide intuition for the linear program. Suppose there exists $P \in \mathcal{MHR}_n$ such $P \in \mathcal{MHR}_n$ such that $\|\mathbf{p} - P\|_1 \leq \varepsilon$ and $\mathrm{d_K}(\mathbf{p}', P) \leq \alpha$. This implies that for all $i \in [n]$, $|P([i, n]) - \mathbf{p}([i, n])| \leq 2\alpha$. Define $I = \{b + 1, \ldots, n\}$ to be the longest interval such that $\mathbf{p}(\{b + 1, \ldots, n\}) \leq \frac{\varepsilon}{2}$. It follows that for every $i \in [n] \setminus I$,

$$\frac{P([i, n])}{\mathbf{p}([i, n])} \leq \frac{\mathbf{p}([i, n]) + 2\alpha}{\mathbf{p}([i, n])} \leq 1 + \frac{2\alpha}{\varepsilon/2} = 1 + 4\varepsilon^2 \leq 1 + \varepsilon \tag{2.10}$$

and similarly $\frac{P([i,n])}{\mathbf{p}([i,n])} \geq \frac{\mathbf{p}([i,n])-2\alpha}{\mathbf{p}([i,n]} \geq 1-\varepsilon$. This means that for the points i in $[n] \setminus I$, we can write constraints asking for multiplicative closeness (within $1 \pm \varepsilon$) between $e^{\sum_{j=1}^{i-1} y_j}$ and $\mathbf{p}([i,n])$, which is very easy to write down as linear constraints on the y_i's.

The linear program Let T and S be respectively the sets of "light" and "heavy" points, defined as $T = \left\{ i \in \{1,\ldots,b\} : \mathbf{p}(i) \leq \varepsilon^2 \right\}$ and $S = \left\{ i \in \{1,\ldots,b\} : \mathbf{p}(i) > \varepsilon^2 \right\}$, where b is as above. (In particular, $|S| \leq 1/\varepsilon^2$.)

Algorithm 7 Linear Program

Find $y_1, \ldots, y_b$

s.t.

$$y_i \leq 0 \tag{2.11}$$

$$y_{i+1} \leq y_i \qquad \forall i \in \{1,\ldots,b-1\} \tag{2.12}$$

$$\ln(1-\varepsilon) \leq \sum_{j=1}^{i-1} y_j - \ln \mathbf{p}([i,n]) \leq \ln(1+\varepsilon) \qquad \forall i \in \{1,\ldots,b\} \tag{2.13}$$

$$\frac{\mathbf{p}(i)-\varepsilon_i}{(1+\varepsilon)\mathbf{p}[i,n]} \leq -y_i \leq (1+4\varepsilon)\frac{\mathbf{p}(i)+\varepsilon_i}{(1-\varepsilon)\mathbf{p}[i,n]} \qquad \forall i \in T \tag{2.14}$$

$$\sum_{i \in T} \varepsilon_i \leq \varepsilon \tag{2.15}$$

$$0 \leq \varepsilon_i \leq 2\alpha \qquad \forall i \in T \tag{2.16}$$

$$\ln\left(1 - \frac{\mathbf{p}(i)+2\alpha}{(1-\varepsilon)\mathbf{p}[i,n]}\right) \leq y_i \leq \ln\left(1 - \frac{\mathbf{p}(i)-2\alpha}{(1+\varepsilon)\mathbf{p}[i,n]}\right) \qquad \forall i \in S \tag{2.17}$$

Given a solution to the linear program above, define $\tilde{P}$ (a non-normalized probability distribution) by setting $\tilde{P}(i) = (1-e^{y_i})e^{\sum_{j=1}^{i-1} y_j}$ for $i \in \{1,\ldots,b\}$, and $\tilde{P}(i) = 0$ for $i \in I = \{b+1,\ldots,n\}$. A MHR distribution is then obtained by normalizing $\tilde{P}$.

Completeness Suppose $P \in \mathcal{MHR}_n$ is as promised. In particular, by the Kolmogorov distance assumption we know that every $i \in T$ has $P(i) \leq \varepsilon^2 + 2\alpha < 2\varepsilon^2$.

- For any $i \in T$, we have that $\frac{P(i)}{P[i,n]} \leq \frac{2\varepsilon^2}{(1-\varepsilon)\varepsilon} \leq 4\varepsilon$, and

$$\frac{\mathbf{p}(i)-\varepsilon_i}{(1+\varepsilon)\mathbf{p}[i,n]} \leq \frac{P(i)}{P[i,n]} \leq \underbrace{-\ln(1-\frac{P(i)}{P[i,n]})}_{-y_i} \leq (1+4\varepsilon)\frac{P(i)}{P[i,n]} = (1+4\varepsilon)\frac{\mathbf{p}(i)+\varepsilon_i}{P[i,n]} \leq \frac{1+4\varepsilon}{1-\varepsilon}\frac{\mathbf{p}(i)+\varepsilon_i}{\mathbf{p}[i,n]} \tag{2.18}$$

  where we used Eq. (2.10) for the two outer inequalities; and so (2.14), (2.15), and (2.16) would follow from setting $\varepsilon_i \stackrel{\text{def}}{=} |P(i) - \mathbf{p}(i)|$ (along with the guarantees on ℓ_1 and Kolmogorov distances between P and $\mathbf{p}$).

- For $i \in S$, Constraint (2.17) is also met, as $\frac{P(i)}{P([i,n])} \in \left[\frac{\mathbf{p}(i)-2\alpha}{P([i,n])}, \frac{\mathbf{p}(i)+2\alpha}{P([i,n])}\right] \subseteq \left[\frac{\mathbf{p}(i)-2\alpha}{(1+\varepsilon)\mathbf{p}([i,n])}, \frac{\mathbf{p}(i)+2\alpha}{(1-\varepsilon)\mathbf{p}([i,n])}\right]$.

Soundness Assume a feasible solution to the linear program is found. We argue that this implies $\mathbf{p}$ is $O(\varepsilon)$-close to some MHR distribution, namely to the distribution obtained by renormalizing $\tilde{P}$.

In order to do so, we bound separately the ℓ_1 distance between $\mathbf{p}$ and $\tilde{P}$, from I, S, and T. First, $\sum_{i\in I}\left|\mathbf{p}(i)-\tilde{P}(i)\right| = \sum_{i\in I}\mathbf{p}(i) \le \frac{\varepsilon}{2}$ by construction. For $i\in T$, we have $\frac{\mathbf{p}(i)}{\mathbf{p}[i,n]} \le \varepsilon$, and

$$\tilde{P}(i) = (1-e^{y_i})e^{\sum_{j=1}^{i-1} y_j} \in [1\pm\varepsilon]\,(1-e^{y_i})\mathbf{p}([i,n]).$$

Now,

$$1-(1-\varepsilon)\frac{\mathbf{p}(i)-\varepsilon_i}{(1+\varepsilon)\mathbf{p}[i,n]} \ge e^{-\frac{\mathbf{p}(i)-\varepsilon_i}{(1+\varepsilon)\mathbf{p}[i,n]}} \ge e^{y_i} \ge e^{-(1+4\varepsilon)\frac{\mathbf{p}(i)+\varepsilon_i}{(1-\varepsilon)\mathbf{p}[i,n]}} \ge 1-(1+4\varepsilon)\frac{\mathbf{p}(i)+\varepsilon_i}{(1-\varepsilon)\mathbf{p}[i,n]}$$

so that

$$(1-\varepsilon)\frac{(1-\varepsilon)}{(1+\varepsilon)}(\mathbf{p}(i)-\varepsilon_i) \le \tilde{P}(i) \le (1+4\varepsilon)\frac{(1+\varepsilon)}{(1-\varepsilon)}(\mathbf{p}(i)+\varepsilon_i)$$

which implies

$$(1-10\varepsilon)(\mathbf{p}(i)-\varepsilon_i) \le \tilde{P}(i) \le (1+10\varepsilon)(\mathbf{p}(i)+\varepsilon_i)$$

so that $\sum_{i\in T}\left|\mathbf{p}(i)-\tilde{P}(i)\right| \le 10\varepsilon\sum_{i\in T}\mathbf{p}(i) + (1+10\varepsilon)\sum_{i\in T}\varepsilon_i \le 10\varepsilon + (1+10\varepsilon)\varepsilon \le 20\varepsilon$ where the last inequality follows from Constraint (2.15).

To analyze the contribution from S, we observe that Constraint (2.17) implies that, for any $i\in S$,

$$\frac{\mathbf{p}(i)-2\alpha}{(1+\varepsilon)\mathbf{p}([i,n])} \le \frac{\tilde{P}(i)}{\tilde{P}([i,n])} \le \frac{\mathbf{p}(i)+2\alpha}{(1-\varepsilon)\mathbf{p}([i,n])}$$

which combined with Constraint (2.13) guarantees

$$\frac{\mathbf{p}(i)-2\alpha}{(1+\varepsilon)^2\tilde{P}([i,n])} \le \frac{\tilde{P}(i)}{\tilde{P}([i,n])} \le \frac{\mathbf{p}(i)+2\alpha}{(1-\varepsilon)^2\tilde{P}([i,n])}$$

which in turn implies that $\left|\tilde{P}(i)-\mathbf{p}(i)\right| \le 3\varepsilon\tilde{P}(i)+2\alpha$. Recalling that $|S| \le \frac{1}{\varepsilon^2}$ and $\alpha=\varepsilon^3$, this yields $\sum_{i\in S}\left|\mathbf{p}(i)-\tilde{P}(i)\right| \le 3\varepsilon\sum_{i\in S}\tilde{P}(i)+2\varepsilon \le 3\varepsilon(1+\varepsilon)+2\varepsilon \le 8\varepsilon$. Summing up, we get $\sum_{i=1}^{n}\left|\mathbf{p}(i)-\tilde{P}(i)\right| \le 30\varepsilon$ which finally implies by the triangle inequality that the ℓ_1 distance between $\mathbf{p}$ and the normalized version of $\tilde{P}$ (a valid MHR distribution) is at most 32ε.

Running time The running time is immediate, from executing the two linear programs on $\mathrm{poly}(n,1/\varepsilon)$ variables and constraints. □

2.1.7.3 Proof of Lemma 2.1.31

Lemma 2.1.31 (Log-concavity). *There exists a procedure* $\textsc{ProjectionDist}^{*}_{\mathcal{LCV}_n}$ *that, on input n as well as the full specifications of a k-histogram distribution $\mathbf{p}$ on $[n]$ and an ℓ-histogram distribution $\mathbf{p}'$ on $[n]$, runs in time* $\mathrm{poly}(n,k,\ell,1/\varepsilon)$*, and satisfies the following.*

- *If there is* $P \in \mathcal{LCV}_n$ *such that* $\|\mathbf{p} - P\|_1 \leq \varepsilon$ and $\mathrm{d_K}(\mathbf{p}', P) \leq \frac{\varepsilon^2}{\log^2(1/\varepsilon)}$, *then the procedure returns* yes*;*
- *If* $\ell_1(\mathbf{p}, \mathcal{LCV}_n) \geq 100\varepsilon$, *then the procedure returns* no*.*

Proof. We set $\alpha \stackrel{\text{def}}{=} \frac{\varepsilon^2}{\log^2(1/\varepsilon)}$, $\beta \stackrel{\text{def}}{=} \frac{\varepsilon^2}{\log(1/\varepsilon)}$, and $\gamma \stackrel{\text{def}}{=} \frac{\varepsilon^2}{10}$ (so that $\alpha \ll \beta \ll \gamma \ll \varepsilon$),

Given the explicit description of a distribution $\mathbf{p}$ on $[n]$, which a k-histogram over a partition $\mathcal{I} = (I_1, \ldots, I_k)$ of $[n]$ with $k = \mathrm{poly}(\log n, 1/\varepsilon)$ and the explicit description of a distribution $\mathbf{p}'$ on $[n]$, one must *efficiently* distinguish between:

(a) $\mathbf{p}$ is ε-close to a log-concave P (in ℓ_1 distance) *and* $\mathbf{p}'$ is α-close to P (in Kolmogorov distance); and

(b) $\mathbf{p}$ is 100ε-far from log-concave.

If we are willing to pay an extra factor of $O(n)$, we can assume without loss of generality that we know the mode of the closest log-concave distribution (which is implicitly assumed in the following: the final algorithm will simply try all possible modes).

Outline First, we argue that we can simplify to the case where $\mathbf{p}$ is unimodal. Then, reduce to the case where where $\mathbf{p}$ and $\mathbf{p}'$ are only one distribution, satisfying both requirements from the completeness case. Both can be done efficiently (Section 2.1.7.3), and make the rest much easier. Then, perform some *ad hoc* partitioning of $[n]$, using our knowledge of $\mathbf{p}$, into $\tilde{O}\left(1/\varepsilon^2\right)$ pieces such that each piece is either a "heavy" singleton, or an interval I with weight very close (multiplicatively) to $\mathbf{p}(I)$ *under the target log-concave distribution, if it exists* (Section 2.1.7.3). This in particular simplifies the type of log-concave distribution we are looking for: it is sufficient to look for distributions putting that very specific weight on each piece, up to a $(1 + o(1))$ factor. Then, in Section 2.1.7.3, we write and solve a linear program to try and find such a "simplified" log-concave distribution, and reject if no feasible solution exists.

Note that the first two sections allow us to argue that instead of additive (in ℓ_1) closeness, we can enforce constraints on *multiplicative* (within a $(1 + \varepsilon)$ factor) closeness between $\mathbf{p}$ and the target log-concave distribution. This is what enables a linear program with variables being the logarithm of the probabilities, which plays very nicely with the log-concavity constraints.

We will require the following result of Chan, Diakonikolas, Servedio, and Sun:

Theorem 2.1.51 ([56, Lemma 4.1])**.** *Let* $\mathbf{p}$ *be a distribution over* $[n]$*, log-concave and non-decreasing over* $\{1, \ldots, b\} \subseteq [n]$*. Let* $a \leq b$ *such that* $\sigma = \mathbf{p}(\{1, \ldots, a-1\}) > 0$*, and write* $\tau = \mathbf{p}(\{a, \ldots, b\})$*. Then* $\frac{\mathbf{p}(b)}{\mathbf{p}(a)} \leq 1 + \frac{\tau}{\sigma}$.

Step 1

Reducing to $\mathbf{p}$ unimodal Using a linear program, find a closest *unimodal* distribution $\tilde{\mathbf{p}}$ to $\mathbf{p}$ (also a k-histogram on $\mathcal{I}$) under the constraint that $\mathrm{d_K}(\mathbf{p}, P) \leq \alpha$: this can be done in time $\mathrm{poly}(k)$. If $\|\mathbf{p} - \tilde{\mathbf{p}}\|_1 > \varepsilon$, output reject.

- If $\mathbf{p}$ is ε-close to a log-concave distribution P as above, then it is in particular ε-close to unimodal and we do not reject. Moreover, by the triangle inequality $\|\tilde{\mathbf{p}} - P\|_1 \leq 2\varepsilon$ and $\mathrm{d_K}(\tilde{\mathbf{p}}, P) \leq 2\alpha$.
- If $\mathbf{p}$ is 100ε-far from log-concave and we do not reject, then $\ell_1(\tilde{\mathbf{p}}, \mathcal{LCV}_n) \geq 99\varepsilon$.

Reducing to $\mathbf{p} = \mathbf{p}'$ First, we note that it is easy to reduce our problem to the case where, in the completeness case, we have $P \in \mathcal{LCV}_n$ such that $\|\mathbf{p} - P\|_1 \leq 4\varepsilon$ and $\mathrm{d_K}(\mathbf{p}, P) \leq 4\alpha$; while in the soundness case $\ell_1(\mathbf{p}, \mathcal{LCV}_n) \geq 97\varepsilon$. Indeed, this can be done with a linear program on $\mathrm{poly}(k, \ell)$ variables and constraints, asking to find a $(k+\ell)$-histogram $\mathbf{p}''$ on a refinement of $\mathbf{p}$ and $\mathbf{p}'$ minimizing the ℓ_1 distance to $\mathbf{p}$, under the constraint that the Kolmogorov distance to $\mathbf{p}'$ be bounded by 2α. (In the completeness case, clearly a feasible solution exists, as (the flattening on this $(k+\ell)$-interval partition) of P is one.) We therefore follow with this new formulation: either

(a) $\mathbf{p}$ is 4ε-close to a log-concave P (in ℓ_1 distance) *and* $\mathbf{p}$ is 4α-close to P (in Kolmogorov distance); and

(b) $\mathbf{p}$ is 97ε-far from log-concave;

where $\mathbf{p}$ is a $(k+\ell)$-histogram.

This way, we have reduced the problem to a slightly more convenient one, that of Section 2.1.7.3.

Reducing to knowing the support $[a, b]$ The next step is to compute a good approximation of the support of any target log-concave distribution. This is easily obtained in time $O(k)$ as the interval $\{a, \cdots, b\}$ such that

- $\mathbf{p}(\{1, \ldots, a-1\}) \leq \alpha$ but $\mathbf{p}(\{1, \ldots, a\}) > \alpha$; and
- $\mathbf{p}(\{b+1, \ldots, \}n) \leq \alpha$ but $\mathbf{p}(\{b, \ldots, n\}) > \alpha$.

Any log-concave distribution that is α-close to $\mathbf{p}$ must include $\{a, \cdots, b\}$ in its support, since otherwise the ℓ_1 distance between $\mathbf{p}$ and P is already greater than α. Conversely, if P is a log-concave distribution α-close to $\mathbf{p}$, it is easy to see that the distribution obtained by setting P to be zero outside $\{a, \cdots, b\}$ and renormalizing the result is still log-concave, and $O(\alpha)$-close to $\mathbf{p}$.

Step 2 Given the explicit description of a *unimodal* distribution $\mathbf{p}$ on $[n]$, which a k-histogram over a partition $\mathcal{I} = (I_1, \ldots, I_k)$ of $[n]$ with $k = \mathrm{poly}(\log n, 1/\varepsilon)$, one must *efficiently* distinguish between:

(a) $\mathbf{p}$ is ε-close to a log-concave P (in ℓ_1 distance) and α-close to P (in Kolmogorov distance); and

(b) $\mathbf{p}$ is 24ε-far from log-concave,

assuming we know the mode of the closest log-concave distribution, which has support $[n]$.

In this stage, we compute a partition $\mathcal{J}$ of $[n]$ into $\tilde{O}\big(1/\varepsilon^2\big)$ intervals (here, we implicitly use the knowledge of the mode of the closest log-concave distribution, in order to apply Theorem 2.1.51 differently on two intervals of the support, corresponding to the non-decreasing and non-increasing parts of the target log-concave distribution).

As $\mathbf{p}$ is unimodal, we can efficiently ($O(\log k)$) find the interval S of heavy points, that is

$$S \stackrel{\text{def}}{=} \{\, x \in [n] :\ \mathbf{p}(x) \geq \beta \,\}.$$

Each point in S will form a singleton interval in our partition. Let $T \stackrel{\text{def}}{=} [n] \setminus S$ be its complement (T is the union of at most two intervals T_1, T_2 on which $\mathbf{p}$ is monotone, the head and tail of the distribution). For convenience, we focus on only one of these two intervals, without loss of generality the "head" T_1 (on which $\mathbf{p}$ is non-decreasing).

1. Greedily find $J = \{1, \dots, a\}$, the smallest prefix of the distribution satisfying $\mathbf{p}(J) \in \left[\frac{\varepsilon}{10} - \beta, \frac{\varepsilon}{10}\right]$.
2. Similarly, partition $T_1 \setminus J$ into intervals $I'_1, \dots, I'_s$ (with $s = O(1/\gamma) = O\big(1/\varepsilon^2\big)$) such that $\frac{\gamma}{10} \leq \mathbf{p}(I'_j) \leq \frac{9}{10}\gamma$ for all $1 \leq j \leq s-1$, and $\frac{\gamma}{10} \leq \mathbf{p}(I'_s) \leq \gamma$. This is possible as all points not in S have weight less than β, and $\beta \ll \gamma$.

Discussion: why doing this? We focus on the completeness case: let $P \in \mathcal{LCV}_n$ be a log-concave distribution such that $\|\mathbf{p} - P\|_1 \leq \varepsilon$ and $\mathrm{d_K}(\mathbf{p}, P) \leq \alpha$. Applying Theorem 2.1.51 on J and the I'_j's, we obtain (using the fact that $\big|P(I'_j) - \mathbf{p}(I'_j)\big| \leq 2\alpha$) that:

$$\frac{\max_{x \in I'_j} P(x)}{\min_{x \in I'_j} P(x)} \leq 1 + \frac{\mathbf{p}(I'_j) + 2\alpha}{\mathbf{p}(J) - 2\alpha} \leq 1 + \frac{\gamma + 2\alpha}{\frac{\varepsilon}{10} - 2\alpha} = 1 + \varepsilon + O\left(\frac{\varepsilon^2}{\log^2(1/\varepsilon)}\right) \stackrel{\text{def}}{=} 1 + \kappa.$$

Moreover, we also get that each resulting interval I'_j will satisfy

$$\mathbf{p}(I'_j)(1 - \kappa_j) = \mathbf{p}(I'_j) - 2\alpha \leq P(I'_j) \leq \mathbf{p}(I'_j) + 2\alpha = \mathbf{p}(I'_j)(1 + \kappa_j)$$

with $\kappa_j \stackrel{\text{def}}{=} \frac{2\alpha}{\mathbf{p}(I'_j)} = \Theta\big(1/\log^2(1/\varepsilon)\big)$.

Summing up, we have a partition of $[n]$ into $|S| + 2 = \tilde{O}\big(1/\varepsilon^2\big)$ intervals such that:

- The (at most) two end intervals have $\mathbf{p}(J) \in \left[\frac{\varepsilon}{10} - \beta, \frac{\varepsilon}{10}\right]$, and thus $P(J) \in \left[\frac{\varepsilon}{10} - \beta - 2\alpha, \frac{\varepsilon}{10} + 2\alpha\right]$;
- the $\tilde{O}\big(1/\varepsilon^2\big)$ singleton-intervals from S are points x with $\mathbf{p}(x) \geq \beta$, so that $P(x) \geq \beta - 2\alpha \geq \frac{\beta}{2}$;
- each other interval $I = I'_j$ satisfies

$$(1 - \kappa_j)\mathbf{p}(I) \leq P(I) \leq (1 + \kappa_j)\mathbf{p}(I) \tag{2.19}$$

with $\kappa_j = O\big(1/\log^2(1/\varepsilon)\big)$; and

$$\frac{\max_{x \in I} P(x)}{\min_{x \in I} P(x)} \leq 1 + \kappa < 1 + \frac{3}{2}\varepsilon. \tag{2.20}$$

We will use in the constraints of the linear program the fact that $(1 + \frac{3}{2}\varepsilon)(1 + \kappa_j) \leq 1 + 2\varepsilon$, and $\frac{1 - \kappa_j}{1 + \frac{3}{2}\varepsilon} \geq \frac{1}{1 + 2\varepsilon}$.

Step 3 We start by computing the partition $\mathcal{J} = (J_1, \dots, J_\ell)$ as in Section 2.1.7.3; with $\ell = \tilde{O}(1/\varepsilon^2)$; and write $J_j = \{a_j, \dots, b_j\}$ for all $j \in [\ell]$. We further denote by S and T the set of heavy and light points, following the notations from Section 2.1.7.3; and let $T' \stackrel{\text{def}}{=} T_1 \cup T_2$ be the set obtained by removing the two "end intervals" (called J in the previous section) from T.

Algorithm 8 Linear Program

Find $x_1, \dots, x_n, \varepsilon_1, \dots, \varepsilon_{|S|}$

s.t.

$$x_i \leq 0 \tag{2.21}$$

$$x_i - x_{i-1} \geq x_{i+1} - x_i \qquad \forall i \in [n] \tag{2.22}$$

$$-\ln(1+2\varepsilon) \leq x_i - \mu_j \leq \ln(1+2\varepsilon), \qquad \forall j \in T', \forall i \in J_j \tag{2.23}$$

$$-2\frac{\varepsilon_i}{\mathbf{p}(i)} \leq x_i - \ln \mathbf{p}(i) \leq \frac{\varepsilon_i}{\mathbf{p}(i)}, \qquad \forall i \in S \tag{2.24}$$

$$\sum_{i \in S} \varepsilon_i \leq \varepsilon \tag{2.25}$$

$$0 \leq \varepsilon_i \leq 2\alpha \qquad \forall i \in S \tag{2.26}$$

(2.27)

where $\mu_j \stackrel{\text{def}}{=} \ln \frac{\mathbf{p}(J_j)}{|J_j|}$ for $j \in T'$.

Lemma 2.1.52 (Soundness)**.** *If the linear program (Algorithm 8) has a feasible solution, then* $\ell_1(\mathbf{p}, \mathcal{LCV}_n) \leq O(\varepsilon)$.

Proof. A feasible solution to this linear program will define (setting $p_i = e^{x_i}$) a sequence $p = (p_1, \dots, p_n) \in (0,1]^n$ such that

- p takes values in $(0, 1]$ (from (2.21));
- p is log-concave (from (2.22));
- p is "$(1 + O(\varepsilon))$-multiplicatively constant" on each interval J_j (from (2.23));
- p puts roughly the right amount of weight on each J_i:
  - weight $(1 \pm O(\varepsilon))\mathbf{p}(J)$ on every J from T (from (2.23)), so that the ℓ_1 distance between $\mathbf{p}$ and p coming from T' is at most $O(\varepsilon)$;
  - it puts weight approximately $\mathbf{p}(J)$ on every singleton J from S, i.e. such that $\mathbf{p}(J) \geq \beta$. To see why, observe that each ε_i is in $[0, 2\alpha]$ by constraints (2.26). In particular, this means that $\frac{\varepsilon_i}{\mathbf{p}(i)} \leq 2\frac{\alpha}{\beta} \ll 1$, and we have

$$\mathbf{p}(i) - 4\varepsilon_i \leq \mathbf{p}(i) \cdot e^{-4\frac{\varepsilon_i}{\mathbf{p}(i)}} \leq p_i = e^{x_i} \leq \mathbf{p}(i) \cdot e^{2\frac{\varepsilon_i}{\mathbf{p}(i)}} \leq \mathbf{p}(i) + 4\varepsilon_i$$

  and together with (2.25) this guarantees that the ℓ_1 distance between $\mathbf{p}$ and p coming from S is at most ε.

Note that the solution obtained this way may not sum to one – i.e., is not necessarily a probability distribution. However, it is easy to renormalize p to obtain a *bona fide* probability distribution $\tilde{P}$ as follows: set $\tilde{P} = \frac{p(i)}{\sum_{i \in S \cup T'} p(i)}$ for all $i \in S \cup T'$, and $p(i) = 0$ for $i \in T \setminus T'$.

Since by the above discussion we know that $p(S \cup T')$ is within $O(\varepsilon)$ of $\mathbf{p}(S \cup T')$ (itself in $[1 - \frac{9\varepsilon}{5}, 1 + \frac{9\varepsilon}{5}]$ by construction of T'), $\tilde{P}$ is a log-concave distribution such that $\|\tilde{P} - \mathbf{p}\|_1 = O(\varepsilon)$. □

Lemma 2.1.53 (Completeness)**.** *If there is P in $\mathcal{LCV}_n$ such that $\|\mathbf{p} - P\|_1 \leq \varepsilon$ and $\mathrm{d_K}(\mathbf{p}, P) \leq \alpha$, then the linear program (Algorithm 8) has a feasible solution.*

Proof. Let $P \in \mathcal{LCV}_n$ such that $\|\mathbf{p} - P\|_1 \leq \varepsilon$ and $\mathrm{d_K}(\mathbf{p}, P) \leq \alpha$. Define $x_i \stackrel{\text{def}}{=} \ln P(i)$ for all $i \in [n]$. Constraints (2.21) and (2.22) are immediately satisfied, since P is log-concave. By the discussion from Section 2.1.7.3 (more specifically, Eq. (2.19) and (2.20)), constraint (2.23) holds as well.

Letting $\varepsilon_i \stackrel{\text{def}}{=} |P(i) - \mathbf{p}(i)|$ for $i \in S$, we also immediately have (2.25) and (2.26) (since $\|P - \mathbf{p}\|_1 \leq \varepsilon$ and $\mathrm{d_K}(\mathbf{p}, P) \leq \alpha$ by assumption). Finally, to see why (2.24) is satisfied, we rewrite

$$x_i - \ln \mathbf{p}(i) = \ln \frac{P(i)}{\mathbf{p}(i)} = \ln \frac{\mathbf{p}(i) \pm \varepsilon_i}{\mathbf{p}(i)} = \ln(1 \pm \frac{\varepsilon_i}{\mathbf{p}(i)})$$

and use the fact that $\ln(1+x) \leq x$ and $\ln(1-x) \geq -2x$ (the latter for $x < \frac{1}{2}$, along with $\frac{\varepsilon_i}{\mathbf{p}(i)} \leq \frac{2\alpha}{\beta} \ll 1$). □

Putting it all together: Proof of Lemma 2.1.31 The algorithm is as follows (keeping the notations from Section 2.1.7.3 to Section 2.1.7.3):

- Set α, β, γ as above.
- Follow Section 2.1.7.3 to reduce it to the case where $\mathbf{p}$ is unimodal and satisfies the conditions for Kolmogorov and ℓ_1 distance; and a good $[a, b]$ approximation of the support is known
- For each of the $O(n)$ possible modes $c \in [a, b]$:
  - Run the linear program Algorithm 8, return accept if a feasible solution is found
- None of the linear programs was feasible: return reject.

The correctness comes from Lemma 2.1.52 and Lemma 2.1.53 and the discussions in Section 2.1.7.3 to Section 2.1.7.3; as for the claimed running time, it is immediate from the algorithm and the fact that the linear program executed each step has $\mathrm{poly}(n, 1/\varepsilon)$ constraints and variables.

□

2.2 The Fourier Knife

The upper bound framework presented in the previous section relied on a *shape* condition on the distributions of the property to be tested, which made it particularly appealing when considering shape-restricted properties such as monotonicity or convexity. In this section, we pursue a different direction, focusing on a completely orthogonal type of structural property: namely, one capturing the *sparsity* of the (discrete) Fourier transform.

As an illustration of the difference between the two approaches: monotone distributions admit, as we saw earlier, very succinct decompositions (in the sense of Definition 2.1.13); yet in general their discrete Fourier transform need not be sparse by any means. On the other hand, it is easy to see that, for $k > 2$, (n, k)-SIIRVs cannot be well-approximated by succinct decompositions; however, as we shall see they enjoy very good sparsity in the "Fourier world."

Our two Swiss Army Knife approaches – the shape restriction one from the previous section, and the "Fourier knife" we are about to describe – can thus be seen as complementary; in conjunction, they provide a thorough and widely applicable toolbox to tackle distribution testing questions.

2.2.1 Introduction

As before, let $\mathcal{P}$ be a family of discrete distributions over a total order (e.g., $[n]$) or a partial order (e.g., $[n]^k$). Recall that the problem of *membership testing for* $\mathcal{P}$ is the following: Given sample access to an unknown distribution $\mathbf{p}$ (effectively supported on the same domain as $\mathcal{P}$), we want to distinguish between the case that $\mathbf{p} \in \mathcal{P}$ versus $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathcal{P}) \geq \varepsilon$. Clearly, the sample complexity of this problem depends on the underlying family $\mathcal{P}$. For example, if $\mathcal{P}$ contains a single distribution over a domain of size n, the sample complexity of the testing problem is $\Theta\big(n^{1/2}/\varepsilon^2\big)$ [138, 58, 82]; while if $\mathcal{P}$ is the set of *all* probability distributions over this domain, the sample complexity drops quite drastically to zero. Thus, in view of Problem 2.0.1 our goal is to abstract a minimal set of structural assumptions on $\mathcal{P}$ that captures this sample complexity.

We give a general technique to test membership in various distribution families over discrete domains, based on properties of the *Fourier spectrum* of the distributions they contain. Before we state our results in full generality, we present concrete applications to a number of well-studied distribution families.

2.2.1.1 Our Results

Our first concrete application is a nearly sample-optimal algorithm for testing sums of independent integer random variables (SIIRVs). Recall from Section 1.3 that an (n, k)-SIIRV is a sum of independent integer random variables, each supported in $[\![k]\!] = \{0, \ldots, k-1\}$. SIIRVs comprise a rich class of distributions that arise in many settings. The special case of $k = 2$, $\mathcal{SIIRV}_{n,2}$, was first considered by Poisson [142] as a non-trivial extension of the Binomial distribution, and is known as Poisson binomial distribution (PBD). In application domains, SIIRVs have many uses in research areas such as survey sampling, case-control studies, and survival analysis, see e.g., [60] for a survey of the many practical uses of these distributions. We remark that these distributions are of fundamental interest and have been extensively studied in probability and statistics [62, 112, 89, 144, 123, 16, 59]. We show the following:

Theorem 2.2.1 (Testing SIIRVs)**.** *Given parameters $k, n \in \mathbb{N}$, $\varepsilon \in (0, 1]$, and sample access to a distribution $\mathbf{p}$ over $\mathbb{N}$, there exists an algorithm (Algorithm 9) which outputs either* accept *or* reject*, and satisfies the following:*

1. *if $\mathbf{p} \in \mathcal{SIIRV}_{n,k}$, then it outputs* accept *with probability at least $3/5$;*

2. *if* $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathcal{SIIRV}_{n,k}) > \varepsilon$*, then it outputs* reject *with probability at least* $3/5$.

Moreover, the algorithm takes $O\left(\frac{kn^{1/4}}{\varepsilon^2}\log^{1/4}\frac{1}{\varepsilon} + \frac{k^2}{\varepsilon^2}\log^2\frac{k}{\varepsilon}\right)$ *samples from* $\mathbf{p}$*, and runs in time* $n(k/\varepsilon)^{O(k\log(k/\varepsilon))}$.

Prior to our work, no non-trivial tester was known for (n,k)-SIIRVs for any $k > 2$. Canonne et al. [51] showed a sample lower bound of $\Omega\left(\frac{k^{1/2}n^{1/4}}{\varepsilon^2}\right)$, that we shall cover in Chapter 3; however, their techniques did not yield a corresponding sample upper bound. The special case of PBDs ($k = 2$) was studied by Acharya and Daskalakis [2] who obtained a tester with sample complexity $O\left(\frac{n^{1/4}}{\varepsilon^2}\sqrt{\log 1/\varepsilon} + \frac{\log^{5/2} 1/\varepsilon}{\varepsilon^6}\right)$ (and running time $O\left(\frac{n^{1/4}}{\varepsilon^2}\sqrt{\log 1/\varepsilon} + (1/\varepsilon)^{O(\log^2 1/\varepsilon)}\right)$ and a sample lower bound of $\Omega(n^{1/4}/\varepsilon^2)$. Our techniques also yield the following corollary:

Theorem 2.2.2 (Testing PBDs)**.** *Given parameters* $n \in \mathbb{N}$*,* $\varepsilon \in (0,1]$*, and sample access to a distribution* $\mathbf{p}$ *over* $\mathbb{N}$*, there exists an algorithm (Algorithm 9) which outputs either* accept *or* reject*, and satisfies the following.*

1. *if* $\mathbf{p} \in \mathcal{PBD}_n$*, then it outputs* accept *with probability at least* $3/5$*;*
2. *if* $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathcal{PBD}_n) > \varepsilon$*, then it outputs* reject *with probability at least* $3/5$.

Moreover, the algorithm takes $O\left(\frac{n^{1/4}}{\varepsilon^2}\log^{1/4}\frac{1}{\varepsilon} + \frac{\log^2 1/\varepsilon}{\varepsilon^2}\right)$ *samples from* $\mathbf{p}$*, and runs in time* $n^{1/4}\cdot\tilde{O}(1/\varepsilon^2) + (1/\varepsilon)^{O(\log\log(1/\varepsilon))}$.

The sample complexity in the theorem above follows from Theorem 2.2.1, for $k = 2$. The improved running time relies on a more efficient computational "projection step" in our general framework, which builds on the geometric structure of Poisson Binomial distributions and allows us to avoid an $(1/\varepsilon)^{O(\log(1/\varepsilon))}$ dependence. In summary, as a special case of Theorem 2.2.1, we obtain a tester for PBDs whose sample complexity is optimal as a function of both n and $1/\varepsilon$ (up to a logarithmic factor).

We further remark that the guarantees provided by the above two theorems are actually stronger than the usual property testing one; namely, whenever the algorithm returns accept, then it also provides a (proper) hypothesis $\mathbf{h}$ such that $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{h}) \leq \varepsilon$ with probability at least $3/5$.

An alternate generalization of PBDs to the high-dimensional setting is the family of Poisson Multinomial Distributions (PMDs). Formally, an (n,k)-PMD is any random variable of the form $X = \sum_{i=1}^{n} X_i$, where the X_i's are independent random vectors supported on the set $\{e_1, e_2, \ldots, e_k\}$ of standard basis vectors in $\mathbb{R}^k$. PMDs comprise a broad class of discrete distributions of fundamental importance in computer science, probability, and statistics. A large body of work in the probability and statistics literature has been devoted to the study of the behavior of PMDs under various structural conditions [15, 126, 16, 26, 152, 151]. PMDs generalize the familiar multinomial distribution, and describe many distributions commonly encountered in computer science (see, e.g., [69, 70, 174, 171]). Recent years have witnessed a flurry of research activity on PMDs and related distributions, from several perspectives of theoretical computer science, including learning [65, 73, 85, 67, 86], property testing [174, 167, 171], computational game theory [69, 70, 37, 71, 68, 99, 61], and derandomization [106, 27, 75, 105].

Theorem 2.2.3 (Testing PMDs)**.** *Given parameters $k, n \in \mathbb{N}$, $\varepsilon \in (0,1]$, and sample access to a distribution $\mathbf{p}$ over $\mathbb{N}$, there exists an algorithm (Algorithm 15) which outputs either* accept *or* reject*, and satisfies the following.*

1. *if $\mathbf{p} \in \mathcal{PMD}_{n,k}$, then it outputs* accept *with probability at least $3/5$;*
2. *if $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathcal{PMD}_{n,k}) > \varepsilon$, then it outputs* reject *with probability at least $3/5$.*

Moreover, the algorithm takes $O\left(\frac{n^{(k-1)/4}k^{2k}\log(k/\varepsilon)^k}{\varepsilon^2}\right)$ samples from $\mathbf{p}$, and runs in time $n^{O(k^3)}\cdot(1/\varepsilon)^{O(k^3\frac{\log(k/\varepsilon)}{\log\log(k/\varepsilon)})^{k-1}}$ or alternatively in time $n^{O(k)}\cdot 2^{O(k^{5k}\log(1/\varepsilon)^{k+2})}$.

We also show a nearly matching sample lower bound[5] of $\Omega_k(n^{(k-1)/4}/\varepsilon^2)$ (Theorem 2.2.28). Finally, we demonstrate the versatility of our techniques by obtaining in Section 2.2.7 a testing algorithm for discrete log-concavity with sample complexity $O(\sqrt{n}/\varepsilon^2 + (\log(1/\varepsilon)/\varepsilon)^{5/2})$; improving on the previous bounds of $O\big(\sqrt{n}/\varepsilon^2 + 1/\varepsilon^5\big)$ [3] and $\tilde{O}\big(\sqrt{n}/\varepsilon^{7/2}\big)$ [51].

2.2.1.2 Our Techniques and Comparison to Previous Work

The common property of these distribution families $\mathcal{P}$ that allows for our unified testing approach is the following: Let $\mathbf{p}$ be the probability mass function of any distribution in $\mathcal{P}$. Then the Fourier transform of $\mathbf{p}$ is approximately sparse, in a well-defined sense.

For concreteness and due to space limitations, we elaborate for the case of SIIRVs. The starting point of our approach is the observation from [85] that (n,k)-SIIRVs, in addition to having a relatively small effective support, also enjoy an approximately sparse Fourier representation. Roughly speaking, most of their Fourier mass is concentrated on a small subset of Fourier coefficients, which can be computed efficiently.

This suggests the following natural approach to testing (n,k)-SIIRVs: first, identify the effective support I of the distribution $\mathbf{p}$ and check that it is as small as it ought to be. Then, compute the corresponding small subset S of the Fourier domain, and check that almost no Fourier mass of $\mathbf{p}$ lies outside S (otherwise, one can safely reject, as this is a certificate that $\mathbf{p}$ is not an (n,k)-SIIRV). Combining the two, one can show that learning (in L_2 norm) the Fourier transform of $\mathbf{p}$ on this small subset S only, is sufficient to learn $\mathbf{p}$ itself in total variation distance. The former goal can be performed with relatively few samples, as S is sufficiently small.

Doing so results in a distribution $\mathbf{h}$, represented succinctly by its Fourier transform on S, such that $\mathbf{p}$ and $\mathbf{h}$ are close in total variation distance. It only remains to perform a computational "projection step" to verify that $\mathbf{h}$ itself is close to some (n,k)-SIIRV. This will clearly be the case if indeed $\mathbf{p} \in \mathcal{SIIRV}_{n,k}$.

We note that although the above idea is at the core of the SIIRV testing algorithm of Algorithm 12, the actual tester has to address separately the case where $\mathbf{p}$ has small variance, which can be handled by a brute-force learning-and-testing approach. Our main contribution is thus to describe how to efficiently perform the second step, i.e., the Fourier sparsity testing. This is done in Theorem 2.2.4, which describes a simple

[5] As mentioned in Chapter 1, we use the notation $\Omega_k(\cdot)$, $O_k(\cdot)$ to indicate that the parameter k is seen as a constant, focusing on the asymptotics with regard to n, ε.

algorithm to perform this step: essentially, by considering the Fourier coefficients of the empirical distribution obtained by taking a small number of samples. Interestingly, the main idea underlying Theorem 2.2.4 is to avoid analyzing directly the behavior of these Fourier coefficients – which would naively require too high a time complexity. Instead, we rely on Plancherel's identity and reduce the problem to the analysis of a different task: that of the sample complexity of L_2 identity testing (Proposition 2.2.5). By a tight analysis of this L_2 tester, we get as a byproduct that several Fourier quantities of interest (of our empirical distribution) simultaneously enjoy good concentration – while arguing concentration of each of these terms separately would yield a suboptimal time complexity.

A nearly identical method works for PMDs as well. Moreover, our approach can be abstracted to yield a general testing framework, as we explain in Section 2.2.5. It is interesting to remark that the Fourier transform has been used to learn PMDs and SIIRVs [85, 67, 86, 72], and therefore it may not be entirely surprising that it has applications to testing as well. However, testing membership in a class using the Fourier transform is significantly more challenging than learning: a fundamental reason being that, in contrast to the learning setting, we need to handle distributions that are *not* SIIRVs and PMDs (but, indeed, are far from those). The learning algorithms, on the other hand, work under the promise that the distribution is in the class, and thus can leverage the specific structure of SIIRVs and PMDs. Moreover, our Fourier testing techniques gives improved algorithms for other structured families as well, e.g., log-concavity, for which no Fourier learning algorithm was known.

Learning and testing the Fourier transform: the advantage One may wonder how the detour via the Fourier transform enables us to obtain better sample complexity than an approach purely based on L_2 testing. Indeed, all distributions in the classes we consider, crucially, have a small L_2 norm: for testing identity to such a distribution $\mathbf{p}$, the standard L_2 identity tester (see, e.g., [58] or Proposition 2.2.5), which works by checking how large the L_2 distance between the empirical and the hypothesis distribution is, will be optimal. We can thus test membership of a class of such distributions by (i) learning $\mathbf{p}$ assuming it belongs to the class, and then (ii) test whether what we learned is indeed close to $\mathbf{p}$ using the L_2 identity tester. The catch is to get guarantees in L_1 distance out of this, applying Cauchy–Schwarz would require us to learn to very small L_2 distance. Namely, if $\mathbf{p}$ has support size n, we would have to learn to L_2 distance $\frac{\varepsilon}{\sqrt{n}}$ in (i), and then in (ii) test that we are within L_2 distance $\frac{\varepsilon}{\sqrt{n}}$ of the learned hypothesis.

However, if a distribution $\mathbf{p}$ has a sparse discrete Fourier transform whose effective support is known, then it is enough to estimate only these few Fourier coefficients [85, 87]. This enables us to learn $\mathbf{p}$ in (i) not just to within L_1 distance ε but indeed crucially within L_2 distance $\frac{\varepsilon}{\sqrt{n}}$ with good sample complexity. Additionally, the identity tester algorithm can be put into a simpler form for a hypothesis with sparse Fourier transform, as previously mentioned. Now, the tester has a higher sample complexity, roughly $\sqrt{n}/\varepsilon^2$; but if it passes, then we have learned the distribution $\mathbf{p}$ to within ε total variation distance, with much fewer samples than the $\Omega\big(n/\varepsilon^2\big)$ required for arbitrary distributions over support size n.

Lastly, we note that instead of $\sqrt{n}/\varepsilon^2$ in the sample complexity above, we can get $n^{1/4}/\varepsilon^2$ for (n, k)-

SIIRVs by considering the effective support of the distribution.

2.2.2 Testing Effective Fourier Support

In this section, we prove the following theorem, which will be invoked as a crucial ingredient of our testing algorithms. Broadly speaking, the theorem ensures one can efficiently test whether an unknown distribution $\mathbf{q}$ has its Fourier transform concentrated on some (small) effective support S (and if this is the case, learn the vector $\widehat{\mathbf{q}}\mathbb{1}_S$, the restriction of this Fourier transform to S, in L_2 distance).

Theorem 2.2.4. *Given parameters $M \geq 1$, $\varepsilon, b \in (0,1]$, as well as a subset $S \subseteq [\![M]\!]$ and sample access to a distribution $\mathbf{q}$ over $[\![M]\!]$, Algorithm 9 outputs either* reject *or a collection of Fourier coefficients $\widehat{\mathbf{h}'} = (\widehat{\mathbf{h}'}(\xi))_{\xi \in S}$ such that with probability at least $7/10$, all the following statements hold simultaneously.*

1. *if $\|\mathbf{q}\|_2^2 > 2b$, then it outputs* reject*;*
2. *if $\|\mathbf{q}\|_2^2 \leq 2b$ and every function $\mathbf{q}^* \colon [\![M]\!] \to \mathbb{R}$ with $\widehat{\mathbf{q}^*}$ supported entirely on S is such that $\|\mathbf{q} - \mathbf{q}^*\|_2 > \varepsilon$, then it outputs* reject*;*
3. *if $\|\mathbf{q}\|_2^2 \leq b$ and there exists a function $\mathbf{q}^* \colon [\![M]\!] \to \mathbb{R}$ with $\widehat{\mathbf{q}^*}$ supported entirely on S such that $\|\mathbf{q} - \mathbf{q}^*\|_2 \leq \frac{\varepsilon}{2}$, then it does not output* reject*;*
4. *if it does not output* reject*, then $\|\widehat{\mathbf{q}}\mathbb{1}_S - \widehat{\mathbf{h}'}\|_2 \leq \frac{\varepsilon\sqrt{M}}{10}$ and the inverse Fourier transform (modulo M) $\mathbf{h}'$ of the Fourier coefficients $\widehat{\mathbf{h}'}$ it outputs satisfies $\|\mathbf{q} - \mathbf{h}'\|_2 \leq \frac{6\varepsilon}{5}$.*

Moreover, the algorithm takes $m = O\left(\frac{\sqrt{b}}{\varepsilon^2} + \frac{|S|}{M\varepsilon^2} + \sqrt{M}\right)$ samples from $\mathbf{q}$, and runs in time $O(m\,|S|)$.

Note that the rejection condition in Item 2 is equivalent to $\|\widehat{\mathbf{q}}\mathbb{1}_{\bar{S}}\|_2 > \varepsilon\sqrt{M}$, that is to having Fourier mass more than ε^2 outside of S; this is because for any $\mathbf{q}^*$ supported on S,

$$M\|\mathbf{q} - \mathbf{q}^*\|_2^2 = \|\widehat{\mathbf{q}} - \widehat{\mathbf{q}^*}\|_2^2 = \|\widehat{\mathbf{q}}\mathbb{1}_S - \widehat{\mathbf{q}^*}\mathbb{1}_S\|_2^2 + \|\widehat{\mathbf{q}}\mathbb{1}_{\bar{S}} - \widehat{\mathbf{q}^*}\mathbb{1}_{\bar{S}}\|_2^2 \geq \|\widehat{\mathbf{q}}\mathbb{1}_{\bar{S}} - \widehat{\mathbf{q}^*}\mathbb{1}_{\bar{S}}\|_2^2 = \|\widehat{\mathbf{q}}\mathbb{1}_{\bar{S}}\|_2^2$$

and the inequality is tight for $\mathbf{q}^*$ being the inverse Fourier transform (modulo M) of $\widehat{\mathbf{q}}\mathbb{1}_S$.

High-level idea. Let $\mathbf{q}$ be an unknown distribution supported on M consecutive integers (we will later apply this to $\mathbf{q} \stackrel{\text{def}}{=} \mathbf{p} \bmod M$), and $S \subseteq [\![M]\!]$ be a set of Fourier coefficients (symmetric with regard to M: $\xi \in S$ implies $-\xi \bmod M \in S$) such that $0 \in S$. We can further assume that we know $b \geq 0$ such that $\|\mathbf{q}\|_2^2 \leq b$.

Given $\mathbf{q}$, we can consider its "truncated Fourier expansion" (with respect to S) $\widehat{\mathbf{h}} = \widehat{\mathbf{q}}\mathbb{1}_S$ defined as

$$\widehat{\mathbf{h}}(\xi) \stackrel{\text{def}}{=} \begin{cases} \widehat{\mathbf{q}}(\xi) & \text{if } \xi \in S \\ 0 & \text{otherwise} \end{cases}$$

for $\xi \in [\![M]\!]$; and let $\mathbf{h}$ be the inverse Fourier transform (modulo M) of $\widehat{\mathbf{h}}$. Note that $\mathbf{h}$ is no longer in general a probability distribution.

To obtain the guarantees of Theorem 2.2.4, a natural idea is to take some number m of samples from $\mathbf{q}$, and consider the empirical distribution $\mathbf{q}'$ they induce over $[\![M]\!]$. By computing the Fourier coefficients (restricted to S) of this $\mathbf{q}'$, as well as the Fourier mass "missed" when doing so (i.e., the Fourier mass $\|\widehat{\mathbf{q}'}\mathbb{1}_{\bar{S}}\|_2^2$ that $\mathbf{q}'$ puts outside of S) to sufficient accuracy, one may hope to prove Theorem 2.2.4 with a reasonable bound on m.

The issue is that analyzing *separately* the behavior of $\|\widehat{\mathbf{q}'}\mathbb{1}_{\bar{S}}\|_2^2$ and $\|\widehat{\mathbf{q}'}\mathbb{1}_S - \widehat{\mathbf{q}}\mathbb{1}_S\|_2^2$ to show that they are both estimated sufficiently accurately, and both small enough, is not immediate. Instead, we will get a bound on both at the same time, by arguing concentration in a different manner – namely, by analyzing a different tester for tolerant identity testing in L_2 norm.

In more detail, letting $\mathbf{h}$ be as above, we have by Plancherel that

$$\sum_{i\in[\![M]\!]} (\mathbf{q}'(i) - \mathbf{h}(i))^2 = \|\mathbf{q}' - \mathbf{h}\|_2^2 = \frac{1}{M}\|\widehat{\mathbf{q}'} - \widehat{\mathbf{h}}\|_2^2 = \frac{1}{M}\sum_{\xi=0}^{M-1} |\widehat{\mathbf{q}'}(\xi) - \widehat{\mathbf{h}}(\xi)|^2$$

and, expanding the definition of $\widehat{\mathbf{h}}$ and using Plancherel again, this can be rewritten as

$$M\sum_{i\in[\![M]\!]} (\mathbf{q}'(i) - \mathbf{h}(i))^2 = \|\widehat{\mathbf{q}}\mathbb{1}_S - \widehat{\mathbf{q}'}\mathbb{1}_S\|_2^2 + \|\mathbf{q}'\|_2^2 - \|\widehat{\mathbf{q}'}\mathbb{1}_S\|_2^2.$$

(The full derivation will be given in the proof.) The left-hand side has two non-negative compound terms: the first, $\|\widehat{\mathbf{p}}\mathbb{1}_S - \widehat{\mathbf{q}'}\mathbb{1}_S\|_2^2$, corresponds to the L_2 error obtained when learning the Fourier coefficients of $\mathbf{q}$ on S. The second, $\|\mathbf{q}'\|_2^2 - \|\widehat{\mathbf{q}'}\mathbb{1}_S\|_2^2 = \|\widehat{\mathbf{q}'}\mathbb{1}_{\bar{S}}\|_2^2$, is the Fourier mass that our empirical $\mathbf{q}'$ puts "outside of S."

So if the LHS is small (say, order ε^2), then in particular both terms of the RHS will be small as well, effectively giving us bounds on our two quantities in one shot. But this very same LHS is very reminiscent of a known statistic [58] for testing identity of distributions in L_2. So, one can analyze the number of samples required by analyzing such an L_2 tester instead. This is what we will do in Proposition 2.2.5.

Algorithm 9 Testing the Fourier Transform Effective Support

Require: parameters $M \geq 1$, $b, \varepsilon \in (0,1]$; set $S \subseteq [\![M]\!]$; sample access to distribution $\mathbf{q}$ over $[\![M]\!]$

Set $m \leftarrow \left\lceil C(\frac{\sqrt{b}}{\varepsilon^2} + \frac{|S|}{M\varepsilon^2} + \sqrt{M})\right\rceil$ ▷ $C > 0$ is an absolute constant

Draw $m' \leftarrow \mathrm{Poisson}(m)$; if $m' > 2m$, **return** reject

Draw m' samples from $\mathbf{q}$, and let $\mathbf{q}'$ be the corresponding empirical distribution over $[\![M]\!]$

Compute $\|\mathbf{q}'\|_2^2$, $\widehat{\mathbf{q}'}(\xi)$ for every $\xi \in S$, and $\|\widehat{\mathbf{q}'}\mathbb{1}_S\|_2^2$ ▷ Takes time $O(m\,|S|)$

if $m'^2\|\mathbf{q}'\|_2^2 - m' > \frac{3}{2}bm^2$ **then return** reject

else if $\|\mathbf{q}'\|_2^2 - \frac{1}{M}\|\widehat{\mathbf{q}'}\mathbb{1}_S\|_2^2 \geq 3\varepsilon^2\left(\frac{m'}{m}\right)^2 + \frac{1}{m'}$ **then return** reject

else

  return $\widehat{\mathbf{h}'} = (\widehat{\mathbf{q}'}(\xi))_{\xi\in S}$

end if

Proof of Theorem 2.2.4. Given $m' \sim \mathrm{Poisson}(m)$ samples from $\mathbf{q}$, let $\mathbf{q}'$ be the empirical distribution they define. We first observe that with probability $2^{-\Omega(\varepsilon^2 m/b)} < \frac{1}{100}$, we have $m' \in [1 \pm \frac{\varepsilon}{100\sqrt{b}}]m$ and thus

the algorithm does not output reject in Step 1 (this follows from standard concentration bounds on Poisson random variables). We will afterwards assume this holds. By Plancherel, we have

$$\sum_{i\in[\![M]\!]}(\mathbf{q}'(i)-\mathbf{h}(i))^2=\|\mathbf{q}'-\mathbf{h}\|_2^2=\frac{1}{M}\|\widehat{\mathbf{q}'}-\widehat{\mathbf{h}}\|_2^2=\frac{1}{M}\sum_{\xi=0}^{M-1}|\widehat{\mathbf{q}'}(\xi)-\widehat{\mathbf{h}}(\xi)|^2$$

and, expanding the definition of $\widehat{\mathbf{h}}$, this yields

$$\begin{aligned}\sum_{i\in[\![M]\!]}(\mathbf{q}'(i)-\mathbf{h}(i))^2&=\frac{1}{M}\sum_{\xi\in S}|\widehat{\mathbf{q}'}(\xi)-\widehat{\mathbf{h}}(\xi)|^2+\frac{1}{M}\sum_{\xi\notin S}|\widehat{\mathbf{q}'}(\xi)|^2\\&=\frac{1}{M}\sum_{\xi\in S}|\widehat{\mathbf{q}'}(\xi)-\widehat{\mathbf{q}}(\xi)|^2+\frac{1}{M}\sum_{\xi=0}^{M-1}|\widehat{\mathbf{q}'}(\xi)|^2-\frac{1}{M}\sum_{\xi\in S}|\widehat{\mathbf{q}'}(\xi)|^2\\&=\frac{1}{M}\left(\|\widehat{\mathbf{q}}\mathbb{1}_S-\widehat{\mathbf{q}'}\mathbb{1}_S\|_2^2+\|\widehat{\mathbf{q}'}\|_2^2-\|\widehat{\mathbf{q}'}\mathbb{1}_S\|_2^2\right)\\&=\frac{1}{M}\|\widehat{\mathbf{q}}\mathbb{1}_S-\widehat{\mathbf{q}'}\mathbb{1}_S\|_2^2+\|\mathbf{q}'\|_2^2-\frac{1}{M}\|\widehat{\mathbf{q}'}\mathbb{1}_S\|_2^2\end{aligned}\tag{2.28}$$

where in the last step we invoked Plancherel again to argue that $\frac{1}{M}\|\widehat{\mathbf{q}'}\|_2^2=\|\mathbf{q}'\|_2^2$.

To analyze the correctness of the algorithm (specifically, the completeness), we will adopt the point of view suggested by (2.28) and analyze instead the statistic $\sum_{i\in[\![M]\!]}(\mathbf{q}'(i)-\mathbf{h}(i))^2$, when $\mathbf{h}$ is an explicit (pseudo) distribution on $[\![M]\!]$ assumed known, and $\mathbf{q}'$ is the empirical distribution obtained by drawing $\mathrm{Poisson}(m)$ samples from some unknown distribution $\mathbf{q}$. (Namely, we want to see this as a tolerant L_2 identity tester between $\mathbf{q}$ and $\mathbf{h}$.)

- We first show that, given that $m'=\Omega\left(\frac{|S|}{M\varepsilon^2}\right)$, with probability at least $\frac{99}{100}$ we have $\|\widehat{\mathbf{q}}\mathbb{1}_S-\widehat{\mathbf{h}'}\|_2\le\frac{\sqrt{M}\varepsilon}{10}$. We note that $m'\widehat{\mathbf{q}'}(\xi)$ is an sum of m' i.i.d. numbers each of absolute value 1 and mean $\widehat{\mathbf{q}}(\xi)$ (which has absolute value less than 1). If X is one of these numbers, $|X-\widehat{\mathbf{q}}(\xi)|\le 2$ with probability 1 and so the variance of the real and imaginary parts of X is at most 4. Thus the variance of the real and imaginary parts of $m'\widehat{\mathbf{q}'}(\xi)$ is at most $4m'$. Then we have $\mathbb{E}[|\widehat{\mathbf{q}}(\xi)-\widehat{\mathbf{q}'}(\xi)|^2]=\mathbb{E}[(\Re(\widehat{\mathbf{q}}(\xi)-\widehat{\mathbf{q}'}(\xi)))^2+(\Im(\widehat{\mathbf{q}}(\xi)-\widehat{\mathbf{q}'}(\xi)))^2]\le 8/m'$. Summing over S, using that $\mathbf{q}'$ and $\mathbf{h}'$ have the same Fourier coefficients there, yields

$$\mathbb{E}\left[\sum_{\xi\in S}\left|\widehat{\mathbf{q}}(\xi)-\widehat{\mathbf{h}'}(\xi)\right|^2\right]\le\frac{8|S|}{m'}\le\frac{M\varepsilon^2}{10000}$$

and by Markov's inequality we get $\Pr\left[\,\|\widehat{\mathbf{q}}\mathbb{1}_S-\widehat{\mathbf{h}'}\|_2^2\le\frac{M\varepsilon^2}{100}\,\right]=\Pr\left[\sum_{\xi\in S}\left|\widehat{\mathbf{q}}(\xi)-\widehat{\mathbf{h}'}(\xi)\right|^2\le\frac{M\varepsilon^2}{100}\right]\ge\frac{1}{100}$, concluding the proof.
- Then, let us consider Item 1: assume $\|\mathbf{q}\|_2^2>2b$, and set $X\stackrel{\text{def}}{=}m'^2\|\mathbf{q}'\|_2^2-m'$. Then,

$$\mathbb{E}[X]=\sum_{i=1}^M\mathbb{E}[m'^2\mathbf{q}'(i)^2]-\sum_{i=1}^M\mathbb{E}[m'\mathbf{q}'(i)]=\sum_{i=1}^M(m\mathbf{q}(i)+m^2\mathbf{q}(i)^2)-\sum_{i=1}^M m\mathbf{q}(i)=m^2\|\mathbf{q}\|_2^2$$

since the $m'\mathbf{q}'(i)$ are distributed as $\mathrm{Poisson}(m\mathbf{q}(i))$. As all $m'\mathbf{q}'(i)$'s are independent by Poissonization, we also have

$$\mathrm{Var}[X] = \sum_{i=1}^{M} \mathrm{Var}[m'^2\mathbf{q}'(i)^2 - m'\mathbf{q}'(i)] = \sum_{i=1}^{M}(2m^2\mathbf{q}(i)^2 + 4m^3\mathbf{q}(i)^3) = 2m^2\|\mathbf{q}\|_2^2 + 4m^3\|\mathbf{q}\|_3^3$$

and by Chebyshev,

$$\Pr[X \le \frac{3}{2}m^2 b] \le \Pr\left[|X - \mathbb{E}[X]| > \frac{1}{4}\mathbb{E}[X]\right] \le 16\frac{\mathrm{Var}[X]}{\mathbb{E}[X]^2} \le \frac{32}{m^2\|\mathbf{q}\|_2^2} + \frac{64\|\mathbf{q}\|_3^3}{m\|\mathbf{q}\|_2^4}$$

Since $\mathbf{q}$ is supported on $[\![M]\!]$, $\|\mathbf{q}\|_2^2 \ge \frac{1}{M}$ and the first term is at most $\frac{32M}{m^2}$. The second term, by monotonicity of ℓ_p-norms, is at most $\frac{64\|\mathbf{q}\|_2^3}{m\|\mathbf{q}\|_2^4} = \frac{48}{m\|\mathbf{q}\|_2} \le \frac{48\sqrt{M}}{m}$. The RHS is then at most $\frac{1}{100}$ for a large enough choice of $C > 0$ in the definition of m. Thus, with probability at least $1 - \frac{1}{100}$ we have $m'^2\|\mathbf{q}'\|_2^2 - m' > \frac{3}{2}b$, and the algorithm outputs reject in Step 5.

Moreover, if $\|\mathbf{q}\|_2^2 \le b$, then the same analysis shows that

$$\Pr[X > \frac{3}{2}m^2 b] \le \Pr\left[|X - \mathbb{E}[X]| > \frac{1}{2}\mathbb{E}[X]\right] \le 4\frac{\mathrm{Var}[X]}{\mathbb{E}[X]^2} \le \frac{1}{100}$$

and with probability at least $1 - \frac{1}{100}$ the algorithm does not output reject in Step 4.

- Turning now to Items 2 to 4: we assume that the algorithm does not output reject in Step 4 (which by the above happens with probability $99/100$ if $\|\mathbf{q}\|_2^2 \le b$; and can be assumed without loss of generality otherwise, since we then want to argue that the algorithm *does* reject at some point in that case).

  By the remark following the statement of the theorem, it is sufficient to show that the algorithm outputs reject (with high probability) if $\|\widehat{\mathbf{q}}\mathbb{1}_{\bar{S}}\|_2^2 > \varepsilon^2 M$, and that if both $\|\mathbf{q}\|_2^2 \le b$ and $\|\widehat{\mathbf{q}}\mathbb{1}_{\bar{S}}\|_2^2 \le \frac{\varepsilon^2}{4}M$ then it does not output reject; and that whenever the algorithm does not output reject, then $\|\widehat{\mathbf{q}} - \widehat{\mathbf{h}}\|_2 \le \varepsilon^2 M$.

  Observe that calling Algorithm 10 with our $m' = \mathrm{Poisson}(m)$ samples from $\mathbf{q}$ (distribution over $[\![M]\!]$), parameters $\frac{\varepsilon}{2}$ and $2b$, and the explicit description of the pseudo distribution $\mathbf{p}^* \stackrel{\mathrm{def}}{=} \frac{m'}{m}\mathbf{h}$ (which one would obtain for $\mathbf{h}$ being the inverse Fourier transform of $\widehat{\mathbf{q}}\mathbb{1}_S$) would result by Proposition 2.2.5 (since $m \ge c\frac{\sqrt{2b}}{(\varepsilon/2)^2} = 244\sqrt{2}\frac{\sqrt{b}}{\varepsilon^2}$, where c is as in Proposition 2.2.5) in having the following guarantees on $\frac{\sqrt{Z}}{m}$, where Z is the statistic defined in Algorithm 10

  - if $\|\mathbf{q} - \mathbf{p}^*\|_2 \le \frac{\varepsilon}{2}$, then $\frac{\sqrt{Z}}{m} \le \sqrt{2.9}\varepsilon$ with probability at least $3/4$;
  - if $\|\mathbf{q} - \mathbf{p}^*\|_2 \ge \varepsilon$, then $\frac{\sqrt{Z}}{m} \ge \sqrt{3.1}\varepsilon$ with probability at least $3/4$;

  as $\|\mathbf{q}\|_2^2 \le 2b$ (note that then $\|\mathbf{h}\|_2^2 \le b$ as well). Since $\sqrt{M}\|\mathbf{q} - \mathbf{p}^*\|_2 = \|\widehat{\mathbf{q}} - \widehat{\mathbf{p}^*}\|_2 = \|\widehat{\mathbf{q}} - \frac{m}{m'}\widehat{\mathbf{q}}\mathbb{1}_S\|_2$ and

$$\frac{Z}{m'^2} = \sum_{i=1}^{M}\left((\mathbf{q}'(i) - \frac{m}{m'}\mathbf{p}^*(i))^2 - \frac{\mathbf{q}'(i)}{m'}\right) = \sum_{i=1}^{M}(\mathbf{q}'(i) - \mathbf{h}(i))^2 - \frac{1}{m'}$$

which is equal to $\frac{1}{M}\|\widehat{\mathbf{q}}\mathbb{1}_S - \widehat{\mathbf{q}'}\mathbb{1}_S\|_2^2 + \|\mathbf{q}'\|_2^2 - \frac{1}{M}\|\widehat{\mathbf{q}'}\mathbb{1}_S\|_2^2 - \frac{1}{m'}$ by Eq. (2.28), we thus get the

following.

– if $\|\widehat{\mathbf{q}}\mathbb{1}_{\bar{S}}\|_2^2 \le \frac{\varepsilon^2 M}{9}$, then $\|\widehat{\mathbf{q}} - \widehat{\mathbf{q}}\mathbb{1}_S\|_2 \le \frac{\varepsilon}{3}\sqrt{M}$, and

$$\sqrt{M}\|\mathbf{p}^* - \mathbf{q}\|_2 = \|\widehat{\mathbf{p}^*} - \widehat{\mathbf{q}}\|_2 \le \|\widehat{\mathbf{p}^*} - \widehat{\mathbf{q}}\mathbb{1}_S\|_2 + \|\widehat{\mathbf{q}}\mathbb{1}_S - \widehat{\mathbf{q}}\|_2 = \left|\frac{m}{m'} - 1\right| \|\widehat{\mathbf{q}}\mathbb{1}_S\|_2 + \|\widehat{\mathbf{q}} - \widehat{\mathbf{q}}\mathbb{1}_S\|_2$$

Since we have $m' \in [1 \pm \frac{\varepsilon}{100\sqrt{b}}]m$ by the above discussion and $\|\widehat{\mathbf{q}}\mathbb{1}_S\|_2 \le \sqrt{2b}\sqrt{M}$, the RHS is upper bounded by $\frac{\varepsilon}{6}\sqrt{M} + \frac{\varepsilon}{3}\sqrt{M} = \frac{\varepsilon}{2}\sqrt{M}$, and $\|\mathbf{p}^* - \mathbf{q}\|_2 \le \frac{\varepsilon}{2}$. Then $\frac{1}{M}\|\widehat{\mathbf{q}}\mathbb{1}_S - \widehat{\mathbf{q}'}\mathbb{1}_S\|_2^2 + \|\mathbf{q}'\|_2^2 - \frac{1}{M}\|\widehat{\mathbf{q}'}\mathbb{1}_S\|_2^2 = \frac{Z}{m'^2} + \frac{1}{m'} \le 2.9\varepsilon^2\left(\frac{m'}{m}\right)^2 + \frac{1}{m'}$ with probability at least $3/4$, and in particular $\|\mathbf{q}'\|_2^2 - \frac{1}{M}\|\widehat{\mathbf{q}'}\mathbb{1}_S\|_2^2 \le 2.9\varepsilon^2\left(\frac{m'}{m}\right)^2 + \frac{1}{m'} < 3\varepsilon^2\left(\frac{m'}{m}\right)^2 + \frac{1}{m'}$;

– if $\|\widehat{\mathbf{q}}\mathbb{1}_{\bar{S}}\|_2^2 > \varepsilon^2 M$, then $\frac{1}{M}\|\widehat{\mathbf{q}}\mathbb{1}_S - \widehat{\mathbf{q}'}\mathbb{1}_S\|_2^2 + \|\mathbf{q}'\|_2^2 - \frac{1}{M}\|\widehat{\mathbf{q}'}\mathbb{1}_S\|_2^2 = \frac{Z}{m'^2} + \frac{1}{m'} > 3.1\varepsilon^2\left(\frac{m'}{m}\right)^2 + \frac{1}{m'}$ with probability at least $3/4$; since by the first part we established we have $\|\widehat{\mathbf{q}}\mathbb{1}_S - \widehat{\mathbf{q}'}\mathbb{1}_S\|_2^2 \le \frac{\varepsilon^2 M}{100}$, this implies $\|\mathbf{q}'\|_2^2 - \frac{1}{M}\|\widehat{\mathbf{q}'}\mathbb{1}_S\|_2^2 > 3.1\varepsilon^2\left(\frac{m'}{m}\right)^2 + \frac{1}{m'} - \frac{\varepsilon^2}{100} > 3\varepsilon^2\left(\frac{m'}{m}\right)^2 + \frac{1}{m'}$.

This immediately takes care of Items 2 and 3; moreover, this implies that whenever Algorithm 9 does *not* output reject, then the inverse Fourier transform $\mathbf{h}'$ of the collection of Fourier coefficients it returns (which are supported on S) satisfies

$$\begin{aligned}\|\mathbf{q} - \mathbf{h}'\|_2^2 = \frac{1}{M}\|\widehat{\mathbf{q}} - \widehat{\mathbf{h}'}\|_2^2 &= \frac{1}{M}\|\widehat{\mathbf{q}}\mathbb{1}_S - \widehat{\mathbf{h}'}\|_2^2 + \frac{1}{M}\|\widehat{\mathbf{q}}\mathbb{1}_{\bar{S}}\|_2^2 \\ &\le \frac{\varepsilon^2}{100} + \frac{1}{M}\|\widehat{\mathbf{q}}\mathbb{1}_{\bar{S}}\|_2^2 \\ &\le \frac{\varepsilon^2}{100} + \varepsilon^2 = \frac{101}{100}\varepsilon^2\end{aligned}$$

and thus $\|\mathbf{q} - \mathbf{h}'\|_2 \le \sqrt{\frac{101}{100}}\varepsilon < \frac{6}{5}\varepsilon$ which establishes Item 4. Finally, by a union bound, all the above holds except with probability $\frac{1}{100} + \frac{1}{100} + \frac{1}{100} + \frac{1}{4} < \frac{3}{10}$. This concludes the proof.

□

2.2.2.1 A tolerant L_2 tester for identity to a pseudodistribution

As previously mentioned, one building block in the proof of Theorem 2.2.4 (and a result that may be of independent interest) is an optimal L_2 identity testing algorithm. Our tester and its analysis are very similar to the tolerant L_2 closeness testing algorithm of Chan et al. [58], with the obvious simplifications pertaining to identity (instead of closeness). The main difference is that we emphasize here the fact that $\mathbf{p}^*$ need not be an actual distribution: any $\mathbf{p}^* : [r] \to \mathbb{R}$ would do, even taking negative values. This will turn out to be crucial for our applications.

Proposition 2.2.5. *There exists an absolute constant $c > 0$ such that the above algorithm (Algorithm 10), when given $\mathrm{Poisson}(m)$ samples drawn from a distribution $\mathbf{p}$ and an explicit function $\mathbf{p}^* : [r] \to \mathbb{R}$ will, with probability at least $3/4$, distinguishes between (a) $\|\mathbf{p} - \mathbf{p}^*\|_2 \le \varepsilon$ and (b) $\|\mathbf{p} - \mathbf{p}^*\|_2 \ge 2\varepsilon$ provided that*

Algorithm 10 Tolerant L_2 identity tester

Require: $\varepsilon \in (0,1)$, m samples from distributions $\mathbf{p}$ over $[r]$, with X_i denoting the number of occurrences of the i-th domain elements in the samples from $\mathbf{p}$, and $\mathbf{p}^*$ being a fixed, known pseudo distribution over $[r]$.
Ensure: Returns accept if $\|\mathbf{p}-\mathbf{p}^*\|_2 \le \varepsilon$ and reject if $\|\mathbf{p}-\mathbf{p}^*\|_2 \ge 2\varepsilon$.
Define $Z = \sum_{i=1}^{r} (X_i - m\mathbf{p}^*(i))^2 - X_i$. ▷ Can actually be computed in $O(m)$ time
Return reject if $\frac{\sqrt{Z}}{m} > \sqrt{3}\varepsilon$, accept otherwise.

$m \ge c\frac{\sqrt{b}}{\varepsilon^2}$, *where b is an upper bound on $\|\mathbf{p}\|_2^2, \|\mathbf{p}^*\|_2^2$. (Moreover, one can take $c = 61$.)*

Moreover, we have the following stronger statement: in case (a), the statistic Z computed in the algorithm satisfies $\frac{\sqrt{Z}}{m} \le \sqrt{2.9}\varepsilon$ with probability at least $3/4$, while in case (b) we have $\frac{\sqrt{Z}}{m} \ge \sqrt{3.1}\varepsilon$ with probability at least $3/4$.

Proof. Letting X_i denote the number of occurrences of the i-th domain element in the samples from $\mathbf{p}$, define $Z_i = (X_i - m\mathbf{p}^*(i))^2 - X_i$. Since X_i is distributed as $\mathrm{Poisson}(m\mathbf{p}(i))$, $\mathbb{E}[Z_i] = m^2(\mathbf{p}(i)-\mathbf{p}^*(i))^2$; thus, Z is an unbiased estimator for $m^2\|\mathbf{p}-\mathbf{p}^*\|_2^2$. (Note that this holds even when $\mathbf{p}^*$ is allowed to take negative values.)

We compute the variance of Z_i via a straightforward calculation involving standard expressions for the moments of a Poisson distribution: getting

$$\mathrm{Var}[Z] = \sum_{i=1}^{r} \mathrm{Var}[Z_i] = \sum_{i=1}^{r} \left(4m^3(\mathbf{p}(i)-\mathbf{p}^*(i))^2\mathbf{p}(i) + 2m^2\mathbf{p}(i)^2\right).$$

By Cauchy–Schwarz, and since $\sum_{i=1}^{r} \mathbf{p}(i)^2 \le b$ by assumption, we have

$$\begin{aligned}\sum_{i=1}^{r}(\mathbf{p}(i)-\mathbf{p}^*(i))^2\mathbf{p}(i) &= \sum_{i=1}^{r}(\mathbf{p}(i)-\mathbf{p}^*(i))\cdot(\mathbf{p}(i)-\mathbf{p}^*(i))\mathbf{p}(i) \\ &\le \sqrt{\sum_{i=1}^{r}(\mathbf{p}(i)-\mathbf{p}^*(i))^2 \sum_{i=1}^{r}\mathbf{p}(i)^2(\mathbf{p}(i)-\mathbf{p}^*(i))^2} \\ &\le \sqrt{\sum_{i=1}^{r}(\mathbf{p}(i)-\mathbf{p}^*(i))^2 b \sum_{i=1}^{r}(\mathbf{p}(i)-\mathbf{p}^*(i))^2} = \sqrt{b}\|\mathbf{p}-\mathbf{p}^*\|_2^2\end{aligned}$$

and so

$$\mathrm{Var}[Z] \le 4m^3\sqrt{b}\|\mathbf{p}-\mathbf{p}^*\|_2^2 + 2m^2 b.$$

For convenience, let $\eta \stackrel{\mathrm{def}}{=} \frac{1}{10}$, and write $\rho \stackrel{\mathrm{def}}{=} \frac{\|\mathbf{p}-\mathbf{p}^*\|_2}{\varepsilon}$ – so that we need to distinguish $\rho \le 1$ from $\rho \ge 2$. If $\rho \le 1$, i.e. $\mathbb{E}[Z] \le m^2\varepsilon^2$, then

$$\Pr[Z > (3-\eta)m^2\varepsilon^2] = \Pr[|Z - \mathbb{E}[Z]| > m^2\varepsilon^2(((3-\eta)-\gamma)-\rho^2)]$$

while if $\rho \geq 2$, i.e. $\mathbb{E}[Z] \geq 4m^2\varepsilon^2$, then

$$\Pr[Z < (3+\eta)m^2\varepsilon^2] = \Pr[\mathbb{E}[Z] - Z > m^2(\|p-q\|_2^2 - (3+\eta)\varepsilon^2)] \leq \Pr[|Z - \mathbb{E}[Z]| > m^2\varepsilon^2(\rho^2 - (3+\eta))].$$

In both cases, by Chebyshev's inequality, the test will be correct with probability at least $3/4$ provided $m \geq c\sqrt{b}/\varepsilon^2$ for some suitable choice of $c > 0$, since (where

$$\begin{aligned}\Pr[|Z - \mathbb{E}[Z]| > m^2\varepsilon^2|\rho^2 - (3 \pm \eta)|] &\leq \frac{\mathrm{Var}[Z]}{m^4\varepsilon^4(\rho^2 - (3 \pm \eta))^2} \\ &\leq \frac{4m^3\sqrt{b}\rho^2\varepsilon^2 + 2m^2 b}{m^4\varepsilon^4(\rho^2 - (3 \pm \eta))^2} = \frac{\rho^2}{(\rho^2 - (3 \pm \eta))^2} \cdot \frac{4\sqrt{b}}{m\varepsilon^2} + \frac{1}{(\rho^2 - (3 \pm \eta))^2} \cdot \frac{2b}{m^2\varepsilon^4} \\ &\leq \frac{20\sqrt{b}}{m\varepsilon^2} + \frac{5b}{2m^2\varepsilon^4} \leq \frac{20}{c} + \frac{5}{2c^2} \leq \frac{1}{3}\end{aligned}$$

as $\max_{\rho \in [0,1]} \frac{\rho^2}{(\rho^2-(3\pm\eta))^2} \leq 5$ and $\max_{\rho \in [0,1]} \frac{1}{(\rho^2-(3\pm\eta))^2} \leq \frac{5}{4}$ and the last inequality holds for $c \geq 61$. □

2.2.3 The Projection Subroutine

2.2.3.1 The projection step for (n,k)-SIIRVs

We can use the proper ε-cover given in [85] to find a (n,k)-SIIRV near $\mathbf{p}$ by looking at $\widehat{\mathbf{h}}$.

Algorithm 11 Algorithm `Project-k-SIIRV`

Require: Parameters n,ε; the approximate Fourier coefficients $(\widehat{\mathbf{h}}(\xi))_{\xi \in S}$ modulo M, of a distribution $\mathbf{p}$ known to be effectively supported on I and to have a Fourier transform effectively supported on S of the form given in Step 13 of Algorithm 12, with $\widetilde{\sigma}^2$ and $\widetilde{\mu}$, an approximation to $\mathbb{E}_{X \sim \mathbf{p}}[X]$ to within half a standard deviation.

1: Compute $\mathcal{C}$, an $\frac{\varepsilon}{5\sqrt{|S|}}$-cover in total variation distance of all (n,k)-SIIRVs.
2: **for** each $\mathbf{q} \in \mathcal{C}$ **do**
3: **if** the mean $\mu_{\mathbf{q}}$ and variance $\sigma_{\mathbf{q}}$ of $\mathbf{q}$ satisfy $|\widetilde{\mu} - \mu_{\mathbf{q}}| \leq \widetilde{\sigma}$ and $2(\sigma_{\mathbf{q}} + 1) \geq \widetilde{\sigma} + 1 \geq (\sigma_{\mathbf{q}} + 1)/2$ **then**
4: Compute $\widehat{\mathbf{q}}(\xi)$ for $\xi \in S$.
5: **if** $\sum_{\xi \in S} |\widehat{\mathbf{h}} - \widehat{\mathbf{q}}|^2 \leq \frac{\varepsilon^2}{5}$ **then return** accept
6: **end if**
7: **end if**
8: **end for**
9: **return** reject ▷ we did not return accept for any $\mathbf{q} \in \mathcal{C}$

Lemma 2.2.6. *If Algorithm* `Project-k-SIIRV` *is given inputs that satisfy its assumptions and we have that* $\sum_{\xi \in S} |\widehat{\mathbf{h}} - \widehat{\mathbf{p}}|^2 \leq (3\varepsilon/25)^2$, $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{h}) \leq 6\varepsilon/25$, *and that if* $\mathbf{p} \in \mathcal{SIIRV}_{n,k}$ *then* $\widetilde{\sigma}^2$ *is a factor-*1.5 *approximation to* $\mathrm{Var}_{X \sim \mathbf{p}}[X] + 1$, *then it distinguishes between (i)* $\mathbf{p} \in \mathcal{SIIRV}_{n,k}$ *and (ii)* $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathcal{SIIRV}_{n,k}) > \varepsilon$. *The algorithm runs in time* $n\,(k/\varepsilon)^{O(k\log(k/\epsilon))}$.

Proof. By Theorem 3.7 of [85], there is an algorithm that can compute an ε-cover of all (n,k)-SIIRVs of size $n\,(k/\varepsilon)^{O(k\log(1/\epsilon))}$ that runs in time $n\,(k/\varepsilon)^{O(k\log(1/\epsilon))}$. Note the way the cover is given, allows us to compute the Fourier coefficients $\widehat{\mathbf{q}}(\xi)$ for any ξ for each $\mathbf{q} \in \mathcal{C}$ in time $\mathrm{poly}(k/\varepsilon)$.

Since $\varepsilon/\sqrt{|S|} = 1/\operatorname{poly}(k/\varepsilon)$, Step 1 takes time $n\,(k/\varepsilon)^{O(k\log(k/\epsilon))}$ and outputs a cover of size $n\,(k/\varepsilon)^{O(k\log(k/\epsilon))}$. As each iteration takes time $|S|$, the whole algorithm takes $n\,(k/\varepsilon)^{O(k\log(k/\epsilon))}$ time.

Note that each $\mathbf{q}$ that passes Step 3 is effectively supported on I by (2.30) and has Fourier transform supported on S by Claim 2.2.14.

- Suppose that $\mathbf{p} \in \mathcal{SIIRV}_{n,k}$. Then there is a (n,k)-SIIRV $\mathbf{q} \in \mathcal{C}$ with $\mathrm{d_{TV}}(\mathbf{p},\mathbf{q}) \le \varepsilon/5\sqrt{|S|}$. We need to show that if the algorithm considers $\mathbf{q}$, it accepts. From standard concentration bounds, one gets that the expectations of $\mathbf{p}$ and $\mathbf{q}$ are within $O(\varepsilon\sqrt{\log(1/\varepsilon)})$ standard deviations of $\mathbf{p}$ and the variances of $\mathbf{p}$ and $\mathbf{q}$ are within $O(\varepsilon\log(1/\varepsilon))$ multiplicative error. Thus $\mathbf{q}$ passes the condition of Step 3. Since $\mathrm{d_{TV}}(\mathbf{p},\mathbf{q}) \le \varepsilon/(5\sqrt{|S|})$, we have that $|\widehat{\mathbf{p}}(\xi) - \widehat{\mathbf{q}}(\xi)| \le \varepsilon/(5\sqrt{|S|})$ for all ξ. In particular, we have $\sum_{\xi\in S}|\widehat{\mathbf{h}} - \widehat{\mathbf{q}}|^2 \le \varepsilon^2/25$. Thus by the triangle inequality for L_2 norm, we have $\sum_{\xi\in S}|\widehat{\mathbf{h}} - \widehat{\mathbf{q}}|^2 \le (\varepsilon/5 + 3\varepsilon/25)^2 \le (\varepsilon/\sqrt{5})^2$. Thus the algorithm accepts.
- Now suppose that the algorithm accepts. We need to show that $\mathbf{p}$ has total variation distance at most ε from some (n,k)-SIIRV. We will show that $\mathrm{d_{TV}}(\mathbf{p},\mathbf{q}) \le \varepsilon$ for the $\mathbf{q}$ which causes the algorithm to accept. Since the algorithm accepts, $\sum_{\xi\in S}|\widehat{\mathbf{h}} - \widehat{\mathbf{q}}|^2 \le \varepsilon^2/25$. For $x \notin S$, $\widehat{\mathbf{h}}(\xi) = 0$ and so $\sum_{\xi\notin S}|\widehat{\mathbf{h}} - \widehat{\mathbf{q}}|^2 = \sum_{\xi\notin S}|\widehat{\mathbf{q}}|^2 \le \varepsilon^2/100$ by Claim 2.2.14. By Plancherel, the distributions $\mathbf{q}' \stackrel{\text{def}}{=} \mathbf{q} \bmod M$, $\mathbf{h}' \stackrel{\text{def}}{=} \mathbf{h} \bmod M$ satisfy

$$\|\mathbf{q}' - \mathbf{h}'\|_2^2 = \frac{1}{M}\sum_{\xi=0}^{M-1}|\widehat{\mathbf{h}} - \widehat{\mathbf{q}}|^2 \le \frac{\varepsilon^2}{20M}.$$

  Thus $\mathrm{d_{TV}}(\mathbf{q}',\mathbf{h}') \le \frac{\varepsilon}{4}$. By definition $\mathbf{h}$ has probability 0 outside I and by (2.30), $\mathbf{q}$ has at most $\frac{\varepsilon}{5}$ probability outside I, Thus $\mathrm{d_{TV}}(\mathbf{q},\mathbf{h}) \le \frac{\varepsilon}{4} + \frac{\varepsilon}{5} \le \frac{\varepsilon}{2}$ and by the triangle inequality $\mathrm{d_{TV}}(\mathbf{p},\mathbf{q}) \le \mathrm{d_{TV}}(\mathbf{q},\mathbf{h}) + \mathrm{d_{TV}}(\mathbf{p},\mathbf{h}) \le \varepsilon/2 + 6\varepsilon/25 \le \varepsilon$ as required.

□

2.2.3.2 The case $k = 2$

For the important case of Poisson Binomial distributions, that is $(n,2)$-SIIRVs, we can dispense with using a cover at all. [86] gives an algorithm that can properly learn Poisson binomial distributions in time $(1/\varepsilon)^{O(\log\log 1/\varepsilon)}$. The algorithm works by first learning the Fourier coefficients in S, which we have already computed here, and checks if one of many systems of polynomial inequalities has a solution: if the Fourier coefficients are close to those of a $(n,2)$-SIIRV, then there will be a solution to one of these systems. This allows us to test whether or not we are close to a $(n,2)$-SIIRV.

More precisely, we can handle this in two cases: the first, when the variance s^2 of $\mathbf{p}$ is relatively small, corresponding to $\widetilde{\sigma} \le \alpha/\varepsilon^2$ (for some absolute constant $\alpha > 0$).

Lemma 2.2.7. *Let $\mathbf{p}$ be a distribution with variance $O(1/\varepsilon^2)$. Let $\widetilde{\mu}$ and $\widetilde{\sigma}^2$ be approximations to the mean μ and variance s^2 of $\mathbf{p}$ with $|\widetilde{\mu} - \mu| \le \widetilde{\sigma}$ and $2(\sigma+1) \ge \widetilde{\sigma} + 1 \ge (\sigma+1)/2$. Suppose that $\mathbf{p}$ is effectively*

supported on an interval I and that its DFT modulo M is effectively supported on S, the set of integers $\xi \leq \ell \stackrel{\text{def}}{=} O(\log(1/\varepsilon))$. Let $\widehat{\mathbf{h}}(\xi)$ be approximations to $\widehat{\mathbf{p}}(\xi)$ for all $\xi \in S$ with $\sum_{\xi \in S} |\widehat{\mathbf{h}}(\xi) - \widehat{\mathbf{p}}(\xi)|^2 \leq \frac{\varepsilon^2}{16}$. There is an algorithm that, given n,ε,$\widetilde{\mu}$, $\widetilde{\sigma}$ and $\widehat{\mathbf{h}}(\xi)$, distinguishes between (i) $\mathbf{p} \in \mathcal{PBD}_n$ and (ii) $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathcal{PBD}_n) > \varepsilon$, in time at most $(1/\varepsilon)^{O(\log\log 1/\varepsilon)}$.

Proof. We use Steps 4 and 5 of Algorithm `Proper-Learn-PBD` in [86]. Step 5 checks if one of a system of polynomials has a solution. If such a solution is found, it corresponds to an $(n, 2)$-SIIRV $\mathbf{q}$ that has $\sum_{|\xi| \leq \ell} |\widehat{\mathbf{h}}(\xi) - \widehat{\mathbf{q}}(\xi)|^2 \leq \varepsilon^2/4$ and so we accept. If no systems have a solution, then there is no such $(n, 2)$-SIIRV and so we reject. The conditions of this lemma are enough to satisfy the conditions of Theorem 11 of [86], though we need that the constant C' used to define $|S|$ is sufficiently large to cover the $\ell = O(\log(1/\varepsilon)$ from that paper. This theorem means that if $\mathbf{p}$ is a $(n, 2)$-SIIRV, then we accept.

We need to show that if the algorithm finds a solution, then it is within ε of a Poisson Binomial distribution. The system of equations ensures that $\sum_{|\xi| \leq \ell} |\widehat{\mathbf{h}}(\xi) - \widehat{\mathbf{q}}(\xi)|^2 \leq \varepsilon^2/4$. Now the argument is similar to that for (n, k)-SIIRVs. For $x \notin S$, $\widehat{\mathbf{h}}(\xi) = 0$ and so $\sum_{\xi \notin S} |\widehat{\mathbf{h}} - \widehat{\mathbf{q}}|^2 = \sum_{\xi \notin S} |\widehat{\mathbf{q}}|^2 \leq \varepsilon^2/100$ by Claim 2.2.14. By Plancherel, the distributions $\mathbf{q}' \stackrel{\text{def}}{=} \mathbf{q} \bmod M$, $\mathbf{h}' \stackrel{\text{def}}{=} \mathbf{h} \bmod M$ satisfy

$$\|\mathbf{q}' - \mathbf{h}'\|_2^2 = \frac{1}{M} \sum_{\xi=0}^{M-1} |\widehat{\mathbf{h}} - \widehat{\mathbf{q}}|^2 \leq \frac{\varepsilon^2}{20M}.$$

Thus $\mathrm{d}_{\mathrm{TV}}(\mathbf{q}', \mathbf{h}') \leq \frac{\varepsilon}{4}$. By definition $\mathbf{h}$ has probability 0 outside I and by (2.30), $\mathbf{q}$ has at most $\frac{\varepsilon}{5}$ probability outside I, Thus $\mathrm{d}_{\mathrm{TV}}(\mathbf{q}, \mathbf{h}) \leq \frac{\varepsilon}{4} + \frac{\varepsilon}{5} \leq \frac{\varepsilon}{2}$ and by the triangle inequality $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{q}) \leq \mathrm{d}_{\mathrm{TV}}(\mathbf{q}, \mathbf{h}) + \mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{h}) \leq \varepsilon/2 + 6\varepsilon/25 \leq \varepsilon$ as required. □

If $\widetilde{\sigma} \geq \alpha/\varepsilon^2$ (corresponding to a "big variance" $s^2 = \Omega(1/\varepsilon^2)$), then we take an additional $O(|S|/\varepsilon^2)$ samples from $\mathbf{p}$ and use them to learn a shifted binomial using algorithms `Learn-Poisson` and `Locate-Binomial` from [65] that is within $O(\varepsilon/\sqrt{|S|})$ total variation distance from $\mathbf{p}$. If these succeed, we can check if its Fourier coefficients are close using the method in Algorithm 11 (`Project-k-SIIRV`). As we can compute the Fourier coefficients of a shifted binomial easily, this overall takes time $\mathrm{poly}(1/\varepsilon)$.

2.2.4 The SIIRV Tester

We are now ready to describe the algorithm behind Theorem 2.2.1, and establish the theorem.

2.2.4.1 Analyzing the subroutines

We start with some useful structural results, which will be necessary to our analysis. The first is the following lemma from [85]:

Lemma 2.2.8 ([85, Lemma 2.3])**.** *Let $\mathbf{p} \in \mathcal{SIIRV}_{n,k}$ with $\sqrt{\mathrm{Var}_{X \sim \mathbf{p}}[X]} = s$, $1/2 > \delta > 0$, and $M \in \mathbb{Z}_+$ with $M > s$. Let $\widehat{\mathbf{p}}$ be the discrete Fourier transform of $\mathbf{p}$ modulo M. Then, we have*

Algorithm 12 Algorithm `Test-SIIRV`

Require: sample access to a distribution $\mathbf{p} \in \Delta(\mathbb{N})$, parameters $n, k \geq 1$ and $\varepsilon \in (0,1]$

▷ Let C, C', C'' be sufficiently large universal constants

Draw $O(k)$ samples from $\mathbf{p}$ and compute as in Claim 2.2.11: (a) $\widetilde{\sigma}^2$, a tentative factor-2 approximation to $\mathrm{Var}_{X\sim\mathbf{p}}[X]+1$, and (b) $\widetilde{\mu}$, a tentative approximation to $\mathbb{E}_{X\sim\mathbf{p}}[X]$ to within one standard deviation.

if If $\widetilde{\sigma} > 2k\sqrt{n}$ **then**

  return reject ▷ Blatant violation of (n,k)-SIIRV-iness

end if

if $\widetilde{\sigma} \leq 2k\sqrt{\ln \frac{10}{\varepsilon}}$ **then**

  Set $M \leftarrow 1 + 2\left\lceil 15k \ln \frac{10}{\varepsilon} \right\rceil$, and let $I \leftarrow [\lfloor \widetilde{\mu} \rfloor - \frac{M-1}{2}, \lfloor \widetilde{\mu} \rfloor + \frac{M-1}{2}]$; and $S \leftarrow [\![M]\!]$

  Draw $O(1/\varepsilon)$ samples from $\mathbf{p}$, to distinguish between $\mathbf{p}(I) \leq 1 - \frac{\varepsilon}{4}$ and $\mathbf{p}(I) > 1 - \frac{\varepsilon}{5}$. If the former is detected, **return** reject

  Take $N = C\left(\frac{|S|}{\varepsilon^2}\right) = O\left(\frac{k}{\varepsilon^2}\log\frac{1}{\varepsilon}\right)$ samples from $\mathbf{p}$ to get an empirical distribution $\mathbf{h}$

else

  Set $M \leftarrow 1 + 2\left\lceil 4\widetilde{\sigma}\sqrt{\ln(4/\varepsilon)} \right\rceil$, and let $I \leftarrow [\lfloor \widetilde{\mu} \rfloor - \frac{M-1}{2}, \lfloor \widetilde{\mu} \rfloor + \frac{M-1}{2}]$

  Draw $O(1/\varepsilon)$ samples from $\mathbf{p}$, to distinguish between $\mathbf{p}(I) \leq 1 - \frac{\varepsilon}{4}$ and $\mathbf{p}(I) > 1 - \frac{\varepsilon}{5}$. If the former is detected, **return** reject

  Let $\delta \leftarrow \frac{\varepsilon}{C''\sqrt{k \log \frac{k}{\varepsilon}}}$, and

$$S \leftarrow \left\{ \xi \in [M-1] : \exists a, b \in \mathbb{Z}, 0 \leq a \leq b < k \text{ s.t. } |\xi/M - a/b| \leq C' \frac{\sqrt{\ln(1/\delta)}}{4\widetilde{\sigma}} \right\} .$$

  Simulating sample access to $\mathbf{p}' \stackrel{\text{def}}{=} \mathbf{p} \bmod M$, call Algorithm 9 on $\mathbf{p}'$ with parameters M, $\frac{\varepsilon}{5\sqrt{M}}$, $b = \frac{16k}{\widetilde{\sigma}}$, and S. If it outputs reject, then **return** reject; otherwise, let $\widehat{\mathbf{h}} = (\widehat{\mathbf{h}}(\xi))_{\xi\in S}$ denote the collection of Fourier coefficients it outputs, and $\mathbf{h}$ their inverse Fourier transform (modulo M) ▷ Do not actually compute $\mathbf{h}$

end if

Projection Step: Check whether $\mathrm{d}_{\mathrm{TV}}(\mathbf{h}, \mathcal{SIIRV}_{n,k}) \leq \frac{\varepsilon}{2}$ (as in Section 2.2.3), and **return** accept if it is the case. If not, **return** reject. ▷ Mostly computational step

(i) Let $\mathcal{L} = \mathcal{L}(\delta, M, s) \stackrel{\text{def}}{=} \left\{ \xi \in [M-1] \mid \exists a, b \in \mathbb{Z}, 0 \leq a \leq b < k \text{ such that } |\xi/M - a/b| < \frac{\sqrt{\ln(1/\delta)}}{2s} \right\}$. *Then,* $|\widehat{\mathbf{p}}(\xi)| \leq \delta$ *for all* $\xi \in [M-1] \setminus \mathcal{L}$. *That is,* $|\widehat{\mathbf{p}}(\xi)| > \delta$ *for at most* $|\mathcal{L}| \leq Mk^2 s^{-1}\sqrt{\log(1/\delta)}$ *values of* ξ .

(ii) At most $4Mks^{-1}\sqrt{\log(1/\delta)}$ *many integers* $0 \leq \xi \leq M-1$ *have* $|\widehat{\mathbf{p}}(\xi)| > \delta$.

Next, we provide a simple structural lemma, bounding the ℓ_2 norm of any (n,k)-SIIRV as a function of k and its variance only:

Lemma 2.2.9 (Any (n,k)-SIIRV modulo M has small ℓ_2 norm)**.** *If* $\mathbf{p} \in \mathcal{S}_{n,k}$ *has variance* s^2*, then the distribution* $\mathbf{p}'$ *defined as* $\mathbf{p}' \stackrel{\text{def}}{=} \mathbf{p} \bmod M$ *satisfies* $\|\mathbf{p}'\|_2 \leq \sqrt{\frac{8k}{s}}$.

Proof of Lemma 2.2.9. By Plancherel, we have $\|\mathbf{p}'\|_2^2 = \frac{1}{M}\sum_{\xi=0}^{M-1}|\widehat{\mathbf{p}'}(\xi)|^2 = \frac{1}{M}\sum_{\xi=0}^{M-1}|\widehat{\mathbf{p}}(\xi)|^2$, the second

equality due to the definition of $\widehat{\mathbf{p}'}$. Indeed, for any $\xi \in [\![M]\!]$,

$$\widehat{\mathbf{p}'}(\xi) = \sum_{j=0}^{M-1} e^{-2i\pi\frac{j\xi}{M}}\mathbf{p}'(j) = \sum_{j=0}^{M-1} e^{-2i\pi\frac{j\xi}{M}} \sum_{\substack{j'\in\mathbb{N}\\ j'=j \bmod M}} \mathbf{p}(j') = \sum_{j=0}^{M-1}\sum_{\substack{j'\in\mathbb{N}\\ j'=j \bmod M}} e^{-2i\pi\frac{j'\xi}{M}}\mathbf{p}(j')$$
$$= \sum_{j\in\mathbb{N}} e^{-2i\pi\frac{j'\xi}{M}}\mathbf{p}(j') = \widehat{\mathbf{p}}(\xi)$$

as $u \mapsto e^{-2i\pi u}$ is 1-periodic. Since $|\widehat{\mathbf{p}}(\xi)| \le 1$ for every $\xi \in [\![M]\!]$ (as $\widehat{\mathbf{p}}(\xi) = \mathbb{E}_{j\sim\mathbf{p}}[e^{-2i\pi\frac{j\xi}{M}}]$), we can upper bound the RHS as

$$\frac{1}{M}\sum_{\xi=0}^{M-1}|\widehat{\mathbf{p}}(\xi)|^2 \le \frac{1}{M}\sum_{r\ge 0}\sum_{\xi:\frac{1}{2^{r+1}}<|\widehat{\mathbf{p}}(\xi)|\le\frac{1}{2^r}}|\widehat{\mathbf{p}}(\xi)|^2 \le \frac{1}{M}\sum_{r\ge 0}\frac{1}{2^{2r}}\left|\left\{\,\xi\in[\![M]\!]:\ \frac{1}{2^{r+1}}<|\widehat{\mathbf{p}}(\xi)|\,\right\}\right|.$$

Invoking Lemma 2.2.8(ii) with parameter δ set to $\frac{1}{2^{r+1}}$, we get that $\left|\left\{\,\xi\in[\![M]\!]:\ \frac{1}{2^{r+1}}<|\widehat{\mathbf{p}}(\xi)|\,\right\}\right| \le 4Mks^{-1}\sqrt{r+1}$, from which

$$\|\mathbf{p}'\|_2^2 \le \frac{4k}{s}\sum_{r\ge 0}\frac{\sqrt{r+1}}{2^{2r}} \le \frac{8k}{s}$$

as desired. □

Finally, we will use the simple fact below – which follows immediately from [85, Claim 2.4] – to bound the running time of our algorithm:

Fact 2.2.10. *For S as defined in Step 13, we have*

$$|S| \le Mk^2\frac{C'}{2\widetilde{\sigma}}\sqrt{\ln\frac{1}{\delta}} \le 100C'k^2\sqrt{\ln\frac{4}{\varepsilon}}\sqrt{\ln\frac{k}{\varepsilon}+\log\log\frac{k}{\varepsilon}+\frac{1}{2}\ln(16C'')} \le C''k^2\log^2\frac{k}{\varepsilon}$$

for a suitably large choice of the constant $C'' > 0$; from which we get $\delta \le \frac{1}{4\sqrt{|S|}}$.

With this in hand, we argue that with high probability, the estimates obtained in Step 2 will be accurate enough for our purposes. (The somewhat odd statement below, stating two distinct guarantees where the second implies the first, is due to the following: Eq. (2.29) will be the guarantee that (the completeness analysis of) our algorithm relies on, while the second, slightly stronger one, will only be used in the particular implementation of the "projection step" (Step 16) from Section 2.2.3.)

Claim 2.2.11 (Estimating the first two moments (if $\mathbf{p}$ is a SIIRV))**.** *With probability at least $19/20$ over the $O(k)$ draws from $\mathbf{p}$ in Step 2, the following holds. If $\mathbf{p} \in \mathcal{SIIRV}_{n,k}$, the estimates $\widetilde{\sigma}, \widetilde{\mu}$ defined as the empirical mean and (unbiased) empirical variance meet the guarantees stated in Step 2 of the algorithm, namely*

$$\frac{1}{2} \le \frac{\widetilde{\sigma}^2}{\mathrm{Var}_{X\sim\mathbf{p}}[X]+1} \le 2, \qquad |\widetilde{\mu}-\mathbb{E}_{X\sim\mathbf{p}}[X]| \le \sqrt{\mathrm{Var}_{X\sim\mathbf{p}}[X]} \tag{2.29}$$

We even have a quantitatively slightly stronger guarantee: $\frac{2}{3} \le \frac{\widetilde{\sigma}^2}{\mathrm{Var}_{X\sim\mathbf{p}}[X]+1} \le \frac{3}{2}$, and $|\widetilde{\mu}-\mathbb{E}_{X\sim\mathbf{p}}[X]| \le$

$\frac{1}{2}\sqrt{\mathrm{Var}_{X\sim\mathbf{p}}[X]}$.

Proof. We handle the estimation of the mean and variance separately.

Estimating the mean. $\widetilde{\mu}$ will be the usual empirical estimator, namely $\widetilde{\mu} \stackrel{\text{def}}{=} \frac{1}{m}\sum_{i=1}^{m} X_i$ for $X_1, \dots, X_m$ independently drawn from $\mathbf{p}$. Since $\mathbb{E}[\widetilde{\mu}] = \mathbb{E}_{X\sim\mathbf{p}}[X]$ and $\mathrm{Var}[\widetilde{\mu}] = \frac{1}{m}\mathrm{Var}_{X\sim\mathbf{p}}[X]$, Chebyshev's inequality guarantees that

$$\Pr[|\widetilde{\mu} - \mathbb{E}_{X\sim\mathbf{p}}[X]| > \frac{1}{2}\sqrt{\mathrm{Var}_{X\sim\mathbf{p}}[X]}] \le \frac{4}{m}$$

which can be made at most $1/200$ by choosing $m \ge 800$.

Estimating the variance. The variance estimation is exactly the same as in [65, Lemma 6], observing that their argument only requires that $\mathbf{p}$ be the distribution of a sum of independent random variables (not necessarily a Poisson Binomial distribution). Namely, they establish that,[6] letting $\widetilde{\sigma}^2 \stackrel{\text{def}}{=} \frac{1}{m-1}\sum_{i=1}^{m}(X_i - \frac{1}{m}\sum_{j=1}^{m} X_j)^2$ be the (unbiased) sample variances, and $s^2 \stackrel{\text{def}}{=} \mathrm{Var}_{X\sim\mathbf{p}}[X]$,

$$\Pr[\left|\widetilde{\sigma}^2 - s^2\right| > \alpha(1+s^2)] \le \frac{4s^4 + k^2s^2}{\alpha^2(1+s^2)^2}\frac{1}{m} \le \frac{4s^4 + s^2}{\alpha^2(1+s^2)^2}\cdot\frac{k^2}{m} \le \frac{4k^2}{\alpha^2 m}$$

which for $\alpha = 1/3$ is at most $9/200$ by choosing $m \ge 800k$.

A union bound completes the proof, giving a probability of error at most $\frac{1}{200} + \frac{9}{200} = \frac{1}{20}$. □

Claim 2.2.12 (Checking the effective support)**.** *With probability at least* $19/20$ *over the draws from* $\mathbf{p}$ *in Step 12, the following holds.*

- *if* $\mathbf{p} \in \mathcal{SIIRV}_{n,k}$ *and* (2.29) *holds, then* $\mathbf{p}(I) \ge 1 - \frac{\varepsilon}{5}$ *and the algorithm does not output* reject *in Step 8 nor 12;*
- *if* $\mathbf{p}$ *puts probability mass more than* $\frac{\varepsilon}{4}$ *outside of* I*, then the algorithm outputs* reject *in Step 8 or 12.*

Proof. Suppose first $\mathbf{p} \in \mathcal{SIIRV}_{n,k}$ and (2.29) holds, and set $s \stackrel{\text{def}}{=} \sqrt{\mathrm{Var}_{X\sim\mathbf{p}}[X]}$ and $\mu \stackrel{\text{def}}{=} \mathbb{E}_{X\sim\mathbf{p}}[X]$ as before. By Bennett's inequality applied to X, we have

$$\Pr[X > \mu + t] \le \exp\left(-\frac{s^2}{k^2}\vartheta\left(\frac{kt}{s^2}\right)\right) \tag{2.30}$$

for any $t > 0$, where $\vartheta\colon \mathbb{R}_+^* \to \mathbb{R}$ is defined by $\vartheta(x) = (1+x)\ln(1+x) - x$.

- If the algorithm reaches Step 8, then $s \le 4k\sqrt{\ln\frac{10}{\varepsilon}}$. Setting $t = \alpha\cdot k\ln\frac{10}{\varepsilon}$ in Eq. (2.30) (for $\alpha > 0$ to be determined shortly), and $u = \frac{kt}{s^2} = \alpha\frac{k^2}{s^2}\ln\frac{10}{\varepsilon} \ge \frac{\alpha}{16}$,

$$\frac{s^2}{k^2}\vartheta\left(\frac{kt}{s^2}\right) = \alpha\ln\frac{10}{\varepsilon}\cdot\frac{\vartheta(u)}{u} \ge \left(16\vartheta\left(\frac{\alpha}{16}\right)\right)\ln\frac{10}{\varepsilon} \ge \ln\frac{10}{\varepsilon}$$

[6] [65, Lemma 6] actually only deals with the case $k = 2$; but the bound we state follows immediately from their proof and the simple

since $\frac{\vartheta(x)}{x} \geq \frac{\vartheta(\alpha/16)}{\alpha/16}$ for all $x \geq \frac{\alpha}{16}$; the last inequality for $\alpha \geq \alpha^* \simeq 2.08$ chosen to be the solution to $16\vartheta\left(\frac{\alpha^*}{16}\right) = 1$. Thus, $\Pr[X > \mu + t] \leq \frac{\varepsilon}{10}$. Similarly, we have $\Pr[X < \mu - t] \leq \frac{\varepsilon}{10}$. As $\mu - 2t \leq \mu - s \leq \widetilde{\mu} \leq \mu + s \leq \mu + 2t$, we get $\Pr[X \in I] \geq 1 - \frac{\varepsilon}{5}$ as claimed.

- If the algorithm reaches Step 12, then $s \geq k\sqrt{\ln\frac{10}{\varepsilon}}$ and $M = 1+2\left\lceil 6\widetilde{\sigma}\sqrt{\ln\frac{10}{\varepsilon}})\right\rceil \geq 1+2\left\lceil 3s\sqrt{\ln\frac{10}{\varepsilon}})\right\rceil$. Setting $t = \beta s\sqrt{\ln\frac{10}{\varepsilon}}$ in Eq. (2.30) (for $\beta > 0$ to be determined shortly), and $u = \frac{kt}{s^2} = \beta\frac{k}{s}\sqrt{\ln\frac{10}{\varepsilon}} \leq \beta$,

$$\frac{s^2}{k^2}\vartheta\left(\frac{kt}{s^2}\right) = \frac{t^2}{s^2}\cdot\frac{\vartheta(u)}{u^2} = \beta^2\ln\frac{10}{\varepsilon}\cdot\frac{\vartheta(u)}{u^2} \geq \ln\frac{10}{\varepsilon}$$

since $\frac{\vartheta(x)}{x^2} \geq \frac{\vartheta(\beta)}{\beta^2}$ for all $x \in (0,\beta]$; the last inequality for $\beta = e - 1 \simeq 1.72$ chosen to be the solution to $\vartheta(\beta) = 1$. Thus, $\Pr[X > \mu + t] \leq \frac{\varepsilon}{10}$. Similarly, it holds $\Pr[X < \mu - t] \leq \frac{\varepsilon}{10}$. Now note that $\lfloor\widetilde{\mu}\rfloor + (M-1)/2 \geq (\mu - s) + \lceil 2s\sqrt{\ln\frac{10}{\varepsilon}})\rceil \geq \mu + t$ and $\lfloor\widetilde{\mu}\rfloor - (M-1)/2 \leq \mu - t$, implying that X is in $[\lfloor\widetilde{\mu}\rfloor - (M-1)/2, \lfloor\widetilde{\mu}\rfloor + (M-1)/2]$ with probability at least $1 - \frac{\varepsilon}{5}$ as desired.

To conclude and establish the conclusion of the first item, as well as the second item, recall that distinguishing with probability $19/20$ between the cases $\mathbf{p}(\bar{I}) \leq \frac{\varepsilon}{5}$ and $\mathbf{p}(\bar{I}) > \frac{\varepsilon}{4}$ can be done with $O(1/\varepsilon)$ samples. □

Claim 2.2.13 (Learning when the effective support is small)**.** *If* $\mathbf{p}$ *satisfies* $\mathbf{p}(I) \geq 1 - \frac{\varepsilon}{4}$*, and the "If" statement at Step 6 holds, then with probability at least* $19/20$ *the empirical distribution* $\mathbf{h}$ *obtained in Step 9 satisfies (i)* $\mathrm{d_{TV}}(\mathbf{p},\mathbf{h}) \leq \frac{\varepsilon}{2}$ *and (ii)* $\|\widehat{\mathbf{p}} - \widehat{\mathbf{h}}\|_2 \leq \frac{\varepsilon^2}{100}$.

Proof. The first item, (i), follows from standard bounds on the rate of convergence of the empirical distribution (namely, that $O(r/\varepsilon^2)$ samples suffice for it to approximate an arbitrary distribution over support of size r up to total variation distance ε). Recalling that in this branch of the algorithm, $S = [\![M]\!]$ with $M = O(k\log(1/\varepsilon))$, the second item, (ii), is proven by the same argument as in (the first bullet in) the proof of Theorem 2.2.4. □

Claim 2.2.14 (Any (n,k)-SIIRV puts near all its Fourier mass in S)**.** *If* $\mathbf{p} \in \mathcal{SIIRV}_{n,k}$ *and* (2.29) *holds, then* $\|\widehat{\mathbf{p}}\mathbb{1}_{\bar{S}}\|_2^2 = \sum_{\xi\notin S}|\widehat{\mathbf{p}}(\xi)|^2 \leq \frac{\varepsilon^2}{100}$.

Proof. Since $\mathbf{p} \in \mathcal{SIIRV}_{n,k}$, our assumptions imply that (with the notations of Lemma 2.2.8) the set of large Fourier coefficients satisfies $\{\,\xi \in [M-1] :\ |\widehat{\mathbf{p}}(\xi)| > \delta\,\} \subseteq \mathcal{L}(\delta, M, s) \subseteq S$. Therefore, $\xi \notin S$ implies $|\widehat{\mathbf{p}}(\xi)| \leq \delta$. We then can conclude as follows: applying Lemma 2.2.8 (ii) with parameter $\delta 2^{-r-1}$ for each $r \geq 0$, this is at most

$$\begin{aligned}\sum_{r\geq 0}(\delta 2^{-r})^2\left|\{\,\xi :\ |\widehat{\mathbf{p}}(\xi)| > \delta 2^{-r-1}\,\}\right| &\leq \frac{4Mk\delta^2}{s}\sum_{r\geq 0}4^{-r}\sqrt{\log(2^{r+2}/\delta)}\\ &\leq \frac{4Mk\delta^2}{s}\sqrt{\log\frac{1}{\delta}}\sum_{r\geq 0}4^{-r}\sqrt{\log(2^{r+1})}\\ &\leq \frac{12Mk\delta^2}{s}\sqrt{\log\frac{1}{\delta}} = O(\varepsilon^2)\end{aligned} \tag{2.31}$$

again at most $\frac{\varepsilon^2}{100}$ for big enough C'' in the definition of δ. □

2.2.4.2 Putting it together

In what follows, we implicitly assume that I (as defined in Step 11 of Algorithm 12) is equal to $[\![M]\!]$. This can be done without loss of generality, as this is just a shifting of the interval and all our Fourier arguments are made modulo M.

Lemma 2.2.15 (Putting it together: completeness)**.** *If $\mathbf{p} \in \mathcal{SIIRV}_{n,k}$, then the algorithm outputs* accept *with probability at least $3/5$.*

Proof. Assume $\mathbf{p} \in \mathcal{SIIRV}_{n,k}$. We condition on the estimates obtained in Step 2 to meet their accuracy guarantees, which by Claim 2.2.11 holds with probability at least $19/20$: that is, we hereafter assume Eq. (2.29) holds. Since the variance of any (n,k)-SIIRV is at most $s^2 \le nk^2$, we consequently have $\widetilde{\sigma} \le 2k\sqrt{n}$ and the algorithm does not output reject in Step 3.

- **Case 1:** the branch in Step 6 is taken. In this case, by Claim 2.2.12 the algorithm does not output reject in Step 8 with probability $19/20$. Since $\mathbf{p}(I) \ge 1 - \frac{\varepsilon}{4}$, by Claim 2.2.13 we get that with probability at least $19/20$ it is the case that $\mathrm{d}_{\mathrm{TV}}(\mathbf{p},\mathbf{h}) \le \frac{\varepsilon}{2}$, and therefore the computational check in Step 16 will succeed, and return accept. Overall, by a union bound the algorithm is successful with probability at least $1 - 3/20 > 3/5$.
- **Case 2:** the branch in Step 10 is taken. In this case, by Claim 2.2.12 the algorithm does not output reject in Step 12 with probability $19/20$. From Lemma 2.2.9, we know that $\mathbf{p}'$ as defined in Step 14 satisfies $\|\mathbf{p}'\|_2^2 \le \frac{8k}{s} \le \frac{16k}{\widetilde{\sigma}} = b$. Moreover, Claim 2.2.14 guarantees that $\|\widehat{\mathbf{p}'}\mathbb{1}_{\bar{S}}\|_2 \le \frac{\varepsilon}{10\sqrt{M}} = \frac{\varepsilon'}{2}$ (for $\varepsilon' = \frac{\varepsilon}{5\sqrt{M}}$). Since Step 14 calls Algorithm 9 with parameters M, ε', b, and S, Item 3 of Theorem 2.2.4 ensures that (with probability at least $7/10$) the algorithm will not output reject in Step 14, but instead return the S-sparse Fourier transform of some $\mathbf{h}$ supported on $[\![M]\!]$ with $\|\mathbf{p}' - \mathbf{h}\|_2 \le \frac{6}{5}\varepsilon' = \frac{6\varepsilon}{25\sqrt{M}}$. By Cauchy–Schwarz, we then have $\|\mathbf{p}' - \mathbf{h}\|_1 \le \sqrt{M}\|\mathbf{p}' - \mathbf{h}\|_2 \le \frac{6\varepsilon}{25}$, i.e. $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}',\mathbf{h}) \le \frac{3\varepsilon}{25}$. But since $\mathrm{d}_{\mathrm{TV}}(\mathbf{p},\mathbf{p}') \le \frac{\varepsilon}{4}$, we get $\mathrm{d}_{\mathrm{TV}}(\mathbf{p},\mathbf{h}) \le \frac{\varepsilon}{4} + \frac{3\varepsilon}{25} < \frac{\varepsilon}{2}$, and the computational check in Step 16 will succeed, and return accept. Overall, by a union bound the algorithm accepts with probability at least $1 - (1/20 + 1/20 + 3/10) = 3/5$.

□

Lemma 2.2.16 (Putting it together: soundness)**.** *If $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathcal{SIIRV}_{n,k}) > \varepsilon$, then the algorithm outputs* reject *with probability at least $3/5$.*

Proof. We will proceed by contrapositive, and show that if the algorithm returns accept with probability at least $3/5$ then $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathcal{SIIRV}_{n,k}) \le \varepsilon$. Depending on the branch of the algorithm followed, we assume the samples taken either in

- Steps 2, 8, 9, meet the guarantees of Claims 2.2.11 to 2.2.13 (by a union bound, this happens with probability at least $1 - 3/20 > 2/3$); or
- Steps 2, 12, 14 meet the guarantees of Claims 2.2.11 and 2.2.12 and Theorem 2.2.4 (by a union bound, this happens with probability at least $1 - (1/20 + 1/20 + 3/10) = 3/5$).

In particular, we hereafter assume that $\widetilde{\sigma} \le 2k\sqrt{n}$.

- **Case 1:** the branch in Step 6 is taken.

  By the above discussion, we have $\mathbf{p}(I) \ge 1 - \frac{\varepsilon}{4}$ by Claim 2.2.12 so Claim 2.2.13 and our conditioning ensure that the empirical distribution $\mathbf{h}$ is such that $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{h}) \le \frac{\varepsilon}{2}$. Since the algorithm did not reject in Step 16, there exists a (n,k)-SIIRV $\mathbf{p}^*$ such that $\mathrm{d}_{\mathrm{TV}}(\mathbf{h}, \mathbf{p}^*) \le \frac{\varepsilon}{2}$: by the triangle inequality, $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathcal{SIIRV}_{n,k}) \le \mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{q}^*) \le \varepsilon$.

- **Case 2:** the branch in Step 10 is taken.

  In this case, we have $\mathbf{p}(I) \ge 1 - \frac{\varepsilon}{4}$ by Claim 2.2.12. Furthermore, as the algorithm did not output reject on Step 14, by Theorem 2.2.4 we know that the inverse Fourier transform (modulo M) $\mathbf{h}$ of the S-sparse collection of Fourier coefficients $\widehat{\mathbf{h}}$ returned satisfies $\|\mathbf{h} - \mathbf{p}'\|_2 \le \frac{6\varepsilon}{25\sqrt{M}}$ which by Cauchy–Schwarz implies, as both $\mathbf{h}$ and $\mathbf{p}'$ are supported on $[\![M]\!]$, that $\|\mathbf{h} - \mathbf{p}'\|_1 \le \frac{6\varepsilon}{25}$, or equivalently $\mathrm{d}_{\mathrm{TV}}(\mathbf{h}, \mathbf{p}') \le \frac{3\varepsilon}{25}$.

  Finally, since the algorithm outputted accept in Step 16, there exists $\mathbf{p}^* \in \mathcal{SIIRV}_{n,k}$ (supported on $[\![M]\!]$) such that $\mathrm{d}_{\mathrm{TV}}(\mathbf{h}, \mathbf{p}^*) \le \frac{\varepsilon}{2}$, and by the triangle inequality

$$\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{p}^*) \le \mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{p}') + \mathrm{d}_{\mathrm{TV}}(\mathbf{h}, \mathbf{p}') + \mathrm{d}_{\mathrm{TV}}(\mathbf{h}, \mathbf{p}^*) \le \frac{\varepsilon}{4} + \frac{3\varepsilon}{25} + \frac{\varepsilon}{2} \le \varepsilon$$

  and thus $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathcal{SIIRV}_{n,k}) \le \mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{p}^*) \le \varepsilon$.

□

Lemma 2.2.17 (Putting it together: sample complexity)**.** *The algorithm has sample complexity* $O\left(\frac{kn^{1/4}}{\varepsilon^2}\log^{1/4}\frac{1}{\varepsilon} + \frac{k^2}{\varepsilon^2}\log^2\frac{k}{\varepsilon}\right)$.

Proof. Algorithm 12 takes samples in Steps 2, 8, 12, and 14. The sample complexity is dominated by Steps 9 and 14, which take respectively N and

$$\begin{aligned} O\left(\frac{\sqrt{b}}{(\varepsilon/\sqrt{M})^2} + \frac{|S|}{M(\varepsilon/\sqrt{M})^2} + \sqrt{M}\right) &= O\left(\frac{\sqrt{k\widetilde{\sigma}}}{\varepsilon^2}\sqrt[4]{\log\frac{1}{\varepsilon}} + \frac{|S|}{\varepsilon^2} + \sqrt{\widetilde{\sigma}}\sqrt[4]{\log\frac{1}{\varepsilon}}\right) \\ &= O\left(\frac{kn^{1/4}}{\varepsilon^2}\log^{1/4}\frac{1}{\varepsilon} + \frac{k^2}{\varepsilon^2}\log^2\frac{k}{\varepsilon}\right) \end{aligned}$$

samples; recalling that Step 3 ensured that $\widetilde{\sigma} \le 2k\sqrt{n}$ and that $|S| = O\big(k^2\log^2\frac{k}{\varepsilon}\big)$ by Fact 2.2.10. □

Lemma 2.2.18 (Putting it together: time complexity)**.** *The algorithm runs in time* $O\Big(\frac{k^4n^{1/4}}{\varepsilon^2}\log^4\frac{k}{\varepsilon}\Big) +$

observation that the excess kurtosis κ of an (n,k)-SIIRV with variance s^2 is at most k^2/s^2.

$T(n,k,\varepsilon)$, *where* $T(n,k,\varepsilon) = n(k/\varepsilon)^{O(k\log(k/\varepsilon))}$ *is the running time of the projection subroutine of Step 16.*

Proof. The running time, depending on the branch taken, is either $O(N) + T(n,k,\varepsilon)$ for the first or $O\left(|S|\left(\frac{kn^{1/4}}{\varepsilon^2}\log^{1/4}\frac{1}{\varepsilon} + \frac{k^2}{\varepsilon^2}\log^2\frac{k}{\varepsilon}\right)\right) + T(n,k,\varepsilon)$ for the second (the latter from the running time of Algorithm 9). Recalling that $|S| = O\left(k^2\log^2\frac{k}{\varepsilon}\right)$ by Fact 2.2.10 yields the claimed running time. □

2.2.5 The General Tester

In this section, we abstract the ideas underlying the (n,k)-SIIRV from Section 2.2.4, to provide a general testing framework. In more detail, our theorem (Theorem 2.2.19) has the following flavor: if $\mathcal{P}$ is a property of distributions such that every $\mathbf{p} \in \mathcal{P}$ has both (i) small effective support and (ii) sparse effective Fourier support, then one can test membership in $\mathcal{P}$ with $O(\sqrt{sM}/\varepsilon^2 + s/\varepsilon^2)$ samples (where M and s are the bounds on the effective support and effective Fourier support, respectively). As a caveat, we do require that the sparse effective Fourier support S be independent of $\mathbf{p} \in \mathcal{P}$, i.e., is a characteristic of the class $\mathcal{P}$ itself.

The high-level idea is then quite simple: the algorithm proceeds in three stages, namely the *effective support test*, the *Fourier effective support test*, and the *projection step*. In the first, it takes some samples from $\mathbf{p}$ to identify what should be the effective support I of $\mathbf{p}$, if $\mathbf{p}$ did have the property: and then checks that indeed $|I| \leq M$ (as it should) and that $\mathbf{p}$ puts probability mass $1 - O(\varepsilon)$ on I.

In the second stage, it invokes the Fourier testing algorithm of Section 2.2.2 to verify that $\widehat{\mathbf{p}}$ indeed puts very little Fourier mass outside of S; and, having verified this, learns very accurately the set of Fourier coefficients of $\mathbf{p}$ on this set S, in L_2 distance.

At this point, either the algorithm has detected that $\mathbf{p}$ violates some required characteristic of the distributions in $\mathcal{P}$, in which case it has rejected already; or is guaranteed to have *learned* a good approximation $\mathbf{h}$ of $\mathbf{p}$, by the Fourier learning performed in the second stage. It only remains to perform the third stage, which "projects" this good approximation $\mathbf{h}$ of $\mathbf{p}$ onto $\mathcal{P}$ to verify that $\mathbf{h}$ is close to some distribution $\mathbf{p}^* \in \mathcal{P}$ (as it should if indeed $\mathbf{p} \in \mathcal{P}$).

Theorem 2.2.19 (General Testing Statement)**.** *Assume* $\mathcal{P} \subseteq \Delta(\mathbb{N})$ *is a property of distributions satisfying the following. There exist* $S\colon (0,1] \to 2^{\mathbb{N}}$, $M\colon (0,1] \to \mathbb{N}$, *and* $q_I\colon (0,1] \to \mathbb{N}$ *such that, for every* $\varepsilon \in (0,1]$,

1. *Fourier sparsity: for all* $\mathbf{p} \in \mathcal{P}$, *the Fourier transform (modulo* $M(\varepsilon)$*) of* $\mathbf{p}$ *is concentrated on* $S(\varepsilon)$*: namely,* $\|\widehat{\mathbf{p}}\mathbb{1}_{\overline{S(\varepsilon)}}\|_2^2 \leq \frac{\varepsilon^2}{100}$.
2. *Support sparsity: for all* $\mathbf{p} \in \mathcal{P}$, *there exists an interval* $I(\mathbf{p}) \subseteq \mathbb{N}$ *with* $|I(\mathbf{p})| \leq M(\varepsilon)$ *such that (i)* $\mathbf{p}$ *is concentrated on* $I(\mathbf{p})$*: namely,* $\mathbf{p}(I(\mathbf{p})) \geq 1 - \frac{\varepsilon}{5}$ *and (ii)* $I(\mathbf{p})$ *can be identified with probability at least* $19/20$ *from* $q_I(\varepsilon)$ *samples from* $\mathbf{p}$.
3. *Projection: there exists a procedure* $\textsc{Project}_{\mathcal{P}}$ *which, on input* $\varepsilon \in (0,1]$ *and the explicit description of a distribution* $\mathbf{h} \in \Delta(\mathbb{N})$, *runs in time* $T(\varepsilon)$*; and outputs* accept *if* $\mathrm{d}_{\mathrm{TV}}(\mathbf{h},\mathcal{P}) \leq \frac{2\varepsilon}{5}$, *and* reject *if* $\mathrm{d}_{\mathrm{TV}}(\mathbf{h},\mathcal{P}) > \frac{\varepsilon}{2}$ *(and can answer either otherwise).*
4. *(Optional)* L_2*-norm bound: there exists* $b \in (0,1]$ *such that, for all* $\mathbf{p} \in \mathcal{P}$, $\|\mathbf{p}\|_2^2 \leq b$.

Algorithm 13 Algorithm `Test-Fourier-Sparse-Class`

Require: sample access to a distribution $\mathbf{p} \in \Delta(\mathbb{N})$, parameter $\varepsilon \in (0,1]$, $b \in (0,1]$, functions $S\colon (0,1] \to 2^{\mathbb{N}}$, $M\colon (0,1] \to \mathbb{N}$, $q_I\colon (0,1] \to \mathbb{N}$, and procedure $\textsc{Project}_{\mathcal{P}}$ as in Theorem 2.2.19

Effective Support

  Take $q_I(\varepsilon)$ samples from $\mathbf{p}$ to identify a "candidate set" I. ▷ Guaranteed to work w.p. $19/20$ if $\mathbf{p} \in \mathcal{P}$.

  Draw $O(1/\varepsilon)$ samples from $\mathbf{p}$, to distinguish between $\mathbf{p}(I) \geq 1 - \frac{\varepsilon}{5}$ and $\mathbf{p}(I) < 1 - \frac{\varepsilon}{4}$. ▷ Correct w.p. $19/20$.

  if $|I| > M(\varepsilon)$ or we detected that $\mathbf{p}(I) > \frac{\varepsilon}{4}$ **then**

    return reject

  end if

Fourier Effective Support

  Simulating sample access to $\mathbf{p}' \stackrel{\text{def}}{=} \mathbf{p} \bmod M(\varepsilon)$, call Algorithm 9 on $\mathbf{p}'$ with parameters $M(\varepsilon)$, $\frac{\varepsilon}{5\sqrt{M(\varepsilon)}}$, b, and $S(\varepsilon)$.

  if Algorithm 9 returned reject **then**

    return reject

  end if

  Let $\widehat{\mathbf{h}} = (\widehat{\mathbf{h}}(\xi))_{\xi \in S(\varepsilon)}$ denote the collection of Fourier coefficients it outputs, and $\mathbf{h}$ their inverse Fourier transform (modulo $M(\varepsilon)$) ▷ Do not actually compute $\mathbf{h}$ here.

Projection Step

  Call $\textsc{Project}_{\mathcal{P}}$ on parameters ε and $\mathbf{h}$, and **return** accept if it does, reject otherwise.

Then, there exists a testing algorithm for $\mathcal{P}$, in the usual standard sense: it outputs either accept *or* reject*, and satisfies the following.*

1. *if $\mathbf{p} \in \mathcal{P}$, then it outputs* accept *with probability at least $3/5$;*
2. *if $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathcal{P}) > \varepsilon$, then it outputs* reject *with probability at least $3/5$.*

The algorithm takes

$$O\left(\frac{\sqrt{|S(\varepsilon)|\, M(\varepsilon)}}{\varepsilon^2} + \frac{|S(\varepsilon)|}{\varepsilon^2} + q_I(\varepsilon)\right)$$

samples from $\mathbf{p}$ (if Item 4 holds, one can replace the above bound by $O\left(\frac{\sqrt{b}M(\varepsilon)}{\varepsilon^2} + \frac{|S(\varepsilon)|}{\varepsilon^2} + q_I(\varepsilon)\right)$); and runs in time $O(m\,|S| + T(\varepsilon))$, where m is the sample complexity.

Moreover, whenever the algorithm outputs accept*, it also* learns $\mathbf{p}$*; that is, it provides a hypothesis $\mathbf{h}$ such that $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{h}) \leq \varepsilon$ with probability at least $3/5$.*

We remark that the statement of Theorem 2.2.19 can be made slightly more general; specifically, one can allow the procedure $\textsc{Project}_{\mathcal{P}}$ to have sample access to $\mathbf{p}$ and err with small probability, and further provide it with the Fourier coefficients $\widehat{\mathbf{h}}$ learnt in the previous step.

Proof of Theorem 2.2.19. For convenience, we hereafter write S and M instead of $S(\varepsilon)$ and $M(\varepsilon)$, respectively. Before establishing the theorem, which will be a generalization of (the second branch of) Algorithm 12, we note that it is sufficient to prove the version including Item 4. This is because, if no bound b is provided,

one can fall back to setting $b \stackrel{\text{def}}{=} \frac{|S|+1}{M}$: indeed, for any $\mathbf{p} \in \mathcal{P}$,

$$\|\mathbf{p}\|_2^2 = \|\widehat{\mathbf{p}}\|_2^2 = \|\widehat{\mathbf{p}}\mathbb{1}_S\|_2^2 + \|\widehat{\mathbf{p}}\mathbb{1}_{\bar{S}}\|_2^2 = \frac{1}{M}\sum_{\xi\in S}|\widehat{\mathbf{p}}(\xi)|^2 + \|\widehat{\mathbf{p}}\mathbb{1}_{\bar{S}}\|_2^2 \leq \frac{|S|}{M} + \frac{\varepsilon^2}{100M} = \frac{|S| + \frac{\varepsilon^2}{100}}{M} \tag{2.32}$$

from Item 1 and the fact that $|\widehat{\mathbf{p}}(\xi)| \leq 1$ for any $\xi \in [\![M]\!]$. Then, we have $\sqrt{b}M \leq \sqrt{2\frac{|S|}{M}}M = \sqrt{2\,|S|\,M}$, concluding the remark.

The algorithm is given in Algorithm 13. Its sample complexity and running time are immediate from the assumptions on the input parameters, and its description; we thus focus on establishing its correctness.

- Completeness: suppose $\mathbf{p} \in \mathcal{P}$. Then, by definition of q_I and M (Item 2 of the theorem), we have that with probability at least $19/20$ the interval I identified in Step 2 satisfies $\mathbf{p}(I) \geq 1 - \frac{\varepsilon}{5}$ and $|I| \leq M$. In this case, also with probability at least $19/20$ the check in Step 3 succeeds, and the algorithm does not output reject there.

  The call to Algorithm 9 in Step 9 then, with probability at least 7/10, does not output reject, but instead Fourier coefficients $\widehat{H}$ (supported on S) of some $\mathbf{h}$ such that $\mathbf{h}' = \mathbf{h} \bmod M$ satisfies $\|\mathbf{h}' - \mathbf{p}'\|_2 \leq \frac{6}{5} \cdot \frac{\varepsilon}{5\sqrt{M}} = \frac{6\varepsilon}{25\sqrt{M}}$ (this is because of the definition of b and Item 1, which ensure the assumptions of Theorem 2.2.4 are met). Thus $\|\mathbf{h}' - \mathbf{p}'\|_1 \leq \sqrt{M}\|\mathbf{h}' - \mathbf{p}'\|_2 \leq \frac{6\varepsilon}{25}$. Since $\|\mathbf{p} - \mathbf{p}'\|_2 \leq 2 \cdot \frac{\varepsilon}{4}$ (as $\mathbf{p}(I) \geq 1 - \frac{\varepsilon}{4}$ and $\mathbf{p}' = \mathbf{p} \bmod M$), by the triangle inequality

  $$\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{h}') = \frac{1}{2}\|\mathbf{h}' - \mathbf{p}'\|_1 \leq \frac{3\varepsilon}{25} + \frac{\varepsilon}{4} < \frac{2\varepsilon}{5}$$

  and the algorithm returns accept in Step 16 (as promised by Item 3).

  Overall, by a union bound the algorithm is correct with probability at least $1 - (\frac{1}{20} + \frac{1}{20} + \frac{3}{10}) \geq \frac{3}{5}$.

- Soundness: we proceed by contrapositive, and show that if the algorithm returns accept with probability at least $3/5$ then $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathcal{P}) \leq \varepsilon$. We hereafter assume the guarantees of Steps 2, 3, and 9 hold, which by a union bound is the case with probability at least $1 - (\frac{1}{20} + \frac{1}{20} + \frac{3}{10}) \geq \frac{3}{5}$.

  Since the algorithm passed Step 5, we have $\mathbf{p}(I) \geq 1 - \frac{\varepsilon}{4}$ and $|I| \leq M$. Furthermore, as the algorithm did not output reject on Step 9, by Theorem 2.2.4 we know that the inverse Fourier transform (modulo M) $\mathbf{h}$ of the S-sparse collection of Fourier coefficients $\widehat{\mathbf{h}}$ returned satisfies, for $\mathbf{h}' \stackrel{\text{def}}{=} \mathbf{h} \bmod M$,

  $$\|\mathbf{h}' - \mathbf{p}'\|_2 \leq \frac{6\varepsilon}{25\sqrt{M}}$$

  which by Cauchy–Schwarz implies that $\|\mathbf{h} - \mathbf{p}'\|_1 \leq \frac{6\varepsilon}{25}$, or equivalently $\mathrm{d}_{\mathrm{TV}}(\mathbf{h}, \mathbf{p}') \leq \frac{3\varepsilon}{25}$.

  Finally, since the algorithm outputted accept in Step 16, there exists $\mathbf{p}^* \in \mathcal{P}$ (supported on $[\![M]\!]$) such that $\mathrm{d}_{\mathrm{TV}}(\mathbf{h}, \mathbf{p}^*) \leq \frac{\varepsilon}{2}$, and by the triangle inequality

  $$\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{p}^*) \leq \mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{p}') + \mathrm{d}_{\mathrm{TV}}(\mathbf{h}, \mathbf{p}') + \mathrm{d}_{\mathrm{TV}}(\mathbf{h}, \mathbf{p}^*) \leq \frac{\varepsilon}{4} + \frac{3\varepsilon}{25} + \frac{\varepsilon}{2} \leq \varepsilon$$

and thus $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathcal{P}) \leq \mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{p}^*) \leq \varepsilon$.

□

2.2.6 The PMD Tester

In this section, we generalize our Fourier testing approach to higher dimensions, and leverage it to design a testing algorithm for the class of Poisson Multinomial distributions – thus establishing Theorem 2.2.3 (restated below).

Theorem 2.2.20 (Testing PMDs)**.** *Given parameters $k, n \in \mathbb{N}$, $\varepsilon \in (0, 1]$, and sample access to a distribution $\mathbf{p}$ over $\mathbb{N}$, there exists an algorithm (Algorithm 15) which outputs either* accept *or* reject*, and satisfies the following.*

1. *if $\mathbf{p} \in \mathcal{PMD}_{n,k}$, then it outputs* accept *with probability at least $3/5$;*
2. *if $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathcal{PMD}_{n,k}) > \varepsilon$, then it outputs* reject *with probability at least $3/5$.*

Moreover, the algorithm takes $O\left(\frac{n^{(k-1)/4} k^{2k} \log(k/\varepsilon)^k}{\varepsilon^2}\right)$ samples from $\mathbf{p}$, and runs in time $n^{O(k^3)} \cdot (1/\varepsilon)^{O(k^3 \frac{\log(k/\varepsilon)}{\log\log(k/\varepsilon)})^{k-1}}$ or alternatively in time $n^{O(k)} \cdot 2^{O(k^{5k} \log(1/\varepsilon)^{k+2}}$.

The reason for the two different running times is that, for the projection step, one can use either the cover given by [86] or that given by [72], which yield the two statements. In contrast to Section 2.2.4 and Section 2.2.5, for PMDs we will have to use a *multidimensional* Fourier transform, which is a little more complicated – and we define next.

Let $M \in \mathbb{Z}^{k \times k}$ be an integer $k \times k$ matrix. We consider the integer lattice $L = L(M) = M\mathbb{Z}^k \stackrel{\text{def}}{=} \{p \in \mathbb{Z}^k \mid p = Mq, q \in \mathbb{Z}^k\}$, and its dual lattice $L^* = L^*(M) \stackrel{\text{def}}{=} \left\{ \xi \in \mathbb{R}^k : \xi \cdot x \in \mathbb{Z} \text{ for all } x \in L \right\}$. Note that $L^* = (M^T)^{-1}\mathbb{Z}^k$, and that L^* is not necessarily integral. The quotient $\mathbb{Z}^k/L$ is the set of equivalence classes of points in $\mathbb{Z}^k$ such that two points $x, y \in \mathbb{Z}^k$ are in the same equivalence class if, and only if, $x - y \in L$. Similarly, the quotient $L^*/\mathbb{Z}^k$ is the set of equivalence classes of points in L^* such that any two points $x, y \in L^*$ are in the same equivalence class if, and only if, $x - y \in \mathbb{Z}^k$.

The *Discrete Fourier Transform (DFT) modulo M*, $M \in \mathbb{Z}^{k \times k}$, of a function $F \colon \mathbb{Z}^k \to \mathbb{C}$ is the function $\widehat{F}_M \colon L^*/\mathbb{Z}^k \to \mathbb{C}$ defined as $\widehat{F}_M(\xi) \stackrel{\text{def}}{=} \sum_{x \in \mathbb{Z}^k} e(\xi \cdot x) F(x)$. (We will omit the subscript M when it is clear from the context.) Similarly, for the case that F is a probability mass function, we can equivalently write $\widehat{F}(\xi) = \mathbb{E}_{X \sim F}\left[e(\xi \cdot X)\right]$. The *inverse DFT* of a function $\widehat{G} \colon L^*/\mathbb{Z}^k \to \mathbb{C}$ is the function $G \colon A \to \mathbb{C}$ defined on a *fundamental domain* A of $L(M)$ as follows: $G(x) = \frac{1}{|\det(M)|} \sum_{\xi \in L^*/\mathbb{Z}^k} \widehat{G}(x) e(-\xi \cdot x)$. Note that these operations are inverse of each other, namely for any function $F \colon A \to \mathbb{C}$, the inverse DFT of $\widehat{F}$ is identified with F.

With this in hand, Algorithm 9 easily generalizes to high dimension:

Crucially, we observe that the proof of Theorem 2.2.4 nowhere requires that $[\![M]\!]$ be a set of M consecutive integers, but only that it is a fundamental domain of the lattice used in the DFT. Consequently, Theorem 2.2.4 also applies in this high dimensional setting, with appropriate notation. Note that the size of any fundamental

Algorithm 14 Testing the Fourier Transform Effective Support in high dimension

Require: parameters, a $k \times k$ matrix M, $b, \varepsilon \in (0,1]$; a fundamental domain A of $L(M)$; sample access to distribution $\mathbf{q}$ over A

Set $m \leftarrow \left\lceil C(\frac{\sqrt{b}}{\varepsilon^2} + \sqrt{\det(M)}) \right\rceil$ ▷ $C > 0$ is an absolute constant; $C = 2000$ works.
Draw $m' \leftarrow \mathrm{Poisson}(m)$; if $m' > 2m$, **return** reject
Draw m' samples from $\mathbf{q}$, and let $\mathbf{q}'$ be the corresponding empirical distribution over $[\![M]\!]$
Compute $\|\mathbf{q}'\|_2^2$, $\widehat{\mathbf{q}'}(\xi)$ for every $\xi \in S$, and $\left\|\widehat{\mathbf{q}'}\mathbb{1}_S\right\|_2^2$ ▷ Takes time $O(m\,|S|)$
if $m'^2\|\mathbf{q}'\|_2^2 - m' > \frac{3}{2}bm^2$ **then return** reject
else if $\|\mathbf{q}'\|_2^2 - \left\|\widehat{\mathbf{q}'}\mathbb{1}_S\right\|_2^2 \geq 3\varepsilon^2 + \frac{1}{m'}$ **then return** reject
else
    return $(\widehat{\mathbf{q}'}(\xi))_{\xi \in S}$
end if

domain is $\det(M)$ which appears in place of M in the sample complexity.

Algorithm 15 Algorithm `Test-PMD`

Require: sample access to a distribution $\mathbf{p} \in \Delta(\mathbb{N}^k)$, parameters $n, k \geq 1$ and $\varepsilon \in (0,1]$

▷ Let C, C', C'' be sufficiently large universal constants
Draw $m_0 = O(k^4)$ samples from X, and let $\widehat{\mu}$ be the sample mean and $\widehat{\Sigma}$ the sample covariance matrix.
Compute an approximate spectral decomposition of $\widehat{\Sigma}$, i.e., an orthonormal eigenbasis v_i with corresponding eigenvalues λ_i, $i \in [k]$.
Set $M \in \mathbb{Z}^{k \times k}$ to be the matrix whose i^{th} column is the closest integer point to the vector $C\left(\sqrt{k\log(k/\varepsilon)\lambda_i + k^2\log^2(k/\varepsilon)}\right) v_i$.
Set $I \leftarrow \mathbb{Z}^k \cap (\widehat{\mu} + M \cdot (-1/2, 1/2]^k)$
Draw $O(1/\varepsilon)$ samples from $\mathbf{p}$, and **return** reject if any falls outside of I
Let $S \subseteq (\mathbb{R}/\mathbb{Z})^k$ to be the set of points $\xi = (\xi_1, \ldots, \xi_k)$ of the form $\xi = (M^T)^{-1} \cdot v + \mathbb{Z}^k$, for some $v \in \mathbb{Z}^k$ with $\|v\|_2 \leq C^2k^2\log(k/\varepsilon)$.
Define $\mathbf{p} \bmod M$ to be the distribution obtained by sampling X from $\mathbf{p}$ and if it lies outside in I, returning X, else returning $X + Mb$ for the uniwue $b \in \mathbb{Z}^k$ such that $X + Mb \in I$.
Simulating sample access to $\mathbf{p}' \stackrel{\text{def}}{=} \mathbf{p} \bmod M$, call Algorithm 14 on $\mathbf{p}'$ with parameters M, $\frac{\varepsilon}{5\sqrt{\det(M)}}$, $b = \frac{|S|+1}{\det(M)}$, and S. If it outputs reject, then **return** reject; otherwise, let $\widehat{\mathbf{h}} = (\widehat{\mathbf{h}}(\xi))_{\xi \in S}$ denote the collection of Fourier coefficients it outputs, and $\mathbf{h}$ their inverse Fourier transform (modulo M) onto I. ▷ Do not actually compute $\mathbf{h}$
Compute a proper $\varepsilon/6\sqrt{|S|}$-cover $\mathcal{C}$ of all PMDs using the algorithm from [87].
for each $\mathbf{q} \in \mathcal{C}$ **do**
    if the mean $\mu_{\mathbf{q}}$ and covariance matrix $\Sigma_{\mathbf{q}}$ satisfy $(\widehat{\mu} - \mu_{\mathbf{q}})^T(\Sigma + I)^{-1}(\widehat{\mu} - \mu_{\mathbf{q}}) \leq 1$ and $2(\Sigma_{\mathbf{q}} + I) \geq \widehat{\Sigma} + I \geq (\Sigma_{\mathbf{q}} + I)/2$. **then**
        Compute $\widehat{\mathbf{q}}(\xi)$ for $\xi \in S$.
        if $\sum_{\xi \in S} |\widehat{\mathbf{h}} - \widehat{\mathbf{q}}|^2 \leq \varepsilon^2/16$ **then return** accept
        end if
    end if
end for
return reject if we do not accept for any $\mathbf{q} \in \mathcal{C}$.

The proof of correctness of Algorithm 15 is very similar to that of Algorithm 12, except that we need results from the proof of correctness of the PMD Fourier learning algorithm of [87]; we will only sketch these ingredients here. That I is an effective support of a PMD whose mean and covariance matrix we have estimated to within approprate error with high probability follows from Lemmas 3.3–3.6 of [87], the

last of which gives that the probability mass outside of I is at most $\varepsilon/10$, smaller than that claimed for I in the (n,k)-SIIRV algorithm. Lemma 3.3 gives, if $\mathbf{p}$ is a PMD, that the mean and covariance satisfy $(\widehat{\mu}-\mu)^T(\Sigma+I)^{-1}(\widehat{\mu}-\mu)=O(1)$ and $2(\Sigma_{\mathbf{q}}+I)\geq\widehat{\Sigma}+I\geq(\Sigma_q+I)/2$. Again, with more samples, we can strengthen this to $(\widehat{\mu}-\mu)^T(\Sigma+I)^{-1}(\widehat{\mu}-\mu)=\frac{1}{2}$ and $(3/2)(\Sigma+I)\geq\widehat{\Sigma}+I\geq(\Sigma+I)/(3/2)$ with $O(k^4)$ samples.

The effective support of the Fourier transform of a PMD is given by the following proposition:

Proposition 2.2.21 (Proposition 2.4 of [87])**.** *Let S be as in the algorithm. With probability at least $99/100$, the Fourier coefficients of $\mathbf{p}$ outside S satisfy $\sum_{\xi\in(L^*/\mathbb{Z}^k)\setminus S}|\widehat{\mathbf{p}}(\xi)|<\varepsilon/10$.*

This holds not just for $\mathbf{p}$, but any (n,k)-PMD $\mathbf{q}$ whose mean $\mu_{\mathbf{q}}$ and covariance matrix $\Sigma_{\mathbf{q}}$ satisfy $(\widehat{\mu}-\mu_{\mathbf{q}})^T(\Sigma+I)^{-1}(\widehat{\mu}-\mu)=O(1)$ and $2(\Sigma_{\mathbf{q}}+I)\geq\widehat{\Sigma}+I\geq(\Sigma_{\mathbf{q}}+I)/2$.

We need to show that this L_1 bound is stronger than the L_2 bound we need. Since every individual $\xi\notin S$ has $|\widehat{\mathbf{p}}(\xi)|<\varepsilon/10$, we have

$$\sum_{\xi\in(L^*/\mathbb{Z}^k)\setminus S}|\widehat{\mathbf{p}}(\xi)|^2\leq\sum_{\xi\in(L^*/\mathbb{Z}^k)\setminus S}\varepsilon|\widehat{\mathbf{p}}(\xi)|/10\leq\varepsilon^2/100$$

and so S is an effective support of the DFT modulo M.

To show that the value of b is indeed a bound on $\|\mathbf{p}\|_2^2$, we can use (2.32), yielding that $\|\mathbf{p}\|_2^2\leq(|S|+1)/\det(M)=b$, where $\det(M)$ here is indeed the size of I.

The proof of correctness of the algorithm and the projection step is now very similar to the (n,k)-SIIRV case. We need to get bounds on the sample and time complexity. We can bound the size of S using

$$\begin{aligned}|S|&\leq\left|\left\{\,v\in\mathbb{Z}^k\,:\,\|v\|_2\leq C^2k^2\log(k/\varepsilon)\,\right\}\right|\leq\left|\left\{\,v\in\mathbb{Z}^k\,:\,\|v\|_\infty\leq C^2k^2\log(k/\varepsilon)\,\right\}\right|\\&=\left(1+2\lfloor C^2k^2\log(k/\varepsilon)\rfloor\right)^k=O(k^2\log(k/\varepsilon))^k\end{aligned}$$

We can bound $\det(M)$ in terms of the L_2 norms of its columns using Hadamard's inequality

$$\det(M)\leq\prod_{i=1}^k\|M_i\|_2\leq\prod_{i=1}^k\left(C\left(\sqrt{k\log(k/\varepsilon)\lambda_i+k^2\log^2(k/\varepsilon)}\right)+\sqrt{k}\right)$$

recalling that λ_i are the eigenvalues of $\widehat{\Sigma}$ which satisfies $2(\Sigma_{\mathbf{q}}+I)\geq\widehat{\Sigma}+I$. We need a bound on $\|\Sigma\|_2$. Each individual summand k-CRV (categorical random variable) is supported on unit vectors, the distance between any two of which is $\sqrt{2}$. Therefore we have that $\|\Sigma\|_2\leq 2n$. Then $\lambda_i\leq 4n+1$ for every $1\leq i\leq k$; moreover, since the k coordinates must sum to n, $\widehat{\Sigma}$ has rank at most $k-1$ and so at least one of the λ_i's is zero. Combining these observations, we obtain

$$\det(M)\leq\sqrt{k^2\log^2\frac{k}{\varepsilon}}\cdot\left(C^2k(4n+2)\log\frac{k}{\varepsilon}+k^2\log^2\frac{k}{\varepsilon}\right)^{\frac{k-1}{2}}=k\log\frac{k}{\varepsilon}\cdot O\left(nk^2\log\frac{k}{\varepsilon}\right)^{\frac{k-1}{2}}.$$

With high constant probability, the number of samples we need is then

$$O\left(\frac{\sqrt{|S|\det M}}{\varepsilon^2}+\frac{|S|}{\varepsilon^2}+q_I(\varepsilon)\right)=\frac{1}{\varepsilon^2}\sqrt{k\log\frac{k}{\varepsilon}}\cdot O\left(nk^2\log\frac{k}{\varepsilon}\right)^{\frac{k-1}{4}}+\frac{O(k^2\log(k/\varepsilon))^k}{\varepsilon^2}+O(k^4)$$
$$=O(n^{(k-1)/4}k^{2k}\log(k/\varepsilon)^k/\varepsilon^2)$$

The time complexity of the algorithm is dominated by the projection step. By Proposition 4.9 and Corollary 4.12 of [87], we can produce a proper ε-cover of $\mathcal{PMD}_{n,k}$ of size $n^{O(k^3)}\cdot(1/\varepsilon)^{O(k^3\frac{\log(k/\varepsilon)}{\log\log(k/\varepsilon)})^{k-1}}$ in time also $n^{O(k^3)}\cdot(1/\varepsilon)^{O(k^3\frac{\log(k/\varepsilon)}{\log\log(k/\varepsilon)})^{k-1}}$. Note that producing an $(\varepsilon/6\sqrt{|S|})$-cover, as $=\varepsilon/O(k^2\log(k/\varepsilon))^{k/2}$, takes time $n^{O(k^3)}\cdot(1/\varepsilon)^{O(k^3\frac{\log(k/\varepsilon)}{\log\log(k/\varepsilon)})^{k-1}}$ (which is also the size of the resulting cover). Hence the running time of the algorithm is at most $n^{O(k^3)}\cdot(1/\varepsilon)^{O(k^3\frac{\log(k/\varepsilon)}{\log\log(k/\varepsilon)})^{k-1}}$.

Alternatively, [72] gives an ε-cover of size $n^{O(k)}\cdot\min 2^{\mathrm{poly}(k/\varepsilon)},2^{O(k^{5k}\log(1/\varepsilon)^{k+2}}$ that can also be constructed in polynomial time. By using this result, one needs to take time $n|S|\,\mathrm{poly}(\log(1/\varepsilon))$ to compute the Fourier coefficients. Applying this to get an $\varepsilon/O(k^2\log(k/\varepsilon))^{k/2}$-cover means that unfortunately we are always doubly exponential in k. In this case, the running time of the algorithm is $n^{O(k)}\cdot 2^{O(k^{5k}\log(1/\varepsilon)^{k+2}}$.

2.2.7 The Discrete Log-Concavity Tester

Theorem 2.2.22 (Testing Log-Concavity)**.** *Given parameters $n\in\mathbb{N}$, $\varepsilon\in(0,1]$, and sample access to a distribution $\mathbf{p}$ over $\mathbb{Z}$, there exists an algorithm which outputs either* accept *or* reject*, and satisfies the following.*

1. *if $\mathbf{p}\in\mathcal{LCV}_n$, then it outputs* accept *with probability at least $3/5$;*
2. *if $\mathrm{d_{TV}}(\mathbf{p},\mathcal{LCV}_n)>\varepsilon$, then it outputs* reject *with probability at least $3/5$.*

where $\mathcal{LCV}_n$ denotes the class of (discrete) log-concave distributions over $[\![n]\!]$. Moreover, the algorithm takes $O(\sqrt{n}/\varepsilon^2)+\tilde{O}\big((\log(n/\varepsilon)/\varepsilon)^{5/2}\big)$ samples from $\mathbf{p}$; and runs in time $O(\sqrt{n}\cdot\mathrm{poly}(1/\varepsilon))$.

We will sketch the proof and algorithm here. We first remark that the Maximum Likelihood Estimator (MLE) for log-concave distributions can be formulated as a convex program [90], which can be solved in sample polynomial time. One advantage of the MLE for log-concave distributions is that it properly learns log-concave distributions (over support size M) to within Hellinger distance ε using $\tilde{O}\big((\log M)/\varepsilon^{5/2}\big)$ samples[7]. Note that the squared Hellinger distance satisfies:

$$\mathrm{d_H}(\mathbf{p},\mathbf{q})^2=\sum_x(\sqrt{(\mathbf{p}(x)}-\sqrt{\mathbf{q}(x)})^2=\sum_x\frac{(\mathbf{p}(x)-\mathbf{q}(x))^2}{(\sqrt{\mathbf{p}}+\sqrt{\mathbf{q}})^2}\geq\frac{\|\mathbf{p}-\mathbf{q}\|_2}{2\max\{\mathbf{p}(x),\mathbf{q}(x)\}}\,.$$

Further, it is known that a log-concave distribution with variance σ^2 is effectively supported in an interval of

[7] We note that a similar, slightly stronger result is already known for *continuous* log-concave distributions, which can be learned to Hellinger distance ε from only $O(\varepsilon^{-5/2})$ samples [122]. The proof of this result, however, does not seem to generalize to discrete log-concave distributions, which is our focus here; thus, we establish in Section 2.2.9 the learning result we require, namely an upper bound on the sample complexity of the MLE estimator for learning the class of log-concave distributions over $[\![M]\!]$ in Hellinger distance (Theorem 2.2.32).

length $M = O(\log(1/\varepsilon)\sigma)$ centered at the mean, and that its maximum probability is $O(1/\sigma)$ (See Fact 2.2.27). Thus, by learning a log-concave distribution properly to within $\varepsilon/\log(1/\varepsilon)$ Hellinger distance, one also learns it to within $\frac{\varepsilon}{\sqrt{M}}$ L_2-distance.

A log-concave distribution $\mathbf{p}$ has L_2 norm bounded by $\|\mathbf{p}\|_2^2 \le \max_x \mathbf{p}(x) \le O(1/\sigma)$. It is easy to show using standard concentration bounds that $\mathbf{p} \bmod M$ also has L_2 norm $O(1/\sqrt{\sigma})$. We will prove in Proposition 2.2.23 that its DFT modulo M is effectively supported on a known set S of size $|S| = O(\log(1/\varepsilon)^2/\varepsilon^2)$.

Thus our algorithm will work as follows: First we estimate the mean and variance under the assumption of log-concavity. We construct an interval I of length $M = O(\log(1/\varepsilon)\sigma)$ which would be containing the effective support if we were log-concave; and reject if it is not the case, i.e., too much probability mass falls outside I. Then we properly learn $\mathbf{p}$ to within $\varepsilon/\log(1/\varepsilon)$ Hellinger distance using the MLE of $\tilde{O}\big((\log M)/\varepsilon^{5/2}\big)$ samples,[8] giving a hypothesis $\mathbf{h}$. At this point, we reject if our estimates for the mean and variance are far from that of $\mathbf{h}$. Then we run an L_2 identity tester between $\mathbf{p}$ and $\mathbf{h}$, i.e., test whether the empirical distribution $\mathbf{q}$ of $O(M/\sigma\varepsilon^2)$ samples is large. To do this efficiently, we compute $\|\mathbf{q}\|_2^2 - \|\widehat{\mathbf{q}}\mathbb{1}_S\|_2^2/M + \|\widehat{\mathbf{q}}\mathbb{1}_S - \widehat{\mathbf{h}}\mathbb{1}_S\|_2^2/M$ (since we know $\widehat{\mathbf{h}}$ is supported on S).

To do this in time $O(\sqrt{n}\cdot\mathrm{poly}(1/\varepsilon))$, we need to compute the Fourier coefficients efficiently. The MLE for log-concave distributions is a piecewise exponential distribution with a number of pieces at most the number of samples [90], which is $\tilde{O}\big((\log M)/\varepsilon^{5/2}\big)$ in this case. Using the expression for the integral of an exponential function gives a simple closed-form expression for $\mathbf{h}(\xi)$ that we can compute in time $\tilde{O}\big((\log M)/\varepsilon^{5/2}\big)$.

Proposition 2.2.23. *Let $\mathbf{p}$ be a discrete log-concave distribution with variance σ^2 and $M = O(\log(1/\varepsilon)\sigma)$ be the size of its effective support. Then its Discrete Fourier transform is effectively supported on a known set S of size $|S| = O(\log(1/\varepsilon)^2/\varepsilon^2)$.*

Proof. First we show that for any unimodal distribution, we can relate the maximum probability to the size of the effective support.

Lemma 2.2.24. *Let $\mathbf{p}$ be a unimodal distribution supported on $\mathbb{Z}$ such that the probability of the mode is $\mathbf{p}_{\max}$. Then the DFT modulo M of $\mathbf{p}$ at $\xi \in [-M/2, M/2)$ has $\widehat{\mathbf{p}}(\xi) = O(\mathbf{p}_{\max}M/|\xi|)$.*

Proof. Let m be the mode of $\mathbf{p}$. Then we have

$$\widehat{\mathbf{p}}(\xi) = \sum_{j=-\infty}^{m-1} \mathbf{p}(j)\exp\left(-2\pi i\frac{\xi j}{M}\right) + \sum_{j=m}^{\infty} \mathbf{p}(j)\exp\left(-2\pi i\frac{\xi j}{M}\right).$$

We will apply summation by parts to these two series. Let $g(x) = \sum_{j=m+1}^{x}\exp(-2\pi i\xi j/M)$ and $g(m) = 0$. By a standard result on geometric series, we have $g(x) = -\frac{\exp(-2\pi i\xi(x+1)/M) - \exp(-2\pi i\xi(m+1)/M)}{1-\exp(-2\pi i\xi/M)}$.

[8] Note that we here invoke the MLE estimator not on the full domain, but on the effective support, which contains at least $1 - O(\varepsilon^2)$ probability mass. This conditioning overall does not affect the sample complexity nor the distances, as it can only cause $O(\varepsilon^2)$ error in total variation (and thus $O(\varepsilon)$ in Hellinger distance).

Claim 2.2.25. *$|g(x)| = O(M/\xi)$ for all integers $x \geq m$.*

Proof. The modulus of the numerator $|\exp(-2\pi i\xi(x+1)/M) - \exp(-2\pi i\xi(m+1)/M)|$ is at most 2. We thus only need to find a lower bound for $|1 - \exp(-2\pi i\xi/M|$.

$$|1 - \exp(-2\pi i\xi/M)|^2 = (1 - \cos(2\pi\xi/M))^2 + \sin(2\pi\xi/M)^2 = 2 - 2\cos(2\pi\xi/M) = \Omega((\xi/M)^2) \ ,$$

and so $|g(x)| \leq 2/\sqrt{\Omega((\xi/M)^2)} = O(M/|\xi|)$. □

Now consider the following, for any $n > m$:

$$\sum_{j=m+1}^{n} \mathbf{p}(j)(g(j) - g(j-1)) + \sum_{j=m+1}^{n} g(j)(\mathbf{p}(j+1) - \mathbf{p}(j)) = \mathbf{p}(n+1)g(n) - \mathbf{p}(m+1)g(m) \ .$$

Now $g(m) = 0$ and $\mathbf{p}(n+1) \to 0$ as $n \to \infty$ while $g(n+1)$ is bounded for all n. Hence, the RHS tends to 0 as $n \to \infty$ and we have:

$$\begin{aligned} |\sum_{j=m+1}^{\infty} \mathbf{p}(j)\exp(-2\pi i\xi j/M)| &= |\sum_{j=m+1}^{\infty} \mathbf{p}(j)(g(j) - g(j-1))| = |\sum_{j=m+1}^{\infty} g(j)(\mathbf{p}(j+1) - \mathbf{p}(j))| \\ &\leq O(M/\xi) \cdot \sum_{j=m+1}^{\infty} (\mathbf{p}(j) - \mathbf{p}(j+1)) = O(\mathbf{p}_{\max} M/\xi) \ . \end{aligned}$$

Similarly, we can show that $\sum_{j=-\infty}^{m-1} \mathbf{p}(j)\exp(-2\pi i\xi j/M) = O(\mathbf{p}_{\max} M/\xi)$ since $\mathbf{p}$ is monotone there as well. □

Then we can get a bound on the size of the effective support:

Lemma 2.2.26. *Let $\mathbf{p}$ be a unimodal distribution supported on $\mathbb{Z}$ such that the probability of the mode is $\mathbf{p}_{\max}$ and let $\varepsilon \leq 1/M$. Then the DFT modulo M of $\mathbf{p}$ has $\sum_{|\xi|>\ell} |\widehat{\mathbf{p}}|^2 \leq \varepsilon^2/100$, where $\ell = \Theta\big(\mathbf{p}_{\max}^2 M^2/\varepsilon^2\big)$.*

Proof.

$$\sum_{|\xi|>\ell} |\widehat{P}|^2 \leq 2 \sum_{\xi=\ell+1}^{M/2} O(\mathbf{p}_{\max} M/\xi)^2 \leq O(\mathbf{p}_{\max} M)^2 \sum_{\xi=\ell+1}^{\infty} 1/\xi^2 \leq O(\mathbf{p}_{\max}^2 M^2/\ell) \leq \frac{\varepsilon^2}{100} \ .$$

□

For log-concave distributions, we can relate $\mathbf{p}_{\max}$ and M as follows,

Fact 2.2.27. *Let $\mathbf{p}$ be a discrete log-concave distribution with mean μ and variance σ^2. Then*

- *$\mathbf{p}$ is unimodal;*
- *its probability mass function satisfies $\mathbf{p}(x) = \exp(-O((x-\mu)/\sigma))/\sigma$; and*
- *$\Pr[|X - \mu| \geq \Omega(\sigma \log(1/\varepsilon))] \leq \varepsilon$.*

Since $\mathbf{p}_{\max} = O(1/\sigma)$, we can take $M = O(\sigma \log(1/\varepsilon))) = O(\log(1/\varepsilon)/\mathbf{p}_{\max})$. Substituting this into Lemma 2.2.26 completes the proof of the proposition. □

2.2.8 Lower Bound for PMD Testing

In this section, we obtain a lower bound to complement our upper bound for testing Poisson Multinomial Distributions. Namely, we prove the following:

Theorem 2.2.28. *There exists an absolute constant $c \in (0,1)$ such that the following holds. For any $k \leq n^c$, any testing algorithm for the class of $\mathcal{PMD}_{n,k}$ must have sample complexity $\Omega\Big(\big(\frac{4\pi}{k}\big)^{k/4} \frac{n^{(k-1)/4}}{\varepsilon^2}\Big)$.*

The proof will rely on the lower bound framework of [51], reducing testing $\mathcal{PMD}_{n,k}$ to testing identity to some suitable hard distribution $\mathbf{p}^* \in \mathcal{PMD}_{n,k}$. To do so, we need to (a) choose a convenient $\mathbf{p}^* \in \mathcal{PMD}_{n,k}$; (b) prove that testing identity to $\mathbf{p}^*$ requires that many samples (we shall do so by invoking the [169] instance-by-instance lower bound method); (c) provide an agnostic learning algorithm for $\mathcal{PMD}_{n,k}$ with small enough sample complexity, for the reduction to go through. Invoking [51, Theorem 18] with these ingredients will then conclude the argument.

Proof of Theorem 2.2.28. In what follows, we choose our "hard instance" $\mathbf{p}^* \in \mathcal{PMD}_{n,k}$ to be the PMD obtained by summing n i.i.d. random variables, all uniformly distributed on $\{e_1, \dots, e_k\}$. This takes care of point (a) above.

To show (b), we will rely on a result of Valiant and Valiant, which showed in [169] that testing identity to any discrete distribution $\mathbf{p}$ required $\Omega\Big(\|\mathbf{p}_{-\varepsilon}^{-\max}\|_{2/3}/\varepsilon^2\Big)$ samples, where $\mathbf{p}_{-\varepsilon}^{-\max}$ is the vector obtained by zeroing out the largest entry of $\mathbf{p}$, as well as a cumulative ε mass of the smallest entries. Since $\|\mathbf{p}_{-\varepsilon}^{-\max}\|_{2/3}$ is rather cumbersome to analyze, we shall instead use a slightly looser bound, considering $\|\mathbf{p}\|_2$ as a proxy.

Fact 2.2.29. *For any discrete distribution $\mathbf{p}$, we have $\|\mathbf{p}\|_{2/3} \geq \frac{1}{\|\mathbf{p}\|_2}$. More generally, for any vector x we have $\|x\|_{2/3} \geq \frac{\|x\|_1^2}{\|x\|_2}$.*

Proof. It is sufficient to prove the second statement, which implies the first. This is in turn a straightforward application of Hölder's inequality, with parameters $(4, \frac{4}{3})$: $\|x\|_1 = \sum_i |x|_i^{1/2} |x|_i^{1/2} \leq \Big(\sum_i |x|_i^2\Big)^{1/4} \Big(|x|_i^{2/3}\Big)^{3/4}$. Squaring both sides yields the claim. □

Fact 2.2.30. *For our distribution $\mathbf{p}^*$, we have $\|\mathbf{p}^*\|_2 = \Theta\Big(\frac{k^{k/4}}{(4\pi n)^{(k-1)/4}}\Big)$.*

Proof. It is not hard to see that, from any $\mathbf{n} = (n_1, \dots, n_k) \in \mathbb{N}^k$ such that $\sum_{i=1}^k n_i = n$, $\mathbf{p}^*(\mathbf{n}) = \frac{1}{k^n}\binom{n}{n_1,\dots,n_k}$ (where $\binom{n}{n_1,\dots,n_k}$ denotes the multinomial coefficient). From there, we have

$$\|\mathbf{p}^*\|_2^2 = \frac{1}{k^{2n}} \sum_{n_1+\cdots+n_k=n} \binom{n}{n_1, \dots, n_k}^2 \underset{n\to\infty}{\sim} \frac{1}{k^{2n}} \cdot k^{2n} \frac{k^{k/2}}{(4\pi n)^{(k-1)/2}}$$

where the equivalent is due to Richmond and Shallit [148]. □

However, from Fact 2.2.29 we want to get a hold on $\|\mathbf{p}^{*-\max}_{-\varepsilon}\|_2$, not $\|\mathbf{p}^*\|_2$ (since $\|\mathbf{p}^{*-\max}_{-\varepsilon}\|_1^2 \geq 1-\Omega(\varepsilon)$, we then will have our lower bound on $\|\mathbf{p}^{*-\max}_{-\varepsilon}\|_{2/3}$). Fortunately, the two are related: namely, $\|\mathbf{p}^{*-\max}_{-\varepsilon}\|_2 \leq \|\mathbf{p}^*\|_2$, so $\frac{1}{\|\mathbf{p}^{*-\max}_{-\varepsilon}\|_2} \geq \frac{1}{\|\mathbf{p}^*\|_2}$ which is the direction we need.

Combining the three facts above establishes (b), providing a lower bound of $q_{\mathrm{hard}}(n,k,\varepsilon) = \Omega\Big(\frac{(4\pi n)^{(k-1)/4}}{k^{k/4}\varepsilon^2}\Big)$ for testing identity to $\mathbf{p}^*$. It only remains to establish (c):

Lemma 2.2.31. *There exists a (not necessarily efficient) agnostic learner for $\mathcal{PMD}_{n,k}$, with sample complexity $q_{\mathrm{agn}}(n,k,\varepsilon) = \frac{1}{\varepsilon^2}\left(O(k^2\log n) + O\Big(\frac{k\log(k/\varepsilon)}{\log\log(k/\varepsilon)}\Big)^k\right)$.*

Proof. This is implied by a result of [87], which establishes the existence of a (proper) ε-cover $\mathcal{M}_{n,k,\varepsilon}$ of $\mathcal{PMD}_{n,k}$ such that $|\mathcal{M}_{n,k,\varepsilon}| \leq n^{O(k^2)} \cdot (1/\varepsilon)^{O\left(\frac{k\log(k/\varepsilon)}{\log\log(k/\varepsilon)}\right)^{k-1}}$. By standard arguments, this yields information-theoretically an agnostic learner with sample complexity $O\Big(\frac{\log|\mathcal{M}_{n,k,\varepsilon}|}{\varepsilon^2}\Big)$. □

Having (a), (b), and (c), an application of [51, Theorem 18] yields that, as long as $q_{\mathrm{agn}}(n,k,\varepsilon) = o(q_{\mathrm{hard}}(n,k,\varepsilon))$ then testing membership in $\mathcal{PMD}_{n,k}$ requires $\Omega(q_{\mathrm{hard}}(n,k,\varepsilon))$ samples as well. This in particular holds for $k = o(n^c)$ (where e.g. $c < 1/9$) and $\varepsilon = 1/2^{O(n)}$.

□

2.2.9 Learning Discrete Log-Concave Distributions in Hellinger Distance

Recall that the Hellinger distance between two probability distributions over a domain Ω is defined as

$$\mathrm{d_H}(p,q) \stackrel{\mathrm{def}}{=} \frac{1}{\sqrt{2}}\|\sqrt{p} - \sqrt{q}\|_2$$

where the 2-norm is to be interpreted as either the ℓ_2 distance or L^2 distance between the pmf or pdf's of p, q, depending on whether Ω is $\mathbb{Z}$ or $\mathbb{R}$. In particular, one can extend this metric to the set of *pseudo*-distributions over Ω, relaxing the requirement that the measures sum to one. We let $\mathcal{F}_\Omega$ denote the set of pseudo-distributions over Ω. The *bracketing entropy* of a family of functions $\mathcal{G} \subseteq \mathbb{R}^\Omega$ with respect to the Hellinger distance (for parameter ε) if then the minimum cardinality of a collection $\mathcal{C}$ of pairs $(g_L, g_U) \in \mathcal{F}_\Omega^2$ such that every $f \in \mathcal{G}$ is "bracketed" between the elements of some pair in $\mathcal{C}$:

$$\mathcal{N}_{[]}(\varepsilon, \mathcal{G}, \mathrm{d_H}) \stackrel{\mathrm{def}}{=} \min\big\{\, N \in \mathbb{N} :\ \exists \mathcal{C} \subseteq \mathcal{F}_\Omega^2,\ |\mathcal{C}| = N,\ \forall f \in \mathcal{G}, \exists (g_L, g_U) \in \mathcal{C} \text{ s.t. } g_L \leq f \leq g_U \text{ and } \mathrm{d_H}(g_L, g_H) \leq \varepsilon \,\big\}$$

Theorem 2.2.32. *Let $\hat{p}_m$ denote the maximum likelihood estimator (MLE) for discrete log-concave distributions on a sample of size m. Then, the minimax supremum risk satisfies*

$$\sup_{p \in \mathcal{LCV}_n} \mathbb{E}_p[\mathrm{d_H}(\hat{p}_m, p)^2] = O\left(\frac{\log^{4/5}(mn)}{m^{4/5}}\right).$$

Note that it is known that for *continuous* log-concave distributions over $\mathbb{R}$, the rate of the MLE is $O(m^{-4/5})$ [122]; this result, however, does not generalize to discrete log-concavity, as it crucially relies on

a scaling argument which does not work in the discrete case. On the other hand, one can derive a rate of convergence to learn discrete log-concave distributions in *total variation distance* (using another estimator than the MLE), getting again $O(m^{-4/5})$ in that case [83]. However, due to the loose upper bound relating total variation and Hellinger distance, this latter result only implies an $O(m^{-2/5})$ convergence rate in Hellinger distance, which is quadratically worse than what we would hope for.

Thus, the result above, while involving a logarithmic dependence on the support size, has the advantage of getting the "right" rate of convergence. (While this additional dependence does not matter for our purposes, we believe a modification of our techniques would allow one to get rid of it, obtaining a rate of $\tilde{O}\big(m^{-4/5}\big)$ instead.) We however conjecture that the tight rate of convergence should be $O(m^{-4/5})$, as in the continuous case (i.e., without the dependence on the domain size n nor the extra logarithmic factors in m).

In order to prove Theorem 2.2.32, we obtain along the way several interesting results on discrete (and continuous) log-concave distributions, namely a bound on their bracketing entropy (Theorem 2.2.33) and an approximation result (Theorem 2.2.34), which we believe are of independent interest.

In what follows, Ω will denote either $\mathbb{R}$ or $\mathbb{Z}$; we let $\mathcal{LCV}_\Omega$ denote the set of log-concave distributions over Ω, and $\mathcal{LCV}_n \subseteq \mathcal{LCV}_\mathbb{Z}$ be the subset of log-concave distributions supported on $[\![n]\!]$.

Theorem 2.2.33. *For every $\varepsilon \in (0,1)$,*

$$\mathcal{N}_{[]}(\varepsilon, \mathcal{LCV}_n, \mathrm{d_H}) \leq \left(\frac{n}{\varepsilon}\right)^{O(1/\sqrt{\varepsilon})}$$

A crucial element in to establish Theorem 2.2.33 will be the following theorem, which shows that log-concave distributions are well-approximated (in Hellinger distance) by piecewise-constant pseudo-distributions with few pieces:

Theorem 2.2.34. *Let Ω be either $\mathbb{R}$ or $\mathbb{Z}$. For every $p \in \mathcal{LCV}_\Omega$ and $\varepsilon \in (0,1)$, there exists a pseudo-distribution g such that (i) g is piecewise-linear with $O(1/\sqrt{\varepsilon})$ pieces; (ii) g is supported on an interval $[a,b]$ with $p(\Omega \setminus [a,b]) = O(\varepsilon^2)$; and (iii) $\mathrm{d_H}(p,g) \leq \varepsilon$. (Moreover, one can choose to enforce $g \leq p$, or $p \leq g$, on $[a,b]$).*

The proof of Theorem 2.2.34 will be very similar to that of [83, Theorem 12]; specifically, we will use the following (reformulation of a) lemma due to Diakonikolas, Kane, and Stewart:

Lemma 2.2.35 ([83, Lemma 14], rephrased)**.** *Let Ω be either $\mathbb{R}$ or $\mathbb{Z}$. Let f be a log-concave function defined on an interval $I \subseteq \Omega$, and suppose that $f(I) \subseteq [a, 2a]$ for some constant $a > 0$. Furthermore, suppose that the logarithmic derivative of f (or, if $\Omega = \mathbb{Z}$, the log-finite difference of f) varies by at most $1/|I|$ on I; then, for any $\varepsilon \in (0,1)$ there exists two piecewise linear functions $g^\ell, g^u \colon I \mapsto \mathbb{R}$ with $O(1/\sqrt{\varepsilon})$ pieces such that*

$$\big|f(x) - g^j(x)\big| = O(\varepsilon) f(x), \qquad j \in \{\ell, u\} \tag{2.33}$$

for all $x \in I$, and with $g^\ell \leq f \leq g^u$.

Proof. Observe that it suffices to establish Eq. (2.33) for a piecewise linear function $g\colon I \mapsto \mathbb{R}$ with $O(1/\sqrt{\varepsilon})$ pieces; indeed, then in order to obtain g^ℓ, g^u from g, it will be sufficient to scale it by respectively $(1+\alpha\varepsilon)^{-1}$ and $(1+\alpha\varepsilon)$ (for a suitably big absolute constant $\alpha > 0$), thus ensuring both Eq. (2.33) and $g^\ell \le f \le g^u$. We therefore focus hereafter on obtaining such a pseudo-distribution g.

For ease of notation, we write h for the logarithmic derivative (or log-finite difference) of f (e.g., in the continuous case, $h = (\ln f)'$). By rescaling f, we may assume without loss of generality that $a = 1$. Note that h is then bounded on I, i.e. $|h| \le c/|I|$ for some absolute constant $c > 0$. We now partition I into subintervals $J_1, J_2, \ldots, J_\ell$ so that (i) each J_i has length at most $\varepsilon^{1/2}\,|I|$, and (ii) h varies by at most $\varepsilon^{1/2}/\,|I|$ on each J_i. This can be achieved with $\ell = O(1/\sqrt{\varepsilon})$ by placing an interval boundary every $\varepsilon^{1/2}\,|I|$ distance as well as every time h passes a multiple of $\varepsilon^{1/2}/\,|I|$.

We now claim that on each interval J_i there exists a linear function g_i so that $|g_i(x) - f(x)| = O(\varepsilon)f(x)$ for all $x \in J_i$. Letting g be g_i on J_i will complete the proof. Fix any i, and write $J_i = [s_i, t_i]$. Letting $\alpha_0 \in h(J_i)$ be an arbitrary value in the range spanned by h on J_i, observe that for any $x \in J_i$ there exists $\alpha_x \in h(J_i)$ such that

$$f(x) = f(s_i)e^{\alpha_x(x-s_i)}$$

from which we have

$$\begin{aligned} f(x) &= f(s_i)e^{\alpha_0(x-s_i)+(\alpha_x-\alpha_0)(x-s_i)} = f(s_i)e^{\alpha_0(x-s_i)}e^{(\alpha_x-\alpha_0)(x-s_i)} \\ &= f(s_i)\left(1+\alpha_0(x-s_i)+O(\varepsilon)\right)\left(1+O((\alpha_x-\alpha_0)(x-s_i))\right) \\ &= f(s_i)\left(1+\alpha_0(x-s_i)+O(\varepsilon)\right)\left(1+O(\varepsilon)\right) \\ &= f(s_i)+\alpha_0 f(s_i)(x-s_i)+O(\varepsilon) \end{aligned}$$

recalling that $|\alpha_0|\,, |\alpha_x| = O(1/\,|I|)$, $|x - s_i| \le \varepsilon^{1/2}\,|I|$, and $|\alpha_x - \alpha_0| \le \varepsilon^{1/2}/\,|I|$, so that $|\alpha_0(x-s_i)| = O(\varepsilon^{1/2})$ and $|(\alpha_x-\alpha_0)(x-s_i)| = O(\varepsilon)$. This motivates defining the affine function g_i as

$$g_i(x) \stackrel{\text{def}}{=} f(s_i) + \alpha_0 f(s_i)(x-s_i), \qquad x \in J_i$$

from which

$$\begin{aligned} \left|\frac{f(x)-g_i(x)}{f(x)}\right| &= \left|1-\frac{f(s_i)+\alpha_0 f(s_i)(x-s_i)}{f(s_i)e^{\alpha_x(x-s_i)}}\right| = \left|1-\frac{1+\alpha_0(x-s_i)}{e^{\alpha_x(x-s_i)}}\right| \\ &= \left|1-\frac{1+\alpha_0(x-s_i)}{1+\alpha_x(x-s_i)+O(\varepsilon)}\right| = \left|1-\left(1+\alpha_0(x-s_i)\right)\left(1-\alpha_x(x-s_i)+O(\varepsilon)\right)\right| \\ &= \left|(\alpha_x-\alpha_0)(x-s_i)+O(\varepsilon)\right| = O(\varepsilon) \end{aligned}$$

as claimed. This concludes the proof. □

We will also rely on the following proposition, from the same paper:

Proposition 2.2.36 ([83, Proposition 15])**.** *Let f be a log-concave distribution on Ω (as before, either $\mathbb{R}$ or $\mathbb{Z}$). Then there exists a partition of Ω into disjoint intervals $I_1, I_2, \ldots$ and a constant $C > 0$ such that*

- *f satisfies the hypotheses of Lemma 2.2.35 on each I_i.*
- *For each m, there are most Cm values of i so that $f(I_i) > 2^{-m}$.*

(Moreover, f is monotone on each I_i.)

We are now ready to prove Theorem 2.2.34:

Proof of Theorem 2.2.34. Fix any $\varepsilon \in (0,1)$, and $p \in \mathcal{LCV}_\Omega$. We divide Ω into intervals as described in Proposition 2.2.36. Call these intervals $I_1, I_2, \ldots$ sorted so that $p(I_i)$ is decreasing in i. Therefore, we have that $p(I_m) \le 2^{-m/C}$.

For $1 \le m \le M \stackrel{\text{def}}{=} 2C\log(1/\varepsilon)$, let $\varepsilon_m \stackrel{\text{def}}{=} \varepsilon 2^{m/(3C)}$; we use Lemma 2.2.35 to approximate p in I_m by two piecewise linear functions g_m^ℓ, g_m^u so that (i) g_m^j has at most $O(1/\sqrt{\varepsilon_m})$ pieces and (ii) p and g_m^j are, on I_m, within a multiplicative $(1 \pm O(\varepsilon_m))$ factor with $g_m^\ell \le p \le g_m^u$. For $j \in \{\ell, u\}$, let g^j be the piecewise linear function that is g_m^j on I_m for $1 \le m \le M$, and 0 elsewhere. g^j is then piecewise linear on

$$\sum_{m=1}^{M} O(\varepsilon_m^{-1/2}) = \sum_{m=1}^{M} O\Big(\varepsilon^{-1/2} 2^{-\frac{m}{6C}}\Big) = O(\varepsilon^{-1/2})$$

intervals.

Let I be defined as the smallest interval such that $\bigcup_{m=1}^{M} I_m \subseteq I$. By definition, g is 0 outside of I, and moreover the total mass of p there is

$$\sum_{m=M+1}^{\infty} p(I_m) \le \sum_{m=M+1}^{\infty} \frac{1}{2^{m/C}} = O\Big(2^{-M/C}\Big) = O\big(\varepsilon^2\big)$$

By replacing g^j by $\max(g^j, 0)$, we may ensure that it is non-negative (while at most doubling the number of pieces without increasing the distance from p). This establishes the first two items of the theorem; we now turn to the third.

The Hellinger distance between p and g^j satisfies, letting $J \stackrel{\text{def}}{=} \bigcup_{m=1}^{M} I_m$,

$$\begin{aligned}
2\mathrm{d}_{\mathrm{H}}\big(p, g^j\big)^2 &= \|\sqrt{p} - \sqrt{g^j}\|_2^2 = \int_{\Omega} \left(\sqrt{p(x)} - \sqrt{g^j(x)}\right)^2 \mu(dx) \\
&= \int_{\Omega \setminus J} \left(\sqrt{p(x)} - \sqrt{g^j(x)}\right)^2 \mu(dx) + \int_{J} \left(\sqrt{p(x)} - \sqrt{g^j(x)}\right)^2 \mu(dx) \\
&= \int_{\Omega \setminus J} p(x)\mu(dx) + \sum_{m=1}^{M} \int_{I_m} p(x) \left(1 - \sqrt{1 \pm O(\varepsilon_m)}\right)^2 \mu(dx) \\
&\leq O(\varepsilon^2) + \sum_{m=1}^{M} \int_{I_m} p(x) \left(1 - \sqrt{1 \pm O(\varepsilon_m)}\right)^2 \mu(dx) \\
&= O(\varepsilon^2) + \sum_{m=1}^{M} \int_{I_m} p(x) O(\varepsilon_m^2) \mu(dx) = O(\varepsilon^2) + \sum_{m=1}^{M} O\big(\varepsilon_m^2 p(I_m)\big) \\
&= O(\varepsilon^2) + \sum_{m=1}^{M} O\Big(\varepsilon^2 2^{\frac{2m}{3C}} 2^{\frac{-m}{C}}\Big) = O(\varepsilon^2) + \sum_{m=1}^{M} O\Big(\varepsilon^2 2^{\frac{-m}{3C}}\Big) \\
&= O(\varepsilon^2) + O(\varepsilon^2) = O(\varepsilon^2)
\end{aligned}$$

establishing the third item. (By dividing ε by a sufficiently big absolute constant before applying the above, one gets (i), (ii), and (iii) with $\mathrm{d}_{\mathrm{H}}\big(p, g^j\big) \leq \varepsilon$ as desired.) For technical reasons (that we will need in the proof of Theorem 2.2.33), instead of defining $[a, b]$ to be our interval I, we choose $[a, b]$ to be I augmented with up to two of the remaining I_m's (those directly on the left and right of I, defining g_m^ℓ, g_m^u on these two additional pieces as before by Lemma 2.2.35). This does not change the fact that the piecewise linear function obtained on $[a, b]$ has $O(\varepsilon^{-1/2})$ pieces (we only added $o(\varepsilon^{-1/2})$ pieces), and $p(\Omega \setminus [a, b]) \leq p(\Omega \setminus I) = O(\varepsilon^2)$. Finally, it is easy to see that this only changes, as per the computation above, the Hellinger distance by $O(\varepsilon^2)$ as well. (The advantage of this technicality is that now, the two end intervals in the union constituting $[a, b]$ have each total probability mass $O(\varepsilon^2)$ under p, which will come in handy later.) It then only remains to choose g to be either g^ℓ or g^u, depending on whether one wants a lower- or upperbound on f (on $[a, b]$). □

We can finally prove Theorem 2.2.33:

Proof of Theorem 2.2.33. We can slightly strengthen the proof of Theorem 2.2.34 for the case of $\mathcal{LCV}_n$, by imposing some restriction on the form of the 'approximating distributions" g. Namely, for any $\varepsilon \in (0, 1)$, fix any $p \in \mathcal{LCV}_n$ and consider the construction of g^ℓ, g^u as in the proof of Theorem 2.2.34. Clearly, we can assume $[a, b] \subseteq [\![n]\!]$.

Now, we modify g^j as follows (for $j \in \{\ell, u\}$): for $1 \leq m \leq M$, consider the interval $I_m = [a_m, b_m]$, and the corresponding "piece" g_m^j of g on I_m. We let $\tilde{g}_m^j$ be the pseudo-distribution defined from g_m^j as follows: it is affine on I_m, with

$$\tilde{g}_m^u(a_m) \stackrel{\text{def}}{=} \left\lceil g^u(a_m) \frac{M\,|I_m|}{2\varepsilon^2} \right\rceil \frac{2\varepsilon^2}{M\,|I_m|}, \qquad \tilde{g}_m^u(a_m) \stackrel{\text{def}}{=} \left\lceil g^u(b_m) \frac{M\,|I_m|}{2\varepsilon^2} \right\rceil \frac{2\varepsilon^2}{M\,|I_m|}$$

and

$$\tilde{g}_m^\ell(a_m) \stackrel{\text{def}}{=} \left\lfloor g^\ell(a_m)\frac{M\,|I_m|}{2\varepsilon^2}\right\rfloor \frac{2\varepsilon^2}{M\,|I_m|}, \qquad \tilde{g}_m^\ell(a_m) \stackrel{\text{def}}{=} \left\lfloor g^\ell(b_m)\frac{M\,|I_m|}{2\varepsilon^2}\right\rfloor \frac{2\varepsilon^2}{M\,|I_m|}$$

i.e. g_m^j is g "rounded up" (resp. down) to the near multiple of $\frac{\varepsilon^2}{M|I_m|}$ on the endpoints. We then let $\tilde{g}^j$ be the correspond piecewise-affine pseudo-distribution defined by piecing together the $\tilde{g}_m^j$'s. Clearly, by construction $\tilde{g}^\ell$ and $\tilde{g}^u$ still satisfies (i) and (ii) of Theorem 2.2.34, and $\tilde{g}^\ell \le p \le \tilde{g}^u$. As for (iii), observe that at all $1 \le m \le M$ and $k \in I_m$ we have $\left|\tilde{g}^j(k) - g^j(k)\right| \le \frac{2\varepsilon^2}{M|I_m|}$, from which

$$\mathrm{d_H}\big(p, \tilde{g}^j\big) \le \mathrm{d_H}\big(p, g^j\big) + \mathrm{d_H}\big(g, \tilde{g}^j\big) \le \varepsilon + \sqrt{\mathrm{d_{TV}}(g^j, \tilde{g}^j)} \le \varepsilon + \sqrt{\frac{1}{2}\sum_{m=1}^{M} |I_m| \cdot \frac{2\varepsilon^2}{M\,|I_m|}} = 2\varepsilon$$

showing that we get (up to a constant factor loss in the distance) (iii) as well. Given this, we get that specifying $(\tilde{g}^\ell, \tilde{g}^u)$ can be done by the list of the $O(1/\sqrt{\varepsilon})$ endpoints along with the value of each $\tilde{g}^j$ for all of these endpoints. Now, given the two endpoints, one gets the size of the corresponding interval I_m (which is at most n), and the two values to specify are a multiple of $\varepsilon^2/(M\,|I_m|)$ in $[0,1]$. (If we were to stop here, we would get the existence of an ε-cover $\mathcal{C}'_\varepsilon$ of $\mathcal{LCV}_n$ in Hellinger distance of size $(n/\varepsilon)^{O(1/\sqrt{\varepsilon})}$.)

One last step: outside $[a,b]$ To get the bracketing bound we seek, we need to do one last modification to our pair $(\tilde{g}^\ell, \tilde{g}^u)$. Specifically, in the above we have one issue when approximating p: namely, that outside of their common support $\{a,\dots,b\}$, both $\tilde{g}^j$'s are 0. While this is fine for the lower bound $\tilde{g}^\ell$, this is not for $\tilde{g}^u$, as it needs to dominate p outside of $\{a,\dots,b\}$ as well, where p may have $O(\varepsilon^2)$ probability mass. Thus, we need to adapt the construction above, as follows (we treat the setting of $\tilde{g}^u$ on $\{b+1,\dots,n\}$, the definition on $[\![a]\!]$ is similar).

First, observe if $p(b+1) = 0$, we are done, as then by monotonicity we must have $(k) = 0$ for all $k \ge b+1$, and so setting $\tilde{g}^u = 0$ on $\{b+1,\dots,n\}$ suffices. Thus, we hereafter assume $p(b+1) > 0$; and, for $b+1 \le k \le n$, set

$$\tilde{g}^u(k) \stackrel{\text{def}}{=} \alpha e^{\beta(k-(b+1))}$$

where $\alpha \stackrel{\text{def}}{=} \left\lceil p(b+1)\frac{n}{2\varepsilon^2}\right\rceil \frac{2\varepsilon^2}{n}$ and $\beta \stackrel{\text{def}}{=} \left\lceil \frac{n}{\varepsilon}\ln\frac{p(b+2)}{p(b+1)}\right\rceil \frac{\varepsilon}{n}$ (so that $\beta \le 0$). Then $\tilde{g}^u(b+1) \ge p(b+1)$, and for $b+1 < k \le n$

$$\frac{\tilde{g}^u(k)}{\tilde{g}^u(k-1)} = e^\beta \ge \frac{p(b+2)}{p(b+1)} \ge \frac{p(k)}{p(k-1)}$$

(the last inequality due to the log-concavity of p). This implies $\tilde{g}^u \ge p$ on $\{b+1,\dots,n\}$ as desired; and, thanks to the rounding, there are only $O(n/\varepsilon^2)$ different possibilities for the tail of $\tilde{g}^u$. In view of bounding the Hellinger distance between p and $\tilde{g}^u$ added by this modification, which is upper bounded by the (square root) of the total variation distance this added, recall that $p(\{b+1,\dots,n\}) = O(\varepsilon^2)$ by construction, and that

$$\tilde{g}^u(\{b+1,\dots,n\}) = \sum_{k=b+1}^{n} \alpha e^{\beta(k-(b+1))} = \frac{\alpha}{1-e^\beta}.$$

Thus, the Hellinger distance incurred on $\{b+1,\dots,n\}$ is at most $\sqrt{O(\varepsilon^2)+\frac{\alpha}{1-e^\beta}}$; and to conclude, it only remains to show that $\frac{\alpha}{1-e^\beta}=O(\varepsilon^2)$.

To show this last point, let $I_m=[c,b]$ be the rightmost interval in the decomposition from Proposition 2.2.36. Recall that we are guaranteed that p is non-increasing on I_m; further, by inspection of the proof of [83, Proposition 15], we also have that I_m is *maximal*, in the sense that b is the rightmost point k such that $[c,k]$ satisfies the assumptions of Lemma 2.2.35. Using first the monotoncity, we have

$$p(b+1)\leq p(b)\leq\frac{p(I_m)}{b-c}\leq\frac{O(\varepsilon^2)}{b-c}$$

that last inequality by construction (from the technicality we enforced in the end of the proof of Theorem 2.2.34); and therefore $\alpha\leq\frac{O(\varepsilon^2)}{b-c}+\frac{\varepsilon^2}{n}=\frac{O(\varepsilon^2)}{b-c}$.
In order to obtain an upper bound on β, we rely on the maximality of I_m, leading to two cases to consider:

- The first is that $p(b+1)<\frac{p(c)}{2}$; in which case $p(b+2)\leq p(b+1)<\frac{p(c)}{2}$; which implies that

$$\frac{1}{2}>\frac{p(b+2)}{p(c)}=\frac{p(b+2)}{p(b+1)}\cdot\frac{p(b+1)}{p(b)}\cdots\frac{p(c+1)}{p(c)}\geq\left(\frac{p(b+2)}{p(b+1)}\right)^{b-c+2}$$

  the last inequality by log-concavity. In turn, we get

$$\beta\leq\ln\frac{p(b+2)}{p(b+1)}+\frac{\varepsilon}{n}\leq-\frac{\ln 2}{b-c+2}+\frac{\varepsilon}{n}.$$

- The second is that $\ln\frac{p(c+1)}{p(c)}-\ln\frac{p(b+1)}{p(b)}>\frac{1}{b-c+1}$. In this case,

$$\ln\frac{p(b+2)}{p(b+1)}\leq\ln\frac{p(b+1)}{p(b)}<\ln\frac{p(c+1)}{p(c)}-\frac{1}{b-c+1}\leq-\frac{1}{b-c+1}<-\frac{\ln 2}{b-c+2}$$

  (the last inequality as $b-c\geq 0$) and therefore $\beta\leq-\frac{\ln 2}{b-c+2}+\frac{\varepsilon}{n}$ as in the first case.

Combining these two bounds, we obtain

$$\frac{\alpha}{1-e^\beta}\leq\frac{O(\varepsilon^2)}{b-c}\cdot\frac{1}{1-e^{\frac{\varepsilon}{n}}e^{-\frac{\ln 2}{b-c+2}}}=O(\varepsilon^2)$$

the last inequality for $\varepsilon<\frac{\ln 2}{2}$ (using the fact that $1\leq b-c\leq n$). This concludes the proof: as discussed, we then have that our setting of $\bar{g}^u$ outside of $[a,b]$ only causes an addition Hellinger distance of $\sqrt{O(\varepsilon^2)+\frac{\alpha}{1-e^\beta}}=\sqrt{O(\varepsilon^2)}=O(\varepsilon)$.

□

We are, at last, ready to prove our main theorem:

Proof of Theorem 2.2.32. Recall the following theorem, due to Wong and Shen [178] (see also [98, Theorem 7.4], [122, Theorem 17]):

Theorem 2.2.37 ([178, Theorem 2]). *There exist positive constants $\tau_1, \tau_2, \tau_3, \tau_4 > 0$ such that, for all $\varepsilon \in (0,1)$, if*

$$\int_{\varepsilon^2/2^8}^{\sqrt{2}\varepsilon} \sqrt{\mathcal{N}_{[]}(u/\tau_1, \mathcal{G}, \mathrm{d_H})}\, du \le \tau_2 m^{1/2} \varepsilon^2 \tag{2.34}$$

and $\tilde{p}_n$ is an estimator that approximates $\hat{p}_m$ within error η (i.e., solves the maximization problem within additive error η) with $\eta \le \tau_3 \varepsilon^2$, then

$$\Pr[\, \mathrm{d_H}(\tilde{p}_m, p) \ge \varepsilon\,] \le 5 \exp(-\tau_4 m \varepsilon^2).$$

To apply this theorem, define the function $J_n \colon (0,1) \to \mathbb{R}$ by $J(x) \stackrel{\text{def}}{=} \int_{x^2}^{x} \sqrt{\ln \frac{n}{u}} u^{-1/4}\, du$. By (tedious) computations, one can verify that $J_n(x) \sim_{x \to 0} \frac{4}{3} x^{3/4} \sqrt{\ln \frac{n}{x}}$; this, combined with the bound of Theorem 2.2.33, yields that for any $\varepsilon \in (0,1)$

$$\int_{\varepsilon^2/2^8}^{\sqrt{2}\varepsilon} \sqrt{\mathcal{N}_{[]}(u/\tau_1, \mathcal{LCV}_n, \mathrm{d_H})}\, du = O\left(\varepsilon^{3/4} \sqrt{\ln \frac{n}{\varepsilon}}\right).$$

Thus, setting, for $m \ge 1$, $\varepsilon_m \stackrel{\text{def}}{=} C m^{-2/5} (\ln(mn))^{2/5}$ for a sufficiently big absolute constant $C > 0$ ensures that ε_m satisfies (2.34). Let $\rho_m \stackrel{\text{def}}{=} 1/\varepsilon_m$. It follows that any estimator which, on a sample of size m, approximates the log-concave MLE to within an additive $\eta_m \stackrel{\text{def}}{=} \tau_3 \varepsilon_m^2$ has minimax error

$$\begin{aligned}
\rho_m^2 \sup_{p \in \mathcal{LCV}_n} \mathbb{E}_p[\mathrm{d_H}(\tilde{p}_m, p)^2] &= \sup_{p \in \mathcal{LCV}_n} \int_0^\infty \Pr\left[\, \rho_m^2 \mathrm{d_H}(\tilde{p}_n, p)^2 \ge t \,\right] dt \\
&= \sup_{p \in \mathcal{LCV}_n} \int_0^\infty \Pr\left[\, \mathrm{d_H}(\tilde{p}_n, p) \ge \sqrt{t} \rho_m^{-1} \,\right] dt \\
&\le 1 + \sup_{p \in \mathcal{LCV}_n} \int_1^\infty \Pr\left[\, \mathrm{d_H}(\tilde{p}_n, p) \ge \sqrt{t} \rho_m^{-1} \,\right] dt \\
&= 1 + \sup_{p \in \mathcal{LCV}_n} \int_1^\infty \Pr\left[\, \mathrm{d_H}(\tilde{p}_n, p) \ge \sqrt{t} \varepsilon_m \,\right] dt \\
&\le= 1 + 5 \sup_{p \in \mathcal{LCV}_n} \int_1^\infty \exp(-\tau_4 m t \varepsilon_m^2)\, dt \\
&= 1 + 5 \sup_{p \in \mathcal{LCV}_n} \int_1^\infty \exp(-\tau_4 C m^{1/2} \ln(mn) t)\, dt \\
&= O(1)
\end{aligned}$$

where we used the fact that if $\varepsilon_t > \varepsilon_m$, then ε_t satisfies (2.34) as well (and applied it to $\varepsilon_t = \sqrt{t}\varepsilon_m$). This concludes the proof. □

Chapter 3

Testing Properties of Distributions: Lower Bounds from Reductions

("That's exactly the method," the Bellman bold
In a hasty parenthesis cried,
"That's exactly the way I have always been told
That the capture of Snarks should be tried!")

Lewis Carroll, *The Hunting of the Snark*

In spite of the considerable interest distribution testing has experienced in recent years, our arsenal of tools for proving lower bounds on the sample complexity of testing problems remains sorely limited. There are only a handful of standard techniques to prove such hardness results; and indeed the vast majority of the lower bounds in the literature are shown via *Le Cam's two-point method* (also known as the "easy direction" of Yao's minimax principle) [180, 143].[1] In view of this scarcity, there has been in recent years a trend towards trying to obtain more, or simpler to use, techniques [174, 80]; however, this state of affairs largely remains the same.

In this chapter, we set out to remedy this situation, by providing two general frameworks to establish distribution testing lower bounds. As we shall see, both are *reduction-based* frameworks – put differently, general techniques enabling us to capitalize on someone else's hard work from a different setting or area, and carry over their impossibility results to our distribution testing problem.

- Section 3.1 describes our first reduction, which establishes a simple criterion under which hardness of testing a subproperty $\mathcal{P}' \subseteq \mathcal{P}$ carries over to testing $\mathcal{P}$ itself. This intuitive and seemingly "obvious" result – *"testing a class is at least as hard as testing anything it contains"* – turns out to be false in general, as is easy to see even for some trivial cases. To remedy this unfortunate state of affairs, we identify a relatively benign assumption sufficient to make it hold; and show how this assumption is satisfied by a large number of natural properties.
- In Section 3.2, we reveal a connection between distribution testing and the simultaneous message passing communication model, leading to our second methodology for proving distribution testing lower bounds. Extending the property testing lower bound framework of Blais, Brody, and Matulef [33], we show a simple way of reducing communication problems to distribution testing ones. (Or, in other words, how to harness Alice and Bob's communication issues in order to prove distribution testing lower bounds.)

[1] In this method, one first defines two distributions $\mathcal{Y}$ and $\mathcal{N}$ over distributions that are respectively yes-instances (having the property) and no-instances (far from having the property). Then it remains to show that with high probability over the choice of the instance, every tester that can distinguish between $\mathbf{p}^{\mathsf{yes}} \sim \mathcal{Y}$ and $\mathbf{p}^{\mathsf{no}} \sim \mathcal{N}$ must use at least a certain number of samples.

As an application, these two reduction-based approaches will allow us to show in a clean and painless fashion that most of the upper bounds obtained in Chapter 2 are optimal or near-optimal, and cannot be significantly improved upon. In an unexpected turn of events, our second reduction will also reveal a connection between distribution testing and the field of interpolation theory, shedding light on an "instance-optimal" testing result of Valiant and Valiant [169].

3.1 The Agnostic Learning Reduction

In this section, we describe our first generic framework for proving lower bounds against testing classes of distributions. Specifically, we describe how to *reduce* – under a mild assumption on the property $\mathcal{C}$ – the problem of testing *membership in* $\mathcal{C}$ ("does $\mathbf{p} \in \mathcal{C}$?") to testing *identity to* $\mathbf{p}^*$ ("does $\mathbf{p} = \mathbf{p}^*$?"), for any explicit distribution $\mathbf{p}^*$ in $\mathcal{C}$. While these two problems need not in general be related,[2] we show that our reduction-based approach applies to a large number of natural properties, and obtain lower bounds that nearly match our upper bounds for all of them. Moreover, this lets us derive a simple proof of the lower bound of [2] on testing the class of PBDs. The reader is referred to Theorem 3.1.1 for the formal statement of our reduction-based lower bound theorem; before proceeding further, we restate below some of the corollaries it lets us to easily derive, both in the standard and tolerant testing settings:

Corollary 2.1.6. *Testing log-concavity, convexity, concavity, MHR, unimodality, t-modality, t-histograms, and t-piecewise degree-d distributions each require $\Omega\big(\sqrt{n}/\varepsilon^2\big)$ samples (the last three for $t = o(\sqrt{n})$ and $t(d+1) = o(\sqrt{n})$, respectively), for any $\varepsilon \geq 1/n^{O(1)}$.*[3]

Corollary 2.1.7. *Testing the classes of Binomial and Poisson Binomial Distributions each require $\Omega\big(n^{1/4}/\varepsilon^2\big)$ samples, for any $\varepsilon \geq 1/n^{O(1)}$.*

Corollary 2.1.8. *There exist absolute constants $c > 0$ and $\varepsilon_0 > 0$ such that testing the class of (n,k)-SIIRV distributions requires $\Omega\big(k^{1/2}n^{1/4}/\varepsilon^2\big)$ samples, for any $k = o(n^c)$ and $1/n^{O(1)} \leq \varepsilon \leq \varepsilon_0$.*

Corollary 2.1.11. *Tolerant testing of log-concavity, convexity, concavity, MHR, unimodality, and t-modality each require $\Omega\Big(\frac{1}{(\varepsilon_2-\varepsilon_1)}\frac{n}{\log n}\Big)$ samples (the latter for $t = o(n)$).*

Corollary 2.1.12. *Tolerant testing of the classes of Binomial and Poisson Binomial Distributions each require $\Omega\Big(\frac{1}{(\varepsilon_2-\varepsilon_1)}\frac{\sqrt{n}}{\log n}\Big)$ samples.*

In order to state our results, we will require the usual definition of *agnostic learning*. Recall that an algorithm is said to be a *semi-agnostic learner* for a class $\mathcal{C}$ if it satisfies the following. Given sample access to an arbitrary distribution $\mathbf{p}$ and parameter ε, it outputs a hypothesis $\hat{\mathbf{p}}$ which (with high probability) does

[2] As a simple example, consider the class $\mathcal{C}$ of *all* distributions, for which testing membership is trivial.

[3] Here, the restriction on ε should be read as "for each of these distribution classes, there exists an absolute constant $c > 0$ (which may depend on the corresponding class) such that the result applies for every $\varepsilon \geq \frac{1}{n^c}$."

"almost as well as it gets":

$$\|\mathbf{p} - \hat{\mathbf{p}}\|_1 \leq c \cdot \mathrm{OPT}_{\mathcal{C},\mathbf{p}} + O(\varepsilon)$$

where $\mathrm{OPT}_{\mathcal{C},\mathbf{p}} \stackrel{\text{def}}{=} \inf_{\mathbf{p}' \in \mathcal{C}} \ell_1(\mathbf{p}', \mathbf{p})$, and $c \geq 1$ is some absolute constant (if $c = 1$, the learner is said to be agnostic).

High-level idea. The motivation for our result is the observation of [19] that "monotonicity is at least as hard as uniformity." Unfortunately, their specific argument does not generalize easily to other classes of distributions, making it impossible to extend it readily. The starting point of our approach is to observe that while uniformity testing is hard in general, it becomes very easy *under the promise that the distribution is monotone, or even only close to monotone* (namely, $O\big(1/\varepsilon^2\big)$ samples suffice.)[4] This can give an alternate proof of the lower bound for monotonicity testing, via a different reduction: first, test if the unknown distribution is monotone; if it is, test whether it is uniform, now assuming closeness to monotone.

More generally, this idea applies to any class $\mathcal{C}$ which (a) contains the uniform distribution, and (b) for which we have a $o(\sqrt{n})$-sample agnostic learner $\mathcal{L}$, as follows. Assuming we have a tester $\mathcal{T}$ for $\mathcal{C}$ with sample complexity $o(\sqrt{n})$, define a uniformity tester as below.

- test if $\mathbf{p} \in \mathcal{C}$ using $\mathcal{T}$; if not, reject (as $\mathbf{u} \in \mathcal{C}$, $\mathbf{p}$ cannot be uniform);
- otherwise, agnostically learn $\mathbf{p}$ with $\mathcal{L}$ (since $\mathbf{p}$ is close to $\mathcal{C}$), and obtain hypothesis $\hat{\mathbf{p}}$;
- check offline if $\hat{\mathbf{p}}$ is close to uniform.

By assumption, $\mathcal{T}$ and $\mathcal{L}$ each use $o(\sqrt{n})$ samples, so does the whole process; but this contradicts the lower bound of [22, 138] on uniformity testing. Hence, $\mathcal{T}$ must use $\Omega(\sqrt{n})$ samples.

This "testing-by-narrowing" reduction argument can be further extended to other properties than to uniformity, as we show below:

Theorem 3.1.1. *Let $\mathcal{C}$ be a class of distributions over $[n]$ for which the following holds:*

(i) there exists a semi-agnostic learner $\mathcal{L}$ for $\mathcal{C}$, with sample complexity $q_L(n, \varepsilon, \delta)$ and "agnostic constant" c;

(ii) there exists a subclass $\mathcal{C}_{\mathrm{Hard}} \subseteq \mathcal{C}$ such that testing $\mathcal{C}_{\mathrm{Hard}}$ requires $q_H(n, \varepsilon)$ samples.

Suppose further that $q_L(n, \varepsilon, 1/6) = o(q_H(n, \varepsilon))$. Then, any tester for $\mathcal{C}$ must use $\Omega(q_H(n, \varepsilon))$ samples.

Proof. The above theorem relies on the reduction outlined above, which we rigorously detail here. Assuming $\mathcal{C}$, $\mathcal{C}_{\mathrm{Hard}}$, $\mathcal{L}$ as above (with semi-agnostic constant $c \geq 1$), and a tester $\mathcal{T}$ for $\mathcal{C}$ with sample complexity $q_T(n, \varepsilon)$, we define a tester $\mathcal{T}_{\mathrm{Hard}}$ for $\mathcal{C}_{\mathrm{Hard}}$. On input $\varepsilon \in (0, 1]$ and given sample access to a distribution $\mathbf{p}$ on $[n]$, $\mathcal{T}_{\mathrm{Hard}}$ acts as follows:

- call $\mathcal{T}$ with parameters n, $\frac{\varepsilon'}{c}$ (where $\varepsilon' \stackrel{\text{def}}{=} \frac{\varepsilon}{3}$) and failure probability $1/6$, to $\frac{\varepsilon'}{c}$-test if $\mathbf{p} \in \mathcal{C}$. If not,

[4] Indeed, it is not hard to show that a monotone distribution can only be ε-far from uniform if it puts probability weight $1/2 + \Omega(\varepsilon)$ on the first half of the domain. Estimating this probability weight to an additive $O(\varepsilon)$ is thus sufficient to conclude.

reject.

- otherwise, agnostically learn a hypothesis $\hat{\mathbf{p}}$ for $\mathbf{p}$, with $\mathcal{L}$ called with parameters n, ε' and failure probability $1/6$;
- check offline if $\hat{\mathbf{p}}$ is ε'-close to $\mathcal{C}_{\mathrm{Hard}}$, accept if and only if this is the case.

We condition on both calls (to $\mathcal{T}$ and $\mathcal{L}$) to be successful, which overall happens with probability at least $2/3$ by a union bound. The completeness is immediate: if $\mathbf{p} \in \mathcal{C}_{\mathrm{Hard}} \subseteq \mathcal{C}$, $\mathcal{T}$ accepts, and the hypothesis $\hat{\mathbf{p}}$ satisfies $\|\hat{\mathbf{p}} - \mathbf{p}\|_1 \leq \varepsilon'$. Therefore, $\ell_1(\hat{\mathbf{p}}, \mathcal{C}_{\mathrm{Hard}}) \leq \varepsilon'$, and $\mathcal{T}_{\mathrm{Hard}}$ accepts.

For the soundness, we proceed by contrapositive. Suppose $\mathcal{T}_{\mathrm{Hard}}$ accepts; it means that each step was successful. In particular, $\ell_1(\hat{\mathbf{p}}, \mathcal{C}) \leq \varepsilon'/c$; so that the hypothesis outputted by the agnostic learner satisfies $\|\hat{\mathbf{p}} - \mathbf{p}\|_1 \leq c \cdot \mathrm{OPT} + \varepsilon' \leq 2\varepsilon'$. In turn, since the last step passed and by a triangle inequality we get, as claimed, $\ell_1(\mathbf{p}, \mathcal{C}_{\mathrm{Hard}}) \leq 2\varepsilon' + \ell_1(\hat{\mathbf{p}}, \mathcal{C}_{\mathrm{Hard}}) \leq 3\varepsilon' = \varepsilon$.

Observing that the overall sample complexity is $q_T(n, \frac{\varepsilon'}{c}) + q_L(n, \varepsilon', \frac{1}{6}) = q_T(n, \frac{\varepsilon'}{c}) + o(q_H(n, \varepsilon'))$ concludes the proof. □

Taking $\mathcal{C}_{\mathrm{Hard}}$ to be the singleton consisting of the uniform distribution, and from the semi-agnostic learners of [56, 55] (each with sample complexity either $\mathrm{poly}(1/\varepsilon)$ or $\mathrm{poly}(\log n, 1/\varepsilon)$), we obtain the following:[5]

Corollary 2.1.6. *Testing log-concavity, convexity, concavity, MHR, unimodality, t-modality, t-histograms, and t-piecewise degree-d distributions each require $\Omega\big(\sqrt{n}/\varepsilon^2\big)$ samples (the last three for $t = o(\sqrt{n})$ and $t(d+1) = o(\sqrt{n})$, respectively), for any $\varepsilon \geq 1/n^{O(1)}$.*[6]

Similarly, we can use another result of [64] which shows how to agnostically learn Poisson Binomial Distributions with $\tilde{O}\big(1/\varepsilon^2\big)$ samples.[7] Taking $\mathcal{C}_{\mathrm{Hard}}$ to be the single $\mathrm{Bin}(n, 1/2)$ distribution (along with the testing lower bound of [169]), this yields the following:

Corollary 2.1.7. *Testing the classes of Binomial and Poisson Binomial Distributions each require $\Omega\big(n^{1/4}/\varepsilon^2\big)$ samples, for any $\varepsilon \geq 1/n^{O(1)}$.*

Finally, we derive a lower bound on testing k-SIIRVs from the agnostic learner of [73] (which has sample complexity $\mathrm{poly}(k, 1/\varepsilon)$, independent of n):

Corollary 2.1.8. *There exist absolute constants $c > 0$ and $\varepsilon_0 > 0$ such that testing the class of (n, k)-SIIRV distributions requires $\Omega\big(k^{1/2}n^{1/4}/\varepsilon^2\big)$ samples, for any $k = o(n^c)$ and $1/n^{O(1)} \leq \varepsilon \leq \varepsilon_0$.*

Proof of Corollary 2.1.8. To prove this result, it is enough by Theorem 3.1.1 to exhibit a particular k-SIIRV S such that testing identity to S requires this many samples. Moreover, from [169] this last part amounts to

[5] Specifically, these lower bounds hold as long as $\varepsilon = \Omega(1/n^\alpha)$ for some absolute constant $\alpha > 0$ (so that the sample complexity of the agnostic learner is indeed negligible in front of $\sqrt{n}/\varepsilon^2$).

[6] Here, the restriction on ε should be read as "for each of these distribution classes, there exists an absolute constant $c > 0$ (which may depend on the corresponding class) such that the result applies for every $\varepsilon \geq \frac{1}{n^c}$."

[7] Note the quasi-quadratic dependence on ε of the learner, which allows us to get ε into our lower bound for $n \gg \mathrm{poly}\log(1/\varepsilon)$.

proving that the (truncated) 2/3-norm $\|S_{-\varepsilon}^{-\max}\|_{2/3}$ of S is $\Omega\big(k^{1/2}n^{1/4}\big)$ (for every $\varepsilon \in (0, \varepsilon_0)$, for some small $\varepsilon_0 > 0$). Our hard instance S will be defined as follows: it is defined as the distribution of $X_1 + \cdots + X_n$, where the X_i's are independent integer random variables uniform on $[\![k]\!]$ (in particular, for $k = 2$ we get a $\mathrm{Bin}(n, 1/2)$ distribution). It is straightforward to verify that $\mathbb{E}S = \frac{n(k-1)}{2}$ and $\sigma^2 \stackrel{\text{def}}{=} \operatorname{Var} S = \frac{(k^2-1)n}{12} = \Theta\big(k^2 n\big)$; moreover, S is log-concave (as the convolution of n uniform distributions). From this last point, we get that (i) the maximum probability of S, attained at its mode, is $\|S\|_\infty = \Theta(1/\sigma)$; and (ii) for every j in an interval I of length 2σ centered at this mode, $S(j) \geq \Omega(\|S\|_\infty)$ (see e.g. [84, Lemma 5.7] for the latter point). Define now ε_0 as an absolute constant such that $2\varepsilon_0 \leq \mathbf{p}(I) = \Omega(1)$.

We want to lower bound $\|S_{-\varepsilon}^{-\max}\|_{2/3}$, for $\varepsilon \leq \varepsilon_0$; as by the above the "$-\max$" part can only change the value by $\|S\|_\infty = o(1)$, we can ignore it. Turning to the $-\varepsilon$ part, i.e. the removal of the ε probability mass of the elements with smallest probability, note that this can only result in zeroing out at most $\frac{\varepsilon}{\mathbf{p}(I)}\,|I| \leq \frac{1}{2}\,|I|$ elements in I (call these $J_\varepsilon \subseteq I$). From this, we obtain that

$$\|S_{-\varepsilon}^{-\max}\|_{2/3} \geq \Big(\sum_{j \in I \setminus J_\varepsilon} S(j)^{2/3}\Big)^{3/2} \geq \left(\frac{1}{2} \cdot 2\sigma \cdot \Omega(1/\sigma)^{2/3}\right)^{3/2} = \Omega\Big(\sigma^{1/2}\Big) = \Omega\Big(k^{1/2}n^{1/4}\Big)$$

which concludes the proof. □

3.1.1 Tolerant Testing

This lower bound framework from the previous section carries to *tolerant* testing as well, resulting in this analogue to Theorem 3.1.1:

Theorem 3.1.2. *Let $\mathcal{C}$ be a class of distributions over $[n]$ for which the following holds:*

(i) there exists a semi-agnostic learner $\mathcal{L}$ for $\mathcal{C}$, with sample complexity $q_L(n, \varepsilon, \delta)$ and "agnostic constant" c;

(ii) there exists a subclass $\mathcal{C}_{\mathrm{Hard}} \subseteq \mathcal{C}$ such that tolerant *testing $\mathcal{C}_{\mathrm{Hard}}$ requires $q_H(n, \varepsilon_1, \varepsilon_2)$ samples for some parameters $\varepsilon_2 > (4c+1)\varepsilon_1$.*

Suppose further that $q_L(n, \varepsilon_2 - \varepsilon_1, 1/10) = o(q_H(n, \varepsilon_1, \varepsilon_2))$. Then, any tolerant *tester for $\mathcal{C}$ must use $\Omega(q_H(n, \varepsilon_1, \varepsilon_2))$ samples (for some explicit parameters $\varepsilon_1', \varepsilon_2'$).*

Proof. The argument follows the same ideas as for Theorem 3.1.1, up to the details of the parameters. Assuming $\mathcal{C}$, $\mathcal{C}_{\mathrm{Hard}}$, $\mathcal{L}$ as above (with semi-agnostic constant $c \geq 1$), and a tolerant tester $\mathcal{T}$ for $\mathcal{C}$ with sample complexity $q(n, \varepsilon_1, \varepsilon_2)$, we define a tolerant tester $\mathcal{T}_{\mathrm{Hard}}$ for $\mathcal{C}_{\mathrm{Hard}}$. On input $0 < \varepsilon_1 < \varepsilon_2 \leq 1$ with $\varepsilon_2 > (4c+1)\varepsilon_1$, and given sample access to a distribution $\mathbf{p}$ on $[n]$, $\mathcal{T}_{\mathrm{Hard}}$ acts as follows. After setting $\varepsilon_1' \stackrel{\text{def}}{=} \frac{\varepsilon_2 - \varepsilon_1}{4}$, $\varepsilon_2' \stackrel{\text{def}}{=} \frac{\varepsilon_2 - \varepsilon_1}{2}$, $\varepsilon' \stackrel{\text{def}}{=} \frac{\varepsilon_2 - \varepsilon_1}{16}$ and $\tau \stackrel{\text{def}}{=} \frac{6\varepsilon_2 + 10\varepsilon_1}{16}$,

- call $\mathcal{T}$ with parameters n, $\frac{\varepsilon_1'}{c}$, $\frac{\varepsilon_2'}{c}$ and failure probability $1/6$, to tolerantly test if $\mathbf{p} \in \mathcal{C}$. If $\ell_1(\mathbf{p}, \mathcal{C}) > \varepsilon_2'/c$, reject.
- otherwise, agnostically learn a hypothesis $\hat{\mathbf{p}}$ for $\mathbf{p}$, with $\mathcal{L}$ called with parameters n, ε' and failure

probability $1/6$;

- check offline if $\hat{\mathbf{p}}$ is τ-close to $\mathcal{C}_{\mathrm{Hard}}$, accept if and only if this is the case.

We condition on both calls (to $\mathcal{T}$ and $\mathcal{L}$) to be successful, which overall happens with probability at least $2/3$ by a union bound. We first argue completeness: assume $\ell_1(\mathbf{p}, \mathcal{C}_{\mathrm{Hard}}) \leq \varepsilon_1$. This implies $\ell_1(\mathbf{p}, \mathcal{C}) \leq \varepsilon_1$, so that $\mathcal{T}$ accepts as $\varepsilon_1 \leq \varepsilon_1'/c$ (which is the case because $\varepsilon_2 > (4c+1)\varepsilon_1$). Thus, the hypothesis $\hat{\mathbf{p}}$ satisfies $\|\hat{\mathbf{p}} - \mathbf{p}\|_1 \leq c \cdot \varepsilon_1'/c + \varepsilon' = \varepsilon_1' + \varepsilon'$. Therefore, $\ell_1(\hat{\mathbf{p}}, \mathcal{C}_{\mathrm{Hard}}) \leq \|\hat{\mathbf{p}} - \mathbf{p}\|_1 + \ell_1(\mathbf{p}, \mathcal{C}_{\mathrm{Hard}}) \leq \varepsilon_1' + \varepsilon' + \varepsilon_1 < \tau$, and $\mathcal{T}_{\mathrm{Hard}}$ accepts.

For the soundness, we again proceed by contrapositive. Suppose $\mathcal{T}_{\mathrm{Hard}}$ accepts; it means that each step was successful. In particular, $\ell_1(\hat{\mathbf{p}}, \mathcal{C}) \leq \varepsilon_2'/c$; so that the hypothesis outputted by the agnostic learner satisfies $\|\hat{\mathbf{p}} - \mathbf{p}\|_1 \leq c \cdot \mathrm{OPT} + \varepsilon' \leq \varepsilon_2' + \varepsilon'$. In turn, since the last step passed and by a triangle inequality we get, as claimed, $\ell_1(\mathbf{p}, \mathcal{C}_{\mathrm{Hard}}) \leq \varepsilon_2' + \varepsilon' + \ell_1(\hat{\mathbf{p}}, \mathcal{C}_{\mathrm{Hard}}) \leq \varepsilon_2' + \varepsilon' + \tau < \varepsilon_2$.

Observing that the overall sample complexity is $q_T(n, \frac{\varepsilon_1'}{c}, \frac{\varepsilon_2'}{c}) + q_L(n, \varepsilon', \frac{1}{10}) = q_T(n, \frac{\varepsilon'}{c}) + o(q_H(n, \varepsilon'))$ concludes the proof. □

As before, we instantiate the general theorem to obtain specific lower bounds for tolerant testing of the classes we covered in this paper. That is, taking $\mathcal{C}_{\mathrm{Hard}}$ to be the singleton consisting of the uniform distribution (combined with the tolerant testing lower bound of [167] (restated in Theorem 3.1.6), which states that tolerant testing of uniformity over $[n]$ requires $\Omega\left(\frac{n}{\log n}\right)$ samples), and again from the semi-agnostic learners of [56, 55] (each with sample complexity either $\mathrm{poly}(1/\varepsilon)$ or $\mathrm{poly}(\log n, 1/\varepsilon)$), we obtain the following:

Corollary 2.1.11. *Tolerant testing of log-concavity, convexity, concavity, MHR, unimodality, and t-modality each require $\Omega\left(\frac{1}{(\varepsilon_2-\varepsilon_1)}\frac{n}{\log n}\right)$ samples (the latter for $t = o(n)$).*

Similarly, we again turn to the class of Poisson Binomial Distributions, for which we can invoke as before the $\tilde{O}(1/\varepsilon^2)$-sample agnostic learner of [64]. As before, we would like to choose for $\mathcal{C}_{\mathrm{Hard}}$ the single $\mathrm{Bin}(n, 1/2)$ distribution; however, as no tolerant testing lower bound for this distribution exists – to the best of our knowledge – in the literature, we first need to establish the lower bound we will rely upon:

Theorem 3.1.3. *There exists an absolute constant $\varepsilon_0 > 0$ such that the following holds. Any algorithm which, given sampling access to an unknown distribution $\mathbf{p}$ on Ω and parameter $\varepsilon \in (0, \varepsilon_0)$, distinguishes with probability at least $2/3$ between (i) $\|\mathbf{p} - \mathrm{Bin}(n, 1/2)\|_1 \leq \varepsilon$ and (ii) $\|\mathbf{p} - \mathrm{Bin}(n, 1/2)\|_1 \geq 100\varepsilon$ must use $\Omega\left(\frac{1}{\varepsilon}\frac{\sqrt{n}}{\log n}\right)$ samples.*

The proof relies on a reduction from tolerant testing of *uniformity*, drawing on a result of Valiant and Valiant [167]; and is deferred to Section 3.1.1.1. With Theorem 3.1.3 in hand, we can apply Theorem 3.1.2 to obtain the desired lower bound:

Corollary 2.1.12. *Tolerant testing of the classes of Binomial and Poisson Binomial Distributions each require $\Omega\left(\frac{1}{(\varepsilon_2-\varepsilon_1)}\frac{\sqrt{n}}{\log n}\right)$ samples.*

We observe that both Corollary 2.1.11 and Corollary 2.1.12 are tight (with regard to the dependence on n), as was shown in the previous chapter (Section 2.1.5).

3.1.1.1 Proof of Theorem 3.1.3

The theorem will be a consequence of the (slightly) more general result below:

Theorem 3.1.4. *There exist absolute constants $\varepsilon_0 > 0$ and $\lambda > 0$ such that the following holds. Any algorithm which, given sample access to an unknown distribution $\mathbf{p}$ on Ω and parameter $\varepsilon \in (0, \varepsilon_0)$, distinguishes with probability at least $2/3$ between (i) $\|\mathbf{p} - \mathrm{Bin}\big(n, \frac{1}{2}\big)\|_1 \leq \varepsilon$ and (ii) $\|\mathbf{p} - \mathrm{Bin}\big(n, \frac{1}{2}\big)\|_1 \geq \lambda\varepsilon^{1/3} - \varepsilon$ must use $\Omega\Big(\varepsilon \frac{\sqrt{n}}{\log(\varepsilon n)}\Big)$ samples.*

By choosing a suitable ε and working out the corresponding parameters, this for instance enables us to derive the following:

Corollary 3.1.5. *There exists an absolute constant $\varepsilon_0 \in (0, 1/1000)$ such that the following holds. Any algorithm which, given sample access to an unknown distribution $\mathbf{p}$ on Ω, distinguishes with probability at least $2/3$ between (i) $\|\mathbf{p} - \mathrm{Bin}\big(n, \frac{1}{2}\big)\|_1 \leq \varepsilon_0$ and (ii) $\|\mathbf{p} - \mathrm{Bin}\big(n, \frac{1}{2}\big)\|_1 \geq 100\varepsilon_0$ must use $\Omega\Big(\frac{\sqrt{n}}{\log n}\Big)$ samples.*

Proof of Corollary 3.1.5. The corollary follows from the proof of Theorem 3.1.4, by choosing $\varepsilon_0 > 0$ sufficiently small so that $\frac{\lambda\varepsilon_0^{1/3} - \varepsilon_0}{\varepsilon_0} \geq 100$. □

By standard techniques, this will in turn imply Theorem 3.1.3.[8]

Proof of Theorem 3.1.4. Hereafter, we write for convenience $B_n \stackrel{\mathrm{def}}{=} \mathrm{Bin}\big(n, \frac{1}{2}\big)$. To prove this lower bound, we will rely on the following:

Theorem 3.1.6 ([167, Theorem 1])**.** *For any constant $\phi \in (0, 1/4)$, following holds. Any algorithm which, given sample access to an unknown distribution $\mathbf{p}$ on $\{1, \ldots, N\}$, distinguishes with probability at least $2/3$ between (i) $\|\mathbf{p} - \mathbf{u}_N\|_1 \leq \phi$ and (ii) $\|\mathbf{p} - \mathbf{u}_N\|_1 \geq \frac{1}{2} - \phi$, must have sample complexity at least $\frac{\phi}{32}\frac{N}{\log N}$.*

Without loss of generality, assume n is even (so that B_n has only one mode located at $\frac{n}{2}$). For $c > 0$, we write $I_{n,c}$ for the interval $\{\frac{n}{2} - c\sqrt{n}, \ldots, \frac{n}{2} + c\sqrt{n}\}$ and $J_{n,c} \stackrel{\mathrm{def}}{=} \Omega \setminus I_{n,c}$.

Fact 3.1.7. *For any $c > 0$,*

$$\frac{B_n(\frac{n}{2} + c\sqrt{n})}{B_n(n/2)}, \frac{B_n(\frac{n}{2} - c\sqrt{n})}{B_n(n/2)} \underset{n\to\infty}{\sim} e^{-2c^2}$$

and

$$B_n(I_{n,c}) \in (1 \pm o(1)) \cdot [e^{-2c^2}, 1] \cdot 2c\sqrt{\frac{2}{\pi}} = \Theta(c)\,.$$

[8] Namely, for $\varepsilon \in (0, \varepsilon_0)$, define the mixture $\mathbf{p}_\varepsilon \stackrel{\mathrm{def}}{=} \frac{\varepsilon}{\varepsilon_0}\mathbf{p} + (1 - \frac{\varepsilon}{\varepsilon_0})\,\mathrm{Bin}(n, 1/2)$. Being able to distinguish $\|\mathbf{p}_\varepsilon - \mathrm{Bin}(n, 1/2)\|_1 \leq \varepsilon$ from $\|\mathbf{p}_\varepsilon - \mathrm{Bin}(n, 1/2)\|_1 \geq 100\varepsilon$ in q samples then allows one to distinguish $\|\mathbf{p} - \mathrm{Bin}(n, 1/2)\|_1 \leq \varepsilon_0$ from $\|\mathbf{p} - \mathrm{Bin}(n, 1/2)\|_1 \geq 100\varepsilon_0$ in $O(\varepsilon \cdot q)$ samples.

The reduction proceeds as follows: given sampling access to $\mathbf{p}$ on $[N]$, we can simulate sampling access to a distribution $\mathbf{p}'$ on $[n]$ (where $n = \Theta(N^2)$) such that

- if $\|\mathbf{p} - \mathbf{u}_N\|_1 \leq \phi$, then $\|\mathbf{p}' - B_n\|_1 < \varepsilon$;
- if $\|\mathbf{p} - \mathbf{u}_N\|_1 \geq \frac{1}{2} - \phi$, then $\|\mathbf{p}' - B_n\|_1 > \varepsilon' - \varepsilon$

for $\varepsilon \stackrel{\text{def}}{=} \Theta(\phi^{3/2})$ and $\varepsilon' \stackrel{\text{def}}{=} \Theta(\phi^{\frac{1}{2}})$; in a way that preserves the sample complexity. The high-level idea is that (by the above fact) the Binomial distribution over Ω is almost uniform on the middle $O(\sqrt{n})$ elements, and has a constant fraction of its probability mass there: we can therefore "embed" the tolerant uniformity testing lower bound (for support $O(\sqrt{n})$) into this middle interval.

More precisely, define $c \stackrel{\text{def}}{=} \sqrt{\frac{1}{2}\ln\frac{1}{1-\phi}} = \Theta(\sqrt{\phi})$ (so that $\phi = 1 - e^{-2c^2}$) and n such that $|I_{n,c}| = N$ (that is, $n = (N/(2c))^2 = \Theta(N^2/\phi)$). From now on, we can therefore identify $[N]$ to $I_{n,c}$ in the obvious way, and see a draw from $\mathbf{p}$ as an element in $I_{n,c}$.

Let $p \stackrel{\text{def}}{=} B_n(I_{n,c}) = \Theta(\sqrt{\phi})$, and $B_{n,c}$, $\bar{B}_{n,c}$ respectively denote the conditional distributions induced by B_n on $I_{n,c}$ and $J_{n,c}$. Intuitively, we want $\mathbf{p}$ to be mapped to the conditional distribution of $\mathbf{p}'$ on $I_{n,c}$, and the conditional distribution of $\mathbf{p}'$ on $J_{n,c}$ to be exactly $\bar{B}_{n,c}$. This is achieved by defining $\mathbf{p}'$ by the process below:

- with probability p, we draw a sample from $\mathbf{p}$ (seen as an element of $I_{n,c}$);
- with probability $1 - p$, we draw a sample from $\bar{B}_{n,c}$.

Let $\tilde{B}_n$ be defined as the distribution which exactly matches B_n on $J_{n,c}$, but is uniform on $I_{n,c}$:

$$\tilde{B}_n(i) = \begin{cases} \frac{p}{|I_{n,c}|} & i \in I_{n,c} \\ B_n(i) & i \in J_{n,c} \end{cases}$$

From the above, we have that $\|\mathbf{p}' - \tilde{B}_n\|_1 = p \cdot \|\mathbf{p} - \mathbf{u}_N\|_1$. Furthermore, by Fact 3.1.7, Lemma 1.4.7 and the definition of $I_{n,c}$, we get that $\|B_n - \tilde{B}_n\|_1 = p \cdot \|(B_n)_{I_{n,c}} - \mathbf{u}_{I_{n,c}}\|_1 \leq p \cdot \phi$. Putting it all together,

- If $\|\mathbf{p} - \mathbf{u}_N\|_1 \leq \phi$, then by the triangle inequality $\|\mathbf{p}' - B_n\|_1 \leq p(\phi + \phi) = 2p\phi$;
- If $\|\mathbf{p} - \mathbf{u}_N\|_1 \geq \frac{1}{2} - \phi$, then similarly $\|\mathbf{p}' - B_n\|_1 \geq p(\frac{1}{2} - \phi - \phi) = \frac{p}{4} - 2p\phi$.

Recalling that $p = \Theta(\sqrt{\phi})$ and setting $\varepsilon \stackrel{\text{def}}{=} 2p\phi$ concludes the reduction. From Theorem 3.1.6, we conclude that

$$\frac{\phi}{32}\frac{N}{\log N} = \Omega\left(\phi\frac{\sqrt{\phi n}}{\log(\phi n)}\right) = \Omega\left(\varepsilon\frac{\sqrt{n}}{\log(\varepsilon n)}\right)$$

samples are necessary. □

3.2 The Communication Complexity Reduction

3.2.1 Introduction

In this section, we reveal a connection between distribution testing and the simultaneous message passing (SMP) communication model, which in turn leads to a new methodology for proving distribution testing lower bounds. Recall that in a private-coin SMP protocol, Alice and Bob are given strings $x, y \in \{0,1\}^k$ (respectively), and each of the players is allowed to send a message to a referee (which depends on the player's input and private randomness) who is then required to decide whether $f(x,y) = 1$ by only looking at the players' messages and flipping coins.

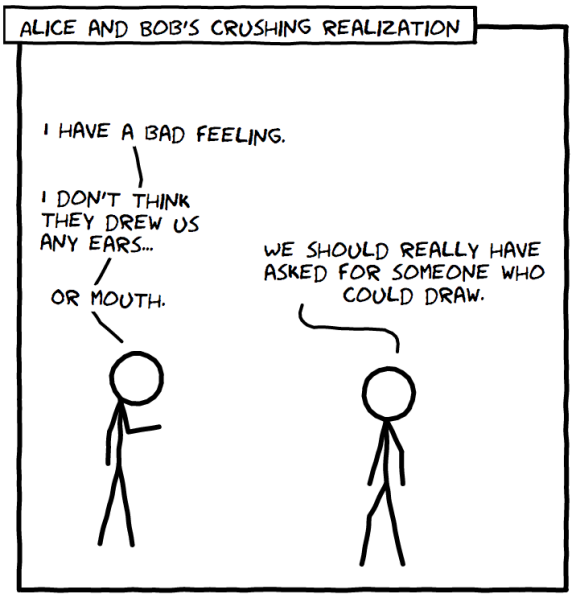

Figure 3.1: Communicating has been somewhat hard for Alice and Bob lately.

Extending the framework of Blais, Brody, and Matulef [33], we show a simple way of reducing (private-coin) SMP problems to distribution testing problems. This foregoing methodology allows us to prove new distribution testing lower bounds, as well as to provide simpler proofs of known lower bounds for problems such as testing uniformity, monotonicity, and k-modality (see Section 3.2.7).

Our main result is a characterization of the sample complexity of the distribution identity testing problem in terms of a key operator in the study of interpolation spaces, which arises naturally from our reduction and for which we are able to provide an intuitive interpretation. Recall that in this problem, the goal is to determine whether a distribution $\mathbf{q}$ over domain Ω (denoted $\mathbf{q} \in \Delta(\Omega)$) is identical to a fixed distribution $\mathbf{p}$; that is, given a full description of $\mathbf{p} \in \Delta(\Omega)$, we ask how many independent samples from $\mathbf{q}$ are needed to decide whether $\mathbf{q} = \mathbf{p}$, or whether $\mathbf{q}$ is ε-far in ℓ_1-distance from $\mathbf{p}$.[9]

[9]Note that this is in fact a family of massively parameterized properties $\{\Pi_p\}_{p \in \Delta(\Omega)}$, where Π_p is the property of being identical to $\mathbf{p}$. See [135] for an excellent survey concerning massively parameterized properties.

Property	Our results	Previous bounds
Uniformity	$\tilde{\Omega}\left(\sqrt{n}\right)$	$\Theta\left(\sqrt{n}\right)$ [104, 138]
Identity to $\mathbf{p}$	$\Omega\left(\kappa_{\mathbf{p}}^{-1}(1-\varepsilon)\right), O\left(\kappa_{\mathbf{p}}^{-1}(1-c\cdot\varepsilon)\right)$	$\Omega\left(\|\mathbf{p}_{\varepsilon}^{-\max}\|_{2/3}\right), O\left(\|\mathbf{p}_{c'\cdot\varepsilon}^{-\max}\|_{2/3}\right)$ [169]
Monotonicity	$\tilde{\Omega}\left(\sqrt{n}\right)$	$\Theta\left(\sqrt{n}\right)$ [19, 3, 51]
k-modal	$\tilde{\Omega}\left(\sqrt{n}\right)$	$\tilde{\Omega}\left(\max(\sqrt{n}, k)\right)$ [43]
Log-concavity, Monotone Hazard Rate	$\tilde{\Omega}\left(\sqrt{n}\right)$	$\Theta\left(\sqrt{n}\right)$ [3, 51]
Binomial, Poisson Binomial	$\tilde{\Omega}\left(n^{1/4}\right)$	$\Theta\left(n^{1/4}\right)$ ([2, 51]
Symmetric sparse support	$\tilde{\Omega}\left(\sqrt{n}\right)$	
Junta distributions (PAIRCOND model)	$\Omega(k)$	

Table 3.1: Summary of results obtained *via* our communication complexity methodology. All the bounds are stated for constant proximity parameter ε.

In a recent and influential work, Valiant and Valiant [169] showed that the sample complexity of the foregoing question is closely related to the $\ell_{2/3}$-quasinorm of $\mathbf{p}$, defined as $\|\mathbf{p}\|_{2/3} = \left(\sum_{\omega\in\Omega} |\mathbf{p}(\omega)|^{2/3}\right)^{3/2}$. That is, viewing a distribution $\mathbf{p} \in \Delta(\Omega)$ as an $|\Omega|$-dimensional vector of probabilities, let $\mathbf{p}_{-\varepsilon}^{-\max}$ be the vector obtained from $\mathbf{p}$ by zeroing out the largest entry as well as the set of smallest entries summing to ε (note that $\mathbf{p}_{-\varepsilon}^{-\max}$ is no longer a probability distribution). Valiant and Valiant gave an ε-tester[10] for testing identity to $\mathbf{p}$ with sample complexity $O\left(\|\mathbf{p}_{-c\varepsilon}^{-\max}\|_{2/3}\right)$, where $c > 0$ is a universal constant, and complemented this result with a lower bound of $\Omega\left(\|\mathbf{p}_{-\varepsilon}^{-\max}\|_{2/3}\right)$.[11]

In this work, using our new methodology, we show alternative and similarly tight bounds on the complexity of identity testing, in terms of a more intuitive measure (as we discuss below) and using simpler arguments. Specifically, we prove that the sample complexity is essentially determined by a fundamental quantity in the theory of interpolation of Banach spaces, known as Peetre's *K-functional*. Formally, for a distribution $\mathbf{p} \in \Delta(\Omega)$, the K-functional between ℓ_1 and ℓ_2 spaces is the operator defined for $t > 0$ by

$$\kappa_{\mathbf{p}}(t) = \inf_{\mathbf{p}'+\mathbf{p}''=\mathbf{p}} \|\mathbf{p}'\|_1 + t\|\mathbf{p}''\|_2.$$

This operator can be thought of as an interpolation norm between the ℓ_1 and ℓ_2 norms of the distribution $\mathbf{p}$ (controlled by the parameter t), naturally inducing a partition of $\mathbf{p}$ into two sub-distributions: $\mathbf{p}'$, which consists of "heavy hitters" in ℓ_1-norm, and $\mathbf{p}''$, which has a bounded ℓ_2-norm. Indeed, the approach of isolating elements with large mass and testing in ℓ_2-norm seems inherent to the problem of identity testing, and is the core component of both early works [104, 21] and more recent ones [82, 80, 102]. As a further connection to the identity testing question, we provide an easily interpretable proxy for this measure $\kappa_{\mathbf{p}}$, showing that the

[10] Throughout the introduction, we fix ε to be small constant and refer to a tester with respect to proximity parameter ε as an *ε-tester*.

[11] We remark that for certain $\mathbf{p}$'s, the asymptotic behavior of $O\left(\|\mathbf{p}_{-c\varepsilon}^{-\max}\|_{2/3}\right)$ strongly depends on the constant c, and so it cannot be omitted from the expression. We further remark that this result was referred to by Valiant and Valiant as "instance-optimal identity testing" as the resulting bounds are phrased as a function of the distribution $\mathbf{p}$ itself – instead of the standard parameter which is the domain size n.

K-functional between the ℓ_1 and ℓ_2 norms of the distribution $\mathbf{p}$ is closely related to the size of the effective support of $\mathbf{p}$, which is the number of supported elements that constitute the vast majority of the mass of $\mathbf{p}$; that is, we say that $\mathbf{p}$ has ε-*effective support* of size T if $1 - O(\varepsilon)$ of the mass of $\mathbf{p}$ is concentrated on T elements (see Section 3.2.2.4 for details).

Having defined the K-functional, we can proceed to state the lower bound we derive for the problem.[12]

Theorem 3.2.1 (Informally stated)**.** *Any ε-tester of identity to $\mathbf{p} \in \Delta(\Omega)$ must have sample complexity $\Omega\big(\kappa_{\mathbf{p}}^{-1}(1-2\varepsilon)\big)$.*

In particular, straightforward calculations show that for the uniform distribution we obtain a tight lower bound of $\Omega(\sqrt{n})$, and for the Binomial distribution we obtain a tight lower bound of $\Omega\big(n^{1/4}\big)$.

To show that tightness of the lower bound above, we complement it with a nearly matching upper bound, also expressed in terms of the K-functional.

Theorem 3.2.2 (Informally stated)**.** *There exist an absolute constant $c > 0$ and an ε-tester of identity to $\mathbf{p} \in \Delta(\Omega)$ that uses $O\big(\kappa_{\mathbf{p}}^{-1}(1-c\varepsilon)\big)$ samples.*[13]

We remark that for some distributions the bounds in Theorems 3.2.1 and 3.2.2 are tighter than the bounds in [169], whereas for other distributions it is the other way around (see discussion in Section 3.2.5).

In the following section, we provide an overview of our new methodology as well as the proofs for the above theorems. We also further discuss the interpretability of the K-functional and show its close connection to the effective support size. We conclude this section by outlining a couple of extensions of our methodology.

Dealing with sub-constant values of the proximity parameter. Similarly to the communication complexity methodology for proving property testing lower bounds [33], our method inherently excels in the regime of *constant* values of the proximity parameter ε. Therefore, in this work we indeed focus on the constant proximity regime. However, in Section 3.2.4.1 we demonstrate how to obtain lower bounds that asymptotically increase as ε tends to zero, via an extension of our general reduction.

Extending the methodology to testing with conditional samples. Testers with sample access are by far the most commonly studied algorithms for distribution testing. However, many scenarios that arise both in theory and practice are not fully captured by this model. In a recent line of works [54, 49, 1, 94, 97], testers with access to *conditional* samples were considered, addressing situations in which one can control the samples that are obtained by requesting samples conditioned on membership on subsets of the domain. In Section 3.2.8, we give an example showing that it is possible to extend our methodology to obtain lower bounds in the conditional sampling model.

[12] As stated, this result is a slight strengthening of our communication complexity reduction, which yields a lower bound of $\Omega\big(\kappa_{\mathbf{p}}^{-1}(1-2\varepsilon)/\log n\big)$. This strengthening is described in Section 3.2.6.3.

[13] Similarly to the [169] bound, for certain $\mathbf{p}$'s, the asymptotic behavior of $O\big(\kappa_{\mathbf{p}}^{-1}(1-2\varepsilon)\big)$ depends on the constant c, and so it cannot be omitted from the expression.

3.2.1.1 Organization

We first give a technical overview in Section 3.2.2, demonstrating the new methodology and presenting our bounds on identity testing. In Section 3.2.3 we formally state and analyze the SMP reduction methodology for proving distribution testing lower bounds. In Section 3.2.4, we instantiate the basic reduction, obtaining a lower bound on uniformity testing, and in Section 3.2.4.1 show how to extend the methodology to deal with sub-constant values of the proximity parameter. (We stress that Section 3.2.4.1 is *not* a prerequisite for the rest of the sections, and can be skipped at the reader's convenience.) In Section 3.2.5 we provide an exposition to the K-functional and generalize inequalities that we shall need for the following sections. Section 3.2.6 then contains the proofs of both lower and upper bounds on the problem of identity testing, in terms of the K-functional. In Section 3.2.7, we demonstrate how to easily obtain lower bounds for other distribution testing problems. Finally, in Section 3.2.8 we discuss extensions to our methodology; specifically, we explain how to obtain lower bounds in various metrics, and show a reduction from communication complexity to distribution testing in the conditional sampling model.

3.2.2 Technical Overview

In this section we provide an overview of the proof of our main result, which consists of new lower and upper bounds on the sample complexity of testing identity to a given distribution, expressed in terms of an intuitive, easily interpretable measure. To do so, we first introduce the key component of this proof, the methodology for proving lower bounds on distribution testing problems via reductions from SMP communication complexity. We then explain how the relation to the theory of interpolation spaces and the so-called K-functional naturally arises when applying this methodology to the identity testing problem.

For the sake of simplicity, throughout the overview we fix the domain $\Omega = [n]$ and fix the proximity parameter ε to be a small constant. We begin in Section 3.2.2.1 by describing a simple "vanilla" reduction for showing an $\tilde{\Omega}(\sqrt{n})$ lower bound on the complexity of testing that a distribution is uniform. Then, in Section 3.2.2.2 we extend the foregoing approach to obtain a new lower bound on the problem of testing identity to a fixed distribution. This lower bound depends on the best rate obtainable by a special type of error-correcting codes, which we call $\mathbf{p}$-*weighted codes*. In Section 3.2.2.3, we show how to relate the construction of such codes to concentration of measure inequalities for weighted sums of Rademacher random variables; furthermore, we discuss how the use of the K-*functional*, an interpolation norm between ℓ_1 and ℓ_2 spaces, leads to stronger concentration inequalities than the ones derived by Chernoff bounds or the central limit theorem. Finally, in Section 3.2.2.4 we establish nearly matching upper bounds for testing distribution identity in terms of this K-functional, using a proxy known as the Q-norm. We then infer that the sample complexity of testing identity to a distribution $\mathbf{p}$ is roughly determined by the size of the *effective support* of $\mathbf{p}$ (which is, loosely speaking, the number of supported elements which together account for the vast majority of the mass of $\mathbf{p}$).

3.2.2.1 Warmup: Uniformity Testing.

Consider the problem of testing whether a distribution $\mathbf{q} \in \Delta([n])$ is the *uniform distribution*; that is, how many (independent) samples from $\mathbf{q}$ are needed to decide whether $\mathbf{q}$ is the uniform distribution over $[n]$, or whether $\mathbf{q}$ is ε-far in ℓ_1-distance from it. We reduce the SMP communication complexity problem of *equality* to the distribution testing problem of uniformity testing.

Recall that in a private-coin SMP protocol for equality, Alice and Bob are given strings $x, y \in \{0,1\}^k$ (respectively), and each of the players is allowed to send a message to a referee (which depends on the player's input and private randomness) who is then required to decide whether $x = y$ by only looking at the players' messages and flipping coins.

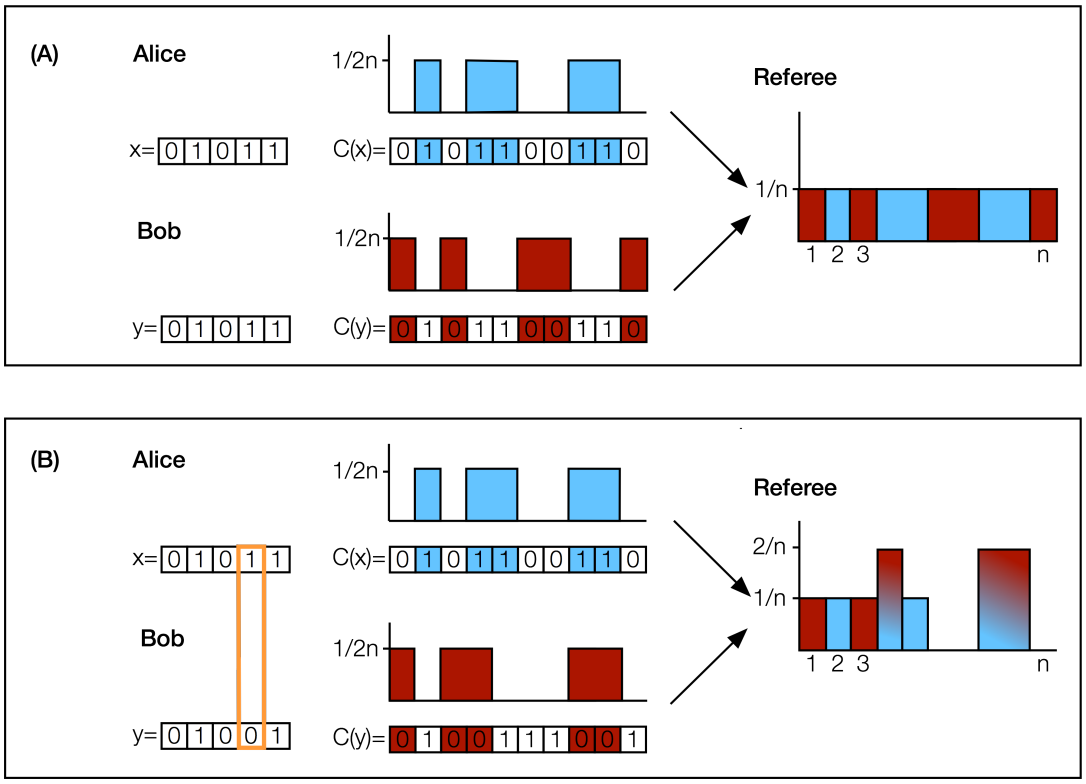

Figure 3.2: The reduction from equality in the SMP model to uniformity testing of distributions. In (A) we see that the uniform distribution is obtained when $x = y$, whereas in (B) we see that when $x \neq y$, we obtain a distribution that is "far" from uniform.

The reduction is as follows. Assume there exists a uniformity tester with sample complexity s. Each of the players encodes its input string via a balanced asymptotically good code C (that is, $C\colon \{0,1\}^k \to \{0,1\}^n$ is an error-correcting code with constant rate and relative distance $\delta = \Omega(1)$, which satisfies the property that each codeword of C contains the same number of 0's and 1's). Denote by $A \subset [n]$ the locations in which $C(x)$ takes the value 1 (i.e., $A = \{\, i \in [n] : \; C(x)_i = 1 \,\}$), and denote by $B \subset [n]$ the locations in which $C(y)$ takes the value 0 (i.e., $B = \{\, i \in [n] : \; C(y)_i = 0 \,\}$). Alice and Bob each send $O(s)$ uniformly distributed samples from A and B, respectively. Finally, the referee invokes the uniformity tester with respect

to the distribution $\mathbf{q} = (\mathbf{u}_A + \mathbf{u}_B)/2$, emulating each draw from $\mathbf{q}$ by tossing a random coin and deciding accordingly whether to use a sample by Alice or Bob. See Fig. 3.2.

The idea is that if $x = y$, then $C(x) = C(y)$, and so A and B are a *partition* of the set $[n]$. Furthermore, since $|C(x)| = |C(y)| = n/2$, this is a equipartition. Now, since Alice and Bob send uniform samples from an equipartition of $[n]$, the distribution $\mathbf{q}$ that the referee emulates is in fact the uniform distribution over $[n]$, and so the uniformity tester will accept. On the other hand, if $x \neq y$, then $C(x)$ and $C(y)$ disagree on a constant fraction of the domain. Thus, A and B intersect on $\delta/2$ elements, as well as do not cover $\delta/2$. Therefore $\mathbf{q}$ is uniform on a $(1-\delta)$-fraction of the domain, unsupported on a $(\delta/2)$-fraction of the domain, and has "double" weight $2/n$ on the remaining $(\delta/2)$-fraction. In particular, since $\delta = \Omega(1)$, the emulated distribution $\mathbf{q}$ is $\Omega(1)$-far (in ℓ_1-distance) from uniform, and it will be rejected by the uniformity tester.

As each sample sent by either Alice or Bob was encoded with $O(\log n)$ bits, the above constitutes an SMP protocol for equality with communication complexity $O(s \log(n))$. Yet it is well known [136] that the players must communicate $\Omega(\sqrt{k})$ bits to solve this problem (see Section 3.2.3), and so we deduce that $s = \Omega(\sqrt{k}/\log(n)) = \tilde{\Omega}(\sqrt{n})$.

3.2.2.2 Revisiting Distribution Identity Testing: A New Lower Bound

Next, consider the problem of testing whether a distribution $\mathbf{q} \in \Delta([n])$ is identical to a fixed distribution $\mathbf{p}$, provided as a (massive) parameter; that is, given a full description of $\mathbf{p} \in \Delta([n])$, we ask how many independent samples from $\mathbf{q}$ are needed to decide whether $\mathbf{q} = \mathbf{p}$, or whether $\mathbf{q}$ is ε-far in ℓ_1-distance from $\mathbf{p}$. As mentioned earlier, Valiant and Valiant [169] established both upper and lower bounds on this problem, involving the $\ell_{2/3}$-quasinorm of $\mathbf{p}$. We revisit this question, and show different – and more interpretable – upper and lower bounds. First, by applying our new communication complexity methodology to the distribution identity problem, we obtain a simple lower bound expressed in terms of a new parameter, which is closely related to the *effective support size* of $\mathbf{p}$.

Consider any fixed $\mathbf{p} \in \Delta([n])$. As a first idea, it is tempting to reduce equality in the SMP model to testing identity to $\mathbf{p}$ by following the uniformity reduction described in Section 3.2.2.1, only instead of having Alice and Bob send *uniform* samples from A and B, respectively, we have them send samples from $\mathbf{p}$ *conditioned* on membership in A and B respectively. That is, as before Alice and Bob encode their inputs x and y via a balanced, asymptotically good code C to obtain the sets $A = \{\, i \in [n] : C(x)_i = 1 \,\}$ and $B = \{\, i \in [n] : C(y)_i = 0 \,\}$, which partition $[n]$ if $x = y$, and intersect on $\Omega(n)$ elements (as well as fail to cover $\Omega(n)$ elements of $[n]$) if $x \neq y$. Only now, Alice sends samples independently drawn from $\mathbf{p}|_A$, i.e., $\mathbf{p}$ conditioned on the samples belonging to A, and Bob sends samples independently drawn from $\mathbf{p}|_B$, i.e., $\mathbf{p}$ conditioned on the samples belonging to B; and the referee emulates the distribution $\mathbf{q} = (\mathbf{p}|_A + \mathbf{p}|_B)/2$.

However, two problems arise in the foregoing approach. The first is that while indeed when $x = y$ the reduction induces an equipartition A, B of the domain, the resulting weights $\mathbf{p}(A)$ and $\mathbf{p}(B)$ in the mixture may still be dramatically different, in which case the referee will need much more samples from one of the

parties to emulate $\mathbf{p}$. The second is a bit more subtle, and has to do with the fact that the properties of this partitioning are with respect to the *size* of the symmetric difference $A \Delta B$, while really we are concerned about its *mass* under the emulated distribution $\mathbf{q}$ (and although both are proportional to each other in the case of the uniform distribution, for general $\mathbf{p}$ we have no such guarantee). Namely, when $x \neq y$ the domain elements which are responsible for the distance from $\mathbf{p}$ (that is, the elements which are covered by both parties ($A \cap B$) and by neither of the parties ($[n] \setminus (A \cup B)$) may only have a small mass according to $\mathbf{p}$, and thus the emulated distribution $\mathbf{q}$ will not be sufficiently far from $\mathbf{p}$. A natural attempt to address these two problems would be to preprocess $\mathbf{p}$ by discarding its light elements, focusing only on the part of the domain where $\mathbf{p}$ puts enough mass pointwise; yet this approach can also be shown to fail, as in this case the reduction may still not generate enough distance.[14]

Instead, we take a different route. The key idea is to consider a new type of codes, which we call $\mathbf{p}$-*weighted codes*, which will allow us to circumvent the second obstacle. These are code whose distance guarantee is weighted according to the distribution $\mathbf{p}$; that is, instead of requiring that every two codewords c, c' in a code C satisfy $\mathrm{dist}(x, y) \stackrel{\text{def}}{=} \sum_{i=1}^{n} |x_i - y_i| \geq \delta$, we consider a code $C_p \colon \{0,1\}^k \to \{0,1\}^n$ such that every $c, c' \in C_p$ satisfy

$$\mathrm{dist}_{\mathbf{p}}(x, y) \stackrel{\text{def}}{=} \sum_{i=1}^{n} \mathbf{p}(i) \cdot |x_i - y_i| \geq \delta.$$

Furthermore, to handle the first issue, we adapt the "balance" property accordingly, requiring that each codeword be balanced according to $\mathbf{p}$, that is, every $c \in C_p$ satisfies $\sum_{i=1}^{n} \mathbf{p}(i) \cdot c_i = 1/2$.

It is straightforward to see that if we invoke the above reduction while letting the parties encode their inputs via a balance $\mathbf{p}$-weighted code C_p, then both of the aforementioned problems are resolved; that is, by the $\mathbf{p}$-balance property the weights $\mathbf{p}(A)$ and $\mathbf{p}(B)$ are equal, and by the $\mathbf{p}$-distance of C_p we obtain that for $x \neq y$ the distribution $\mathbf{q} = (\mathbf{p}_A + \mathbf{p}_B)/2$ is $\Omega(1)$-far from $\mathbf{p}$. Hence we obtain a lower bound of $\Omega\left(\sqrt{k}/\log(n)\right)$ on the query complexity of testing identity to $\mathbf{p}$. To complete the argument, it remains to construct such codes, and determine what the best rate k/n that can be obtained by $\mathbf{p}$-weighted codes is.

3.2.2.3 Detour: $\mathbf{p}$-weighted Codes, Peetre's K-functional, and beating the CLT

The discussion of previous section left us with the task of constructing high-rate $\mathbf{p}$-weighted codes. Note that unlike standard (uniformly weighted) codes, for which we can easily obtain constant rate, there exist some $\mathbf{p}$'s for which high rate is impossible (for example, if $\mathbf{p} \in \Delta([n])$ is only supported on one element, we can only obtain rate $1/n$). In particular, by the sphere packing bound, every $\mathbf{p}$-weighted code $C \colon \{0,1\}^k \to \{0,1\}^n$ with distance δ must satisfy

$$\underbrace{2^k}_{\#\text{codewords}} \leq \frac{2^n}{\mathrm{Vol}_{\mathbb{F}_2^n, \mathrm{dist}_{\mathbf{p}}}(\delta/2)},$$

[14] In more detail, this approach would consider the distribution $\mathbf{p}'$ obtained by iteratively removing the lightest elements of $\mathbf{p}$ until a total of ε probability mass was removed. This way, every element i in the support of $\mathbf{p}'$ is guaranteed to have mass $\mathbf{p}'_i \geq \varepsilon/n$: this implies that the weights $\mathbf{p}'(A)$ and $\mathbf{p}'(B)$ are proportional, and that each element that is either covered by both parties or not covered at all will contribute ε/n to the distance from $\mathbf{p}'$. However, the total distance of $\mathbf{q}$ from $\mathbf{p}$ would only be $\Omega(|\operatorname{supp}(\mathbf{p}')| \cdot \varepsilon/n)$; and this only suffices if $\mathbf{p}$ and $\mathbf{p}'$ have comparable support size, i.e. $|\operatorname{supp}(\mathbf{p})| = O(|\operatorname{supp}(\mathbf{p}')|)$.

where $\mathrm{Vol}_{\mathbb{F}_2^n,\mathrm{dist}_{\mathbf{p}}}(r)$ is the volume of the $\mathbf{p}$-ball of radius r in the n-dimensional hypercube, given by

$$\mathrm{Vol}_{\mathbb{F}_2^n,\mathrm{dist}_{\mathbf{p}}}(r) \stackrel{\text{def}}{=} \left| \left\{ w \in \mathbb{F}_2^n : \sum_{i=1}^{n} \mathbf{p}_i \cdot w_i \leq r \right\} \right|.$$

Hence, we must have $k \leq n - \log \mathrm{Vol}_{\mathbb{F}_2^n,\mathrm{dist}_{\mathbf{p}}}(\delta/2)$.

In Section 3.2.6.1 we show that there exist (roughly) balanced $\mathbf{p}$-weighted codes with nearly-optimal rate,[15] and so it remains to determine the volume of the $\mathbf{p}$-ball of radius ε in the n-dimensional hypercube, where recall that ε is the proximity parameter of the test. To this end, it will be convenient to represent this quantity as a concentration inequality of sums of weighted Rademacher random variables, as follows

$$\mathrm{Vol}_{\mathbb{F}_2^n,\mathrm{dist}_{\mathbf{p}}}(\varepsilon) = 2^n \Pr_{Y \sim \{0,1\}^n}\left[\sum_{i=1}^{n} \mathbf{p}_i Y_i \leq \varepsilon\right] = 2^n \Pr_{X \sim \{-1,1\}^n}\left[\sum_{i=1}^{n} \mathbf{p}_i X_i \geq 1 - 2\varepsilon\right]. \tag{3.1}$$

Applying standard tail bounds derived from the central limit theorem (CLT), we have that

$$\Pr_{X \sim \{-1,1\}^n}\left[\sum_{i=1}^{n} \mathbf{p}_i X_i \geq 1 - 2\varepsilon\right] \leq e^{\frac{-(1-2\varepsilon)^2}{2\|\mathbf{p}\|_2^2}}, \tag{3.2}$$

and so we can obtain a $\mathbf{p}$-weighted code $C_p \colon \{0,1\}^k \to \{0,1\}^n$ with dimension $k = O(1/\|\mathbf{p}\|_2^2)$, which in turn, by the reduction described in Section 3.2.2.2, implies a lower bound of $\Omega(1/(\|\mathbf{p}\|_2 \cdot \log(n)))$ on the complexity of testing identity to $\mathbf{p}$.

Unfortunately, the above lower bound is not as strong as hoped, and in particular, far weaker than the $\|\mathbf{p}_{-\varepsilon}^{-\max}\|_{2/3}$ bound of [169].[16] Indeed, it turns out that the CLT-based bound in Eq. (3.2) is only tight for distributions satisfying $\|\mathbf{p}\|_\infty = O(\|\mathbf{p}\|_2^2)$, and is in general too crude for our purposes. Instead, we look for stronger concentration of measure inequalities that "beat" the CLT. To this end, we shall use powerful tools from the theory of interpolation spaces. Specifically, we consider Peetre's *K-functional* between ℓ_1 and ℓ_2 spaces. Loosely speaking, this is the operator defined for $t > 0$ by

$$\kappa_{\mathbf{p}}(t) = \inf_{\mathbf{p}'+\mathbf{p}''=\mathbf{p}} \|\mathbf{p}'\|_1 + t\|\mathbf{p}''\|_2.$$[17]

This K-functional can be thought of as an interpolation norm between the ℓ_1 and ℓ_2 norms of the distribution $\mathbf{p}$ (and accordingly, for any fixed t it defines a norm on the space $\ell_1 + \ell_2$). In particular, note

[15] We remark that since these codes are not perfectly $\mathbf{p}$-balanced, a minor modification to the reduction needs to be done. See Section 3.2.6.1 for details.

[16] For example, fix $\alpha \in (0,1)$, and consider the distribution $\mathbf{p} \in \Delta([n])$ in which $n/2$ elements are of mass $1/n$, and $n^\alpha/2$ elements are of mass $1/n^\alpha$. It is straightforward to verify that $\|\mathbf{p}\|_2^{-1} = \Theta\left((\sqrt{n})^\alpha\right)$, whereas $\|\mathbf{p}\|_{2/3} = \Theta\left(\sqrt{n}\right)$. (Intuitively, this is because the ℓ_2-norm is mostly determined by the few heavy elements, whereas the $\ell_{2/3}$-quasinorm is mostly determined by the numerous light elements.)

[17] Interestingly, Holmstedt [113] showed that the infimum is *approximately* obtained by partitioning $\mathbf{p} = (\mathbf{p}', \mathbf{p}'')$ such that $\mathbf{p}'$ consists of the heaviest t^2 coordinates of $\mathbf{p}$ and $\mathbf{p}''$ consists of the rest (for more detail, see Proposition 3.2.12).

that for large values of t the function $\kappa_{\mathbf{p}}(t)$ is close to $\|\mathbf{p}\|_1$, whereas for small values of t it will behave like $t\|\mathbf{p}\|_2$.

The foregoing connection is due to Montgomery-Smith [133], who established the following concentration of measure inequality for weighted sums of Rademacher random variables,

$$\Pr\left[\sum_{i=1}^{n} \mathbf{p}_i X_i \geq \kappa_{\mathbf{p}}(t)\right] \leq e^{-\frac{t^2}{2}}. \tag{3.3}$$

Furthermore, he proved that this concentration bound is essentially tight (see Section 3.2.5 for a precise statement). Plugging (3.3) into (3.1), we obtain a lower bound of $\Omega(\kappa_{\mathbf{p}}^{-1}(1-2\varepsilon)/\log(n))$ on the complexity of testing identity to $\mathbf{p}$.

To understand and complement this result, we describe in the next subsection a nearly tight upper bound for this problem, also expressed in terms of this K-functional; implying that this unexpected connection is in fact not a coincidence, but instead capturing an intrinsic aspect of the identity testing question. We also give a natural interpretation of this bound, showing that the size of the *effective support* of $\mathbf{p}$ (roughly, the number of supported elements that constitute the vast majority of the mass of $\mathbf{p}$) is a good proxy for this parameter $\kappa_{\mathbf{p}}^{-1}(1-2\varepsilon)$ – and thus for the complexity of testing identity to $\mathbf{p}$.

3.2.2.4 Using the Q-norm Proxy to Obtain an Upper Bound

To the end of obtaining an upper bound on the sample complexity of testing identity to $\mathbf{p}$, in terms of the K-functional, it will actually be convenient to look at a related quantity, known as the Q-*norm* [133]. At a high-level, the Q-norm of a distribution $\mathbf{p}$, for a given parameter $T \in \mathbb{N}$, is the maximum one can reach by partitioning the domain of $\mathbf{p}$ into T sets and taking the sum of the ℓ_2 norms of these T subvectors. That is

$$\|\mathbf{p}\|_{Q(T)} \stackrel{\text{def}}{=} \sup\left\{\sum_{j=1}^{T}\left(\sum_{i\in A_j}\mathbf{p}_i^2\right)^{1/2} : (A_j)_{1\leq j\leq T} \text{ partition of } \mathbb{N}\right\}.$$

Astashkin [11], following up Montgomery-Smith [133], showed that the Q-norm constitutes a good approximation of K-functional, by proving that

$$\|\mathbf{p}\|_{Q(t^2)} \leq \kappa_{\mathbf{p}}(t) \leq \sqrt{2}\|\mathbf{p}\|_{Q(t^2)}.$$

In Section 3.2.5 we further generalize this claim and show it is possible to get a tradeoff in the upper bound; specifically, we prove that $\kappa_{\mathbf{p}}(t) \leq \|\mathbf{p}\|_{Q(2t^2)}$. Thus, it suffices to prove an upper bound on distribution identity testing in terms of the Q-norm.

From an algorithmic point of view, it is not immediately clear that switching to this Q-norm is of any help. However, we will argue that this value captures – in a very quantitative sense – the notion of the *sparsity* of $\mathbf{p}$. As a first step, observe that if $\|\mathbf{p}\|_{Q(T)} = 1$, then the distribution $\mathbf{p}$ is supported on at most T

elements. To see this, denote by $\mathbf{p}_{A_j}$ the restriction of the sequence $\mathbf{p}$ to the indices in A_j, and note that if $\|\mathbf{p}\|_{Q(T)} \stackrel{\text{def}}{=} \sum_{j=1}^{T} \|\mathbf{p}_{A_j}\|_2 = 1$, then by the monotonicity of ℓ_p norms and since $\sum_{j=1}^{T} \|\mathbf{p}_{A_j}\|_1 = \|\mathbf{p}\|_1 = 1$ we have that

$$\sum_{j=1}^{T} (\underbrace{\|\mathbf{p}_{A_j}\|_1 - \|\mathbf{p}_{A_j}\|_2}_{\geq 0}) = 0,$$

which implies that $\|\mathbf{p}_{A_j}\|_1 = \|\mathbf{p}_{A_j}\|_2$ for all $j \in [T]$.

Now, it turns out that it is possible to obtain a *robust* version of the foregoing observation, yielding a sparsity lemma that, roughly speaking, shows thats if $\|\mathbf{p}\|_{Q(T)} \geq 1 - \varepsilon$, then $1 - O(\varepsilon)$ of the mass of $\mathbf{p}$ is concentrated on T elements: in this case we say that $\mathbf{p}$ has $O(\varepsilon)$-*effective support* of size T. (See Lemma 3.2.28 for precise statement of the sparsity lemma.)

This property of the Q-norm suggests the following natural test for identity to a distribution $\mathbf{p}$: Simply fix T such that $\|\mathbf{p}\|_{Q(T)} = 1 - \varepsilon$, and apply one of the standard procedures for testing identity to a distribution with support size T, which require $O(\sqrt{T})$ samples. But by the previous discussion, we have $\|\mathbf{p}\|_{Q(2t^2)} \geq \kappa_{\mathbf{p}}(t)$, so that setting $T = 2t^2$ for the "right" choice of $t = \kappa_{\mathbf{p}}^{-1}(1 - 2\varepsilon)$ will translate to an $O(t)$ upper bound – which is what we were aiming for.

3.2.3 The Methodology: From Communication Complexity to Distribution Testing

In this section we adapt the methodology for proving property testing lower bounds via reductions from communication complexity, due to Blais, Brody, and Matulef [33], to the setting of distribution testing. As observed in [33, 40], to prove lower bounds on the query complexity of *non-adaptive* testers it suffices to reduce from one-sided communication complexity. We show that for distribution testers (which are inherently non-adaptive), it suffices to reduce from the more restricted communication complexity model of *private-coin* simultaneous message passing (SMP).

Recall that a private-coin SMP protocol for a communication complexity predicate $f\colon \{0,1\}^k \times \{0,1\}^k \to \{0,1\}$ consists of three computationally unbounded parties: Two players (commonly referred to as Alice and Bob), and a Referee. Alice and Bob receive inputs $x, y \in \{0,1\}^k$. Each of the players simultaneously (and independently) sends a message to the referee, based on its input and (private) randomness. The referee is then required to successfully compute $f(x, y)$ with probability at least $2/3$, using its private randomness and the messages received from Alice and Bob. The communication complexity of an SMP protocol is the total number of bits sent by Alice and Bob. The private-coin SMP complexity of f, denoted $\mathsf{SMP}(f)$, is the minimum communication complexity of all SMP protocols that solve f with probability at least $2/3$.

Generally, to reduce an SMP problem f to ε-testing a distribution property Π, Alice and Bob can send messages $m_A(x, r_A, \varepsilon)$ and $m_B(y, r_B, \varepsilon)$ (respectively) to the Referee, where r_A and r_B are the private random strings of Alice and Bob. Subsequently, the Referee uses the messages $m_A(x, r_A, \varepsilon)$ and $m_A(y, r_B, \varepsilon)$, as well as its own private randomness, to feed the property tester samples from a distribution $\mathbf{p}$ that satisfies the following conditions: (1) *completeness*: if $f(x, y) = 1$, then $\mathbf{p} \in \Pi$; and (2) *soundness*: if $f(x, y) = 0$,

then $\mathbf{p}$ is ε-far from Π in ℓ_1-distance.

We shall focus on a special type of the foregoing reductions, which is particularly convenient to work with and suffices for all of the our lower bounds. Loosely speaking, in these reductions Alice and Bob both send the prover samples from sub-distributions that can be combined by the Referee to obtain samples from a distribution that satisfies the completeness and soundness conditions. The following lemma gives a framework for proving lower bounds based on such reductions.

Lemma 3.2.3. *Let $\varepsilon > 0$, and let Ω be a finite domain of cardinality n. Fix a property $\Pi \subseteq \Delta(\Omega)$ and a communication complexity predicate $f\colon \{0,1\}^k \times \{0,1\}^k \to \{0,1\}$. Suppose that there exists a mapping $\mathbf{p}\colon \{0,1\}^k \times \{0,1\}^k \to \Delta(\Omega)$ that satisfies the following conditions.*

1. Decomposability: *For every $x, y \in \{0,1\}^k$, there exist constants $\alpha = \alpha(x), \beta = \beta(y) \in [0,1]$ and distributions $\mathbf{p}_A(x), \mathbf{p}_B(y)$ such that*

$$p(x,y) = \frac{\alpha}{\alpha+\beta} \cdot \mathbf{p}_A(x) + \frac{\beta}{\alpha+\beta} \cdot \mathbf{p}_B(y)$$

   and α, β can each be encoded with $O(\log n)$ bits.

2. Completeness: *For every $(x,y) \in f^{-1}(1)$, it holds that $\mathbf{p}(x,y) \in \Pi$.*

3. Soundness: *For every $(x,y) \in f^{-1}(0)$, it holds that $\mathbf{p}(x,y)$ is ε-far from Π in ℓ_1 distance.*

Then, every ε-tester for Π needs $\Omega\Big(\frac{\mathsf{SMP}(f)}{\log(n)}\Big)$ samples.

Proof. Supose there exists an ε-tester for Π with sample complexity s'; assume without loss of generality that the soundness of the foregoing tester is $5/6$, at the cost of increasing the query complexity to $s = O(s')$. Let $x, y \in \{0,1\}^k$ be the inputs of Alice and Bob (respectively) for the SMP problem. Alice computes the distribution $\mathbf{p}_A(x)$ and the "decomposability parameter" $\alpha = \alpha(x)$ and sends $6s$ independent samples from $\mathbf{p}_A(x)$, as well as the parameter α. Analogously, Bob computes $\mathbf{p}_B(y)$ and its parameter $\beta = \beta(y)$, and sends $6s$ independent samples from $\mathbf{p}_B(y)$ as well as the parameter β. Subsequently, the referee generates a sequence of $\mathbf{q}$ independent samples from $\mathbf{p}(x,y)$, where each sample is drawn as follows: with probability $\frac{\alpha}{\alpha+\beta}$ use a (fresh) sample from Alice's samples, and with probability $1 - \frac{\alpha}{\alpha+\beta}$ use a (fresh) sample from Bob's samples. Finally the referee feeds the generated samples to the ε-tester for Π.

By Markov's inequality, the above procedure indeed allows the referee to retrieve, with probability at least $1 - \frac{\alpha s}{6s} \geq \frac{5}{6}$, at least s independent samples from the distribution $\frac{\alpha}{\alpha+\beta} \cdot \mathbf{p}_A(x) + \frac{\beta}{\alpha+\beta} \cdot \mathbf{p}_B(y)$, which equals to $\mathbf{p}(x,y)$, by the decomposability condition. If $(x,y) = f^{-1}(1)$, then by the completeness condition $\mathbf{p}(x,y) \in \Pi$, and so the ε-tester for Π is successful with probability at least $\frac{5}{6} \cdot \frac{5}{6}$. Similarly, if $(x,y) = f^{-1}(0)$, then by the soundness condition $\mathbf{p}(x,y)$ is ε-far from Π, and so the ε-tester for Π is successful with probability at least $\frac{5}{6} \cdot \frac{5}{6}$. Finally, note that since each one of the samples provided by Alice and Bob requires sending $\log n$ bits, the total communication complexity of the protocol is $12s \log n + O(\log n)$ (the last term from the cost of sending α, β), hence $s' = \Omega\Big(\frac{\mathsf{SMP}(f)}{\log(n)}\Big)$. □

We conclude this section by stating a well-known SMP lower bound on the equality problem. Let $\mathrm{EQ}_k \colon \{0,1\}^k \times \{0,1\}^k \to \{0,1\}$ be the equality predicate, i.e., $\mathrm{EQ}_k(x,y) = 1$ if and only if $x = y$. In this work, we shall frequently use the following (tight) lower bound on the EQ_k predicate:

Theorem 3.2.4 (Newman and Szegedy [136])**.** *For every $k \in \mathbb{N}$ it holds that* $\mathsf{SMP}(\mathrm{EQ}_k) = \Omega\left(\sqrt{k}\right)$.

3.2.4 The Basic Reduction: The Case of Uniformity

Theorem 3.2.5. *For any $\varepsilon \in (0, 1/2)$ and finite domain Ω, testing that $\mathbf{p} \in \Delta(\Omega)$ is uniform, with respect to proximity parameter ε, requires $\tilde{\Omega}(\sqrt{n})$ samples, where $n = |\Omega|$.*

Proof. Assume there exists a $\mathbf{q}$-query ε-tester for the uniform distribution, with error probability $1/6$. For a sufficiently large $k \in \mathbb{N}$, let $C \colon \{0,1\}^k \to \{0,1\}^n$ be a balanced code as promised by Proposition 1.5.1 with distance ε. Namely, there exists an absolute constant $\rho > 0$ such that

(i) $|C(x)| = \frac{n}{2}$ for all $x \in \{0,1\}^k$;

(ii) $\mathrm{dist}(C(x), C(y)) > \varepsilon$ for all distinct $x, y \in \{0,1\}^k$;

(iii) $\frac{k}{n} \geq \rho$.

Given their respective inputs $x, y \in \{0,1\}^k$ from EQ_k, Alice and Bob separately create inputs $(C(x), C(y)) \in \{0,1\}^n \times \{0,1\}^n$, and the corresponding sets $A \stackrel{\text{def}}{=} \{\, i \in [n] \,:\, C(x)_i = 1 \,\}$, $B \stackrel{\text{def}}{=} \{\, i \in [n] \,:\, C(y)_i = 0 \,\}$. We then invoke the general reduction of Lemma 3.2.3 as follows: we set $\alpha = \beta = \frac{1}{2}$, and $\mathbf{p}_A(x) \in \Delta(\Omega)$ (respectively $\mathbf{p}_B(y) \in \Delta(\Omega)$) to be the uniform distribution on the set A (respectively B). It is clear that the decomposability condition of the lemma is satisfied for $\mathbf{p}(x,y) = \frac{\alpha}{\alpha+\beta} \cdot \mathbf{p}_A(x) + \frac{\beta}{\alpha+\beta} \cdot \mathbf{p}_B(y) = \frac{1}{2}(\mathbf{p}_A(x) + \mathbf{p}_B(y))$; we thus turn to the second and third conditions.

Completeness. If $(x,y) \in \mathrm{EQ}_k^{-1}(1)$, then $C(x) = C(y)$ and $A = [n] \setminus B$. This implies that $\mathbf{p}(x,y)$ is indeed the uniform distribution on $[n]$, as desired.

Soundness. If $(x,y) \in \mathrm{EQ}_k^{-1}(0)$, then $\mathrm{dist}(C(x), C(y)) > \varepsilon$, and therefore $|A \triangle \bar{B}| > \varepsilon n$ by construction. Since $\mathbf{p}(x,y)$ assigns mass $2/n$ to each element in $A \cap B = A \setminus \bar{B}$, and mass 0 to any element in $\bar{A} \cap \bar{B} = \bar{B} \setminus A$, we have $\|\mathbf{p}(x,y) - u\|_1 = \frac{1}{n} \cdot |A \triangle \bar{B}| > \varepsilon$; that is, $\mathbf{p}(x,y)$ is ε-far from uniform.

The desired $\Omega\left(\frac{\sqrt{n}}{\log n}\right)$ lower bound then immediately follows from Lemma 3.2.3 and Theorem 3.2.4. □

3.2.4.1 Obtaining ε-Dependency

In this section, we explain how to generalize the reduction from the previous section to obtain some dependence (albeit non optimal) on the distance parameter ε in the lower bound. This generalization will rely on an extension of the methodology of Lemma 3.2.3: instead of having the referee define the distribution $\mathbf{p}(x,y)$ as a mixture of $\mathbf{p}_A(x)$ and $\mathbf{p}_B(y)$ (namely, $\mathbf{p}(x,y) = \frac{\alpha(x)}{\alpha(x)+\beta(y)}\mathbf{p}_A(x) + \frac{\beta(y)}{\alpha(x)+\beta(y)}\mathbf{p}_B(y)$), he will instead use a (random) combination function F_ε, function of ε and its private coins only. Given this function, which maps

a larger domain of size $m = \Theta\big(n/\varepsilon^2\big)$ to $[n]$, $\mathbf{p}(x,y)$ will be defined as the mixture

$$\mathbf{p}(x,y) = \frac{\alpha(x)}{\alpha(x)+\beta(y)}\mathbf{p}_A(x)\circ F_\varepsilon^{-1} + \frac{\beta(y)}{\alpha(x)+\beta(y)}\mathbf{p}_B(y)\circ F_\varepsilon^{-1}.$$

More simply, this allows Alice and Bob to send to the referee samples from their respective distributions on a much larger domain $m \gg n$; the referee, who has on its side chosen how to randomly partition this large domain into only n different "buckets," converts these draws from Alice and Bob into samples from the induced distributions on the n buckets, and takes a mixture of these two distributions instead. By choosing each bucket to chave size roughly $1/\varepsilon^2$, we expect this random "coarsening" of Alice and Bob's distributions to yield a distribution at distance only $\Omega(\varepsilon)$ from uniformity (instead of constant distance) in the no-case; but now letting us get a lower bound on the *original* support size m, i.e. $\tilde{\Omega}\Big(\sqrt{n/\varepsilon^2}\Big)$, instead of $\tilde{\Omega}(\sqrt{n})$ as before.

Theorem 3.2.6. *For any $\varepsilon \in (0,1/2)$ and finite domain Ω, testing that $\mathbf{p} \in \Delta(\Omega)$ is uniform, with respect to proximity parameter ε, requires $\tilde{\Omega}(\sqrt{n}/\varepsilon)$ samples, where $n = |\Omega|$.*

Proof of Theorem 3.2.6. We will reduce from EQ_k, where $k \in \mathbb{N}$ is again assumed big enough (in particular, with regard to $1/\varepsilon^2$). Alice and Bob act as in Section 3.2.4, separately creating $(a,b) = (C(x), C(y)) \in \{0,1\}^m \times \in \{0,1\}^m$ from their respective inputs $x, y \in \{0,1\}^k$ (where $C\colon \{0,1\}^k \to \{0,1\}^m$ is a balanced code with linear rate and distance $\delta \stackrel{\text{def}}{=} 1/3$). As before, they consider the sets $A \stackrel{\text{def}}{=} \{\, i \in [m] \,:\, C(x)_i = 1 \,\}$, $B \stackrel{\text{def}}{=} \{\, i \in [m] \,:\, C(y)_i = 0 \,\}$, set $\alpha = \beta = \frac{1}{2}$, and consider the distributions $\mathbf{p}_A(x), \mathbf{p}_B(y) \in \Delta([m])$ which are uniform respectively on A and B.

This is where we deviate from the proof of Theorem 3.2.5: indeed, setting $n \stackrel{\text{def}}{=} c\varepsilon^2 m$ (where $c > 0$ is an absolute constant determined later), the referee will combine the samples from $\mathbf{p}_A(x)$ and $\mathbf{p}_B(y)$ in a different way to emulate a distribution $\mathbf{p}(x,y) \in \Delta([n])$ – that is, with a much smaller support than that of $\mathbf{p}_A(x), \mathbf{p}_B(y)$ (instead of setting $\mathbf{p}(x,y)$ to be, as before, a mixture of the two).

To do so, the referee randomly partitions $[m]$ into n sets $B_1, \dots, B_n$ of equal size $r \stackrel{\text{def}}{=} |B_j| = \frac{m}{n} = \frac{1}{c\varepsilon^2}$, $j \in [n]$, by choosing a uniformly random equipartition of $[m]$. He then defines the distribution $\mathbf{p} = \mathbf{p}(x,y) \in \Delta([n])$ by $\mathbf{p}(j) = \Pr[\, i \in B_j \,]$ (where $i \in [m]$ is received from either Alice or Bob). Viewed differently, the random equipartition chosen by the referee induces a mapping $F_\varepsilon\colon [m] \to [n]$ such that $\big|F^{-1}(j)\big| = r$ for all $j \in [n]$; and, setting $\mathbf{p}'(x,y) = \frac{1}{2}(\mathbf{p}_A(x) + \mathbf{p}_B(y)) \in \Delta([m])$, we obtain $\mathbf{p}(x,y)$ as the *coarsening* of $\mathbf{p}'(x,y)$ defined as

$$\mathbf{p}(x,y)(j) = \sum_{i \in F_\varepsilon^{-1}(j)} p'(x,y)(i) = p'(x,y)(F_\varepsilon^{-1}(j)) = \frac{1}{2}\left(\mathbf{p}_A(x)(F_\varepsilon^{-1}(j)) + \mathbf{p}_B(y)(F_\varepsilon^{-1}(j))\right), \qquad j \in [n].$$

Note furthermore that each sample sent by Alice and Bob (who have no knowledge of the randomly chosen F_ε) can be encoded with $O(\log m) = O(\log \frac{n}{\varepsilon})$ bits.

We then turn to establish the analogue in this generalized reduction of the last two conditions of Lemma 3.2.3, i.e. the completeness and soundness. The former, formally stated below, will be an easy consequence of the

previous section.

Claim 3.2.7. *If $x = y$, then $\mathbf{p}(x, y)$ is uniform on $[n]$.*

Proof. As in the proof of Theorem 3.2.5, in this case the distribution $\mathbf{p}'(x, y) = \frac{1}{2}(\mathbf{p}_A(x) + \mathbf{p}_B(y)) \in \Delta([m])$ is uniform; since each "bucket" $B_j = F_\varepsilon^{-1}(j)$ has the same size, this implies that $\mathbf{p}(x, y)(j) = p'(x, y)(B_j) = \frac{1}{n}$ for all $j \in [n]$. □

Establishing the soundness, however, is not as straightforward:

Claim 3.2.8. *If $x \neq y$, then with probability at least $1/100$ (over the choice of the equipartition $(B_1, \ldots, B_n)$), $\mathbf{p}(x, y)$ is ε-far from uniform.*

Proof. Before delving into the proof, we provide a high-level idea of why this holds. Since the partition was chosen uniformly at random, on expectation each element $j \in [n]$ will have probability $\mathbb{E}[\mathbf{p}(x, y)(j)] = \mathbb{E}[\mathbf{p}'(x, y)(B_j)] = \frac{1}{n}$. However, since a constant fraction of elements $i \in [m]$ (before the random partition) has probability mass either 0 or $2/m$ (as in the proof of Theorem 3.2.5), and each bucket B_j contains $r = 1/(c\varepsilon^2)$ many elements chosen uniformly at random, we expect the fluctuations of $\mathbf{p}'(x, y)(B_j)$ around its expectation to be of the order of $\Omega(\sqrt{r}/m) = \Omega(\varepsilon/n)$ with constant probability, and summing over all j's this will give us the distance $\Omega(\varepsilon)$ we want.

To make this argument precise, we assume $x \neq y$, so that $A \triangle \bar{B} > \delta m$; and define $H \stackrel{\text{def}}{=} A \cap B, L \stackrel{\text{def}}{=} \bar{A} \cap \bar{B}$ (so that $|H| = |L| > \frac{\delta}{2}m$). For any $j \in [n]$, we then let the random variables $H^{(j)}, L^{(j)}$ be the number of "high" and "low" elements of $[m]$ in the bucket B_j, respectively:

$$H^{(j)} \stackrel{\text{def}}{=} |B_j \cap H|, \qquad L^{(j)} \stackrel{\text{def}}{=} |B_j \cap L|.$$

From the definition, we get that $\mathbf{p} = \mathbf{p}(x, y)$ satisfies $\mathbf{p}(j) = \frac{1}{m}\left(2H^{(j)} + (r - H^{(j)} - L^{(j)}\right) = \frac{r}{m} + \frac{H^{(j)} - L^{(j)}}{m}$ for $j \in [n]$. Furthermore, it is easy to see that $\mathbb{E}[\mathbf{p}(j)] = \frac{r}{m} = \frac{1}{n}$ for all $j \in [n]$, where the expectation is over the choice of the equipartition by the referee.

As previously discussed, we will analyze the deviation from this expectation; more precisely, we want to show that with good probability, a constant fraction of the j's will be such that $\mathbf{p}(j)$ deviates from $1/n$ by at least an additive $\Omega(\sqrt{r}/m) = \varepsilon/n$. This anticoncentration guarantee will be a consequence of the Paley–Zygmund inequality (Theorem 1.4.13) to $Z^{(j)} \stackrel{\text{def}}{=} (H^{(j)} - L^{(j)})^2 \geq 0$; in view of applying it, we need to analyze the first two moments of this random variable.

Lemma 3.2.9. *For any $j \in [n]$, we have the following. (i) $\mathbb{E}\left[(H^{(j)} - L^{(j)})^2\right] = \delta r \frac{m-r}{m-1}$, and (ii) $\mathbb{E}\left[(H^{(j)} - L^{(j)})^4\right] = 3(1 + o(1))\delta^2 r^2$.*

Proof. Fix any $j \in [n]$. We write for convenience X and Y for respectively $H^{(j)}$ and $L^{(j)}$. The distribution

of $(X, Y, r-(X-Y))$ is then a *multivariate hypergeometric distribution* [176] with 3 classes:

$$(X, Y, r-(X+Y)) \sim \text{MultivHypergeom}_3(\underbrace{(\tfrac{1}{2}\delta m, \tfrac{1}{2}\delta m, (1-\delta)m)}_{(K_1,K_2,K_3)}, m, r).$$

Conditioning on $U \stackrel{\text{def}}{=} X+Y$, we have that $\mathbb{E}[X \mid U]$ follows a hypergeometric distribution, specifically $\mathbb{E}[X \mid U] \sim \text{Hypergeom}(U, \frac{1}{2}\delta m, \delta m)$. Moreover, U itself is hypergeometrically distributed, with $U \sim \text{Hypergeom}(r, \delta m, m)$. We can thus write

$$\mathbb{E}\big[(X-Y)^2\big] = \mathbb{E}\big[\mathbb{E}\big[(X-Y)^2 \mid U\big]\big] = \mathbb{E}\big[\mathbb{E}\big[(2X-U)^2 \mid U\big]\big]$$

and

$$\mathbb{E}\big[(X-Y)^4\big] = \mathbb{E}\big[\mathbb{E}\big[(X-Y)^4 \mid U\big]\big] = \mathbb{E}\big[\mathbb{E}\big[(2X-U)^4 \mid U\big]\big].$$

By straightforward, yet tedious, calculations involving the computation of $\mathbb{E}\big[(2X-U)^2 \mid U\big]$ and $\mathbb{E}\big[(2X-U)^4 \mid U\big]$ (after expanding and using the known moments of the hypergeometric distribution),[18] we obtain

$$\begin{aligned}
\mathbb{E}\big[(X-Y)^2\big] &= \delta r \frac{m-r}{m-1} \underset{m\to\infty}{=} (1+o(1))\delta r \\
\mathbb{E}\big[(X-Y)^4\big] &= \frac{(\delta r(r-m)((-1+3\delta(m-1)-m)m + 6r^2(\frac{1}{2}\delta m - 1) - 6rm(\frac{1}{2}\delta m - 1)))}{(m-3)(m-2)(m-1)} \\
&\xrightarrow[m\to\infty]{} 3\delta^2 r^2 + (1-3\delta)\delta r = 3\delta^2 r^2
\end{aligned}$$

the last equality as $\delta = 1/3$. □

We can now apply the Paley–Zygmund inequality to $Z^{(j)}$. Doing so, we obtain that for $r \le \frac{m}{4}$ (with some slack), and any $\theta \in [0,1]$,

$$\Pr\left[\left|H^{(j)} - L^{(j)}\right| \ge \theta\sqrt{\frac{1}{2}\delta r}\right] \ge \Pr\left[\left|H^{(j)} - L^{(j)}\right| \ge \theta\sqrt{\delta r \frac{m-r}{m-1}}\right] \ge (1-\theta^2)^2 \frac{\mathbb{E}\big[(H^{(j)} - L^{(j)})^2\big]^2}{\mathbb{E}\big[(H^{(j)} - L^{(j)})^4\big]}.$$

By the lemma above, the RHS converges to $\frac{(1-\theta^2)^2}{3}$ when $m \to \infty$, and therefore is at least $\frac{(1-\theta^2)^2}{4}$ for m big enough. We set $\theta \stackrel{\text{def}}{=} 1/\sqrt{2}$ to obtain the following: there exists $M \ge 0$ such that

$$\Pr\left[\left|H^{(j)} - L^{(j)}\right| \ge \sqrt{\frac{\delta r}{4}}\right] \ge \frac{1}{16} \tag{3.4}$$

[18] One can also use a formal computation system, e.g. Mathematica:

```
Expectation[ Expectation[(2 X - U)^2, {X \[Distributed] HypergeometricDistribution[U, a*m, 2 a*m]}],
  {U \[Distributed] HypergeometricDistribution[r, 2*a*m, m]}]
Expectation[ Expectation[(2 X - U)^4, {X \[Distributed] HypergeometricDistribution[U, a*m, 2 a*m]}],
  {U \[Distributed] HypergeometricDistribution[r, 2*a*m, m]}]
```

for every $m \geq M$.

Eq. (3.4) implies that the number K of *good* indices $j \in [n]$ satisfying $\left|H^{(j)} - L^{(j)}\right| \geq \sqrt{\frac{\delta r}{4}}$ is on expectation at least $\frac{n}{16}$, and by an averaging argument[19] we get that $K \geq \frac{n}{20}$ with probability at least $\frac{1}{76} > \frac{1}{100}$.

Whenever this happens, the distance from $\mathbf{p}$ to uniform is at least

$$\sum_{j \text{ good}} \left|\mathbf{p}(j) - \frac{1}{n}\right| = \sum_{j \text{ good}} \frac{\left|H^{(j)} - L^{(j)}\right|}{m} \geq \frac{n}{20} \cdot \frac{\sqrt{\frac{\delta r}{4}}}{m} = \frac{\sqrt{\delta r}}{40}\frac{n}{m} = \frac{\sqrt{c}}{40\sqrt{3}}\varepsilon$$

and choosing $c \geq 4800$ so that $\frac{\sqrt{c}}{40\sqrt{3}} \geq 1$ yields the claim. □

From this lemma, we can complete the reduction: given a tester $\mathcal{T}$ for uniformity with query complexity $\mathbf{q}$, we first convert it by standard amplification into a tester $\mathcal{T}'$ with failure probability $\delta \stackrel{\text{def}}{=} 1/1000$ and sample complexity $O(q)$. The referee can provide samples from the distribution $\mathbf{p}(x,t)$, and on input ε:

- If $x = y$, then $\mathcal{T}'$ will return reject with probability at most $1/200$;
- If $x \neq y$, then $\mathcal{T}'$ will return reject with probability at least $199/200 \cdot 1/100 > 1/200$;

so repeating independently the protocol a constant (fixed in advance) number of times and taking a majority vote enables the referee to solve EQ_k with probability at least $2/3$. Since $\Omega\left(\sqrt{k}\right) = \Omega\left(\sqrt{n/\varepsilon^2}\right)$ bits of communication are required for this, and each sample sent by Alice or Bob to the referee only requires $\Theta\left(\log \frac{n}{\varepsilon}\right)$ bits, we get a lower bound of

$$\Omega\left(\frac{\sqrt{n}}{\varepsilon \log \frac{n}{\varepsilon}}\right) = \tilde{\Omega}\left(\frac{\sqrt{n}}{\varepsilon}\right)$$

on the sample complexity of $\mathcal{T}'$, and therefore of $\mathcal{T}$. □

3.2.5 The K-Functional: An Unexpected Journey

A quantity that will play a major role in our results is the *K-functional between ℓ_1 and ℓ_2*, a specific case of the key operator in interpolation theory introduced by Peetre [141]. We start by recalling below the definition and some of its properties, before establishing (for our particular setting) results that will be crucial to us. (For more on the K-functional and its use in functional analysis, the reader is referred to [25] and [11].)

Definition 3.2.10 (K-functional)**.** Fix any two Banach spaces $(X_0, \|\cdot\|_0), (X_1, \|\cdot\|_1)$. The *$K$-functional* between X_0 and X_1 is the function $K_{X_0,X_1} \colon (X_0 + X_1) \times (0,\infty) \to [0,\infty)$ defined by

$$K_{X_0,X_1}(x,t) \stackrel{\text{def}}{=} \inf_{\substack{(x_0,x_1) \in X_0 \times X_1 \\ x_0 + x_1 = x}} \|x_0\|_0 + t\|x_1\|_1.$$

[19] Applying Markov's inequality: $\Pr\left[K < \frac{n}{20}\right] = \Pr\left[n - K > \frac{19n}{20}\right] \leq \frac{n - \mathbb{E}[K]}{19n/20} \leq \frac{15/16}{19/20} = \frac{75}{76}$.

For $a \in \ell_1 + \ell_2$, we denote by κ_a the function $t \mapsto K_{\ell_1,\ell_2}(a,t)$.

In other terms, as t varies the quantity $\kappa_a(t)$ interpolates between the ℓ_1 and ℓ_2 norms of the sequence a (and accordingly, for any fixed t it defines a norm on $\ell_1 + \ell_2$). In particular, note that for large values of t the function $\kappa_a(t)$ is close to $\|x\|_1$, whereas for small values of t the function $\kappa_a(t)$ is close to $t\|x\|_2$ (see Corollary 3.2.14). We henceforth focus on the case of K_{ℓ_1,ℓ_2}, although some of the results mentioned hold for the general setting of arbitrary Banach X_0, X_1.

Proposition 3.2.11 ([25, Proposition 1.2])**.** *For any $a \in \ell_1 + \ell_2$, κ_a is continuous, increasing, and concave. Moreover, the function $t \in (0,1) \mapsto \frac{\kappa_a}{t}$ is decreasing.*

Although no closed-form expression is known for κ_a, it will be necessary for us to understand its behavior, and therefore seek good upper and lower bounds on its value. We start with the following inequality, due to Holmstedt [113], which, loosely speaking, shows that the infimum in the definition of $\kappa_a(t)$ is *roughly* obtained by partitioning $a = (a_1, a_2)$ such that a_1 consists of heaviest t^2 coordinates of a, and a_2 consists of the rest.

Proposition 3.2.12 ([11, Proposition 2.2], after [113, Theorem 4.2])**.** *For any $a \in \ell_2$ and $t > 0$,*

$$\frac{1}{4}\left(\sum_{i=1}^{\lfloor t^2 \rfloor} a_i^* + t\left(\sum_{i=\lfloor t^2 \rfloor+1}^{\infty} a_i^{*2}\right)^{\frac{1}{2}}\right) \leq \kappa_a(t) \leq \sum_{i=1}^{\lfloor t^2 \rfloor} a_i^* + t\left(\sum_{i=\lfloor t^2 \rfloor+1}^{\infty} a_i^{*2}\right)^{\frac{1}{2}} \tag{3.5}$$

where a^* is a non-increasing permutation of the sequence $(|a_i|)_{i\in\mathbb{N}}$.

(We remark that for our purposes, this constant factor gap between left-hand and right-hand side is not innocuous, as we will later need to study te behavior of the *inverse* of the function κ_a.)

Incomparable bounds on κ_a were obtained [133], relating it to a different quantity, the "Q-norm," which we discuss and generalize next.

3.2.5.1 Approximating the K-Functional by the Q-norm

Loosely speaking, the Q-norm of a vector a (for a given parameter T) is a *mixed* ℓ_1/ℓ_2 norm: it is the maximum one can reach by partitioning the components of a into T sets, and taking the sum of the ℓ_2 norms of these T subvectors. Although not straightforward to interpret, this intuitively captures the notion of *sparsity* of a: indeed, if a is supported on k elements then its Q-norm becomes equal to the ℓ_1 norm for parameter $T \geq k$.

Proposition 3.2.13 ([11, Lemma 2.2], after [133, Lemma 2])**.** *For arbitrary $a \in \ell_2$ and $t \in \mathbb{N}$, define the norm*

$$\|a\|_{Q(t)} \stackrel{\text{def}}{=} \sup\left\{ \sum_{j=1}^{t}\left(\sum_{i\in A_j} a_i^2\right)^{1/2} \;:\; (A_j)_{1\leq j\leq t} \text{ partition of } \mathbb{N} \right\}.$$

Then, for any $a \in \ell_2$, and $t > 0$ such that $t^2 \in \mathbb{N}$, we have

$$\|a\|_{Q(t^2)} \leq \kappa_a(t) \leq \sqrt{2}\|a\|_{Q(t^2)}. \tag{3.6}$$

As we shall see shortly, one can generalize this result further, obtaining a tradeoff in the upper bound. Before turning to this extension in Lemma 3.2.15 and Lemma 3.2.18, we first state several other properties of the K-functional implied by the above:

Corollary 3.2.14. *For any $a \in \ell_2$,*

(i) $\kappa_a(t) = t\|a\|_2$ *for all* $t \in (0,1)$

(ii) $\lim_{t\to 0^+} \kappa_a(t) = 0$

(iii) $\frac{1}{4}\|a\|_1 \leq \lim_{t\to\infty} \kappa_a(t) \leq \|a\|_1$.

Moreover, for a supported on finitely many elements, it is the case that $\lim_{t\to\infty} \kappa_a(t) = \|a\|_1$.

Proof. The first two points follow by definition; turning to Item (iii), we first note the upper bound is a direct consequence of the definition of κ_a as an infimum (as, for all $t > 0$, $\kappa_a(t) \leq \|a\|_1$). (This itself ensures the limit as $t \to \infty$ exists by monotone convergence, as κ_a is a non-decreasing bounded function.) The lower bound follows from that of Proposition 3.2.12, which guarantees that for all $t > 0$ $\kappa_a(t) \geq \frac{1}{4}\sum_{i=1}^{\lfloor t^2 \rfloor} a_i^* \xrightarrow[t\to\infty]{} \frac{1}{4}\|a\|_1$. Finally, the last point can be obtained immediately from, e.g., the lower bound side of Proposition 3.2.13 and the upper bound given on Item (iii) above. □

Lemma 3.2.15. *For any $a \in \ell_2$ and t such that $t^2 \in \mathbb{N}$, we have*

$$\|a\|_{Q(t^2)} \leq \kappa_a(t) \leq \|a\|_{Q(2t^2)}. \tag{3.7}$$

Proof of Lemma 3.2.15. We follow and adapt the proof of [11, Lemma 2.2] (itself similar to that of [133, Lemma 2]). The first inequality is immediate: indeed, for any sequence $c \in \ell_2$, by the definition of $\|a\|_{Q(t^2)}$ and the monotonicity of the $\mathbf{p}$-norms, we have $\|c\|_{Q(t^2)} \leq \|c\|_1$; and by Cauchy–Schwarz, for any partition $(A_j)_{1\leq j\leq t^2}$ of $\mathbb{N}$,

$$\sum_{j=1}^{t^2}\left(\sum_{i\in A_j} c_i^2\right)^{1/2} \leq t\left(\sum_{j=1}^{t^2}\sum_{i\in A_j} c_i^2\right)^{1/2} = t\|c\|_2$$

and thus $\|c\|_{Q(t^2)} \leq t\|c\|_2$. This yields the lower bound, as

$$\kappa_a(t) = \inf_{\substack{a'+a''=a \\ a'\in\ell_1, a''\in\ell_2}} \|a'\|_1 + t\|a''\|_2 \geq \inf_{\substack{a'+a''=a \\ a'\in\ell_1, a''\in\ell_2}} \|a'\|_{Q(t^2)} + \|a''\|_{Q(t^2)} \geq \|a\|_{Q(t^2)}$$

by the triangle inequality.

We turn to the upper bound. As $\ell_2(\mathbb{R})$ is a symmetric space and $\kappa_a = \kappa_{|a|}$, without loss of generality, we can assume that $(a_k)_{k\in\mathbb{N}}$ is non-negative and monotone non-increasing, i.e. $a_1 \geq a_2 \geq \cdots \geq a_k \geq \ldots$. We

will rely on the characterization of κ_a as

$$\kappa_a(t) = \sup\left\{ \sum_{k=1}^{\infty} a_k b_k \ : \ b \in \ell_2, \max(\|b\|_\infty, t^{-1}\|b\|_2) \le 1 \right\}, \qquad t > 0$$

(see e.g. [11, Lemma 2.2] for a proof). The first step is to establish the existence of a "nice" sequence $b \in \ell_2$ arbitrarily close to this supremum:

Claim 3.2.16. *For any $\delta > 0$, there exists a* non-increasing, *non-negative sequence $b^* \in \ell_2$ with $\max(\|b^*\|_\infty, t^{-1}\|b^*\|_2) \le 1$ such that*

$$(1-\delta)\kappa_a \le \sum_{k=1}^{\infty} a_k b_k^*.$$

Proof. By the above characterization, there exists a sequence $b \in \ell_2$ with $\max(\|b\|_\infty, t^{-1}\|b\|_2) \le 1$ such that $(1-\delta)\kappa_a \le \sum_{k=1}^{\infty} a_k b_k$. We now claim that we can further take b to be non-negative and monotone non-increasing as well. The first part is immediate, as replacing negative terms by their absolute values can only increase the sum (since a is itself non-negative). For the second part, we will invoke the Hardy–Littlewood rearrangement inequality (Theorem 1.4.15), which states that for any two non-negative functions f, g vanishing at infinity, the integral $\int_{\mathbb{R}} fg$ is maximized when f and g are non-increasing. We now apply this inequality to a, b, letting a^*, b^* be the non-increasing rearrangements of a, b (in particular, we have $a = a^*$) and introducing the functions f_a, f_b:

$$f_a = \sum_{j=1}^{\infty} a_j \mathbb{1}_{(j-1,j]}, \quad f_b = \sum_{j=1}^{\infty} b_j \mathbb{1}_{(j-1,j]}$$

which satisfy the hypotheses of Theorem 1.4.15. Thus, we get $\int_{\mathbb{R}} f_a f_b \le \int_{\mathbb{R}} f_a^* f_b^*$; as it is easily seen that $f_a^* = f_{a^*}$ and $f_b^* = f_{b^*}$, this yields

$$\sum_{k=1}^{\infty} a_k b_k = \int_{\mathbb{R}} f_a f_b \le \int_{\mathbb{R}} f_a^* f_b^* = \sum_{k=1}^{\infty} a_k^* b_k^* = \sum_{k=1}^{\infty} a_k b_k^*.$$

Moreover, it is immediate to check that $\max(\|b^*\|_\infty, t^{-1}\|b^*\|_2) \le 1$. □

The next step is to relate the inner product $\sum_{k=1}^{\infty} a_k b_k^*$ to the Q-norm of a:

Claim 3.2.17. *Fix $t > 0$ such that $t^2 \in \mathbb{N}$, and let $b^* \in \ell_2$ be any non-increasing, non-negative sequence with $\max(\|b^*\|_\infty, t^{-1}\|b^*\|_2) \le 1$. Then*

$$\sum_{k=1}^{\infty} a_k b_k^* \le \|a\|_{Q(2t^2)}.$$

Proof. We proceed constructively, by exhibiting a partition of $\mathbb{N}$ into $2t^2$ sets $A_1, \dots, A_{2t^2}$ satisfying $\sum_{k=1}^{\infty} a_k b_k^* \le \sum_{j=1}^{2t^2} \left(\sum_{i \in A_j} {b_i^*}^2\right)^{1/2}$. This will prove the claim, by definition of $\|a\|_{Q(2t^2)}$ as the supremum over all such partitions.

Specifically, we inductively choose $n_0, n_1, \dots, n_T \in \{0, \dots, \infty\}$ as follows, where $T \stackrel{\text{def}}{=} \frac{t^2}{c}$ for some

$c > 0$ to be chosen later (satisfying $T \in \mathbb{N}$). If $0 = n_0 < n_1 < \cdots < n_m$ are already set, then

$$n_{m+1} \stackrel{\text{def}}{=} 1 + \sup\left\{ \ell \geq n_m \,:\, \sum_{i=n_m+1}^{\ell} b_i^{*2} \leq c \right\}.$$

From $\|b^*\|_2 \leq t$, it follows that $n_T = \infty$. Let m^* be the first index such that $n_{m^*+1} > n_{m^*} + 1$. Note that this implies (by monotonicity of b^*) that $b_i^{*2} > c$ for all $i \leq n_{m^*}$, and $b_i^{*2} \leq c$ for all $i \geq n_{m^*} + 1$. We can write

$$\sum_{i=1}^{\infty} a_i b_i^* = \sum_{m=1}^{T} \sum_{i=n_{m-1}+1}^{n_m} a_i b_i^* = \sum_{i=1}^{n_{m^*}} a_i b_i^* + \sum_{m=m^*+1}^{T} \sum_{i=n_{m-1}+1}^{n_m} a_i b_i^*$$

Since $\|b^*\|_\infty \leq 1$ and $n_{m-1} + 1 = n_m$ for all $m \leq m^*$, the first term can be bounded as

$$\sum_{i=1}^{n_{m^*}} a_i b_i^* \leq \sum_{i=1}^{n_{m^*}} \sqrt{a_i^2} = \sum_{m=1}^{m^*} \left(\sum_{i=n_{m-1}+1}^{n_m} a_i^2 \right)^{1/2}.$$

Turning to the second term, we recall that $b_i^{*2} \leq c$ for all $i \geq n_{m^*} + 1$, so that $\sum_{i=n_{m-1}+1}^{n_m} b_i^{*2} \leq 2c$ for all $m \geq m^* + 1$. This allows us to bound the second term as

$$\sum_{m=m^*+1}^{T} \sum_{i=n_{m-1}+1}^{n_m} a_i b_i^* \leq \sum_{m=m^*+1}^{T} \left(\sum_{i=n_{m-1}+1}^{n_m} b_i^{*2} \right)^{1/2} \left(\sum_{i=n_{m-1}+1}^{n_m} a_i^2 \right)^{1/2} \leq \sqrt{2c} \sum_{m=m^*+1}^{T} \left(\sum_{i=n_{m-1}+1}^{n_m} a_i^2 \right)^{1/2}$$

Therefore, by combining the two we get that

$$\begin{aligned}(1-\delta)\kappa_a(t) &\leq \sum_{m=1}^{m^*} \left(\sum_{i=n_{m-1}+1}^{n_m} a_i^2 \right)^{1/2} + \sqrt{2c} \sum_{m=m^*+1}^{T} \left(\sum_{i=n_{m-1}+1}^{n_m} a_i^2 \right)^{1/2} \leq \max(1, \sqrt{2c}) \sum_{m=1}^{T} \left(\sum_{i=n_{m-1}+1}^{n_m} a_i^2 \right)^{1/2} \\ &\leq \max(1, \sqrt{2c}) \|a\|_{Q(T)} = \|a\|_{Q(2t^2)}\end{aligned}$$

the last equality by choosing $c \stackrel{\text{def}}{=} \frac{1}{2}$. □

We now fix an arbitrary $\delta > 0$, and let b^* be as promised by Claim 3.2.16. As this sequence satisfies the assumptions of Claim 3.2.17, putting the two results together leads to

$$(1-\delta)\kappa_a(t) \leq \sum_{k=1}^{\infty} a_k b_k^* \leq \|a\|_{Q(2t^2)}.$$

Since this holds for all $\delta > 0$, taking the limit as $\delta \searrow 0$ gives the (upper bound of the) lemma. □

We observe that, with similar techniques, one can also establish the following generalization of Proposition 3.2.13:

Lemma 3.2.18 (Generalization of Proposition 3.2.13)**.** *For any $a \in \ell_2$, t, and $\alpha \in [1, \infty)$ such that $t^2, \alpha t^2 \in$*

$\mathbb{N}$, *we have*

$$\|a\|_{Q(t^2)} \leq \kappa_a(t) \leq \sqrt{1+\alpha^{-1}}\|a\|_{Q(\alpha t^2)}. \tag{3.8}$$

Proof of Lemma 3.2.18 (Sketch). We again follow the proof of [11, Lemma 2.2], up to the inductive definition of $n_1, \ldots, n_j$, which we change as

$$n_{m+1} = 1 + \sup\left\{ \ell \geq n_m \,:\, \sum_{i=n_m+1}^{\ell} b_i^2 \leq \frac{1}{\alpha} \right\}.$$

Since $\|b\|_\infty \leq 1$, we have $\sum_{i=n_m+1}^{n_{m+1}} b_i^2 \leq 1 + \frac{1}{\alpha}$. From $\|b\|_2 \leq t$, it follows that $n_{\alpha t^2} = \infty$. Therefore, for any $\delta > 0$,

$$(1-\delta)\kappa_a(t) \leq \sum_{i=1}^{\infty} a_i b_i \leq \sum_{m=1}^{T} \left(\sum_{i=n_{m-1}+1}^{n_m} b_i^2 \right)^{1/2} \left(\sum_{i=n_{m-1}+1}^{n_m} a_i^2 \right)^{1/2} \leq \sqrt{1+\frac{1}{\alpha}}\|a\|_{Q(\alpha t^2)}.$$

Since this holds for all $\delta > 0$, taking the limit gives the (upper bound of the) lemma. □

We note that further inequalities relating κ_a to other functionals of a were obtained in [111].

3.2.5.2 Concentration Inequalities for Weighted Rademacher Sums

The connection between the K-functional and tail bounds on weighted sums of Rademacher random variables was first made by Montgomery-Smith [133], to which the following result is due (we here state a version with slightly improved constants):

Theorem 3.2.19. *Let $(X_i)_{i\in\mathbb{N}}$ be a sequence of independent Rademacher random variables, i.e. uniform on $\{-1, 1\}$. Then, for any $a \in \ell_2$ and $t > 0$,*

$$\Pr\left[\sum_{i=1}^{\infty} a_i X_i \geq \kappa_a(t) \right] \leq e^{-\frac{t^2}{2}}. \tag{3.9}$$

and, for any fixed $c > 0$ and all $t \geq 1$,

$$\Pr\left[\sum_{i=1}^{\infty} a_i X_i \geq \frac{1}{1+c}\kappa_a(t) \right] \geq e^{-\left(\frac{2}{c}\ln\frac{\sqrt{6}(1+c)}{c}\right)(t^2+c)}. \tag{3.10}$$

In particular,

$$\Pr\left[\sum_{i=1}^{\infty} a_i X_i \geq \frac{1}{2}\kappa_a(t) \right] \geq e^{-(\ln 24)(t^2+1)} \geq e^{-(2\ln 24)t^2}.$$

One can intepret the above theorem as stating that the (inverse of the) K-functional κ_a is the "right" parameter to consider in these tail bounds; while standard Chernoff or Hoeffding bounds will depend instead on the quantity $\|a\|_2$. Before giving the proof of this theorem, we remark that similar statements or improvements can be found in [111] and [11]; below, we closely follow the argument of the latter.

Proof of Theorem 3.2.19. The upper bound can be found in e.g. [133], or [11, Theorem 2.2]. For the lower bound, we mimic the proof due to Astashkin, improving the parameters of some of the lemmas it relies on.

Lemma 3.2.20 (Small improvement of (2.14) in [11, Lemma 2.3])**.** *If $a = (a_k)_{k\geq 1} \in \ell_2$, then, for any $\lambda \in (0,1)$,*

$$\Pr\left[\left|\sum_{k=1}^{\infty} a_k X_k\right|^2 \geq \lambda \sum_{k=1}^{\infty} a_k^2\right] \geq \frac{1}{3}(1-\lambda)^2. \tag{3.11}$$

Proof of Lemma 3.2.20. The proof is exactly the same, but when invoking (1.10) for $p = 4$ we use the actual tight version proven there for $p = 2m$ (instead of the more general version that also applies to odd values of p): since $m = 2$, we get $\frac{(2m)!}{2^m m!} = 3$, and $\mathbb{E}[f]^2 \geq \frac{1}{3}\mathbb{E}\big[f^2\big]$ in the proof (instead of $(\frac{p}{2}+1)^{-\frac{p}{2}} = \frac{1}{9}$). □

Using the lemma above along with Lemma 3.2.15 in the proof of [11, Theorem 2.2], we can strenghten it as follows: letting $T \stackrel{\text{def}}{=} \frac{t^2}{c}$, for arbitrary $\delta > 0$ we fix a partition $A_1, \ldots, A_T$ of $\mathbb{N}$ such that $\|a\|_{Q(T)} \leq (1+\delta)\sum_{j=1}^{T}\left(\sum_{k\in A_j} a_k^2\right)^{1/2}$,

$$\begin{aligned}
\Pr\left[\sum_{k=1}^{\infty} a_k X_k > \frac{1}{1+c}\kappa_a(t)\right] &\geq \Pr\left[\sum_{k=1}^{\infty} a_k X_k > \frac{1}{\sqrt{1+c}}\|a\|_{Q(T)}\right] && \text{(by (3.7))}\\
&\geq \Pr\left[\sum_{j=1}^{T}\sum_{k\in A_j} a_k X_k > \frac{1+\delta}{\sqrt{1+c}}\sum_{j=1}^{T}\left(\sum_{k\in A_j} a_k^2\right)^{1/2}\right]\\
&\geq \prod_{j=1}^{T}\Pr\left[\sum_{k\in A_j} a_k X_k > \frac{1+\delta}{\sqrt{1+c}}\left(\sum_{k\in A_j} a_k^2\right)^{1/2}\right]\\
&= \prod_{j=1}^{T}\frac{1}{2}\Pr\left[\left|\sum_{k\in A_j} a_k X_k\right|^2 > \left(\frac{1+\delta}{\sqrt{1+c}}\right)^2\left(\sum_{k\in A_j} a_k^2\right)\right] && \text{(symmetry)}\\
&\geq \prod_{j=1}^{T}\frac{1}{6}\left(1-\frac{(1+\delta)^2}{1+c}\right)^2. && \text{(Lemma 3.2.20)}
\end{aligned}$$

By taking the limit as $\delta \to 0^+$, we then obtain

$$\Pr\left[\sum_{k=1}^{\infty} a_k X_k > \frac{1}{1+c}\kappa_a(t)\right] \geq \left(\frac{1}{6}\left(1-\frac{1}{1+c}\right)^2\right)^T = \left(\frac{c}{\sqrt{6}(1+c)}\right)^{\frac{2t^2}{c}} = e^{-\left(\frac{2}{c}\ln\frac{\sqrt{6}(1+c)}{c}\right)t^2}. \tag{3.12}$$

This takes care of the case where $\frac{t^2}{c}$ is an integer. If this is not the case, we consider $s \stackrel{\text{def}}{=} \sqrt{c\left(\left\lfloor\frac{t^2}{c}\right\rfloor + 1\right)}$, so that $t^2 \leq s^2 \leq t^2 + c$. The monotonicity of κ_a then ensures that

$$\Pr\left[\sum_{k=1}^{\infty} a_k X_k > \frac{1}{1+c}\kappa_a(t)\right] \geq \Pr\left[\sum_{k=1}^{\infty} a_k X_k > \frac{1}{1+c}\kappa_a(s)\right] \underset{(3.12)}{\geq} e^{-\left(\frac{2}{c}\ln\frac{\sqrt{6}(1+c)}{c}\right)s^2} \geq e^{-\left(\frac{2}{c}\ln\frac{\sqrt{6}(1+c)}{c}\right)(t^2+c)}$$

which concludes the proof. □

3.2.5.3 Some Examples

To gain intuition about the behavior of κ_a, we now compute tight asymptotic expressions for it in several instructive cases, specifically for some natural examples of probability distributions in $\Delta(\Omega)$.

From the lower bound of Proposition 3.2.13 and the fact that $\kappa_{\mathbf{p}} \leq \|\mathbf{p}\|_1$ for any $\mathbf{p} \in \ell_1$, it is clear that as soon as $t \geq \sqrt{n}$, $\kappa_{\mathbf{p}}(t) = 1$ for any $\mathbf{p} \in \Delta(\Omega)$. It suffices then to consider the case $0 \leq t \leq \sqrt{n}$.

The uniform distribution. We let $\mathbf{p}$ be the uniform distribution on $[n]$: $\mathbf{p}_k = \frac{1}{n}$ for all $i \in [n]$. By considering a partition of $[n]$ into t^2 sets of size $\frac{n}{t^2}$, the lower bound of Proposition 3.2.13 yields $\kappa_{\mathbf{p}}(t) \geq \|\mathbf{p}\|_{Q(t^2)} \geq \frac{t}{\sqrt{n}}$. On the other hand, by definition $\kappa_{\mathbf{p}}(t) = \inf_{\mathbf{p}'+\mathbf{p}''=\mathbf{p}} \|\mathbf{p}'\|_1 + t\|\mathbf{p}''\|_2 \leq t\|\mathbf{p}\|_2 = \frac{t}{\sqrt{n}}$, and thus

$$\kappa_{\mathbf{p}}(t) = \begin{cases} \frac{t}{\sqrt{n}} & \text{if } t \leq \sqrt{n} \\ 1 & \text{if } t \geq \sqrt{n}. \end{cases}$$

We remark that in this case, the upper bound of Holmstedt from Proposition 3.2.12 only results in

$$\kappa_{\mathbf{p}}(t) \leq \frac{t^2}{n} + t\sqrt{\frac{n-t^2}{n^2}} = f\left(\frac{t}{\sqrt{n}}\right)$$

where $f\colon x \in [0,1] \mapsto x^2 + x\sqrt{1-x^2}$. It is instructive to note this shows that this could not possibly have been the right upper bound (and therefore that Proposition 3.2.12 cannot be tight in general), as f is neither concave nor non-decreasing, and not even bounded by 1:

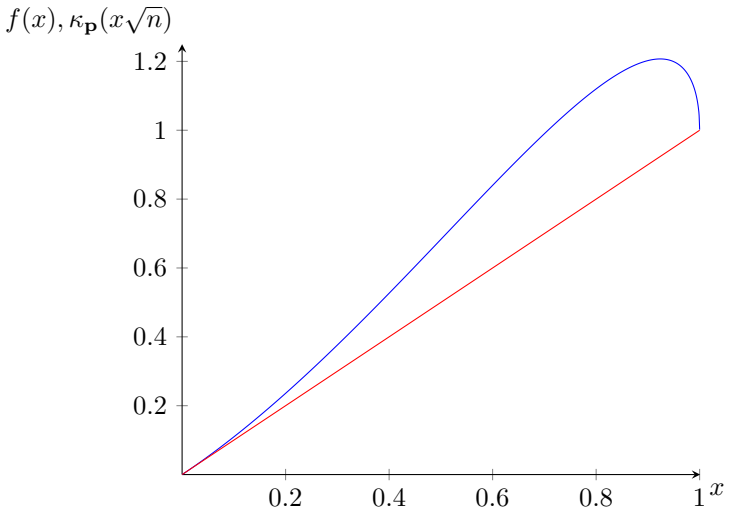

Figure 3.3: Example of the K-functional for the uniform distribution over $[n]$: Holmstedt's upper bound (in blue) vs. true behavior of $\kappa_{\mathbf{p}}$ (in red).

From the above, we can now compare the behavior of $\kappa_{\mathbf{p}}^{-1}(1-2\varepsilon)$ to the "2/3-norm functional" introduced

by Valiant and Valiant [169]: for $\varepsilon \in (0, 1/2)$,

$$\kappa_{\mathbf{p}}^{-1}(1-2\varepsilon) = (1-2\varepsilon)\sqrt{n}, \quad \|\mathbf{p}_{-\varepsilon}^{-\max}\|_{2/3} = (1-\varepsilon)^{3/2}\sqrt{n} + o(1). \tag{3.13}$$

The Harmonic distribution. We now consider the case of the (truncated) Harmonic distribution, letting $\mathbf{p} \in \Delta([n])$ be defined as $\mathbf{p}_k = \frac{1}{kH_n}$ for all $i \in [n]$ (H_n being the n-th Harmonic number). By considering a partition of $[n]$ into $t^2 - 1$ sets of size 1 and one of size $n - t^2$, the lower bound of Proposition 3.2.13 yields

$$H_n \kappa_{\mathbf{p}}(t) \geq \|\mathbf{p}\|_{Q(t^2)} \geq \sum_{k=1}^{t^2-1} \frac{1}{k} + \sqrt{\sum_{k=t^2}^{n} \frac{1}{k^2}}$$

while Holmstedt's upper bound gives

$$H_n \kappa_{\mathbf{p}}(t) \leq \sum_{k=1}^{t^2-1} \frac{1}{k} + t\sqrt{\sum_{k=t^2}^{n} \frac{1}{k^2}}.$$

For $t = O(1)$, this implies that $\kappa_{\mathbf{p}}(t) = o(1)$; however, for $t = \omega(1)$ (but still less than $\sqrt{n}$), an asymptotic development of both upper and lower bounds shows that

$$\kappa_{\mathbf{p}}(t) = \frac{2\ln t + O(1)}{\ln n}.$$

Using this expression, we can again compare the behavior of $\kappa_{\mathbf{p}}^{-1}(1-2\varepsilon)$ to the 2/3-norm functional of [169]: for $\varepsilon \in (0, 1/2)$,

$$\kappa_{\mathbf{p}}^{-1}(1-2\varepsilon) = \Theta\left(n^{\frac{1}{2}-\varepsilon}\right), \quad \|\mathbf{p}_{-\varepsilon}^{-\max}\|_{2/3} = \Theta\left(\frac{n^{\frac{1-\varepsilon}{2}}}{\log n}\right) = \Theta\left(n^{\frac{1-\varepsilon}{2}-o(1)}\right). \tag{3.14}$$

3.2.6 Identity Testing, revisited

For any $x \in (0, 1/2)$ and sequence $a \in \ell_1$, we let $t_x \stackrel{\text{def}}{=} \kappa_a^{-1}(1-2x)$, where κ_a is the K-functional of a as previously defined. Armed with the results and characterizations from the previous section, we will first in Section 3.2.6.1 describe an elegant reduction from communication complexity leading to a lower bound on instance-optimal identity testing parameterized by the quantity t_ε. Guided by this lower bound, we then will in Section 3.2.6.2 consider this result from the *upper bound* viewpoint, and in Theorem 3.2.27 establish that indeed this parameter captures the sample complexity of this problem. Finally, Section 3.2.6.3 is concerned with tightening our lower bound by using different arguments: specifically, showing that the bound that appeared naturally as a consequence of our communication complexity approach can, in hindsight, be established and slightly strenghtened with standard distribution testing arguments.

3.2.6.1 The Communication Complexity Lower Bound

In this subsection we prove the following lower bound on identity testing, via reduction from SMP communication complexity.

Theorem 3.2.21. *Let Ω be a finite domain, and let $\mathbf{p} = (\mathbf{p}_1, \ldots, \mathbf{p}_n) \in \Delta(\Omega)$ be a distribution, given as a parameter. Let $\varepsilon \in (0, 1/5)$, and set $t_\varepsilon \stackrel{\text{def}}{=} \kappa_{\mathbf{p}}^{-1}(1 - 2\varepsilon)$. Then, given sample access to a distribution $\mathbf{q} = (\mathbf{q}_1, \ldots, \mathbf{q}_n) \in \Delta(\Omega)$, testing $\mathbf{p} = \mathbf{q}$ versus $\|\mathbf{p} - \mathbf{q}\|_1 > \varepsilon$ requires $\Omega(t_\varepsilon / \log(n))$ samples from $\mathbf{q}$.*

We will follow the argument outlined in Section 3.2.2.2: namely, applying the same overall idea as in the reduction for uniformity testing, but with an error-correcting code specifically designed for the distribution $\mathbf{p}$ instead of a standard Hamming one. To prove Theorem 3.2.21 we thus first need to define and obtain codes with properties that are tailored for our reduction; which we do next.

Balanced $\mathbf{p}$-weighted codes Recall that in our reductions so far, the first step is for Alice and Bob to apply a code to their inputs; typically, we chose that code to be a balanced code with constant rate, and linear distance *with respect to the uniform distribution* (i.e., with good Hamming distance). In order to obtain better bounds on a case-by-case basis, it will be useful to consider a generalization of these codes, under a different distribution:

Definition 3.2.22 ($\mathbf{p}$-distance)**.** For any $n \in \mathbb{N}$, given a probability distribution $\mathbf{p} \in \Delta([n])$ we define the $\mathbf{p}$-*distance* on $\{0,1\}^n$, denoted $\mathrm{dist}_{\mathbf{p}}$, as the weighted Hamming distance

$$\mathrm{dist}_{\mathbf{p}}(x, y) \stackrel{\text{def}}{=} \sum_{i=1}^{n} \mathbf{p}(i) \cdot |x_i - y_i|$$

for $x, y \in \{0,1\}^n$. (In particular, this is a pseudometric on $\{0,1\}^n$.) The $\mathbf{p}$-*weight* of $x \in \{0,1\}^n$ is given by $\mathrm{weight}_{\mathbf{p}}(x) \stackrel{\text{def}}{=} \mathrm{dist}_{\mathbf{p}}(x, 0^n)$.

A $\mathbf{p}$-weighted code is a code whose distance guarantee is with respect to the $\mathbf{p}$-distance.

Definition 3.2.23 ($\mathbf{p}$-weighted codes)**.** Fix a probability distribution $\mathbf{p} \in \Delta([n])$. We say that $C \colon \{0,1\}^k \to \{0,1\}^n$ is a (binary) $\mathbf{p}$-*weighted code with relative distance $\gamma = \gamma(n)$ and rate $\rho = k/n$* if

$$\mathrm{dist}_{\mathbf{p}}(C(x), C(y)) > \gamma$$

for all distinct $x, y \in \{0,1\}^k$.

Recall that the "vanilla" reduction in Section 3.2.4 relies on *balanced* codes. We generalize the balance property to the $\mathbf{p}$-distance and allow the following relaxation.

Definition 3.2.24 ($\mathbf{p}$-weighted τ-balance)**.** A $\mathbf{p}$-weighted code $C \colon \{0,1\}^k \to \{0,1\}^n$ is τ-*balanced* if there exists $\tau \in (0,1)$ such that $\mathrm{weight}_{\mathbf{p}}(C(x)) \in \left(\frac{1}{2} - \tau, \frac{1}{2} + \tau\right)$ for all $x \in \{0,1\}^k$.

Now, for a distribution $\mathbf{p}$, the volume of the $\mathbf{p}$-ball in $\{0,1\}^n$ is given by

$$\mathrm{Vol}_{\mathbb{F}_2^n,\mathrm{dist}_{\mathbf{p}}}(\varepsilon) \stackrel{\text{def}}{=} \left|\left\{\, w \in \mathbb{F}_2^n \;:\; \mathrm{weight}_{\mathbf{p}}(w) \le \varepsilon \,\right\}\right|.$$

Next, we show that there exist nearly balanced $\mathbf{p}$-weighted codes with constant relative distance and nearly optimal rate.

Proposition 3.2.25 (Existence of nearly balanced $\mathbf{p}$-weighted codes)**.** *Fix a probability distribution $\mathbf{p} \in \Delta([n])$, constants $\gamma, \tau \in (0, \frac{1}{3})$, and $\varepsilon = \max\{\gamma, \frac{1}{2} - \tau\}$. There exists a $\mathbf{p}$-weighted τ-balanced code $C\colon \{0,1\}^k \to \{0,1\}^n$ with relative distance γ such that $k = \Omega(n - \log \mathrm{Vol}_{\mathbb{F}_2^n,\mathrm{dist}_{\mathbf{p}}}(\varepsilon))$.*

In contrast, by the sphere packing bound, every $\mathbf{p}$-weighted code $C\colon \{0,1\}^k \to \{0,1\}^n$ with distance γ satisfies

$$\underbrace{2^k}_{\#\text{codewords}} \le \frac{2^n}{\mathrm{Vol}_{\mathbb{F}_2^n,\mathrm{dist}_{\mathbf{p}}}(\gamma/2)}.$$

Hence, we have $k \le n - \log \mathrm{Vol}_{\mathbb{F}_2^n,\mathrm{dist}_{\mathbf{p}}}(\gamma/2)$.

Proof of Proposition 3.2.25. Note that

$$\mathrm{Vol}_{\mathbb{F}_2^n,\mathrm{dist}_{\mathbf{p}}}(\varepsilon) = \left|\left\{\, w \in \mathbb{F}_2^n \;:\; \mathrm{weight}_{\mathbf{p}}(w) \le \varepsilon \,\right\}\right| = 2^n \cdot \Pr_{w \sim \{0,1\}^n}\left[\sum_{i=1}^n \mathbf{p}_i w_i \le \varepsilon\right].$$

The probability that a randomly chosen code $C\colon \{0,1\}^k \to \{0,1\}^n$ does *not* have distance γ is

$$\begin{aligned}
\Pr_C\left[\,\exists x, y \in \{0,1\}^k \text{ such that } \mathrm{dist}_{\mathbf{p}}(C(x), C(y)) \le \gamma\,\right] &\le 2^{2k} \cdot \Pr_{w,w' \sim \{0,1\}^n}\left[\,\mathrm{dist}_{\mathbf{p}}(w, w') \le \gamma\,\right] \\
&\le 2^{2k} \cdot \Pr_{w \sim \{0,1\}^n}\left[\sum_{i=1}^n \mathbf{p}_i w_i \le \varepsilon\right] \\
&= \frac{\mathrm{Vol}_{\mathbb{F}_2^n,\mathrm{dist}_{\mathbf{p}}}(\varepsilon)}{2^{n-2k}}.
\end{aligned}$$

Hence, for sufficiently small $k = \Omega(n - \log \mathrm{Vol}_{\mathbb{F}_2^n,\mathrm{dist}_{\mathbf{p}}}(\varepsilon))$, the probability that a random code is a $\mathbf{p}$-weighted code with relative distance γ is at least $2/3$; fix such k. Similarly, the probability that a random code $C\colon \{0,1\}^k \to \{0,1\}^n$ is not τ-balanced (under the $\mathbf{p}$-distance) is

$$\begin{aligned}
\Pr_C\left[\exists x \in \{0,1\}^k \text{ such that } \mathrm{weight}_{\mathbf{p}}(C(x)) \notin \left(\frac{1}{2} - \tau, \frac{1}{2} + \tau\right)\right] &\le 2^k \cdot \Pr_{w \in \{0,1\}^n}\left[\left|\mathrm{weight}_{\mathbf{p}}(w) - \frac{1}{2}\right| > \tau\right] \\
&\le 2^{k+1} \cdot \Pr_{w \in \{0,1\}^n}\left[\sum_{i=1}^n \mathbf{p}_i w_i < \varepsilon\right] \\
&\le \frac{\mathrm{Vol}_{\mathbb{F}_2^n,\mathrm{dist}_{\mathbf{p}}}(\varepsilon)}{2^{n-k-1}}.
\end{aligned}$$

Thus, the probability that a random code is τ-balanced (under the $\mathbf{p}$-distance) is at least $2/3$, and so, with probability at least $\frac{1}{3}$, a random code satisfies the proposition's hypothesis. □

We now establish a connection between the rate of $\mathbf{p}$-weighted codes and the K-functional of $\mathbf{p}$, as introduced in Section 3.2.5:

Claim 3.2.26. *Let $\mathbf{p} \in \Delta(\Omega)$ be a probability distribution. Then, for any $\gamma \in (0, \frac{1}{2})$ we have*

$$n - \log \mathrm{Vol}_{\mathbb{F}_2^n, \mathrm{dist}_{\mathbf{p}}}(\gamma) \geq \frac{1}{2 \ln 2} \kappa_{\mathbf{p}}^{-1}(1-2\gamma)^2$$

where $\kappa_{\mathbf{p}}^{-1}(u) = \inf \{\, t \in (0,\infty) : \kappa_a(t) \geq u \,\}$ for $u \in [0,\infty)$.

Proof. From the definition,

$$\mathrm{Vol}_{\mathbb{F}_2^n, \mathrm{dist}_{\mathbf{p}}}(\gamma) = \left|\left\{ w \in \mathbb{F}_2^n : \ \mathrm{weight}_{\mathbf{p}}(w) \leq \gamma \right\}\right| = \left|\left\{ w \in \mathbb{F}_2^n : \ \sum_{i=1}^n \mathbf{p}_i w_i \leq \gamma \right\}\right| = 2^n \Pr_{Y \sim \{0,1\}^n}\left[\sum_{i=1}^n \mathbf{p}_i Y_i \leq \gamma\right]$$
$$= 2^n \Pr_{X \sim \{-1,1\}^n}\left[\sum_{i=1}^n \mathbf{p}_i X_i \geq 1 - 2\gamma\right] = 2^n \Pr_{X \sim \{-1,1\}^n}\left[\sum_{i=1}^n \mathbf{p}_i X_i \geq \kappa_{\mathbf{p}}(u_\gamma)\right]$$

where we set $u_\gamma \stackrel{\mathrm{def}}{=} \kappa_{\mathbf{p}}^{-1}(1-2\gamma)$. From Theorem 3.2.19, we then get $\mathrm{Vol}_{\mathbb{F}_2^n, \mathrm{dist}_{\mathbf{p}}}(\gamma) \leq 2^n e^{-\frac{u_\gamma^2}{2}}$, from which

$$n - \log \mathrm{Vol}_{\mathbb{F}_2^n, \mathrm{dist}_{\mathbf{p}}}(\gamma) \geq -\log e^{-\frac{u_\gamma^2}{2}} = \frac{1}{2 \ln 2} u_\gamma^2$$

as claimed. □

The Reduction Equipped with the nearly balanced $\mathbf{p}$-weighted codes in Proposition 3.2.25, we are ready to prove Theorem 3.2.21. Assume there exists an s-sample ε-tester for identity to $\mathbf{p}$, with error probability $1/6$, and assume, without loss of generality, that ε is a constant (independent of n).

Fix $\gamma = \varepsilon$ and $\tau = (1-2\varepsilon)/2$. For a sufficiently large $k \in \mathbb{N}$, let $C \colon \{0,1\}^k \to \{0,1\}^n$ be a τ-balanced $\mathbf{p}$-weighted code with relative distance γ, as guaranteed by Proposition 3.2.25; namely, the code C satisfies the following conditions.

(i) *Balance*: $\mathrm{weight}_{\mathbf{p}}(C(x)) \in \left(\frac{1}{2} - \tau, \frac{1}{2} + \tau\right)$ for all $x \in \{0,1\}^k$;

(ii) *Distance*: $\mathrm{dist}_{\mathbf{p}}(C(x), C(y)) > \gamma$ for all distinct $x, y \in \{0,1\}^k$;

(iii) *Rate*: $k = \Omega(n - \log \mathrm{Vol}_{\mathbb{F}_2^n, \mathrm{dist}_{\mathbf{p}}}(\varepsilon))$.

We reduce from the problem of equality in the (private coin) SMP model. Given their respective inputs $x, y \in \{0,1\}^k \times \in \{0,1\}^k$ from EQ_k, Alice and Bob separately create inputs $(a, b) = (C(x), C(y)) \in \{0,1\}^n \times \in \{0,1\}^n$. Let $A \subseteq [n]$ denote the set indicated by a, and let $B \subseteq [n]$ denote the set indicated by $\bar{b}$. Alice and Bob then each send to the referee the $\mathbf{p}$-weight of their encoded input, $\mathrm{weight}_{\mathbf{p}}(a) = \mathbf{p}(A)$ and $\mathrm{weight}_{\mathbf{p}}(\bar{b}) = \mathbf{p}(B)$ respectively,[20] as well as a sequence of $6cs$ samples independently drawn from the distribution $\mathbf{p}$ *restricted* to the subsets A and B respectively, where c is the constant such that $\frac{1}{c} p(B) \cdot \leq$

[20] A standard argument shows it suffices to specify $\mathbf{p}(A)$ and $\mathbf{p}(B)$ with precision roughly $1/n^2$, and so sending the weights only costs $O(\log n)$ bits.

$\mathbf{p}(A) \leq c \cdot \mathbf{p}(B)$, guaranteed by the balance property of C. Finally, the referee checks that $\mathbf{p}(A) + \mathbf{p}(B) = 1$ (and otherwise rejects) and generates a sequence of $\mathbf{q}$ samples by choosing independently, for each of them, Alice's element with probability $\mathbf{p}(A)$ and Bob's with probability $\mathbf{p}(B)$, and feeds these samples to the ε-tester for identity to $\mathbf{p}$.

By Markov's inequality, the above procedure indeed allows the referee to retrieve, with probability at least $1 - \frac{cs}{6cs} = \frac{5}{6}$, at least s independent samples from the distribution

$$q \stackrel{\text{def}}{=} \mathbf{p}(A) \cdot \mathbf{p}|_A + \mathbf{p}(B) \cdot \mathbf{p}|_B,$$

at the cost of $O(s \log n)$ bits of communication in total.

For correctness, note that if $x = y$, then $A = \bar{B}$, which implies $\mathbf{q} = \mathbf{p}$. On the other hand, if $x \neq y$, by the ($\mathbf{p}$-weighted) distance of C we have $\mathrm{dist}_{\mathbf{p}}(C(x), C(y)) > \gamma$, and so $\mathbf{p}(A \cap B) + \mathbf{p}(\overline{A \cup B}) > \gamma$. Note that every $i \in A \cap B$ satisfies $\mathbf{q}_i = 2\mathbf{p}_i$ and every $i \in \overline{A \cup B}$ is not supported in $\mathbf{q}$. Therefore, we have $\|\mathbf{p} - \mathbf{q}\|_1 > \varepsilon$. The referee can therefore invoke the identity testing algorithm to distinguish between $\mathbf{p}$ and $\mathbf{q}$ with probability $1 - (\frac{1}{6} + \frac{1}{6}) = \frac{2}{3}$. This implies that the number of samples $\mathbf{q}$ used by any such tester must satisfy $s \log n = \Omega\Big(\sqrt{k}\Big)$. Finally, by Claim 3.2.26 we have

$$k = \Omega\big(n - \log \mathrm{Vol}_{\mathbb{F}_2^n, \mathrm{dist}_{\mathbf{p}}}(\varepsilon)\big) = \Omega\big(\kappa_{\mathbf{p}}^{-1}(1 - 2\varepsilon)^2\big),$$

and therefore we obtain a lower bound of $s = \Omega(t_\varepsilon / \log(n))$.

3.2.6.2 The Upper Bound

Inspired by the results of the previous section, it is natural to wonder whether the dependence on t_ε of the lower bound is the "right" one. Our next theorem shows that this is the case: the parameter t_ε does, in fact, capture the sample complexity of the problem.

Theorem 3.2.27. *There exists an absolute constant $c > 0$ such that the following holds. Given any fixed distribution $\mathbf{p} \in \Delta([n])$ and parameter $\varepsilon \in (0, 1]$, and granted sample access to an unknown distribution $\mathbf{q} \in \Delta([n])$, one can test $\mathbf{p} = q$ vs. $\|\mathbf{p} - \mathbf{q}\|_1 > \varepsilon$ with $O\big(\max\big(\frac{t_{c\varepsilon}}{\varepsilon^2}, \frac{1}{\varepsilon}\big)\big)$ samples from $\mathbf{q}$. (Moreover, one can take $c = \frac{1}{18}$).*

High-level idea As discussed in Section 3.2.2.4, the starting point of the proof is the connection between the K-functional and the "Q-norm" obtained in Lemma 3.2.15: indeed, this result ensures that for $T = 2t^2_{O(\varepsilon)}$, there exists a partition of the domain into sets $A_1, \ldots, A_T$ such that

$$1 - O(\varepsilon) \leq \|\mathbf{p}\|_{Q(T)} = \sum_{j=1}^{T} \sqrt{\sum_{i \in A_j} \mathbf{p}_i^2} = \sum_{j=1}^{T} \|\mathbf{p}_{A_j}\|_2$$

where $\mathbf{p}_{A_j}$ is the restriction of the sequence $\mathbf{p}$ to the indices in A_j. But by the monotonicity of ℓ_p norms, we know that $\sum_{j=1}^{T} \|\mathbf{p}_{A_j}\|_2 \leq \sum_{j=1}^{T} \|\mathbf{p}_{A_j}\|_1 = \sum_{j=1}^{T} \sum_{i \in A_j} \mathbf{p}_i = \|\mathbf{p}\|_1 = 1$. Therefore, what we obtain is in fact that

$$0 \leq \sum_{j=1}^{T} (\underbrace{\|\mathbf{p}_{A_j}\|_1 - \|\mathbf{p}_{A_j}\|_2}_{\geq 0}) \leq O(\varepsilon).$$

Now, if the right-hand side were *exactly* 0, then this would imply $\|\mathbf{p}_{A_j}\|_1 = \|\mathbf{p}_{A_j}\|_2$ for all j, and thus that $\mathbf{p}$ has (at most) one non-zero element in each A_j. Therefore, testing identity to $\mathbf{p}$ would boil down to testing identity on a distribution with support size T, which can be done with $O(\sqrt{T}/\varepsilon^2)$ samples.

This is not actually the case, of course: the right-hand-side is only small and not exactly zero. Yet, one can show that a robust version of the above holds, making this intuition precise: in Lemma 3.2.28, we show that on average, *most* of the probability mass of $\mathbf{p}$ is concentrated on a single point from each A_j. This sparsity implies that testing identity to $\mathbf{p}$ on this set of T points is indeed enough – leading to the theorem.

Proof of Theorem 3.2.27 Let $\mathbf{p} \in \Delta([n])$ be a fixed, known distribution, assumed monotone non-increasing without loss of generality: $\mathbf{p}_1 \geq \mathbf{p}_2 \geq \cdots \geq \mathbf{p}_n$. Given $\varepsilon \in (0, 1/2)$, we let t_ε be as above, namely such that

$$\kappa_{\mathbf{p}}(t_\varepsilon) \geq 1 - 2\varepsilon.$$

From this, it follows by Lemma 3.2.15 that

$$\|\mathbf{p}\|_{Q(T)} \geq 1 - 2\varepsilon, \tag{3.15}$$

where we set $T \stackrel{\text{def}}{=} 2t_\varepsilon^2$. Choose $A_1, \ldots, A_T$ to be a partition of $[n]$ achieving the maximum (since we are in the finite, discrete case) defining $\|\mathbf{p}\|_{Q(T)}$; and let $\tilde{\mathbf{p}}$ be the subdistribution on T elements defined as follows. For each $j \in [T]$, choose $i_j \stackrel{\text{def}}{=} \arg\max_{i \in A_j} \mathbf{p}_i$, and set $\tilde{\mathbf{p}}(j) \stackrel{\text{def}}{=} \mathbf{p}(i_j)$.

Lemma 3.2.28 (Sparsity Lemma)**.** *There exists an absolute constant $\kappa > 0$ such that $\tilde{\mathbf{p}}([T]) = \sum_{j=1}^{T} \mathbf{p}(i_j) \geq 1 - \kappa\varepsilon$. (Moreover, one can take $\kappa \stackrel{\text{def}}{=} \frac{2}{3-\sqrt{7}} \simeq 5.65$.)*

Proof. Fix any $j \in [T]$, and for convenience let $A \stackrel{\text{def}}{=} A_j$. Write a^* for the maximum element for $\mathbf{p}$ in A, so that $\mathbf{p}(i_j) = \max_{a \in A} \mathbf{p}(a) = \mathbf{p}(a^*)$. We have by monotonicity $\mathbf{p}(A) \geq \sqrt{\sum_{a \in A} \mathbf{p}(a)^2}$, and moreover, letting $\alpha \stackrel{\text{def}}{=} \mathbf{p}(A) - \mathbf{p}(a^*) = \mathbf{p}(A \setminus \{a^*\})$,

$$\mathbf{p}(A) - \sqrt{\sum_{a \in A} \mathbf{p}(a)^2} = \mathbf{p}(a^*) + \alpha - \sqrt{\mathbf{p}(a^*)^2 + \sum_{a \neq a^*} \mathbf{p}(a)^2} \geq \mathbf{p}(a^*) + \alpha - \sqrt{\mathbf{p}(a^*)^2 + \alpha^2}.$$

We let $s > 1$ be a (non-integer) parameter to be chosen later. Suppose first that $\alpha \leq \frac{s}{s+1} p(A)$, or equivalenty

$\alpha \le s\mathbf{p}(a^*)$. In that case, we have

$$\mathbf{p}(A) - \sqrt{\sum_{a\in A}\mathbf{p}(a)^2} \ge \mathbf{p}(a^*) + \alpha - \mathbf{p}(a^*)\sqrt{1+\left(\frac{\alpha}{\mathbf{p}(a^*)}\right)^2} \ge \mathbf{p}(a^*) + \alpha - \mathbf{p}(a^*)\left(1 + \frac{\sqrt{s^2+1}-1}{s}\frac{\alpha}{\mathbf{p}(a^*)}\right)$$
$$= \left(1 - \frac{\sqrt{s^2+1}-1}{s}\right)\alpha \stackrel{\text{def}}{=} L_1(s)\alpha$$

where we relied on the inequality $\sqrt{1+x^2} \le 1 + \frac{\sqrt{s^2+1}-1}{s}x$ for $x \in [0,s]$. However, if $\alpha > s\mathbf{p}(a^*)$, then we have

$$\mathbf{p}(A) - \sqrt{\sum_{a\in A}\mathbf{p}(a)^2} = \mathbf{p}(a^*) + \alpha - \sqrt{\mathbf{p}(a^*)^2 + \sum_{a\neq a^*}\mathbf{p}(a)^2} \ge \alpha - \sqrt{\sum_{a\neq a^*}\mathbf{p}(a)^2}$$
$$\ge \alpha - \sqrt{\lfloor s\rfloor\left(\frac{\alpha}{s}\right)^2 + 1\cdot\left(\alpha - \frac{\lfloor s\rfloor}{s}\alpha\right)^2} = \left(1 - \sqrt{\frac{\lfloor s\rfloor}{s^2} + \left(1-\frac{\lfloor s\rfloor}{s}\right)^2}\right)\alpha \stackrel{\text{def}}{=} L_2(s)\alpha.$$

using the fact that $\mathbf{p}(a^*)$ is the maximum probability value of any element, so that the total α has to be spread among at least $\lfloor s\rfloor + 1$ elements (recall that s will be chosen not to be an integer). Optimizing these two bounds leads to the choice of $s \stackrel{\text{def}}{=} \frac{4+\sqrt{7}}{3} \notin \mathbb{N}$, for which $L_1(s) = L_2(s) = 3 - \sqrt{7} \simeq 0.35$.

Putting it together, we obtain, summing over all $j \in [T]$, that

$$1 - \|\mathbf{p}\|_{Q(T)} = \sum_{j=1}^{T}\mathbf{p}(A_j) - \sum_{j=1}^{T}\sqrt{\sum_{i\in A_j}\mathbf{p}(i)^2} = \sum_{j=1}^{T}\left(\mathbf{p}(A_j) - \sqrt{\sum_{i\in A_j}\mathbf{p}(i)^2}\right) \ge (3-\sqrt{7})\sum_{j=1}^{T}\left(\mathbf{p}(A_j) - \mathbf{p}(i_j)\right)$$
$$= (3-\sqrt{7})\left(1 - \tilde{\mathbf{p}}([T])\right)$$

which implies $\tilde{\mathbf{p}}([T]) \ge \frac{1}{3-\sqrt{7}}\|\mathbf{p}\|_{Q(T)} - \frac{1}{3-\sqrt{7}} + 1 \ge 1 - \frac{2}{3-\sqrt{7}}\varepsilon$ by Eq. (3.15). □

Lemma 3.2.29. *Fix $\mathbf{p}$, ε as above, let $S \stackrel{\text{def}}{=} \{i_1, \dots, i_T\}$ be the corresponding set of T elements, and take κ as in Lemma 3.2.28. For any $\mathbf{q} \in \Delta([n])$, if (i) $\sum_{j=1}^{T}\mathbf{q}(i_j) \ge 1 - (\kappa + \frac{1}{3})\varepsilon$ and (ii) $\sum_{j=1}^{T}\left|\frac{\tilde{\mathbf{p}}(j)}{\mathbf{p}(S)} - \frac{\tilde{\mathbf{q}}(j)}{\mathbf{q}(S)}\right| \le \frac{1}{3}\varepsilon$, then $\|\mathbf{p}-\mathbf{q}\|_1 \le (3\kappa+1)\varepsilon$.*

Proof. Unrolling the definition, and as $\mathbf{p}(\bar{S}) \leq \kappa\varepsilon$ by Lemma 3.2.28,

$$\begin{aligned}
\|\mathbf{p}-\mathbf{q}\|_1 &= \sum_{i=1}^{n} |\mathbf{p}(i)-\mathbf{q}(i)| = \sum_{j=1}^{T} |\mathbf{p}(i_j)-\mathbf{q}(i_j)| + \sum_{i\notin S} |\mathbf{p}(i)-\mathbf{q}(i)| \leq \sum_{j=1}^{T} |\mathbf{p}(i_j)-\mathbf{q}(i_j)| + \mathbf{p}(\bar{S}) + \mathbf{q}(\bar{S}) \\
&\leq \sum_{j=1}^{T} |\mathbf{p}(i_j)-\mathbf{q}(i_j)| + \kappa\varepsilon + (\kappa+\tfrac{1}{3})\varepsilon = \sum_{j=1}^{T} \left|\mathbf{p}(S)\frac{\tilde{\mathbf{p}}(j)}{\mathbf{p}(S)} - \mathbf{q}(S)\frac{\tilde{\mathbf{q}}(j)}{\mathbf{q}(S)}\right| + (2\kappa+\tfrac{1}{3})\varepsilon \\
&\leq \mathbf{p}(S)\sum_{j=1}^{T}\left|\frac{\tilde{\mathbf{p}}(j)}{\mathbf{p}(S)} - \frac{\tilde{\mathbf{q}}(j)}{\mathbf{q}(S)}\right| + \sum_{j=1}^{T}\frac{\tilde{\mathbf{q}}(j)}{\mathbf{q}(S)}|\mathbf{p}(S)-\mathbf{q}(S)| + (2\kappa+\tfrac{1}{3})\varepsilon \\
&= \mathbf{p}(S)\cdot\sum_{j=1}^{T}\left|\frac{\tilde{\mathbf{p}}(j)}{\mathbf{p}(S)} - \frac{\tilde{\mathbf{q}}(j)}{\mathbf{q}(S)}\right| + |\mathbf{p}(S)-\mathbf{q}(S)| + (2\kappa+\tfrac{1}{3})\varepsilon \\
&\leq \tfrac{1}{3}\varepsilon + (\kappa+\tfrac{1}{3})\varepsilon + (2\kappa+\tfrac{1}{3})\varepsilon = (3\kappa+1)\varepsilon
\end{aligned}$$

concluding the proof of the lemma. □

Let $\kappa > 0$ be the constant from Lemma 3.2.28. We let $\varepsilon' \stackrel{\text{def}}{=} \frac{\varepsilon}{3\kappa+1}$, and $T \stackrel{\text{def}}{=} 2t_{\varepsilon'}^2$, $\{i_1,\ldots,i_T\} \subseteq [n]$ the corresponding value and elements (i.e., T and the i_j's are as in the foregoing discussion (chosen with regard to ε' and the known distribution $\mathbf{p}$)). For convenience, denote by $\tilde{\mathbf{q}}$ the (unknown) subdistribution on $[T]$ defined by $\tilde{\mathbf{q}}(j) \stackrel{\text{def}}{=} \mathbf{q}(i_j)$ for $j \in [T]$.

We first verify that $\tilde{\mathbf{q}}([T]) \geq 1-\kappa\varepsilon'$, with $O(1/\varepsilon')$ samples (specifically, we distinguish, with probability at least $9/10$, between $\tilde{\mathbf{q}}([T]) \geq 1-\kappa\varepsilon'$ and $\tilde{\mathbf{q}}([T]) \leq 1-(\kappa+\frac{1}{3})\varepsilon'$; and reject in the latter case). Once this is done, we apply one of the known identity testing algorithms to $\bar{\mathbf{p}}, \bar{\mathbf{q}} \in \Delta([T])$, renormalized versions of $\tilde{\mathbf{p}}, \tilde{\mathbf{q}}$:

$$\bar{\mathbf{p}} = \frac{\tilde{\mathbf{p}}}{\tilde{\mathbf{p}}([T])}, \quad \bar{\mathbf{q}} = \frac{\tilde{\mathbf{q}}}{\tilde{\mathbf{q}}([T])}$$

using rejection sampling (note that we have the explicit description of $\bar{\mathbf{p}}$; and, since $\tilde{\mathbf{q}}([T]) \geq 1-(\kappa+\frac{1}{3})\varepsilon'$ (conditioning on the first test meeting its guarantee), we can obtain m independent samples from $\bar{\mathbf{q}}$ with an expected $O(m)$ number of samples from $\mathbf{q}$). This is done with parameter ε' and failure probability $1/10$; and costs $O\left(\frac{\sqrt{T}}{\varepsilon'^2}\right) = O\left(\frac{t_{\varepsilon'}}{\varepsilon'^2}\right)$ samples from $\mathbf{q}$.

Turning to the correctness: we condition on both tests meeting their guarantees, which by a union bound holds with probability at least $4/5$.

- If $\mathbf{p} = q$, then $\mathbf{q}(S) = \mathbf{p}(S) \geq 1-\kappa\varepsilon'$, and $\bar{\mathbf{q}} = \bar{\mathbf{p}}$: neither the first nor the second test reject, and the overall algorithm accepts.
- If the algorithm accepts, then $\mathbf{q}(S) \geq 1-(\kappa+\frac{1}{3})\varepsilon'$ (by the first test) and $\sum_{j=1}^{T}\left|\frac{\tilde{\mathbf{p}}(j)}{\mathbf{p}(S)} - \frac{\tilde{\mathbf{q}}(j)}{\mathbf{p}(S)}\right| \leq \varepsilon'$ (by the second): Lemma 3.2.29 then guarantees that $\|\mathbf{p}-\mathbf{q}\|_1 \leq 3\kappa+1\varepsilon' = \varepsilon$.

Observing that for $\kappa = \frac{2}{3-\sqrt{7}}$ (as suggested by Lemma 3.2.28) we have $3\kappa+1 \leq 18$ establishes the last part of the theorem.

Remark 3.2.30. We observe that, although efficiently computing $\kappa_{\mathbf{p}}(\cdot)$ (and *a fortiori* $\kappa_{\mathbf{p}}^{-1}(\cdot)$) or $\|\mathbf{p}\|_{Q(\cdot)}$ is not

immediate, the above algorithm *is* efficient, and can be implemented to run in time $O(n + T \log n + \sqrt{T}/\varepsilon^2)$. The reason is that knowing beforehand the value of T is not necessary: given $\mathbf{p}$ (e.g., as an unsorted sequence of n values) and ε, it is enough to retrieve the biggest values of $\mathbf{p}$ until they sum to $1 - O(\varepsilon)$: the number of elements retrieved will, by our proof, be at most T (and this can be done in time $O(n + T \log n)$ by using e.g. a max-heap). It only remains to apply the above testing algorithm to the set of (at most) T elements thus obtained.

3.2.6.3 Tightening the Lower Bound

As a last step, one may want to strenghten the lower bound obtained by the communication complexity reduction of Theorem 3.2.21. We here describe how this can be achieved using more standard arguments from distribution testing. However, we stress that these arguments in some sense are applicable "'after the fact," that is after Section 3.2.6.1 revealed the connection to the K-functional, and the bound we should aim for. Specifically, we prove the following:

Theorem 3.2.31. *For any $\mathbf{p} \in \Delta([n])$, and any $\varepsilon \in (0, 1/2)$ any algorithm testing identity to $\mathbf{p}$ must have sample complexity $\Omega\left(\frac{t_\varepsilon}{\varepsilon}\right)$.*

Proof. Fix $\mathbf{p} \in \Delta(\Omega)$ and $\varepsilon \in (0, 1/2)$ as above, and consider the corresponding value t_ε; we assume that $t_\varepsilon \geq 2$, as otherwise there is nothing to prove.[21] Without loss of generality – as we could always consider a sufficiently small approximation, and take the limit in the end, we further assume the infimum defining $\kappa_{\mathbf{p}}$ is attained: let $h, \ell \in [0, 1]^n$ be such that $\mathbf{p} = h + \ell$ and $\kappa_{\mathbf{p}}(t_\varepsilon) = \|h\|_1 + t_\varepsilon \|\ell\|_2 = 1 - 2\varepsilon$.

Since $\|\ell\|_1 = 1 - \|h\|_1$, from the definition of h, ℓ, we have that $1 - 2\varepsilon = 1 - \|\ell\|_1 + t_\varepsilon \|\ell\|_2$, from which

$$0 < \|\ell\|_2 = \frac{\|\ell\|_1 - 2\varepsilon}{t_\varepsilon} \leq \frac{1}{t_\varepsilon} \tag{3.16}$$

(note that the right inequality is strict because $\varepsilon > 0$: since if $\|\ell\|_2 = 0$, then $\|\ell\|_1 = 0$ and $h = \mathbf{p}$; but then $\kappa_{t_\varepsilon} = \|\mathbf{p}\|_1 = 1$.) In particular, this implies $\|\ell\|_1 - 2\varepsilon > 0$.

With this in hand, we will apply the following theorem, due to Valiant and Valiant:

Theorem 3.2.32 ([169, Theorem 4])**.** *Given a distribution $\mathbf{p} \in \Delta(\Omega)$, and associated values $(\varepsilon_i)_{i \in [n]}$ such that $\varepsilon_i \in [0, \mathbf{p}_i]$ for each i, define the distribution over distributions $\mathcal{Q}$ by the process: independently for each domain element i, set uniformly at random $\mathbf{q}_i = \mathbf{p}_i \pm \varepsilon_i$, and then normalize $\mathbf{q}$ to be a distribution. Then there exists a constant $c > 0$ such that is takes at least $c\left(\sum_{i=1}^n \varepsilon_i^4/\mathbf{p}_i^2\right)^{-1/2}$ samples to distinguish $\mathbf{p}$ from $\mathcal{Q}$ with success probability $2/3$. Further, with probability at least $1/2$ the ℓ_1 distance between $\mathbf{p}$ and a uniformly random distribution from $\mathcal{Q}$ is at least $\min\left(\sum_{i=1}^n \varepsilon_i - \max_i \varepsilon_i, \frac{1}{2}\sum_{i=1}^n \varepsilon_i\right)$.*

We want to invoke the above theorem with ℓ being, roughly speaking, the "random perturbation" to $\mathbf{p}$. Indeed, since ℓ has small ℓ_2 norm of order $O(1/t_\varepsilon)$ by (3.16) (which gives a good lower bound) and has ℓ_1

[21] Indeed, an immediate lower bound of $\Omega(1/\varepsilon)$ on this problem holds.

sum $\Omega(\varepsilon)$ (which gives distance), this seems to be a natural choice.
In view of this, set $\alpha \stackrel{\text{def}}{=} \frac{2\varepsilon}{\|\ell\|_1} \in (0,1)$ and, for $i \in [n]$, $\varepsilon_i \stackrel{\text{def}}{=} \alpha\ell_i \leq \ell_i \in [0, \mathbf{p}_i]$. Theorem 3.2.31 will then be a direct consequence of the next two claims:

Claim 3.2.33 (Distance)**.** *We have* $\min\left(\sum_{i=1}^n \varepsilon_i - \max_i \varepsilon_i, \frac{1}{2}\sum_{i=1}^n \varepsilon_i\right) \geq \varepsilon$.

Proof. Since by our choice of α it is immediate that $\sum_{i=1}^n \varepsilon_i = \frac{2\varepsilon}{\|\ell\|_1}\sum_{i=1}^n \ell_i = 2\varepsilon$, it suffices to show that $\max_i \varepsilon_i \leq \varepsilon$, or equivalently that $\max_i \ell_i \leq \frac{1}{2}\|\ell\|_1$. But this follows from the fact that $\|\ell\|_\infty \leq \|\ell\|_2 \leq \frac{\|\ell\|_1}{t_\varepsilon}$, and our assumption that $t_\varepsilon \geq 2$. □

It then remains to analyze the lower bound obtained through the application of Theorem 3.2.32:

Claim 3.2.34 (Lower bound)**.** *With the* ε_i*'s defined as before,* $\left(\sum_{i=1}^n \varepsilon_i^4/\mathbf{p}_i^2\right)^{-1/2} \geq \frac{2t_\varepsilon}{\varepsilon}$.

Proof. Unrolling the definition of the ε_i's,

$$\sum_{i=1}^n \frac{\varepsilon_i^4}{\mathbf{p}_i^2} = \alpha^4 \sum_{i=1}^n \frac{\ell_i^4}{\mathbf{p}_i^2} = \alpha^4 \sum_{i=1}^n \frac{\ell_i^2}{\mathbf{p}_i^2}\ell_i^2 \leq \alpha^4 \sum_{i=1}^n \ell_i^2 = \frac{2^4\varepsilon^4}{\|\ell\|_1^4}\|\ell\|_2^2 = \left(\frac{4\varepsilon^2}{\|\ell\|_1^2}\frac{\|\ell\|_1 - 2\varepsilon}{t_\varepsilon}\right)^2$$

where the last equality is (3.16). This yields

$$\left(\sum_{i=1}^n \frac{\varepsilon_i^4}{\mathbf{p}_i^2}\right)^{-1/2} \geq \frac{t_\varepsilon}{4\varepsilon^2}\cdot\frac{\|\ell\|_1^2}{\|\ell\|_1 - 2\varepsilon} = \frac{t_\varepsilon}{2\varepsilon}\cdot\frac{\left(\frac{\|\ell\|_1}{2\varepsilon}\right)^2}{\frac{\|\ell\|_1}{2\varepsilon} - 1} \geq \frac{2t_\varepsilon}{\varepsilon}$$

where the last inequality comes from $f\colon x > 1 \mapsto \frac{x^2}{x-1}$ achieving its minimum, 4, at $x = 2$. □

Combining the two claims with Theorem 3.2.32 implies, by a standard argument, the lower bound of Theorem 3.2.31. □

Remark 3.2.35. A straightforward modification of the proof of Theorem 3.2.31 allows one to prove a somewhat more general statement, namely a lower bound of $\Omega\left(\gamma t_\gamma/\varepsilon^2\right)$ for any $\gamma \in [\varepsilon, 1/2]$ such that $t_\gamma \geq 2$. In particular, this implies an incomparable bound of $\Omega\left(t_{1/4}/\varepsilon^2\right)$ as long as $\mathbf{p}$ does not put almost all its probability weight on $O(1)$ elements.

On the optimality of our bound. We conclude this section by briefly discussing the optimality of our bound, and specifically whether one could hope to strenghten Theorem 3.2.31 to obtain an $\Omega\left(t_\varepsilon/\varepsilon^2\right)$ lower bound. Unfortunately, it is easy to come up with simple (albeit contrived) counterexamples: e.g., fix $\varepsilon \in (0, 1/3)$, and let $\mathbf{p} \in \Delta([n])$ be the distribution that puts mass $1 - 3\varepsilon$ on the first element and uniformly spreads the rest among the remaining $n - 1$ elements. A straightforward calculation shows that, for this distribution $\mathbf{p} = \mathbf{p}(\varepsilon)$, one has $\kappa_{\mathbf{p}}^{-1}(1 - 2\varepsilon) = \Theta(\sqrt{n})$; and it is not hard to check that one can indeed test identity to $\mathbf{p}$ with $O(\sqrt{n}/\varepsilon)$ samples only,[22] and so the $\Omega(t_\varepsilon/\varepsilon)$ lower bound is tight in this case.

[22] Indeed, any distribution $\mathbf{q}$ such that $\|\mathbf{q} - \mathbf{p}\|_1 > \varepsilon$ must either be such that $|\mathbf{p}(1) - \mathbf{q}(1)| = \Omega(\varepsilon)$ or $\left|\mathbf{p}|_{[n]\setminus\{1\}} - \mathbf{q}|_{[n]\setminus\{1\}}\right| =$

Although this specific instance is somewhat unnatural, as it fails to be a counterexample for any distance parameter $\varepsilon' \ll \varepsilon$, it does rule out an improvement of Theorem 3.2.31 for the full range of parameters. On the other hand, it is also immediate to see that the upper bound $O\big(t_\varepsilon/\varepsilon^2\big)$ cannot be improved in general, as demonstrated by choosing $\mathbf{p}$ to be the uniform distribution (yet, in this case, the extension provided by Remark 3.2.35 does provide the optimal bound).

3.2.7 Lower Bounds on Other Properties

In this section we demonstrate how our methodology can be used to easily obtain lower bounds on the sample complexity of various properties of distributions. To this end, we provide sketches of proofs of lower bounds for monotonicity testing, k-modality, and the "symmetric sparse support" property (that we define below). We remark that using minor variations on the reductions presented in Section 3.2.4 and Section 3.2.6, it is also straightforward to obtain lower bounds for properties of distributions such as being binomially distributed, Poisson binomially distributed, and having a log-concave probability mass function. Throughout this section, we fix ε to be a small constant and refer to testing with respect to proximity $\Theta(\varepsilon)$.

Monotonicity on the integer line and the Boolean hypercube. We start with the problem of testing monotonicity on the integer line, that is, testing whether a distribution $\mathbf{p} \in \Delta([n])$ has a monotone probability mass function. Consider the "vanilla" reduction, presented in Section 3.2.4. Note that for yes-instances, we obtain the uniform distribution, which is monotone. For no-instances, however, we obtain a distribution $\mathbf{p}$ that has mass $1/n$ on a $(1-\varepsilon)$-fraction of the domain, is unsupported on a $(\varepsilon/2)$-fraction of the domain, and has mass $2/n$ on the remaining $(\varepsilon/2)$-fraction. Typically, $\mathbf{p}$ is $\Omega(1)$-far from being monotone; however, it could be the case that the first (respectively, last) $\varepsilon n/2$ elements are of 0 mass, and the last (respectively, first) $\varepsilon n/2$ elements are of mass $2/n$, in which case $\mathbf{p}$ is perfectly monotone. To remedy this, all we have to do is let the referee emulate a distribution $\mathbf{p}' \in \Delta([3n])$ such that $\mathbf{p}'_i = \begin{cases} \frac{1}{3}\mathbf{p}_{i-n} & i \in \{n+1,\ldots,2n\} \\ \frac{1}{3n} & \text{otherwise} \end{cases}$. It is immediate to see that the probability mass functions of $\mathbf{p}'$ is $(\varepsilon/3)$-far from monotone.

The idea above can be extended to monotonicity over the hypercube as follows. We start with the uniformity reduction, this time over the domain $\{0,1\}^n$. As before, yes-instances will be mapped to the uniform distribution over the hypercube, which is monotone, and no-instances will be mapped to a distribution that has mass $1/2^n$ on a $(1-\varepsilon)$-fraction of the domain, is unsupported on a $(\varepsilon/2)$-fraction of the domain, and has mass $1/2^{n-1}$ on the remaining $(\varepsilon/2)$-fraction – but could potentially be monotonously *strictly* increasing (or decreasing). This time, however, the "boundary" is larger than the "edges" of the integer line, and we cannot afford to pad it with elements of weight $1/2^n$. Instead, the referee, who receives for the players samples drawn from a distribution $\mathbf{p} \in \Delta(\{0,1\}^n)$, emulates a distribution $\mathbf{p}'' \in \Delta\big(\{0,1\}^{n+1}\big)$ over a larger hypercube whose additional coordinate determines between a negated or regular copy of $\mathbf{p}$; that is,

$\Omega(1)$. The first case only takes $O(1/\varepsilon)$ samples, while the second can be achieved by rejection sampling with $O(1/\varepsilon) \cdot O(\sqrt{n})$ samples.

$\mathbf{p}'(z) = \begin{cases} \mathbf{p}(z) & z_1 = 0 \\ \frac{1}{2^{n-1}} - \mathbf{p}(z) & z_1 = 1 \end{cases}$ (where the referee chooses $z_1 \in \{0,1\}$ independently and uniformly at random for each new sample). Hence, even if $\mathbf{p}$ is monotonously increasing (or decreasing), the emulated distribution $\mathbf{p}''$ is $\Omega(\varepsilon)$-far from monotone. By the above, we obtain $\tilde{\Omega}(\sqrt{n})$ and $\tilde{\Omega}(2^{n/2})$ lower bounds on the sample complexity of testing monotonicity on the line and on the hypercube, respectively.

k-modality. Recall that a distribution $\mathbf{p} \in \Delta([n])$ is said to be *k-modal* if its probability mass function has at most k "peaks" and "valleys." Such distributions are natural generalizations of monotone (for $k = 0$) and unimodal (for $k = 1$) distributions. Fix a sublinear k, and consider the uniformity reduction presented in Section 3.2.4, with the additional step of letting the prover apply a random permutation to the domain $[n]$ (similarly to the reduction shown in Section 3.2.4.1). Note that yes-instances are still mapped to the uniform distribution (which is clearly k-modal), and no-instances are mapped to distributions with mass $1/n$, $2/n$, and 0 on a $(1-\varepsilon)$, $(\varepsilon/2)$, and $(\varepsilon/2)$ (respectively) fractions of the domain. Intuitively, applying a random permutation of the domain to such a distribution "spreads" the elements with masses 0 and $2/n$ nearly uniformly, causing many level changes (i.e., high modality); indeed, it is straightforward to verify that with high probability over the choice of a random permutation of the domain, such a distribution will indeed be $\Omega(\varepsilon)$-far from k-modal. This yields an $\tilde{\Omega}(\sqrt{n})$ lower bound on the sample complexity of testing k-modality, nearly matching the best known lower bound of $\Omega(\max(\sqrt{n}, k/\log k))$ following from [43], for $k/\log(k) = O(\sqrt{n})$.

Symmetric sparse support. Consider the property of distributions $\mathbf{p} \in \Delta([n])$ such that when projected to its support, $\mathbf{p}$ is mirrored around the middle of the domain. That is, $\mathbf{p}$ is said to have a *symmetric sparse support* if there exists $S = \{i_0 < i_2 < \cdots < i_{2\ell}\} \subseteq [n]$ with $i_\ell = \frac{n}{2}$ such that: (1) $\mathbf{p}(i) = 0$ for all $i \in [n] \setminus S$, and (2) $\mathbf{p}(i_{\ell+1-j}) = \mathbf{p}(i_{\ell+j})$ for all $0 \leq j \leq \ell$. We sketch a proof of an $\tilde{\Omega}(\sqrt{n})$ lower bound on the sample complexity of testing this property. Once again, we shall begin with the uniformity reduction presented in Section 3.2.4, obtaining samples from a distribution $\mathbf{p} \in \Delta([n/2])$. Then the referee emulates samples from the distribution $\mathbf{p}' \in \Delta([n])$ that is distributed as $\mathbf{p}$ on its left half, and uniformly distributed on its right half; that is, $\mathbf{p}'_i = \begin{cases} \mathbf{p}_i/2 & i \in [n] \\ 1/n & \text{otherwise} \end{cases}$. Note that yes-instances are mapped to the uniform distribution, which has symmetric sparse support, and no-instances are mapped to distributions in which the right half is uniformly distributed and the left half contains $\varepsilon n/2$ elements of mass $2/n$, and hence it is $\Omega(\varepsilon)$-far from having symmetric sparse support.

Other properties. As aforementioned, similar techniques as in the reductions above (as well as in the identity testing reduction of Section 3.2.6, invoked on a specific $\mathbf{p}$, e.g., the $\mathrm{Bin}(n, 1/2)$ distribution) can be applied to obtain nearly-tight lower bounds of $\tilde{\Omega}(\sqrt{n})$ (respectively $\tilde{\Omega}(n^{1/4})$) for the properties of being log-concave and monotone hazard rate (respectively Binomially and Poisson Binomially distributed). See

e.g., [51] for the formal definitions of these properties.

3.2.8 Testing with Conditional Samples

In this section we show that reductions from communication complexity protocols can be used to obtain lower bounds on the sample complexity of distribution testers that are augmented with conditional samples. These testing algorithms, first introduced in [54, 48], aim to address scenarios that arise both in theory and practice yet are not fully captured by the standard distribution testing model.

In more detail, algorithms for testing with conditional samples are distribution testers that, in addition to sample access to a distribution $\mathbf{p} \in \Delta(\Omega)$, can ask for samples from $\mathbf{p}$ conditioned on the sample belonging to a subset $S \subseteq \Omega$. It turns out that testers with conditional samples are much stronger than standard distribution testers, leading in many cases to exponential savings (or even more) in the sample complexity. In fact, these testing algorithms can often maintain their power even if they only have the ability to query subsets of a particular structure.

One of the most commonly studied restricted conditional samples models is the PAIRCOND model [49]. In this model, the testers can either obtain standard samples from $\mathbf{p}$, or specify two distinct indices $i, j \in \Omega$ and get a sample from $\mathbf{p}$ conditioned on membership in $S = \{i, j\}$. As shown in [49, 42], even under this restriction one can obtain constant- or $\operatorname{poly}\log(n)$-query testers for many properties, such as uniformity, identity, closeness, and monotonicity (all of which require $\Omega(\sqrt{n})$ or more samples in the standard sampling setting). This, along with the inherent difficulty of proving hardness results against *adaptive* algorithms, makes proving lower bounds in this setting a challenging task; and indeed the PAIRCOND lower bounds established in the aforementioned works are quite complex and intricate.

We will prove, via a reduction from communication complexity, a strong lower bound on the sample complexity of any PAIRCOND algorithm for testing *junta distributions*, a class of distributions introduced in [8] (see definition below).

Since PAIRCOND algorithms are stronger than standard distribution testers (in particular, they can make adaptive queries), we shall reduce from the general randomized communication complexity model (rather than from the SMP model, as we did for standard distribution testers). In this model, Alice and Bob are given inputs x and y as well as a common random string, and the parties aim to compute a function $f(x, y)$ using the minimum amount of communication.

We say that a distribution $\mathbf{p} \in \Delta(\{0,1\}^n)$ is a *k-junta distribution* (with respect to the uniform distribution) if its probability mass function is only influenced by k of its variables. We outline below a proof of the following lower bound.

Theorem 3.2.36. *Every* PAIRCOND *algorithm for testing k-junta distributions must make $\Omega(k)$ queries.*

Sketch of proof. We closely follow the k-linearity lower bound in [33] and reduce from the unique $(k/2)$-disjointness problem. In this promise problem, Alice and Bob get inputs $x \in \{0,1\}^n$ and $y \in \{0,1\}^n$

(respectively) of Hamming weight $k/2$ each, and the parties are required to decide whether $\sum_{i=1}^{n} x_i y_i = 1$ or $\sum_{i=1}^{n} x_i y_i = 0$. It is well-known that in every randomized protocol for this problem the parties must communicate $\Omega(k)$ bits.

Assume there exists a PAIRCOND algorithm for testing k-junta distributions, with query complexity $\mathbf{q}$. The reduction is as follows. Alice sets $A = \{\, i \in [n] : \; x_i = 1 \,\}$ and considers the character function $\chi_A(z) = \oplus_{i \in A} z_i$, and similarly Bob sets $B = \{\, i \in [n] : \; y_i = 1 \,\}$ and considers the character function $\chi_B(z) = \oplus_{i \in B} z_i$. Both players then invoke the tester for k-junta distributions, feeding it samples emulated from the distribution $\mathbf{p} \in \Delta(\{0,1\}^n)$ given by $\mathbf{p}(z) = \chi_{A \triangle B}(z)/2^{n-1}$ (where $\chi_{A \triangle B}(z) = \oplus_{i \in A \triangle B} z_i$); note that since the non-zero character functions are balanced, $\mathbf{p}$ is indeed a probability distribution. Recall that each query of a PAIRCOND algorithm is performed by either setting $S = \{0,1\}^n$, or choosing $z, z' \in \{0,1\}^n$ and setting $S = \{z, z'\}$, then sampling from $\mathbf{p}|_S$. The players emulate each PAIRCOND query by the following rejection sampling procedure:

Sampling query ($S = \{0,1\}^n$): Alice and Bob proceed as follows.

1. Choose $z \in S$ uniformly at random, using shared randomness;
2. Exchange $\chi_A(z)$ and $\chi_B(z)$ between the players, and compute $\chi_{A \triangle B}(z) = \chi_A(z) \cdot \chi_B(z)$;
3. If $\chi_{A \triangle B}(z) = 1$, feed the tester with the sample z. Otherwise repeat the process.

Note that since $\chi_{A \triangle B}(z)$ is a balanced function, then on expectation each PAIRCOND query to $\mathbf{p}$ can be emulated by exchanging $O(1)$ bits.

Pairwise query ($S = \{z, z'\}$ for some $z, z' \in \{0,1\}^n$): exchange $\chi_A(z), \chi_A(z')$ and $\chi_B(z), \chi_B(z')$ between the players, compute $\chi_{A \triangle B}(z)$ and $\chi_{A \triangle B}(z')$, and use shared randomness to sample from S with the corresponding (now fully known) conditional probabilities.

The above gives a protocol with *expected* communication complexity $O(q)$, correct with probability $5/6$. To convert it to a honest-to-goodness protocol with communication complexity $O(q)$ and success probability $2/3$, it suffices for Alice and Bob to run the above protocol and stop (and output reject) as soon as they go over Ck bits of communication, for some absolute constant $C > 0$. An application of Markov's inequality guarantees that this happens with probability at most $1/6$, yielding the claimed bound on the error probability of the protocol.

Finally, note that on the one hand, if (x, y) is such that $\sum_{i=1}^{n} x_i y_i = 0$, then $\chi_{A \triangle B}(z)$ is a degree-k character, and in particular, a k-junta. Hence, by definition $\mathbf{p}$ is a k-junta distribution. On the other hand, if (x, y) is such that $\sum_{i=1}^{n} x_i y_i = 1$, then $\chi_{A \triangle B}(z)$ is a degree-$(k-2)$ character, which in particular disagrees with every k-junta on $\Omega(1)$-fraction of the inputs. Therefore, since $\mathbf{p}$ is uniform over its support, we can deduce that that $\mathbf{p}$ is $\Omega(1)$-far in ℓ_1-distance from any k-junta distribution. $\square$

Chapter 4

Testing Properties of Distributions: Changing the Rules

> You may seek it with thimbles—and seek it with care;
> You may hunt it with forks and hope;
> You may threaten its life with a railway-share;
> You may charm it with smiles and soap–
>
> Lewis Carroll, *The Hunting of the Snark*

In the standard distribution testing setting considered so far, the "massive object" is an arbitrary probability distribution $\mathbf{p}$ over an n-element set, and the algorithm accesses the distribution by drawing independent samples from it. One broad insight that has emerged from this past decade of work in this setting is that, while sublinear-sample algorithms do exist for many distribution testing problems, the number of samples required remains in general quite large. Indeed, even the basic problem of testing whether $\mathbf{p}$ is the uniform distribution $\mathbf{u}$ over $[n]$ versus ε-far from uniform requires $\Omega_\varepsilon(\sqrt{n})$ samples, and most other problems have sample complexities at least this high, and in some cases *almost linear in the domain size* n [146, 174, 172]. Since such sample complexities could be and routinely are prohibitively high in real-world settings where n can be extremely large (see e.g. [22, 104, 127, 154], and references within), it is natural to explore problem variants where it may be possible for algorithms to succeed using fewer samples.

Indeed, researchers have studied distribution testing in settings where the unknown distribution is guaranteed to have some special structure, such as being monotone, k-modal or a "k-histogram" over $[n]$ [19, 74, 117], or being monotone over $\{0,1\}^n$ [155] or over other posets [30], and have obtained significantly more sample-efficient algorithms using these additional assumptions.

In this chapter we pursue a different line of investigation: rather than restricting the *class* of probability distributions under consideration, we consider testing algorithms that may use a more powerful form of *access* to the unknown distribution $\mathbf{p}$. In particular, we introduce and analyze two of these stronger types of access, the *conditional* and *extended* models (and some of their variants), where the algorithm can respectively obtain samples conditioned on certain events of its choosing, and inspect directly the probability mass or cumulative distribution function of the unknown probability distribution. The conditional sampling model will be the focus of Section 4.1; then, Section 4.2 contains the details of our work on the extended access model.

4.1 Conditional Sampling: Focusing on What Matters

4.1.1 Introduction

In this section, we consider our first generalization of the standard sampling model of distribution testing, granting the testing algorithms a more flexible access to the underlying probability distribution. This is a *conditional sampling oracle*, which allows the algorithm to obtain a draw from $\mathbf{p}_S$, the conditional distribution of $\mathbf{p}$ restricted to a subset S of the domain (where S is specified by the algorithm). More precisely, we have:

Definition 4.1.1. Fix a distribution $\mathbf{p}$ over $[n]$. A COND *oracle for* $\mathbf{p}$, denoted $\mathsf{COND}_{\mathbf{p}}$, is defined as follows: The oracle is given as input a *query set* $S \subseteq [n]$, chosen by the algorithm, that has $\mathbf{p}(S) > 0$. The oracle returns an element $i \in S$, where the probability that element i is returned is $\mathbf{p}_S(i) = \mathbf{p}(i)/\mathbf{p}(S)$, independently of all previous calls to the oracle.[1]

We remark that a recent work of Chakraborty et al. [54] introduced a very similar conditional model; we discuss their results and how they relate to ours in Section 4.1.1.2. For compatibility with our $\mathsf{COND}_{\mathbf{p}}$ notation we will write $\mathsf{SAMP}_{\mathbf{p}}$ to denote an oracle that takes no input and, each time it is invoked, returns an element from $[n]$ drawn according to $\mathbf{p}$ independently from all previous draws. This is the sample access to $\mathbf{p}$ that is used in the standard model of testing distributions, and this is of course the same as a call to $\mathsf{COND}_{\mathbf{p}}([n])$.

Motivation and Discussion. One purely theoretical motivation for the study of the COND model is that it may further our understanding regarding what forms of information (beyond standard sampling) can be helpful for testing properties of distributions. In both learning and property testing it is generally interesting to understand how much power algorithms can gain by making queries, and COND queries are a natural type of query to investigate in the context of distributions. As we discuss in more detail below, in several of our results we actually consider restricted versions of COND queries that do not require the full power of obtaining conditional samples from arbitrary sets.

A second attractive feature of the COND model is that it enables a new level of richness for algorithms that deal with probability distributions. In the standard model where only access to $\mathsf{SAMP}_{\mathbf{p}}$ is provided, all algorithms must necessarily be non-adaptive, with the same initial step of simply drawing a sample of points from $\mathsf{SAMP}_{\mathbf{p}}$, and the difference between two algorithms comes only from how they process their samples. In contrast, the essence of the COND model is to allow algorithms to *adaptively* determine later query sets S based on the outcomes of earlier queries.

A natural question about the COND model is its plausibility: are there settings in which an investigator could actually make conditional samples from a distribution of interest? We feel that the COND framework

[1] Note that as described above the behavior of $\mathsf{COND}_{\mathbf{p}}(S)$ is undefined if $\mathbf{p}(S) = 0$, i.e., the set S has zero probability under $\mathbf{p}$. While various definitional choices could be made to deal with this, we shall assume that in such a case, the oracle (and hence the algorithm) outputs "failure" and terminates. This will not be a problem for us throughout this paper, as (a) our lower bounds deal only with distributions that have $\mathbf{p}(i) > 0$ for all $i \in [n]$, and (b) in our algorithms $\mathsf{COND}_{\mathbf{p}}(S)$ will only ever be called on sets S which are "guaranteed" to have $\mathbf{p}(S) > 0$. (More precisely, each time an algorithm calls $\mathsf{COND}_{\mathbf{p}}(S)$ it will either be on the set $S = [n]$, or will be on a set S which contains an element i which has been returned as the output of an earlier call to $\mathsf{COND}_{\mathbf{p}}$.)

provides a reasonable first approximation for scenarios that arise in application areas (e.g., in biology or chemistry) where the parameters of an experiment can be adjusted so as to restrict the range of possible outcomes. For example, a scientist growing bacteria or yeast cells in a controlled environment may be able to deliberately introduce environmental factors that allow only cells with certain desired characteristics to survive, thus restricting the distribution of all experimental outcomes to a pre-specified subset. We further note that techniques which are broadly reminiscent of COND sampling have long been employed in statistics and polling design under the name of "stratified sampling" (see e.g. [177, 137]). We thus feel that the study of distribution testing in the COND model is well motivated both by theoretical and practical considerations.

Given the above motivations, the central question is whether the COND model enables significantly more efficient algorithms than are possible in the weaker SAMP model. Our results (see Section 4.1.1.1) show that this is indeed the case.

Before detailing our results, we note that several of them will in fact deal with a weaker variant of the COND model, which we now describe. In designing COND-model algorithms it is obviously desirable to have algorithms that only invoke the COND oracle on query sets S which are "simple" in some sense. Of course there are many possible notions of simplicity; in this work we consider the size of a set as a measure of its simplicity, and consider algorithms which only query small sets. More precisely, we consider the following restriction of the general COND model:

PAIRCOND **oracle:** We define a PAIRCOND (short for "pair-cond") *oracle for* $\mathbf{p}$ is a restricted version of $\mathsf{COND}_{\mathbf{p}}$ that only accepts input sets S which are either $S = [n]$ (thus providing the power of a $\mathsf{SAMP}_{\mathbf{p}}$ oracle) or $S = \{i, j\}$ for some $i, j \in [n]$, i.e. sets of size two. The PAIRCOND oracle may be viewed as a minimalist variant of COND that essentially permits an algorithm to compare the relative weights of two items under $\mathbf{p}$ (and to draw random samples from $\mathbf{p}$, by setting $S = [n]$).

INTCOND **oracle:** We define an INTCOND (short for "interval-cond") *oracle for* $\mathbf{p}$ as a restricted version of $\mathsf{COND}_{\mathbf{p}}$ that only accepts input sets S which are intervals $S = [a, b] = \{a, a+1, \ldots, b\}$ for some $a \leq b \in [n]$ (note that taking $a = 1$, $b = n$ this provides the power of a $\mathsf{SAMP}_{\mathbf{p}}$ oracle). This is a natural restriction on COND queries in settings where the n points are endowed with a total order.

To motivate the PAIRCOND model (which essentially gives the ability to compare two elements), one may consider a setting in which a human domain expert can provide an estimate of the relative likelihood of two distinct outcomes in a limited-information prediction scenario.

4.1.1.1 Our results

We give a detailed study of a range of natural distribution testing problems in the COND model and its variants described above, establishing both upper and lower bounds on their query complexity. Our results show that the ability to do conditional sampling provides a significant amount of power to property testers, enabling $\mathrm{polylog}(n)$-query, or even constant-query, algorithms for problems whose sample complexities in the standard

Problem	Our results		Standard model
Is $\mathbf{p}$ uniform?	$\mathsf{COND}_{\mathbf{p}}$	$\Omega\left(\frac{1}{\varepsilon^2}\right)$	$\Theta\left(\frac{\sqrt{n}}{\varepsilon^2}\right)$ [104, 20, 138]
	$\mathsf{PAIRCOND}_{\mathbf{p}}$	$\tilde{O}\left(\frac{1}{\varepsilon^2}\right)$	
	$\mathsf{INTCOND}_{\mathbf{p}}$	$\tilde{O}\left(\frac{\log^3 n}{\varepsilon^3}\right)$ $\Omega\left(\frac{\log n}{\log\log n}\right)$	
Is $\mathbf{p} = \mathbf{p}^*$ for a known $\mathbf{p}^*$?	$\mathsf{COND}_{\mathbf{p}}$	$\tilde{O}\left(\frac{1}{\varepsilon^4}\right)$	$\Theta\left(\frac{\sqrt{n}}{\varepsilon^2}\right)$ [21, 138, 169]
	$\mathsf{PAIRCOND}_{\mathbf{p}}$	$\tilde{O}\left(\frac{\log^4 n}{\varepsilon^4}\right)$ $\Omega\left(\sqrt{\frac{\log n}{\log\log n}}\right)$	
Are $\mathbf{p}_1, \mathbf{p}_2$ (both unknown) equivalent?	$\mathsf{COND}_{\mathbf{p}_1,\mathbf{p}_2}$	$\tilde{O}\left(\frac{\log^5 n}{\varepsilon^4}\right)$	$\Theta\left(\max\left(\frac{n^{2/3}}{\varepsilon^{4/3}}, \frac{\sqrt{n}}{\varepsilon^2}\right)\right)$ [20, 174, 58]
	$\mathsf{PAIRCOND}_{\mathbf{p}_1,\mathbf{p}_2}$	$\tilde{O}\left(\frac{\log^6 n}{\varepsilon^{21}}\right)$	
How far is $\mathbf{p}$ from uniform?	$\mathsf{PAIRCOND}_{\mathbf{p}}$	$\tilde{O}\left(\frac{1}{\varepsilon^{20}}\right)$	$O\left(\frac{1}{\varepsilon^2}\frac{n}{\log n}\right)$ [172, 170] $\Omega\left(\frac{n}{\log n}\right)$ [172, 167]

Table 4.1: Comparison between the COND model and the standard model on a variety of distribution testing problems over $[n]$. The upper bounds for the first three problems are for testing whether the property holds (i.e. $\mathrm{d_{TV}} = 0$) versus $\mathrm{d_{TV}} \geq \varepsilon$, and for the last problem the upper bound is for estimating the distance to uniformity to within an additive $\pm\varepsilon$.

model are $n^{\Omega(1)}$; see Table 4.1. While we have considered a variety of distribution testing problems in the COND model, our results are certainly not exhaustive, and many directions remain to be explored; we discuss some of these in Section 4.1.9.

Testing distributions over unstructured domains In this early work on the COND model our main focus has been on the simplest (and, we think, most fundamental) problems in distribution testing, such as testing whether $\mathbf{p}$ is the uniform distribution $\mathbf{u}$; testing whether $\mathbf{p} = \mathbf{p}^*$ for an explicitly provided $\mathbf{p}^*$; testing whether $\mathbf{p}_1 = \mathbf{p}_2$ given $\mathsf{COND}_{\mathbf{p}_1}$ and $\mathsf{COND}_{\mathbf{p}_2}$ oracles; and estimating the variation distance between $\mathbf{p}$ and the uniform distribution. In what follows $\mathrm{d_{TV}}$ denotes the variation distance.

Testing uniformity. We give a $\mathsf{PAIRCOND}_{\mathbf{p}}$ algorithm that tests whether $\mathbf{p} = \mathbf{u}$ versus $\mathrm{d_{TV}}(\mathbf{p}, \mathbf{u}) \geq \varepsilon$ using $\tilde{O}(1/\varepsilon^2)$ calls to $\mathsf{PAIRCOND}_{\mathbf{p}}$, independent of n. We show that this $\mathsf{PAIRCOND}_{\mathbf{p}}$ algorithm is nearly optimal by proving that any $\mathsf{COND}_{\mathbf{p}}$ tester (which may use arbitrary subsets $S \subseteq [n]$ as its query sets) requires $\Omega(1/\varepsilon^2)$ queries for this testing problem.

Testing equivalence to a known distribution. As stated above, for the simple problem of testing uniformity we have an essentially optimal PAIRCOND testing algorithm and a matching lower bound. A more general and challenging problem is that of testing whether $\mathbf{p}$ (accessible via a PAIRCOND or COND oracle) is equivalent to $\mathbf{p}^*$, where $\mathbf{p}^*$ is an arbitrary "known" distribution over $[n]$ that is explicitly provided to the testing algorithm at no cost (say as a vector $(\mathbf{p}^*(1), \ldots, \mathbf{p}^*(n))$ of probability values). For this "known $\mathbf{p}^*$" problem,

we give a $\mathsf{PAIRCOND}_{\mathbf{p}}$ algorithm testing whether $\mathbf{p} = \mathbf{p}^*$ versus $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{p}^*) \geq \varepsilon$ using $\tilde{O}((\log n)^4/\varepsilon^4)$ queries. We further show that the $(\log n)^{\Omega(1)}$ query complexity of our $\mathsf{PAIRCOND}_{\mathbf{p}}$ algorithm is inherent in the problem, by proving that any $\mathsf{PAIRCOND}_{\mathbf{p}}$ algorithm for this problem must use $\sqrt{\log(n)/\log\log(n)}$ queries for constant ε.

Given these $(\log n)^{\Theta(1)}$ upper and lower bounds on the query complexity of $\mathsf{PAIRCOND}_{\mathbf{p}}$-testing equivalence to a known distribution, it is natural to ask whether the full $\mathsf{COND}_{\mathbf{p}}$ oracle provides more power for this problem. We show that this is indeed the case, by giving a $\tilde{O}(1/\varepsilon^4)$-query algorithm (independent of n) that uses unrestricted $\mathsf{COND}_{\mathbf{p}}$ queries.

Testing equivalence between two unknown distributions. We next consider the more challenging problem of testing whether two unknown distributions $\mathbf{p}_1, \mathbf{p}_2$ over $[n]$ (available via $\mathsf{COND}_{\mathbf{p}_1}$ and $\mathsf{COND}_{\mathbf{p}_2}$ oracles) are identical versus ε-far. We give two very different algorithms for this problem. The first uses $\mathsf{PAIRCOND}$ oracles and has query complexity $\tilde{O}((\log n)^6/\varepsilon^{21})$, while the second uses COND oracles and has query complexity $\tilde{O}((\log n)^5/\varepsilon^4)$. We believe that the proof technique of the second algorithm is of independent interest, since it shows how a $\mathsf{COND}_{\mathbf{p}}$ oracle can efficiently simulate an "approximate $\mathsf{EVAL}_{\mathbf{p}}$ oracle." (An $\mathsf{EVAL}_{\mathbf{p}}$ oracle takes as input a point $i \in [n]$ and outputs the probability mass $\mathbf{p}(i)$ that $\mathbf{p}$ puts on i; we briefly explain our notion of approximating such an oracle in Section 4.1.1.1.)

Estimating the distance to uniformity. We also consider the problem of estimating the variation distance between $\mathbf{p}$ and the uniform distribution $\mathbf{u}$ over $[n]$, to within an additive error of $\pm\varepsilon$. In the standard $\mathsf{SAMP}_{\mathbf{p}}$ model this is known to be a very difficult problem, with an $\Omega(n/\log n)$ lower bound established in [172, 167]. In contrast, we give a $\mathsf{PAIRCOND}_{\mathbf{p}}$ algorithm that makes only $\tilde{O}(1/\varepsilon^{20})$ queries, independent of n.

Testing distributions over structured domains In the final portion of the section we view the domain $[n]$ as an ordered set $1 \leq \cdots \leq n$. (Note that in all the testing problems and results described previously, the domain could just as well have been viewed as an unstructured set of abstract points $x_1, \ldots, x_n$.) With this perspective it is natural to consider an additional oracle. We define an $\mathsf{INTCOND}$ (short for "interval-cond") *oracle for* $\mathbf{p}$ as a restricted version of $\mathsf{COND}_{\mathbf{p}}$, which only accepts input sets S that are intervals $S = [a, b] = \{a, a+1, \ldots, b\}$ for some $a \leq b \in [n]$ (note that taking $a = 1$, $b = n$ this provides the power of a $\mathsf{SAMP}_{\mathbf{p}}$ oracle).

We give an $\tilde{O}((\log n)^3/\varepsilon^3)$-query $\mathsf{INTCOND}_{\mathbf{p}}$ algorithm for testing whether $\mathbf{p}$ is uniform versus ε-far from uniform. We show that a $(\log n)^{\Omega(1)}$ query complexity is inherent for uniformity testing using $\mathsf{INTCOND}_{\mathbf{p}}$, by proving an $\Omega(\log n/\log\log n)$-query $\mathsf{INTCOND}_{\mathbf{p}}$ lower bound.

Along the way to establishing our main testing results described above, we develop several powerful tools for analyzing distributions in the COND and $\mathsf{PAIRCOND}$ models, which we believe may be of independent interest and utility in subsequent work on the COND and $\mathsf{PAIRCOND}$ models. These include as mentioned

above a procedure for approximately simulating an "evaluation oracle", as well as a procedure for estimating the weight of the "neighborhood" of a given point in the domain of the distribution. (See further discussion of these tools in Section 4.1.1.1.)

A high-level discussion of our algorithms To maintain focus here we describe only the ideas behind our algorithms; intuition for each of our lower bounds can be found in an informal discussion preceding the formal proof, see the beginnings of Sections 4.1.3.2, 4.1.4.2 and 4.1.8. As can be seen in the following discussion, our algorithms share some common themes, though each has its own unique idea/technique, which we emphasize below.

Our simplest testing algorithm is the algorithm for **testing whether $\mathbf{p}$ is uniform** over $[n]$ (using $\mathsf{PAIRCOND}_{\mathbf{p}}$ queries). The algorithm is based on the observation that if a distribution is ε-far from uniform, then the total weight (according to $\mathbf{p}$) of points $y \in [n]$ for which $\mathbf{p}(y) \geq (1+\Omega(\varepsilon))/n$ is $\Omega(\varepsilon)$, and the fraction of points $x \in [n]$ for which $\mathbf{p}(x) \leq (1-\Omega(\varepsilon))/n$ is $\Omega(\varepsilon)$. If we obtain such a pair of points (x, y), then we can detect this deviation from uniformity by performing $\Theta(1/\varepsilon^2)$ $\mathsf{PAIRCOND}_{\mathbf{p}}$ queries on the pair. Such a pair can be obtained with high probability by making $\Theta(1/\varepsilon)$ $\mathsf{SAMP}_{\mathbf{p}}$ queries (so as to obtain y) as well as selecting $\Theta(1/\varepsilon)$ points uniformly (so as to obtain x). This approach yields an algorithm whose complexity grows like $1/\varepsilon^4$. To actually get an algorithm with query complexity $\tilde{O}(1/\varepsilon^2)$ (which, as our lower bound shows, is tight), a slightly more refined approach is applied.

When we take the next step to **testing equality to an arbitrary (but fully specified) distribution $\mathbf{p}^*$**, the abovementioned observation generalizes so as to imply that if we sample $\Theta(1/\varepsilon)$ points from $\mathbf{p}$ and $\Theta(1/\varepsilon)$ from $\mathbf{p}^*$, then with high probability we shall obtain a pair of points (x, y) such that $\mathbf{p}(x)/\mathbf{p}(y)$ differs by at least $(1 \pm \Omega(\varepsilon))$ from $\mathbf{p}^*(x)/\mathbf{p}^*(y)$. Unfortunately, this cannot necessarily be detected by a small number of $\mathsf{PAIRCOND}_{\mathbf{p}}$ queries since (as opposed to the uniform case), $\mathbf{p}^*(x)/\mathbf{p}^*(y)$ may be very large or very small. However, we show that by sampling from both $\mathbf{p}$ and $\mathbf{p}^*$ and allowing the number of samples to grow with $\log n$, with high probability we either obtain a pair of points as described above for which $\mathbf{p}^*(x)/\mathbf{p}^*(y)$ is a constant, or we detect that for some set of points B we have that $|\mathbf{p}(B) - \mathbf{p}^*(B)|$ is relatively large.[2]

As noted previously, we prove a lower bound showing that a polynomial dependence on $\log n$ is unavoidable if only $\mathsf{PAIRCOND}_{\mathbf{p}}$ queries (in addition to standard sampling) are allowed. To obtain our more efficient $\mathrm{poly}(1/\varepsilon)$-queries algorithm, which uses more general $\mathsf{COND}_{\mathbf{p}}$ queries, we extend the observation from the uniform case in a different way. Specifically, rather than comparing the relative weight of pairs of points, we compare the relative weight of pairs in which one element is a point and the other is a subset of points. Roughly speaking, we show how points can be paired with subsets of points of comparable weight (according to $\mathbf{p}^*$) such that the following holds. If $\mathbf{p}$ is far from $\mathbf{p}^*$, then by taking $\tilde{O}(1/\varepsilon)$ samples from $\mathbf{p}$ and selecting subsets of points in an appropriate manner (depending on $\mathbf{p}^*$), we can obtain (with high probability) a point x and a subset Y such that $\mathbf{p}(x)/\mathbf{p}(Y)$ differs significantly from $\mathbf{p}^*(x)/\mathbf{p}^*(Y)$ *and* $\mathbf{p}^*(x)/\mathbf{p}^*(Y)$ is a constant.

[2]Here we use B for "Bucket", as we consider a bucketing of the points in $[n]$ based on their weight according to $\mathbf{p}^*$. We note that bucketing has been used extensively in the context of testing properties of distributions, see e.g. [20, 21].

In our next step, to **testing equality between two unknown distributions $\mathbf{p}_1$ and $\mathbf{p}_2$**, we need to cope with the fact that we no longer "have a hold" on a known distribution. Our PAIRCOND algorithm can be viewed as creating such a hold in the following sense. By sampling from $\mathbf{p}_1$ we obtain (with high probability) a (relatively small) set of points R that *cover* the distribution $\mathbf{p}_1$. By "covering" we mean that except for a subset having small weight according to $\mathbf{p}_1$, all points y in $[n]$ have a *representative* $r \in R$, i.e. a point r such that $\mathbf{p}_1(y)$ is close to $\mathbf{p}_1(r)$. We then show that if $\mathbf{p}_2$ is far from $\mathbf{p}_1$, then one of the following must hold: (1) There is relatively large weight, either according to $\mathbf{p}_1$ or according to $\mathbf{p}_2$, on points y such that for some $r \in R$ we have that $\mathbf{p}_1(y)$ is close to $\mathbf{p}_1(r)$ but $\mathbf{p}_2(y)$ is not sufficiently close to $\mathbf{p}_2(r)$; (2) There exists a point $r \in R$ such that the set of points y for which $\mathbf{p}_1(y)$ is close to $\mathbf{p}_1(r)$ has significantly different weight according to $\mathbf{p}_2$ as compared to $\mathbf{p}_1$. We note that this algorithm can be viewed as a variant of the PAIRCOND algorithm for the case when one of the distributions is known (where the "buckets" B, which were defined by $\mathbf{p}^*$ in that algorithm (and were disjoint), are now defined by the points in R (and are not necessarily disjoint)).

As noted previously, our (general) COND algorithm for testing the equality of two (unknown) distributions is based on a subroutine that estimates $\mathbf{p}(x)$ (to within $(1 \pm O(\varepsilon))$) for a given point x given access to $\mathsf{COND}_{\mathbf{p}}$. Obtaining such an estimate for *every* $x \in [n]$ cannot be done efficiently for some distributions.[3] However, we show that if we allow the algorithm to output unknown on some subset of points with total weight $O(\varepsilon)$, then the relaxed task can be performed using $\mathrm{poly}(\log n, 1/\varepsilon)$ queries, by performing a kind of randomized binary search "with exceptions". This relaxed version, which we refer to as an *approximate* EVAL *oracle*, suffices for our needs in distinguishing between the case that $\mathbf{p}_1$ and $\mathbf{p}_2$ are the same distribution and the case in which they are far from each other. It is possible that this procedure will be useful for other tasks as well.

The algorithm for **estimating the distance to uniformity** (which uses $\mathrm{poly}(1/\varepsilon)$ $\mathsf{PAIRCOND}_{\mathbf{p}}$ queries) is based on a subroutine for finding a *reference point* x together with an estimate $\widehat{\mathbf{p}}(x)$ of $\mathbf{p}(x)$. A reference point should be such that $\mathbf{p}(x)$ is relatively close to $1/n$ (if such a point cannot be found then it is evidence that $\mathbf{p}$ is very far from uniform). Given a reference point x (together with $\widehat{\mathbf{p}}(x)$) it is possible to estimate the distance to uniformity by obtaining (using PAIRCOND queries) estimates of the ratio between $\mathbf{p}(x)$ and $\mathbf{p}(y)$ for $\mathrm{poly}(1/\varepsilon)$ uniformly selected points y. The procedure for finding a reference point x together with $\widehat{\mathbf{p}}(x)$ is based on estimating both the weight and the size of a subset of points y such that $\mathbf{p}(y)$ is close to $\mathbf{p}(x)$. The procedure shares a common subroutine, ESTIMATE-NEIGHBORHOOD, with the PAIRCOND algorithm for testing equivalence between two unknown distributions.

Finally, the $\mathsf{INTCOND}_{\mathbf{p}}$ algorithm for testing uniformity is based on a version of the approximate EVAL oracle mentioned previously, which on one hand uses only $\mathsf{INTCOND}_{\mathbf{p}}$ (rather than general $\mathsf{COND}_{\mathbf{p}}$) queries, and on the other hand exploits the fact that we are dealing with the uniform distribution rather than an arbitrary distribution.

[3] As an extreme case consider a distribution $\mathbf{p}$ for which $\mathbf{p}(1) = 1 - \phi$ and $\mathbf{p}(2) = \cdots = \mathbf{p}(n) = \phi/(n-1)$ for some very small ϕ (which in particular may depend on n), and for which we are interested in estimating $\mathbf{p}(2)$. This requires $\Omega(1/\phi)$ queries.

4.1.1.2 The work of Chakraborty et al. [54]

Chakraborty et al. [54] proposed essentially the same COND model that we study, differing only in what happens on query sets S such that $\mathbf{p}(S) = 0$. In our model such a query causes the COND oracle and algorithm to return fail, while in their model such a query returns a uniform random $i \in S$.

Related to testing equality of distributions, [54] provides an (adaptive) algorithm for testing whether $\mathbf{p}$ is equivalent to a specified distribution $\mathbf{p}^*$ using $\mathrm{poly}(\log^* n, 1/\varepsilon)$ COND queries. Recall that we give an algorithm for this problem that performs $\tilde{O}(1/\varepsilon^4)$ COND queries. [54] also gives a *non-adaptive* algorithm for this problem that performs $\mathrm{poly}(\log n, 1/\varepsilon)$ COND queries.[4] Testing equivalence between two unknown distributions is not considered in [54], and the same is true for testing in the PAIRCOND model.

[54] also presents additional results for a range of other problems, which we discuss below:

- An (adaptive) algorithm for testing uniformity that performs $\mathrm{poly}(1/\varepsilon)$ queries.[5] The sets on which the algorithms performs COND queries are of size linear in $1/\varepsilon$. Recall that our algorithm for this problem performs $\tilde{O}(1/\varepsilon^2)$ PAIRCOND queries and that we show that every algorithm must perform $\Omega(1/\varepsilon^2)$ queries (when there is no restriction on the types of queries). We note that their analysis uses the same observation that ours does regarding distributions that are far from uniform (see the discussion in Section 4.1.1.1), but exploits it in a different manner.

  They also give a non-adaptive algorithm for this problem that performs $\mathrm{poly}(\log n, 1/\varepsilon)$ COND queries and show that $\Omega(\log \log n)$ is a lower bound on the necessary number of queries for non-adaptive algorithms.

- An (adaptive) algorithm for testing whether $\mathbf{p}$ is equivalent to a specified distribution $\mathbf{p}^*$ using $\mathrm{poly}(\log^* n, 1/\varepsilon)$ COND queries. Recall that we give an algorithm for this problem that performs $\tilde{O}(1/\varepsilon^4)$ COND queries.

  They also give a non-adaptive algorithm for this problem that performs $\mathrm{poly}(\log n, 1/\varepsilon)$ COND queries.

- An (adaptive) algorithm for testing any label-invariant (i.e., invariant under permutations of the domain) property that performs $\mathrm{poly}(\log n, 1/\varepsilon)$ COND queries. As noted in [54], this in particular implies an algorithm with this complexity for estimating the distance to uniformity. Recall that we give an algorithm for this estimation problem that performs $\mathrm{poly}(1/\varepsilon)$ PAIRCOND queries.

  The algorithm for testing any label-invariant property is based on learning a certain approximation of the distribution $\mathbf{p}$ and in this process defining some sort of approximate EVAL oracle. To the best of our understanding, our notion of an approximate EVAL oracle (which is used to obtain one of our results for testing equivalence between two unknown distributions) is quite different.

  They also show that there exists a label-invariant property for which any adaptive algorithm must

[4] We note that it is only possible for them to give a non-adaptive algorithm because their model is more permissive than ours if a query set S is proposed for which $\mathbf{p}(S) = 0$, their model returns a uniform random element of S while our model returns fail). In our stricter model, any non-adaptive algorithm which queries a proper subset $S \subsetneq [n]$ would output fail on some distribution $\mathbf{p}$.

[5] The precise polynomial is not specified – we believe it is roughly $1/\varepsilon^4$ as it follows from an application of the identity tester of [21] with distance $\Theta(\varepsilon^2)$ on a domain of size $O(1/\varepsilon)$.

perform $\Omega(\sqrt{\log\log n})$ COND queries.

- Finally they show that there exist general properties that require $\Omega(n)$ COND queries.

4.1.2 Some useful procedures

In this section we describe some procedures that will be used by our algorithms. On a first pass the reader may wish to focus on the explanatory prose and performance guarantees of these procedures (i.e., the statements of Lemma 4.1.2 and Lemma 4.1.3, as well as Definition 4.1.4 and Theorem 4.1.5) and otherwise skip to p.162; the internal details of the proofs are not necessary for the subsequent sections that use these procedures.

4.1.2.1 The procedure COMPARE

We start by describing a procedure that estimates the ratio between the weights of two disjoint sets of points by performing COND queries on the union of the sets. More precisely, it estimates the ratio (to within $1 \pm \eta$) if the ratio is not too high and not too low. Otherwise, it may output high or low, accordingly. In the special case when each set is of size one, the queries performed are PAIRCOND queries.

Algorithm 16 COMPARE

Require: COND query access to a distribution $\mathbf{p}$ over $[n]$, disjoint subsets $X, Y \subset [n]$, parameter $\eta \in (0,1]$, $K \geq 1$, and $\delta \in (0, 1/2]$.

1: Perform $\Theta\left(\frac{K\log(1/\delta)}{\eta^2}\right)$ $\mathsf{COND}_{\mathbf{p}}$ queries on the set $S = X \cup Y$, and let $\hat{\mu}$ be the fraction of times that a point $y \in Y$ is returned.
2: **if** $\hat{\mu} < \frac{2}{3} \cdot \frac{1}{K+1}$ **then return** low.
3: **else if** $1 - \hat{\mu} < \frac{2}{3} \cdot \frac{1}{K+1}$ **then return** high.
4: **else**
5: **return** $\rho = \frac{\hat{\mu}}{1-\hat{\mu}}$
6: **end if**

Lemma 4.1.2. *Given as input two disjoint subsets of points X, Y together with parameters $\eta \in (0,1]$, $K \geq 1$, and $\delta \in (0, 1/2]$, as well as* COND *query access to a distribution $\mathbf{p}$, the procedure* COMPARE *(Algorithm 16) either outputs a value $\rho > 0$ or outputs* high *or* low*, and satisfies the following:*

1. *If $\mathbf{p}(X)/K \leq \mathbf{p}(Y) \leq K \cdot \mathbf{p}(X)$ then with probability at least $1 - \delta$ the procedure outputs a value $\rho \in [1-\eta, 1+\eta]\mathbf{p}(Y)/\mathbf{p}(X)$;*
2. *If $\mathbf{p}(Y) > K \cdot \mathbf{p}(X)$ then with probability at least $1 - \delta$ the procedure outputs either* high *or a value $\rho \in [1-\eta, 1+\eta]\mathbf{p}(Y)/\mathbf{p}(X)$;*
3. *If $\mathbf{p}(Y) < \mathbf{p}(X)/K$ then with probability at least $1 - \delta$ the procedure outputs either* low *or a value $\rho \in [1-\eta, 1+\eta]\mathbf{p}(Y)/\mathbf{p}(X)$.*

The procedure performs $O\left(\frac{K\log(1/\delta)}{\eta^2}\right)$ COND queries on the set $X \cup Y$.

Proof. The bound on the number of queries performed by the algorithm follows directly from the description of the algorithm, and hence we turn to establish its correctness.

Let $w(X) = \frac{\mathbf{p}(X)}{\mathbf{p}(X)+\mathbf{p}(Y)}$ and let $w(Y) = \frac{\mathbf{p}(Y)}{\mathbf{p}(X)+\mathbf{p}(Y)}$. Observe that $\frac{w(Y)}{w(X)} = \frac{\mathbf{p}(Y)}{\mathbf{p}(X)}$ and that for $\hat{\mu}$ as defined in Line 1 of the algorithm, $\mathbb{E}[\hat{\mu}] = w(Y)$ and $\mathbb{E}[1-\hat{\mu}] = w(X)$. Also observe that for any $B \geq 1$, if $\mathbf{p}(Y) \geq \mathbf{p}(X)/B$, then $w(Y) \geq \frac{1}{B+1}$ and if $\mathbf{p}(Y) \leq B \cdot \mathbf{p}(X)$, then $w(X) \geq \frac{1}{B+1}$.

Let E_1 be the event that $\hat{\mu} \in [1-\eta/3, 1+\eta/3]w(Y)$ and let E_2 be the event that $(1-\hat{\mu}) \in [1-\eta/3, 1+\eta/3]w(X)$. Given the number of COND queries performed on the set $X \cup Y$, by applying a multiplicative Chernoff bound (see Theorem 1.4.10), if $w(Y) \geq \frac{1}{4K}$ then with probability at least $1-\delta/2$ the event E_1 holds, and if $w(X) \geq \frac{1}{4K}$, then with probability at least $1-\delta/2$ the event E_2 holds. We next consider the three cases in the lemma statement.

1. If $\mathbf{p}(X)/K \leq \mathbf{p}(Y) \leq K\mathbf{p}(X)$, then by the discussion above, $w(Y) \geq \frac{1}{K+1}$, $w(X) \geq \frac{1}{K+1}$, and with probability at least $1-\delta$ we have that $\hat{\mu} \in [1-\eta/3, 1+\eta/3]w(Y)$ and $(1-\hat{\mu}) \in [1-\eta/3, 1+\eta/3]w(X)$. Conditioned on these bounds holding,

$$\hat{\mu} \geq \frac{1-\eta/3}{K+1} \geq \frac{2}{3} \cdot \frac{1}{K+1} \quad \text{and} \quad 1-\hat{\mu} \geq \frac{2}{3} \cdot \frac{1}{K+1}\,.$$

   It follows that the procedure outputs a value $\rho = \frac{\hat{\mu}}{1-\hat{\mu}} \in [1-\eta, 1+\eta]\frac{w(Y)}{w(X)}$ as required by Item 1.

2. If $\mathbf{p}(Y) > K \cdot \mathbf{p}(X)$, then we consider two subcases.

   a) If $\mathbf{p}(Y) > 3K \cdot \mathbf{p}(X)$, then $w(X) < \frac{1}{3K+1}$, so that by a multiplicative Chernoff bound (stated in Claim 1.4.11), with probability at least $1-\delta$ we have that

$$1-\hat{\mu} < \frac{1+\eta/3}{3K+1} \leq \frac{4}{3} \cdot \frac{1}{3K+1} \leq \frac{2}{3} \cdot \frac{1}{K+1},$$

      causing the algorithm to output high. Thus Item 2 is established for this subcase.

   b) If $K \cdot \mathbf{p}(X) < \mathbf{p}(Y) \leq 3K \cdot \mathbf{p}(X)$, then $w(X) \geq \frac{1}{3K+1}$ and $w(Y) \geq \frac{1}{2}$, so that the events E_1 and E_2 both hold with probability at least $1-\delta$. Assume that these events in fact hold. This implies that $\hat{\mu} \geq \frac{1-\eta/3}{2} \geq \frac{2}{3} \cdot \frac{1}{K+1}$, and the algorithm either outputs high or outputs $\rho = \frac{\hat{\mu}}{1-\hat{\mu}} \in [1-\eta, 1+\eta]\frac{w(Y)}{w(X)}$, so Item 2 is established for this subcase as well.

3. If $\mathbf{p}(Y) < \mathbf{p}(X)/K$, so that $\mathbf{p}(X) > K \cdot \mathbf{p}(Y)$, then the exact same arguments are applied as in the previous case, just switching the roles of Y and X and the roles of $\hat{\mu}$ and $1-\hat{\mu}$ so as to establish Item 3.

We have thus established all items in the lemma. □

4.1.2.2 The procedure ESTIMATE-NEIGHBORHOOD

In this subsection we describe a procedure that, given a point x, provides an estimate of the weight of a set of points y such that $\mathbf{p}(y)$ is similar to $\mathbf{p}(x)$. In order to specify the behavior of the procedure more precisely, we introduce the following notation. For a distribution $\mathbf{p}$ over $[n]$, a point $x \in [n]$ and a parameter $\gamma \in [0,1]$, let

$$U_\gamma^{\mathbf{p}}(x) \stackrel{\text{def}}{=} \left\{ y \in [n] : \frac{1}{1+\gamma}\mathbf{p}(x) \leq \mathbf{p}(y) \leq (1+\gamma)\mathbf{p}(x) \right\} \tag{4.1}$$

denote the set of points whose weight is "γ-close" to the weight of x. If we take a sample of points distributed according to $\mathbf{p}$, then the expected fraction of these points that belong to $U_\gamma^{\mathbf{p}}(x)$ is $\mathbf{p}(U_\gamma^{\mathbf{p}}(x))$. If this value is not too small, then the actual fraction in the sample is close to the expected value. Hence, if we could efficiently determine for any given point y whether or not it belongs to $U_\gamma^{\mathbf{p}}(x)$, then we could obtain a good estimate of $\mathbf{p}(U_\gamma^{\mathbf{p}}(x))$. The difficulty is that it is not possible to perform this task efficiently for "boundary" points y such that $\mathbf{p}(y)$ is very close to $(1+\gamma)\mathbf{p}(x)$ or to $\frac{1}{1+\gamma}\mathbf{p}(x)$. However, for our purposes, it is not important that we obtain the weight and size of $U_\gamma^{\mathbf{p}}(x)$ for a specific γ, but rather it suffices to do so for γ in a given range, as stated in the next lemma. The parameter β in the lemma is the threshold above which we expect the algorithm to provide an estimate of the weight, while $[\kappa, 2\kappa)$ is the range in which γ is permitted to lie; finally, η is the desired (multiplicative) accuracy of the estimate, while δ is a bound on the probability of error allowed to the subroutine.

Lemma 4.1.3. *Given as input a point x together with parameters $\kappa, \beta, \eta, \delta \in (0, 1/2]$ as well as* PAIRCOND *query access to a distribution $\mathbf{p}$, the procedure* ESTIMATE-NEIGHBORHOOD *(Algorithm 17) outputs a pair $(\hat{w}, \alpha) \in [0,1] \times (\kappa, 2\kappa)$ such that α is uniformly distributed in $\{\kappa + i\theta\}_{i=0}^{\kappa/\theta - 1}$ for $\theta = \frac{\kappa\eta\beta\delta}{64}$, and such that the following holds:*

1. *If $\mathbf{p}(U_\alpha^{\mathbf{p}}(x)) \geq \beta$, then with probability at least $1-\delta$ we have $\hat{w} \in [1-\eta, 1+\eta] \cdot \mathbf{p}(U_\alpha^{\mathbf{p}}(x))$, and $\mathbf{p}(U_{\alpha+\theta}^{\mathbf{p}}(x) \setminus U_\alpha^{\mathbf{p}}(x)) \leq \eta\beta/16$;*
2. *If $\mathbf{p}(U_\alpha^{\mathbf{p}}(x)) < \beta$, then with probability at least $1-\delta$ we have $\hat{w} \leq (1+\eta)\cdot\beta$, and $\mathbf{p}(U_{\alpha+\theta}^{\mathbf{p}}(x) \setminus U_\alpha^{\mathbf{p}}(x)) \leq \eta\beta/16$.*

The number of PAIRCOND *queries performed by the procedure is $O\left(\frac{\log(1/\delta)\cdot\log(\log(1/\delta)/(\delta\beta\eta^2))}{\kappa^2\eta^4\beta^3\delta^2}\right)$.*

Algorithm 17 ESTIMATE-NEIGHBORHOOD

Require: PAIRCOND query access to a distribution $\mathbf{p}$ over $[n]$, a point $x \in [n]$ and parameters $\kappa, \beta, \eta, \delta \in (0, 1/2]$

1: Set $\theta = \frac{\kappa\eta\beta\delta}{64}$ and $r = \frac{\kappa}{\theta} = \frac{64}{\eta\beta\delta}$.
2: Select a value $\alpha \in \{\kappa + i\theta\}_{i=0}^{r-1}$ uniformly at random.
3: Call the $\mathsf{SAMP}_{\mathbf{p}}$ oracle $\Theta(\log(1/\delta)/(\beta\eta^2))$ times and let S be the set of points obtained.
4: For each point y in S call $\textsc{Compare}_{\mathbf{p}}(\{x\}, \{y\}, \theta/4, 4, \delta/(4|S|))$ (if a point y appears more than once in S, then COMPARE is called only once on y).
5: Let $\hat{w}$ be the fraction of occurrences of points y in S for which COMPARE returned a value $\rho(y) \in [1/(1+\alpha+\theta/2), (1+\alpha+\theta/2)]$. (That is, S is viewed as a multiset.)
6: Return $(\hat{w}, \alpha)$.

Proof of Lemma 4.1.3. The number of PAIRCOND queries performed by ESTIMATE-NEIGHBORHOOD is the size of S times the number of PAIRCOND queries performed in each call to COMPARE. By the setting of the parameters in the calls to COMPARE, the total number of PAIRCOND queries is $O\left(\frac{(|S|)\cdot\log|S|/\delta)}{\theta^2}\right) = O\left(\frac{\log(1/\delta)\cdot\log(\log(1/\delta)/(\delta\beta\eta^2))}{\kappa^2\eta^4\beta^3\delta^2}\right)$. We now turn to establishing the correctness of the procedure.

Since $\mathbf{p}$ and x are fixed, in what follows we shall use the shorthand U_γ for $U_\gamma^{\mathbf{p}}(x)$. For $\alpha \in \{\kappa + i\theta\}_{i=0}^{r-1}$, let $\Delta_\alpha \stackrel{\text{def}}{=} U_{\alpha+\theta} \setminus U_\alpha$. We next define several "desirable" events. In all that follows we view S as a multiset.

1. Let E_1 be the event that $\mathbf{p}(\Delta_\alpha) \leq 4/(\delta r)$. Since there are r disjoint sets Δ_α for $\alpha \in \{\kappa + i\theta\}_{i=0}^{r-1}$, the probability that E_1 occurs (taken over the uniform choice of α) is at least $1 - \delta/4$. From this point on we fix α and assume E_1 holds.
2. The event E_2 is that $|S \cap \Delta_\alpha|/|S| \leq 8/(\delta r)$ (that is, at most twice the upper bound on the expected value). By applying the multiplicative Chernoff bound using the fact that $|S| = \Theta(\log(1/\delta)/(\beta\eta^2)) = \Omega(\log(1/\delta) \cdot (\delta r))$, we have that $\Pr_S[E_2] \geq 1 - \delta/4$.
3. The event E_3 is defined as follows: If $\mathbf{p}(U_\alpha) \geq \beta$, then $|S \cap U_\alpha|/|S| \in [1-\eta/2, 1+\eta/2] \cdot \mathbf{p}(U_\alpha)$, and if $\mathbf{p}(U_\alpha) < \beta$, then $|S \cap U_\alpha|/|S| < (1+\eta/2) \cdot \beta$. Once again applying the multiplicative Chernoff bound (for both cases) and using that fact that $|S| = \Theta(\log(1/\delta)/(\beta\eta^2))$, we have that $\Pr_S[E_3] \geq 1 - \delta/4$.
4. Let E_4 be the event that all calls to COMPARE return an output as specified in Lemma 4.1.2. Given the setting of the confidence parameter in the calls to COMPARE we have that $\Pr[E_4] \geq 1 - \delta/4$ as well.

Assume from this point on that events E_1 through E_4 all hold where this occurs with probability at least $1 - \delta$. By the definition of Δ_α and E_1 we have that $\mathbf{p}(U_{\alpha+\theta} \setminus U_\alpha) \leq 4/(\delta r) = \eta\beta/16$, as required (in both items of the lemma). Let T be the (multi-)subset of points y in S for which COMPARE returned a value $\rho(y) \in [1/(1+\alpha+\theta/2), (1+\alpha+\theta/2)]$ (so that $\hat{w}$, as defined in the algorithm, equals $|T|/|S|$). Note first that conditioned on E_4 we have that for every $y \in U_{2\kappa}$ it holds that the output of COMPARE when called on $\{x\}$ and $\{y\}$, denoted $\rho(y)$, satisfies $\rho(y) \in [1 - \theta/4, 1 + \theta/4](\mathbf{p}(y)/\mathbf{p}(x))$, while for $y \notin U_{2\kappa}$ either COMPARE outputs high or low or it outputs a value $\rho(y) \in [1 - \theta/4, 1 + \theta/4](\mathbf{p}(y)/\mathbf{p}(x))$. This implies that if $y \in U_\alpha$, then $\rho(y) \leq (1 + \alpha) \cdot (1 + \theta/4) \leq 1 + \alpha + \theta/2$ and $\rho(y) \geq (1 + \alpha)^{-1} \cdot (1 - \theta/4) \geq (1 + \alpha + \theta/2)^{-1}$, so that $S \cap U_\alpha \subseteq T$. On the other hand, if $y \notin U_{\alpha+\theta}$ then either $\rho(y) > (1 + \alpha + \theta) \cdot (1 - \theta/4) \geq 1 + \alpha + \theta/2$ or $\rho(y) < (1 + \alpha + \theta)^{-1} \cdot (1 + \theta/4) \leq (1 + \alpha + \theta/2)^{-1}$ so that $T \subseteq S \cap U_{\alpha+\theta}$. Combining the two we have:

$$S \cap U_\alpha \;\subseteq\; T \;\subseteq\; S \cap U_{\alpha+\theta}\,. \tag{4.2}$$

Recalling that $\hat{w} = \frac{|T|}{|S|}$, the left-hand side of Eq. (4.2) implies that

$$\hat{w} \geq \frac{|S \cap U_\alpha|}{|S|}\,, \tag{4.3}$$

and by E_1 and E_2, the right-hand-side of Eq. (4.2) implies that

$$\hat{w} \;\leq\; \frac{|S \cap U_\alpha|}{|S|} + \frac{8}{\delta r} \;\leq\; \frac{|S \cap U_\alpha|}{|S|} + \frac{\beta\eta}{8}\,. \tag{4.4}$$

We consider the two cases stated in the lemma:

1. If $\mathbf{p}(U_\alpha) \geq \beta$, then by Eq. (4.3), Eq. (4.4) and (the first part of) E_3, we have that $\hat{w} \in [1 - \eta, 1 + \eta] \cdot \mathbf{p}(U_\alpha)$.
2. If $\mathbf{p}(U_\alpha) < \beta$, then by Eq. (4.4) and (the second part of) E_3, we have that $\hat{w} \leq (1 + \eta)\beta$.

The lemma is thus established. □

4.1.2.3 The procedure APPROX-EVAL-SIMULATOR

Approximate EVAL oracles. We begin by defining the notion of an "approximate EVAL oracle" that we will use. Intuitively this is an oracle which gives a multiplicatively $(1 \pm \varepsilon)$-accurate estimate of the value of $\mathbf{p}(i)$ for all i in a fixed set of probability weight at least $1 - \varepsilon$ under $\mathbf{p}$. More precisely, we have the following definition:

Definition 4.1.4. Let $\mathbf{p}$ be a distribution over $[n]$. An (ε, δ)-*approximate* $\mathsf{EVAL}_{\mathbf{p}}$ *simulator* is a randomized procedure ORACLE with the following property: For each $0 < \varepsilon < 1$, there is a fixed set $S^{(\varepsilon,\mathbf{p})} \subsetneq [n]$ with $\mathbf{p}(S^{(\varepsilon,\mathbf{p})}) < \varepsilon$ for which the following holds. Given as input an element $i^* \in [n]$, the procedure ORACLE either outputs a value $\alpha \in [0, 1]$ or outputs unknown or fail. The following holds for all $i^* \in [n]$:

(i) If $i^* \notin S^{(\varepsilon,\mathbf{p})}$ then with probability at least $1 - \delta$ the output of ORACLE on input i^* is a value $\alpha \in [0, 1]$ such that $\alpha \in [1 - \varepsilon, 1 + \varepsilon]\mathbf{p}(i^*)$;

(i) If $i^* \in S^{(\varepsilon,\mathbf{p})}$ then with probability at least $1 - \delta$ the procedure either outputs unknown or outputs a value $\alpha \in [0, 1]$ such that $\alpha \in [1 - \varepsilon, 1 + \varepsilon]\mathbf{p}(i^*)$.

We note that according to the above definition, it may be the case that different calls to ORACLE on the same input element $i^* \in [n]$ may return different values. However, the "low-weight" set $S^{(\varepsilon,\mathbf{p})}$ is an *a priori* fixed set that does not depend in any way on the input point i^* given to the algorithm. The key property of an (ε, δ)-approximate $\mathsf{EVAL}_{\mathbf{p}}$ oracle is that it reliably gives a multiplicatively $(1 \pm \varepsilon)$-accurate estimate of the value of $\mathbf{p}(i)$ for all i in some fixed set of probability weight at least $1 - \varepsilon$ under $\mathbf{p}$.

Constructing an approximate $\mathsf{EVAL}_{\mathbf{p}}$ simulator using $\mathsf{COND}_{\mathbf{p}}$ In this subsection we show that a $\mathsf{COND}_{\mathbf{p}}$ oracle can be used to obtain an approximate EVAL simulator:

Theorem 4.1.5. *Let* $\mathbf{p}$ *be any distribution over* $[n]$ *and let* $0 < \varepsilon, \delta < 1$. *The algorithm* APPROX-EVAL-SIMULATOR *has the following properties: It uses*

$$\tilde{O}\left(\frac{(\log n)^5 \cdot (\log(1/\delta))^2}{\varepsilon^3}\right)$$

calls to $\mathsf{COND}_{\mathbf{p}}$ *and it is an* (ε, δ)-*approximate* $\mathsf{EVAL}_{\mathbf{p}}$ *simulator.*

A few notes: First, in the proof we give below of Theorem 4.1.5 we assume throughout that $0 < \varepsilon \leq 1/40$. This incurs no loss of generality because if the desired ε parameter is in $(1/40, 1)$ then the parameter can simply be set to $1/40$. We further note that in keeping with our requirement on a $\mathsf{COND}_{\mathbf{p}}$ algorithm, the algorithm APPROX-EVAL-SIMULATOR only ever calls the $\mathsf{COND}_{\mathbf{p}}$ oracle on sets S which are either $S = [n]$ or else contain at least one element i that has been returned as the output of an earlier call to $\mathsf{COND}_{\mathbf{p}}$. To see this, note that Line 6 is the only line when $\mathsf{COND}_{\mathbf{p}}$ queries are performed. In the first execution of the outer

"For" loop clearly all COND queries are on set $S_0 = [n]$. In subsequent stages the only way a set S_j is formed is if either (i) S_j is set to $\{i^*\}$ in Line 10, in which case clearly i^* was previously received as the response of a $\mathsf{COND}_{\mathbf{p}}(S_{j-1})$ query, or else (ii) a nonzero fraction of elements $i_1, \ldots, i_m$ received as responses to $\mathsf{COND}_{\mathbf{p}}(S_{j-1})$ queries belong to S_j (see Line 19).

A preliminary simplification. Fix a distribution $\mathbf{p}$ over $[n]$. Let Z denote $\mathrm{supp}(\mathbf{p})$, i.e. $Z = \{i \in [n] : \mathbf{p}(i) > 0\}$. We first claim that in proving Theorem 4.1.5 we may assume without loss of generality that no two distinct elements $i, j \in Z$ have $\mathbf{p}(i) = \mathbf{p}(j)$ – in other words, we shall prove the theorem under this assumption on $\mathbf{p}$, and we claim that this implies the general result. To see this, observe that if Z contains elements $i \neq j$ with $\mathbf{p}(i) = \mathbf{p}(j)$, then for any arbitrarily small $\xi > 0$ and any arbitrarily large M we can perturb the weights of elements in Z to obtain a distribution $\mathbf{p}'$ supported on Z such that (i) no two elements of Z have the same probability under $\mathbf{p}'$, and (ii) for every $S \subseteq [n]$, $S \cap Z \neq \emptyset$ we have $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}_S, \mathbf{p}'_S) \leq \xi/M$. Since the variation distance between $\mathbf{p}'_S$ and $\mathbf{p}_S$ is at most ξ/M for an arbitrarily small ξ, the variation distance between (a) the execution of any M-query COND algorithm run on $\mathbf{p}$ and (b) the execution of any M-query COND algorithm run on $\mathbf{p}'$ will be at most ξ. Since ξ can be made arbitrarily small this means that indeed without loss of generality we may work with $\mathbf{p}'$ in what follows. Thus, we henceforth assume that the distribution $\mathbf{p}$ has no two elements in $\mathrm{supp}(\mathbf{p})$ with the same weight. For such a distribution we can explicitly describe the set $S^{(\varepsilon,\mathbf{p})}$ from Definition 4.1.4 that our analysis will deal with. Let $\pi : \{1, \ldots, |Z|\} \to Z$ be the bijection such that $\mathbf{p}(\pi(1)) > \cdots > \mathbf{p}(\pi(|Z|))$ (note that the bijection π is uniquely defined by the assumption that $\mathbf{p}(i) \neq \mathbf{p}(j)$ for all distinct $i, j \in Z$). Given a value $0 < \tau < 1$ we define the set $L_{\tau,D}$ to be $([n] \setminus Z) \cup \{\pi(s), \ldots, \pi(|Z|)\}$ where s is the smallest index in $\{1, \ldots, |Z|\}$ such that $\sum_{j=s}^{|Z|} \mathbf{p}(\pi(j)) < \tau$ (if $\mathbf{p}(\pi(|Z|))$ itself is at least τ then we define $L_{\tau,\mathbf{p}} = [n] \setminus Z$). Thus intuitively $L_{\tau,\mathbf{p}}$ contains the τ fraction (w.r.t. $\mathbf{p}$) of $[n]$ consisting of the lightest elements. The desired set $S^{(\varepsilon,\mathbf{p})}$ is precisely $L_{\varepsilon,\mathbf{p}}$.

Intuition for the algorithm. The high-level idea of the EVAL_D simulation is the following: Let $i^* \in [n]$ be the input element given to the $\mathsf{EVAL}_{\mathbf{p}}$ simulator. The algorithm works in a sequence of stages. Before performing the j-th stage it maintains a set S_{j-1} that contains i^*, and it has a high-accuracy estimate $\hat{\mathbf{p}}(S_{j-1})$ of the value of $\mathbf{p}(S_{j-1})$. (The initial set S_0 is simply $[n]$ and the initial estimate $\hat{\mathbf{p}}(S_0)$ is of course 1.) In the j-th stage the algorithm attempts to construct a subset S_j of S_{j-1} in such a way that (i) $i^* \in S_j$, and (ii) it is possible to obtain a high-accuracy estimate of $\mathbf{p}(S_j)/\mathbf{p}(S_{j-1})$ (and thus a high-accuracy estimate of $\mathbf{p}(S_j)$). If the algorithm cannot construct such a set S_j then it outputs $\mathsf{unknown}$; otherwise, after at most (essentially) $O(\log n)$ stages, it reaches a situation where $S_j = \{i^*\}$ and so the high-accuracy estimate of $\mathbf{p}(S_j) = \mathbf{p}(i^*)$ is the desired value.

A natural first idea towards implementing this high-level plan is simply to split S_{j-1} randomly into two pieces and use one of them as S_j. However this simple approach may not work; for example, if S_{j-1} has one or more elements which are very heavy compared to i^*, then with a random split it may not be possible to

efficiently estimate $\mathbf{p}(S_j)/\mathbf{p}(S_{j-1})$ as required in (ii) above. Thus we follow a more careful approach which first identifies and removes "heavy" elements from S_{j-1} in each stage.

In more detail, during the j-th stage, the algorithm first performs $\mathsf{COND}_{\mathbf{p}}$ queries on the set S_{j-1} to identify a set $H_j \subseteq S_{j-1}$ of "heavy" elements; this set essentially consists of all elements which individually each contribute at least a κ fraction of the total mass $\mathbf{p}(S_{j-1})$. (Here κ is a "not-too-small" quantity but it is significantly less than ε.) Next, the algorithm performs additional $\mathsf{COND}_{\mathbf{p}}$ queries to estimate $\mathbf{p}(i^*)/\mathbf{p}(S_{j-1})$. If this fraction exceeds $\kappa/20$ then it is straightforward to estimate $\mathbf{p}(i^*)/\mathbf{p}(S_{j-1})$ to high accuracy, so using $\hat{\mathbf{p}}(S_{j-1})$ it is possible to obtain a high-quality estimate of $\mathbf{p}(i^*)$ and the algorithm can conclude. However, the typical case is that $\mathbf{p}(i^*)/\mathbf{p}(S_{j-1}) < \kappa/20$. In this case, the algorithm next estimates $\mathbf{p}(H_j)/\mathbf{p}(S_{j-1})$. If this is larger than $1 - \varepsilon/10$ then the algorithm outputs $\mathsf{unknown}$ (see below for more discussion of this). If $\mathbf{p}(H_j)/\mathbf{p}(S_{j-1})$ is less than $1 - \varepsilon/10$ then $\mathbf{p}(S_{j-1} \setminus H_j)/\mathbf{p}(S_{j-1}) \geq \varepsilon/10$ (and so $\mathbf{p}(S_{j-1} \setminus H_j)/\mathbf{p}(S_{j-1})$ can be efficiently estimated to high accuracy), but each element k of $S_{j-1} \setminus H_j$ has $\mathbf{p}(k)/\mathbf{p}(S_{j-1}) \leq \kappa \ll \varepsilon/10 \leq \mathbf{p}(S_{j-1} \setminus H_j)/\mathbf{p}(S_{j-1})$. Thus it must be the case that the weight under $\mathbf{p}$ of $S_{j-1} \setminus H_j$ is "spread out" over many "light" elements.

Given that this is the situation, the algorithm next chooses S'_j to be a random subset of $S_{j-1} \setminus (H_j \cup \{i^*\})$, and sets S_j to be $S'_j \cup \{i^*\}$. It can be shown that with high probability (over the random choice of S_j) it will be the case that $\mathbf{p}(S_j) \geq \frac{1}{3}\mathbf{p}(S_{j-1} \setminus H_j)$ (this relies crucially on the fact that the weight under $\mathbf{p}$ of $S_{j-1} \setminus H_j$ is "spread out" over many "light" elements). This makes it possible to efficiently estimate $\mathbf{p}(S_j)/\mathbf{p}(S_{j-1} \setminus H_j)$; together with the high-accuracy estimate of $\mathbf{p}(S_{j-1} \setminus H_j)/\mathbf{p}(S_{j-1})$ noted above, and the high-accuracy estimate $\hat{\mathbf{p}}(S_{j-1})$ of $\mathbf{p}(S_{j-1})$, this means it is possible to efficiently estimate $\mathbf{p}(S_j)$ to high accuracy as required for the next stage. (We note that after defining S_j but before proceeding to the next stage, the algorithm actually checks to be sure that S_j contains at least one point that was returned from the $\mathsf{COND}_{\mathbf{p}}(S_{j-1})$ calls made in the past stage. This check ensures that whenever the algorithm calls $\mathsf{COND}_{\mathbf{p}}(S)$ on a set S, it is guaranteed that $\mathbf{p}(S) > 0$ as required by our $\mathsf{COND}_{\mathbf{p}}$ model. Our analysis shows that doing this check does not affect correctness of the algorithm since with high probability the check always passes.)

Intuition for the analysis. We require some definitions to give the intuition for the analysis establishing correctness. Fix a nonempty subset $S \subseteq [n]$. Let π_S be the bijection mapping $\{1, \ldots, |S|\}$ to S in such a way that $\mathbf{p}_S(\pi_S(1)) > \cdots > \mathbf{p}_S(\pi_S(|S|))$, i.e. $\pi_S(1), \ldots, \pi_S(|S|)$ is a listing of the elements of S in order from heaviest under $\mathbf{p}_S$ to lightest under $\mathbf{p}_S$. Given $j \in S$, we define the *S-rank of j*, denoted $\mathrm{rank}_S(j)$, to be the value $\sum_{i:\mathbf{p}_S(\pi(i)) \leq \mathbf{p}_S(j)} \mathbf{p}_S(\pi(i))$, i.e. $\mathrm{rank}_S(j)$ is the sum of the weights (under $\mathbf{p}_S$) of all the elements in S that are no heavier than j under $\mathbf{p}_S$. Note that having $i^* \notin L_{\varepsilon,n}$ implies that $\mathrm{rank}_{[n]}(i^*) \geq \varepsilon$.

We first sketch the argument for correctness. (It is easy to show that the algorithm only outputs fail with very small probability so we ignore this possibility below.) Suppose first that $i^* \notin L_{\varepsilon,\mathbf{p}}$. A key lemma shows that if $i^* \notin L_{\varepsilon,\mathbf{p}}$ (and hence $\mathrm{rank}_{[n]}(i^*) \geq \varepsilon$), then with high probability every set S_{j-1} constructed by the algorithm is such that $\mathrm{rank}_{S_{j-1}}(i^*) \geq \varepsilon/2$. (In other words, if i^* is not initially among the ε-fraction lightest elements (under $\mathbf{p}$), then for any j it never "falls too far" from becoming part of the $\varepsilon/2$-fraction lightest

elements of S_{j-1} (under $\mathbf{p}_{S_{j-1}}$).) Given that (with high probability) i^* always satisfies $\mathrm{rank}_{S_{j-1}}(i^*) \geq \varepsilon/2$, it must be the case that (with high probability) the procedure does not output unknown (and hence it must (with high probability) output a numerical value). This is because there are only two places where the procedure can output unknown, in Lines 14 and 19; we consider both cases below.

1. In order for the procedure to output unknown in Line 14, it must be the case that the elements of H_j – each of which individually has weight at least $\kappa/2$ under $\mathbf{p}_{S_{j-1}}$ – collectively have weight at least $1-3\varepsilon/20$ under $\mathbf{p}_{S_{j-1}}$ by Line 13. But i^* has weight at most $3\kappa/40$ under $\mathbf{p}_{S_{j-1}}$ (because the procedure did not go to Line 2 in Line 10), and thus i^* would need to be in the bottom $3\varepsilon/20$ of the lightest elements, i.e. it would need to have $\mathrm{rank}_{S_{j-1}}(i^*) \leq 3\varepsilon/20$; but this contradicts $\mathrm{rank}_{S_{j-1}}(i^*) \geq \varepsilon/2$.
2. In order for the procedure to output unknown in Line 19, it must be the case that all elements $i_1, \ldots, i_m$ drawn in Line 6 are not chosen for inclusion in S_j. In order for the algorithm to reach Line 19, though, it must be the case that at least $(\varepsilon/10 - \kappa/20)m$ of these draws do not belong to $H_j \cup \{i^*\}$; since these draws do not belong to H_j each one occurs only a small number of times among the m draws, so there must be many distinct values, and hence the probability that none of these distinct values is chosen for inclusion in S'_j is very low.

Thus we have seen that if $i^* \notin L_{\varepsilon,\mathbf{p}}$, then with high probability the procedure outputs a numerical value; it remains to show that with high probability this value is a high-accuracy estimate of $\mathbf{p}(i^*)$. However, this follows easily from the fact that we inductively maintain a high-quality estimate of $\mathbf{p}(S_{j-1})$ and the fact that the algorithm ultimately constructs its estimate of $\hat{\mathbf{p}}(i^*)$ only when it additionally has a high-quality estimate of $\mathbf{p}(i^*)/\mathbf{p}(S_{j-1})$. This fact also handles the case in which $i^* \in L_{\varepsilon,\mathbf{p}}$ – in such a case it is allowable for the algorithm to output unknown, so since the algorithm with high probability outputs a high-accuracy estimate when it outputs a numerical value, this means the algorithm performs as required in Case (ii) of Definition 4.1.4.

We now sketch the argument for query complexity. We will show that the heavy elements can be identified in each stage using $\mathrm{poly}(\log n, 1/\varepsilon)$ queries. Since the algorithm constructs S_j by taking a random subset of S_{j-1} (together with i^*) at each stage, the number of stages is easily bounded by (essentially) $O(\log n)$. Since the final probability estimate for $\mathbf{p}(i^*)$ is a product of $O(\log n)$ conditional probabilities, it suffices to estimate each of these conditional probabilities to within a multiplicative factor of $(1 \pm O\left(\frac{\varepsilon}{\log n}\right))$. We show that each conditional probability estimate can be carried out to this required precision using only $\mathrm{poly}(\log n, 1/\varepsilon)$ calls to $\mathsf{COND}_{\mathbf{p}}$; given this, the overall $\mathrm{poly}(\log n, 1/\varepsilon)$ query bound follows straightforwardly.

Now we enter into the actual proof. We begin our analysis with a simple but useful lemma about the "heavy" elements identified in Line 7.

Lemma 4.1.6. *With probability at least $1 - \delta/9$, every set H_j that is ever constructed in Line 7 satisfies the following for all $\ell \in S_{j-1}$:*

(i) If $\mathbf{p}(\ell)/\mathbf{p}(S_{j-1}) > \kappa$, then $\ell \in H_j$;

Algorithm 18 APPROX-EVAL-SIMULATOR

Require: access to $\mathsf{COND}_{\mathbf{p}}$; parameters $0 < \varepsilon, \delta < 1$; input element $i^* \in [n]$

1: Set $S_0 = [n]$ and $\hat{\mathbf{p}}(S_0) = 1$. Set $M = \log n + \log(9/\delta) + 1$. Set $\kappa = \Theta(\varepsilon/(M^2 \log(M/\delta)))$.
2: **for** $j = 1$ to M **do**
3: **if** $|S_{j-1}| = 1$ **then**
4: return $\hat{\mathbf{p}}(S_{j-1})$ (and exit)
5: **end if**
6: Perform $m = \Theta(\max\{M^2 \log(M/\delta)/(\varepsilon^2\kappa), \log(M/(\delta\kappa))/\kappa^2\})$ $\mathsf{COND}_{\mathbf{p}}$ queries on S_{j-1} to obtain points $i_1, \ldots, i_m \in S_{j-1}$.
7: Let $H_j = \{k \in [n] : k$ appears at least $\frac{3}{4}\kappa m$ times in the list $i_1, \ldots, i_m\}$
8: Let $\hat{\mathbf{p}}_{S_{j-1}}(i^*)$ denote the fraction of times that i^* appears in $i_1, \ldots, i_m$
9: **if** $\hat{\mathbf{p}}_{S_{j-1}}(i^*) \geq \frac{\kappa}{20}$ **then**
10: Set $S_j = \{i^*\}$, set $\hat{\mathbf{p}}(S_j) = \hat{\mathbf{p}}_{S_{j-1}}(i^*) \cdot \hat{\mathbf{p}}(S_{j-1})$, increment j, and go to Line 2.
11: **end if**
12: Let $\hat{\mathbf{p}}_{S_{j-1}}(H_j)$ denote the fraction of elements among $i_1, \ldots, i_m$ that belong to H_j.
13: **if** $\hat{\mathbf{p}}_{S_{j-1}}(H_j) > 1 - \varepsilon/10$ **then**
14: return unknown (and exit)
15: **end if**
16: Set S'_j to be a uniform random subset of $S_{j-1} \setminus (H_j \cup \{i^*\})$ and set S_j to be $S'_j \cup \{i^*\}$.
17: Let $\hat{\mathbf{p}}_{S_{j-1}}(S_j)$ denote the fraction of elements among $i_1, \ldots, i_m$ that belong to S_j
18: **if** $\hat{\mathbf{p}}_{S_{j-1}}(S_j) = 0$ **then**
19: return unknown (and exit)
20: **end if**
21: Set $\hat{\mathbf{p}}(S_j) = \hat{\mathbf{p}}_{S_{j-1}}(S_j) \cdot \hat{\mathbf{p}}(S_{j-1})$
22: **end for**
23: **return** fail.

(ii) If $\mathbf{p}(\ell)/\mathbf{p}(S_{j-1}) < \kappa/2$ *then* $\ell \notin H_j$.

Proof. Fix an iteration j. By Line 7 in the algorithm, a point ℓ is included in H_j if it appears at least $\frac{3}{4}\kappa m$ times among $i_1, \ldots, i_m$ (which are the output of $\mathsf{COND}_{\mathbf{p}}$ queries on S_{j-1}). For the first item, fix an element ℓ such that $\mathbf{p}(\ell)/\mathbf{p}(S_{j-1}) > \kappa$. Recall that $m = \Omega(M^2 \log(M/\delta)/(\varepsilon^2\kappa)) = \Omega(\log(Mn/\delta)/\kappa)$ (since $M = \Omega(\log(n))$). By a multiplicative Chernoff bound, the probability (over the choice of $i_1, \ldots, i_m$ in S_{j-1}) that ℓ appears less than $\frac{3}{4}\kappa m$ times among $i_1, \ldots, i_m$ (that is, less than $3/4$ times the lower bound on the expected value) is at most $\delta/(9Mn)$ (for an appropriate constant in the setting of m). On the other hand, for each fixed ℓ such that $\mathbf{p}(\ell)/\mathbf{p}(S_{j-1}) < \kappa/2$, the probability that ℓ appears at least $\frac{3}{4}\kappa m$ times (that is, at least $3/2$ times the upper bound on the expected value) is at most $\delta/(9Mn)$ as well. The lemma follows by taking a union bound over all (at most n) points considered above and over all M settings of j. □

Next we show that with high probability Algorithm APPROX-EVAL-SIMULATOR returns either unknown or a numerical value (as opposed to outputting fail in Line 23):

Lemma 4.1.7. *For any* $\mathbf{p}$, ε, δ *and* i^*, *Algorithm* APPROX-EVAL-SIMULATOR *outputs* fail *with probability at most* $\delta/9$.

Proof. Fix any element $i \neq i^*$. The probability (taken only over the choice of the random subset in each execution of Line 16) that i is placed in S'_j in each of the first $\log n + \log(9/\delta)$ executions of Line 16 is at most

$\frac{\delta}{9n}$. Taking a union bound over all $n-1$ points $i \neq i^*$, the probability that any point other than i^* remains in S_{j-1} through all of the first $\log n + \log(9/\delta)$ executions of the outer "for" loop is at most $\frac{\delta}{9}$. Assuming that this holds, then in the execution of the outer "for" loop when $j = \log n + \log(9/\delta) + 1$, the algorithm will return $\hat{\mathbf{p}}(S_{j-1}) = \hat{\mathbf{p}}(i^*)$ in Line 4. □

For the rest of the analysis it will be helpful for us to define several "desirable" events and show that they all hold with high probability:

1. Let E_1 denote the event that every set H_j that is ever constructed in Line 7 satisfies both properties (i) and (ii) stated in Lemma 4.1.6. By Lemma 4.1.6 the event E_1 holds with probability at least $1 - \delta/9$.
2. Let E_2 denote the event that in every execution of Line 8, the estimate $\hat{\mathbf{p}}_{S_{j-1}}(i^*)$ is within an additive $\pm\frac{\kappa}{40}$ of the true value of $\mathbf{p}(i^*)/\mathbf{p}(S_{j-1})$. By the choice of m in Line 6 (i.e., using $m = \Omega(\log(M/\delta)/\kappa^2)$), an additive Chernoff bound, and a union bound over all iterations, the event E_2 holds with probability at least $1 - \delta/9$.
3. Let E_3 denote the event that if Line 10 is executed, the resulting value $\hat{\mathbf{p}}_{S_{j-1}}(i^*)$ lies in $[1 - \frac{\varepsilon}{2M}, 1 + \frac{\varepsilon}{2M}]\mathbf{p}(i^*)/\mathbf{p}(S_{j-1})$. Assuming that event E_2 holds, if Line 10 is reached then the true value of $\mathbf{p}(i^*)/\mathbf{p}(S_{j-1})$ must be at least $\kappa/40$, and consequently a multiplicative Chernoff bound and the choice of m (i.e. using $m = \Omega(M^2 \log(M/\delta)/(\varepsilon^2\kappa))$) together imply that $\hat{\mathbf{p}}_{S_{j-1}}(i^*)$ lies in $[1 - \frac{\varepsilon}{2M}, 1 + \frac{\varepsilon}{2M}]\mathbf{p}(i^*)/\mathbf{p}(S_{j-1})$ except with failure probability at most $\delta/9$.
4. Let E_4 denote the event that in every execution of Line 12, the estimate $\hat{\mathbf{p}}_{S_{j-1}}(H_j)$ is within an additive error of $\pm\frac{\varepsilon}{20}$ from the true value of $\mathbf{p}(H_j)/\mathbf{p}(S_{j-1})$. By the choice of m in Line 6 (i.e., using $m = \Omega(\log(M/\delta)/\varepsilon^2)$) and an additive Chernoff bound, the event E_4 holds with probability at least $1 - \delta/9$.

The above arguments show that E_1, E_2, E_3 and E_4 all hold with probability at least $1 - 4\delta/9$.

Let E_5 denote the event that in every execution of Line 16, the set S'_j which is drawn satisfies $\mathbf{p}(S'_j)/\mathbf{p}(S_{j-1} \setminus (H_j \cup \{i^*\})) \geq 1/3$

The following lemma says that conditioned on E_1 through E_4 all holding, event E_5 holds with high probability:

Lemma 4.1.8. *Conditioned on E_1 through E_4 the probability that E_5 holds is at least $1 - \delta/9$.*

Proof. Fix a value of j and consider the j-th iteration of Line 16. Since events E_2 and E_4 hold, it must be the case that $\mathbf{p}(S_{j-1} \setminus (H_j \cup \{i^*\}))/\mathbf{p}(S_{j-1}) \geq \varepsilon/40$. Since event E_1 holds, it must be the case that every $i \in (S_{j-1} \setminus (H_j \cup \{i^*\}))$ has $\mathbf{p}(i)/\mathbf{p}(S_{j-1}) \leq \kappa$. Now since S'_j is chosen by independently including each element of $S_{j-1} \setminus (H_j \cup \{i^*\})$ with probability $1/2$, we can apply the first part of Corollary 1.4.12 and get

$$\Pr\left[\mathbf{p}(S'_j) < \frac{1}{3}\mathbf{p}(S_{j-1} \setminus (H_j \cup \{i^*\}))\right] \leq e^{-4\varepsilon/(40 \cdot 9 \cdot 4\kappa)} < \frac{\delta}{9M},$$

where the last inequality follows by the setting of $\kappa = \Omega(\varepsilon/(M^2 \log(1/\delta)))$. □

Thus we have established that E_1 through E_5 all hold with probability at least $1 - 5\delta/9$.

Next, let E_6 denote the event that the algorithm never returns unknown and exits in Line 19. Our next lemma shows that conditioned on events E_1 through E_5, the probability of E_6 is at least $1 - \delta/9$:

Lemma 4.1.9. *Conditioned on E_1 through E_5 the probability that E_6 holds is at least $1 - \delta/9$.*

Proof. Fix any iteration j of the outer "For" loop. In order for the algorithm to reach Line 18 in this iteration, it must be the case (by Lines 9 and 13) that at least $(\varepsilon/10 - \kappa/20)m > (\varepsilon/20)m$ points in $i_1, \ldots, i_m$ do not belong to $H_j \cup \{i^*\}$. Since each point not in H_j appears at most $\frac{3}{4}\kappa m$ times in the list $i_1, \ldots, i_m$, there must be at least $\frac{\varepsilon}{15\kappa}$ distinct such values. Hence the probability that none of these values is selected to belong to S'_j is at most $1/2^{\varepsilon/(15\kappa)} < \delta/(9M)$. A union bound over all (at most M) values of j gives that the probability the algorithm ever returns unknown and exits in Line 19 is at most $\delta/9$, so the lemma is proved. □

Now let E_7 denote the event that in every execution of Line 17, the estimate $\hat{\mathbf{p}}_{S_{j-1}}(S_j)$ lies in $[1 - \frac{\varepsilon}{2M}, 1 + \frac{\varepsilon}{2M}]\mathbf{p}(S_j)/\mathbf{p}(S_{j-1})$. The following lemma says that conditioned on E_1 through E_5, event E_7 holds with probability at least $1 - \delta/9$:

Lemma 4.1.10. *Conditioned on E_1 through E_5, the probability that E_7 holds is at least $1 - \delta/9$.*

Proof. Fix a value of j and consider the j-th iteration of Line 17.The expected value of $\hat{\mathbf{p}}_{S_{j-1}}(S_j)$ is precisely

$$\frac{\mathbf{p}(S_j)}{\mathbf{p}(S_{j-1})} = \frac{\mathbf{p}(S_j)}{\mathbf{p}(S_{j-1} \setminus (H_j \cup \{i^*\}))} \cdot \frac{\mathbf{p}(S_{j-1} \setminus (H_j \cup \{i^*\}))}{\mathbf{p}(S_{j-1})}. \tag{4.5}$$

Since events E_2 and E_4 hold we have that $\frac{\mathbf{p}(S_{j-1} \setminus (H_j \cup \{i^*\}))}{\mathbf{p}(S_{j-1})} \geq \varepsilon/40$, and since event E_5 holds we have that $\frac{\mathbf{p}(S_j)}{\mathbf{p}(S_{j-1} \setminus (H_j \cup \{i^*\}))} \geq 1/3$ (note that $\mathbf{p}(S_j) \geq \mathbf{p}(S'_j)$). Thus we have that (4.5) is at least $\varepsilon/120$. Recalling the value of m (i.e., using $m = \Omega(M^2 \log(M/\delta)/\varepsilon^2\kappa) = \Omega(M^2 \log(M/\delta)/\varepsilon^3)$) a multiplicative Chernoff bound gives that indeed $\hat{\mathbf{p}}_{S_{j-1}}(S_j) \in [1 - \frac{\varepsilon}{2M}, 1 + \frac{\varepsilon}{2M}]\mathbf{p}(S_j)/\mathbf{p}(S_{j-1})$ with failure probability at most $\delta/(9M)$. A union bound over all M possible values of j finishes the proof. □

At this point we have established that events E_1 through E_7 all hold with probability at least $1 - 7\delta/9$.

We can now argue that each estimate $\hat{\mathbf{p}}(S_j)$ is indeed a high-accuracy estimate of the true value $\mathbf{p}(S_j)$:

Lemma 4.1.11. *With probability at least $1 - 7\delta/9$ each estimate $\hat{\mathbf{p}}(S_j)$ constructed by* APPROX-EVAL-SIMULATOR *lies in $[(1 - \frac{\varepsilon}{2M})^j, (1 + \frac{\varepsilon}{2M})^j]\mathbf{p}(S_j)$.*

Proof. We prove the lemma by showing that if all events E_1 through E_7 hold, then the conclusion of the lemma (denoted (*) for the sake of succinctness) holds: each estimate $\hat{\mathbf{p}}(S_j)$ constructed by APPROX-EVAL-SIMULATOR lies in $[(1 - \frac{\varepsilon}{2M})^j, (1 + \frac{\varepsilon}{2M})^j]\mathbf{p}(S_j)$. Thus for the rest of the proof we assume that indeed all events E_1 through E_7 hold.

The claim (*) is clearly true for $j = 0$. We prove (*) by induction on j assuming it holds for $j - 1$. The only places in the algorithm where $\hat{\mathbf{p}}(S_j)$ may be set are Lines 10 and 21. If $\hat{\mathbf{p}}(S_j)$ is set in Line 21 then (*) follows from the inductive claim for $j - 1$ and Lemma 4.1.10. If $\hat{\mathbf{p}}(S_j)$ is set in Line 10, then (*) follows from the inductive claim for $j - 1$ and the fact that event E_3 holds. This concludes the proof of the lemma. □

Finally, we require the following crucial lemma which establishes that if $i^* \notin L_{\varepsilon,n}$ (and hence the initial rank $\mathrm{rank}_{[n]}$ of i^* is at least ε), then with very high probability the rank of i^* never becomes too low during the execution of the algorithm:

Lemma 4.1.12. *Suppose $i^* \notin L_{\varepsilon,n}$. Then with probability at least $1 - \delta/9$, every set S_{j-1} constructed by the algorithm has $\mathrm{rank}_{S_{j-1}}(i^*) \geq \varepsilon/2$.*

We prove Lemma 4.1.12 in Section 4.1.2.3 below.

With these pieces in place we are ready to prove Theorem 4.1.5.

Proof of Theorem 4.1.5: It is straightforward to verify that algorithm APPROX-EVAL-SIMULATOR has the claimed query complexity. We now argue that APPROX-EVAL-SIMULATOR meets the two requirements (i) and (ii) of Definition 4.1.4. Throughout the discussion below we assume that all the "favorable events" in the above analysis (i.e. events E_1 through E_7, Lemma 4.1.7, and Lemma 4.1.12) indeed hold as desired (incurring an overall failure probability of at most δ).

Suppose first that $i^* \notin L_{\varepsilon,\mathbf{p}}$. We claim that by Lemma 4.1.12 it must be the case that the algorithm does not return unknown in Line 14. To verify this, observe that in order to reach Line 14 it would need to be the case that $\mathbf{p}(i^*)/\mathbf{p}(S_{j-1}) \leq 3\kappa/40$ (so the algorithm does not instead go to Line 2 in Line 10). Since by Lemma 4.1.6 every element k in H_j satisfies $\mathbf{p}(k)/\mathbf{p}(S_{j-1}) \geq \kappa/2$, this means that i^* does not belong to H_j. In order to reach Line 14, by event E_4 we must have $\mathbf{p}(H_j)/\mathbf{p}(S_{j-1}) \geq 1 - 3\varepsilon/20$. Since every element of H_j has more mass under $\mathbf{p}$ (at least $\kappa/2$) than i^* (which has at most $3\kappa/40$), this would imply that $\mathrm{rank}_{S_{j-1}}(i^*) \leq 3\varepsilon/20$, contradicting Lemma 4.1.12. Furthermore, by Lemma 4.1.9 it must be the case that the algorithm does not return unknown in Line 19. Thus the algorithm terminates by returning an estimate $\hat{\mathbf{p}}(S_j) = \hat{\mathbf{p}}(i^*)$ which, by Lemma 4.1.11, lies in $[(1 - \frac{\varepsilon}{2M})^j, (1 + \frac{\varepsilon}{2M})^j]\mathbf{p}(i^*)$. Since $j \leq M$ this estimate lies in $[1 - \varepsilon, 1 + \varepsilon]\mathbf{p}(i^*)$ as required.

Now suppose that $i^* \in L_{\varepsilon,\mathbf{p}}$. By Lemma 4.1.7 we may assume that the algorithm either outputs unknown or a numerical value. As above, Lemma 4.1.11 implies that if the algorithm outputs a numerical value then the value lies in $[1 - \varepsilon, 1 + \varepsilon]\mathbf{p}(i^*)$ as desired. This concludes the proof of Theorem 4.1.5. □

Proof of Lemma 4.1.12. The key to proving Lemma 4.1.12 will be proving the next lemma. (In the following, for S a set of real numbers we write $\mathrm{sum}(S)$ to denote $\sum_{\alpha \in S} \alpha$.)

Lemma 4.1.13. *Fix $0 < \varepsilon \leq 1/40$. Set $\kappa = \Theta(\varepsilon/(M^2 \log(M/\delta)))$. Let $T = \{\alpha_1, \ldots, \alpha_n\}$ be a set of values $\alpha_1 < \cdots < \alpha_n$ such that $\mathrm{sum}(T) = 1$. Fix $\ell \in [n]$ and let $T_L = \{\alpha_1, \ldots, \alpha_\ell\}$ and let $T_R = \{\alpha_{\ell+1}, \ldots, \alpha_n\}$, so $T_L \cup T_R = T$. Assume that $\mathrm{sum}(T_L) \geq \varepsilon/2$ and that $\alpha_\ell \leq \kappa/10$.*

Fix H to be any subset of T satisfying the following two properties: (i) H includes every α_j such that $\alpha_j \geq \kappa$; and (ii) H includes no α_j such that $\alpha_j < \kappa/2$. (Note that consequently H does not intersect T_L.)

Let T' be a subset of $(T \setminus (H \cup \{\alpha_\ell\})$ selected uniformly at random. Let $T'_L = T' \cap T_L$ and let $T'_R = T' \cap T_R$.

Then we have the following:

1. *If* $\mathrm{sum}(T_L) \geq 20\varepsilon$*, then with probability at least* $1 - \delta/M$ *(over the random choice of* T'*) it holds that*

$$\frac{\mathrm{sum}(T'_L \cup \{\alpha_\ell\})}{\mathrm{sum}(T' \cup \{\alpha_\ell\})} \geq 9\varepsilon;$$

2. *If* $\varepsilon/2 \leq \mathrm{sum}(T_L) < 20\varepsilon$*, then with probability at least* $1 - \delta/M$ *(over the random choice of* T'*) it holds that*

$$\frac{\mathrm{sum}(T'_L \cup \{\alpha_\ell\})}{\mathrm{sum}(T' \cup \{\alpha_\ell\})} \geq \mathrm{sum}(T_L)\,(1-\rho)\,,$$

where $\rho = \frac{\ln 2}{M}$.

Proof of Lemma 4.1.12 using Lemma 4.1.13: We apply Lemma 4.1.13 repeatedly at each iteration j of the outer "For" loop. The set H of Lemma 4.1.13 corresponds to the set H_j of "heavy" elements that are removed at a given iteration, the set of values T corresponds to the values $\mathbf{p}(i)/\mathbf{p}(S_{j-1})$ for $i \in S_{j-1}$, and the element α_ℓ of Lemma 4.1.13 corresponds to $\mathbf{p}(i^*)/\mathbf{p}(S_{j-1})$. The value $\mathrm{sum}(T_L)$ corresponds to $\mathrm{rank}_{S_{j-1}}(i^*)$ and the value

$$\frac{\mathrm{sum}(T'_L \cup \{\alpha_\ell\})}{\mathrm{sum}(T' \cup \{\alpha_\ell\})}$$

corresponds to $\mathrm{rank}_{S_j}(i^*)$. Observe that since $i^* \notin L_{\varepsilon,n}$ we know that initially $\mathrm{rank}_{[n]}(i^*) \geq \varepsilon$, which means that the first time we apply Lemma 4.1.13 (with $T = \{\mathbf{p}(i) : i \in [n]\}$) we have $\mathrm{sum}(T_L) \geq \varepsilon$.

By Lemma 4.1.13 the probability of failure in any of the (at most M) iterations is at most $\delta/9$, so we assume that there is never a failure. Consequently for all j we have that if $\mathrm{rank}_{S_{j-1}}(i^*) \geq 20\varepsilon$ then $\mathrm{rank}_{S_j}(i^*) \geq 9\varepsilon$, and if $\varepsilon/2 \leq \mathrm{rank}_{S_{j-1}}(i^*) < 20\varepsilon$ then $\mathrm{rank}_{S_j}(i^*) \geq \mathrm{rank}_{S_j}(i^*) \cdot (1-\rho)$. Since $\mathrm{rank}_{S_0}(i^*) \geq \varepsilon$, it follows that for all $j \leq M$ we have $\mathrm{rank}_{S_j}(i^*) \geq \varepsilon \cdot (1-\rho)^M > \varepsilon/2$. □

Proof of Lemma 4.1.13. We begin with the following claim:

Claim 4.1.14. *With probability at least* $1 - \delta/(2M)$ *(over the random choice of* T'*) it holds that* $\mathrm{sum}(T'_L) \geq \frac{1}{2} \cdot \mathrm{sum}(T_L) \cdot (1 - \rho/2)$.

Proof. Recall from the setup that every element $\alpha_i \in T_L$ satisfies $\alpha_i \leq \kappa/10$, and $\mathrm{sum}(T_L) \geq \varepsilon/2$. Also recall that $\kappa = O(\varepsilon/(M^2 \log(M/\delta)))$ and that $\rho = \frac{\ln 2}{M}$, so that $\rho^2\varepsilon/(6\kappa) \geq \ln(2M/\delta)$. The claim follows by applying the first part of Corollary 1.4.12 (with $\gamma = \rho/2$). □

Part (1) of Lemma 4.1.13 is an immediate consequence of Claim 4.1.14, since in part (1) we have

$$\frac{\mathrm{sum}(T'_L \cup \{\alpha_\ell\})}{\mathrm{sum}(T' \cup \{\alpha_\ell\})} \geq \mathrm{sum}(T'_L) \geq \frac{1}{2} \cdot \mathrm{sum}(T_L) \cdot \left(1 - \frac{\rho}{2}\right) \geq \frac{1}{2} \cdot 20\varepsilon \cdot \left(1 - \frac{\rho}{2}\right) \geq 9\varepsilon.$$

It remains to prove Part (2) of the lemma. We will do this using the following claim:

Claim 4.1.15. *Suppose $\varepsilon/2 \le \operatorname{sum}(T_L) \le 20\varepsilon$. Then with probability at least $1-\delta/(2M)$ (over the random choice of T') it holds that $\operatorname{sum}(T'_R) \le \frac{1}{2}\operatorname{sum}(T_R)\cdot(1+\rho/2)$.*

Proof. Observe first that $\alpha_i < \kappa$ for each $\alpha_i \in T_R \setminus H$. We consider two cases.

If $\operatorname{sum}(T_R \setminus H) \ge 4\varepsilon$, then we apply the first part of Corollary 1.4.12 to the α_i's in $T_R \setminus H$ and get that

$$\begin{aligned}\Pr\left[\operatorname{sum}(T'_R) > \frac{1}{2}\operatorname{sum}(T_R)\cdot(1+\rho/2)\right] &\le \Pr\left[\operatorname{sum}(T'_R) > \frac{1}{2}\operatorname{sum}(T_R\setminus H)\cdot(1+\rho/2)\right] \\ &< \exp(-\rho^2\operatorname{sum}(T_R\setminus H)/24\kappa) \qquad (4.6)\\ &\le \exp(-\rho^2\varepsilon/(6\kappa)) \le \frac{\delta}{2M} \qquad (4.7)\end{aligned}$$

(recall from the proof of Claim 4.1.14 that $\rho^2\varepsilon/(6\kappa) \ge \ln(2M/\delta)$).

If $\operatorname{sum}(T_R\setminus H) < 4\varepsilon$, (so that the expected value of $\operatorname{sum}(T'_R)$ is less than 2ε) then we can apply the second part of Corollary 1.4.12 as we explain next. Observe that by the premise of the lemma, $\operatorname{sum}(T_R) \ge 1-20\varepsilon$ which is at least $1/2$ (recalling that ε is at most 1/40). Consequently, the event "$\operatorname{sum}(T'_R) \ge \frac{1}{2}\cdot\operatorname{sum}(T_R)\cdot(1+\rho/2)$" implies the event "$\operatorname{sum}(T'_R) \ge \frac{1}{4}$", and by applying the second part of Corollary 1.4.12 we get

$$\Pr\left[\operatorname{sum}(T'_R) > \frac{1}{2}\operatorname{sum}(T_R)\cdot(1+\rho/2)\right] \le \Pr\left[\operatorname{sum}(T'_R) > \frac{1}{4}\right] < 2^{-1/4\kappa} < \frac{\delta}{2M}\,, \qquad (4.8)$$

as required. □

Now we can prove Lemma 4.1.13. Using Claims 4.1.14 and 4.1.15 we have that with probability at least $1-\delta/M$,

$$\operatorname{sum}(T'_L) \ge \frac{1}{2}\cdot\operatorname{sum}(T_L)\cdot(1-\rho/2) \qquad \text{and} \qquad \operatorname{sum}(T'_R) \le \frac{1}{2}\operatorname{sum}(T_R)\cdot(1+\rho/2);$$

we assume that both these inequalities hold going forth. Since

$$\frac{\operatorname{sum}(T'_L\cup\{\alpha_\ell\})}{\operatorname{sum}(T'\cup\{\alpha_\ell\})} = \frac{\operatorname{sum}(T'_L)+\alpha_\ell}{\operatorname{sum}(T')+\alpha_\ell} > \frac{\operatorname{sum}(T'_L)}{\operatorname{sum}(T')},$$

it is sufficient to show that $\frac{\operatorname{sum}(T'_L)}{\operatorname{sum}(T')} \ge \operatorname{sum}(T_L)(1-\rho)$; we now show this. As $\operatorname{sum}(T') = \operatorname{sum}(T'_L)+\operatorname{sum}(T'_R)$,

$$\begin{aligned}\frac{\operatorname{sum}(T'_L)}{\operatorname{sum}(T')} &= \frac{\operatorname{sum}(T'_L)}{\operatorname{sum}(T'_L)+\operatorname{sum}(T'_R)} = \frac{1}{1+\frac{\operatorname{sum}(T'_R)}{\operatorname{sum}(T'_L)}} \\ &\ge \frac{1}{1+\frac{(1/2)\cdot\operatorname{sum}(T_R)\cdot(1+\rho/2)}{(1/2)\cdot\operatorname{sum}(T_L)\cdot(1-\rho/2)}} \\ &= \frac{\operatorname{sum}(T_L)\cdot(1-\rho/2)}{\operatorname{sum}(T_L)\cdot(1-\rho/2)+\operatorname{sum}(T_R)\cdot(1+\rho/2)} \\ &\ge \frac{\operatorname{sum}(T_L)\cdot(1-\rho/2)}{\operatorname{sum}(T_L)\cdot(1+\rho/2)+\operatorname{sum}(T_R)\cdot(1+\rho/2)} \\ &= \operatorname{sum}(T_L)\cdot\frac{1-\rho/2}{1+\rho/2} > \operatorname{sum}(T_L)\cdot(1-\rho).\end{aligned}$$

This concludes the proof of Lemma 4.1.13. □

4.1.3 Algorithms and lower bounds for testing uniformity

4.1.3.1 A $\tilde{O}(1/\varepsilon^2)$-query PAIRCOND algorithm for testing uniformity

In this subsection we present an algorithm $\mathsf{PAIRCOND}_{\mathbf{p}}$-TEST-UNIFORM and prove the following theorem:

Theorem 4.1.16. $\mathsf{PAIRCOND}_{\mathbf{p}}$-TEST-UNIFORM *is a* $\tilde{O}(1/\varepsilon^2)$*-query* $\mathsf{PAIRCOND}_{\mathbf{p}}$ *testing algorithm for uniformity, i.e. it outputs* accept *with probability at least* $2/3$ *if* $\mathbf{p} = \mathbf{u}$ *and outputs* reject *with probability at least* $2/3$ *if* $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{u}) \geq \varepsilon$.

Intuition. For the sake of intuition we first describe a simpler approach that yields a $\tilde{O}(1/\varepsilon^4)$-query algorithm, and then build on those ideas to obtain our real algorithm with its improved $\tilde{O}(1/\varepsilon^2)$ bound. Fix $\mathbf{p}$ to be a distribution over $[n]$ that is ε-far from uniform. Let

$$H = \left\{ h \in [n] \,:\, \mathbf{p}(h) \geq \frac{1}{n} \right\} \text{ and } L = \left\{ \ell \in [n] \,:\, \mathbf{p}(\ell) < \frac{1}{n} \right\}.$$

It is easy to see that since $\mathbf{p}$ is ε-far from uniform, we have

$$\sum_{h \in H} \left(\mathbf{p}(h) - \frac{1}{n} \right) = \sum_{\ell \in L} \left(\frac{1}{n} - \mathbf{p}(\ell) \right) \geq \frac{\varepsilon}{2} \,. \tag{4.9}$$

From this it is not hard to show that

(i) many elements of $[n]$ must be "significantly light" in the following sense: Define $L' \subseteq L$ to be $L' = \left\{ \ell \in [n] \,:\, \mathbf{p}(\ell) < \frac{1}{n} - \frac{\varepsilon}{4n} \right\}$. Then it must be the case that $|L'| \geq (\varepsilon/4)n$.

(ii) $\mathbf{p}$ places significant weight on elements that are "significantly heavy" in the following sense: Define $H' \subseteq H$ to be $H' = \left\{ h \in [n] \,:\, \mathbf{p}(h) \geq \frac{1}{n} + \frac{\varepsilon}{4n} \right\}$. Then it must be the case that $\mathbf{p}(H') \geq (\varepsilon/4)$.

Using (i) and (ii) it is fairly straightforward to give a $\tilde{O}\big(1/\varepsilon^4\big)$-query $\mathsf{PAIRCOND}_{\mathbf{p}}$ testing algorithm as follows: we can get a point in L' with high probability by randomly sampling $O(1/\varepsilon)$ points uniformly at random from $[n]$, and we can get a point in H' with high probability by drawing $O(1/\varepsilon)$ points from $\mathsf{SAMP}_{\mathbf{p}}$. Then at least one of the $O(1/\varepsilon^2)$ pairs that have one point from the first sample and one point from the second will have a multiplicative factor difference of $1 + \Omega(\varepsilon)$ between the weight under $\mathbf{p}$ of the two points, and this can be detected by calling the procedure COMPARE (see Section 4.1.2.1). Since there are $O(1/\varepsilon^2)$ pairs and for each one the invocation of COMPARE uses $\tilde{O}(1/\varepsilon^2)$ queries, the overall sample complexity of this simple approach is $\tilde{O}(1/\varepsilon^4)$.

Our actual algorithm $\mathsf{PAIRCOND}_{\mathbf{p}}$-TEST-UNIFORM for testing uniformity extends the above ideas to get a $\tilde{O}(1/\varepsilon^2)$-query algorithm. More precisely, the algorithm works as follows: it first draws a "reference sample" of $O(1)$ points uniformly from $[n]$. Next, repeatedly for $O\big(\log \frac{1}{\varepsilon}\big)$ iterations, the algorithm draws two other samples, one uniformly from $[n]$ and the other from $\mathsf{SAMP}_{\mathbf{p}}$. (These samples have different sizes

at different iterations; intuitively, each iteration is meant to deal with a different "scale" of probability mass that points could have under $\mathbf{p}$.) At each iteration it then uses COMPARE to do comparisons between pairs of elements, one from the reference sample and the other from one of the two other samples. If $\mathbf{p}$ is ε-far from uniform, then with high probability at some iteration the algorithm will either draw a point from $\mathsf{SAMP}_{\mathbf{p}}$ that has "very big" mass under $\mathbf{p}$, or draw a point from the uniform distribution over $[n]$ that has "very small" mass under $\mathbf{p}$, and this will be detected by the comparisons to the reference points. Choosing the sample sizes and parameters for the COMPARE calls carefully at each iteration yields the improved query bound.

Algorithm 19 $\mathsf{PAIRCOND}_{\mathbf{p}}$-TEST-UNIFORM

Require: error parameter $\varepsilon > 0$; query access to $\mathsf{PAIRCOND}_{\mathbf{p}}$ oracle

1: Set $t = \log(\frac{4}{\varepsilon}) + 1$.
2: Select $q = \Theta(1)$ points $i_1, \ldots, i_q$ independently and uniformly from $[n]$.
3: **for** $j = 1$ to t **do**
4: Call the $\mathsf{SAMP}_{\mathbf{p}}$ oracle $s_j = \Theta\big(2^j \cdot t\big)$ times to obtain points $h_1, \ldots, h_{s_j}$ distributed according to $\mathbf{p}$.
5: Select s_j points $\ell_1, \ldots, \ell_{s_j}$ independently and uniformly from $[n]$.
6: **for all** pairs $(x, y) = (i_r, h_{r'})$ and $(x, y) = (i_r, \ell_{r'})$ (where $1 \le r \le q$, $1 \le r' \le s_j$) **do**
7: Call $\text{COMPARE}_{\mathbf{p}}(\{x\}, \{y\}, \Theta(\varepsilon 2^j), 2, \exp(-\Theta(t)))$.
8: **if** the COMPARE call does not return a value in $[1 - 2^{j-5}\frac{\varepsilon}{4}, 1 + 2^{j-5}\frac{\varepsilon}{4}]$ **then**
9: **return** reject (and exit).
10: **end if**
11: **end for**
12: **end for**
13: **return** accept

Proof of Theorem 4.1.16. Let m_j denote the number of $\mathsf{PAIRCOND}_{\mathbf{p}}$ queries used to run $\text{COMPARE}_{\mathbf{p}}$ in a given execution of Line 7 during the j-th iteration of the outer loop. By the setting of the parameters in each such call and Lemma 4.1.2, $m_j = O\big(\frac{t}{\varepsilon^2 2^{2j}}\big)$. It is easy to see that the algorithm only performs $\mathsf{PAIRCOND}_{\mathbf{p}}$ queries and that the total number of queries that the algorithm performs is

$$O\left(\sum_{j=1}^{t} q \cdot s_j \cdot m_j\right) = O\left(\sum_{j=1}^{t} 2^j \log\left(\frac{1}{\varepsilon}\right) \cdot \frac{\log(\frac{1}{\varepsilon})}{\varepsilon^2 2^{2j}}\right) = O\left(\frac{(\log(\frac{1}{\varepsilon}))^2}{\varepsilon^2}\right).$$

We prove Theorem 4.1.16 by arguing completeness and soundness below.

Completeness: Suppose that $\mathbf{p}$ is the uniform distribution. Then for any fixed pair of points (x, y), Lemma 4.1.2 implies that the call to COMPARE on $\{x\}, \{y\}$ in Line 7 causes the algorithm to output reject in Line 9 with probability at most $e^{-\Theta(t)} = \text{poly}(\varepsilon)$. By taking a union bound over all $\text{poly}(1/\varepsilon)$ pairs of points considered by the algorithm, the algorithm will accept with probability at least $2/3$, as required.

Soundness: Now suppose that $\mathbf{p}$ is ε-far from uniform (we assume throughout the analysis that $\varepsilon = 1/2^k$ for some integer k, which is clearly without loss of generality). We define H, L as above and further partition

H and L into "buckets" as follows: for $j = 1, \ldots, t-1 = \log(\frac{4}{\varepsilon})$, let

$$H_j \stackrel{\text{def}}{=} \left\{ h : \left(1 + 2^{j-1} \cdot \frac{\varepsilon}{4}\right) \cdot \frac{1}{n} \leq \mathbf{p}(h) < \left(1 + 2^{j} \cdot \frac{\varepsilon}{4}\right) \cdot \frac{1}{n} \right\},$$

and for $j = 1, \ldots, t-2$ let

$$L_j \stackrel{\text{def}}{=} \left\{ \ell : \left(1 - 2^{j} \cdot \frac{\varepsilon}{4}\right) \cdot \frac{1}{n} < \mathbf{p}(\ell) \leq \left(1 - 2^{j-1} \cdot \frac{\varepsilon}{4}\right) \cdot \frac{1}{n} \right\}.$$

Also define

$$H_0 \stackrel{\text{def}}{=} \left\{ h : \frac{1}{n} \leq \mathbf{p}(h) < \left(1 + \frac{\varepsilon}{4}\right) \cdot \frac{1}{n} \right\}, \quad L_0 \stackrel{\text{def}}{=} \left\{ \ell : \left(1 - \frac{\varepsilon}{4}\right) \cdot \frac{1}{n} < \mathbf{p}(\ell) < \frac{1}{n} \right\},$$

and

$$H_t \stackrel{\text{def}}{=} \left\{ h : \mathbf{p}(h) \geq \frac{2}{n} \right\}, \quad L_{t-1} \stackrel{\text{def}}{=} \left\{ \ell : \mathbf{p}(\ell) \leq \frac{1}{2n} \right\}.$$

First observe that by the definition of H_0 and L_0, we have

$$\sum_{h \in H_0} \left(\mathbf{p}(h) - \frac{1}{n}\right) \leq \frac{\varepsilon}{4} \text{ and } \sum_{\ell \in L_0} \left(\frac{1}{n} - \mathbf{p}(\ell)\right) \leq \frac{\varepsilon}{4}.$$

Therefore (by Eq. (4.9)) we have

$$\sum_{j=1}^{t} \sum_{h \in H_j} \left(\mathbf{p}(h) - \frac{1}{n}\right) \geq \frac{\varepsilon}{4} \text{ and } \sum_{j=1}^{t-1} \sum_{\ell \in L_j} \left(\frac{1}{n} - \mathbf{p}(\ell)\right) \geq \frac{\varepsilon}{4}.$$

This implies that for some $1 \leq j(H) \leq t$, and some $1 \leq j(L) \leq t-1$, we have

$$\sum_{h \in H_{j(H)}} \left(\mathbf{p}(h) - \frac{1}{n}\right) \geq \frac{\varepsilon}{4t} \text{ and } \sum_{\ell \in L_{j(L)}} \left(\frac{1}{n} - \mathbf{p}(\ell)\right) \geq \frac{\varepsilon}{4t}. \tag{4.10}$$

The rest of the analysis is divided into two cases depending on whether $|L| \geq \frac{n}{2}$ or $|H| > \frac{n}{2}$.

Case 1: $|L| \geq \frac{n}{2}$**.** In this case, with probability at least $99/100$, in Line 2 the algorithm will select at least one point $i_r \in L$. We consider two subcases: $j(H) = t$, and $j(H) \leq t-1$.

- $j(H) = t$: In this subcase, by Eq. (4.10) we have that $\sum_{h \in H_{j(H)}} \mathbf{p}(h) \geq \frac{\varepsilon}{4t}$. This implies that when $j = j(H) = t = \log(\frac{4}{\varepsilon}) + 1$, so that $s_j = s_t = \Theta\left(\frac{t}{\varepsilon}\right)$, with probability at least $99/100$ the algorithm selects a point $h_{r'} \in H_t$ in Line 4. Assume that indeed such a point $h_{r'}$ is selected. Since $\mathbf{p}(h_{r'}) \geq \frac{2}{n}$, while $\mathbf{p}(i_r) < \frac{1}{n}$, Lemma 4.1.2 implies that with probability at least $1 - \text{poly}(\varepsilon)$ the COMPARE call in Line 7 outputs either high or a value that is at least $\frac{7}{12} = \frac{1}{2} + \frac{1}{12}$. Since $\frac{7}{12} > \frac{1}{2} + 2^{j-5}\frac{\varepsilon}{4}$ for $j = t$, the algorithm will output reject in Line 9.

- $j(H) < t$: By Eq. (4.10) and the definition of the buckets, we have

$$\sum_{h \in H_{j(H)}} \left(\left(1 + 2^{j(H)} \frac{\varepsilon}{4}\right) \frac{1}{n} - \frac{1}{n} \right) \geq \frac{\varepsilon}{4t} ,$$

implying that $\left|H_{j(H)}\right| \geq \frac{n}{2^{j(H)}t}$ so that $\mathbf{p}(H_{j(H)}) \geq \frac{1}{2^{j(H)}t}$. Therefore, when $j = j(H)$ so that $s_j = \Theta\left(2^{j(H)}t\right)$, with probability at least $99/100$ the algorithm will get a point $h_{r'} \in H_{j(H)}$ in Line 4. Assume that indeed such a point $h_{r'}$ is selected. Since $\mathbf{p}(h_{r'}) \geq \left(1 + 2^{j(H)-1}\frac{\varepsilon}{4}\right)\frac{1}{n}$, while $\mathbf{p}(i_r) \leq \frac{1}{n}$, for $\alpha_{j(H)} = 2^{j(H)-1}\frac{\varepsilon}{4}$, we have

$$\frac{\mathbf{p}(h_{r'})}{\mathbf{p}(i_r)} \geq 1 + \alpha_{j(H)} .$$

Since COMPARE is called in Line 7 on the pair $\{i_r\}, \{h_{r'}\}$ with the "δ" parameter set to $\Theta(\varepsilon 2^j)$, with probability $1 - \mathrm{poly}(\varepsilon)$ the algorithm outputs reject as a result of this COMPARE call.

Case 2: $|H| > \frac{n}{2}$. This proceeds similarly to Case 1. In this case we have that with high constant probability the algorithm selects a point $i_r \in H$ in Line 2. Here we consider the subcases $j(L) = t - 1$ and $j(L) \leq t - 2$. In the first subcase we have that $\sum_{\ell \in L_t} \frac{1}{n} \geq \frac{\varepsilon}{4t}$, so that $|L_t| \geq (\frac{\varepsilon}{4t})n$, and in the second case we have that $\sum_{\ell \in L_{j(L)}} (2^{j(L)}\frac{\varepsilon}{4})\frac{1}{n} \geq \frac{\varepsilon}{4t}$, so that $\left|L_{j(L)}\right| \geq \frac{n}{2^{j(L)}t}$. The analysis of each subcase is similar to Case 1. This concludes the proof of Theorem 4.1.16. □

4.1.3.2 An $\Omega(1/\varepsilon^2)$ lower bound for $\mathsf{COND}_{\mathbf{p}}$ algorithms that test uniformity

In this subsection we give a lower bound showing that the query complexity of the $\mathsf{PAIRCOND}_{\mathbf{p}}$ algorithm of the previous subsection is essentially optimal, even for algorithms that may make general $\mathsf{COND}_{\mathbf{p}}$ queries:

Theorem 4.1.17. *Any $\mathsf{COND}_{\mathbf{p}}$ algorithm for testing whether $\mathbf{p} = \mathbf{u}$ versus $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{u}) \geq \varepsilon$ must make $\Omega(1/\varepsilon^2)$ queries.*

The high-level idea behind Theorem 4.1.17 is to reduce it to the well-known fact that distinguishing a fair coin from a $(\frac{1}{2} + 4\varepsilon)$-biased coin requires $\Omega\left(\frac{1}{\varepsilon^2}\right)$ coin tosses. We show that any q-query algorithm $\mathsf{COND}_{\mathbf{p}}$ testing algorithm A can be transformed into an algorithm A' that successfully distinguishes q tosses of a fair coin from q tosses of a $(\frac{1}{2} + 4\varepsilon)$-biased coin.

Proof of Theorem 4.1.17: First note that we may assume without loss of generality that $0 < \varepsilon \leq 1/8$. Let A be any q-query algorithm that makes $\mathsf{COND}_{\mathbf{p}}$ queries and tests whether $\mathbf{p} = \mathbf{u}$ versus $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{u}) \geq \varepsilon$. We may assume without loss of generality that in every possible execution algorithm A makes precisely q queries (this will be convenient later).

Let $\mathbf{p}_{\mathsf{no}}$ be the distribution that has $\mathbf{p}_{\mathsf{no}}(i) = \frac{1+2\varepsilon}{n}$ for each $i \in \left[1, \frac{n}{2}\right]$ and has $\mathbf{p}_{\mathsf{no}}(i) = \frac{1-2\varepsilon}{n}$ for each $i \in \left[\frac{n}{2} + 1, n\right]$. (This is the "no"-distribution for our lower bound; it is ε-far in variation distance from the

uniform distribution $\mathbf{u}$.) By the guarantee of a testing algorithm, it must be the case that

$$Z := \left|\Pr\left[A^{\mathsf{COND}_{\mathbf{p_{no}}}} \text{ outputs } \mathsf{accept}\right] - \Pr\left[A^{\mathsf{COND}_{\mathbf{u}}} \text{ outputs } \mathsf{accept}\right]\right| \geq 1/3.$$

The proof works by showing that given A as described above, there must exist an algorithm A' with the following properties: A' is given as input a q-bit string $(b_1, \ldots, b_q) \in \{0,1\}^q$. Let $\mathbf{p}_0$ denote the uniform distribution over $\{0,1\}^q$ and let $\mathbf{p}_{4\varepsilon}$ denote the distribution over $\{0,1\}^q$ in which each coordinate is independently set to 1 with probability $1/2 + 4\varepsilon$. Then algorithm A' has

$$\left|\Pr_{b\sim\mathbf{p}_0}[A'(b) \text{ outputs } \mathsf{accept}] - |\Pr_{b\sim\mathbf{p}_{4\varepsilon}}[A'(b) \text{ outputs } \mathsf{accept}]\right| = Z. \tag{4.11}$$

Given (4.11), by the data processing inequality for total variation distance (Fact 1.4.2) we have that $Z \leq \mathrm{d_{TV}}(\mathbf{p}_0, \mathbf{p}_{4\varepsilon})$. It is easy to see that $\mathrm{d_{TV}}(\mathbf{p}_0, \mathbf{p}_{4\varepsilon})$ is precisely equal to the variation distance $\mathrm{d_{TV}}(\mathrm{Bin}(q, 1/2), \mathrm{Bin}(q, 1/2 + 4\varepsilon))$. However, in order for the variation distance between these two binomial distributions to be as large as $1/3$ it must be the case that $q \geq \Omega(1/\varepsilon^2)$:

Fact 4.1.18 (Distinguishing Fair from Biased Coin)**.** *Suppose $m \leq \frac{c}{\varepsilon^2}$, with c a sufficiently small constant and $\varepsilon \leq 1/8$. Then,*

$$\mathrm{d_{TV}}\left(\mathrm{Bin}\left(m, \frac{1}{2}\right), \mathrm{Bin}\left(m, \frac{1}{2} + 4\varepsilon\right)\right) \leq \frac{1}{3}.$$

(Fact 4.1.18 is well known; it follows, for example, as an immediate consequence of Equations (2.15) and (2.16) of [6].) Thus to prove Theorem 4.1.17 it remains only to describe algorithm A' and prove Eq. (4.11).

As suggested above, algorithm A' uses algorithm A; in order to do this, it must perfectly simulate the $\mathsf{COND}_{\mathbf{p}}$ oracle that A requires, both in the case when $\mathbf{p} = \mathbf{u}$ and in the case when $\mathbf{p} = \mathbf{p_{no}}$. We show below that when its input b is drawn from $\mathbf{p}_0$ then A' can perfectly simulate the execution of A when it is run on the $\mathsf{COND}_{\mathbf{u}}$ oracle, and when b is drawn from $\mathbf{p}_{4\varepsilon}$ then A' can perfectly simulate the execution of A when it is run on the $\mathsf{COND}_{\mathbf{p_{no}}}$ oracle.

Fix any step $1 \leq t \leq q$. We now describe how A' perfectly simulates the t-th step of the execution of A (i.e. the t-th call to $\mathsf{COND}_{\mathbf{p}}$ that A makes, and the response of $\mathsf{COND}_{\mathbf{p}}$). We may inductively assume that A' has perfectly simulated the first $t-1$ steps of the execution of A.

For each possible prefix of $t-1$ query-response pairs to $\mathsf{COND}_{\mathbf{p}}$

$$PREFIX = ((S_1, s_1), ..., (S_{t-1}, s_{t-1}))$$

(where each $S_i \subseteq [n]$ and each $s_i \in S_i$), there is some distribution $\P_{A,PREFIX}$ over possible t-th query sets S_t that A would make given that its first $t-1$ query-response pairs were $PREFIX$. So for a set $S_t \subseteq [n]$ and a possible prefix $PREFIX$, the value $P_{A,PREFIX}(S_t)$ is the probability that algorithm A, having had the transcript of its execution thus far be $PREFIX$, generates set S_t as its t-th query set. For any query set $S \subseteq [n]$, let us write S as a disjoint union $S = S_0 \sqcup S_1$, where $S_0 = S \cap \left[1, \frac{n}{2}\right]$ and $S_1 = S \cap \left[\frac{n}{2} + 1, n\right]$. We

may assume that every query S ever used by A has $|S_0|, |S_1| \geq 1$ (for otherwise A could perfectly simulate the response of $\mathsf{COND}_{\mathbf{p}}(S)$ whether $\mathbf{p}$ were $\mathbf{u}$ or $\mathbf{p}_{\mathsf{no}}$ by simply choosing a uniform point from S, so there would be no need to call $\mathsf{COND}_{\mathbf{p}}$ on such an S). Thus we may assume that $P_{A,PREFIX}(S)$ is nonzero only for sets S that have $|S_0|, |S_1| \geq 1$.

Consider the bit $b_t \in \{0,1\}$. As noted above, we inductively have that (whether $\mathbf{p}$ is $\mathbf{u}$ or $\mathbf{p}_{\mathsf{no}}$) the algorithm A' has perfectly simulated the execution of A for its first $t-1$ query-response pairs; in this simulation some prefix $PREFIX = ((S_1, s_1), \ldots, (S_{t-1}, s_{t-1}))$ of query-response pairs has been constructed. If $b = (b_1, \ldots, b_q)$ is distributed according to $\mathbf{p}_0$ then $PREFIX$ is distributed exactly according to the distribution of A's prefixes of length $t-1$ when A is run with $\mathsf{COND}_{\mathbf{u}}$, and if $b = (b_1, \ldots, b_q)$ is distributed according to $\mathbf{p}_{4\varepsilon}$ then the distribution of $PREFIX$ is exactly the distribution of A's prefixes of length $t-1$ when A is run with $\mathsf{COND}_{\mathbf{p}_{\mathsf{no}}}$.

Algorithm A' simulates the t-th stage of the execution of A as follows:

1. Randomly choose a set $S \subseteq [n]$ according to the distribution $P_{A,PREFIX}$; let $S = S_0 \sqcup S_1$ be the set that is selected. Let us write $\alpha(S)$ to denote $|S_1|/|S_0|$ (so $\alpha(S) \in [2/n, n/2]$).
2. If $b_t = 1$ then set the bit $\sigma \in \{0,1\}$ to be 1 with probability u_t and to be 0 with probability $1 - u_t$. If $b_t = 0$ then set σ to be 1 with probability v_t and to be 0 with probability $1 - v_t$. (We specify the exact values of u_t, v_t below.)
3. Set s to be a uniform random element of S_σ. Output the query-response pair $(S_t, s_t) = (S, s)$.

It is clear that Step 1 above perfectly simulates the t-th query that algorithm A would make (no matter what is the distribution $\mathbf{p}$). To show that the t-th response is simulated perfectly, we must show that

(i) if b_t is uniform random over $\{0,1\}$ then s is distributed exactly as it would be distributed if A were being run on $\mathsf{COND}_{\mathbf{u}}$ and had just proposed S as a query to $\mathsf{COND}_{\mathbf{u}}$; i.e. we must show that s is a uniform random element of S_1 with probability $p(\alpha) \stackrel{\text{def}}{=} \frac{\alpha}{\alpha+1}$ and is a uniform random element of S_0 with probability $1 - p(\alpha)$.

(ii) if $b_t \in \{0,1\}$ has $\Pr[b_t = 1] = 1/2 + 4\varepsilon$, then s is distributed exactly as it would be distributed if A were being run on $\mathsf{COND}_{\mathbf{p}_{\mathsf{no}}}$ and had just proposed S as a query to $\mathsf{COND}_{\mathbf{u}}$; i.e. we must show that s is a uniform random element of S_1 with probability $q(\alpha) \stackrel{\text{def}}{=} \frac{\alpha}{\alpha+(1+2\varepsilon)/(1-2\varepsilon)}$ and is a uniform random element of S_0 with probability $1 - q(\alpha)$.

By (i), we require that

$$\frac{u_t}{2} + \frac{v_t}{2} = p(\alpha) = \frac{\alpha}{\alpha+1}, \tag{4.12}$$

and by (ii) we require that

$$\left(\frac{1}{2} + 4\varepsilon\right) u_t + \left(\frac{1}{2} - 4\varepsilon\right) v_t = q(\alpha) = \frac{\alpha}{\alpha + \frac{1+2\varepsilon}{1-2\varepsilon}} \tag{4.13}$$

It is straightforward to check that

$$u_t = \frac{\alpha}{\alpha+1}\left(1 - \frac{1}{2((1-2\varepsilon)\alpha+1+2\varepsilon)}\right), \qquad v_t = \frac{\alpha}{\alpha+1}\left(1 + \frac{1}{2((1-2\varepsilon)\alpha+1+2\varepsilon)}\right)$$

satisfy the above equations, and that for $0 < \alpha$, $0 < \varepsilon \leq 1/8$ we have $0 \leq u_t, v_t \leq 1$. So indeed A' perfectly simulates the execution of A in all stages $t = 1, \ldots, q$. Finally, after simulating the t-th stage algorithm A' outputs whatever is output by its simulation of A, so Equation (4.11) indeed holds. This concludes the proof of Theorem 4.1.17. □

4.1.4 Testing equivalence to a known distribution $\mathbf{p}^*$

4.1.4.1 A $\mathrm{poly}(\log n, 1/\varepsilon)$-query $\mathsf{PAIRCOND}_{\mathbf{p}}$ algorithm

In this subsection we present an algorithm PAIRCOND-TEST-KNOWN and prove the following theorem:

Theorem 4.1.19. PAIRCOND-TEST-KNOWN *is a* $\tilde{O}((\log n)^4/\varepsilon^4)$*-query* $\mathsf{PAIRCOND}_{\mathbf{p}}$ *testing algorithm for testing equivalence to a known distribution* $\mathbf{p}^*$*. That is, for every pair of distributions* $\mathbf{p}, \mathbf{p}^*$ *over* $[n]$ *(such that* $\mathbf{p}^*$ *is fully specified and there is* PAIRCOND *query access to* $\mathbf{p}$*) the algorithm outputs* ***accept*** *with probability at least* $2/3$ *if* $\mathbf{p} = \mathbf{p}^*$ *and outputs* ***reject*** *with probability at least* $2/3$ *if* $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{p}^*) \geq \varepsilon$*.*

Intuition. Let $\mathbf{p}^*$ be a fully specified distribution, and let $\mathbf{p}$ be a distribution that may be accessed via a $\mathsf{PAIRCOND}_{\mathbf{p}}$ oracle. The high-level idea of the PAIRCOND-TEST-KNOWN algorithm is the following: As in the case of testing uniformity, we shall try to "catch" a pair of points x, y such that $\frac{\mathbf{p}(x)}{\mathbf{p}(y)}$ differs significantly from $\frac{\mathbf{p}^*(x)}{\mathbf{p}^*(y)}$ (so that calling $\textsc{Compare}_{\mathbf{p}}$ on $\{x\}, \{y\}$ will reveal this difference). In the uniformity case, where $\mathbf{p}^*(z) = 1/n$ for every z (so that $\frac{\mathbf{p}^*(x)}{\mathbf{p}^*(x)+\mathbf{p}^*(y)} = 1/2$), to get a $\mathrm{poly}(1/\varepsilon)$-query algorithm it was sufficient to show that sampling $\Theta(1/\varepsilon)$ points uniformly (i.e., according to $\mathbf{p}^*$) with high probability yields a point x for which $\mathbf{p}(x) < \mathbf{p}^*(x) - \Omega(\varepsilon/n)$, and that sampling $\Theta(1/\varepsilon)$ points from $\mathsf{SAMP}_{\mathbf{p}}$ with high probability yields a point y for which $\mathbf{p}(x) > \mathbf{p}^*(y) + \Omega(\varepsilon/n)$. However, for general $\mathbf{p}^*$ it is not sufficient to get such a pair because it is possible that $\mathbf{p}^*(y)$ could be much larger than $\mathbf{p}^*(x)$. If this were the case then it might happen that both $\frac{\mathbf{p}^*(x)}{\mathbf{p}^*(y)}$ and $\frac{\mathbf{p}(x)}{\mathbf{p}(y)}$ are very small, so calling $\textsc{Compare}_{\mathbf{p}}$ on $\{x\}, \{y\}$ cannot efficiently demonstrate that $\frac{\mathbf{p}^*(x)}{\mathbf{p}^*(y)}$ differs from $\frac{\mathbf{p}(x)}{\mathbf{p}(y)}$.

To address this issue we partition the points into $O(\log n/\varepsilon)$ "buckets" so that within each bucket all points have similar probability according to $\mathbf{p}^*$. We show that if $\mathbf{p}$ is ε-far from $\mathbf{p}^*$, then either the probability weight of one of these buckets according to $\mathbf{p}$ differs significantly from what it is according to $\mathbf{p}^*$ (which can be observed by sampling from $\mathbf{p}$), or we can get a pair $\{x, y\}$ *that belong to the same bucket* and for which $\mathbf{p}(x)$ is sufficiently smaller than $\mathbf{p}^*(x)$ and $\mathbf{p}(y)$ is sufficiently larger than $\mathbf{p}^*(y)$. For such a pair COMPARE will efficiently give evidence that $\mathbf{p}$ differs from $\mathbf{p}^*$.

The algorithm and its analysis. We define some quantities that are used in the algorithm and its analysis. Let $\eta \stackrel{\text{def}}{=} \varepsilon/c$ for some sufficiently large constant c that will be determined later. As described above we partition the domain elements $[n]$ into "buckets" according to their probability weight in $\mathbf{p}^*$. Specifically, for $j = 1, \ldots, \lceil \log(n/\eta) + 1 \rceil$, we let

$$B_j \stackrel{\text{def}}{=} \{\, x \in [n] \,:\, 2^{j-1} \cdot \eta/n \le \mathbf{p}^*(x) < 2^j \cdot \eta/n \,\} \tag{4.14}$$

and we let $B_0 \stackrel{\text{def}}{=} \{x \in [n] : \mathbf{p}^*(x) < \eta/n\}$. Let $b \stackrel{\text{def}}{=} \lceil \log(n/\eta) + 1 \rceil + 1$ denote the number of buckets.

We further define $J^h \stackrel{\text{def}}{=} \{j : \mathbf{p}^*(B_j) \ge \eta/b\}$ to denote the set of indices of "heavy" buckets, and let $J^\ell \stackrel{\text{def}}{=} \{\, j \,:\, \mathbf{p}^*(B_j) < \eta/b \,\}$ denote the set of indices of "light" buckets. Note that we have

$$\sum_{j \in J^\ell \cup \{0\}} \mathbf{p}^*(B_j) < 2\eta. \tag{4.15}$$

Algorithm 20 $\mathsf{PAIRCOND}_{\mathbf{p}}$-TEST-KNOWN

Require: error parameter $\varepsilon > 0$; query access to $\mathsf{PAIRCOND}_{\mathbf{p}}$ oracle; explicit description $(\mathbf{p}^*(1), \ldots, \mathbf{p}^*(n))$ of distribution $\mathbf{p}^*$

Call the $\mathsf{SAMP}_{\mathbf{p}}$ oracle $m = \Theta(b^2(\log b)/\eta^2)$ times to obtain points $h_1, \ldots, h_m$ distributed according to $\mathbf{p}$.
for $j = 0$ to b **do**
    Let $\widehat{\mathbf{p}}(B_j)$ be the fraction of points $h_1, \ldots, h_m$ that lie in B_j (where the buckets B_j are as defined in Eq. (4.14)).
    if some j has $|\mathbf{p}^*(B_j) - \widehat{\mathbf{p}}(B_j)| > \eta/b$ **then**
        return reject and exit
    end if
end for
Select $s = \Theta(b/\varepsilon)$ points $x_1, \ldots, x_s$ independently from $\mathbf{p}^*$.
Call the $\mathsf{SAMP}_{\mathbf{p}}$ oracle $s = \Theta(b/\varepsilon)$ times to obtain points $y_1, \ldots, y_s$ distributed according to $\mathbf{p}$.
for all pairs (x_i, y_j) (where $1 \le i, j \le s$) such that $\frac{D^*(x)}{D^*(y)} \in [1/2, 2]$ **do**
    Call COMPARE$(\{x\}, \{y\}, \eta/(4b), 2, 1/(10s^2))$
    if COMPARE returns low or a value smaller than $(1 - \eta/(2b)) \cdot \frac{D^*(x)}{D^*(y)}$ **then**
        return reject (and exit)
    end if
end for
return accept

The query complexity of the algorithm is dominated by the number of $\mathsf{PAIRCOND}_{\mathbf{p}}$ queries performed in the executions of COMPARE, which by Lemma 4.1.2 is upper bounded by

$$O(s^2 \cdot b^2 \cdot (\log s)/\eta^2) = O\left(\frac{(\log \frac{n}{\varepsilon})^4 \cdot \log\left((\log \frac{n}{\varepsilon})/\varepsilon\right)}{\varepsilon^4}\right).$$

We argue completeness and soundness below.

Completeness: Suppose that $\mathbf{p} = \mathbf{p}^*$. Since the expected value of $\widehat{\mathbf{p}}(B_j)$ (defined in Line 3) is precisely $\mathbf{p}^*(B_j)$, for any fixed value of $j \in \{0, \ldots, \lceil \log(n/\eta) + 1 \rceil\}$ an additive Chernoff bound implies that $|\mathbf{p}^*(B_j) - \widehat{\mathbf{p}}(B_j)| > \eta/b$ with failure probability at most $1/(10b)$. By a union bound over all b values of j, the algorithm outputs reject in Line 5 with probability at most $1/10$. Later in the algorithm, since $\mathbf{p} = \mathbf{p}^*$, no matter what points x_i, y_j are sampled from $\mathbf{p}^*$ and $\mathbf{p}$ respectively, the following holds for each pair (x_i, y_j) such that $\mathbf{p}^*(x)/\mathbf{p}^*(y) \in [1/2, 2]$. By Lemma 4.1.2 (and the setting of the parameters in the calls to COMPARE), the probability that COMPARE returns low or a value smaller than $(1 - \delta/(2b)) \cdot (\mathbf{p}^*(x)/\mathbf{p}^*(y))$, is at most $1/(10s^2)$. A union bound over all (at most s^2) pairs (x_i, y_j) for which $\mathbf{p}^*(x)/\mathbf{p}^*(y) \in [1/2, 2]$, gives that the probability of outputting reject in Line 13 is at most $1/10$. Thus with overall probability at least $8/10$ the algorithm outputs accept.

Soundness: Now suppose that $\mathrm{d_{TV}}(\mathbf{p}, \mathbf{p}^*) \geq \varepsilon$; our goal is to show that the algorithm rejects with probability at least $2/3$. Since the algorithm rejects if any estimate $\widehat{\mathbf{p}}(B_j)$ obtained in Line 3 deviates from $\mathbf{p}^*(B_j)$ by more than $\pm\eta/b$, we may assume that all these estimates are indeed $\pm\eta/b$-close to the values $\mathbf{p}^*(B_j)$ as required. Moreover, by an additive Chernoff bound (as in the completeness analysis), we have that with overall failure probability at most $1/10$, each j has $|\widehat{\mathbf{p}}(B_j) - \mathbf{p}(B_j)| \leq \eta/b$; we condition on this event going forth. Thus, for every $0 \leq j \leq b$,

$$\mathbf{p}^*(B_j) - 2\eta/b \leq \mathbf{p}(B_j) \leq \mathbf{p}^*(B_j) + 2\eta/b \,. \tag{4.16}$$

Recalling the definition of J^ℓ and Eq. (4.15), we see that

$$\sum_{j \in J^\ell \cup \{0\}} \mathbf{p}(B_j) < 4\eta \,. \tag{4.17}$$

Let

$$d_j \stackrel{\text{def}}{=} \sum_{x \in B_j} |\mathbf{p}^*(x) - \mathbf{p}(x)| \,, \tag{4.18}$$

so that $\|\mathbf{p}^* - \mathbf{p}\|_1 = \sum_j d_j$. By Eqs. (4.15) and (4.17), we have

$$\sum_{j \in J^\ell \cup \{0\}} d_j \leq \sum_{j \in J^\ell \cup \{0\}} (\mathbf{p}^*(B_j) + \mathbf{p}(B_j)) \leq 6\eta \,. \tag{4.19}$$

Since we have (by assumption) that $\|\mathbf{p}^* - \mathbf{p}\|_1 = 2\,\mathrm{d_{TV}}(\mathbf{p}^*, \mathbf{p}) \geq 2\varepsilon$, we get that

$$\sum_{j \in J^h \setminus \{0\}} d_j > 2\varepsilon - 6\eta \,. \tag{4.20}$$

Let $N_j \stackrel{\text{def}}{=} |B_j|$ and observe that $N_j \leq \mathbf{p}^*(B_j)/p_j \leq 1/p_j$, where $p_j \stackrel{\text{def}}{=} 2^{j-1} \cdot \eta/n$ is the lower bound on the probability (under $\mathbf{p}^*$) of all elements in B_j. For each B_j such that $j \in J^h \setminus \{0\}$, let

$H_j \stackrel{\text{def}}{=} \{x \in B_j : \mathbf{p}(x) > \mathbf{p}^*(x)\}$ and $L_j \stackrel{\text{def}}{=} \{x \in B_j : \mathbf{p}(x) < \mathbf{p}^*(x)\}$. Similarly to the "testing uniformity" analysis, we have that

$$\sum_{x \in L_j} (\mathbf{p}^*(x) - \mathbf{p}(x)) + \sum_{x \in H_j} (\mathbf{p}(x) - \mathbf{p}^*(x)) = d_j \,. \tag{4.21}$$

Eq. (4.16) may be rewritten as

$$\left| \sum_{x \in L_j} (\mathbf{p}^*(x) - \mathbf{p}(x)) - \sum_{x \in H_j} (\mathbf{p}(x) - \mathbf{p}^*(x)) \right| \leq 2\eta/b \,, \tag{4.22}$$

and so we have both

$$\sum_{x \in L_j} (\mathbf{p}^*(x) - \mathbf{p}(x)) \geq d_j/2 - \eta/b \quad \text{and} \quad \sum_{x \in H_j} (\mathbf{p}(x) - \mathbf{p}^*(x)) \geq d_j/2 - \eta/b \,. \tag{4.23}$$

Also similarly to what we had before, let $H'_j \stackrel{\text{def}}{=} \{x \in B_j : \mathbf{p}(x) > \mathbf{p}^*(x) + \eta/(bN_j)\}$, and $L'_j \stackrel{\text{def}}{=} \{x \in B_j : \mathbf{p}(x) < \mathbf{p}^*(x) - \eta/(bN_j)\}$ (recall that $N_j = |B_j|$); these are the elements of B_j that are "significantly heavier" (lighter, respectively) under $\mathbf{p}$ than under $\mathbf{p}^*$. We have

$$\sum_{x \in L_j \setminus L'_j} (\mathbf{p}^*(x) - \mathbf{p}(x)) \leq \eta/b \quad \text{and} \quad \sum_{x \in H_j \setminus H'_j} (\mathbf{p}(x) - \mathbf{p}^*(x)) \leq \eta/b \,. \tag{4.24}$$

By Eq. (4.20), there exists $j^* \in J^h \setminus \{0\}$ for which $d_{j^*} \geq (2\varepsilon - 6\eta)/b$. For this index, applying Eqs. (4.23) and (4.24), we get that

$$\sum_{x \in L'_{j^*}} \mathbf{p}^*(x) \geq \sum_{x \in L'_{j^*}} (\mathbf{p}^*(x) - \mathbf{p}(x)) \geq (\varepsilon - 5\eta)/b \,, \tag{4.25}$$

and similarly,

$$\sum_{x \in H'_{j^*}} \mathbf{p}(x) \geq \sum_{x \in H'_{j^*}} (\mathbf{p}(x) - \mathbf{p}^*(x)) \geq (\varepsilon - 5\eta)/b \,. \tag{4.26}$$

Recalling that $\eta = \varepsilon/c$ and setting the constant c to 6, we have that $(\varepsilon - 5\eta)/b = \varepsilon/6b$. Since $s = \Theta(b/\varepsilon)$, with probability at least 9/10 it is the case both that some x_i drawn in Line 8 belongs to L'_{j^*} and that some $y_{i'}$ drawn in Line 9 belongs to H'_{j^*}. By the definitions of L'_{j^*} and H'_{j^*} and the fact for each $j > 0$ it holds that $N_j \leq 1/p_j$ and $p_j \leq \mathbf{p}^*(x) < 2p_j$ for each $x_i \in B_j$, we have that

$$\mathbf{p}(x_i) < \mathbf{p}^*(x_i) - \eta/(bN_{j^*}) \leq \mathbf{p}^*(x_i) - (\eta/b)p_{j^*} \leq (1 - \eta/(2b))\mathbf{p}^*(x_i) \tag{4.27}$$

and

$$\mathbf{p}(y_{i'}) > \mathbf{p}^*(y_{i'}) + \eta/(bN_{j^*}) \geq \mathbf{p}^*(y_{i'}) + (\eta/b)p_j \geq (1 + \eta/(2b))\mathbf{p}^*(y_{i'}) \,. \tag{4.28}$$

Therefore,

$$\frac{\mathbf{p}(x_i)}{\mathbf{p}(y_{i'})} < \frac{1-\eta/(2b)}{1+\eta/(2b)} \cdot \frac{\mathbf{p}^*(x_i)}{\mathbf{p}^*(y_{i'})} < \left(1-\frac{3\eta}{4b}\right) \cdot \frac{\mathbf{p}^*(x_i)}{\mathbf{p}^*(y_{i'})} \,. \tag{4.29}$$

By Lemma 4.1.2, with probability at least $1 - 1/(10s^2)$, the output of COMPARE is either low or is at most $\left(1-\frac{3\eta}{4b}\right) \cdot \left(1+\frac{\eta}{4b}\right) < \left(1-\frac{\eta}{2b}\right)$, causing the algorithm to reject. Thus the overall probability that the algorithm outputs reject is at least $8/10 - 1/(10s^2) > 2/3$, and the theorem is proved. □

4.1.4.2 A $(\log n)^{\Omega(1)}$ lower bound for $\mathsf{PAIRCOND}_{\mathbf{p}}$

In this subsection we prove that any $\mathsf{PAIRCOND}_{\mathbf{p}}$ algorithm for testing equivalence to a known distribution must have query complexity at least $(\log n)^{\Omega(1)}$:

Theorem 4.1.20. *Fix $\varepsilon = 1/2$. There is a distribution $\mathbf{p}^*$ over $[n]$ (described below), which is such that any $\mathsf{PAIRCOND}_{\mathbf{p}}$ algorithm for testing whether $\mathbf{p} = \mathbf{p}^*$ versus $\mathrm{d_{TV}}(\mathbf{p}, \mathbf{p}^*) \geq \varepsilon$ must make $\Omega\left(\sqrt{\frac{\log n}{\log\log n}}\right)$ queries.*

The distribution $\mathbf{p}^*$. Fix parameters $r = \Theta\left(\frac{\log n}{\log\log n}\right)$ and $K = \Theta(\log n)$. We partition $[n]$ from left (1) to right (n) into $2r$ consecutive intervals $B_1, \ldots, B_{2r}$, which we henceforth refer to as "buckets." The i-th bucket has $|B_i| = K^i$ (we may assume without loss of generality that n is of the form $\sum_{i=1}^{2r} K^i$). The distribution $\mathbf{p}^*$ assigns equal probability weight to each bucket, so $\mathbf{p}^*(B_i) = 1/(2r)$ for all $1 \leq i \leq 2r$. Moreover $\mathbf{p}^*$ is uniform within each bucket, so for all $j \in B_i$ we have $\mathbf{p}^*(j) = 1/(2rK^i)$. This completes the specification of $\mathbf{p}^*$.

To prove the lower bound we construct a probability distribution $\mathcal{D}_{\mathsf{no}}$ over possible no-distributions. To define the distribution $\mathcal{D}_{\mathsf{no}}$ it will be useful to have the notion of a "bucket-pair." A bucket-pair U_i is $U_i = B_{2i-1} \cup B_{2i}$, i.e. the union of the i-th pair of consecutive buckets.

A distribution $\mathbf{p}$ drawn from $\mathcal{D}_{\mathsf{no}}$ is obtained by selecting a string $\pi = (\pi_1, \ldots, \pi_r)$ uniformly at random from $\{\downarrow\uparrow, \uparrow\downarrow\}^r$ and setting $\mathbf{p}$ to be $\mathbf{p}_\pi$, which we now define. The distribution $\mathbf{p}_\pi$ is obtained by perturbing $\mathbf{p}^*$ in the following way: for each bucket-pair $U_i = (B_{2i-1}, B_{2i})$,

- If $\pi_i = \uparrow\downarrow$ then the weight of B_{2i-1} is uniformly "scaled up" from $1/(2r)$ to $3/(4r)$ (keeping the distribution uniform within B_{2i-1}) and the weight of B_{2i} is uniformly "scaled down" from $1/(2r)$ to $1/(4r)$ (likewise keeping the distribution uniform within B_{2i}).
- If $\pi_i = \downarrow\uparrow$ then the weight of B_{2i-1} is uniformly "scaled down" from $1/(2r)$ to $1/(4r)$ and the weight of B_{2i} is uniformly "scaled up" from $1/(2r)$ to $3/(4r)$.

Note that for any distribution $\mathbf{p}$ in the support of $\mathcal{D}_{\mathsf{no}}$ and any $1 \leq i \leq r$ we have that $\mathbf{p}(U_i) = \mathbf{p}^*(U_i) = 1/r$.

Every distribution $\mathbf{p}$ in the support of $\mathcal{D}_{\mathsf{no}}$ has $\mathrm{d_{TV}}(\mathbf{p}^*, \mathbf{p}) = 1/2$. Thus Theorem 4.1.20 follows immediately from the following:

Theorem 4.1.21. *Let A be any (possibly adaptive) algorithm. which makes at most $q \leq \frac{1}{3} \cdot \sqrt{r}$ calls to*

$\mathsf{PAIRCOND}_{\mathbf{p}}$. *Then*

$$\left| \Pr_{D \leftarrow \mathcal{D}_{no}} \left[A^{\mathsf{PAIRCOND}_{\mathbf{p}}} \textit{ outputs } \textsf{accept} \right] - \Pr \left[A^{\mathsf{PAIRCOND}_{\mathbf{p}^*}} \textit{ outputs } \textsf{accept} \right] \right| \leq 1/5. \tag{4.30}$$

Note that in the first probability of Eq. (4.30) the randomness is over the draw of $\mathbf{p}$ from $\mathcal{D}_{\mathsf{no}}$, the internal randomness of A in selecting its query sets, and the randomness of the responses to the $\mathsf{PAIRCOND}_{\mathbf{p}}$ queries. In the second probability the randomness is just over the internal coin tosses of A and the randomness of the responses to the $\mathsf{PAIRCOND}_{\mathbf{p}}$ queries.

Intuition for Theorem 4.1.21. A very high-level intuition for the lower bound is that $\mathsf{PAIRCOND}_{\mathbf{p}}$ queries are only useful for "comparing" points whose probabilities are within a reasonable multiplicative ratio of each other. But $\mathbf{p}^*$ and every distribution $\mathbf{p}$ in the support of $\mathcal{D}_{\mathsf{no}}$ are such that every two points either have the same probability mass under all of these distributions (so a $\mathsf{PAIRCOND}_{\mathbf{p}}$ query is not informative), or else the ratio of their probabilities is so skewed that a small number of $\mathsf{PAIRCOND}_{\mathbf{p}}$ queries is not useful for comparing them.

In more detail, we may suppose without loss of generality that in every possible execution, algorithm A first makes q calls to $\mathsf{SAMP}_{\mathbf{p}}$ and then makes q (possibly adaptive) calls to $\mathsf{PAIRCOND}_{\mathbf{p}}$. The more detailed intuition for the lower bound is as follows: First consider the $\mathsf{SAMP}_{\mathbf{p}}$ calls. Since every possible $\mathbf{p}$ (whether $\mathbf{p}^*$ or a distribution drawn from $\mathcal{D}_{\mathsf{no}}$) puts weight $1/r$ on each bucket-pair $U_1, \ldots, U_r$, a birthday paradox argument implies that in both scenarios, with probability at least $9/10$ (over the randomness in the responses to the $\mathsf{SAMP}_{\mathbf{p}}$ queries) no two of the $q \leq \frac{1}{3}\sqrt{r}$ calls to $\mathsf{SAMP}_{\mathbf{p}}$ return points from the same bucket-pair. Conditioned on this, the distribution of responses to the $\mathsf{SAMP}_{\mathbf{p}}$ queries is exactly the same under $\mathbf{p}^*$ and under $\mathbf{p}$ where $\mathbf{p}$ is drawn randomly from $\mathcal{D}_{\mathsf{no}}$.

For the pair queries, the intuition is that in either setting (whether the distribution $\mathbf{p}$ is $\mathbf{p}^*$ or a randomly chosen distribution from $\mathcal{D}_{\mathsf{no}}$), making q pair queries will with $1 - o(1)$ probability provide no information that the tester could not simulate for itself. This is because any pair query $\mathsf{PAIRCOND}_{\mathbf{p}}(\{x, y\})$ either has x, y in the same bucket B_i or in different buckets $B_i \neq B_j$ with $i < j$. If x, y are both in the same bucket B_i then in either setting $\mathsf{PAIRCOND}_{\mathbf{p}}(\{x, y\})$ is equally likely to return x or y. If they belong to buckets B_i, B_j with $i < j$ then in either setting $\mathsf{PAIRCOND}_{\mathbf{p}}(\{x, y\})$ will return the one that belongs to P_i with probability $1 - 1/\Theta(K^{j-i}) \geq 1 - 1/\Omega(K)$.

Proof of Theorem 4.1.21: As described above, we may fix A to be any $\mathsf{PAIRCOND}_{\mathbf{p}}$ algorithm that makes exactly q calls to $\mathsf{SAMP}_{\mathbf{p}}$ followed by exactly q adaptive calls to $\mathsf{PAIRCOND}_{\mathbf{p}}$.

A *transcript* for A is a full specification of the sequence of interactions that A has with the $\mathsf{PAIRCOND}_{\mathbf{p}}$ oracle in a given execution. More precisely, it is a pair (Y, Z) where $Y = (s_1, \ldots, s_q) \in [n]^q$ and $Z = (\{x_1, y_1\}, p_1), \ldots, (\{x_q, y_q\}, p_q))$, where $p_i \in \{x_i, y_i\}$ and $x_i, y_i \in [n]$. The idea is that Y is a possible sequence of responses that A might receive to the initial q $\mathsf{SAMP}_{\mathbf{p}}$ queries, $\{x_i, y_i\}$ is a possible pair that

could be the input to an i-th $\mathsf{PAIRCOND}_{\mathbf{p}}$ query, and p_i is a possible response that could be received from that query.

We say that a *length-i transcript prefix* is a pair (Y, Z^i) where Y is as above and $Z^i = ((\{x_1, y_1\}, p_1), \ldots, (\{x_i, y_i\}, p_i))$. A $\mathsf{PAIRCOND}$ algorithm A may be viewed as a collection of distributions over pairs $\{x, y\}$ in the following way: for each length-i transcript-prefix (Y, Z^i) $(0 \leq i \leq q-1)$, there is a distribution over pairs $\{x_{i+1}, y_{i+1}\}$ that A would use to select the $(i+1)$-st query pair for $\mathsf{PAIRCOND}_{\mathbf{p}}$ given that the length-i transcript prefix of A's execution thus far was (Y, Z^i). We write $\mathrm{T}_{(Y,Z^i)}$ to denote this distribution over pairs.

Let $\P^*$ denote the distribution over transcripts induced by running A with oracle $\mathsf{PAIRCOND}_{D^*}$. Let $\P^{\mathsf{no}}$ denote the distribution over transcripts induced by first (i) drawing $\mathbf{p}$ from $\mathcal{D}_{\mathsf{no}}$, and then (ii) running A with oracle $\mathsf{PAIRCOND}_{\mathbf{p}}$. To prove Theorem 4.1.21 it is sufficient to prove that the distribution over transcripts of A is statistically close whether the oracle is $\mathbf{p}^*$ or is a random $\mathbf{p}$ drawn from $\mathcal{D}_{\mathsf{no}}$, i.e. it is sufficient to prove that

$$\mathrm{d}_{\mathrm{TV}}(\P^*, \P^{\mathsf{no}}) \leq 1/5. \tag{4.31}$$

For our analysis we will need to consider variants of algorithm A that, rather than making q calls to $\mathsf{PAIRCOND}_{\mathbf{p}}$, instead "fake" the final $q-k$ of these $\mathsf{PAIRCOND}_{\mathbf{p}}$ queries as described below. For $0 \leq k \leq q$ we define $A^{(k)}$ to be the algorithm that works as follows:

1. $A^{(k)}$ exactly simulates the execution of A in making an initial q $\mathsf{SAMP}_{\mathbf{p}}$ calls and making the first k $\mathsf{PAIRCOND}_{\mathbf{p}}$ queries precisely like A. Let (Y, Z^k) be the length-k transcript prefix of A's execution thus obtained.
2. Exactly like A, algorithm $A^{(k)}$ draws a pair $\{x_{k+1}, y_{k+1}\}$ from $\mathrm{T}_{(Y,Z^k)}$. However, instead of calling $\mathsf{PAIRCOND}_{\mathbf{p}}(\{x_{k+1}, y_{k+1}\})$ to obtain p_{k+1}, algorithm $A^{(k)}$ generates p_{k+1} in the following manner:

   (i) If x_{k+1} and y_{k+1} both belong to the same bucket B_ℓ then p_{k+1} is chosen uniformly from $\{x_{k+1}, y_{k+1}\}$.

   (ii) If one of $\{x_{k+1}, y_{k+1}\}$ belongs to B_ℓ and the other belongs to $B_{\ell'}$ for some $\ell < \ell'$, then p_{k+1} is set to be the element of $\{x_{k+1}, y_{k+1}\}$ that belongs to B_ℓ.

   Let (Y, Z^{k+1}) be the length-$(k+1)$ transcript prefix obtained by appending $(\{x_{k+1}, y_{k+1}\}, p_{k+1})$ to Z^k. Algorithm $A^{(k)}$ continues in this way for a total of $q-k$ stages; i.e. it next draws $\{x_{k+2}, y_{k+2}\}$ from $\mathrm{T}_{(Y,Z^{k+1})}$ and generates p_{k+2} as described above; then (Y, Z^{k+2}) is the length-$(k+2)$ transcript prefix obtained by appending $(\{x_{k+2}, y_{k+2}\}, p_{k+2})$ to Z^{k+1}; and so on. At the end of the process a transcript (Y, Z^q) has been constructed.

Let $\P^{*,(k)}$ denote the distribution over final transcripts (Y, Z^q) that are obtained by running $A^{(k)}$ on a $\mathsf{PAIRCOND}_{\mathbf{p}^*}$ oracle. Let $\P^{\mathsf{no},(k)}$ denote the distribution over final transcripts (Y, Z^q) that are obtained by (i) first drawing $\mathbf{p}$ from $\mathcal{D}_{\mathsf{no}}$, and then (ii) running $A^{(k)}$ on a $\mathsf{PAIRCOND}_{\mathbf{p}}$ oracle. Note that $\P^{*,(q)}$ is identical to $\P^*$ and $\P^{\mathsf{no},(q)}$ is identical to $\P^{\mathsf{no}}$ (since algorithm $A^{(q)}$, which does not fake any queries, is identical to algorithm A).

Recall that our goal is to prove Eq. (4.31). Since $\P^{*,(q)} = \P^{*}$ and $\P^{\mathsf{no},(q)} = \P^{\mathsf{no}}$, Eq. (4.31) is an immediate consequence (using the triangle inequality for total variation distance) of the following two lemmas, which we prove below:

Lemma 4.1.22. $\mathrm{d_{TV}}(\P^{*,(0)}, \P^{no,(0)}) \leq 1/10$.

Lemma 4.1.23. *For all* $0 \leq k < q$*, we have* $\mathrm{d_{TV}}(\P^{*,(k)}, \P^{*,(k+1)}) \leq 1/(20q)$ *and* $\mathrm{d_{TV}}(\P^{no,(k)}, \P^{no,(k+1)}) \leq 1/(20q)$.

Proof of Lemma 4.1.22: Define $\P^{*}_0$ to be the distribution over outcomes of the q calls to $\mathsf{SAMP}_{\mathbf{p}}$ (i.e. over length-0 transcript prefixes) when $\mathbf{p} = \mathbf{p}^*$. Define $\P^{\mathsf{no}}$ to be the distribution over outcomes of the q calls to $\mathsf{SAMP}_{\mathbf{p}}$ when $\mathbf{p}$ is drawn from $\mathcal{D}_{\mathsf{no}}$. We begin by noting that by the data processing inequality for total variation distance (Fact 1.4.2), we have $\mathrm{d_{TV}}(\P^{*,(0)}, \P^{\mathsf{no},(0)}) \leq \mathrm{d_{TV}}(\P^{*}_0, \P^{\mathsf{no}}_0)$ (indeed, after the calls to respectively $\mathsf{SAMP}_{\mathbf{p}}$ and $\mathsf{SAMP}_{\mathbf{p}^*}$, the same randomized function F – which fakes all remaining oracle calls – is applied to the two resulting distributions over length-0 transcript prefixes $\P^{*}_0$ and $\P^{\mathsf{no}}_0$). In the rest of the proof we show that $\mathrm{d_{TV}}(\P^{*}_0, \P^{\mathsf{no}}_0) \leq 1/10$.

Let E denote the event that the q calls to $\mathsf{SAMP}_{\mathbf{p}}$ yield points $s_1, \ldots, s_q$ such that no bucket-pair U_i contains more than one of these points. Since $\mathbf{p}^*(U_i) = 1/r$ for all i,

$$\P^{*}_0(E) = \prod_{j=0}^{q-1} \left(1 - \frac{j}{r}\right) \geq 9/10\ , \tag{4.32}$$

where Eq. (4.32) follows from a standard birthday paradox analysis and the fact that $q \leq \frac{1}{3}\sqrt{r}$. Since for each possible outcome of $\mathbf{p}$ drawn from $\mathcal{D}_{\mathsf{no}}$ we have $\mathbf{p}(U_i) = 1/r$ for all i, we further have that also

$$\P^{\mathsf{no}}_0(E) = \prod_{j=0}^{q-1} \left(1 - \frac{j}{r}\right). \tag{4.33}$$

We moreover claim that the two conditional distributions $(\P^{*}_0|E)$ and $(\P^{\mathsf{no}}_0|E)$ are identical, i.e.

$$(\P^{*}_0|E) = (\P^{\mathsf{no}}_0|E). \tag{4.34}$$

To see this, fix any sequence $(\ell_1, \ldots, \ell_q) \in [r]^q$ such that $\ell_i \neq \ell_j$ for all $i \neq j$. Let $(s_1, \ldots, s_q) \in [n]^q$ denote a draw from $(\P^{*}_0|E)$. The probability that $(s_i \in U_{\ell_i}$ for all $1 \leq i \leq q)$ is precisely $(r-q)!/r!$. Now given that $s_i \in U_{\ell_i}$ for all i, it is clear that s_i is equally likely to lie in $B_{2\ell_i - 1}$ and in $B_{2\ell_i}$, and given that it lies in a particular one of the two buckets, it is equally likely to be any element in that bucket. This is true independently for all $1 \leq i \leq q$.

Now let $(s_1, \ldots, s_q) \in [n]^q$ denote a draw from $(\P^{\mathsf{no}}_0|E)$. Since each distribution $\mathbf{p}$ in the support of $\mathcal{D}_{\mathsf{no}}$ has $\mathbf{p}(U_i) = 1/r$ for all i, we likewise have that the probability that $(s_i \in U_{\ell_i}$ for all $1 \leq i \leq q)$ is precisely $(r-q)!/r!$. Now given that $s_i \in U_{\ell_i}$ for all i, we have that s_i is equally likely to lie in $B_{2\ell_i - 1}$

and in $B_{2\ell_i}$; this is because π_i (recall that π determines $\mathbf{p} = \mathbf{p}_\pi$) is equally likely to be $\uparrow\downarrow$ (in which case $\mathbf{p}(B_{2\ell_i-1}) = 3/(4r)$ and $\mathbf{p}(B_{2\ell_i}) = 1/(4r)$) as it is to be $\downarrow\uparrow$ (in which case $\mathbf{p}(B_{2\ell_i-1}) = 1/(4r)$ and $\mathbf{p}(B_{2\ell_i}) = 3/(4r)$). Additionally, given that s_i lies in a particular one of the two buckets, it is equally likely to be any element in that bucket. This is true independently for all $1 \le i \le q$ (because conditioning on E ensures that no two elements of $s_1, \ldots, s_q$ lie in the same bucket-pair, so there is "fresh randomness for each i"), and so indeed the two conditional distributions $(\P_0^*|E)$ and $(\P_0^{\mathsf{no}}|E)$ are identical.

Finally, the claimed bound $\mathrm{d_{TV}}(\P_0^*, \P_0^{\mathsf{no}}) \le 1/10$ follows directly from Eqs. (4.32) to (4.34). □

Proof of Lemma 4.1.23: Consider first the claim that $\mathrm{d_{TV}}(\P^{*,(k)}, \P^{*,(k+1)}) \le 1/(20q)$. Fix any $0 \le k < q$. The data processing inequality for total variation distance implies that $\mathrm{d_{TV}}(\P^{*,(k)}, \P^{*,(k+1)})$ is at most the variation distance between random variables X and X', where

- X is the random variable obtained by running A on $\mathsf{COND}_{\mathbf{p}^*}$ to obtain a length-k transcript prefix (Y, Z^k), then drawing $\{x_{k+1}, y_{k+1}\}$ from $\mathrm{T}_{(Y,Z^k)}$, then setting p_{k+1} to be the output of $\mathsf{PAIRCOND}_{\mathbf{p}^*}(\{x_{k+1}, y_{k+1}\})$; and
- X' is the random variable obtained by running A on $\mathsf{COND}_{\mathbf{p}^*}$ to obtain a length-k transcript prefix (Y, Z^k), then drawing $\{x_{k+1}, y_{k+1}\}$ from $\mathrm{T}_{(Y,Z^k)}$, then setting p_{k+1} according to the aforementioned rules 2(i) and 2(ii).

Consider any fixed outcome of (Y, Z^k) and $\{x_{k+1}, y_{k+1}\}$. If rule 2(i) is applied (x_{k+1} and y_{k+1} are in the same bucket), then there is zero contribution to the variation distance between X and X', because choosing a uniform element of $\{x_{k+1}, y_{k+1}\}$ is a perfect simulation of $\mathsf{PAIRCOND}_{\mathbf{p}}(\{x_{k+1}, y_{k+1}\})$. If rule 2(ii) is applied, then the contribution is upper bounded by $O(1/K) < 1/20q$, because $\mathsf{PAIRCOND}_{\mathbf{p}^*}(\{x_{k+1}y_{k+1}\})$ would return a different outcome from rule 2(ii) with probability $1/\Theta(K^{\ell'-\ell}) = O(1/K)$. Averaging over all possible outcomes of (Y, Z^k) and $\{x_{k+1}, y_{k+1}\}$ we get that the variation distance between X and X' is at most $1/20q$ as claimed.

An identical argument shows that similarly $\mathrm{d_{TV}}(\P^{\mathsf{no},(k)}, \P^{\mathsf{no},(k+1)}) \le 1/(20q)$. The key observation is that for any distribution $\mathbf{p}$ in the support of $\mathcal{D}_{\mathsf{no}}$, as with $\mathbf{p}^*$ it is the case that points in the same bucket have equal probability under $\mathbf{p}$ and for a pair of points $\{x, y\}$ such that $x \in B_\ell$ and $y \in B_{\ell'}$ for $\ell' > \ell$, the probability that a call to $\mathsf{PAIRCOND}_{\mathbf{p}}(\{x, y\})$ returns y is only $1/\Theta(K^{\ell'-\ell})$. This concludes the proof of Lemma 4.1.23 and of Theorem 4.1.20. □

4.1.4.3 A $\mathrm{poly}(1/\varepsilon)$-query $\mathsf{COND}_{\mathbf{p}}$ algorithm

In this subsection we present an algorithm COND-TEST-KNOWN and prove the following theorem:

Theorem 4.1.24. COND-TEST-KNOWN *is a $\tilde{O}(1/\varepsilon^4)$-query $\mathsf{COND}_{\mathbf{p}}$ testing algorithm for testing equivalence to a known distribution $\mathbf{p}^*$. That is, for every pair of distributions $\mathbf{p}, \mathbf{p}^*$ over $[n]$ (such that $\mathbf{p}^*$ is fully specified and there is* COND *query access to $\mathbf{p}$), the algorithm outputs* accept *with probability at least $2/3$ if $\mathbf{p} = \mathbf{p}^*$ and outputs* reject *with probability at least $2/3$ if $\mathrm{d_{TV}}(\mathbf{p}, \mathbf{p}^*) \ge \varepsilon$.*

This constant-query testing algorithm stands in interesting contrast to the $(\log n)^{\Omega(1)}$-query lower bound for $\mathsf{PAIRCOND}_{\mathbf{p}}$ algorithms for this problem.

High-level overview of the algorithm and its analysis: First, we note that by reordering elements of $[n]$ we may assume without loss of generality that $\mathbf{p}^*(1) \leq \cdots \leq \mathbf{p}^*(n)$; this will be convenient for us.

Our $(\log n)^{\Omega(1)}$ query lower bound for $\mathsf{PAIRCOND}_{\mathbf{p}}$ algorithms exploited the intuition that comparing two points using the $\mathsf{PAIRCOND}_{\mathbf{p}}$ oracle might not provide much information (e.g. if one of the two points was a priori "known" to be much heavier than the other). In contrast, with a general $\mathsf{COND}_{\mathbf{p}}$ oracle at our disposal, we can compare a given point $j \in [n]$ with *any subset* of $[n] \setminus \{j\}$. Thus the following definition will be useful:

Definition 4.1.25 (comparable points)**.** Fix $0 < \lambda \leq 1$. A point $j \in \mathrm{supp}(D^*)$ is said to be λ-*comparable* if there exists a set $S \subseteq ([n] \setminus \{j\})$ such that

$$\mathbf{p}^*(j) \in [\lambda \mathbf{p}^*(S), \mathbf{p}^*(S)/\lambda].$$

Such a set S is then said to be a λ-*comparable-witness for* j (according to $\mathbf{p}^*$), which is denoted $S \cong^* j$. We say that a set $T \subseteq [n]$ is λ-comparable if every $i \in T$ is λ-comparable.

We stress that the notion of being λ-comparable deals only with the known distribution $\mathbf{p}^*$; this will be important later.

Fix $\varepsilon_1 = \Theta(\varepsilon)$ (we specify ε_1 precisely in Eq. (4.37) below). Our analysis and algorithm consider two possible cases for the distribution $\mathbf{p}^*$ (where it is not hard to verify, and we provide an explanation subsequently, that one of the two cases must hold):

1. The first case is that for some $i^* \in [n]$ we have

$$\mathbf{p}^*(\{1, \ldots, i^*\}) > 2\varepsilon_1 \quad \text{but} \quad \mathbf{p}^*(\{1, \ldots, i^* - 1\}) \leq \varepsilon_1. \tag{4.35}$$

   In this case $1 - \varepsilon_1$ of the total probability mass of $\mathbf{p}^*$ must lie on a set of at most $1/\varepsilon_1$ elements, and in such a situation it is easy to efficiently test whether $\mathbf{p} = \mathbf{p}^*$ using $\mathrm{poly}(1/\varepsilon)$ queries (see Algorithm $\mathsf{COND}_{\mathbf{p}}$-TEST-KNOWN-HEAVY and Lemma 4.1.29).

2. The second case is that there exists an element $k^* \in [n]$ such that

$$\varepsilon_1 < \mathbf{p}^*(\{1, \ldots, k^*\}) \leq 2\varepsilon_1 < D^*(\{1, \ldots, k^* + 1\}). \tag{4.36}$$

   This is the more challenging (and typical) case. In this case, it can be shown that every element $j > k^*$ has at least one ε_1-comparable-witness within $\{1, \ldots, j\}$. In fact, we show (see Claim 4.1.26) that either (a) $\{1, \ldots, j-1\}$ is an ε_1-comparable witness for j, or (b) the set $\{1, \ldots, j-1\}$ can be partitioned

into disjoint sets[6] $S_1, \ldots, S_t$ such that *each* S_i, $1 \le i \le t$, is a $\frac{1}{2}$-comparable-witness for j. Case (a) is relatively easy to handle so we focus on (b) in our informal description below.

The partition $S_1, \ldots, S_t$ is useful to us for the following reason: Suppose that $\mathrm{d_{TV}}(\mathbf{p}, \mathbf{p}^*) \ge \varepsilon$. It is not difficult to show (see Claim 4.1.28) that unless $\mathbf{p}(\{1, \ldots, k^*\}) > 3\varepsilon_1$ (which can be easily detected and provides evidence that the tester should reject), a random sample of $\Theta(1/\varepsilon)$ draws from $\mathbf{p}$ will with high probability contain a "heavy" point $j > k^*$, that is, a point $j > k^*$ such that $\mathbf{p}(j) \ge (1+\varepsilon_2)\mathbf{p}^*(j)$ (where $\varepsilon_2 = \Theta(\varepsilon)$). Given such a point j, there are two possibilities:

1. The first possibility is that a significant fraction of the sets $S_1, \ldots, S_t$ have $\mathbf{p}(j)/\mathbf{p}(S_i)$ "noticeably different" from $\mathbf{p}^*(j)/\mathbf{p}^*(S_i)$. (Observe that since each set S_i is a $\frac{1}{2}$-comparable witness for j, it is possible to efficiently check whether this is the case.) If this is the case then our tester should reject since this is evidence that $\mathbf{p} \ne \mathbf{p}^*$.
2. The second possibility is that almost every S_i has $\mathbf{p}(j)/\mathbf{p}(S_i)$ very close to $\mathbf{p}^*(j)/\mathbf{p}^*(S_i)$. If this is the case, though, then since $\mathbf{p}(j) \ge (1+\varepsilon_2)\mathbf{p}^*(j)$ and the union of $S_1, \ldots, S_t$ is $\{1, \ldots, j-1\}$, it must be the case that $\mathbf{p}(\{1, \ldots, j\})$ is "significantly larger" than $\mathbf{p}^*(\{1, \ldots, j\})$. This will be revealed by random sampling from $\mathbf{p}$ and thus our testing algorithm can reject in this case as well.

Key quantities and useful claims. We define some quantities that are used in the algorithm and its analysis. Let

$$\varepsilon_1 \stackrel{\text{def}}{=} \frac{\varepsilon}{10}; \quad \varepsilon_2 \stackrel{\text{def}}{=} \frac{\varepsilon}{2}; \quad \varepsilon_3 \stackrel{\text{def}}{=} \frac{\varepsilon}{48}; \quad \varepsilon_4 \stackrel{\text{def}}{=} \frac{\varepsilon}{6}. \tag{4.37}$$

Claim 4.1.26. *Suppose there exists an element $k^* \in [n]$ that satisfies Eq. (4.36). Fix any $j > k^*$. Then*

1. *If $\mathbf{p}^*(j) \ge \varepsilon_1$, then $S_1 \stackrel{\text{def}}{=} \{1, \ldots, j-1\}$ is an ε_1-comparable witness for j;*
2. *If $\mathbf{p}^*(j) < \varepsilon_1$ then the set $\{1, \ldots, j-1\}$ can be partitioned into disjoint sets $S_1, \ldots, S_t$ such that each S_i, $1 \le i \le t$, is a $\frac{1}{2}$-comparable-witness for j.*

Proof. First consider the case that $\mathbf{p}^*(j) \ge \varepsilon_1$. In this case $S_1 = \{1, \ldots, j-1\}$ is an ε_1-comparable witness for j because $\mathbf{p}^*(j) \ge \varepsilon_1 \ge \varepsilon_1 \mathbf{p}^*(\{1, \ldots, j-1\})$ and $\mathbf{p}^*(j) \le 1 \le \frac{1}{\varepsilon_1}\mathbf{p}^*(\{1, \ldots, k^*\}) \le \frac{1}{\varepsilon_1}\mathbf{p}^*(\{1, \ldots, j-1\})$, where the last inequality holds since $k^* \le j-1$.

Next, consider the case that $\mathbf{p}^*(j) < \varepsilon_1$. In this case we build our intervals iteratively from right to left, as follows. Let $j_1 = j-1$ and let j_2 be the minimum index in $[\![j_1]\!]$ such that

$$\mathbf{p}^*(\{j_2+1, \ldots, j_1\}) \le \mathbf{p}^*(j).$$

(Observe that we must have $j_2 \ge 1$, because $\mathbf{p}^*(\{1, \ldots, k^*\}) > \varepsilon_1 > \mathbf{p}^*(j)$.) Since $\mathbf{p}^*(\{j_2, \ldots, j_1\}) > \mathbf{p}^*(j)$

[6] In fact the sets are intervals (under the assumption $\mathbf{p}^*(1) \le \cdots \le \mathbf{p}^*(n)$), but that is not really important for our arguments.

and the function $\mathbf{p}^*(\cdot)$ is monotonically increasing, it must be the case that

$$\frac{1}{2}\mathbf{p}^*(j) \le \mathbf{p}^*(\{j_2+1,\dots,j_1\}) \le \mathbf{p}^*(j).$$

Thus the interval $S_1 \stackrel{\text{def}}{=} \{j_2+1,\dots,j_1\}$ is a $\frac{1}{2}$-comparable witness for j as desired.

We continue in this fashion from right to left; i.e. if we have defined $j_2,\dots,j_t$ as above and there is an index $j' \in [\![j_t]\!]$ such that $\mathbf{p}^*(\{j'+1,\dots,j_t\}) > \mathbf{p}^*(j)$, then we define j_{t+1} to be the minimum index in $[\![j_t]\!]$ such that

$$\mathbf{p}^*(\{j_{t+1}+1,\dots,j_t\}) \le \mathbf{p}^*(j),$$

and we define S_t to be the interval $\{j_{t+1}+1,\dots,j_t\}$. The argument of the previous paragraph tells us that

$$\frac{1}{2}\mathbf{p}^*(j) \le \mathbf{p}^*(\{j_{t+1}+1,\dots,j_t\}) \le \mathbf{p}^*(j) \tag{4.38}$$

and hence S_t is an $\frac{1}{2}$-comparable witness for j.

At some point, after intervals $S_1 = \{j_2+1,\dots,j_1\},\dots,S_t = \{j_{t+1}+1,\dots,j_t\}$ have been defined in this way, it will be the case that there is no index $j' \in [\![j_{t+1}]\!]$ such that $\mathbf{p}^*(\{j'+1,\dots,j_{t+1}\}) > \mathbf{p}^*(j)$. At this point there are two possibilities: first, if $j_{t+1}+1 = 1$, then $S_1,\dots,S_t$ give the desired partition of $\{1,\dots,j-1\}$. If $j_{t+1}+1 > 1$ then it must be the case that $\mathbf{p}^*(\{1,\dots,j_{t+1}\}) \le \mathbf{p}^*(j)$. In this case we simply add the elements $\{1,\dots,j_{t+1}\}$ to S_t, i.e. we redefine S_t to be $\{1,\dots,j_t\}$. By Eq. (4.38) we have that

$$\frac{1}{2}\mathbf{p}^*(j) \le \mathbf{p}^*(S_t) \le 2\mathbf{p}^*(j)$$

and thus S_t is an $\frac{1}{2}$-comparable witness for j as desired. This concludes the proof. □

Definition 4.1.27 (Heavy points)**.** A point $j \in \mathrm{supp}(D^*)$ is said to be *η-heavy* if $\mathbf{p}(j) \ge (1+\eta)D^*(j)$.

Claim 4.1.28. *Suppose that $\mathrm{d_{TV}}(\mathbf{p}, D^*) \ge \varepsilon$ and Eq. (4.36) holds. Suppose moreover that $\mathbf{p}(\{1,\dots,k^*\}) \le 4\varepsilon_1$. Let $i_1,\dots,i_\ell$ be i.i.d. points drawn from $\mathbf{p}$. Then for $\ell = \Theta(1/\varepsilon)$, with probability at least $99/100$ (over the i.i.d. draws of $i_1,\dots,i_\ell \sim \mathbf{p}$) there is some point $i_j \in \{i_1,\dots,i_\ell\}$ such that $i_j > k^*$ and i_j is ε_2-heavy.*

Proof. Define H_1 to be the set of all ε_2-heavy points and H_2 to be the set of all "slightly lighter" points as follows:

$$H_1 = \{\, i \in [n] : \mathbf{p}(i) \ge (1+\varepsilon_2)D^*(i) \,\}$$
$$H_2 = \{\, i \in [n] : (1+\varepsilon_2)D^*(i) > \mathbf{p}(i) \ge \mathbf{p}^*(i) \,\}$$

By definition of the total variation distance, we have

$$\begin{aligned}\varepsilon \le \mathrm{d}_{\mathrm{TV}}(\mathbf{p}, D^*) &= \sum_{i:\mathbf{p}(i)\ge\mathbf{p}^*(i)} (\mathbf{p}(i) - \mathbf{p}^*(i)) = (\mathbf{p}(H_1) - \mathbf{p}^*(H_1)) + (\mathbf{p}(H_2) - \mathbf{p}^*(H_2)) \\ &\le \mathbf{p}(H_1) + ((1+\varepsilon_2)D^*(H_2) - \mathbf{p}^*(H_2)) \\ &= \mathbf{p}(H_1) + \varepsilon_2 \mathbf{p}^*(H_2) < \mathbf{p}(H_1) + \varepsilon_2 = \mathbf{p}(H_1) + \frac{\varepsilon}{2}.\end{aligned}$$

So it must be the case that $\mathbf{p}(H_1) \ge \varepsilon/2 = 5\varepsilon_1$. Since by assumption we have $\mathbf{p}(\{1,\ldots,k^*\}) \le 4\varepsilon_1$, it must be the case that $\mathbf{p}(H_1 \setminus \{1,\ldots,k^*\}) \ge \varepsilon_1$. The claim follows from the definition of H_1 and the size, ℓ, of the sample. □

Algorithm 21 $\mathsf{COND}_{\mathbf{p}}$-TEST-KNOWN

Require: error parameter $\varepsilon > 0$; query access to $\mathsf{COND}_{\mathbf{p}}$ oracle; explicit description $(\mathbf{p}^*(1),\ldots,\mathbf{p}^*(n))$ of distribution $\mathbf{p}^*$ satisfying $\mathbf{p}^*(1) \le \cdots \le \mathbf{p}^*(n)$

1: Let i^* be the minimum index $i \in [n]$ such that $\mathbf{p}^*(\{1,\ldots,i\}) > 2\varepsilon_1$.
2: **if** $\mathbf{p}^*(\{1,\ldots,i^*-1\}) \le \varepsilon_1$ **then**
3: Call algorithm $\mathsf{COND}_{\mathbf{p}}$-Test-Known-Heavy$(\varepsilon, \mathsf{COND}_{\mathbf{p}}, \mathbf{p}^*, i^*)$ (and exit)
4: **else**
5: Call algorithm $\mathsf{COND}_{\mathbf{p}}$-Test-Known-Main$(\varepsilon, \mathsf{COND}_{\mathbf{p}}, \mathbf{p}^*, i^*-1)$ (and exit).
6: **end if**

Algorithm 22 $\mathsf{COND}_{\mathbf{p}}$-TEST-KNOWN-HEAVY

Require: error parameter $\varepsilon > 0$; query access to $\mathsf{COND}_{\mathbf{p}}$ oracle; explicit description $(\mathbf{p}^*(1),\ldots,\mathbf{p}^*(n))$ of distribution $\mathbf{p}^*$ satisfying $\mathbf{p}^*(1) \le \cdots \le \mathbf{p}^*(n)$; value $i^* \in [n]$ satisfying $\mathbf{p}^*(\{1,\ldots,i^*-1\}) \le \varepsilon_1$, $\mathbf{p}^*(\{1,\ldots,i^*\}) > 2\varepsilon_1$

1: Call the $\mathsf{SAMP}_{\mathbf{p}}$ oracle $m = \Theta((\log(1/\varepsilon))/\varepsilon^4)$ times. For each $i \in [i^*, n]$ let $\widehat{\mathbf{p}}(i)$ be the fraction of the m calls to $\mathsf{SAMP}_{\mathbf{p}}$ that returned i. Let $\widehat{\mathbf{p}}' = 1 - \sum_{i\in[i^*,n]} \widehat{\mathbf{p}}(i)$ be the fraction of the m calls that returned values in $\{1,\ldots,i^*-1\}$.
2: **if** either (any $i \in [i^*, n]$ has $|\widehat{\mathbf{p}}(i) - \mathbf{p}^*(i)| > {\varepsilon_1}^2$) or $(\widehat{\mathbf{p}}' - \mathbf{p}^*(\{1,\ldots,i^*-1\}) > \varepsilon_1)$ **then**
3: **return** reject (and exit)
4: **end if**
5: **return** accept

Proof of Theorem 4.1.24 It is straightforward to verify that the query complexity of $\mathsf{COND}_{\mathbf{p}}$-Test-Known-Heavy is $\tilde{O}(1/\varepsilon^4)$ and the query complexity of $\mathsf{COND}_{\mathbf{p}}$-Test-Known-Main is also $\tilde{O}(1/\varepsilon^4)$, so the overall query complexity of COND-TEST-KNOWN is as claimed.

By the definition of i^* (in the first line of the algorithm), either Eq. (4.35) holds for this setting of i^*, or Eq. (4.36) holds for $k^* = i^* - 1$. To prove correctness of the algorithm, we first deal with the simpler case, which is that Eq. (4.35) holds:

Lemma 4.1.29. *Suppose that $\mathbf{p}^*$ is such that $\mathbf{p}^*(\{1,\ldots,i^*\}) > 2\varepsilon_1$ but $\mathbf{p}^*(\{1,\ldots,i^*-1\}) \le \varepsilon_1$. Then $\mathsf{COND}_{\mathbf{p}}$-TEST-KNOWN-HEAVY$(\varepsilon, \mathsf{COND}_D, \mathbf{p}^*, i^*)$ returns accept with probability at least $2/3$ if $\mathbf{p} = \mathbf{p}^*$ and returns reject with probability at least $2/3$ if $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{p}^*) \ge \varepsilon$.*

Algorithm 23 $\mathsf{COND}_{\mathbf{p}}$-TEST-KNOWN-MAIN

Require: error parameter $\varepsilon > 0$; query access to $\mathsf{COND}_{\mathbf{p}}$ oracle; explicit description $(\mathbf{p}^*(1), \ldots, \mathbf{p}^*(n))$ of distribution $\mathbf{p}^*$ satisfying $\mathbf{p}^*(1) \leq \cdots \leq \mathbf{p}^*(n)$; value $k^* \in [n]$ satisfying $\varepsilon_1 < \mathbf{p}^*(\{1, \ldots, k^*\}) \leq 2\varepsilon_1 < \mathbf{p}^*(\{1, \ldots, k^*+1\})$

1: Call the $\mathsf{SAMP}_{\mathbf{p}}$ oracle $\Theta(1/\varepsilon^2)$ times and let $\widehat{\mathbf{p}}(\{1, \ldots, k^*\})$ denote the fraction of responses that lie in $\{1, \ldots, k^*\}$. If $\widehat{\mathbf{p}}(\{1, \ldots, k^*\}) \notin [\frac{\varepsilon_1}{2}, \frac{5\varepsilon_1}{2}]$ then **return** reject (and exit).
2: Call the $\mathsf{SAMP}_{\mathbf{p}}$ oracle $\ell = \Theta(1/\varepsilon)$ times to obtain points $i_1, \ldots, i_\ell$.
3: **for** all $j \in \{1, \ldots, \ell\}$ such that $i_j > k^*$ **do**
4: Call the $\mathsf{SAMP}_{\mathbf{p}}$ oracle $m = \Theta(\log(1/\varepsilon)/\varepsilon^2)$ times and let $\widehat{\mathbf{p}}(\{1, \ldots, i_j\})$ be the fraction of responses that lie in $\{1, \ldots, i_j\}$. If $\widehat{\mathbf{p}}(\{1, \ldots, i_j\}) \notin [1-\varepsilon_3, 1+\varepsilon_3]\mathbf{p}^*(\{1, \ldots, i_j\})$ then **return** reject (and exit).
5: **if** $\mathbf{p}^*(i_j) \geq \varepsilon_1$ **then**
6: Run $\textsc{Compare}(\{i_j\}, \{1, \ldots, i_j - 1\}, \frac{\varepsilon_2}{16}, \frac{2}{\varepsilon_1}, \frac{1}{10\ell})$ and let v denote its output. If $v \notin [1 - \frac{\varepsilon_2}{8}, 1 + \frac{\varepsilon_2}{8}]\frac{\mathbf{p}^*(\{1,\ldots,i_j-1\})}{\mathbf{p}^*(\{i_j\})}$ then **return** reject (and exit).
7: **else**
8: Let $S_1, \ldots, S_t$ be the partition of $\{1, \ldots, i_j - 1\}$ such that each S_i is an ε_1-comparable witness for i_j, which is provided by Claim 4.1.26.
9: Select a list of $h = \Theta(1/\varepsilon)$ elements $S_{a_1}, \ldots, S_{a_h}$ independently and uniformly from $\{S_1, \ldots, S_j\}$.
10: For each S_{a_r}, $1 \leq r \leq h$, run $\textsc{Compare}(\{i_j\}, S_{a_r}, \frac{\varepsilon_4}{8}, 4, \frac{1}{10\ell h})$ and let v denote its output. If $v \notin [1 - \frac{\varepsilon_4}{4}, 1 + \frac{\varepsilon_4}{4}]\frac{\mathbf{p}^*(S_{a_r})}{\mathbf{p}^*(\{i_j\})}$ then **return** reject (and exit).
11: **end if**
12: **end for**
13: **return** accept.

Proof. The conditions of Lemma 4.1.29, together with the fact that $\mathbf{p}^*(\cdot)$ is monotone non-decreasing, imply that each $i \geq i^*$ has $\mathbf{p}^*(i) \geq \varepsilon_1$. Thus there can be at most $1/\varepsilon_1$ many values $i \in \{i^*, \ldots, n\}$, i.e. it must be the case that $i^* \geq n - 1/\varepsilon_1 + 1$. Since the expected value of $\widehat{\mathbf{p}}(i)$ (defined in Line 1 of $\mathsf{COND}_{\mathbf{p}}$-TEST-KNOWN-HEAVY) is precisely $\mathbf{p}(i)$, for any fixed value of $i \in \{i^*, \ldots, n\}$ an additive Chernoff bound implies that $|\mathbf{p}(i) - \widehat{\mathbf{p}}(i)| \leq (\varepsilon_1)^2$ with failure probability at most $\frac{1}{10\left(1+\frac{1}{\varepsilon_1}\right)}$. Similarly $|\widehat{\mathbf{p}}' - \mathbf{p}(\{1, \ldots, i^*-1\})| \leq \varepsilon_1$ with failure probability at most $\frac{1}{10\left(1+\frac{1}{\varepsilon_1}\right)}$. A union bound over all failure events gives that with probability at least $9/10$ each value $i \in \{i^*, \ldots, n\}$ has $|\mathbf{p}(i) - \widehat{\mathbf{p}}(i)| \leq {\varepsilon_1}^2$ and additionally $|\widehat{\mathbf{p}}' - \mathbf{p}(\{1, \ldots, i^*-1\})| \leq \varepsilon_1$; we refer to this compound event as (*).

If $\mathbf{p}^* = \mathbf{p}$, by (*) the algorithm outputs accept with probability at least $9/10$.

Now suppose that $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{p}^*) \geq \varepsilon$. With probability at least 9/10 we have (*) so we suppose that indeed (*) holds. In this case we have

$$\begin{aligned}
\varepsilon \leq \mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{p}^*) &= \sum_{i<i^*} |\mathbf{p}(i) - \mathbf{p}^*(i)| + \sum_{i \geq i^*} |\mathbf{p}(i) - \mathbf{p}^*(i)| \\
&\leq \sum_{i<i^*} (\mathbf{p}(i) + \mathbf{p}^*(i)) + \sum_{i \geq i^*} |\mathbf{p}(i) - \mathbf{p}^*(i)| \\
&\leq \mathbf{p}(\{1, \ldots, i^*-1\}) + \varepsilon_1 + \sum_{i \geq i^*} \left(|\widehat{\mathbf{p}}(i) - \mathbf{p}^*(i)| + {\varepsilon_1}^2\right) \\
&\leq \widehat{\mathbf{p}}' + \varepsilon_1 + 2\varepsilon_1 + \sum_{i \geq i^*} (|\widehat{\mathbf{p}}(i) - \mathbf{p}^*(i)|)
\end{aligned}$$

where the first inequality is by the triangle inequality, the second is by (*) and the fact that $\mathbf{p}^*(\{1,\dots,i^*-1\}) \le \varepsilon_1$, and the third inequality is by (*) and the fact that there are at most $1/\varepsilon_1$ elements in $\{i^*,\dots,n\}$. Since $\varepsilon_1 = \varepsilon/10$, the above inequality implies that

$$\frac{7}{10}\varepsilon \le \widehat{\mathbf{p}}' + \sum_{i \ge i^*} \left(|\widehat{\mathbf{p}}(i) - \mathbf{p}^*(i)|\right).$$

If any $i \in \{i^*,\dots,n\}$ has $|\widehat{\mathbf{p}}(i) - \mathbf{p}^*(i)| > (\varepsilon_1)^2$ then the algorithm outputs reject so we may assume that $|\widehat{\mathbf{p}}(i) - \mathbf{p}^*(i)| \le {\varepsilon_1}^2$ for all i. This implies that

$$6\varepsilon_1 = \frac{6}{10}\varepsilon \le \widehat{\mathbf{p}}'$$

but since $\mathbf{p}^*(\{1,\dots,i^*-1\}) \le \varepsilon_1$ the algorithm must reject. □

Now we turn to the more difficult (and typical) case, that Eq. (4.36) holds (for $k^* = i^* - 1$), i.e.

$$\varepsilon_1 < \mathbf{p}^*(\{1,\dots,k^*\}) \le 2\varepsilon_1 < \mathbf{p}^*(\{1,\dots,k^*+1\}).$$

With the claims we have already established it is straightforward to argue completeness:

Lemma 4.1.30. *Suppose that $\mathbf{p} = \mathbf{p}^*$ and Eq. (4.36) holds. Then with probability at least $2/3$ algorithm* $\mathsf{COND}_{\mathbf{p}}$-TEST-KNOWN-MAIN *outputs accept.*

Proof. We first observe that the expected value of the quantity $\widehat{\mathbf{p}}(\{1,\dots,k^*\})$ defined in Line 1 is precisely $\mathbf{p}(\{1,\dots,k^*\}) = \mathbf{p}^*(\{1,\dots,k^*\})$ and hence lies in $[\varepsilon_1, 2\varepsilon_1]$ by Eq. (4.36). The additive Chernoff bound implies that the probability the algorithm outputs reject in Line 1 is at most $1/10$. Thus we may assume the algorithm continues to Line 2.

In any given execution of Line 4, since the expected value of $\widehat{\mathbf{p}}(\{1,\dots,i_j\})$ is precisely $\mathbf{p}(\{1,\dots,i_j\}) = \mathbf{p}^*(\{1,\dots,i_j\}) > \varepsilon_1$, a multiplicative Chernoff bound gives that the algorithm outputs reject with probability at most $1/(10\ell)$. Thus the probability that the algorithm outputs reject in any execution of Line 4 is at most $1/10$. We henceforth assume that the algorithm never outputs reject in this step.

Fix a setting of $j \in \{1,\dots,\ell\}$ such that $i_j > k^*$. Consider first the case that $\mathbf{p}^*(i_j) \ge \varepsilon_1$ so the algorithm enters Line 6. By item (1) of Claim 4.1.26 and item (1) of Lemma 4.1.2, we have that with probability at least $1 - \frac{1}{10\ell}$ COMPARE outputs a value v in the range $[1 - \frac{\varepsilon_2}{16}, 1 + \frac{\varepsilon_2}{16}]\frac{\mathbf{p}^*(\{1,\dots,i_j-1\})}{\mathbf{p}^*(\{i_j\})}$ (recall that $\mathbf{p} = \mathbf{p}^*$), so the algorithm does not output reject in Line 6. Now suppose that $\mathbf{p}^*(i_j) < \varepsilon_1$ so the algorithm enters Line 8. Fix a value $1 \le r \le h$ in Line 10. By Claim 4.1.26 we have that S_{a_r} is a $\frac{1}{2}$-comparable witness for i_j. By item (1) of Lemma 4.1.2, we have that with probability at least $1 - \frac{1}{10\ell h}$ COMPARE outputs a value v in the range $[1 - \frac{\varepsilon_4}{4}, 1 + \frac{\varepsilon_4}{4}]\frac{\mathbf{p}^*(S_{a_r})}{\mathbf{p}^*(\{i_j\})}$ (recall that $\mathbf{p} = \mathbf{p}^*$). A union bound over all h values of r gives that the algorithm outputs reject in Line 10 with probability at most $1/(10\ell)$. So in either case, for this setting of j, the algorithm outputs reject on that iteration of the outer loop with probability at most $1/(10\ell)$. A union bound over all ℓ

iterations of the outer loop gives that the algorithm outputs reject at any execution of Line 6 or Line 10 is at most $1/10$.

Thus the overall probability that the algorithm outputs reject is at most $3/10$, and the lemma is proved. □

Next we argue soundness:

Lemma 4.1.31. *Suppose that* $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{p}^*) \geq \varepsilon$ *and Eq. (4.36) holds. Then with probability at least* $2/3$ *algorithm* $\mathsf{COND}_{\mathbf{p}}$-TEST-KNOWN-MAIN *outputs reject.*

Proof. If $\mathbf{p}(\{1, \ldots, k^*\}) \notin [\varepsilon_1/3, 3\varepsilon_1]$ then a standard additive Chernoff bound implies that the algorithm outputs reject in Line 1 with probability at least $9/10$. Thus we may assume going forward in the argument that $\mathbf{p}(\{1, \ldots, k^*\}) \in [\varepsilon_1/3, 3\varepsilon_1]$. As a result we may apply Claim 4.1.28, and we have that with probability at least $99/100$ there is an element $i_j \in \{i_1, \ldots, i_\ell\}$ such that $i_j > k^*$ and i_j is ε_2-heavy, i.e. $\mathbf{p}(i_j) \geq (1+\varepsilon_2)\mathbf{p}^*(i_j)$. We condition on this event going forward (the rest of our analysis will deal with this specific element i_j).

We now consider two cases:

Case 1: Distribution $\mathbf{p}$ has $\mathbf{p}(\{1, \ldots, i_j\}) \notin [1 - 3\varepsilon_3, 1 + 3\varepsilon_3]\mathbf{p}^*(\{1, \ldots, i_j\})$. Since the quantity $\widehat{\mathbf{p}}(\{1, \ldots, i_j\})$ obtained in Line 4 has expected value $\mathbf{p}(\{1, \ldots, i_j\}) \geq \mathbf{p}(\{1, \ldots, k^*\}) \geq \varepsilon_1/3$, applying the multiplicative Chernoff bound implies that $\widehat{\mathbf{p}}(\{1, \ldots, i_j\}) \in [1 - \varepsilon_3, 1 + \varepsilon_3]\mathbf{p}(\{1, \ldots, i_j\})$ except with failure probability at most $\varepsilon/10 \leq 1/10$. If this failure event does not occur then since $\mathbf{p}(\{1, \ldots, i_j\}) \notin [1 - 3\varepsilon_3, 1 + 3\varepsilon_3]\mathbf{p}^*(\{1, \ldots, i_j\})$ it must hold that $\widehat{\mathbf{p}}(\{1, \ldots, i_j\}) \notin [1 - \varepsilon_3, 1 + \varepsilon_3]\mathbf{p}^*(\{1, \ldots, i_j\})$ and consequently the algorithm outputs reject. Thus in Case 1 the algorithm outputs reject with overall failure probability at least $89/100$.

Case 2: Distribution $\mathbf{p}$ has $\mathbf{p}(\{1, \ldots, i_j\}) \in [1 - 3\varepsilon_3, 1 + 3\varepsilon_3]\mathbf{p}^*(\{1, \ldots, i_j\})$. This case is divided into two sub-cases depending on the value of $\mathbf{p}^*(i_j)$.

Case 2(a): $\mathbf{p}^*(i_j) \geq \varepsilon_1$. In this case the algorithm reaches Line 6. e use the following claim:

Claim 4.1.32. *In Case 2(a), suppose that* $i_j > k^*$ *is such that* $\mathbf{p}(i_j) \geq (1 + \varepsilon_2)\mathbf{p}^*(i_j)$*, and* $\mathbf{p}(\{1, \ldots, i_j\}) \in [1 - 3\varepsilon_3, 1 + 3\varepsilon_3]\mathbf{p}^*(\{1, \ldots, i_j\})$*. Then*

$$\frac{\mathbf{p}(\{1, \ldots, i_j - 1\})}{\mathbf{p}(i_j)} \leq \left(1 - \frac{\varepsilon_2}{4}\right) \cdot \frac{\mathbf{p}^*(\{1, \ldots, i_j - 1\})}{\mathbf{p}^*(i_j)}.$$

Proof. To simplify notation we write

$$a \stackrel{\text{def}}{=} \mathbf{p}(i_j); \quad b \stackrel{\text{def}}{=} \mathbf{p}^*(i_j); \quad c \stackrel{\text{def}}{=} \mathbf{p}(\{1, \ldots, i_j - 1\}); \quad d \stackrel{\text{def}}{=} \mathbf{p}^*(\{1, \ldots, i_j - 1\}).$$

We have that

$$a \geq (1 + \varepsilon_2)b \quad \text{and} \quad a + c \leq (1 + 3\varepsilon_3)(b + d). \tag{4.39}$$

This gives

$$c \leq (1+3\varepsilon_3)(b+d) - (1+\varepsilon_2)b = (1+3\varepsilon_3)d + (3\varepsilon_3 - \varepsilon_2)b \ < (1+3\varepsilon_3)d\,, \tag{4.40}$$

where in the last inequality we used $\varepsilon_2 > 3\varepsilon_3$. Recalling that $a \geq (1+\varepsilon_2)b$ and using $\varepsilon_3 = \varepsilon_2/24$ we get

$$\frac{c}{a} < \frac{(1+3\varepsilon_3)d}{(1+\varepsilon_2)b} = \frac{d}{b}\cdot\frac{1+\varepsilon_2/8}{1+\varepsilon_2} < \frac{d}{b}\cdot\left(1-\frac{\varepsilon_2}{4}\right)\,. \tag{4.41}$$

This proves the claim. □

Applying Claim 4.1.32, we get that in Line 6 we have

$$\frac{\mathbf{p}(\{1,\ldots,i_j-1\})}{\mathbf{p}(i_j)} \leq \left(1-\frac{\varepsilon_2}{4}\right)\cdot\frac{\mathbf{p}^*(\{1,\ldots,i_j-1\})}{\mathbf{p}^*(i_j)}\,. \tag{4.42}$$

Recalling that by the premise of this case $\mathbf{p}^*(i_j) \geq \varepsilon_1$, by applying Claim 4.1.26 we have that $\{1,\ldots,i_j-1\}$ is an ε_1-comparable witness for i_j. Therefore, by Lemma 4.1.2, with probability at least $1-\frac{1}{10\ell}$ the call to $\textsc{Compare}(\{i_j\},\{1,\ldots,i_j-1\},\frac{\varepsilon_2}{16},\frac{2}{\varepsilon_1},\frac{1}{10\ell})$ in Line 6 either outputs an element of $\{\mathsf{high},\mathsf{low}\}$ or outputs a value $v \leq (1-\frac{\varepsilon_2}{4})(1+\frac{\varepsilon_2}{16})\frac{\mathbf{p}^*(\{1,\ldots,i_j-1\})}{\mathbf{p}^*(i_j)} < (1-\frac{\varepsilon_2}{8})\frac{\mathbf{p}^*(\{1,\ldots,i_j-1\})}{\mathbf{p}^*(i_j)}$. In either case the algorithm outputs reject in Line 6, so we are done with Case 2(a).

Case 2(b): $\mathbf{p}^*(i_j) < \varepsilon_1$. In this case the algorithm reaches Line 10, and by item 2 of Claim 4.1.26, we have that $S_1,\ldots,S_t$ is a partition of $\{1,\ldots,i_j-1\}$ and each set $S_1,\ldots,S_t$ is a $\frac{1}{2}$-comparable witness for i_j, i.e.,

$$\text{for all } i \in \{1,\ldots,t\}, \quad \frac{1}{2}\mathbf{p}^*(i_j) \leq \mathbf{p}^*(S_i) \leq 2\mathbf{p}^*(i_j). \tag{4.43}$$

We use the following claim:

Claim 4.1.33. *In Case 2(b) suppose $i_j > k^*$ is such that $\mathbf{p}(i_j) \geq (1+\varepsilon_2)\mathbf{p}^*(i_j)$ and $\mathbf{p}(\{1,\ldots,i_j\}) \in [1-3\varepsilon_3, 1+3\varepsilon_3]\mathbf{p}^*(\{1,\ldots,i_j\})$. Then at least $(\varepsilon_4/8)$-fraction of the sets $S_1,\ldots,S_t$ are such that*

$$\mathbf{p}(S_i) \leq (1+\varepsilon_4)\mathbf{p}^*(S_i).$$

Proof. The proof is by contradiction. Let $\rho = 1-\varepsilon_4/8$ and suppose that there are w sets (without loss of generality we call them $S_1,\ldots,S_w$) that satisfy $\mathbf{p}(S_i) > (1+\varepsilon_4)\mathbf{p}^*(S_i)$, where $\rho' = \frac{w}{t} > \rho$. We first observe that the weight of the w subsets $S_1,\ldots,S_w$ under $\mathbf{p}^*$, as a fraction of $\mathbf{p}^*(\{1,\ldots,i_j-1\})$, is at least

$$\frac{\mathbf{p}^*(S_1\cup\cdots\cup S_w)}{\mathbf{p}^*(S_1\cup\cdots\cup S_w)+(t-w)\cdot 2\mathbf{p}^*(i_j)} \geq \frac{w\frac{\mathbf{p}^*(i_j)}{2}}{w\frac{\mathbf{p}^*(i_j)}{2}+(t-w)\cdot 2\mathbf{p}^*(i_j)} = \frac{w}{4t-3w} = \frac{\rho'}{4-3\rho'},$$

where we used the right inequality in Eq. (4.43) on $S_{w+1},\ldots,S_t$ to obtain the leftmost expression above, and the left inequality in Eq. (4.43) (together with the fact that $\frac{x}{x+c}$ is an increasing function of x for all $c>0$) to

obtain the inequality above. This implies that

$$\begin{aligned}\mathbf{p}(\{1,\dots,i_j-1\}) &= \sum_{i=1}^{w}\mathbf{p}(S_i)+\sum_{i=w+1}^{t}\mathbf{p}(S_i) \geq (1+\varepsilon_4)\sum_{i=1}^{w}\mathbf{p}^*(S_i)+\sum_{i=w+1}^{t}\mathbf{p}(S_i)\\ &\geq (1+\varepsilon_4)\frac{\rho'}{4-3\rho'}\mathbf{p}^*(\{1,\dots,i_j-1\})\\ &\geq (1+\varepsilon_4)\frac{\rho}{4-3\rho}\mathbf{p}^*(\{1,\dots,i_j-1\})\,. \end{aligned} \tag{4.44}$$

From Section 4.1.4.3 we have

$$\begin{aligned}\mathbf{p}(\{1,\dots,i_j\}) &\geq (1+\varepsilon_4)\frac{\rho}{4-3\rho}\mathbf{p}^*(\{1,\dots,i_j-1\})+(1+\varepsilon_2)\mathbf{p}^*(i_j)\\ &\geq \left(1+\frac{3\varepsilon_4}{8}\right)\mathbf{p}^*(\{1,\dots,i_j-1\})+(1+\varepsilon_2)\mathbf{p}^*(i_j)\end{aligned}$$

where for the first inequality above we used $\mathbf{p}(i_j)\geq(1+\varepsilon_2)\mathbf{p}^*(i_j)$ and for the second inequality we used $(1+\varepsilon_4)\frac{\rho}{4-3\rho}\geq 1+\frac{3\varepsilon_4}{8}$. This implies that

$$\mathbf{p}(\{1,\dots,i_j\})>\left(1+\frac{3\varepsilon_4}{8}\right)\mathbf{p}^*(\{1,\dots,i_j-1\})+\left(1+\frac{3\varepsilon_4}{8}\right)\mathbf{p}^*(i_j)=\left(1+\frac{3\varepsilon_4}{8}\right)\mathbf{p}^*(\{1,\dots,i_j\})$$

where the inequality follows from $\varepsilon_2>\frac{3\varepsilon_4}{8}$. Since $\frac{3\varepsilon_4}{8}=3\varepsilon_3$, though, this is a contradiction and the claim is proved. □

Applying Claim 4.1.33, and recalling that $h=\Theta(1/\varepsilon)=\Theta(1/\varepsilon_4)$ sets are chosen randomly in Line 9, we have that with probability at least $9/10$ there is some $r\in\{1,\dots,h\}$ such that $\mathbf{p}(S_{a_r})\leq(1+\varepsilon_4)\mathbf{p}^*(S_{a_r})$. Combining this with $\mathbf{p}(i_j)\geq(1+\varepsilon_2)\mathbf{p}^*(i_j)$, we get that

$$\frac{\mathbf{p}(S_{a_r})}{\mathbf{p}(i_j)}\leq\frac{1+\varepsilon_4}{1+\varepsilon_2}\cdot\frac{\mathbf{p}^*(S_{a_r})}{\mathbf{p}^*(i_j)}\leq\left(1-\frac{\varepsilon_4}{2}\right)\cdot\frac{\mathbf{p}^*(S_{a_r})}{\mathbf{p}^*(i_j)}.$$

By Lemma 4.1.2, with probability at least $1-\frac{1}{10\ell h}$ the call to $\textsc{Compare}(\{i_j\},S_{a_r},\frac{\varepsilon_4}{8},4,\frac{1}{10\ell h})$ in Line 10 either outputs an element of $\{\mathsf{high},\mathsf{low}\}$ or outputs a value $v\leq(1-\frac{\varepsilon_4}{2})(1+\frac{\varepsilon_4}{8})\frac{\mathbf{p}^*(S_{a_r})}{\mathbf{p}^*(i_j)}<(1-\frac{\varepsilon_4}{4})\frac{\mathbf{p}^*(S_{a_r})}{\mathbf{p}^*(i_j)}$. In either case the algorithm outputs reject in Line 10, so we are done in Case 2(b). This concludes the proof of soundness and the proof of Theorem 4.1.19. □

4.1.5 Testing equality between two unknown distributions

4.1.5.1 An approach based on PAIRCOND queries

In this subsection we consider the problem of testing whether two unknown distributions $\mathbf{p}_1,\mathbf{p}_2$ are identical versus ε-far, given PAIRCOND access to these distributions. Although this is known to require $\Omega(n^{2/3})$ many samples in the standard model [20, 174], we are able to give a $\mathrm{poly}(\log n,1/\varepsilon)$-query algorithm using PAIRCOND queries, by taking advantage of comparisons to perform some sort of *clustering* of the domain.

On a high level the algorithm works as follows. First it obtains (with high probability) a small set of points R such that almost every element in $[n]$, except possibly for some negligible subset according to $\mathbf{p}_1$, has probability weight (under $\mathbf{p}_1$) close to some "representative" in R. Next, for each representative r in R it obtains an estimate of the weight, according to $\mathbf{p}_1$, of a set of points $U(r)$ such that $\mathbf{p}_1(u)$ is close to $\mathbf{p}_1(r)$ for each u in $U(r)$ (i.e., r's "neighborhood under $\mathbf{p}_1$"). This is done using the procedure ESTIMATE-NEIGHBORHOOD from Section 4.1.2.2. Note that these neighborhoods can be interpreted roughly as a succinct *cover* of the support of $\mathbf{p}_1$ into (not necessarily disjoint) sets of points, where within each set the points have similar weight (according to $\mathbf{p}_1$). Our algorithm is based on the observation that, if $\mathbf{p}_1$ and $\mathbf{p}_2$ are far from each other, it must be the case that one of these sets, denoted $U(r^*)$, reflects it in one of the following ways: (1) $\mathbf{p}_2(U(r^*))$ differs significantly from $\mathbf{p}_1(U(r^*))$; (2) $U(r^*)$ contains a subset of points $V(r^*)$ such that $\mathbf{p}_2(v)$ differs significantly from $\mathbf{p}_2(r^*)$ for each v in $V(r^*)$, and either $\mathbf{p}_1(V(r^*))$ is relatively large or $\mathbf{p}_2(V(r^*))$ is relatively large. (This structural result is made precise in Lemma 4.1.36). We thus take additional samples, both from $\mathbf{p}_1$ and from $\mathbf{p}_2$, and compare the weight (according to both distributions) of each point in these samples to the representatives in R (using the procedure COMPARE from Section 4.1.2.1). In this manner we detect (with high probability) that either (1) or (2) holds.

We begin by formalizing the notion of a cover discussed above:

Definition 4.1.34 (Weight-Cover)**.** Given a distribution $\mathbf{p}$ on $[n]$ and a parameter $\varepsilon_1 > 0$, we say that a point $i \in [n]$ is ε_1-*covered* by a set $R = \{r_1, \ldots, r_t\} \subseteq [n]$ if there exists a point $r_j \in R$ such that $\mathbf{p}(i) \in [1/(1+\varepsilon_1), 1+\varepsilon_1]\mathbf{p}(r_j)$. Let the set of points in $[n]$ that are ε_1-covered by R be denoted by $U^{\mathbf{p}}_{\varepsilon_1}(R)$. We say that R is an $(\varepsilon_1, \varepsilon_2)$-*cover for* $\mathbf{p}$ if $\mathbf{p}([n] \setminus U^{\mathbf{p}}_{\varepsilon_1}(R)) \leq \varepsilon_2$.

For a singleton set $R = \{r\}$ we slightly abuse notation and write $U^{\mathbf{p}}_{\varepsilon}(r)$ to denote $U^{\mathbf{p}}_{\varepsilon}(R)$; note that this aligns with the notation established in (4.1).

The following lemma says that a small sample of points drawn from $\mathbf{p}$ gives a cover with high probability:

Lemma 4.1.35. *Let $\mathbf{p}$ be any distribution over $[n]$. Given any fixed $c > 0$, there exists a constant $c' > 0$ such that with probability at least $99/100$, a sample R of size $m = c' \frac{\log(n/\varepsilon)}{\varepsilon^2} \cdot \log\left(\frac{\log(n/\varepsilon)}{\varepsilon}\right)$ drawn according to distribution $\mathbf{p}$ is an $(\varepsilon/c, \varepsilon/c)$-cover for $\mathbf{p}$.*

Proof. Let t denote $\lceil \ln(2cn/\varepsilon) \cdot \frac{c}{\varepsilon} \rceil$. We define t "buckets" of points with similar weight under $\mathbf{p}$ as follows: for $i = 0, 1, \ldots, t-1$, define $B_i \subseteq [n]$ to be

$$B_i \stackrel{\text{def}}{=} \left\{ x \in [n] : \frac{1}{(1+\varepsilon/c)^{i+1}} < \mathbf{p}(x) \leq \frac{1}{(1+\varepsilon/c)^i} \right\}.$$

Let L be the set of points x which are not in any of $B_0, \ldots, B_{t-1}$ (because $\mathbf{p}(x)$ is too small); since every point in L has $\mathbf{p}(x) < \frac{\varepsilon}{2cn}$, one can see that $\mathbf{p}(L) \leq \frac{\varepsilon}{2c}$.

It is easy to see that if the sample R contains a point from a bucket B_j then every point $y \in B_j$ is $\frac{\varepsilon}{c}$-covered by R. We say that bucket B_i is *insignificant* if $\mathbf{p}(B_i) \leq \frac{\varepsilon}{2ct}$; otherwise bucket B_i is *significant*. It is clear that

the total weight under $\mathbf{p}$ of all insignificant buckets is at most $\varepsilon/2c$. Thus if we can show that for the claimed sample size, with probability at least $99/100$ every significant bucket has at least one of its points in R, we will have established the lemma.

This is a simple probabilistic calculation: fix any significant bucket B_j. The probability that m random draws from $\mathbf{p}$ all miss B_j is at most $(1 - \frac{\varepsilon}{2ct})^m$, which is at most $\frac{1}{100t}$ for a suitable (absolute constant) choice of c'. Thus a union bound over all (at most t) significant buckets gives that with probability at least $99/100$, no significant bucket is missed by R. □

Lemma 4.1.36. *Suppose $\mathrm{d_{TV}}(\mathbf{p}_1, \mathbf{p}_2) \geq \varepsilon$, and let $R = \{r_1, \ldots, r_t\}$ be an $(\tilde{\varepsilon}, \tilde{\varepsilon})$-cover for $\mathbf{p}_1$ where $\tilde{\varepsilon} \leq \varepsilon/100$. Then, there exists $j \in [t]$ such that at least one of the following conditions holds for every $\alpha \in [\tilde{\varepsilon}, 2\tilde{\varepsilon}]$:*

1. *$\mathbf{p}_1(U_\alpha^{\mathbf{p}_1}(r_j)) \geq \frac{\tilde{\varepsilon}}{t}$ and $\mathbf{p}_2(U_\alpha^{\mathbf{p}_1}(r_j)) \notin [1-\tilde{\varepsilon}, 1+\tilde{\varepsilon}]\mathbf{p}_1(U_\alpha^{\mathbf{p}_1}(r_j))$, or $\mathbf{p}_1(U_\alpha^{\mathbf{p}_1}(r_j)) < \frac{\tilde{\varepsilon}}{t}$ and $\mathbf{p}_2(U_\alpha^{\mathbf{p}_1}(r_j)) > \frac{2\tilde{\varepsilon}}{t}$;*
2. *$\mathbf{p}_1(U_\alpha^{\mathbf{p}_1}(r_j)) \geq \frac{\tilde{\varepsilon}}{t}$, and at least a $\tilde{\varepsilon}$-fraction of the points i in $U_\alpha^{\mathbf{p}_1}(r_j)$ satisfy $\frac{\mathbf{p}_2(i)}{\mathbf{p}_2(r_j)} \notin [1/(1+\alpha+\tilde{\varepsilon}), 1+\alpha+\tilde{\varepsilon}]$;*
3. *$\mathbf{p}_1(U_\alpha^{\mathbf{p}_1}(r_j)) \geq \frac{\tilde{\varepsilon}}{t}$, and the total weight according to $\mathbf{p}_2$ of the points i in $U_\alpha^{\mathbf{p}_1}(r_j)$ for which $\frac{\mathbf{p}_2(i)}{\mathbf{p}_2(r_j)} \notin [1/(1+\alpha+\tilde{\varepsilon}), 1+\alpha+\tilde{\varepsilon}]$ is at least $\frac{\tilde{\varepsilon}^2}{t}$;*

Proof. Without loss of generality, we can assume that $\varepsilon \leq 1/4$. Suppose, contrary to the claim, that for each r_j there exists $\alpha_j \in [\tilde{\varepsilon}, 2\tilde{\varepsilon}]$ such that if we let $U_j \stackrel{\mathrm{def}}{=} U_{\alpha_j}^{\mathbf{p}_1}(r_j)$, then the following holds:

1. If $\mathbf{p}_1(U_j) < \frac{\tilde{\varepsilon}}{t}$, then $\mathbf{p}_2(U_j) \leq \frac{2\tilde{\varepsilon}}{t}$;
2. If $\mathbf{p}_1(U_j) \geq \frac{\tilde{\varepsilon}}{t}$, then:
   a) $\mathbf{p}_2(U_j) \in [1-\tilde{\varepsilon}, 1+\tilde{\varepsilon}]\mathbf{p}_1(U_j)$;
   b) Less than an $\tilde{\varepsilon}$-fraction of the points y in U_j satisfy $\frac{\mathbf{p}_2(y)}{\mathbf{p}_2(r_j)} \notin [1/(1+\alpha_j+\tilde{\varepsilon}), 1+\alpha_j+\tilde{\varepsilon}]$;
   c) The total weight according to $\mathbf{p}_2$ of the points y in U_j for which $\frac{\mathbf{p}_2(y)}{\mathbf{p}_2(r_j)} \notin [1/(1+\alpha_j+\tilde{\varepsilon}), 1+\alpha_j+\tilde{\varepsilon}]$ is at most $\frac{\tilde{\varepsilon}^2}{t}$;

We show that in such a case $\mathrm{d_{TV}}(\mathbf{p}_1, D_2) < \varepsilon$, contrary to the premise of the claim.

Consider each point $r_j \in R$ such that $\mathbf{p}_1(U_j) \geq \frac{\tilde{\varepsilon}}{t}$. By the foregoing discussion (point 2(a)), $\mathbf{p}_2(U_j) \in [1-\tilde{\varepsilon}, 1+\tilde{\varepsilon}]\mathbf{p}_1(U_j)$. By the definition of U_j (and since $\alpha_j \leq 2\tilde{\varepsilon}$),

$$\mathbf{p}_1(r_j) \in [1/(1+2\tilde{\varepsilon}), 1+2\tilde{\varepsilon}]\,\frac{\mathbf{p}_1(U_j)}{|U_j|}\,. \tag{4.45}$$

Turning to bound $\mathbf{p}_2(r_j)$, on one hand (by 2(b))

$$\mathbf{p}_2(U_j) = \sum_{y \in U_j} \mathbf{p}_2(y) \geq \tilde{\varepsilon}|U_j| \cdot 0 + (1-\tilde{\varepsilon})|U_j| \cdot \frac{\mathbf{p}_2(r_j)}{1+3\tilde{\varepsilon}}\,, \tag{4.46}$$

and so

$$\mathbf{p}_2(r_j) \leq \frac{(1+3\tilde{\varepsilon})\mathbf{p}_2(U_j)}{(1-\tilde{\varepsilon})\,|U_j|} \leq (1+6\tilde{\varepsilon})\frac{\mathbf{p}_1(U_j)}{|U_j|}\,. \tag{4.47}$$

On the other hand (by 2(c)),

$$\mathbf{p}_2(U_j) = \sum_{y \in U_j} \mathbf{p}_2(y) \leq \frac{\tilde{\varepsilon}^2}{t} + |U_j| \cdot (1+3\tilde{\varepsilon})\mathbf{p}_2(r_j) \;, \tag{4.48}$$

and so

$$\mathbf{p}_2(r_j) \geq \frac{\mathbf{p}_2(U_j) - \tilde{\varepsilon}^2/t}{(1+3\tilde{\varepsilon})\,|U_j|} \geq \frac{(1-\tilde{\varepsilon})\mathbf{p}_1(U_j) - \tilde{\varepsilon}\mathbf{p}_1(U_j)}{(1+3\tilde{\varepsilon})\,|U_j|} \geq (1-5\tilde{\varepsilon})\frac{\mathbf{p}_1(U_j)}{|U_j|} \;. \tag{4.49}$$

Therefore, for each such r_j we have

$$\mathbf{p}_2(r_j) \in [1-8\tilde{\varepsilon}, 1+10\tilde{\varepsilon}]\mathbf{p}_1(r_j) \;. \tag{4.50}$$

Let $C \stackrel{\text{def}}{=} \bigcup_{j=1}^{t} U_j$. We next partition the points in C so that each point $i \in C$ is assigned to some $r_{j(i)}$ such that $i \in U_{j(i)}$. We define the following "bad" subsets of points in $[n]$:

1. $B_1 \stackrel{\text{def}}{=} [n] \setminus C$, so that $\mathbf{p}_1(B_1) \leq \tilde{\varepsilon}$ (we later bound $\mathbf{p}_2(B_1)$);
2. $B_2 \stackrel{\text{def}}{=} \left\{i \in C : \mathbf{p}_1(U_{j(i)}) < \tilde{\varepsilon}/t\right\}$, so that $\mathbf{p}_1(B_2) \leq \tilde{\varepsilon}$ and $\mathbf{p}_2(B_2) \leq 2\tilde{\varepsilon}$;
3. $B_3 \stackrel{\text{def}}{=} \left\{i \in C \setminus B_2 : \mathbf{p}_2(i) \notin [1/(1+3\tilde{\varepsilon}), 1+3\tilde{\varepsilon}]\mathbf{p}_2(r_{j(i)})\right\}$, so that $\mathbf{p}_1(B_3) \leq 2\tilde{\varepsilon}$ and $\mathbf{p}_2(B_3) \leq \tilde{\varepsilon}^2$.

Let $B \stackrel{\text{def}}{=} B_1 \cup B_2 \cup B_3$. Observe that for each $i \in [n] \setminus B$ we have that

$$\mathbf{p}_2(i) \in [1/(1+3\tilde{\varepsilon}), 1+3\tilde{\varepsilon}]\mathbf{p}_2(r_{j(i)}) \subset [1-15\tilde{\varepsilon}, 1+15\tilde{\varepsilon}]\mathbf{p}_1(r_{j(i)}) \subset [1-23\tilde{\varepsilon}, 1+23\tilde{\varepsilon}]\mathbf{p}_1(i) \;, \tag{4.51}$$

where the first containment follows from the fact that $i \notin B$, the second follows from Eq. (4.50), and the third from the fact that $i \in U_{j(i)}$. In order to complete the proof we need a bound on $\mathbf{p}_2(B_1)$, which we obtain next.

$$\begin{aligned} \mathbf{p}_2(B_1) &= 1 - \mathbf{p}_2([n] \setminus B_1) \leq 1 - \mathbf{p}_2([n] \setminus B) \leq 1 - (1-23\tilde{\varepsilon})\mathbf{p}_1([n] \setminus B) \\ &\leq 1 - (1-23\tilde{\varepsilon})(1-4\tilde{\varepsilon}) \leq 27\tilde{\varepsilon} \;. \end{aligned} \tag{4.52}$$

Therefore,

$$\begin{aligned} \mathrm{d}_{\mathrm{TV}}(\mathbf{p}_1, \mathbf{p}_2) &= \frac{1}{2}\sum_{i=1}^{n} |\mathbf{p}_1(i) - \mathbf{p}_2(i)| \\ &\leq \frac{1}{2}\Big(\mathbf{p}_1(B) + \mathbf{p}_2(B) + \sum_{i \notin B} 23\tilde{\varepsilon}\mathbf{p}_1(i)\Big) < \varepsilon \;, \end{aligned} \tag{4.53}$$

and we have reached a contradiction. □

Theorem 4.1.37. *If $\mathbf{p}_1 = \mathbf{p}_2$ then with probability at least $2/3$ Algorithm* PAIRCOND-TEST-EQUALITY-UNKNOWN *returns* accept*, and if $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}_1, \mathbf{p}_2) \geq \varepsilon$, then with probability at least $2/3$ Algorithm* PAIRCOND-TEST-EQUALITY-UNKNOWN *returns* reject*. The number of* PAIRCOND *queries performed by the algorithm is $\tilde{O}\left(\frac{\log^6 n}{\varepsilon^{21}}\right)$.*

Algorithm 24 Algorithm $\text{PAIRCOND}_{\mathbf{p}_1,\mathbf{p}_2}$-TEST-EQUALITY-UNKNOWN

Require: PAIRCOND query access to distributions $\mathbf{p}_1$ and $\mathbf{p}_2$ and a parameter ε.

1: Set $\tilde{\varepsilon} = \varepsilon/100$.
2: Draw a sample R of size $t = \tilde{\Theta}\left(\frac{\log n}{\varepsilon^2}\right)$ from $\mathbf{p}_1$.
3: **for all** $r_j \in R$ **do**
4: Call ESTIMATE-NEIGHBORHOOD$_{\mathbf{p}_1}$ on r_j with $\kappa = \tilde{\varepsilon}$, $\eta = \frac{\tilde{\varepsilon}}{8}$, $\beta = \frac{\tilde{\varepsilon}}{2t}$, $\delta = \frac{1}{100t}$ and let the output be denoted by $(\hat{w}_j^{(1)}, \alpha_j)$.
5: Set $\theta = \kappa\eta\beta\delta/64 = \tilde{\Theta}(\varepsilon^7/\log^2 n)$.
6: Draw a sample S_1 from $\mathbf{p}_1$, of size $s_1 = \Theta\left(\frac{t}{\varepsilon^2}\right) = \tilde{\Theta}\left(\frac{\log n}{\varepsilon^4}\right)$.
7: Draw a sample S_2 from $\mathbf{p}_2$, of size $s_2 = \Theta\left(\frac{t \log t}{\varepsilon^3}\right) = \tilde{\Theta}\left(\frac{\log n}{\varepsilon^5}\right)$.
8: For each point $i \in S_1 \cup S_2$ call $\text{COMPARE}_{\mathbf{p}_1}(\{r_j\}, \{i\}, \theta/4, 4, 1/(200t(s_1+s_2)))$ and $\text{COMPARE}_{\mathbf{p}_2}(\{r_j\}, \{i\}, \theta/4, 4, 1/(200t(s_1+s_2)))$, and let the outputs be denoted $\rho_{r_j}^{(1)}(i)$ and $\rho_{r_j}^{(2)}(i)$, respectively (where in particular these outputs may be high or low).
9: Let $\hat{w}_j^{(2)}$ be the fraction of occurrences of $i \in S_2$ such that $\rho_{r_j}^{(1)}(i) \in [1/(1+\alpha_j+\theta/2), 1+\alpha_j+\theta/2]$.
10: **if** ($\hat{w}_j^{(1)} \leq \frac{3}{4}\frac{\tilde{\varepsilon}}{t}$ and $\hat{w}_j^{(2)} > \frac{3}{2}\frac{\tilde{\varepsilon}}{t}$) or ($\hat{w}_j^{(1)} > \frac{3}{4}\frac{\tilde{\varepsilon}}{t}$ and $\hat{w}_j^{(2)}/\hat{w}_j^{(1)} \notin [1-\tilde{\varepsilon}/2, 1+\tilde{\varepsilon}/2]$) **then**
11: **return** reject
12: **end if**
13: **if** there exists $i \in S_1 \cup S_2$ such that $\rho_{r_j}^{(1)}(i) \in [1/(\alpha_j+\tilde{\varepsilon}/2), 1+\alpha_j+\tilde{\varepsilon}/2]$ and $\rho_{r_j}^{(2)}(i) \notin [1/(\alpha_j+3\tilde{\varepsilon}/2), 1+\alpha_j+3\tilde{\varepsilon}/2]$, **then**
14: **return** reject
15: **end if**
16: **end for**
17: **return** accept.

Proof. The number of queries performed by the algorithm is the sum of: (1) t times the number of queries performed in each execution of ESTIMATE-NEIGHBORHOOD (in Line 4) and (2) $t \cdot (s_1+s_2) = O(t \cdot s_2)$ times the number of queries performed in each execution of COMPARE (in Line 8). By Lemma 4.1.3 (and the settings of the parameters in the calls to ESTIMATE-NEIGHBORHOOD), the first term is $O\left(t \cdot \frac{\log(1/\delta)\cdot\log(\log(1/\delta)/(\delta\beta\eta^2))}{\kappa^2\eta^4\beta^3\delta^2}\right) = \tilde{O}\left(\frac{\log^6 n}{\varepsilon^{21}}\right)$, and by Lemma 4.1.2 (and the settings of the parameters in the calls to COMPARE), the second term is $O\left(t \cdot s_2 \cdot \frac{\log(t\cdot s_2)}{\theta^2}\right) = \tilde{O}\left(\frac{\log^6 n}{\varepsilon^{21}}\right)$, so that we get the bound stated in the theorem.

We now turn to establishing the correctness of the algorithm. We shall use the shorthand U_j for $U_{\alpha_j}^{\mathbf{p}_1}(r_j)$, and U'_j for $U_{\alpha_j+\theta}^{\mathbf{p}_1}(r_j)$. We consider the following "desirable" events.

1. The event E_1 is that the sample R is a $(\tilde{\varepsilon}, \tilde{\varepsilon})$-weight-cover for $\mathbf{p}_1$ (for $\tilde{\varepsilon} = \varepsilon/100$). By Lemma 4.1.35 (and an appropriate constant in the $\Theta(\cdot)$ notation for the size of R), the probability that E_1 holds is at least $99/100$.
2. The event E_2 is that all calls to the procedure ESTIMATE-NEIGHBORHOOD are as specified by Lemma 4.1.3. By the setting of the confidence parameter in the calls to the procedure, the event E_2 holds with probability at least $99/100$.
3. The event E_3 is that all calls to the procedure COMPARE are as specified by Lemma 4.1.2. By the setting of the confidence parameter in the calls to the procedure, the event E_3 holds with probability at least $99/100$.

4. The event E_4 is that $\mathbf{p}_2(U'_j \setminus U_j) \leq \eta\beta/16 = \tilde{\varepsilon}^2/(256t)$ for each j. If $\mathbf{p}_2 = \mathbf{p}_1$ then this event follows from E_2. Otherwise, it holds with probability at least $99/100$ by the setting of θ and the choice of α_j (as shown in the proof of Lemma 4.1.3 in the analysis of the event E_1 there.

5. The event E_5 is defined as follows. For each j, if $\mathbf{p}_2(U_j) \geq \tilde{\varepsilon}/(4t)$, then $|S_2 \cap U_j|/|S_2| \in [1 - \tilde{\varepsilon}/10, 1 + \tilde{\varepsilon}/10]\mathbf{p}_2(U_j)$, and if $\mathbf{p}_2(U_j) < \tilde{\varepsilon}/(4t)$ then $|S_2 \cap U_j|/|S_2| < (1 + \tilde{\varepsilon}/10)\tilde{\varepsilon}/(4t)$. This event holds with probability at least $99/100$ by applying a multiplicative Chernoff bound in the first case, and Claim 1.4.11 in the second.

6. The event E_6 is that for each j we have $|S_2 \cap (U'_j \setminus U_j)|/|S_2| \leq \tilde{\varepsilon}^2/(128t)$. Conditioned on E_4, the event E_6 holds with probability at least $99/100$ by applying Claim 1.4.11.

From this point on we assume that events $E_1 - E_6$ all hold. Note that in particular this implies the following:

1. By E_2, for every j:
   - If $\mathbf{p}_1(U_j) \geq \beta = \tilde{\varepsilon}/(2t)$, then $\hat{w}_j^{(1)} \in [1 - \eta, 1 + \eta]\mathbf{p}_1(U_j) = [1 - \tilde{\varepsilon}/8, 1 + \tilde{\varepsilon}/8]\mathbf{p}_1(U_j)$.
   - If $\mathbf{p}_1(U_j) < \tilde{\varepsilon}/(2t)$, then $\hat{w}_j^{(1)} \leq (1 + \tilde{\varepsilon}/8)(\tilde{\varepsilon}/(2t))$.
2. By E_3, for every j and for each point $i \in S_1 \cup S_2$:
   - If $i \in U_j$, then $\rho_{r_j}^{(1)}(i) \in [1/(1 + \alpha_j + \frac{\theta}{2}), 1 + \alpha_j + \frac{\theta}{2}]$.
   - If $i \notin U'_j$, then $\rho_{r_j}^{(1)}(i) \notin [1/(1 + \alpha_j + \frac{\theta}{2}), 1 + \alpha_j + \frac{\theta}{2}]$.
3. By the previous item and E_4–E_6:
   - If $\mathbf{p}_2(U_j) \geq \tilde{\varepsilon}/(4t)$, then $\hat{w}_j^{(2)} \geq (1-\tilde{\varepsilon}/10)\mathbf{p}_2(U_j)$ and $\hat{w}_j^{(2)} \leq (1+\tilde{\varepsilon}/10)\mathbf{p}_2(U_j)+\tilde{\varepsilon}^2/(128t) \leq (1 + \tilde{\varepsilon}/8)\mathbf{p}_2(U_j)$.
   - If $\mathbf{p}_2(U_j) < \tilde{\varepsilon}/(4t)$ then $\hat{w}_j^{(2)} \leq (1 + \tilde{\varepsilon}/10)\tilde{\varepsilon}/(4t) + \tilde{\varepsilon}^2/(128t) \leq (1 + \tilde{\varepsilon}/4)(\tilde{\varepsilon}/(4t))$.

Completeness. Assume $\mathbf{p}_1$ and $\mathbf{p}_2$ are the same distribution $\mathbf{p}$. For each j, if $\mathbf{p}(U_j) \geq \tilde{\varepsilon}/t$, then by the foregoing discussion, $\hat{w}_j^{(1)} \geq (1 - \tilde{\varepsilon}/8)\mathbf{p}(U_j) > 3\tilde{\varepsilon}/(4t)$ and $\hat{w}_j^{(2)}/\hat{w}_j^{(1)} \in [(1 - \tilde{\varepsilon}/8)^2, (1 + \tilde{\varepsilon}/8)^2] \subset [1 - \tilde{\varepsilon}/2, 1 + \tilde{\varepsilon}/2]$, so that the algorithm does not reject in Line 10. Otherwise (i.e., $\mathbf{p}(U_j) < \tilde{\varepsilon}/t$), we consider two subcases. Either $\mathbf{p}(U_j) \leq \tilde{\varepsilon}/(2t)$, in which case $\hat{w}_j^{(1)} \leq 3\tilde{\varepsilon}/(4t)$, or $\tilde{\varepsilon}/(2t) < \mathbf{p}(U_j) < \tilde{\varepsilon}/t$, and then $\hat{w}_j^{(1)} \in [1-\tilde{\varepsilon}/8, 1+\tilde{\varepsilon}/8]\mathbf{p}_1(U_j)$. Since in both cases $\hat{w}_j^{(2)} \leq 3\tilde{\varepsilon}/(2t)$, the algorithm does not reject in Line 10. By E_3, the algorithm does not reject in Line 13 either. We next turn to establish soundness.

Soundness. Assume $\mathrm{d_{TV}}(\mathbf{p}_1, \mathbf{p}_2) \geq \varepsilon$. By applying Lemma 4.1.36 on R (and using E_1), there exists an index j for which one of the items in the lemma holds. We denote this index by j^*, and consider the three items in the lemma.

1. If Item 1 holds, then we consider its two cases:
   a) In the first case, $\mathbf{p}_1(U_{j^*}) \geq \tilde{\varepsilon}/t$ and $\mathbf{p}_2(U_{j^*}) \notin [1 - \tilde{\varepsilon}, 1 + \tilde{\varepsilon}]\mathbf{p}_1(U_{j^*})$. Due to the lower bound on $\mathbf{p}_1(U_{j^*})$ we have that $\hat{w}_{j^*}^{(1)} \in [1 - \tilde{\varepsilon}/8, 1 + \tilde{\varepsilon}/8]\mathbf{p}_1(U_{j^*})$, so that in particular $\hat{w}_{j^*}^{(1)} > 3\tilde{\varepsilon}/(4t)$. As

for $\hat{w}_{j^*}^{(2)}$, either $\hat{w}_{j^*}^{(2)} < (1-\tilde{\varepsilon})(1+\tilde{\varepsilon}/8)\mathbf{p}_1(U_{j^*})$ (this holds both when $\mathbf{p}_2(U_{j^*}) \geq \tilde{\varepsilon}/(4t)$ and when $\mathbf{p}_2(U_{j^*}) < \tilde{\varepsilon}/(4t)$) or $\hat{w}_{j^*}^{(2)} > (1+\tilde{\varepsilon})(1-\tilde{\varepsilon}/10)\mathbf{p}_1(U_{j^*})$. In either (sub)case $\hat{w}_{j^*}^{(2)}/\hat{w}_{j^*}^{(1)} \notin [1-\tilde{\varepsilon}/2, 1+\tilde{\varepsilon}/2]$, causing the algorithm to reject in (the second part of) Line 10.

b) In the second case, $\mathbf{p}_1(U_{j^*}) < \tilde{\varepsilon}/t$ and $\mathbf{p}_2(U_{j^*}) > 2\tilde{\varepsilon}/t$. Due to the lower bound on $\mathbf{p}_2(U_{j^*})$ we have that $\hat{w}_{j^*}^{(2)} \geq (1-\tilde{\varepsilon}/10)\mathbf{p}_2(U_{j^*}) > (1-\tilde{\varepsilon}/10)(2\tilde{\varepsilon}/t)$, so that in particular $\hat{w}_{j^*}^{(2)} > (3\tilde{\varepsilon}/(2t))$. As for $\hat{w}_{j^*}^{(1)}$, if $\mathbf{p}_1(U_{j^*}) \leq \tilde{\varepsilon}/(2t)$, then $\hat{w}_{j^*}^{(1)} \leq 3\tilde{\varepsilon}/(4t)$, causing the algorithm to reject in (the first part of) Line 10. If $\tilde{\varepsilon}/(2t) < \mathbf{p}_1(U_{j^*}) \leq \tilde{\varepsilon}/t$, then $\hat{w}_{j^*}^{(1)} \in [1-\tilde{\varepsilon}/8, 1+\tilde{\varepsilon}/8]\mathbf{p}_1(U_{j^*}) \leq (1+\tilde{\varepsilon}/8)(\tilde{\varepsilon}/t)$, so that $\hat{w}_{j^*}^{(2)}/\hat{w}_{j^*}^{(1)} \geq \frac{(1-\tilde{\varepsilon}/10)(2\tilde{\varepsilon}/t)}{(1+\tilde{\varepsilon}/8)\tilde{\varepsilon}/t} > (1+\tilde{\varepsilon}/2)$, causing the algorithm to reject in (either the first or second part of) Line 10.

2. If Item 2 holds, then, by the choice of the size of S_1, which is $\Theta(t/\tilde{\varepsilon}^2)$, and since all points in U_{j^*} have approximately the same weight according to $\mathbf{p}_1$, with probability at least $99/100$, the sample S_1 will contain a point i for which $\frac{\mathbf{p}_2(i)}{\mathbf{p}_2(r_{j^*})} \notin [1/(1+\alpha_{j^*}+\tilde{\varepsilon}), 1+\alpha_{j^*}+\tilde{\varepsilon}]$, and by E_3 this will be detected in Line 13.

3. Similarly, if Item 3 holds, then by the choice of the size of S_2, with probability at least $99/100$, the sample S_2 will contain a point i for which $\frac{\mathbf{p}_2(i)}{\mathbf{p}_2(r_{j^*})} \notin [1/(1+\alpha_{j^*}+\tilde{\varepsilon}), 1+\alpha_{j^*}+\tilde{\varepsilon}]$, and by E_3 this will be detected in Line 13.

The theorem is thus established. □

4.1.5.2 An approach based on simulating EVAL

In this subsection we present an alternate approach for testing whether two unknown distributions $\mathbf{p}_1, \mathbf{p}_2$ are identical versus ε-far. We prove the following theorem:

Theorem 4.1.38. *There exists an algorithm that has the following properties: given query access to* $\mathsf{COND}_{\mathbf{p}_1}$ *and* $\mathsf{COND}_{\mathbf{p}_2}$ *oracles for any two distributions* $\mathbf{p}_1, \mathbf{p}_2$ *over* $[n]$*, the algorithm outputs* ***accept*** *with probability at least* $2/3$ *if* $\mathbf{p}_1 = \mathbf{p}_2$ *and outputs* ***reject*** *with probability at least* $2/3$ *if* $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}_1, \mathbf{p}_2) \geq \varepsilon$*. The algorithm performs*

$$\tilde{O}\left(\frac{(\log n)^5}{\varepsilon^4}\right)$$

queries.

At the heart of this result is our efficient simulation of an "approximate $\mathsf{EVAL}_{\mathbf{p}}$ oracle" using a $\mathsf{COND}_{\mathbf{p}}$ oracle. (Recall that an $\mathsf{EVAL}_{\mathbf{p}}$ oracle is an oracle which, given as input an element $i \in [n]$, outputs the numerical value $\mathbf{p}(i)$.) We feel that this efficient simulation of an approximate EVAL oracle using a COND oracle is of independent interest since it sheds light on the relative power of the COND and EVAL models.

In more detail, the starting point of our approach to prove Theorem 4.1.38 is a simple algorithm from [155] that uses an $\mathsf{EVAL}_{\mathbf{p}}$ oracle to test equality between $\mathbf{p}$ and a known distribution $\mathbf{p}^*$. We first show (see Theorem 4.1.39) that a modified version of the algorithm, which uses a SAMP oracle and an "approximate" EVAL oracle, can be used to efficiently test equality between two unknown distributions $\mathbf{p}_1$ and $\mathbf{p}_2$. As we

show (in Section 4.1.2.3) the required "approximate" EVAL oracle can be efficiently implemented using a COND oracle, and so Theorem 4.1.38 follows straightforwardly by combining Theorems 4.1.39 and 4.1.5.

Testing equality between $\mathbf{p}_1$ and $\mathbf{p}_2$ using an approximate EVAL oracle. We now show how an approximate $\mathsf{EVAL}_{\mathbf{p}_1}$ oracle, an approximate $\mathsf{EVAL}_{\mathbf{p}_2}$ oracle, and a $\mathsf{SAMP}_{\mathbf{p}_1}$ oracle can be used together to test whether $\mathbf{p}_1 = \mathbf{p}_2$ versus $\mathrm{d_{TV}}(\mathbf{p}_1, \mathbf{p}_2) \geq \varepsilon$. As mentioned earlier, the approach is a simple extension of the EVAL algorithm given in [155, Observation 24].

Theorem 4.1.39. *Let* ORACLE_1 *be an* $(\varepsilon/100, \varepsilon/100)$*-approximate* $\mathsf{EVAL}_{\mathbf{p}_1}$ *simulator and let* ORACLE_2 *be an* $(\varepsilon/100, \varepsilon/100)$*-approximate* $\mathsf{EVAL}_{\mathbf{p}_2}$ *simulator. There is an algorithm* TEST-EQUALITY-UNKNOWN *with the following properties: for any distributions* $\mathbf{p}_1, \mathbf{p}_2$ *over* $[n]$*, algorithm* TEST-EQUALITY-UNKNOWN *makes* $O(1/\varepsilon)$ *queries to* ORACLE_1, ORACLE_2, $\mathsf{SAMP}_{\mathbf{p}_1}$, $\mathsf{SAMP}_{\mathbf{p}_2}$*, and it outputs* ***accept*** *with probability at least* $7/10$ *if* $\mathbf{p}_1 = \mathbf{p}_2$ *and outputs* ***reject*** *with probability at least* $7/10$ *if* $\mathrm{d_{TV}}(\mathbf{p}_1, \mathbf{p}_2) \geq \varepsilon$.

Algorithm 25 TEST-EQUALITY-UNKNOWN

Require: query access to ORACLE_1, to ORACLE_2, and access to $\mathsf{SAMP}_{\mathbf{p}_1}$, $\mathsf{SAMP}_{\mathbf{p}_2}$ oracles

1: Call the $\mathsf{SAMP}_{\mathbf{p}_1}$ oracle $m = 5/\varepsilon$ times to obtain points $h_1, \ldots, h_m$ distributed according to $\mathbf{p}_1$.
2: Call the $\mathsf{SAMP}_{\mathbf{p}_2}$ oracle $m = 5/\varepsilon$ times to obtain points $h_{m+1}, \ldots, h_{2m}$ distributed according to $\mathbf{p}_2$.
3: **for** $j = 1$ to $2m$ **do**
4: Call $\mathsf{ORACLE}_1(h_j)$. If it returns unknown then **return** reject, otherwise let $v_{1,i} \in [0, 1]$ be the value it outputs.
5: Call $\mathsf{ORACLE}_2(h_j)$. If it returns unknown then **return** reject, otherwise let $v_{2,i} \in [0, 1]$ be the value it outputs.
6: **if** $v_{1,j} \notin [1 - \varepsilon/8, 1 + \varepsilon/8]v_{2,j}$ **then**
7: **return** reject and exit
8: **end if**
9: **end for**
10: **return** accept

It is clear that TEST-EQUALITY-UNKNOWN makes $O(1/\varepsilon)$ queries as claimed. To prove Theorem 4.1.39 we argue completeness and soundness below.

Completeness: Suppose that $\mathbf{p}_1 = \mathbf{p}_2$. Since ORACLE_1 is an $(\varepsilon/100, \varepsilon/100)$-approximate $\mathsf{EVAL}_{\mathbf{p}_1}$ simulator, the probability that any of the $2m = 10/\varepsilon$ points $h_1, \ldots, h_{2m}$ drawn in Lines 1 and 2 lies in $S^{(\varepsilon/100, \mathbf{p}_1)}$ is at most $1/10$. Going forth, let us assume that all points h_i indeed lie outside $S^{(\varepsilon/100, \mathbf{p}_1)}$. Then for each execution of Line 4 we have that with probability at least $1 - \varepsilon/100$ the call to $\mathsf{ORACLE}(h_i)$ yields a value $v_{1,i}$ satisfying $v_{1,i} \in [1 - \frac{\varepsilon}{100}, 1 + \frac{\varepsilon}{100}]\mathbf{p}_1(i)$. The same holds for each execution of Line 5. Since there are $20/\varepsilon$ total executions of Lines 4 and 5, with overall probability at least 7/10 we have that each $1 \leq j \leq m$ has $v_{1,j}, v_{2,j} \in [1 - \frac{\varepsilon}{100}, 1 + \frac{\varepsilon}{100}]\mathbf{p}_1(i)$. If this is the case then $v_{1,j}, v_{2,j}$ pass the check in Line 6, and thus the algorithm outputs accept with overall probability at least $7/10$.

Soundness: Now suppose that $\mathrm{d_{TV}}(\mathbf{p}_1, \mathbf{p}_2) \geq \varepsilon$. Let us say that $i \in [n]$ is *good* if $\mathbf{p}_1(i) \in [1-\varepsilon/5, 1+\varepsilon/5]\mathbf{p}_2(i)$. Let BAD $\subseteq [n]$ denote the set of all $i \in [n]$ that are not good. We have

$$2\,\mathrm{d_{TV}}(\mathbf{p}_1, \mathbf{p}_2) = \sum_{i \text{ is good}} |\mathbf{p}_1(i) - \mathbf{p}_2(i)| + \sum_{i \text{ is bad}} |\mathbf{p}_1(i) - \mathbf{p}_2(i)| \geq 2\varepsilon.$$

Since

$$\sum_{i \text{ is good}} |\mathbf{p}_1(i) - \mathbf{p}_2(i)| \leq \sum_{i \text{ is good}} \frac{\varepsilon}{5}|\mathbf{p}_2(i)| \leq \frac{\varepsilon}{5},$$

we have

$$\sum_{i \text{ is bad}} (|\mathbf{p}_1(i)| + |\mathbf{p}_2(i)|) \geq \sum_{i \text{ is bad}} |\mathbf{p}_1(i) - \mathbf{p}_2(i)| \geq \frac{9}{5}\varepsilon.$$

Consequently it must be the case that either $\mathbf{p}_1(\text{BAD}) \geq \frac{9}{10}\varepsilon$ or $\mathbf{p}_2(\text{BAD}) \geq \frac{9}{10}\varepsilon$. For the rest of the argument we suppose that $\mathbf{p}_1(\text{BAD}) \geq \frac{9}{10}\varepsilon$ (by the symmetry of the algorithm, an identical argument to the one we give below but with the roles of $\mathbf{p}_1$ and $\mathbf{p}_2$ flipped throughout handles the other case).

Since $\mathbf{p}_1(\text{BAD}) \geq \frac{9}{10}\varepsilon$, a simple calculation shows that with probability at least $98/100$ at least one of the $5/\varepsilon$ points $h_1, \ldots, h_m$ drawn in Line 1 belongs to BAD. For the rest of the argument we suppose that indeed (at least) one of these points is in BAD; let h_{i^*} be such a point. Now consider the execution of Line 4 when ORACLE_1 is called on h_{i^*}. By Definition 4.1.4, whether or not i^* belongs to $S^{(\varepsilon/100, \mathbf{p}_1)}$, with probability at least $1 - \varepsilon/100$ the call to ORACLE_1 either causes TEST-EQUALITY-UNKNOWN to reject in Line 4 (because ORACLE_1 returns unknown) or it returns a value $v_{1,i^*} \in [1 - \frac{\varepsilon}{100}, 1 + \frac{\varepsilon}{100}]\mathbf{p}_1(i^*)$. We may suppose that it returns a value $v_{1,i^*} \in [1 - \frac{\varepsilon}{100}, 1 + \frac{\varepsilon}{100}]\mathbf{p}_1(i^*)$. Similarly, in the execution of Line 5 when ORACLE_2 is called on h_{i^*}, whether or not i^* belongs to $S^{(\varepsilon/100, \mathbf{p}_2)}$, with probability at least $1 - \varepsilon/100$ the call to ORACLE_2 either causes TEST-EQUALITY-UNKNOWN to reject in Line 5 or it returns a value $v_{2,i^*} \in [1 - \frac{\varepsilon}{100}, 1 + \frac{\varepsilon}{100}]\mathbf{p}_2(i^*)$. We may suppose that it returns a value $v_{2,i^*} \in [1 - \frac{\varepsilon}{100}, 1 + \frac{\varepsilon}{100}]\mathbf{p}_2(i^*)$. But recalling that $i^* \in$ BAD, an easy calculation shows that the values v_{1,i^*} and v_{2,i^*} must be multiplicatively far enough from each other that the algorithm will output reject in Line 7. Thus with overall probability at least $96/100$ the algorithm outputs reject. □

4.1.6 An algorithm for estimating the distance to uniformity

In this section we describe an algorithm that estimates the distance between a distribution $\mathbf{p}$ and the uniform distribution $\mathbf{u}$ by performing $\mathrm{poly}(1/\varepsilon)$ PAIRCOND (and SAMP) queries. We start by giving a high level description of the algorithm.

By the definition of the variation distance (and the uniform distribution),

$$\mathrm{d_{TV}}(\mathbf{p}, \mathbf{u}) = \sum_{i:\mathbf{p}(i)<1/n} \left(\frac{1}{n} - \mathbf{p}(i)\right). \tag{4.54}$$

We define the following function over $[n]$:

$$\psi^{\mathbf{p}}(i) = (1 - n \cdot \mathbf{p}(i)) \text{ for } \mathbf{p}(i) < \frac{1}{n}, \text{ and } \psi^{\mathbf{p}}(i) = 0 \text{ for } \mathbf{p}(i) \geq \frac{1}{n}. \tag{4.55}$$

Observe that $\psi^{\mathbf{p}}(i) \in [0,1]$ for every $i \in [n]$ and

$$\mathrm{d_{TV}}(\mathbf{p}, \mathbf{u}) = \frac{1}{n} \sum_{i=1}^{n} \psi^{\mathbf{p}}(i). \tag{4.56}$$

Thus $\mathrm{d_{TV}}(\mathbf{p}, \mathbf{u})$ can be viewed as an average value of a function whose range is in $[0,1]$. Since $\mathbf{p}$ is fixed throughout this subsection, we shall use the shorthand $\psi(i)$ instead of $\psi^{\mathbf{p}}(i)$. Suppose we were able to compute $\psi(i)$ exactly for any i of our choice. Then we could obtain an estimate $\hat{d}$ of $\mathrm{d_{TV}}(\mathbf{p}, \mathbf{u})$ to within an additive error of $\varepsilon/2$ by simply selecting $\Theta(1/\varepsilon^2)$ points in $[n]$ uniformly at random and setting $\hat{d}$ to be the average value of $\psi(\cdot)$ on the sampled points. By an additive Chernoff bound (for an appropriate constant in the $\Theta(\cdot)$ notation), with high constant probability the estimate $\hat{d}$ would deviate by at most $\varepsilon/2$ from $\mathrm{d_{TV}}(\mathbf{p}, \mathbf{u})$.

Suppose next that instead of being able to compute $\psi(i)$ exactly, we were able to compute an estimate $\hat{\psi}(i)$ such that $|\hat{\psi}(i) - \psi(i)| \leq \varepsilon/2$. By using $\hat{\psi}(i)$ instead of $\psi(i)$ for each of the $\Theta(1/\varepsilon^2)$ sampled points we would incur an additional additive error of at most $\varepsilon/2$. Observe first that for i such that $\mathbf{p}(i) \leq \varepsilon/(2n)$ we have that $\psi(i) \geq 1 - \varepsilon/2$, so the estimate $\hat{\psi}(i) = 1$ meets our requirements. Similarly, for i such that $\mathbf{p}(i) \geq 1/n$, any estimate $\hat{\psi}(i) \in [0, \varepsilon/2]$ can be used. Finally, for i such that $\mathbf{p}(i) \in [\varepsilon/(2n), 1/n]$, if we can obtain an estimate $\widehat{\mathbf{p}}(i)$ such that $\widehat{\mathbf{p}}(i) \in [1 - \varepsilon/2, 1 + \varepsilon/2]\mathbf{p}(i)$, then we can use $\hat{\psi}(i) = 1 - n \cdot \widehat{\mathbf{p}}(i)$.

In order to obtain such estimates $\hat{\psi}(i)$, we shall be interested in finding a *reference point* x. Namely, we shall be interested in finding a pair $(x, \widehat{\mathbf{p}}(x))$ such that $\widehat{\mathbf{p}}(x) \in [1 - \varepsilon/c, 1 + \varepsilon/c]\mathbf{p}(x)$ for some sufficiently large constant c, and such that $\mathbf{p}(x) = \Omega(\varepsilon/n)$ and $\mathbf{p}(x) = O(1/(\varepsilon n))$. In Section 4.1.6.1 we describe a procedure for finding such a reference point. More precisely, the procedure is required to find such a reference point (with high constant probability) only under a certain condition on $\mathbf{p}$. It is not hard to verify (and we show this subsequently), that if this condition is not met, then $\mathrm{d_{TV}}(\mathbf{p}, \mathbf{u})$ is very close to 1. In order to state the lemma we introduce the following notation. For $\gamma \in [0,1]$, let

$$H_{\gamma}^{\mathbf{p}} \stackrel{\text{def}}{=} \left\{ i : \mathbf{p}(i) \geq \frac{1}{\gamma n} \right\}. \tag{4.57}$$

Lemma 4.1.40. *Given an input parameter $\kappa \in (0, 1/4]$ as well as* SAMP *and* PAIRCOND *query access to a distribution $\mathbf{p}$, the procedure* FIND-REFERENCE *(Section 4.1.6.1) either returns a pair $(x, \widehat{\mathbf{p}}(x))$ where $x \in [n]$ and $\widehat{\mathbf{p}}(x) \in [0,1]$ or returns* No-Pair. *The procedure satisfies the following:*

1. *If $\mathbf{p}(H_{\kappa}^{\mathbf{p}}) \leq 1 - \kappa$, then with probability at least $9/10$, the procedure returns a pair $(x, \widehat{\mathbf{p}}(x))$ such that $\widehat{\mathbf{p}}(x) \in [1 - 2\kappa, 1 + 3\kappa]\mathbf{p}(x)$ and $\mathbf{p}(x) \in \left[\frac{\kappa}{8}, \frac{4}{\kappa}\right] \cdot \frac{1}{n}$.*

2. *If* $\mathbf{p}(H_\kappa^{\mathbf{p}}) > 1-\kappa$*, then with probability at least* $9/10$*, the procedure either returns* No-Pair *or it returns a pair* $(x, \widehat{\mathbf{p}}(x))$ *such that* $\widehat{\mathbf{p}}(x) \in [1-2\kappa, 1+3\kappa]\mathbf{p}(x)$ *and* $\mathbf{p}(x) \in \left[\frac{\kappa}{8}, \frac{4}{\kappa}\right] \cdot \frac{1}{n}$.

The procedure performs $\tilde{O}\left(1/\kappa^{20}\right)$ PAIRCOND *and* SAMP *queries.*

Once we have a reference point x we can use it to obtain an estimate $\hat{\psi}(i)$ for any i of our choice, using the procedure COMPARE, whose properties are stated in Lemma 4.1.2 (see Section 4.1.2.1).

Algorithm 26 Estimating the Distance to Uniformity

Require: PAIRCOND and SAMP query access to a distribution $\mathbf{p}$ and a parameter $\varepsilon \in [0,1]$.

Call the procedure FIND-REFERENCE (Section 4.1.6.1) with κ set to $\varepsilon/8$. If it returns No-Pair, then output $\hat{d} = 1$ as the estimate for the distance to uniformity. Otherwise, let $(x, \widehat{\mathbf{p}}(x))$ be its output.
Select a sample S of $\Theta(1/\varepsilon^2)$ points uniformly.
Let $K = \max\left\{\frac{2/n}{\widehat{\mathbf{p}}(x)}, \frac{\widehat{\mathbf{p}}(x)}{\varepsilon/(4n)}\right\}$.
for all point $y \in S$ **do**
    Call COMPARE$\left(\{x\}, \{y\}, \kappa, K, \frac{1}{10|S|}\right)$.
    if COMPARE returns high or it returns a value $\rho(y)$ such that $\rho(y) \cdot \widehat{\mathbf{p}}(x) \geq 1/n$ **then**
        set $\hat{\psi}(y) = 0$
    else if COMPARE returns low or it returns a value $\rho(y)$ such that $\rho(y) \cdot \widehat{\mathbf{p}}(x) \leq \varepsilon/4n$ **then**
        set $\hat{\psi}(y) = 1$;
    else
        set $\hat{\psi}(y) = 1 - n \cdot \rho(y) \cdot \widehat{\mathbf{p}}(x)$
    end if
end for
return $\hat{d} = \frac{1}{|S|}\sum_{y \in S} \hat{\psi}(y)$.

Theorem 4.1.41. *With probability at least* $2/3$*, the estimate* $\hat{d}$ *returned by Algorithm 26 satisfies* $\hat{d} = \mathrm{d_{TV}}(\mathbf{p}, \mathbf{u}) \pm \varepsilon$*. The number of queries performed by the algorithm is* $\tilde{O}(1/\varepsilon^{20})$.

Proof. In what follows we shall use the shorthand H_γ instead of $H_\gamma^{\mathbf{p}}$. Let E_0 denote the event that the procedure FIND-REFERENCE (Section 4.1.6.1) obeys the requirements in Lemma 4.1.40, where by Lemma 4.1.40 the event E_0 holds with probability at least $9/10$. Conditioned on E_0, the algorithm outputs $\hat{d} = 1$ right after calling the procedure (because the procedure returns No-Pair) only when $\mathbf{p}(H_\kappa) > 1-\kappa = 1-\varepsilon/8$. We claim that in this case $\mathrm{d_{TV}}(\mathbf{p}, \mathbf{u}) \geq 1 - 2\varepsilon/8 = 1 - \varepsilon/4$. To verify this, observe that

$$\mathrm{d_{TV}}(\mathbf{p}, \mathbf{u}) = \sum_{i:\mathbf{p}(i) > 1/n} \left(\mathbf{p}(i) - \frac{1}{n}\right) \geq \sum_{i \in H_\kappa} \left(\mathbf{p}(i) - \frac{1}{n}\right) = \mathbf{p}(H_\kappa) - \frac{|H_\kappa|}{n} \geq \mathbf{p}(H_\kappa) - \kappa\,. \tag{4.58}$$

Thus, in this case the estimate $\hat{d}$ is as required.

We turn to the case in which the procedure FIND-REFERENCE returns a pair $(x, \widehat{\mathbf{p}}(x))$ such that $\widehat{\mathbf{p}}(x) \in [1-2\kappa, 1+3\kappa]\mathbf{p}(x)$ and $\mathbf{p}(x) \in \left[\frac{\kappa}{8}, \frac{4}{\kappa}\right] \cdot \frac{1}{n}$.

We start by defining two more "desirable" events, which hold (simultaneously) with high constant probability, and then show that conditioned on these events holding (as well as E_0), the output of the algorithm

is as required. Let E_1 be the event that the sample S satisfies

$$\left| \frac{1}{|S|} \sum_{y \in S} \psi(y) - \mathrm{d_{TV}}(\mathbf{p}, \mathbf{u}) \right| \leq \varepsilon/2 \ . \tag{4.59}$$

By an additive Chernoff bound, the event E_1 holds with probability at least $9/10$.

Next, let E_2 be the event that all calls to the procedure COMPARE return answers as specified in Lemma 4.1.2. Since COMPARE is called $|S|$ times, and for each call the probability that it does not return an answer as specified in the lemma is at most $1/(10|S|)$, by the union bound the probability that E_2 holds is at least $9/10$.

From this point on assume events E_0, E_1 and E_2 all occur, which holds with probability at least $1-3/10 \geq 2/3$. Since E_2 holds, we get the following.

1. When COMPARE returns high for $y \in S$ (so that $\hat{\psi}(y)$ is set to 0) we have that

$$\mathbf{p}(y) > K \cdot \mathbf{p}(x) \geq \frac{2/n}{\widehat{\mathbf{p}}(x)} \cdot \mathbf{p}(x) > \frac{1}{n} \ , \tag{4.60}$$

   implying that $\hat{\psi}(y) = \psi(y)$.

2. When COMPARE returns low for $y \in S$ (so that $\hat{\psi}(y)$ is set to 1) we have that

$$\mathbf{p}(y) < \frac{\mathbf{p}(x)}{K} \leq \frac{\mathbf{p}(x)}{\widehat{\mathbf{p}}(x)/(\varepsilon/4n)} \leq \frac{\varepsilon}{2n} \ , \tag{4.61}$$

   implying that $\hat{\psi}(y) \leq \psi(y) + \varepsilon/2$ (and clearly $\psi(y) \leq \hat{\psi}(y)$).

3. When COMPARE returns a value $\rho(y)$ it holds that $\rho(y) \in [1-\kappa, 1+\kappa](\mathbf{p}(y)/\mathbf{p}(x))$, so that $\rho(y) \cdot \widehat{\mathbf{p}}(x) \in [(1-\kappa)(1-2\kappa), (1+\kappa)(1+3\kappa)]\mathbf{p}(y)$. Since $\kappa = \varepsilon/8$, if $\rho(y) \cdot \widehat{\mathbf{p}}(x) \geq 1/n$ (so that $\hat{\psi}(y)$ is set to 0), then $\psi(y) < \varepsilon/2$, if $\rho(y) \cdot \widehat{\mathbf{p}}(x) \leq \varepsilon/4n$ (so that $\hat{\psi}(y)$ is set to 1), then $\psi(y) \geq 1 - \varepsilon/2$, and otherwise $|\hat{\psi}(y) - \psi(y)| \leq \varepsilon/2$.

It follows that

$$\hat{d} = \frac{1}{|S|} \sum_{y \in S} \hat{\psi}(y) \in \left[\frac{1}{|S|} \sum_{y \in S} \psi(y) - \varepsilon/2, \frac{1}{|S|} \sum_{y \in S} \psi(y) + \varepsilon/2 \right] \subseteq [\mathrm{d_{TV}}(\mathbf{p}, \mathbf{u}) - \varepsilon, \mathrm{d_{TV}}(\mathbf{p}, \mathbf{u}) + \varepsilon] \tag{4.62}$$

as required.

The number of queries performed by the algorithm is the number of queries performed by the procedure FIND-REFERENCE, which is $\tilde{O}\big(1/\varepsilon^{20}\big)$, plus $\Theta(1/\varepsilon^2)$ times the number of queries performed in each call to COMPARE. The procedure COMPARE is called with the parameter K, which is bounded by $O(1/\varepsilon^2)$, the parameter η, which is $\Omega(\varepsilon)$, and δ, which is $\Omega(1/\varepsilon^2)$. By Lemma 4.1.2, the number of queries performed in each call to COMPARE is $O(\log(1/\varepsilon)/\varepsilon^4)$. The total number of queries performed is hence $\tilde{O}\big(1/\varepsilon^{20}\big)$. $\square$

4.1.6.1 Finding a reference point

In this subsection we prove Lemma 4.1.40. We start by giving the high-level idea behind the procedure. For a point $x \in [n]$ and $\gamma \in [0,1]$, let $U^{\mathbf{p}}_{\gamma}(x)$ be as defined in Eq. (4.1). Since $\mathbf{p}$ is fixed throughout this subsection, we shall use the shorthand $U_\gamma(x)$ instead of $U^{\mathbf{p}}_{\gamma}(x)$. Recall that κ is a parameter given to the procedure. Assume we had a point x for which $\mathbf{p}(U_\kappa(x)) \geq \kappa^{d_1}$ and $|U_\kappa(x)| \geq \kappa^{d_2} n$ for some constants d_1 and d_2 (so that necessarily $\mathbf{p}(x) = \Omega(\kappa^{d_1}/n)$ and $\mathbf{p}(x) = O(1/(\kappa^{d_2} n))$. It is not hard to verify (and we show this in detail subsequently), that if $\mathbf{p}(H) \leq 1-\kappa$, then a sample of size $\Theta(1/\operatorname{poly}(\kappa))$ distributed according to $\mathbf{p}$ will contain such a point x with high constant probability. Now suppose that we could obtain an estimate $\hat{w}$ of $\mathbf{p}(U_\kappa(x))$ such that $\hat{w} \in [1-\kappa, 1+\kappa]\mathbf{p}(U_\kappa(x))$ and an estimate $\hat{u}$ of $|U_\kappa(x)|$ such that $\hat{u} \in [1-\kappa, 1+\kappa]|U_\kappa(x)|$. By the definition of $U_\kappa(x)$ we have that $(\hat{w}/\hat{u}) \in [1-O(\kappa), 1+O(\kappa)]\mathbf{p}(x)$.

Obtaining good estimates of $\mathbf{p}(U_\kappa(x))$ and $|U_\kappa(x)|$ (for x such that both $|U_\kappa(x)|$ and $\mathbf{p}(U_\kappa(x))$ are sufficiently large) might be infeasible. This is due to the possible existence of many points y for which $\mathbf{p}(y)$ is very close to $(1+\kappa)\mathbf{p}(x)$ or $\mathbf{p}(x)/(1+\kappa)$ which define the boundaries of the set $U_\kappa(x)$. For such points it is not possible to efficiently distinguish between those among them that belong to $U_\kappa(x)$ (so that they are within the borders of the set) and those that do not belong to $U_\kappa(x)$ (so that they are just outside the borders of the set). However, for our purposes it suffices to estimate the weight and size of *some* set $U_\alpha(x)$ such that $\alpha \geq \kappa$ (so that $U_\kappa(x) \subseteq U_\alpha(x)$) and α is not much larger than κ (e.g., $\alpha \leq 2\kappa$). To this end we can apply Procedure ESTIMATE-NEIGHBORHOOD (see Section 4.1.2.2), which (conditioned on $\mathbf{p}(U_\kappa(x))$ being above a certain threshold), returns a pair $(\hat{w}(x), \alpha)$ such that $\hat{w}(x)$ is a good estimate of $\mathbf{p}(U_\alpha(x))$. Furthermore, α is such that for α' slightly larger than α, the weight of $U_{\alpha'}(x) \setminus U_\alpha(x)$ is small, allowing us to obtain also a good estimate $\hat{\mu}(x)$ of $|U_\alpha(x)|/n$.

Proof of Lemma 4.1.40. We first introduce the following notation.

$$L \stackrel{\text{def}}{=} \left\{ i : \ \mathbf{p}(i) < \frac{\kappa}{2n} \right\}, \ M \stackrel{\text{def}}{=} \left\{ i : \ \frac{\kappa}{2n} \leq \mathbf{p}(i) < \frac{1}{\kappa n} \right\} . \tag{4.63}$$

Let $H = H^{\mathbf{p}}_{\kappa}$ where $H^{\mathbf{p}}_{\kappa}$ is as defined in Eq. (4.57). Observe that $\mathbf{p}(L) < \kappa/2$, so that if $\mathbf{p}(H) \leq 1-\kappa$, then $\mathbf{p}(M) \geq \kappa/2$. Consider further partitioning the set M of "medium weight" points into buckets $M_1, \ldots, M_r$ where $r = \log_{1+\kappa}(2/\kappa^2) = \Theta(\log(1/\kappa)/\kappa)$ and the bucket M_j is defined as follows:

$$M_j \stackrel{\text{def}}{=} \left\{ i : \ (1+\kappa)^{j-1} \cdot \frac{\kappa}{2n} \leq \mathbf{p}(i) < (1+\kappa)^{j} \cdot \frac{\kappa}{2n} \right\} . \tag{4.64}$$

We consider the following "desirable" events.

1. Let E_1 be the event that conditioned on the existence of a bucket M_j such that $\mathbf{p}(M_j) \geq \kappa/2r = \Omega(\kappa^2/\log(1/\kappa))$, there exists a point $x^* \in X$ that belongs to M_j. By the setting of the size of the sample X, the (conditional) event E_1 holds with probability at least $1 - 1/40$.
2. Let E_2 be the event that all calls to ESTIMATE-NEIGHBORHOOD return an output as specified

Algorithm 27 Procedure FIND-REFERENCE

Require: PAIRCOND and SAMP query access to a distribution $\mathbf{p}$ and a parameter $\kappa \in (0, 1/4]$

1: Select a sample X of $\Theta(\log(1/\kappa)/\kappa^2)$ points distributed according to $\mathbf{p}$.
2: **for all** $x \in X$ **do**
3: Call ESTIMATE-NEIGHBORHOOD with the parameters κ as in the input to FIND-REFERENCE, $\beta = \kappa^2/(40\log(1/\kappa))$, $\eta = \kappa$, and $\delta = 1/(40|X|)$.
4: Let $\theta = \kappa\eta\beta\delta/64 = \Theta(\kappa^6/\log^2(1/\kappa))$ (as in FIND-REFERENCE).
5: **if** ESTIMATE-NEIGHBORHOOD returns a pair $(\hat{w}(x), \alpha(x))$ such that $\hat{w}(x) < \kappa^2/20\log(1/\kappa)$ **then** go to Line 2 and continue with the next $x \in X$.
6: **end if**
7: Select a sample Y_x of size $\Theta(\log^2(1/\kappa)/\kappa^5)$ distributed uniformly.
8: **for all** $y \in Y_x$ **do**
9: call COMPARE$(\{x\}, \{y\}, \theta/4, 4, 1/40|X||Y_x|)$, and let the output be denoted $\rho_x(y)$
10: **end for**
11: Let $\hat{\mu}(x)$ be the fraction of occurrences of $y \in Y_x$ such that $\rho_x(y) \in [1/(1+\alpha+\theta/2), 1+\alpha+\theta/2]$.
12: Set $\widehat{\mathbf{p}}(x) = \hat{w}(x)/(\hat{\mu}(x)n)$.
13: **end for**
14: **if** for some point $x \in X$ we have $\hat{w}(x) \geq \kappa^2/20\log(1/\kappa)$, $\hat{\mu}(x) \geq \kappa^3/20\log(1/\kappa)$, and $\kappa/4n \leq \widehat{\mathbf{p}}(x) \leq 2/(\kappa n)$, **then**
15: **return** $(x, \widehat{\mathbf{p}}(x))$
16: **else**
17: **return** No-Pair
18: **end if**

by Lemma 4.1.3. By Lemma 4.1.3, the setting of the confidence parameter δ in each call and a union bound over all $|X|$ calls, E_2 holds with probability at least $1 - 1/40$.

3. Let E_3 be the event that for each $x \in X$ we have the following.

   a) If $\frac{|U_{\alpha(x)}(x)|}{n} \geq \frac{\kappa^3}{40\log(1/\kappa)}$, then $\frac{|Y_x \cap U_{\alpha(x)}(x)|}{|Y_x|} \in [1-\eta/2, 1+\eta/2]\frac{|U_{\alpha(x)}(x)|}{n}$;
   If $\frac{|U_{\alpha(x)}(x)|}{n} < \frac{\kappa^3}{40\log(1/\kappa)}$, then $\frac{|Y_x \cap U_{\alpha(x)}(x)|}{|Y_x|} < \frac{\kappa^3}{30\log(1/\kappa)}$;

   b) Let $\Delta_{\alpha(x),\theta}(x) \stackrel{\text{def}}{=} U_{\alpha(x)+\theta}(x) \setminus U_{\alpha(x)}(x)$ (where θ is as specified by the algorithm).
   If $\frac{|\Delta_{\alpha(x),\theta}(x)|}{n} \geq \frac{\kappa^4}{240\log(1/\kappa)}$, then $\frac{|Y_x \cap \Delta_{\alpha(x),\theta}(x)|}{|Y_x|} \leq 2 \cdot \frac{|\Delta_{\alpha(x),\theta}(x)|}{n}$;
   If $\frac{|\Delta_{\alpha(x),\theta}(x)|}{n} < \frac{\kappa^4}{240\log(1/\kappa)}$, then $\frac{|Y_x \cap \Delta_{\alpha(x),\theta}(x)|}{|Y_x|} < \frac{\kappa^4}{120\log(1/\kappa)}$.

   By the size of each set Y_x and a union bound over all $x \in X$, the event E_3 holds with probability at least $1 - 1/40$.

4. Let E_4 be the event that all calls to COMPARE return an output as specified by Lemma 4.1.2. By Lemma 4.1.2, the setting of the confidence parameter δ in each call and a union bound over all (at most) $|X| \cdot |Y|$ calls, E_3 holds with probability at least $1 - 1/40$.

Assuming events E_1–E_4 all hold (which occurs with probability at least $9/10$) we have the following.

1. By E_2, for each $x \in X$ such that $\hat{w}(x) \geq \kappa^2/20\log(1/\kappa)$ (so that x may be selected for the output of the procedure) we have that $\mathbf{p}(U_{\alpha(x)}(x)) \geq \kappa^2/40\log(1/\kappa)$.

   The event E_2 also implies that for each $x \in X$ we have that $\mathbf{p}(\Delta_{\alpha(x),\theta}(x)) \leq \eta\beta/16 \leq (\eta/16) \cdot$

$\mathbf{p}(U_{\alpha(x)}(x))$, so that

$$\frac{|\Delta_{\alpha(x),\theta}(x)|}{n} \leq \frac{\eta(1+\alpha(x))(1+\alpha(x)+\theta)}{16} \cdot \frac{|U_{\alpha(x)}(x)|}{n} \leq \frac{\eta}{6} \cdot \frac{|U_{\alpha(x)}(x)|}{n} \,. \tag{4.65}$$

2. Consider any $x \in X$ such that $\hat{w}(x) \geq \kappa^2/20\log(1/\kappa)$. Let $T_x \stackrel{\text{def}}{=} \{\, y \in Y_x : \rho_x(y) \in [1/(1+\alpha+\theta/2), (1+\alpha+\theta/2]\,\}$, so that $\hat{\mu}(x) = |T_x|/|Y_x|$. By E_4, for each $y \in Y_x \cap U_{\alpha(x)}(x)$ we have that $\rho_x(y) \leq (1+\alpha)(1+\theta/4) \leq (1+\alpha+\theta/2)$ and $\rho_x(y) \geq (1+\alpha)^{-1}(1-\theta/4) \geq (1+\alpha+\theta/2)^{-1}$, so that $y \in T_x$. On the other hand, for each $y \notin Y_x \cap U_{\alpha(x)+\theta}(x)$ we have that $\rho_x(y) > (1+\alpha+\theta)(1-\theta/4) \geq 1+\alpha+\theta/2$ or $\rho_x(y) < (1+\alpha+\theta)^{-1}(1-\theta/4) < (1+\alpha+\theta/2)^{-1}$, so that $y \notin T_x$. It follows that

$$Y_x \cap U_{\alpha(x)}(x) \;\subseteq\; T_x \;\subseteq\; Y_x \cap (U_{\alpha(x)}(x) \cup \Delta_{\alpha(x),\theta}(x)) \,. \tag{4.66}$$

   By E_3, when $\hat{\mu}(x) = |T_x|/|Y_x| \geq \kappa^3/20\log(1/\kappa)$, then necessarily $\hat{\mu}(x) \in [1-\eta, 1+\eta]|U_{\alpha(x)}(x)|/n$. To verify this consider the following cases.

   a) If $\frac{|U_{\alpha(x)}(x)|}{n} \geq \frac{\kappa^3}{40\log(1/\kappa)}$, then (by the left-hand-side of Eq. (4.66)) and the definition of E_3) we get that $\hat{\mu}(x) \geq (1-\eta/2)\frac{|U_{\alpha(x)}(x)|}{n}$, and (by the right-hand-side of Eq. (4.66), Eq. (4.65), and E_3) we get that $\hat{\mu}(x) \leq (1+\eta/2)\frac{|U_{\alpha(x)}(x)|}{n} + 2(\eta/6)\frac{|U_{\alpha(x)}(x)|}{n} < (1+\eta)\frac{|U_{\alpha(x)}(x)|}{n}$.

   b) If $\frac{|U_{\alpha(x)}(x)|}{n} < \frac{\kappa^3}{40\log(1/\kappa)}$, then (by the right-hand-side of Eq. (4.66), Eq. (4.65), and E_3) we get that $\hat{\mu}(x) < \frac{\kappa^3}{30\log(1/\kappa)} + \frac{\kappa^4}{120\log(1/\kappa)} < \kappa^3/20\log(1/\kappa)$.

3. If $\mathbf{p}(H) \leq 1-\kappa$, so that $\mathbf{p}(M) \geq \kappa/2$, then there exists at least one bucket M_j such that $\mathbf{p}(M_j) \geq \kappa/2r = \Omega(\kappa^2/\log(1/\kappa))$. By E_1, the sample X contains a point $x^* \in M_j$. By the definition of the buckets, for this point x^* we have that $\mathbf{p}(U_\kappa(x^*)) \geq \kappa/2r \geq \kappa^2/(10\log(1/\kappa))$ and $|U_\kappa(x^*)| \geq (\kappa^2/2r)n \geq \kappa^3/(10\log(1/\kappa))n$.

By the first two items above and the setting $\eta = \kappa$ we have that for each x such that $\hat{w}(x) \geq \kappa^2/20\log(1/\kappa)$ and $\hat{\mu}(x) \geq \kappa^3/20\log(1/\kappa)$,

$$\widehat{\mathbf{p}}(x) \in \left[\frac{1-\kappa}{1+\kappa}, \frac{1+\kappa}{1-\kappa}\right] \mathbf{p}(x) \subset [1-2\kappa, 1+3\kappa]\mathbf{p}(x) \,.$$

Thus, if the algorithm outputs a pair $(x, \widehat{\mathbf{p}}(x))$ then it satisfies the condition stated in both items of the lemma. This establishes the second item in the lemma. By combining all three items we get that if $\mathbf{p}(H) \geq 1-\kappa$ then the algorithm outputs a pair $(x, \widehat{\mathbf{p}}(x))$ (where possibly, but not necessarily, $x = x^*$), and the first item is established as well.

Turning to the query complexity, the total number of PAIRCOND queries performed in the $|X| = O(\log(1/\kappa)/\kappa^2)$ calls to ESTIMATE-NEIGHBORHOOD is $O\Big(\frac{|X|\log(1/\delta)^2\log(1/(\beta\eta))}{\kappa^2\eta^4\beta^3\delta^2}\Big) = \tilde{O}\big(1/\kappa^{18}\big)$, and the total number of PAIRCOND queries performed in the calls to COMPARE (for at most all pairs $x \in X$ and $y \in Y_x$) is $\tilde{O}\big(1/\kappa^{20}\big)$. □

4.1.7 A $\tilde{O}\big((\log^3 n)/\varepsilon^3\big)$-query $\mathsf{INTCOND}_{\mathbf{p}}$ algorithm for testing uniformity

In this and the next section we consider INTCOND algorithms for testing whether an unknown distribution $\mathbf{p}$ over $[n]$ is the uniform distribution versus ε-far from uniform. Our results show that INTCOND algorithms are not as powerful as PAIRCOND algorithms for this basic testing problem; in this section we give a $\mathrm{poly}(\log n, 1/\varepsilon)$-query $\mathsf{INTCOND}_{\mathbf{p}}$ algorithm, and in the next section we prove that any $\mathsf{INTCOND}_{\mathbf{p}}$ algorithm must make $\tilde{\Omega}(\log n)$ queries.

In more detail, in this section we describe an algorithm $\mathsf{INTCOND}_{\mathbf{p}}$-TEST-UNIFORM and prove the following theorem:

Theorem 4.1.42. $\mathsf{INTCOND}_{\mathbf{p}}$-TEST-UNIFORM *is a* $\tilde{O}\big(\frac{\log^3 n}{\varepsilon^3}\big)$*-query* $\mathsf{INTCOND}_{\mathbf{p}}$ *testing algorithm for uniformity, i.e. it outputs* accept *with probability at least* $2/3$ *if* $\mathbf{p} = \mathbf{u}$ *and outputs* reject *with probability at least* $2/3$ *if* $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{u}) \geq \varepsilon$.

Intuition. Recall that, as mentioned in Section 4.1.3.1, any distribution $\mathbf{p}$ which is ε-far from uniform must put $\Omega(\varepsilon)$ probability mass on "significantly heavy" elements (that is, if we define $H' = \big\{\, h \in [n] : \ \mathbf{p}(h) \geq \frac{1}{n} + \frac{\varepsilon}{4n} \,\big\}$, it must hold that $\mathbf{p}(H') \geq \varepsilon/4$). Consequently a sample of $O(1/\varepsilon)$ points drawn from $\mathbf{p}$ will contain such a point with high probability. Thus, a natural approach to testing whether $\mathbf{p}$ is uniform is to devise a procedure that, given an input point y, can distinguish between the case that $y \in H'$ and the case that $\mathbf{p}(y) = 1/n$ (as it is when $\mathbf{p} = \mathbf{u}$).

We give such a procedure, which uses the $\mathsf{INTCOND}_{\mathbf{p}}$ oracle to perform a sort of binary search over intervals. The procedure successively "weighs" narrower and narrower intervals until it converges on the single point y. In more detail, we consider the *interval tree* whose root is the whole domain $[n]$, with two children $\{1, \ldots, n/2\}$ and $\{n/2+1, \ldots, n\}$, and so on, with a single point at each of the n leaves. Our algorithm starts at the root of the tree and goes down the path that corresponds to y; at each child node it uses COMPARE to compare the weight of the current node to the weight of its sibling under $\mathbf{p}$. If at any point the estimate deviates significantly from the value it should have if $\mathbf{p}$ were uniform (namely the weights should be essentially equal, with slight deviations because of even/odd issues), then the algorithm rejects. Assuming the algorithm does not reject, it provides a $(1 \pm O(\varepsilon))$-accurate multiplicative estimate of $\mathbf{p}(y)$, and the algorithm checks whether this estimate is sufficiently close to $1/n$ (rejecting if this is not the case). If no point in a sample of $\Theta(1/\varepsilon)$ points (drawn according to $\mathbf{p}$) causes rejection, then the algorithm accepts.

The algorithm we use to perform the "binary search" described above is Algorithm 28, BINARY-DESCENT. We begin by proving correctness for it:

Lemma 4.1.43. *Suppose the algorithm* BINARY-DESCENT *is run with inputs* $\varepsilon \in (0, 1]$, $a = 1$, $b = n$, *and* $y \in [n]$, *and is provided* INTCOND *oracle access to distribution* $\mathbf{p}$ *over* $[n]$. *It performs* $\tilde{O}\big(\log^3 n/\varepsilon^2\big)$ *queries and either outputs a value* $\hat{\mathbf{p}}(y)$ *or* reject, *where the following holds:*

1. *if* $\mathbf{p}(y) \geq \frac{1}{n} + \frac{\varepsilon}{4n}$, *then with probability at least* $1 - \frac{\varepsilon}{100}$ *the procedure either outputs a value* $\hat{\mathbf{p}}(y) \in [1 - \varepsilon/12, 1 + \varepsilon/12]\mathbf{p}(y)$ *or* reject;

Algorithm 28 BINARY-DESCENT

Require: parameter $\varepsilon > 0$; integers $1 \le a \le b \le n$; $y \in [a, b]$; query access to $\mathsf{INTCOND}_{\mathbf{p}}$ oracle

1: **if** $a = b$ **then return** 1
2: **end if**
3: Let $c = \lfloor \frac{a+b}{2} \rfloor$; $\Delta = (b - a + 1)/2$.
4: **if** $y \le c$ **then**
5: Define $I_y = \{a, \ldots, c\}$, $I_{\bar{y}} = \{c + 1, \ldots, b\}$ and $\rho = \lceil \Delta \rceil / \lfloor \Delta \rfloor$
6: **else**
7: Define $I_{\bar{y}} = \{a, \ldots, c\}$, $I_y = \{c + 1, \ldots, b\}$ and $\rho = \lfloor \Delta \rfloor / \lceil \Delta \rceil$
8: **end if**
9: Call COMPARE on I_y, $I_{\bar{y}}$ with parameters $\eta = \frac{\varepsilon}{48 \log n}$, $K = 2$, $\delta = \frac{\varepsilon}{100(1+\log n)}$ to get an estimate $\hat{\rho}$ of $\mathbf{p}(I_y)/\mathbf{p}(I_{\bar{y}})$
10: **if** $\hat{\rho} \notin [1 - \frac{\varepsilon}{48 \log n}, 1 + \frac{\varepsilon}{48 \log n}] \cdot \rho$ (this includes the case that $\hat{\rho}$ is high or low) **then return** reject
11: **end if**
12: Call recursively BINARY-DESCENT on input (ε, the endpoints of I_y, y);
13: **if** BINARY-DESCENT returns a value ν **then return** $\frac{\hat{\rho}}{1+\hat{\rho}} \cdot \nu$
14: **elsereturn** reject
15: **end if**

Algorithm 29 $\mathsf{INTCOND}_{\mathbf{p}}$-TEST-UNIFORM

Require: error parameter $\varepsilon > 0$; query access to $\mathsf{INTCOND}_{\mathbf{p}}$ oracle

1: Draw $t = \frac{20}{\varepsilon}$ points $y_1, \ldots, y_t$ from $\mathsf{SAMP}_{\mathbf{p}}$.
2: **for** $j = 1$ to t **do**
3: Call BINARY-DESCENT$(\varepsilon, 1, n, y_j)$ and return reject if it rejects, otherwise let $\hat{d}_j$ be the value it returns as its estimate of $\mathbf{p}(y_j)$
4: **if** $\hat{d}_j \notin [1 - \frac{\varepsilon}{12}, 1 + \frac{\varepsilon}{12}] \cdot \frac{1}{n}$ **then return** reject
5: **end if**
6: **end forreturn** accept

2. *if* $\mathbf{p} = \mathbf{u}$*, then with probability at least* $1 - \frac{\varepsilon}{100}$ *the procedure outputs a value* $\hat{\mathbf{p}}(y) \in [1 - \varepsilon/12, 1 + \varepsilon/12] \cdot \frac{1}{n}$.

Proof of Lemma 4.1.43. The claimed query bound is easily verified, since the recursion depth is at most $1 + \log n$ and the only queries made are during calls to COMPARE, each of which performs $O(\log(1/\delta)/\eta^2) = \tilde{O}\big(\log^2 n/\varepsilon^2\big)$ queries.

Let E_0 be the event that all calls to COMPARE satisfy the conditions in Lemma 4.1.2; since each of them succeeds with probability at least $1 - \delta = 1 - \frac{\varepsilon}{100(1+\log n)}$, a union bound shows that E_0 holds with probability at least $1 - \varepsilon/100$. We hereafter condition on E_0.

We first prove the second part of the lemma where $\mathbf{p} = \mathbf{u}$. Fix any specific recursive call, say the j-th, during the execution of the procedure. The intervals $I_y^{(j)}, I_{\bar{y}}^{(j)}$ used in that execution of the algorithm are easily seen to satisfy $\mathbf{p}(I_y)/\mathbf{p}(I_{\bar{y}}) \in [1/K, K]$ (for $K = 2$), so by event E_0 it must be the case that COMPARE returns an estimate $\hat{\rho}_j \in [1 - \frac{\varepsilon}{48 \log n}, 1 + \frac{\varepsilon}{48 \log n}] \cdot \mathbf{p}(I_y^{(j)})/\mathbf{p}(I_{\bar{y}}^{(j)})$. Since $\mathbf{p} = U$, we have that $\mathbf{p}(I_y^{(j)})/\mathbf{p}(I_{\bar{y}}^{(j)}) = \rho^{(j)}$, so the overall procedure returns a numerical value rather than reject.

Let $M = \lceil \log n \rceil$ be the number of recursive calls (i.e., the number of executions of Line 12). Note that

we can write $\mathbf{p}(y)$ as a product

$$\mathbf{p}(y) = \prod_{j=1}^{M} \frac{\mathbf{p}(I_y^{(j)})}{\mathbf{p}(I_y^{(j)}) + \mathbf{p}(I_{\bar{y}}^{(j)})} = \prod_{j=1}^{M} \frac{\mathbf{p}(I_y^{(j)})/\mathbf{p}(I_{\bar{y}}^{(j)})}{\mathbf{p}(I_y^{(j)})/\mathbf{p}(I_{\bar{y}}^{(j)}) + 1} \,. \tag{4.67}$$

We next observe that for any $0 \leq \varepsilon' < 1$ and $\rho, d > 0$, if $\hat{\rho} \in [1-\varepsilon', 1+\varepsilon']d$ then we have $\frac{\hat{\rho}}{\hat{\rho}+1} \in [1-\frac{\varepsilon'}{2}, 1+\varepsilon']\frac{d}{d+1}$ (by straightforward algebra). Applying this M times, we get

$$\begin{aligned}
\prod_{j=1}^{M} \frac{\hat{\rho}_j}{\hat{\rho}_j + 1} &\in \left[\left(1 - \frac{\varepsilon}{96\log n}\right)^M, \left(1 + \frac{\varepsilon}{48\log n}\right)^M\right] \cdot \prod_{j=1}^{M} \frac{\mathbf{p}(I_y^{(j)})/\mathbf{p}(I_{\bar{y}}^{(j)})}{\mathbf{p}(I_y^{(j)})/\mathbf{p}(I_{\bar{y}}^{(j)}) + 1} \\
&\in \left[\left(1 - \frac{\varepsilon}{96\log n}\right)^M, \left(1 + \frac{\varepsilon}{48\log n}\right)^M\right] \cdot \mathbf{p}(y) \\
&\in \left[1 - \frac{\varepsilon}{12}, 1 + \frac{\varepsilon}{12}\right] \mathbf{p}(y).
\end{aligned}$$

Since $\prod_{j=1}^{M} \frac{\hat{\rho}_j}{\hat{\rho}_j+1}$ is the value that the procedure outputs, the second part of the lemma is proved.

The proof of the first part of the lemma is virtually identical. The only difference is that now it is possible that COMPARE outputs high or low at some call (since $\mathbf{p}$ is not uniform it need not be the case that $\mathbf{p}(I_y^{(j)})/\mathbf{p}(I_{\bar{y}}^{(j)}) = \rho^{(j)}$), but this is not a problem for (i) since in that case BINARY-DESCENT would output reject. □

See Algorithm 28 for a description of the testing algorithm $\mathsf{INTCOND}_{\mathbf{p}}$-TEST-UNIFORM. We now prove Theorem 4.1.42:

Proof of Theorem 4.1.42. Define E_1 to be the event that all calls to BINARY-DESCENT satisfy the conclusions of Lemma 4.1.43. With a union bound over all these $t = 20/\varepsilon$ calls, we have $\Pr[E_1] \geq 8/10$.

Completeness: Suppose $\mathbf{p} = \mathbf{u}$, and condition again on E_1. Since this implies that BINARY-DESCENT will always return a value, the only case $\mathsf{INTCOND}_{\mathbf{p}}$-TEST-UNIFORM might reject is by reaching Line 4. However, since it is the case that every value $\hat{d}_j$ returned by the procedure satisfies $\hat{\mathbf{p}}(y) \in [1-\varepsilon/12, 1+\varepsilon/12] \cdot \frac{1}{n}$, this can never happen.

Soundness: Suppose $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{u}) \geq \varepsilon$. Let E_2 be the event that at least one of the y_i's drawn in Line 1 belongs to H'. As $\mathbf{p}(H') \geq \varepsilon/4$, we have $\Pr[E_2] \geq 1 - (1-\varepsilon/4)^{20/\varepsilon} \geq 9/10$. Conditioning on both E_1 and E_2, for such a y_j, one of two cases below holds:

- either the call to BINARY-DESCENT outputs reject and $\mathsf{INTCOND}_{\mathbf{p}}$-TEST-UNIFORM outputs reject;
- or a value $\hat{d}_j$ is returned, for which $\hat{d}_j \geq (1-\frac{\varepsilon}{12})(1+\frac{\varepsilon}{4}) \cdot \frac{1}{n} > (1+\varepsilon/12)/n$ (where we used the fact that E_1 holds); and $\mathsf{INTCOND}_{\mathbf{p}}$-TEST-UNIFORM reaches Line 4 and rejects.

Since $\Pr[E_1 \cup E_2] \geq 7/10$, $\mathsf{INTCOND}_{\mathbf{p}}$-TEST-UNIFORM is correct with probability at least $2/3$. Finally, the claimed query complexity directly follows from the $t = \Theta(1/\varepsilon)$ calls to BINARY-DESCENT, each of which makes $\tilde{O}\big(\log^3 n/\varepsilon^2\big)$ queries to $\mathsf{INTCOND}_{\mathbf{p}}$. □

4.1.8 An $\Omega(\log n/\log\log n)$ lower bound for $\mathsf{INTCOND}_{\mathbf{p}}$ algorithms that test uniformity

In this section we prove that any $\mathsf{INTCOND}_{\mathbf{p}}$ algorithm that ε-tests uniformity even for constant ε must have query complexity $\tilde{\Omega}(\log n)$. This shows that our algorithm in the previous subsection is not too far from optimal, and sheds light on a key difference between INTCOND and PAIRCOND oracles.

Theorem 4.1.44. *Fix $\varepsilon = 1/3$. Any $\mathsf{INTCOND}_{\mathbf{p}}$ algorithm for testing whether $\mathbf{p} = \mathbf{u}$ versus $\mathrm{d_{TV}}(\mathbf{p}, \mathbf{p}^*) \geq \varepsilon$ must make $\Omega\Big(\frac{\log n}{\log\log n}\Big)$ queries.*

To prove this lower bound we define a probability distribution $\mathcal{D}_{\mathsf{no}}$ over possible no-distributions (i.e. distributions that have variation distance at least $1/3$ from $\mathbf{u}$). A distribution drawn from $\mathcal{D}_{\mathsf{no}}$ is constructed as follows: first (assuming without loss of generality that n is a power of 2), we partition $[n]$ into $b = 2^X$ consecutive intervals of the same size $\Delta = \frac{n}{2^X}$, which we refer to as "blocks", where X is a random variable distributed uniformly on the set $\{\frac{1}{3}\log n, \frac{1}{3}\log n + 1, \ldots, \frac{2}{3}\log n\}$. Once the block size Δ is determined, a random offset y is drawn uniformly at random in $[n]$, and all block endpoints are shifted by y modulo $[n]$ (intuitively, this prevents the testing algorithm from "knowing" a priori that specific points are endpoints of blocks). Finally, independently for each block, a fair coin is thrown to determine its *profile*: with probability $1/2$, each point in the first half of the block will have probability weight $\frac{1-2\varepsilon}{n}$ and each point in the second half will have probability $\frac{1+2\varepsilon}{n}$ (such a block is said to be a *low-high* block, with profile $\downarrow\uparrow$). With probability $1/2$ the reverse is true: each point in the first half has probability $\frac{1+2\varepsilon}{n}$ and each point in the second half has probability $\frac{1-2\varepsilon}{n}$ (a *high-low* block $\uparrow\downarrow$). It is clear that each distribution $\mathbf{p}$ in the support of $\mathcal{D}_{\mathsf{no}}$ defined in this way indeed has $\mathrm{d_{TV}}(\mathbf{p}, \mathbf{u}) = \varepsilon$.

To summarize, each no-distribution $\mathbf{p}$ in the support of $\mathcal{D}_{\mathsf{no}}$ is parameterized by $(b+2)$ parameters: its block size Δ, offset y, and profile $\vartheta \in \{\downarrow\uparrow, \uparrow\downarrow\}^b$. Note that regardless of the profile vector, each block always has weight exactly Δ/n.

We note that while there is only one yes-distribution $\mathbf{u}$, it will sometimes be convenient for the analysis to think of $\mathbf{u}$ as resulting from the same initial process of picking a block size and offset, but without the subsequent choice of a profile vector. We sometimes refer to this as the "fake construction" of the uniform distribution $\mathbf{u}$ (the reason for this will be clear later).

The proof of Theorem 4.1.44 will be carried out in two steps. First we shall restrict the analysis to *non-adaptive algorithms*, and prove the lower bound for such algorithms. This result will then be extended to the general setting by introducing (similarly to Section 4.1.4.2) the notion of a *query-faking algorithm*, and reducing the behavior of adaptive algorithms to non-adaptive ones through an appropriate sequence of such query-faking algorithms.

Before proceeding, we define the *transcript* of the interaction between an algorithm and a $\mathsf{INTCOND}_{\mathbf{p}}$

oracle. Informally, the transcript captures the entire history of interaction between the algorithm and the $\mathsf{INTCOND}_{\mathbf{p}}$ oracle during the whole sequence of queries.

Definition 4.1.45. Fix any (possibly adaptive) testing algorithm $\mathcal{A}$ that queries an $\mathsf{INTCOND}_{\mathbf{p}}$ oracle. The *transcript* of $\mathcal{A}$ is a sequence $\mathcal{T} = (I_\ell, s_\ell)_{\ell \in \mathbb{N}^*}$ of pairs, where I_ℓ is the ℓ-th interval provided by the algorithm as input to $\mathsf{INTCOND}_{\mathbf{p}}$, and $s_\ell \in I_\ell$ is the response that $\mathsf{INTCOND}_{\mathbf{p}}$ provides to this query. Given a transcript $\mathcal{T}$, we shall denote by $\mathcal{T}|_k$ the partial transcript induced by the first k queries, i.e. $\mathcal{T}|_k = (I_\ell, s_\ell)_{1 \leq \ell \leq k}$.

Equipped with these definitions, we now turn to proving the theorem in the special case of non-adaptive testing algorithms. Observe that there are three different sources of randomness in our arguments: (i) the draw of the no-instance from $\mathcal{D}_{\mathsf{no}}$, (ii) the internal randomness of the testing algorithm; and (iii) the random draws from the oracle. Whenever there could be confusion we shall explicitly state which probability space is under discussion.

4.1.8.1 A lower bound against non-adaptive algorithms

Throughout this subsection we assume that $\mathcal{A}$ is an arbitrary, fixed, non-adaptive, randomized algorithm that makes exactly $q \leq \tau \cdot \frac{\log n}{\log \log n}$ queries to $\mathsf{INTCOND}_{\mathbf{p}}$; here $\tau \in (0,1)$ is some absolute constant that will be determined in the course of the analysis. (The assumption that $\mathcal{A}$ always makes exactly q queries is without loss of generality since if in some execution the algorithm makes $q' < q$ queries, it can perform additional "dummy" queries). In this setting algorithm $\mathcal{A}$ corresponds to a distribution P_A over q-tuples $\bar{I} = (I_1, \ldots, I_q)$ of query intervals. The following theorem will directly imply Theorem 4.1.44 in the case of non-adaptive algorithms:

Theorem 4.1.46.

$$\left| \Pr_{\mathbf{p} \sim \mathcal{D}_{no}} [\mathcal{A}^{\mathsf{INTCOND}_{\mathbf{p}}} \textit{ outputs } \textsf{accept}] - \Pr[\mathcal{A}^{\mathsf{INTCOND}_{\mathbf{u}}} \textit{ outputs } \textsf{accept}] \right| \leq 1/5. \tag{4.68}$$

Observe that in the first probability of Equation (4.68) the randomness is taken over the draw of $\mathbf{p}$ from $\mathcal{D}_{\mathsf{no}}$, the draw of $\bar{I} \sim P_A$ that $\mathcal{A}$ performs to select its sequence of query intervals, and the randomness of the $\mathsf{INTCOND}_{\mathbf{p}}$ oracle. In the second one the randomness is just over the draw of $\bar{I}$ from P_A and the randomness of the $\mathsf{INTCOND}_{\mathbf{u}}$ oracle.

Intuition for Theorem 4.1.46. The high-level idea is that the algorithm will not be able to distinguish between the uniform distribution and a no-distribution unless it manages to learn something about the "structure" of the blocks in the no-case, either by guessing (roughly) the right block size, or by guessing (roughly) the location of a block endpoint and querying a short interval containing such an endpoint.

In more detail, we define the following "bad events" (over the choice of $\mathbf{p}$ and the points s_i) for a fixed sequence $\bar{I} = (I_1, \ldots, I_q)$ of queries (the dependence on $\bar{I}$ is omitted in the notation for the sake of

readability):

$$
\begin{aligned}
B^{\mathrm{n}}_{\mathrm{size}} &= \{\ \exists \ell \in [q]\ :\ \Delta/\log n \le |I_\ell| \le \Delta \cdot (\log n)^2\ \} \\
B^{\mathrm{n}}_{\mathrm{boundary}} &= \{\ \exists \ell \in [q]\ :\ |I_\ell| < \Delta/\log n \text{ and } I_\ell \text{ intersects two blocks}\ \} \\
B^{\mathrm{n}}_{\mathrm{middle}} &= \{\ \exists \ell \in [q]\ :\ |I_\ell| < \Delta/\log n \text{ and } I_\ell \text{ intersects both halves of the same block}\ \} \\
B^{\mathrm{n}}_{\ell,\mathrm{outer}} &= \{\Delta \cdot (\log n)^2 < |I_\ell| \text{ and } s_\ell \text{ belongs to a block not contained entirely in } I_\ell\} \qquad \ell \in [q] \\
B^{\mathrm{n}}_{\ell,\mathrm{collide}} &= \{\Delta \cdot (\log n)^2 < |I_\ell| \text{ and } \exists j < \ell,\ s_\ell \text{ and } s_j \text{ belong to the same block}\} \qquad \ell \in [q]
\end{aligned}
$$

The first three events depend only on the draw of $\mathbf{p}$ from $\mathcal{D}_{\mathsf{no}}$, which determines Δ and y, while the last $2q$ events also depend on the random draws of s_ℓ from the $\mathsf{INTCOND}_{\mathbf{p}}$ oracle. We define in the same fashion the corresponding bad events for the yes-instance (i.e. the uniform distribution $\mathbf{u}$) $B^{\mathrm{Y}}_{\mathrm{size}}$, $B^{\mathrm{Y}}_{\mathrm{boundary}}$, $B^{\mathrm{Y}}_{\mathrm{middle}}$, $B^{\mathrm{Y}}_{\ell,\mathrm{outer}}$ and $B^{\mathrm{Y}}_{\ell,\mathrm{collide}}$, using the notion of the "fake construction" of $\mathbf{u}$ mentioned above.

Events $B^{\mathrm{n}}_{\mathrm{size}}$ and $B^{\mathrm{Y}}_{\mathrm{size}}$ correspond to the possibility, mentioned above, that algorithm $\mathcal{A}$ "guesses" essentially the right block size, and events $B^{\mathrm{n}}_{\mathrm{boundary}}$, $B^{\mathrm{Y}}_{\mathrm{boundary}}$ and $B^{\mathrm{n}}_{\mathrm{middle}}$, $B^{\mathrm{Y}}_{\mathrm{middle}}$ correspond to the possibility that algorithm $\mathcal{A}$ "guesses" a short interval containing respectively a block endpoint or a block midpoint. The final bad events correspond to $\mathcal{A}$ guessing a "too-large" block size but "getting lucky" with the sample returned by $\mathsf{INTCOND}$, either because the sample belongs to one of the (at most two) outer blocks not entirely contained in the query interval, or because $\mathcal{A}$ has already received a sample from the same block as the current sample.

We can now describe the *failure events* for both the uniform distribution and for a no-distribution as the union of the corresponding bad events:

$$
\begin{aligned}
B^{\mathrm{n}}_{(\bar{I})} &= B^{\mathrm{n}}_{\mathrm{size}} \cup B^{\mathrm{n}}_{\mathrm{boundary}} \cup B^{\mathrm{n}}_{\mathrm{middle}} \cup \left(\bigcup_{\ell=1}^{q} B^{\mathrm{n}}_{\ell,\mathrm{outer}}\right) \cup \left(\bigcup_{\ell=1}^{q} B^{\mathrm{n}}_{\ell,\mathrm{collide}}\right) \\
B^{\mathrm{Y}}_{(\bar{I})} &= B^{\mathrm{Y}}_{\mathrm{size}} \cup B^{\mathrm{Y}}_{\mathrm{boundary}} \cup B^{\mathrm{Y}}_{\mathrm{middle}} \cup \left(\bigcup_{\ell=1}^{q} B^{\mathrm{Y}}_{\ell,\mathrm{outer}}\right) \cup \left(\bigcup_{\ell=1}^{q} B^{\mathrm{Y}}_{\ell,\mathrm{collide}}\right)
\end{aligned}
$$

These failure events can be interpreted, from the point of view of the algorithm $\mathcal{A}$, as the "opportunity to potentially learn something;" we shall argue below that if the failure events do not occur then the algorithm gains no information about whether it is interacting with the uniform distribution or with a no-distribution.

Structure of the proof of Theorem 4.1.46. First, observe that since the transcript is the result of the interaction of the algorithm and the oracle on a randomly chosen distribution, it is itself a random variable; we will be interested in the distribution over this random variable induced by the draws from the oracle and the choice of $\mathbf{p}$. More precisely, for a fixed sequence of query sets $\bar{I}$, let $Z^{\mathrm{n}}_{\bar{I}}$ denote the random variable over no-transcripts generated when $\mathbf{p}$ is drawn from $\mathcal{D}_{\mathsf{no}}$. Note that this is a random variable over the probability space defined by the random draw of $\mathbf{p}$ and the draws of s_i by $\mathsf{INTCOND}_{\mathbf{p}}(I_\ell)$. We define $\mathfrak{A}^{\mathrm{n}}_{\bar{I}}$ as the resulting distribution over these no-transcripts. Similarly, $Z^{\mathrm{Y}}_{\bar{I}}$ will be the random variable over yes-transcripts, with

corresponding distribution $\mathfrak{A}^{\mathrm{Y}}_{\bar{I}}$.

As noted earlier, the nonadaptive algorithm $\mathcal{A}$ corresponds to a distribution P_A over q-tuples $\bar{I}$ of query intervals. We define $\mathfrak{A}^{\mathrm{n}}$ as the distribution over transcripts corresponding to first drawing $\bar{I}$ from P_A and then making a draw from $\mathfrak{A}^{\mathrm{n}}_{\bar{I}}$. Similarly, we define $\mathfrak{A}^{\mathrm{Y}}$ as the distribution over transcripts corresponding to first drawing $\bar{I}$ from P_A and then making a draw from $\mathfrak{A}^{\mathrm{Y}}_{\bar{I}}$.

To prove Theorem 4.1.46 it is sufficient to show that the two distributions over transcripts described above are statistically close:

Lemma 4.1.47. $\mathrm{d}_{\mathrm{TV}}\left(\mathfrak{A}^{\mathrm{Y}}, \mathfrak{A}^{\mathrm{n}}\right) \leq 1/5$.

The proof of this lemma is structured as follows: first, for any *fixed* sequence of q queries $\bar{I}$, we bound the probability of the failure events, both for the uniform and the no-distributions:

Claim 4.1.48. *For each fixed sequence $\bar{I}$ of q query intervals, we have*

$$\Pr\left[B^{\mathrm{Y}}_{(\bar{I})}\right] \leq 1/10 \qquad \textit{and} \qquad \Pr_{\mathbf{p}\leftarrow\mathcal{D}_{no}}\left[B^{\mathrm{n}}_{(\bar{I})}\right] \leq 1/10.$$

(Note that the first probability above is taken over the randomness of the $\mathsf{INTCOND}_{\mathbf{u}}$ responses and the choice of offset and size in the "fake construction" of $\mathbf{u}$, while the second is over the random draw of $\mathbf{p} \sim \mathcal{D}_{\mathsf{no}}$ and over the $\mathsf{INTCOND}_{\mathbf{p}}$ responses.)

Next we show that, provided the failure events do not occur, the distribution over transcripts is exactly the same in both cases:

Claim 4.1.49. *Fix any sequence $\bar{I} = (I_1, \ldots, I_q)$ of q queries. Then, conditioned on their respective failure events not happening, $Z^{\mathrm{n}}_{\bar{I}}$ and $Z^{\mathrm{Y}}_{\bar{I}}$ are identically distributed:*

$$\textit{for every transcript } \mathcal{T} = ((I_1, s_1), \ldots, (I_q, s_q)), \quad \Pr\left[Z^{\mathrm{n}}_{\bar{I}} = \mathcal{T} \;\middle|\; \overline{B^{\mathrm{n}}_{(\bar{I})}}\right] = \Pr\left[Z^{\mathrm{Y}}_{\bar{I}} = \mathcal{T} \;\middle|\; \overline{B^{\mathrm{Y}}_{(\bar{I})}}\right].$$

Finally we combine these two claims to show that the two overall distributions of transcripts are statistically close:

Claim 4.1.50. *Fix any sequence of q queries $\bar{I} = (I_1, \ldots, I_q)$. Then $\mathrm{d}_{\mathrm{TV}}\left(\mathfrak{A}^{\mathrm{n}}_{\bar{I}}, \mathfrak{A}^{\mathrm{Y}}_{\bar{I}}\right) \leq 1/5$.*

Lemma 4.1.47 (and thus Theorem 4.1.46) directly follows from Claim 4.1.50 since, using the notation $\bar{s} = (s_1, \ldots, s_q)$ for a sequence of q answers to a sequence $\bar{I} = (I_1, \ldots, I_q)$ of q queries, which together define a transcript $\mathcal{T}(\bar{I}, \bar{s}) = ((I_1, s_1), \ldots, (I_q, s_q))$,

$$\begin{aligned}
\mathrm{d}_{\mathrm{TV}}\left(\mathfrak{A}^{\mathrm{Y}}, \mathfrak{A}^{\mathrm{n}}\right) &= \frac{1}{2}\sum_{\bar{I}}\sum_{\bar{s}}\left|P_A(\bar{I}) \cdot \Pr\left[Z^{Y}_{\bar{I}} = \mathcal{T}(\bar{I}, \bar{s})\right] - P_A(\bar{I}) \cdot \Pr\left[Z^{N}_{\bar{I}} = \mathcal{T}(\bar{I}, \bar{s})\right]\right| \\
&= \frac{1}{2}\sum_{\bar{I}} P_A(\bar{I}) \cdot \sum_{\bar{s}}\left|\Pr\left[Z^{Y}_{\bar{I}} = \mathcal{T}(\bar{I}, \bar{s})\right] - \Pr\left[Z^{N}_{\bar{I}} = \mathcal{T}(\bar{I}, \bar{s})\right]\right| \\
&\leq \max_{\bar{I}}\left\{\mathrm{d}_{\mathrm{TV}}\left(\mathfrak{A}^{\mathrm{Y}}_{\bar{I}}, \mathfrak{A}^{\mathrm{n}}_{\bar{I}}\right)\right\} \leq 1/5\,. \qquad (4.69)
\end{aligned}$$

This concludes the proof of Lemma 4.1.47 modulo the proofs of the above claims; we give those proofs in Section 4.1.8.1 below.

Proof of Claims 4.1.48 to 4.1.50 To prove Claim 4.1.48 we bound the probability of each of the bad events separately, starting with the no-case.

(i) Defining the event $B^{\mathrm{n}}_{\ell,\mathrm{size}}$ as

$$B^{\mathrm{n}}_{\ell,\mathrm{size}} = \{\Delta/\log n \le |I_\ell| \le \Delta \cdot (\log n)^2\} \ ,$$

we can use a union bound to get $\Pr[B^{\mathrm{n}}_{\mathrm{size}}] \le \sum_{\ell=1}^{q} \Pr[B^{\mathrm{n}}_{\ell,\mathrm{size}}]$. For any fixed setting of I_ℓ there are $O(\log\log n)$ values of $\Delta \in \{\frac{n}{2^X} \mid X \in \{\frac{1}{3}\log n, \ldots, \frac{2}{3}\log n\}\}$ for which $\Delta/\log n \le I_\ell \le \Delta \cdot (\log n)^2$. Hence we have $\Pr[B^{\mathrm{n}}_{\ell,\mathrm{size}}] = O((\log\log n)/\log n)$, and consequently $\Pr[B^{\mathrm{n}}_{\mathrm{size}}] = O(q(\log\log n)/\log n)$.

(ii) Similarly, defining the event $B^{\mathrm{n}}_{\ell,\mathrm{boundary}}$ as

$$B^{\mathrm{n}}_{\ell,\mathrm{boundary}} = \{|I_\ell| < \Delta/\log n \text{ and } I_\ell \text{ intersects two blocks}\} \ ,$$

we have $\Pr[B^{\mathrm{n}}_{\mathrm{boundary}}] \le \sum_{\ell=1}^{q} \Pr[B^{\mathrm{n}}_{\ell,\mathrm{boundary}}]$. For any fixed setting of I_ℓ, recalling the choice of a uniform random offset $y \in [n]$ for the blocks, we have that $\Pr[B^{\mathrm{n}}_{\ell,\mathrm{boundary}}] \le O(1/\log n)$, and consequently $\Pr[B^{\mathrm{n}}_{\mathrm{boundary}}] = O(q/\log n)$.

(iii) The analysis of $B^{\mathrm{n}}_{\mathrm{middle}}$ is identical (by considering the midpoint of a block instead of its endpoint), yielding directly $\Pr[B^{\mathrm{n}}_{\mathrm{middle}}] = O(q/\log n)$.

(iv) Fix $\ell \in [q]$ and recall that $B^{\mathrm{n}}_{\ell,\mathrm{outer}} = \{\Delta \cdot (\log n)^2 < |I_\ell| \text{ and } s_\ell \text{ is drawn from a block} \subsetneq I_\ell\}$. Fix any outcome for Δ such that $\Delta \cdot (\log n)^2 < |I_\ell|$ and let us consider only the randomness over the draw of s_ℓ from I_ℓ. Since there are $\Omega((\log n)^2)$ blocks contained entirely in I_ℓ, the probability that s_ℓ is drawn from a block not contained entirely in I_ℓ (there are at most two such blocks, one at each end of I_ℓ) is $O(1/(\log n)^2)$. Hence we have $\Pr[B^{\mathrm{n}}_{\ell,\mathrm{outer}}] \le O(1)/(\log n)^2$.

(v) Finally, recall that

$$B^{\mathrm{n}}_{\ell,\mathrm{collide}} = \{\Delta \cdot (\log n)^2 < |I_\ell| \text{ and } \exists j < \ell \text{ s.t. } s_\ell \text{ and } s_j \text{ belong to the same block } \} \ .$$

Fix $\ell \in [q]$ and a query interval I_ℓ. Let r_ℓ be the number of blocks in I_ℓ within which resides some previously sampled point s_j, $j \in [\ell - 1]$. Since there are $\Omega((\log n)^2)$ blocks in I_ℓ and $r_\ell \le \ell - 1$, the probability that s_ℓ is drawn from a block containing any s_j, $j < \ell$, is $O(\ell/(\log n)^2)$. Hence we have $\Pr[B^{\mathrm{n}}_{\ell,\mathrm{collide}}] = O(\ell/(\log n)^2)$.

With these probability bounds for bad events in hand, we can prove Claim 4.1.48:

Proof of Claim 4.1.48. Recall that $q \leq \tau \cdot \frac{\log n}{\log \log n}$. Recalling the definition of $B^{\mathrm{n}}_{(\bar{I})}$, a union bound yields

$$\begin{aligned}
\Pr[B^{\mathrm{n}}_{(\bar{I})}] &\leq \Pr[B^{\mathrm{n}}_{\mathrm{size}}] + \Pr[B^{\mathrm{n}}_{\mathrm{boundary}}] + \Pr[B^{\mathrm{n}}_{\mathrm{middle}}] + \sum_{\ell=1}^{q} \Pr[B^{\mathrm{n}}_{\ell,\mathrm{outer}}] + \sum_{\ell=1}^{q} \Pr[B^{\mathrm{n}}_{\ell,\mathrm{collide}}] \\
&= O\left(\frac{q \cdot \log\log n}{\log n}\right) + O\left(\frac{q}{\log n}\right) + O\left(\frac{q}{\log n}\right) + \sum_{\ell=1}^{q} O\left(\frac{1}{(\log n)^2}\right) + \sum_{\ell=1}^{q} O\left(\frac{\ell}{(\log n)^2}\right) \\
&\leq \frac{1}{10}\,,
\end{aligned}$$

where the last inequality holds for a sufficiently small choice of the absolute constant τ.

The same analysis applies unchanged for $\Pr[B^{\mathrm{Y}}_{\mathrm{size}}]$, $\Pr[B^{\mathrm{Y}}_{\mathrm{middle}}]$ and $\Pr[B^{\mathrm{Y}}_{\mathrm{boundary}}]$, using the "fake construction" view of $\mathbf{u}$ as described earlier. The arguments for $\Pr[B^{\mathrm{Y}}_{\ell,\mathrm{outer}}]$ and $\Pr[B^{\mathrm{Y}}_{\ell,\mathrm{collide}}]$ go through unchanged as well, and Claim 4.1.48 is proved. □

Proof of Claim 4.1.49. Fix any $\bar{I} = (I_1, \ldots, I_q)$ and any transcript $\mathcal{T} = ((I_1, s_1), \ldots, (I_q, s_q))$. Recall that the length-ℓ partial transcript $\mathcal{T}|_\ell$ is defined to be $((I_1, s_1), \ldots, (I_\ell, s_\ell))$. We define the random variables $Z^{\mathrm{n}}_{\bar{I},\ell}$ and $Z^{\mathrm{Y}}_{\bar{I},\ell}$ to be the length-ℓ prefixes of $Z^{\mathrm{n}}_{\bar{I}}$ and $Z^{\mathrm{Y}}_{\bar{I}}$ respectively. We prove Claim 4.1.49 by establishing the following, which we prove by induction on ℓ:

$$\Pr\left[Z^{\mathrm{n}}_{\bar{I},\ell} = \mathcal{T}|_\ell \,\middle|\, \overline{B^{\mathrm{n}}_{(\bar{I})}}\right] = \Pr\left[Z^{\mathrm{Y}}_{\bar{I},\ell} = \mathcal{T}|_\ell \,\middle|\, \overline{B^{\mathrm{Y}}_{(\bar{I})}}\right]. \tag{4.70}$$

For the base case, it is clear that (4.70) holds with $\ell = 0$. For the inductive step, suppose (4.70) holds for all $k \in [\ell - 1]$. When querying I_ℓ at the ℓ-th step, one of the following cases must hold (since we conditioned on the "bad events" not happening):

(1) I_ℓ is contained within a half-block (more precisely, either entirely within the first half of a block or entirely within the second half). In this case the "yes" and "no" distribution oracles behave exactly the same since both generate s_ℓ by sampling uniformly from I_ℓ.

(2) The point s_ℓ belongs to a block, contained entirely in I_ℓ, which is "fresh" in the sense that it contains no s_j, $j < \ell$. In the no-case this block may either be high-low or low-high; but since both outcomes have the same probability, there is another transcript with equal probability in which the two profiles are switched. Consequently (over the randomness in the draw of $\mathbf{p} \sim \mathcal{P}_{\mathsf{no}}$) the probability of picking s_ℓ in the no-distribution case is the same as in the uniform distribution case (i.e., uniform on the fresh blocks contained in I_ℓ).

This concludes the proof of Claim 4.1.49. □

Proof of Claim 4.1.50. Given Claims 4.1.48 and 4.1.49, Claim 4.1.50 is an immediate consequence of the following basic fact:

Fact 4.1.51. *Let $\mathbf{p}_1$, $\mathbf{p}_2$ be two distributions over the same finite set X. Let E_1, E_2, be two events*

such that $\mathbf{p}_i[E_i] = \alpha_i \leq \alpha$ *for* $i = 1, 2$ *and the conditional distributions* $(\mathbf{p}_i)_{\overline{E_i}}$ *are identical, i.e.* $\mathrm{d_{TV}}((\mathbf{p}_1)_{\overline{E_1}}, (\mathbf{p}_2)_{\overline{E_2}}) = 0$. *Then* $\mathrm{d_{TV}}(\mathbf{p}_1, \mathbf{p}_2) \leq \alpha$.

Proof. We first observe that since $(\mathbf{p}_2)_{\overline{E_2}}(E_2) = 0$ and $(\mathbf{p}_1)_{\overline{E_1}}$ is identical to $(\mathbf{p}_2)_{\overline{E_2}}$, it must be the case that $(\mathbf{p}_1)_{\overline{E_1}}(E_2) = 0$, and likewise $(\mathbf{p}_2)_{\overline{E_2}}(E_1) = 0$. This implies that $\mathbf{p}_1(E_2 \setminus E_1) = \mathbf{p}_2(E_1 \setminus E_2) = 0$. Now let us write

$$\begin{aligned} 2\,\mathrm{d_{TV}}(\mathbf{p}_1, \mathbf{p}_2) \quad = \quad & \sum_{x \in X \setminus (E_1 \cup E_2)} |\mathbf{p}_1(x) - \mathbf{p}_2(x)| + \sum_{x \in E_1 \cap E_2} |\mathbf{p}_1(x) - \mathbf{p}_2(x)| + \\ & \sum_{x \in E_1 \setminus E_2} |\mathbf{p}_1(x) - \mathbf{p}_2(x)| + \sum_{x \in E_2 \setminus E_1} |\mathbf{p}_1(x) - \mathbf{p}_2(x)|. \end{aligned}$$

We may upper bound $\sum_{x \in E_1 \cap E_2} |\mathbf{p}_1(x) - \mathbf{p}_2(x)|$ by $\sum_{x \in E_1 \cap E_2} (\mathbf{p}_1(x) + \mathbf{p}_2(x)) = \mathbf{p}_1(E_1 \cap E_2) + \mathbf{p}_2(E_1 \cap E_2)$, and the above discussion gives $\sum_{x \in E_1 \setminus E_2} |\mathbf{p}_1(x) - \mathbf{p}_2(x)| = \mathbf{p}_1(E_1 \setminus E_2)$ and $\sum_{x \in E_2 \setminus E_1} |\mathbf{p}_1(x) - \mathbf{p}_2(x)| = \mathbf{p}_2(E_2 \setminus E_1)$. We thus have

$$\begin{aligned} 2\,\mathrm{d_{TV}}(\mathbf{p}_1, \mathbf{p}_2) \quad & \leq \quad \sum_{x \in X \setminus (E_1 \cup E_2)} |\mathbf{p}_1(x) - \mathbf{p}_2(x)| + \mathbf{p}_1(E_1) + \mathbf{p}_2(E_2) \\ & \leq \quad \sum_{x \in X \setminus (E_1 \cup E_2)} |\mathbf{p}_1(x) - \mathbf{p}_2(x)| + \alpha_1 + \alpha_2. \end{aligned}$$

Finally, since $\mathrm{d_{TV}}((\mathbf{p}_1)_{\overline{E_1}}, (\mathbf{p}_2)_{\overline{E_2}}) = 0$, we have

$$\begin{aligned} \sum_{x \in X \setminus (E_1 \cup E_2)} |\mathbf{p}_1(x) - \mathbf{p}_2(x)| &= |\mathbf{p}_1(X \setminus (E_1 \cup E_2)) - \mathbf{p}_2(X \setminus (E_1 \cup E_2))| \\ &= \left|\mathbf{p}_1(\overline{E_1}) - \mathbf{p}_2(\overline{E_2})\right| = |\alpha_1 - \alpha_2|. \end{aligned}$$

Thus $2\,\mathrm{d_{TV}}(\mathbf{p}_1, \mathbf{p}_2) \leq |\alpha_1 - \alpha_2| + \alpha_1 + \alpha_2 = 2\max\{\alpha_1, \alpha_2\} \leq 2\alpha$, and the fact is established. □

This concludes the proof of Claim 4.1.50. □

4.1.8.2 A lower bound against adaptive algorithms: Outline of the proof of Theorem 4.1.44

Throughout this subsection $\mathcal{A}$ denotes a general adaptive algorithm that makes $q \leq \tau \cdot \frac{\log n}{\log \log n}$ queries, where as before $\tau \in (0, 1)$ is an absolute constant. Theorem 4.1.44 is a consequence of the following theorem, which deals with adaptive algorithms:

Theorem 4.1.52.

$$\left| \Pr_{\mathbf{p} \sim \mathcal{D}_{no}} [\mathcal{A}^{\mathsf{INTCOND}_{\mathbf{p}}} \textit{ outputs accept}] - \Pr[\mathcal{A}^{\mathsf{INTCOND}_{\mathbf{u}}} \textit{ outputs accept}] \right| \leq 1/5. \tag{4.71}$$

The idea here is to extend the previous analysis for non-adaptive algorithms, and argue that "adaptiveness

does not really help" to distinguish between $\mathbf{p} = \mathbf{u}$ and $\mathbf{p} \sim \mathcal{D}_{\mathsf{no}}$ given access to $\mathsf{INTCOND}_{\mathbf{p}}$.

As in the non-adaptive case, in order to prove Theorem 4.1.52, it is sufficient to prove that the transcripts for uniform and no-distributions are close in total variation distance; i.e., that

$$\mathrm{d}_{\mathrm{TV}}\left(\mathfrak{A}^{\mathrm{Y}}, \mathfrak{A}^{\mathrm{n}}\right) \leq 1/5. \tag{4.72}$$

The key idea used to prove this will be to introduce a *sequence* $\mathfrak{A}_{\mathrm{otf}}^{(k),\mathrm{n}}$ of distributions over transcripts (where "otf" stands for "on the fly"), for $0 \leq k \leq q$, such that (i) $\mathfrak{A}_{\mathrm{otf}}^{(0),\mathrm{n}} = \mathfrak{A}^{\mathrm{Y}}$ and $\mathfrak{A}_{\mathrm{otf}}^{(q),\mathrm{n}} = \mathfrak{A}^{\mathrm{n}}$, and (ii) the distance $\mathrm{d}_{\mathrm{TV}}\left(\mathfrak{A}_{\mathrm{otf}}^{(k),\mathrm{n}}, \mathfrak{A}_{\mathrm{otf}}^{(k+1),\mathrm{n}}\right)$ for each $0 \leq k \leq q-1$ is "small". This will enable us to conclude by the triangle inequality, as

$$\mathrm{d}_{\mathrm{TV}}\left(\mathfrak{A}^{\mathrm{n}}, \mathfrak{A}^{\mathrm{Y}}\right) = \mathrm{d}_{\mathrm{TV}}\left(\mathfrak{A}_{\mathrm{otf}}^{(0),\mathrm{n}}, \mathfrak{A}_{\mathrm{otf}}^{(q),\mathrm{n}}\right) \leq \sum_{k=0}^{q-1} \mathrm{d}_{\mathrm{TV}}\left(\mathfrak{A}_{\mathrm{otf}}^{(k),\mathrm{n}}, \mathfrak{A}_{\mathrm{otf}}^{(k+1),\mathrm{n}}\right). \tag{4.73}$$

To define this sequence, in the next subsection we will introduce the notion of an *extended transcript*, which in addition to the queries and samples includes additional information about the "local structure" of the distribution at the endpoints of the query intervals and the sample points. Intuitively, this extra information will help us analyze the interaction between the adaptive algorithm and the oracle. We will then describe an alternative process according to which a "faking algorithm" (reminiscent of the similar notion from Section 4.1.4.2) can interact with an oracle to generate such an extended transcript. More precisely, we shall define a sequence of such faking algorithms, paramaterized by "how much faking" they perform. For both the original ("non-faking") algorithm $\mathcal{A}$ and for the faking algorithms, we will show how extended transcripts can be generated "on the fly". The aforementioned distributions $\mathfrak{A}_{\mathrm{otf}}^{(k),\mathrm{n}}$ over (regular) transcripts are obtained by *truncating* the extended transcripts that are generated on the fly (i.e., discarding the extra information), and we shall argue that they satisfy requirements (i) and (ii) above.

Before turning to the precise definitions and the analysis of extended transcripts and faking algorithms, we provide the following variant of Fact 4.1.51, which will come in handy when we bound the right hand side of Equation (4.73).

Fact 4.1.53. *Let $\mathbf{p}_1$, $\mathbf{p}_2$ be two distributions over the same finite set X. Let E be an event such that $\mathbf{p}_i[E] = \alpha_i \leq \alpha$ for $i = 1, 2$ and the conditional distributions $(\mathbf{p}_1)_{\overline{E}}$ and $(\mathbf{p}_2)_{\overline{E}}$ are statistically close, i.e. $\mathrm{d}_{\mathrm{TV}}((\mathbf{p}_1)_{\overline{E}}, (\mathbf{p}_2)_{\overline{E}}) = \beta$. Then $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}_1, \mathbf{p}_2) \leq \alpha + \beta$.*

Proof. As in the proof of Fact 4.1.51, let us write

$$2\,\mathrm{d}_{\mathrm{TV}}(\mathbf{p}_1, \mathbf{p}_2) = \sum_{x \in X \setminus E} |\mathbf{p}_1(x) - \mathbf{p}_2(x)| + \sum_{x \in E} |\mathbf{p}_1(x) - \mathbf{p}_2(x)|.$$

We may upper bound $\sum_{x \in E} |\mathbf{p}_1(x) - \mathbf{p}_2(x)|$ by $\sum_{x \in E} (\mathbf{p}_1(x) + \mathbf{p}_2(x)) = \mathbf{p}_1(E) + \mathbf{p}_2(E) = \alpha_1 + \alpha_2$;

furthermore,

$$\begin{aligned}\sum_{x\in\bar{E}}|\mathbf{p}_1(x)-\mathbf{p}_2(x)| &= \sum_{x\in\bar{E}}\left|(\mathbf{p}_1)_{\bar{E}}(x)\cdot\mathbf{p}_1(\bar{E})-(\mathbf{p}_2)_{\bar{E}}(x)\cdot\mathbf{p}_2(\bar{E})\right| \\ &\le \mathbf{p}_1(\bar{E})\cdot\sum_{x\in\bar{E}}|(\mathbf{p}_1)_{\bar{E}}(x)-(\mathbf{p}_2)_{\bar{E}}(x)|+\left|\mathbf{p}_1(\bar{E})-\mathbf{p}_2(\bar{E})\right|\cdot(\mathbf{p}_2)_{\bar{E}}(\bar{E}) \\ &\le (1-\alpha_1)\cdot(2\beta)+|\alpha_2-\alpha_1|\cdot 1 \le 2\beta+|\alpha_2-\alpha_1|\end{aligned}$$

Thus $2\,\mathrm{d_{TV}}(\mathbf{p}_1,\mathbf{p}_2)\le 2\beta+|\alpha_1-\alpha_2|+\alpha_1+\alpha_2 = 2\beta+2\max\{\alpha_1,\alpha_2\}\le 2(\alpha+\beta)$, and the fact is established. □

4.1.8.3 Extended transcripts and drawing $\mathbf{p}\sim\mathcal{D}_{\mathsf{no}}$ on the fly.

Observe that the testing algorithm, seeing only pairs of queries and answers, does not have direct access to all the underlying information – namely, in the case of a no-distribution, whether the profile of the block that the sample point comes from is $\downarrow\uparrow$ or $\uparrow\downarrow$. It will be useful for us to consider an "extended" version of the transcripts, which includes this information along with information about the profile of the "boundary" blocks for each queried interval, even though this information is not directly available to the algorithm.

Definition 4.1.54. With the same notation as in Definition 4.1.45, the *extended transcript* of a sequence of queries made by $\mathcal{A}$ and the corresponding responses is a sequence $\mathcal{E}=(I_\ell,s_\ell,b_\ell)_{\ell\in[q]}$ of triples, where I_ℓ and s_ℓ are as before, and $b_\ell=(b_\ell^L,b_\ell^{\mathrm{samp}},b_\ell^R)\in\{\downarrow\uparrow,\uparrow\downarrow\}^3$ is a triple defined as follows: Let $B_{i^L},\ldots,B_{i^R}$ be the blocks that I_ℓ intersects, going from left to right. Then

1. b_ℓ^L is the profile of the block B_{i^L};
2. b_ℓ^R is the profile of the block B_{i^R};
3. b_ℓ^{samp} is the profile of the block $B_\ell\in\{B_{i^L},\ldots,B_{i^R}\}$ that s_ℓ belongs to.

We define $\mathcal{E}|_k$ to be the length-k prefix of an extended transcript $\mathcal{E}$.

As was briefly discussed prior to the current subsection, we shall be interested in considering algorithms that *fake* some answers to their queries. Specifically, given an adaptive algorithm $\mathcal{A}$, we define $\mathcal{A}^{(1)}$ as the algorithm that *fakes* its first query, in the following sense: If the first query made by $\mathcal{A}$ to the oracle is some interval I, then the algorithm $\mathcal{A}^{(1)}$ does not call $\mathsf{INTCOND}$ on I but instead chooses a point s uniformly at random from I and then behaves exactly as $\mathcal{A}$ would behave if the $\mathsf{INTCOND}$ oracle had returned s in response to the query I. More generally, we define $\mathcal{A}^{(k)}$ for all $0\le k\le q$ as the algorithm behaving like $\mathcal{A}$ but faking its first k queries (note that $\mathcal{A}^{(0)}=\mathcal{A}$).

In Section 4.1.8.3 we explain how extended transcripts can be generated for $\mathcal{A}^{(0)}=\mathcal{A}$ in an "on the fly" fashion so that the resulting distribution over extended transcripts is the same as the one that would result from first drawing $\mathbf{p}$ from $\mathcal{D}_{\mathsf{no}}$ and then running algorithm $\mathcal{A}$ on it. It follows that when we remove the extension to the transcript so as to obtain a regular transcript, we get a distribution over transcripts that is identical to

$\mathfrak{A}^{\mathrm{n}}$. In Section 4.1.8.3 we explain how to generate extended transcripts for $\mathcal{A}^{(k)}$ where $0 \leq k \leq q$. We note that for $k \geq 1$ the resulting distribution over extended transcripts is *not* the same as the one that would result from first drawing $\mathbf{p}$ from $\mathcal{D}_{\mathsf{no}}$ and then running algorithm $\mathcal{A}^{(k)}$ on it. However, this is not necessary for our purposes. For our purposes it is sufficient that the distributions corresponding to pairs of consecutive indices $(k, k+1)$ are similar (including the pair $(0,1)$), and that for $k = q$ the distribution over regular transcripts obtained by removing the extension to the transcript is identical to $\mathfrak{A}^{\mathrm{Y}}$.

Extended transcripts for $\mathcal{A} = \mathcal{A}^{(0)}$ Our proof of Equation (4.72) takes advantage of the fact that one can view the draw of a no-distribution from $\mathcal{D}_{\mathsf{no}}$ as being done "on the fly" during the course of algorithm $\mathcal{A}$'s execution. First, the size Δ and the offset y are drawn at the very beginning, but we may view the profile vector ϑ as having its components chosen independently, coordinate by coordinate, only as $\mathcal{A}$ interacts with $\mathsf{INTCOND}$ – each time an element s_ℓ is obtained in response to the ℓ-th query I_ℓ, only then are the elements of the profile vector ϑ corresponding to the three coordinates of b_ℓ chosen (if they were not already completely determined by previous calls to $\mathsf{INTCOND}$). More precise details follow.

Consider the ℓ-th query I_ℓ that $\mathcal{A}$ makes to $\mathsf{INTCOND}_{\mathbf{p}}$. Inductively some coordinates of ϑ may have been already set by previous queries. Let $B_{i^L}, \ldots, B_{i^R}$ be the blocks that I_ℓ intersects. First, if the coordinate of ϑ corresponding to block B_{i^L} was not already set by a previous query, a fair coin is tossed to choose a setting from $\{\downarrow\uparrow, \uparrow\downarrow\}$ for this coordinate. Likewise, if the coordinate of ϑ corresponding to block B_{i^R} was not already set (either by a previous query or because $i^R = i^L$), a fair coin is tossed to choose a setting from $\{\downarrow\uparrow, \uparrow\downarrow\}$ for this coordinate.

At this point, the values of b_ℓ^L and b_ℓ^R have been set. A simple but important observation is that these outcomes of b_ℓ^L and b_ℓ^R completely determine the probabilities (call them α^L and α^R respectively) that the block B_ℓ from which s_ℓ will be chosen is B_{i^L} (is B_{i^R} respectively), as we explain in more detail next. If $i^R = i^L$ then there is no choice to be made, and so assume that $i^R > i^L$. For $K \in \{L, R\}$ let $\rho_1^K \cdot \Delta$ be the size of the intersection of I_ℓ with the first (left) half of B_{i^K} and let $\rho_2^K \cdot \Delta$ be the size of the intersection of I_ℓ with the second (right) half of B_{i^K}. Note that $0 < \rho_1^K + \rho_2^K \leq 1$ and that $\rho_1^L = 0$ when $\rho_2^L \leq 1/2$ and similarly $\rho_2^R = 0$ when $\rho_1^R \leq 1/2$. If $b_\ell^K = \uparrow\downarrow$ then let $w^K = \rho_1^K \cdot (1+2\varepsilon) + \rho_2^K \cdot (1-2\varepsilon) = \rho_1^K + \rho_2^K + 2\varepsilon(\rho_1^K - \rho_2^K)$, and if $b_\ell^K = \downarrow\uparrow$ then let $w^K = \rho_1^K + \rho_2^K - 2\varepsilon(\rho_1^K - \rho_2^K)$. We now set $\alpha^K = \frac{w^K}{w^L + w^R + (i^L - i^R - 1)}$. The block B_{i^L} is selected with probability α^L, the block B_{i^R} is selected with probability α^R, and for $i^R \geq i^L + 2$, each of the other blocks is selected with equal probability, $\frac{1}{w^L + w^R + (i^L - i^R - 1)}$.

Given the selection of the block B_ℓ as described above, the element s_ℓ and the profile b_ℓ^{samp} of the block to which it belongs are selected as follows. If the coordinate of ϑ corresponding to B_ℓ has already been determined, then b_ℓ^{samp} is set to this value and s_ℓ is drawn from B_ℓ as determined by the $\downarrow\uparrow$ or $\uparrow\downarrow$ setting of b_ℓ^{samp}. Otherwise, a fair coin is tossed, b_ℓ^{samp} is set either to $\downarrow\uparrow$ or to $\uparrow\downarrow$ depending on the outcome, and s_ℓ is drawn from B_ℓ as in the previous case (as determined by the setting of b_ℓ^{samp}). Now all of I_ℓ, s_ℓ, and $b_\ell = (b_\ell^L, b_\ell^{\mathrm{samp}}, b_\ell^R)$ have been determined and the triple (I_ℓ, s_ℓ, b_ℓ) is taken as the ℓ-th element of the extended transcript.

We now define $\mathfrak{A}_{\text{otf}}^{(0),\text{n}}$ as follows. A draw from this distribution over (non-extended) transcripts is obtained by first drawing an extended transcript $(I_1, s_1, b_1), \ldots, (I_q, s_q, b_q)$ from the on-the-fly process described above, and then removing the third element of each triple to yield $(I_1, s_1), \ldots, (I_q, s_q)$. This is exactly the distribution over transcripts that is obtained by first drawing $\mathbf{p}$ from $\mathcal{D}_{\mathsf{no}}$ and then running $\mathcal{A}$ on it.

Extended transcripts for $\mathcal{A}^{(k)}$, $k \geq 0$ In this subsection we define the distribution $\mathfrak{A}_{\text{otf}}^{(k),n}$ for $0 \leq k \leq q$ (the definition we give below will coincide with our definition from the previous subsection for $k = 0$). Here too the size Δ and the offset y are drawn at the very beginning, and the coordinates of the profile vector ϑ are chosen on the fly, together with the sample points. For each $\ell > k$, the pair (s_ℓ, b_ℓ) is selected exactly as was described for $\mathcal{A}$, conditioned on the length-k prefix of the extended transcript and the new query I_ℓ (as well as the choice of (Δ, y)). It remains to explain how the selection is made for $1 \leq \ell \leq k$.

Consider a value $1 \leq \ell \leq k$ and the ℓ-th query interval I_ℓ. As in our description of the "on-the-fly" process for $\mathcal{A}$, inductively some coordinates of ϑ may have been already set by previous queries. Let $B_{i^L}, \ldots, B_{i^R}$ be the blocks that I_ℓ intersects. As in the process for $\mathcal{A}$, if the coordinate of ϑ corresponding to block B_{i^L} was not already set by a previous query, a fair coin is tossed to choose a setting from $\{\downarrow\uparrow, \uparrow\downarrow\}$ for this coordinate. Likewise, if the coordinate of ϑ corresponding to block B_{i^R} was not already set (either by a previous query or because $i^L = i^R$), a fair coin is tossed to choose a setting from $\{\downarrow\uparrow, \uparrow\downarrow\}$ for this coordinate. Hence, b_ℓ^L and b_ℓ^R are set exactly the same as described for $\mathcal{A}$.

We now explain how to set the probabilities α^L and α^R of selecting the block B_ℓ (from which s_ℓ is chosen) to be B_{i^L} and B_{i^R}, respectively. Since the "faking" process should choose s_ℓ to be a uniform point from I_ℓ, the probability α^L is simply $|B_{i^L} \cap I_\ell|/|I_\ell|$, and similarly for α^R. (If $i^L = i^R$ we take $\alpha^L = 1$ and $\alpha^R = 0$.) Thus the values of α^L and α^R are completely determined by the number of blocks j and the relative sizes of the intersection of I_ℓ with B_{i^L} and with B_{i^R}. Now, with probability α^L the block B_ℓ is chosen to be B_{i^L}, with probability α^R it is chosen to be B_{i^R} and with probability $1 - \alpha^L - \alpha^R$ it is chosen uniformly among $\{B_{i^L+1}, \ldots, B_{i^R-1}\}$.

Given the selection of the block B_ℓ as described above, s_ℓ is chosen to be a uniform random element of $B_\ell \cap I_\ell$. The profile b_ℓ^{samp} of B_ℓ is selected as follows:

1. If the coordinate of ϑ corresponding to B_ℓ has already been determined (either by a previous query or because $B_\ell \in \{B_{i^L}, B_{i^R}\}$), then b_ℓ^{samp} is set accordingly.
2. Otherwise, the profile of B_ℓ was not already set; note that in this case it must hold that $B_\ell \notin \{B_{i^L}, B_{i^R}\}$. We look at the half of B_ℓ that s_ℓ belongs to, and toss a biased coin to set its profile $b_\ell^{\text{samp}} \in \{\downarrow\uparrow, \uparrow\downarrow\}$: If s_ℓ belongs to the first half, then the coin toss's probabilities are $((1-2\varepsilon)/2, (1+2\varepsilon)/2)$; otherwise, they are $((1+2\varepsilon)/2, (1-2\varepsilon)/2)$.

Let $\mathfrak{E}_{\text{otf}}^{(k),\text{n}}$ denote the distribution induced by the above process over extended transcripts, and let $\mathfrak{A}_{\text{otf}}^{(k),\text{n}}$ be the corresponding distribution over regular transcripts (that is, when removing the profiles from the transcript). As noted in Section 4.1.8.3, for $k = 0$ we have that $\mathfrak{A}_{\text{otf}}^{(0),\text{n}} = \mathfrak{A}^{\text{n}}$. In the other extreme, for $k = q$, since each point

s_ℓ is selected uniformly in I_ℓ (with no dependence on the selected profiles) we have that $\mathfrak{A}_{\mathrm{otf}}^{(q),\mathrm{n}} = \mathfrak{A}^{\mathrm{Y}}$. In the next subsection we bound the total variation distance between $\mathfrak{A}_{\mathrm{otf}}^{(k),\mathrm{n}}$ and $\mathfrak{A}_{\mathrm{otf}}^{(k+1),\mathrm{n}}$ for every $0 \leq k \leq q-1$ by bounding the distance between the corresponding distributions $\mathfrak{E}_{\mathrm{otf}}^{(k),\mathrm{n}}$ and $\mathfrak{E}_{\mathrm{otf}}^{(k+1),\mathrm{n}}$. Roughly speaking, the only difference between the two (for each $0 \leq k \leq q-1$) is in the distribution over $(s_{k+1}, b_{k+1}^{\mathrm{samp}})$. As we argue in more detail and formally in the next subsection, conditioned on certain events (determined, among other things, by the choice of (Δ, y)), we have that $(s_{k+1}, b_{k+1}^{\mathrm{samp}})$ are distributed the same under $\mathfrak{E}_{\mathrm{otf}}^{(k),\mathrm{n}}$ and $\mathfrak{E}_{\mathrm{otf}}^{(k+1),\mathrm{n}}$.

4.1.8.4 Bounding $\mathrm{d}_{\mathrm{TV}}\left(\mathfrak{A}_{\mathrm{otf}}^{(k),\mathrm{n}}, \mathfrak{A}_{\mathrm{otf}}^{(k+1),\mathrm{n}}\right)$

As per the foregoing discussion, we can focus on bounding the total variation distance between extended transcripts

$$\mathrm{d}_{\mathrm{TV}}\left(\mathfrak{E}_{\mathrm{otf}}^{(k),\mathrm{n}}, \mathfrak{E}_{\mathrm{otf}}^{(k+1),\mathrm{n}}\right)$$

for arbitrary fixed $k \in \{0, \ldots, q-1\}$. Before diving into the proof, we start by defining the probability space we shall be working in, as well as explaining the different sources of randomness that are in play and how they fit into the random processes we end up analyzing.

The probability space. Recall the definition of an extended transcript: for notational convenience, we reserve the notation $\mathcal{E} = (I_\ell, s_\ell, b_\ell)_{\ell \in [q]}$ for extended transcript valued random variables, and will write $E = (\iota_\ell, \sigma_\ell, \pi_\ell)_{\ell \in [q]}$ for a fixed outcome. We denote by Σ the space of all such tuples E, and by Λ the set of all possible outcomes for (Δ, y). The sample space we are considering is now defined as $X \stackrel{\mathrm{def}}{=} \Sigma \times \Lambda$: that is, an extended transcript along with the underlying choice of block size and offset[7]. The two probability measures on X we shall consider will be induced by the execution of $\mathcal{A}^{(k)}$ and $\mathcal{A}^{(k+1)}$, as per the process detailed below.

A key thing to observe is that, as we focus on two "adjacent" faking algorithms $\mathcal{A}^{(k)}$ and $\mathcal{A}^{(k+1)}$, it will be sufficient to consider the following equivalent view of the way an extended transcript is generated:

1. up to (and including) stage k, the faking algorithm generates on its own both the queries ι_ℓ and the uniformly distributed samples $\sigma_\ell \in \iota_\ell$; it also chooses its $(k+1)$-st query ι_{k+1};
2. then, at that point only is the choice of (Δ, y) made; and the profiles π_ℓ ($1 \leq \ell \leq k$) of the *previous* blocks decided upon, as described in Section 4.1.8.3;
3. after this, the sampling and block profile selection is made exactly according to the previous "on-the-fly process" description.

The reason that we can defer the choice of (Δ, y) and the setting of the profiles in the manner described above is the following: For both $\mathcal{A}^{(k)}$ and $\mathcal{A}^{(k+1)}$, the choice of each σ_ℓ for $1 \leq \ell \leq k$ depends only on ι_ℓ

[7] We emphasize the fact that the algorithm, whether faking or not, has access neither to the "extended" part of the transcript nor to the choice of (Δ, y); however, these elements are part of the events we analyze.

and the choice of each ι_ℓ for $1 \leq \ell \leq k+1$ depends only on $(\iota_1, \sigma_1), \ldots, (\iota_{\ell-1}, \sigma_{\ell-1})$. That is, there is no dependence on (Δ, y) nor on any $\pi_{\ell'}$ for $\ell' \leq \ell$. By deferring the choice of the pair (Δ, y) we may consider the randomness coming in its draw only at the $(k+1)$-st stage (which is the pivotal stage here). Note that, both for $\mathcal{A}^{(k)}$ and $\mathcal{A}^{(k+1)}$, the resulting distribution over X induced by the description above exactly matches the one from the "on-the-fly" process. In the next paragraph, we go into more detail, and break down further the randomness and choices happening in this new view.

Sources of randomness. To define the probability measure on this space, we describe the process that, up to stage $k+1$, generates the corresponding part of the extended transcript and the (Δ, y) for $\mathcal{A}^{(m)}$ (where $m \in \{k, k+1\}$) (see the previous subsections for precise descriptions of how the following random choices are made):

(R1) $\mathcal{A}^{(m)}$ draws $\iota_1, \sigma_1, \ldots, \iota_k, \sigma_k$ and finally ι_{k+1} by itself;

(R2) the outcome of (Δ, y) is chosen: this "retroactively" fixes the partition of the ι_ℓ's ($1 \leq \ell \leq k+1$) into blocks $B_{i_L}^{(\ell)}, \ldots, B_{i_R}^{(\ell)}$;

(R3) the profiles of $B_{i_L}^{(\ell)}$, $B_{i_R}^{(\ell)}$ and B_ℓ (i.e., the values of the triples π_ℓ, for $1 \leq \ell \leq k$) are drawn;

(R4) the profiles of $B_{i_L}^{(k+1)}$, $B_{i_R}^{(k+1)}$ are chosen;

(R5) the block selection (choice of the block B_{k+1} to which σ_{k+1} will belong to) is made:

- a) whether it will be one of the two end blocks, or one of the inner ones (for $\mathcal{A}^{(k+1)}$ this is based on the respective sizes of the end blocks, and for $\mathcal{A}^{(k)}$ this is based on the weights of the end blocks, using the profiles of the end blocks);
- b) the choice of the block itself is performed:
  - if the block has to be one of the outer ones, draw it based on either the respective sizes (for $\mathcal{A}^{(k+1)}$) or the respective weights (for $\mathcal{A}^{(k)}$, using the profiles of the end blocks)
  - if the block has to be one of the inner ones, draw it uniformly at random among all inner blocks;

(R6) the sample σ_{k+1} and the profile π_{k+1}^{samp} are chosen;

(R7) the rest of the transcript, for $k+1, \ldots, q$, is iteratively chosen (in the same way for $\mathcal{A}^{(k)}$ and $\mathcal{A}^{(k+1)}$) according to the on-the-fly process discussed before.

Note that the only differences between the processes for $\mathcal{A}^{(k)}$ and $\mathcal{A}^{(k+1)}$ lie in steps (R5a), (R5b) and (R6) of the $(k+1)$-st stage.

Bad events and outline of the argument Let $G(\iota_{k+1})$ (where 'G' stands for 'Good') denote the settings of (Δ, y) that satisfy the following: Either (i) $|\iota_{k+1}| > \Delta \cdot (\log n)^2$ or (ii) $|\iota_{k+1}| < \Delta / \log n$ and ι_{k+1} is contained entirely within a single half block. We next define three indicator random variables for a given element $\omega = (E, (\Delta, y))$ of the sample space X, where $E = ((\iota_1, \sigma_1, \pi_1), \ldots, (\iota_q, \sigma_q, \pi_q))$. The first, Γ_1, is

zero when $(\Delta, y) \notin G(\iota_{k+1})$. Note that the randomness for Γ_1 is over the choice of (Δ, y) and the choice of ι_{k+1}. The second, Γ_2, is zero when ι_{k+1} intersects at least two blocks and the block B_{k+1} is one of the two extreme blocks intersected by ι_{k+1}. The third, Γ_3, is zero when ι_{k+1} is not contained entirely within a single half block and B_{k+1} is a block whose profile had already been set (either because it contains a selected point σ_ℓ for $\ell \leq k$ or because it belongs to one of the two extreme blocks for some queried interval ι_ℓ for $\ell \leq k$). For notational ease we write $\overline{\Gamma}(E)$ to denote the triple $(\Gamma_1, \Gamma_2, \Gamma_3)$. Observe that these indicator variables are well defined, and correspond to events that are indeed subsets of our space X: given any element $\omega \in X$, whether $\Gamma_i(\omega) = 1$ (for $i \in \{1, 2, 3\}$) is fully determined.

Define $\mathbf{p}_1$, $\mathbf{p}_2$ as the two distributions over X induced by the executions of respectively $\mathcal{A}^{(k)}$ and $\mathcal{A}^{(k+1)}$ (in particular, by only keeping the first marginal of $\mathbf{p}_1$ we get back $\mathfrak{E}^{(k),n}$). Applying Fact 4.1.53 to $\mathbf{p}_1$ and $\mathbf{p}_2$, we obtain that

$$\begin{aligned}
\mathrm{d}_{\mathrm{TV}}(\mathbf{p}_1, \mathbf{p}_2) &\leq \Pr\big[\,\overline{\Gamma} \neq (1,1,1)\,\big] + \mathrm{d}_{\mathrm{TV}}\big(\mathbf{p}_1 \mid \overline{\Gamma} = (1,1,1), \mathbf{p}_2 \mid \overline{\Gamma} = (1,1,1)\big) \\
&\leq \Pr[\,\Gamma_1 = 0\,] + \Pr[\,\Gamma_2 = 0 \mid \Gamma_1 = 1\,] + \Pr[\,\Gamma_3 = 0 \mid \Gamma_1 = \Gamma_2 = 1\,] \\
&\quad + \mathrm{d}_{\mathrm{TV}}\big(\mathbf{p}_1 \mid \overline{\Gamma} = (1,1,1), \mathbf{p}_2 \mid \overline{\Gamma} = (1,1,1)\big)\,. \qquad (4.74)
\end{aligned}$$

To conclude, we can now deal which each of these 4 summands separately:

Claim 4.1.55. *We have that* $\Pr[\,\Gamma_1 = 0\,] \leq \eta(n)$*, where* $\eta(n) = O\left(\frac{\log\log n}{\log n}\right)$.

Proof. Similarly to the proof of Claim 4.1.48, for any fixed setting of ι_{k+1}, there are $O(\log\log n)$ values of $\Delta \in \left\{\, \frac{n}{2^j} \,:\, j \in \{\frac{1}{3}\log n, \ldots, \frac{2}{3}\log n\}\,\right\}$ for which $\Delta/\log n \leq \iota_{k+1} \leq \Delta \cdot (\log n)^2$. Therefore, the probability that one of these ("bad") values of Δ is selected is $O\left(\frac{\log\log n}{\log n}\right)$. If the choice of Δ is such that $|\iota_{k+1}| < \Delta/\log n$, then, by the choice of the random offset y, the probability that ι_{k+1} is not entirely contained within a single half block is $O(1/\log n)$. The claim follows. □

Claim 4.1.56. *We have that* $\Pr[\,\Gamma_2 = 0 \mid \Gamma_1 = 1\,] \leq \eta(n)$.

Proof. If $\Gamma_1 = 1$ because $|\iota_{k+1}| < \Delta/(\log n)^2$ and ι_{k+1} is entirely contained within a single half block, then $\Gamma_2 = 1$ (with probability 1). Otherwise, $|\iota_{k+1}| > \Delta \cdot (\log n)^2$, so that ι_{k+1} intersects at least $(\log n)^2$ blocks. The probability that one of the two extreme blocks is selected is hence $O(1/(\log n)^2)$, and the claim follows. □

Claim 4.1.57. *We have that* $\Pr[\,\Gamma_3 = 0 \mid \Gamma_1 = \Gamma_2 = 1\,] \leq \eta(n)$.

Proof. If $\Gamma_1 = 1$ because $|\iota_{k+1}| < \Delta/(\log n)^2$ and ι_{k+1} is entirely contained within a single half block, then $\Gamma_3 = 1$ (with probability 1). Otherwise, $|\iota_{k+1}| > \Delta \cdot (\log n)^2$, so that ι_{k+1} intersects at least $(\log n)^2$ blocks. Since $\Gamma_2 = 1$, the block B_{k+1} is uniformly selected from $(\log n)^2 - 2$ non-extreme blocks. Among them there are at most $3k = O\left(\frac{\log n}{\log\log n}\right)$ blocks whose profiles were already set. The probability that one of them is selected (so that $\Gamma_3 = 1$) is $O\left(\frac{1}{\log n \log\log n}\right) = O\left(\frac{\log\log n}{\log n}\right)$, and the claim follows. □

We are left with only the last term, $\mathrm{d_{TV}}\big(\mathbf{p}_1 \mid \overline{\Gamma} = (1,1,1), \mathbf{p}_2 \mid \overline{\Gamma} = (1,1,1)\big)$. But as we are now ruling out all the "bad events" that would induce a difference between the distributions of the extended transcripts under $\mathcal{A}^{(k)}$ and $\mathcal{A}^{(k+1)}$, it becomes possible to argue that this distance is actually zero:

Claim 4.1.58. $\mathrm{d_{TV}}\big(\mathbf{p}_1 \mid \overline{\Gamma} = (1,1,1), \mathbf{p}_2 \mid \overline{\Gamma} = (1,1,1)\big) = 0.$

Proof. Unrolling the definition, we can write $\mathrm{d_{TV}}\big(\mathbf{p}_1 \mid \overline{\Gamma} = (1,1,1), \mathbf{p}_2 \mid \overline{\Gamma} = (1,1,1)\big)$ as

$$\sum_{E,(\Delta,y)} \left|\Pr\Big[\,\mathcal{E}^{(k)} = E, \mathcal{Y}^{(m)} = (\Delta,y) \;\Big|\; \overline{\Gamma} = (1,1,1)\,\Big] - \Pr\Big[\,\mathcal{E}^{(k+1)} = E, \mathcal{Y}^{(m)} = (\Delta,y) \;\Big|\; \overline{\Gamma} = (1,1,1)\,\Big]\right| .$$

where $\mathcal{Y}^{(m)}$ denotes the Λ-valued random variable corresponding to $\mathcal{A}^{(m)}$. In order to bound this sum, we will show that each of its terms is zero: i.e., that for any fixed $(E,(\Delta,y)) \in \Sigma \times \Lambda$ we have

$$\Pr\Big[\,\mathcal{E}^{(k)} = E, \mathcal{Y}^{(k)} = (\Delta,y) \;\Big|\; \overline{\Gamma} = (1,1,1)\,\Big] = \Pr\Big[\,\mathcal{E}^{(k+1)} = E, \mathcal{Y}^{(k+1)} = (\Delta,y) \;\Big|\; \overline{\Gamma} = (1,1,1)\,\Big] .$$

We start by observing that, for $m \in \{k, k+1\}$,

$$\begin{aligned}
&\Pr\Big[\,\mathcal{E}^{(m)} = E, \mathcal{Y}^{(m)} = (\Delta,y) \;\Big|\; \overline{\Gamma} = (1,1,1)\,\Big] \\
&\qquad = \Pr\Big[\,\mathcal{E}^{(m)} = E \;\Big|\; \overline{\Gamma} = (1,1,1), \mathcal{Y}^{(m)} = (\Delta,y)\,\Big] \Pr\Big[\,\mathcal{Y}^{(m)} = (\Delta,y) \;\Big|\; \overline{\Gamma} = (1,1,1)\,\Big]
\end{aligned}$$

and that the term $\Pr\big[\,\mathcal{Y}^{(m)} = (\Delta,y) \;\big|\; \overline{\Gamma} = (1,1,1)\,\big] = \Pr\big[\,\mathcal{Y}^{(m)} = (\Delta,y)\,\big]$ is identical for $m = k$ and $m = k+1$. Therefore, it is sufficient to show that

$$\Pr\Big[\,\mathcal{E}^{(k)} = E \;\Big|\; \overline{\Gamma} = (1,1,1), \mathcal{Y}^{(k)} = (\Delta,y)\,\Big] = \Pr\Big[\,\mathcal{E}^{(k+1)} = E \;\Big|\; \overline{\Gamma} = (1,1,1), \mathcal{Y}^{(k+1)} = (\Delta,y)\,\Big] .$$

Let $\omega = (E,(\Delta,y)) \in X$ be arbitrary, with $E = ((\iota_1,\sigma_1,\pi_1),\ldots,(\iota_q,\sigma_q,\pi_q)) \in \Sigma$, and let $m \in \{k,k+1\}$. We can express $\Phi^{(m)}(\omega) \stackrel{\text{def}}{=} \Pr\big[\,\mathcal{E}^{(m)} = E \;\big|\; \overline{\Gamma} = (1,1,1), \mathcal{Y}^{(m)} = (\Delta,y)\,\big]$ as the product of the following 5 terms:

(T1) $p_k^{(m),\mathrm{int},\mathrm{samp}}(\omega)$, defined as

$$\begin{aligned}
p_k^{(m),\mathrm{int},\mathrm{samp}}(\omega) &\stackrel{\text{def}}{=} \Pr\Big[\,\mathcal{E}^{(m),\mathrm{int},\mathrm{samp}}|_k = E^{\mathrm{int},\mathrm{samp}}|_k \;\Big|\; \overline{\Gamma} = (1,1,1), \mathcal{Y}^{(m)} = (\Delta,y)\,\Big] \\
&= \Pr\Big[\,\mathcal{E}^{(m),\mathrm{int},\mathrm{samp}}|_k = E^{\mathrm{int},\mathrm{samp}}|_k\,\Big] ,
\end{aligned}$$

where $E_\ell^{\mathrm{int},\mathrm{samp}}$ denotes (ι_ℓ,σ_ℓ) and $E^{\mathrm{int},\mathrm{samp}}|_k$ denotes $(E_1^{\mathrm{int},\mathrm{samp}},\ldots,E_k^{\mathrm{int},\mathrm{samp}})$;

(T2) $p_k^{(m),\mathrm{prof}}(\omega)$, defined as

$$\begin{aligned}
p_k^{(m),\mathrm{prof}}(\omega) &\stackrel{\text{def}}{=} \Pr\Big[\,\mathcal{E}^{(m),\mathrm{prof}}|_k = E^{\mathrm{prof}}|_k \;\Big|\; \mathcal{E}^{(m),\mathrm{int},\mathrm{samp}}|_k = E^{\mathrm{int},\mathrm{samp}}|_k, \overline{\Gamma} = (1,1,1), \mathcal{Y}^{(m)} = (\Delta,y)\,\Big] \\
&= \Pr\Big[\,\mathcal{E}^{(m),\mathrm{prof}}|_k = E^{\mathrm{prof}}|_k \;\Big|\; \mathcal{E}^{(m),\mathrm{int},\mathrm{samp}}|_k = E^{\mathrm{int},\mathrm{samp}}|_k, \mathcal{Y}^{(m)} = (\Delta,y)\,\Big] ,
\end{aligned}$$

where $E^{\text{prof}}|_k$ denotes $(\pi_1, \ldots, \pi_k)$;

(T3) $p_{k+1}^{(m),\text{int}}(\omega)$, defined as

$$p_{k+1}^{(m),\text{int}}(\omega) \stackrel{\text{def}}{=} \Pr\Big[\, I_{k+1} = \iota_{k+1} \;\Big|\; \mathcal{E}^{(m),\text{int,samp}}|_k = E^{\text{int,samp}}|_k, \overline{\Gamma} = (1,1,1), \mathcal{Y}^{(m)} = (\Delta, y) \,\Big] \\ = \Pr\Big[\, I_{k+1} = \iota_{k+1} \;\Big|\; \mathcal{E}^{(m),\text{int,samp}}|_k = E^{\text{int,samp}}|_k \,\Big] \;;$$

(T4) $p_{k+1}^{(m),\text{samp,prof}}(\omega)$, defined as

$$p_{k+1}^{(m),\text{samp,prof}}(\omega) \\ \stackrel{\text{def}}{=} \Pr\Big[\, (s_{k+1}, b_{k+1}) = (\sigma_{k+1}, \pi_{k+1}) \;\Big|\; I_{k+1} = \iota_{k+1}, \mathcal{E}|_k^{(m)} = E|_k, \overline{\Gamma} = (1,1,1), \mathcal{Y}^{(m)} = (\Delta, y) \,\Big] \;;$$

(T5) and the last term $p_{k+2}^{(m)}(\omega)$, defined as

$$p_{k+2}^{(m)}(\omega) \stackrel{\text{def}}{=} \Pr\Big[\, \mathcal{E}^{(m)}|_{k+2,\ldots,q} = E|_{k+2,\ldots,q} \;\Big|\; \mathcal{E}^{(m)}|_{k+1} = E|_{k+1}, \overline{\Gamma} = (1,1,1), \mathcal{Y}^{(m)} = (\Delta, y) \,\Big] \,,$$

where $E|_{k+1} = ((\iota_{k+1}, \sigma_{k+1}, \pi_{k+1}), \ldots, (\iota_q, \sigma_q, \pi_q))$.

Note that we could remove the conditioning on $\overline{\Gamma}$ for the first three terms, as they only depend on the length-k prefix of the (extended) transcript and the choice of ι_{k+1}, that is, on the randomness from (R1). The important observation is that the above probabilities are independent of whether $m = k$ or $m = k+1$. We first verify this for (T1), (T2), (T3) and (T5), and then turn to the slightly less straightforward term (T4). This is true for $p_k^{(m),\text{int,samp}}(E)$ because $\mathcal{A}^{(k)}$ and $\mathcal{A}^{(k+1)}$ select their interval queries in exactly the same manner, and for $1 \le \ell \le k$, the ℓ-th sample point is uniformly selected in the ℓ-th queried interval. Similarly we get that $p_{k+1}^{(k),\text{int}}(E) = p_{k+1}^{(k+1),\text{int}}(E)$. The probabilities $p_k^{(k),\text{prof}}(E)$ and $p_k^{(k+1),\text{prof}}(E)$ are induced in the same manner by (R2) and (R3), and $p_{k+2}^{(k)}(E) = p_{k+2}^{(k+1)}(E)$ since for both $\mathcal{A}^{(k)}$ and $\mathcal{A}^{(k+1)}$, the pair (s_ℓ, b_ℓ) is distributed the same for every $\ell \ge k+2$ (conditioned on any length-$(k+1)$ prefix of the (extended) transcript and the choice of (Δ, y)).

Turning to (T4), observe that $\Gamma_1 = \Gamma_2 = \Gamma_3 = 1$ (by conditioning). Consider first the case that $\Gamma_1 = 1$ because $|\iota_{k+1}| < \Delta/\log n$ and ι_{k+1} is contained entirely within a single half block. For this case there are two subcases. In the first subcase, the profile of the block that contains ι_{k+1} was already set. This implies that b_{k+1} is fully determined (in the same manner) for both $m = k$ and $m = k+1$. In the second subcase, the profile of the block that contains ι_{k+1} (which is an extreme block) is set independently and with equal probability to either $\downarrow\uparrow$ or $\uparrow\downarrow$ for both $m = k$ and $m = k+1$. In either subcase, s_{k+1} is uniformly distributed in ι_{k+1} for both $m = k$ and $m = k+1$.

Next, consider the remaining case that $\Gamma_1 = 1$ because $|\iota_{k+1}| > \Delta \cdot (\log n)^2$. In this case, since $\Gamma_2 = 1$, the block B_{k+1} is not an extreme block, and since $\Gamma_3 = 1$, the profile of the block B_{k+1} was not previously set. Given this, it follows from the discussion at the end of Section 4.1.8.3 that the distribution of (s_{k+1}, b_{k+1}) is identical whether $m = k$ (and $\mathcal{A}^{(m)}$ does not fake the $(k+1)$-th query) or $m = k+1$ (and $\mathcal{A}^{(m)}$ fakes the

$(k+1)$-th query). $\square$

Assembling the pieces, the 4 claims above together with Equation (4.74) yield $\mathrm{d_{TV}}\big(\mathfrak{E}^{(k),\mathrm{n}}, \mathfrak{E}^{(k+1),\mathrm{n}}\big) \leq \mathrm{d_{TV}}(\mathbf{p}_1, \mathbf{p}_2) \leq 3\eta(n)$, and finally

$$\begin{aligned}\mathrm{d_{TV}}\big(\mathfrak{A}^{\mathrm{n}}, \mathfrak{A}^{\mathrm{Y}}\big) = \mathrm{d_{TV}}\Big(\mathfrak{A}_{\mathrm{otf}}^{(0),\mathrm{n}}, \mathfrak{A}_{\mathrm{otf}}^{(q),\mathrm{n}}\Big) &\leq \sum_{k=0}^{q-1} \mathrm{d_{TV}}\Big(\mathfrak{A}_{\mathrm{otf}}^{(k),\mathrm{n}}, \mathfrak{A}_{\mathrm{otf}}^{(k+1),\mathrm{n}}\Big) \\ &\leq \sum_{k=0}^{q-1} \mathrm{d_{TV}}\Big(\mathfrak{E}^{(k),\mathrm{n}}, \mathfrak{E}^{(k+1),\mathrm{n}}\Big) \leq 3q \cdot \eta(n) \\ &\leq 1/5\end{aligned}$$

for a suitable choice of the absolute constant τ. $\square$

4.1.9 Conclusion

We have introduced a new conditional sampling framework for testing probability distributions and shown that it allows significantly more query-efficient algorithms than the standard framework for a range of problems. This new framework presents many potential directions for future work.

One specific goal is to strengthen the upper and lower bounds for problems studied in this paper. As a concrete question along these lines, we conjecture that COND algorithms for testing equality of two unknown distributions $\mathbf{p}_1$ and $\mathbf{p}_2$ over $[n]$ require $(\log n)^{\Omega(1)}$ queries. A broader goal is to study more properties of distributions beyond those considered in this paper; natural candidates here, which have been well-studied in the standard model, are monotonicity (for which we have preliminary results), independence between marginals of a joint distribution, and entropy. Yet another goal is to study distributions over other structured domains such as the Boolean hypercube $\{0,1\}^n$ – here it would seem natural to consider "subcube" queries, analogous to the INTCOND queries we considered when the structured domain is the linearly ordered set $[n]$. A final broad goal is to study distribution *learning* (rather than testing) problems in the conditional sampling framework.[8]

4.2 Dual Sampling: When You Can Query Too

4.2.1 Introduction

In this section, we consider the power of two natural oracles, again generalizing (although in a different and incomparable direction than the conditional sampling model of Section 4.1) the standard sampling setting.

[8] We note that, subsequent to our work and concurrent to the writing of this thesis, some of the directions and questions outlined here were addressed in several papers, some by the author of this dissertation. Specifically, our conjecture on the query complexity of testing identity of two unknown distributions was answered in the affirmative, and resolved up to a quadratic factor [1, 94]. Monotonicity testing of distributions under the COND, PAIRCOND, and INTCOND models (as well as in the oracle models we shall cover in Section 4.2) was studied in [44] and [42]. Finally, recent work of Bhattacharyya and Chakraborty [31] set out to pursue the particular direction of "subcube queries" over $\{0,1\}^n$, and more generally Σ^n.

The first is a *dual oracle*, which combines the standard model for distributions and the familiar one commonly assumed for testing Boolean and real-valued functions. In more detail, the testing algorithm is granted access to the unknown distribution $\mathbf{p}$ through two independent oracles, one providing samples of the distribution, while the other, on query i in the domain of the distribution, provides the value of the probability density function at i.[9]

Definition 4.2.1 (Dual access model)**.** Let $\mathbf{p}$ be a fixed distribution over $[n] = \{1, \ldots, n\}$. A *dual oracle for* $\mathbf{p}$ is a pair of oracles $(\mathsf{SAMP}_{\mathbf{p}}, \mathsf{EVAL}_{\mathbf{p}})$ defined as follows: when queried, the *sampling* oracle $\mathsf{SAMP}_{\mathbf{p}}$ returns an element $i \in [n]$, where the probability that i is returned is $\mathbf{p}(i)$ independently of all previous calls to any oracle; while the *evaluation* oracle $\mathsf{EVAL}_{\mathbf{p}}$ takes as input a query element $j \in [n]$, and returns the probability weight $\mathbf{p}(j)$ that the distribution puts on j.

It is worth noting that this type of dual access to a distribution has been considered (under the name *combined oracle*) in [23] and [108], where they address the task of estimating (multiplicatively) the entropy of the distribution, or the f-divergence between two of them (see Section 4.2.3 for a discussion of their results).

The second oracle that we consider provides samples of the distribution as well as queries to the *cumulative distribution function* (cdf) at any point in the domain.[10]

Definition 4.2.2 (Cumulative Dual access model)**.** Let $\mathbf{p}$ be a fixed distribution over $[n]$. A *cumulative dual oracle for* $\mathbf{p}$ is a pair of oracles $(\mathsf{SAMP}_{\mathbf{p}}, \mathsf{CEVAL}_{\mathbf{p}})$ defined as follows: the *sampling* oracle $\mathsf{SAMP}_{\mathbf{p}}$ behaves as before, while the *evaluation* oracle $\mathsf{CEVAL}_{\mathbf{p}}$ takes as input a query element $j \in [n]$, and returns the probability weight that the distribution puts on $[j]$, that is $\mathbf{p}([j]) = \sum_{i=1}^{j} \mathbf{p}(i)$.

Remark 4.2.3. We will sometimes refer as a *multiplicatively noisy* $\mathsf{EVAL}_{\mathbf{p}}$ (or similarly for $\mathsf{CEVAL}_{\mathbf{p}}$) to an evaluation oracle with takes an additional input parameter $\tau > 0$ and returns a value $\hat{d}_i$ within a multiplicative factor $(1 + \tau)$ of the true $\mathbf{p}(i)$. Note however that this notion of noisy oracle does not preserve the two-query simulation of a dual oracle by a cumulative dual one.

4.2.1.1 Motivation and discussion

As a first motivation to this hybrid model, consider the following scenario: There is a huge and freely available dataset, which a computationally-limited party – call it Arthur – needs to process. Albeit all the data is public and Arthur can view any element of his choosing, extracting further information from the dataset (such as the number of occurrences of a particular element) takes too much time. However, a third-party, Merlin, has already spent resources in preprocessing this dataset and is willing to disclose such information – yet at a price. This leaves Arthur with the following question: *how can he get his work done as quickly as possible, paying as*

[9] Note that in both definitions, one can decide to disregard the corresponding evaluation oracle, which in effect amounts to falling back to the standard sampling model; moreover, for our domain $[n]$, any $\mathsf{EVAL}_{\mathbf{p}}$ query can be simulated by (at most) two queries to a $\mathsf{CEVAL}_{\mathbf{p}}$ oracle – in other terms, the cumulative dual model is at least as powerful as the dual one.

[10] We observe that such a cumulative evaluation oracle CEVAL appears in [19, Section 8].

little as possible? This type of question is captured by our new model, and can be analyzed in this framework. For instance, if the samples are stored in sorted order, implementing either of our oracles becomes possible with only a logarithmic overhead per query. It is worth noting that Google has published their N-gram models, which describe their distribution model on 5-word sequences in the English language. In addition, they have made available the texts on which their model was constructed. Thus, samples of the distribution in addition to query access to probabilities of specific domain elements may be extracted from the Google model.

A second and entirely theoretical motivation for studying distribution testing in these two dual oracle settings arises from attempting to understand the limitations and underlying difficulties of the standard sampling model. Indeed, by circumventing the lower bound, one may get a better grasp on the core issues whence the hardness stemmed in the first place. Another motivation arises from data privacy, when a curator administers a database of highly sensitive records (e.g, healthcare information, or financial records). Differential privacy [88, 93, 92] studies mechanisms which allow the curator to release relevant information about its database without without jeopardizing the privacy of the individual records. In particular, mechanisms have been considered that enable the curator to *release* a sanitized approximation $\tilde{\mathbf{p}}$ of its database $\mathbf{p}$, which "behaves" essentially the same for all queries of a certain type – such as *counting* or *interval queries* [35].[11] Specifically, if the user needs to test a property of a database, it is sufficient to test whether the sanitized database has the property, using now both samples and interval (i.e., CEVAL) or counting (EVAL) queries. As long as the tester has some tolerance (in that it accepts databases that are close to having the property), it is then possible to decide whether the true database itself is close to having the property of interest.

Finally, a further motivation is the tight connection between the dual access model and the *data-stream model*, as shown by Guha et al. ([108, Theorem 25]): more precisely, they prove that any (multiplicative) approximation algorithm for a large class of functions of the distribution (functions that are invariant by relabeling of any two elements of the support) in the dual access model yields a space-efficient, $O(1)$-pass approximation algorithm for the same function in the data-stream model.

4.2.1.2 Our results and techniques

We focus here on four fundamental and pervasive problems in distribution testing, which are testing *uniformity*, *identity* to a known distribution $\mathbf{p}^*$, *closeness* between two (unknown) distributions $\mathbf{p}_1$, $\mathbf{p}_2$, and finally *entropy and support size*. As usual in the distribution testing literature, the notion of distance we use is the *total variation distance* (or statistical distance), which is essentially the ℓ_1 distance between the probability distributions. Testing closeness is thus the problem of deciding if two distributions are equal or far from each other in total variation distance; while tolerant testing aims at deciding whether they are sufficiently close versus far from each other.

As shown in Table 4.2, which summarizes our results and compares them to the corresponding bounds for the standard sampling-only (SAMP), evaluation-only (EVAL) and conditional sampling (COND) models,

[11] A counting query is of the form "how many records in the database satisfy predicate χ?" – or, equivalently, "what is the probability that a random record drawn from the database satisfies χ?".

we indeed manage to bypass the aforementioned limitations of the sampling model, and give (often tight) algorithms with sample complexity either constant (with relation to n) or logarithmic, where a polynomial dependence was required in the standard setting.

Our main finding overall is that *both dual models allow testing algorithms to significantly outperform both* SAMP *and* COND *algorithms*, either with relation to the dependence on n or, for the latter, in $1/\varepsilon$; further, these testing algorithms are *significantly simpler*, both conceptually and in their analysis, and can often be made robust to some multiplicative noise in the evaluation oracle. Another key observation is that this new flexibility not only allows us to tell whether two distributions are close or far, but also to efficiently estimate their distance.[12]

In more detail, we show that for the problem of testing equivalence between distributions, both our models allow to get rid of any dependence on n, with a (tight) sample complexity of $\Theta(1/\varepsilon)$. The upper bound is achieved by adapting an EVAL-only algorithm of [155] (for identity testing) to our setting, while the lower bound is obtained by designing a far-from-uniform instance which "defeats" simultaneously both oracles of our models. Turning to tolerant testing of equivalence, we describe algorithms whose sample complexity is again independent of n, in sharp contrast with the $n^{1-o(1)}$ lower bound of the standard sampling model. Moreover, we are able to show that, at least in the Dual access model, our quadratic dependence on ε is optimal. The same notable improvements apply to the query complexity of estimating the support size of the distribution, which becomes constant (with relation to n) in both of our access models – versus quasilinear if one only allows sampling.

As for the task of (additively) estimating the entropy of an arbitrary distribution, we give an algorithm whose sample complexity is only polylogarithmic in n, and show that this is tight in the Dual access model, up to the exponent of the logarithm. Once more, this is to be compared to the $n^{1-o(1)}$ lower bound for sampling.

Problem	SAMP	COND [49, 48]	EVAL	Dual	Cumulative Dual
Testing uniformity	$\Theta\left(\frac{\sqrt{n}}{\varepsilon^2}\right)$ [104, 20, 138]	$\tilde{O}\left(\frac{1}{\varepsilon^2}\right), \Omega\left(\frac{1}{\varepsilon^2}\right)$	$O\left(\frac{1}{\varepsilon}\right)$ [155], $\Omega\left(\frac{1}{\varepsilon}\right)^*$	$\Theta\left(\frac{1}{\varepsilon}\right)$ (†)	$\Theta\left(\frac{1}{\varepsilon}\right)$ (†)
Testing $\equiv \mathbf{p}^*$	$\tilde{\Theta}\left(\frac{\sqrt{n}}{\varepsilon^2}\right)$ [21, 138]	$\tilde{O}\left(\frac{1}{\varepsilon^4}\right)$			
Testing $\mathbf{p}_1 \equiv \mathbf{p}_2$	$\Theta\left(\left(\max\left(\frac{N^{2/3}}{\varepsilon^{4/3}}, \frac{\sqrt{N}}{\varepsilon^2}\right)\right)\right)$ [20, 174, 58]	$\tilde{O}\left(\frac{\log^5 n}{\varepsilon^4}\right)$	$\Omega\left(\frac{1}{\varepsilon}\right)^*$		
Tolerant uniformity	$O\left(\frac{1}{(\varepsilon_2-\varepsilon_1)^2}\frac{n}{\log n}\right)$ [172, 170] $\Omega\left(\frac{n}{\log n}\right)$ [172, 167]	$\tilde{O}\left(\frac{1}{(\varepsilon_2-\varepsilon_1)^{20}}\right)$	$\Omega\left(\frac{1}{(\varepsilon_2-\varepsilon_1)^2}\right)^*$	$\Theta\left(\frac{1}{(\varepsilon_2-\varepsilon_1)^2}\right)$ (†)	$O\left(\frac{1}{(\varepsilon_2-\varepsilon_1)^2}\right)$ (†)
Tolerant $\mathbf{p}^*$ Tolerant $\mathbf{p}_1, \mathbf{p}_2$	$\Omega\left(\frac{n}{\log n}\right)$ [172, 167]				
Estimating entropy to $\pm\Delta$	$\Theta\left(\frac{n}{\log n}\right)$ [172, 167]			$O\left(\frac{\log^2 \frac{n}{\Delta}}{\Delta^2}\right)$ (†), $\Omega(\log n)$	$O\left(\frac{\log^2 \frac{n}{\Delta}}{\Delta^2}\right)$ (†)
Estimating support size to $\pm\varepsilon n$	$\Theta\left(\frac{n}{\log n}\right)$ [172, 167]			$\Theta\left(\frac{1}{\varepsilon^2}\right)$	$O\left(\frac{1}{\varepsilon^2}\right)$

Table 4.2: Summary of results in the dual and cumulative dual models. (†) stands for "robust to multiplicative noise". The bounds with an asterisk are those which, in spite of being for different models, derive from the results of the last two columns.

While it is not clear, looking at these problems, whether the additional flexibility that the Cumulative Dual

[12] For details on the equivalence between tolerant testing and distance estimation, the reader is referred to [140].

access grants over the Dual one can *unconditionally* yield strictly more sample-efficient testing algorithms, we do provide a separation between the two models in Section 4.2.3.2 by showing an exponential improvement in the query complexity for estimating the entropy of a distribution given the promise that the latter is (close to) monotone. This leads us to suspect that for the task of testing monotonicity (for which we have preliminary results), under a structural assumption on the distribution, or more generally for properties intrinsically related to the underlying total order of the domain, such a speedup holds. Moreover, we stress out the fact that our $\Omega\big(1/(\varepsilon_2-\varepsilon_1)^2\big)$ lower bound for tolerant identity testing does not apply to the Cumulative Dual setting.

One of the main techniques we use for algorithms in the dual model is a general approach[13] for estimating very efficiently any quantity of the form $\mathbb{E}_{i\sim\mathbf{p}}\left[\Phi(i,\mathbf{p}(i))\right]$, for any *bounded* function Φ. In particular, in the light of our lower bounds, this technique is both an intrinsic and defining feature of the Dual model, as it gives essentially tight upper bounds for the problems we consider.

On the other hand, for the task of proving lower bounds, we no longer can take advantage of the systematic characterizations known for the sampling model (see e.g. [14], Sect. 2.4.1). For this reason, we have to rely on reductions from known-to-be-hard problems (such as estimating the bias of a coin), or prove indistinguishability in a *customized* fashion.

4.2.1.3 Organization

We begin with the first three problems of testing equivalence of distributions in Section 4.2.2, where we describe our testing upper and lower bounds. We then turn to the harder problem of *tolerant* testing. Finally, we tackle in Section 4.2.3 the task of performing entropy and support size estimation, and give for the latter matching upper and lower bounds.

4.2.2 Uniformity and identity of distributions

4.2.2.1 Testing

In this section, we consider the three following testing problems, each of them a generalization of the previous:

Uniformity testing: given oracle access to $\mathbf{p}$, decide whether $\mathbf{p}=\mathbf{u}$ (the uniform distribution on $[n]$) or is far from it;

Identity testing: given oracle access to $\mathbf{p}$ and the full description of a fixed $\mathbf{p}^*$, decide whether $\mathbf{p}=\mathbf{p}^*$ or is far from it;

Closeness testing: given independent oracle accesses to $\mathbf{p}_1$, $\mathbf{p}_2$ (both unknown), decide whether $\mathbf{p}_1=\mathbf{p}_2$ or $\mathbf{p}_1$, $\mathbf{p}_2$ are far from each other.

We begin by stating here two results from the literature that transpose straightforwardly in our setting. Observe that since the problem of testing closeness between two unknown distributions $\mathbf{p}_1,\mathbf{p}_2$ in particular

[13] We note that a similar method was utilized in [23], albeit in a less systematic way.

encompasses the identity to known $\mathbf{p}^*$ testing (and a fortiori the uniformity testing) one, this upper bound automatically applies to these as well.

Theorem 4.2.4 ([155, Theorem 24])**.** *In the query access model, there exists a tester for identity to a known distribution $\mathbf{p}^*$ with query complexity $O\left(\frac{1}{\varepsilon}\right)$.*

Note that the tester given in [155] is neither tolerant nor robust; however, it only uses query access. [49] later adapt this algorithm to give a tester for closeness between two unknown distributions, in a setting which can be seen as "relaxed" dual access model:[14]

Theorem 4.2.5 ([49, Theorem 12], and Theorem 4.1.39)**.** *In the dual access model, there exists a tester for closeness between two unknown distributions $\mathbf{p}_1$, $\mathbf{p}_2$ with sample complexity $O\left(\frac{1}{\varepsilon}\right)$.*

It is worth noting that the algorithm in question is conceptually very simple – namely, it consists in drawing samples from both distributions and then querying the respective probability mass both distributions put on them, hoping to detect a violation.

Remark 4.2.6. As mentioned, the setting of the theorem is slightly more general than stated – indeed, it only assumes "approximate" query access to $\mathbf{p}_1$, $\mathbf{p}_2$ (in their terminology, this refers to an evaluation oracle that outputs, on query $x \in [n]$, a good *multiplicative* estimate of $\mathbf{p}_i(x)$, for *most* of the points x).

Lower bound Getting more efficient testing seems unlikely – the dependence on $1/\varepsilon$ being "as good as it gets." The following result formalizes this, showing that indeed both Theorems 4.2.4 and 4.2.5 are tight, even for the least challenging task of testing uniformity:

Theorem 4.2.7 (Lower bound for dual oracle testers)**.** *In the dual access model, any tester for uniformity must have query complexity $\Omega\left(\frac{1}{\varepsilon}\right)$.*

Although the lower bound above applies only to the dual model, one can slightly adapt the proof to get the following improvement:

Theorem 4.2.8 (Lower bound for cumulative dual oracle testers)**.** *In the cumulative dual access model, any tester for uniformity must have sample complexity $\Omega\left(\frac{1}{\varepsilon}\right)$.*

Sketch. Theorem 4.2.8 directly implies Theorem 4.2.7, so we focus on the former. The high-level idea is to trick the algorithm by somehow "disabling" the additional flexibility coming from the oracles.

To do so, we start with a distribution that is far from uniform, but easy to recognize when given evaluation queries. We then shuffle its support randomly in such a way that (a) sampling will not, with overwhelming probability, reveal anything, while (b) evaluation queries essentially need to find a needle in a haystack. Note that the choice of the shuffling must be done carefully, as the tester has access to the cumulative distribution

[14] In the sense that the evaluation oracle, being simulated via another type of oracle, is not only noisy but also allowed to err on a small set of points.

function of any no-instance $\mathbf{p}$: in particular, using a random permutation will not work. Indeed, it is crucial for the cumulative distribution function to be as close as the linear function $x \in [n] \mapsto \frac{x}{n}$ as possible; meaning that the set of elements on which $\mathbf{p}$ differs from $\mathbf{u}$ had better be a consecutive "chunk" (otherwise, looking at the value of the cdf at a uniformly random point would give away the difference with uniform with non-negligible probability: such a point x is likely to have at least a "perturbed point" before *and* after it, so that $\sum_{i \leq x} \mathbf{p}(x) \neq \frac{x}{n}$).

Fix any $\varepsilon \in (0, \frac{1}{2}]$; for $n \geq \frac{1}{\varepsilon}$, set $m \stackrel{\text{def}}{=} (1-\varepsilon)n - 1$, and consider testing a distribution $\mathbf{p}$ on $[n]$ which is either (a) the uniform distribution or (b) chosen uniformly at random amongst the family of distributions $(\mathbf{p}_r)_{0 \leq r \leq m}$, defined this way: for any offset $0 \leq r < m$, $\mathbf{p}_r$ is obtained as follows:

1. Set $\mathbf{p}(1) = \varepsilon + \frac{1}{n}$, $\mathbf{p}(2) = \cdots = \mathbf{p}(\varepsilon n + 1) = 0$, and $\mathbf{p}(k) = \frac{1}{n}$ for the remaining $m = (1-\varepsilon)n - 1$ points;
2. Shift the whole support (modulo n) by adding r.

At a high-level, what this does is keeping the "chunk" on which the cdf of the no-instance grouped together, and just place it at a uniformly random position; outside this interval, the cdf's are exactly the same, and the only way to detect a difference with CEVAL is to make a query in the "chunk." Furthermore, it is not hard to see that any no-instance distribution will be exactly ε-far from uniform, so that any tester $\mathcal{T}$ must distinguish between cases (a) and (b) with probability at least 2/3.

Suppose by contradiction that there exists a tester $\mathcal{T}$ making $q = o\big(\frac{1}{\varepsilon}\big)$ queries (without loss of generality, we can further assume $\mathcal{T}$ makes exactly q queries; and that for any SAMP query, the tester also gets "for free" the result of an evaluation query on the sample). Given dual access to a $\mathbf{p} = \mathbf{p}_r$ generated as in case (b), observe first that, since the outputs of the sample queries are independent of the results of the evaluation queries, one can assume that some evaluation queries are performed first, followed by some sample queries, before further evaluation queries (where the evaluation points may depend arbitrarily on the sample query results) are made. That is, we subdivide the queries in 3: first, q_1 consecutive EVAL queries, then a sequence of q_2 SAMP queries, and finally q_3 EVAL queries. Define the following "bad" events:

- E_1: one of the first q_1 evaluation queries falls outside the set $G \stackrel{\text{def}}{=} \{\varepsilon n + 2 + r, \ldots, n + r\} \mod n$;
- E_2: one of the q_2 sampling queries returns a sample outside G, conditioned on $\overline{E_1}$;
- E_3: one of the q_3 evaluation queries is on a point outside G, conditioned on $\overline{E_1} \cap \overline{E_2}$.

It is clear that, conditioned on $\overline{E_1} \cap \overline{E_2} \cap \overline{E_3}$, all the tester sees is exactly what its view would have been in case (a) (probabilities equal to $\frac{1}{n}$ for any EVAL query, and uniform sample from G for any SAMP one). It is thus sufficient to show that $\Pr\big[\,\overline{E_1} \cap \overline{E_2} \cap \overline{E_3}\,\big] = 1 - o(1)$.

- As r is chosen uniformly at random, $\Pr[\,E_1\,] \leq q_1 \frac{n-m}{n} = q_1(\varepsilon + \frac{1}{n})$;
- since $\mathbf{p}(G) = \frac{m}{n} = 1 - \varepsilon - \frac{1}{n} \geq 1 - 2\varepsilon$, $\Pr[\,E_2\,] \leq 1 - (1-2\varepsilon)^{q_2}$;
- finally, $\Pr[\,E_3\,] \leq q_3(\varepsilon + \frac{1}{n})$;

we therefore have $\Pr[\,E_1 \cup E_2 \cup E_3\,] \leq (q_1 + q_3)(\varepsilon + \frac{1}{n}) + 1 - (1-2\varepsilon)^{q_2} = O(q\varepsilon) = o(1)$, as claimed. $\square$

4.2.2.2 Tolerant testing

In this section, we describe tolerant testing algorithms for the three problems of uniformity, identity and closeness; note that by a standard reduction (see Parnas et al. ([140], Section 3.1), this is equivalent to estimating the distance between the corresponding distributions. As hinted in the introduction, our algorithm relies on a general estimation approach that will be illustrated further in Section 4.2.3, and which constitutes a fundamental feature of the dual oracle: namely, the ability to estimate cheaply quantities of the form $\mathbb{E}_{i\sim\mathbf{p}}\left[\Phi(i,\mathbf{p}(i))\right]$ for any *bounded* function Φ.

Theorem 4.2.9. *In the dual access model, there exists a tolerant tester for uniformity with query complexity* $O\left(\frac{1}{(\varepsilon_2-\varepsilon_1)^2}\right)$.

Proof. We describe such a tester $\mathcal{T}$; it will start by estimating the quantity $2\mathrm{d}_{\mathrm{TV}}(\mathbf{p},\mathbf{u})$ up to some additive $\gamma \stackrel{\text{def}}{=} \varepsilon_2-\varepsilon_1$ (and then accept if and only if its estimate $\hat{d}$ is at most $2\varepsilon_1+\gamma=\varepsilon_1+\varepsilon_2$).
In order to approximate this quantity, observe that[15]

$$\begin{aligned}\mathrm{d}_{\mathrm{TV}}(\mathbf{p},\mathbf{u}) &= \frac{1}{2}\sum_{i\in[n]}\left|\mathbf{p}(i)-\frac{1}{n}\right| = \sum_{i:\mathbf{p}(i)>\frac{1}{n}}\left(\mathbf{p}(i)-\frac{1}{n}\right) = \sum_{i:\mathbf{p}(i)>\frac{1}{n}}\left(1-\frac{1}{n\mathbf{p}(i)}\right)\cdot\mathbf{p}(i)\\ &= \mathbb{E}_{i\sim\mathbf{p}}\left[\left(1-\frac{1}{n\mathbf{p}(i)}\right)\mathbb{1}_{\left\{\mathbf{p}(i)>\frac{1}{n}\right\}}\right] \qquad (4.75)\end{aligned}$$

where $\mathbb{1}_E$ stands for the indicator function of set (or event) E; thus, $\mathcal{T}$ only has to do get an empirical estimate of this expected value, which can be done by taking $m=O\left(1/(\varepsilon_2-\varepsilon_1)^2\right)$ samples s_i from $\mathbf{p}$, querying $\mathbf{p}(s_i)$ and computing $X_i=\left(1-\frac{1}{n\mathbf{p}(s_i)}\right)\mathbb{1}_{\left\{\mathbf{p}(s_i)>\frac{1}{n}\right\}}$ (cf. Algorithm 30).

Algorithm 30 Tester $\mathcal{T}$: ESTIMATE-L_1

Require: $\mathsf{SAMP}_{\mathbf{p}}$ and $\mathsf{EVAL}_{\mathbf{p}}$ oracle access, parameters $0\le\varepsilon_1<\varepsilon_2$
  Set $m\stackrel{\text{def}}{=}\Theta\left(\frac{1}{\gamma^2}\right)$, where $\gamma\stackrel{\text{def}}{=}\frac{\varepsilon_2-\varepsilon_1}{2}$.
  Draw $s_1,\ldots,s_m$ from $\mathbf{p}$
  for $i=1$ **to** m **do**
    With EVAL, get $X_i\stackrel{\text{def}}{=}\left(1-\frac{1}{n\mathbf{p}(s_i)}\right)\mathbb{1}_{\left\{\mathbf{p}(s_i)>\frac{1}{n}\right\}}$
  end for
  Compute $\hat{d}\stackrel{\text{def}}{=}\frac{1}{m}\sum_{i=1}^m X_i$.
  if $\hat{d}\le\frac{\varepsilon_1+\varepsilon_2}{2}$ **then**
    return accept
  else
    return reject
  end if

[15] Note that dividing by $\mathbf{p}(i)$ is "legal", since if $\mathbf{p}(i)=0$ for some $i\in[n]$, this point will never be sampled, and thus no division by 0 will ever occur.

Analysis Define the random variable X_i as above; from Eq.(4.75), we can write its expectation as

$$\mathbb{E}[X_i] = \sum_{k=1}^{n} \mathbf{p}(k) \left| 1 - \frac{1}{n\mathbf{p}(k)} \right| \mathbb{1}_{\{\mathbf{p}(k) > \frac{1}{n}\}} = \mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{u}). \tag{4.76}$$

Since the X_i's are independent and take value in $[0, 1]$, an additive Chernoff bound ensures that

$$\Pr\Big[\left| \hat{d} - \mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{u}) \right| \geq \gamma \Big] \leq 2e^{-2\gamma^2 m} \tag{4.77}$$

which is at most $1/3$ by our choice of m. Conditioning from now on on the event $\left| \hat{d} - \mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{u}) \right| < \gamma$:

- if $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{u}) \leq \varepsilon_1$, then $\hat{d} \leq \varepsilon_1 + \gamma$, and $\mathcal{T}$ outputs accept;
- if $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{u}) > \varepsilon_2$, then $\hat{d} > \varepsilon_2 - \gamma$, and $\mathcal{T}$ outputs reject.

Furthermore, the algorithm makes m SAMP queries, and m EVAL queries. □

Remark 4.2.10. Note that we can also do it with EVAL queries only (same query complexity), by internally drawing uniform samples: indeed,

$$2\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{u}) = \sum_{i \in [n]} \left| \mathbf{p}(i) - \frac{1}{n} \right| = \sum_{i \in [n]} |n\mathbf{p}(i) - 1| \cdot \frac{1}{n} = 2\mathbb{E}_{x \sim \mathbf{u}} \left[|n\mathbf{p}(x) - 1| \, \mathbb{1}_{\left\{\frac{1}{n} > \mathbf{p}(x)\right\}} \right]$$

This also applies to the first corollary below, as long as the known distribution is efficiently samplable by the algorithm.

Indeed, the proof above can be easily extended to other distributions than uniform, and even to the case of two unknown distributions:

Corollary 4.2.11. *In the dual access model, there exists a tolerant tester for identity to a known distribution with query complexity* $O\left(\frac{1}{(\varepsilon_2 - \varepsilon_1)^2}\right)$.

Corollary 4.2.12. *In the dual access model, there exists a tolerant tester for closeness between two unknown distributions with query complexity* $O\left(\frac{1}{(\varepsilon_2 - \varepsilon_1)^2}\right)$. *As noted in the next subsection, this is optimal (up to constant factors).*

Interestingly, this tester can be made robust to multiplicative noise, i.e. can be shown to work even when the answers to the EVAL queries are only accurate up to a factor $(1 + \gamma)$ for $\gamma > 0$: it suffices to set $\gamma = \varepsilon/2$, getting on each point $\hat{\mathbf{p}}(i) \in [(1+\gamma)^{-1}, 1+\gamma]\mathbf{p}(i)$, and work with $X_i = (1 - \mathbf{p}^*(s_i)/\hat{\mathbf{p}}(s_i)) \, \mathbb{1}_{\{\hat{\mathbf{p}}(s_i) > \mathbf{p}^*(s_i)\}}$ and estimate the expectation up to $\pm\gamma$ (or, for closeness between two unknown distributions, setting $\gamma = \varepsilon/4$).

4.2.2.3 Lower bound

In this subsection, we show that the upper bounds of Theorem 4.2.9 and Corollaries 4.2.11 and 4.2.12 are tight.

Theorem 4.2.13. *In the dual access model, performing* $(\varepsilon_1, \varepsilon_2)$*-testing for uniformity requires sample*

complexity $\Omega\left(\frac{1}{(\varepsilon_2-\varepsilon_1)^2}\right)$ *(the bound holds even when only asking* ε_1 *to be* $\Omega(1)$*).*

Proof. The overall idea lies on a reduction from distinguishing between two types of biased coins to tolerant testing for uniformity. In more detail, given access to samples from a fixed coin (promised to be of one of these two types), we define a probability distribution as follows: the domain $[n]$ is randomly partitioned into $K = 1/\varepsilon^2$ pairs of buckets, each bucket having same number of elements; the distribution is uniform within each bucket, and the two buckets of each pair are balanced to have total weight $2/K$. However, within each pair of buckets (A, B), the probability mass is divided according to a coin toss (performed "on-the-fly" when a query is made by the tolerant tester), so that either (a) $\mathbf{p}(A) = (1+\alpha)/K$ and $\mathbf{p}(B) = (1-\alpha)/K$, or (b) $\mathbf{p}(A) = \mathbf{p}(B) = 1/K$. Depending on whether the coin used for this choice is fair or $(\frac{1}{2}+\varepsilon)$)biased, the resulting distribution will (with high probability) have different distance from uniformity – sufficiently for a tolerant tester to distinguish between the two cases.

Construction We start by defining the instances of distributions we shall consider. Fix any $\varepsilon \in (0, \frac{1}{100})$; without loss of generality, assume n is even, and $n \gg 1/\varepsilon$. Define $\alpha = 1/(1+\varepsilon) \in (0,1)$, $K = 1/\varepsilon^2$, $p^+ = (1+\varepsilon)/2$ and $p^- = (1+30\varepsilon)/2$, and consider the family of distributions $\mathcal{D}^+$ (resp. $\mathcal{D}^-$) defined by the following construction:

- pick uniformly at random a partition[16] of $[n]$ in $2K$ sets of size $n/(2K)$ $A_1, \dots, A_K, B_1, \dots, B_K$;
- for all $k \in [K]$, draw independently at random $X_k \sim \mathrm{Bern}(p^+)$ (resp. $X_k \sim \mathrm{Bern}(p^-)$), and set for all $x \in A_k$, $y \in B_k$

$$\mathbf{p}^+(x) = \begin{cases} \frac{1+\alpha}{n} & \text{if } X_i = 1 \\ \frac{1}{n} & \text{o.w.} \end{cases} \qquad \text{and} \qquad \mathbf{p}^+(y) = \begin{cases} \frac{1-\alpha}{n} & \text{if } X_i = 1 \\ \frac{1}{n} & \text{o.w.} \end{cases}$$

(the pairing between A_k and B_k ensures the final measure indeed sums to one). Regardless of the choice of the initial partition, but with fluctuations over the random coin tosses $X_1, \dots, X_k$, we have that the total variation distance between a distribution $\mathbf{p}^+ \in \mathcal{D}^+$ (resp. $\mathbf{p}^- \in \mathcal{D}^-$) and uniform is on expectation what we aimed for:

$$\mathbb{E}\big[\mathrm{d}_{\mathrm{TV}}(\mathbf{p}^+, \mathbf{u})\big] = \frac{1}{2} \cdot 2 \cdot \sum_{k=1}^{K} \frac{n}{2K} \cdot \frac{\alpha}{n} p^+ = \frac{1}{2}\alpha p^+ = \frac{1}{4}$$
$$\mathbb{E}\big[\mathrm{d}_{\mathrm{TV}}(\mathbf{p}^-, \mathbf{u})\big] = \frac{1}{2} p^- \alpha = \frac{1+30\varepsilon}{1+\varepsilon} \cdot \frac{1}{4} > \frac{1}{4} + 7\varepsilon$$

[16] For convenience, it will be easier to think of the A_i's and B_i's as consecutive intervals, the first ones covering $[\frac{n}{2}]$ while the former cover $[n] \setminus [\frac{n}{2}]$ (see Fig. 4.1).

and with an additive Chernoff bound on the sum of $K = 1/\varepsilon^2$ i.i.d. choices for the X_k's, we have that for $(\mathbf{p}^+, \mathbf{p}^-)$: for any choice of the initial partition $\Pi = (A_k, B_k)_{k\in[K]}$, with probability at least $99/100$,

$$\mathrm{d}_{\mathrm{TV}}\big(\mathbf{p}^+_\Pi, \mathbf{u}\big) < \frac{1}{4} + 3\varepsilon$$
$$\mathrm{d}_{\mathrm{TV}}\big(\mathbf{p}^-_\Pi, \mathbf{u}\big) > \frac{1}{4} + 4\varepsilon$$

where by $\mathbf{p}^\pm_\Pi$ we denote the distribution defined as above, but fixing the partition for the initial step to be Π. We will further implicitly condition on this event happening; any tolerant tester for uniformity called with $(\varepsilon', \varepsilon' + c\varepsilon)$ must therefore distinguish between $\mathbf{p}^+$ and $\mathbf{p}^-$. Suppose we have such a tester $\mathcal{T}$, with (without loss of generality) exact sample complexity $q = q(\varepsilon) = o\big(\frac{1}{\varepsilon^2}\big)$.

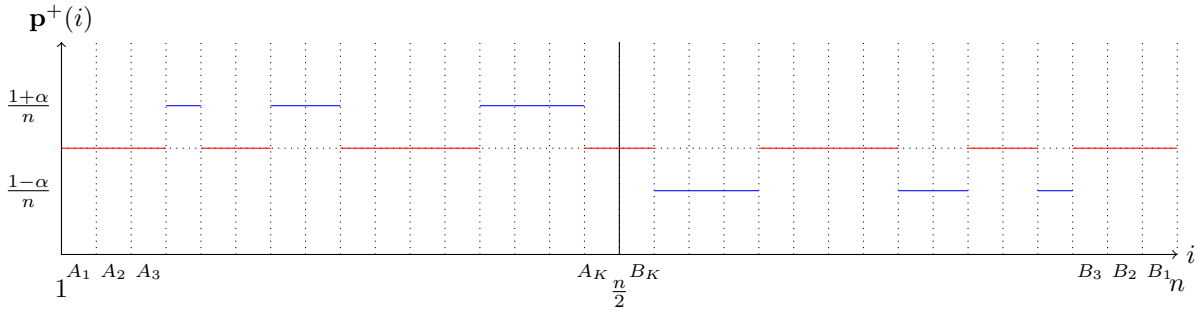

Figure 4.1: Lower bound for tolerant uniformity testing in the dual access model: The yes-instance $\mathbf{p}^+$ (for a fixed Π, taken to be consecutive intervals).

Reduction We will reduce the problem of distinguishing between (a) a p^+- and (b) a p^--biased coin to telling $\mathbf{p}^+$ and $\mathbf{p}^-$ apart.

Given $\mathsf{SAMP}_{\mathrm{coin}}$ access to i.i.d. coin tosses coming either from one of those two situations, define a distinguisher $\mathcal{A}$ as follows:

- choose uniformly at random a partition $\Pi = (A^0_k, A^1_k)_{k\in[K]}$ of $[n]$; for convenience, for any $i \in [n]$, we shall write $\pi(i)$ for the index $k \in [K]$ such that $i \in A^0_k \cup A^1_k$, and $\varsigma(i) \in \{0, 1\}$ for the part in which it belongs – so that $i \in A^{\varsigma(i)}_{\pi(i)}$ for all i;
- run $\mathcal{T}$, maintaining a set C of triples[17] $(k, \mathbf{p}^0_k, \mathbf{p}^1_k)$ (initially empty), containing the information about the (A^0_k, A^1_k) for which the probabilities have already be decided;
- EVAL: whenever asked an evaluation query on some $i \in [n]$:
  - if $\pi(i) \in C$, return $\mathbf{p}^{\varsigma(i)}_{\pi(i)}$;
  - otherwise, let $k = \pi(i)$; ask a fresh sample b_k from $\mathsf{SAMP}_{\mathrm{coin}}$ and draw a uniform random bit b'_k;

[17] Abusing the notation, we will sometimes write "$k \in C$" for "there is a triple in C with first component k."

set

$$(\mathbf{p}_k^0, \mathbf{p}_k^1) = \begin{cases} (\frac{1}{n}, \frac{1}{n}) & \text{if } b_k = 0 \\ (\frac{1+\alpha}{n}, \frac{1-\alpha}{n}) & \text{if } b_k = 1, b'_k = 1 \\ (\frac{1-\alpha}{n}, \frac{1+\alpha}{n}) & \text{if } b_k = 1, b'_k = 0 \end{cases} \qquad \text{(“Choosing the profile”)}$$

then add $(k, \mathbf{p}_k^0, \mathbf{p}_k^1)$ to C; and return $\mathbf{p}_k^{\varsigma(i)}$.

- SAMP: whenever asked a sample: let $\gamma = \frac{n}{2K} \sum_{k \in C} d_k$ the current probability mass of the “committed points”; observe that the distribution $\mathbf{p}_C$ induced by the d_k's on $\{\ i \in [n]\ :\ \pi(i) \in C\ \}$ is fully known by $\mathcal{A}$;
  - with probability γ, $\mathcal{A}$ draws $i \sim \mathbf{p}_C$ and returns it;
  - otherwise, $\mathcal{A}$ draws $k \sim \mathbf{u}([K] \setminus C)$. As before, it gets b_k from $\mathsf{SAMP}_{\text{coin}}$ and a uniform random bit b'_k; gets $(\mathbf{p}_k^0, \mathbf{p}_k^1)$ as in the EVAL case, commits to it as above by $(k, \mathbf{p}_k^0, \mathbf{p}_k^1)$ to C. Finally, it draws a random sample i from the piecewise constant distribution induced by $(\mathbf{p}_k^0, \mathbf{p}_k^1)$ on $A_k^0 \cup A_k^1$, where each $j \in A_k^0$ (resp. A_k^1) has equal probability mass $\mathbf{p}_k^0 \cdot \frac{n}{2K}$ (resp. $\mathbf{p}_k^1 \cdot \frac{n}{2K}$), and returns i.

Observe that $\mathcal{A}$ makes at most q queries to $\mathsf{SAMP}_{\text{coin}}$; provided we can argue that $\mathcal{A}$ answers $\mathcal{T}$'s queries consistently to what a corresponding $\mathbf{p}^{\pm}$ (depending on whether we are in case (a) or (b)) would look like, we can conclude.

This is the case, as (i) $\mathcal{A}$ is always consistent with what its previous answers induce on the distribution (because of the maintaining of the set C); (ii) any EVAL query on a new point exactly simulates the “on-the-fly” construction of a $\mathbf{p}^{\pm}$; and any SAMP query is either consistent with the part of $\mathbf{p}^{\pm}$ already built, or in case of a new point gets a sample exactly distributed according to the $\mathbf{p}^{\pm}$ built “on-the-fly”; this is because in any $\mathbf{p}^{\pm}$, every $A_k \cup B_k$ has same probability mass $1/(2K)$; therefore, in order to get one sample, tossing K i.i.d. coins to decide the “profiles” of *every* $A_k \cup B_k$ before sampling from the overall support $[n]$ is equivalent to first choosing uniformly at random a particular $S = A_k \cup B_k$, tossing one coin to decide *only its particular profile*, and then drawing a point accordingly from S.

In other terms, $\mathcal{A}$ will distinguish, with only $o\big(1/\varepsilon^2\big)$ i.i.d. samples, between cases (a) ($\frac{1}{2}$-biased coin) and (b) ($\frac{1}{2} + \Omega(\varepsilon)$)-biased coin with probability at least $6/10$ – task which, for ε sufficiently small, is known to require $\Omega\big(1/\varepsilon^2\big)$ samples (cf. Fact 1.4.9), thus leading to a contradiction. □

4.2.3 Entropy and support size

4.2.3.1 Additive and multiplicative estimations of entropy

In this section, we describe simple algorithms to perform both additive and multiplicative estimation (which in turns directly implies tolerant testing) of the *entropy* $H(\mathbf{p})$ of the unknown distribution $\mathbf{p}$, defined as

$$H(\mathbf{p}) \stackrel{\text{def}}{=} -\sum_{i \in [n]} \mathbf{p}(i) \log \mathbf{p}(i) \in [0, \log n]$$

We remark that Batu et al. ([23, Theorem 14]) gives a similar algorithm, based on essentially the same approach but relying on a Chebyshev bound, yielding a $(1+\gamma)$-multiplicative approximation algorithm for entropy with sample complexity $O\big((1+\gamma)^2 \log^2 n/\gamma^2 h^2\big)$, given a lower bound $h > 0$ on $H(\mathbf{p})$.

Guha et al. ([108, Theorem 5.2]) then refined their result, using as above a threshold for the estimation along with a multiplicative Chernoff bound to get the sample complexity down to $O\big(\log n/\gamma^2 h\big)$ – thus matching the $\Omega(\log n/\gamma(2+\gamma)h)$ lower bound of [23, Theorem 18]; we recall their results for multiplicative estimation of the entropy below.[18]

Theorem 4.2.14 (Upper bound [108, Theorem 5.2]]). *Fix $\gamma > 0$. In the dual access model, there exists an algorithm that, given a parameter $h > 0$ and the promise that $H(\mathbf{p}) \geq h$, estimates the entropy within a multiplicative $(1+\gamma)$ factor, with sample complexity $\Theta\Big(\frac{\log n}{\gamma^2 h}\Big)$.*

Theorem 4.2.15 (Lower bound [23, Theorem 18]). *Fix $\gamma > 0$. In the dual access model, any algorithm that, given a parameter $h > 0$ and the promise that $H(\mathbf{p}) = \Omega(h)$, estimates the entropy within a multiplicative $(1+\gamma)$ factor must have sample complexity $\Omega\Big(\frac{\log n}{\gamma(2+\gamma)h}\Big)$.*

Observe that the additive bound we give (based on a different cutoff threshold), however, still performs better in many cases, e.g. $\Delta = \gamma h > 1$ and $h > 1$; and does not require any *a priori* knowledge on a lower bound $h > 0$. Moreover, we believe that this constitutes a good illustration of the more general technique used, and a good example of what the dual model allows: approximation of quantities of the form $\mathbb{E}_{i\sim\mathbf{p}}\left[\Phi(i, \mathbf{p}(i))\right]$, where Φ is any *bounded* function of both an element of the domain and its probability mass under the distribution $\mathbf{p}$.

Additive estimate The key idea is to observe that for a distribution $\mathbf{p}$, the entropy $H(\mathbf{p})$ can be rewritten as

$$H(\mathbf{p}) = \sum_{x\in[n]} \mathbf{p}(x) \log \frac{1}{\mathbf{p}(x)} = \mathbb{E}_{x\sim\mathbf{p}}\left[\log \frac{1}{\mathbf{p}(x)}\right] \tag{4.78}$$

The quantity $\log \frac{1}{\mathbf{p}(x)}$ cannot be easily upperbounded, which we need for concentration results. However, recalling that the function $x \mapsto x\log(1/x)$ is increasing for $x \in (0, \frac{1}{e})$ (and has limit 0 when $x \to 0^+$), one can refine the above identity as follows: for any *cutoff threshold* $\tau \in (0, \frac{1}{e})$, write

$$H(\mathbf{p}) = \sum_{x:\mathbf{p}(x)\geq\tau} \mathbf{p}(x) \log \frac{1}{\mathbf{p}(x)} + \sum_{x:\mathbf{p}(x)<\tau} \mathbf{p}(x) \log \frac{1}{\mathbf{p}(x)} \tag{4.79}$$

so that

$$\begin{aligned} H(\mathbf{p}) \geq \sum_{x:\mathbf{p}(x)\geq\tau} \mathbf{p}(x) \log \frac{1}{\mathbf{p}(x)} &\geq H(\mathbf{p}) - \sum_{x:\mathbf{p}(x)<\tau} \mathbf{p}(x) \log \frac{1}{\mathbf{p}(x)} \\ &\geq H(\mathbf{p}) - n\cdot\tau \log \frac{1}{\tau} \end{aligned}$$

[18] In particular, note that translating their lower bound for additive estimation implies that the dependence on n of our algorithm is tight.

Without loss of generality, assume $\frac{\Delta}{n} < \frac{1}{2}$. Fix $\tau \stackrel{\text{def}}{=} \frac{\frac{\Delta}{n}}{10 \log \frac{n}{\Delta}}$, so that $n \cdot \tau \log \frac{1}{\tau} \leq \frac{\Delta}{2}$; and set

$$\varphi \colon y \mapsto \log \frac{1}{y} \mathbb{1}_{\{y \geq \tau\}}$$

Then, the above discussion gives us

$$H(\mathbf{p}) \geq \mathbb{E}_{x \sim \mathbf{p}}[\varphi(\mathbf{p}(x))] \geq H(\mathbf{p}) - \frac{\Delta}{2} \tag{4.80}$$

and getting an additive $\Delta/2$-approximation of $\mathbb{E}_{x \sim \mathbf{p}}[\varphi(\mathbf{p}(x))]$ is enough for estimating $H(\mathbf{p})$ within $\pm\Delta$; further, we now have

$$0 \leq \varphi(\mathbf{p}(x)) \leq \log \frac{1}{\tau} \sim \log \frac{n}{\Delta} \text{ a.s.} \tag{4.81}$$

so using an additive Chernoff bound, taking $m = \Theta\left(\frac{\log^2 \frac{n}{\Delta}}{\Delta^2}\right)$ samples $x_1, \ldots, x_m$ from SAMP_D and computing the quantities $\varphi(\mathbf{p}(x_i))$ using $\mathsf{EVAL}_{\mathbf{p}}$ implies

$$\Pr\left[\left|\frac{1}{m}\sum_{i=1}^{m} \varphi(\mathbf{p}(x_i)) - \mathbb{E}_{x \sim \mathbf{p}}[\varphi(\mathbf{p}(x))\right|] \geq \frac{\Delta}{2}\right] \leq 2e^{-\frac{\Delta^2 m}{\log^2 \frac{1}{\tau}}} \leq \frac{1}{3}$$

This leads to the following theorem:

Theorem 4.2.16. *In the dual access model, there exists an algorithm estimating the entropy up to an additive Δ, with sample complexity $\Theta\left(\frac{\log^2 \frac{n}{\Delta}}{\Delta^2}\right)$.*

or, in terms of tolerant testing:

Corollary 4.2.17. *In the dual access model, there exists an (Δ_1, Δ_2)-tolerant tester for entropy with sample complexity $\tilde{\Theta}\left(\frac{\log^2 n}{(\Delta_1 - \Delta_2)^2}\right)$.*

Proof. We describe such a $\mathcal{T}$ in Algorithm 31; the claimed query complexity is straightforward. □

Algorithm 31 Tester $\mathcal{T}$: ESTIMATE-ENTROPY

Require: $\mathsf{SAMP}_{\mathbf{p}}$ and $\mathsf{EVAL}_{\mathbf{p}}$ oracle access, parameters $0 \leq \Delta \leq \frac{n}{2}$
Ensure: Outputs $\hat{H}$ s.t. w.p. at least $2/3$, $\hat{H} \in [H(\mathbf{p}) - \Delta, H(\mathbf{p}) + \Delta/2]$
  Set $\tau \stackrel{\text{def}}{=} \frac{\frac{\Delta}{n}}{10 \log \frac{n}{\Delta}}$ and $m = \left\lceil \frac{\ln 6}{\Delta^2} \log^2 \frac{1}{\tau} \right\rceil$.
  Draw $s_1, \ldots, s_m$ from $\mathbf{p}$
  for $i = 1$ **to** m **do**
    With EVAL, get $X_i \stackrel{\text{def}}{=} \log \frac{1}{\mathbf{p}(s_i)} \mathbb{1}_{\{\mathbf{p}(s_i) \geq \tau\}}$
  end for
  return $\hat{H} \stackrel{\text{def}}{=} \frac{1}{m} \sum_{i=1}^{m} X_i$

Remark 4.2.18. The tester above can easily be adapted to be made multiplicatively robust; indeed, it is enough that the EVAL oracle only provide $(1+\gamma)$-accurate estimates $\hat{\mathbf{p}}(i)$ of the probabilities $\mathbf{p}(i)$, where γ is chosen to be $\gamma \stackrel{\text{def}}{=} \min(2^{\Delta/3} - 1, 1)$ so that the algorithm will output with high probability an additive $(\Delta/2)$-estimate

of a quantity

$$H(\mathbf{p}) \geq \mathbb{E}_{x\sim\mathbf{p}}\left[\widehat{\varphi}(x)\right] \geq \sum_{x:\mathbf{p}(x)\geq(1+\gamma)\tau} \mathbf{p}(x)\log\frac{1}{\mathbf{p}(x)} - \log(1+\gamma) \geq H(\mathbf{p}) + n\underbrace{\cdot(1+\gamma)\tau\log(1+\gamma)\tau}_{\geq -2\tau\log\frac{1}{2\tau}} - \frac{\Delta}{3}$$

and taking for instance $\tau \stackrel{\text{def}}{=} \frac{\frac{\Delta}{n}}{30\log\frac{n}{\Delta}}$ ensures the right-hand-side is at least $H(\mathbf{p}) - \frac{\Delta}{6} - \frac{\Delta}{3} = H(\mathbf{p}) - \frac{\Delta}{2}$.

4.2.3.2 Additive estimation of entropy for monotone distributions

In the previous section, we saw how to obtain an additive estimate of the entropy of the unknown distribution, using essentially $O\big(\log^2 n\big)$ sampling and evaluation queries; moreover, this dependence on n is optimal. However, one may wonder if, by taking advantage of *cumulative* queries, it becomes possible to obtain a better query complexity. We partially answer this question, focusing on a particular class of distributions for which the cumulative dual query access seems particularly well-suited: namely the class of *monotone* distributions.[19]

Before describing how this assumption can be leveraged to obtain an exponential improvement in the sample complexity for cumulative dual query algorithms, we first show that given only *dual* access to a distribution promised to be $o(1)$-close to monotone, no such speedup can hold. By establishing (see Remark 4.2.22) that the savings obtained for (close to) monotone distributions are only possible with cumulative dual access, this will yield a separation between the two oracles, proving the latter is strictly more powerful.

4.2.3.3 Lower bound for dual oracles

Theorem 4.2.19. *In the dual access model, any algorithm that estimates the entropy of distributions $O(1/\log n)$-close to monotone even to an additive constant must make $\Omega(\log n)$ queries to the oracle.*

Proof. We will define two families of distributions, $\mathcal{D}_1$ and $\mathcal{D}_2$, such that for any two $\mathbf{p}_1$, $\mathbf{p}_2$ drawn uniformly at random from $\mathcal{D}_1$ and $\mathcal{D}_2$:

1. $\mathbf{p}_1$ and $\mathbf{p}_2$ are $(2/\log n)$-close to monotone;
2. $|H(\mathbf{p}_1) - H(\mathbf{p}_2)| = 1/4$;
3. no algorithm making $o(\log n)$ queries to a dual oracle can distinguish between $\mathbf{p}_1$ and $\mathbf{p}_2$ with constant probability.

In more detail, the families are defined by the following process: for $K_n \stackrel{\text{def}}{=} n^{1/4}$, $\ell_n \stackrel{\text{def}}{=} \log n$ and $\gamma_n \stackrel{\text{def}}{=} 1/\log n$,

- Draw a subset $S \subset \{2,\dots,n\}$ of size ℓ_n uniformly at random;
- Set $\mathbf{p}_1(1) = 1-\gamma_n$, and $\mathbf{p}_1(i) = \gamma_n/\ell_n = 1/\log^2 n$ for all $i \in S$.

($\mathbf{p}_2$ is obtained similarly, but with a subset S of size $K_n\ell_n = n^{1/4}\log n$ and $\mathbf{p}_2(i) = \gamma_n/(\ell_n K_n)$) Roughly, both distributions have a very heavy first element (whose role is to "disable" sampling queries by hogging them with high probability), and then a random subset of size respectively logarithmic or polynomial, on

[19] Recall that a distribution $\mathbf{p}$ over a totally ordered domain is said to be monotone if for all $i \in [n-1]$ $\mathbf{p}(i) \geq \mathbf{p}(i+1)$

which they are uniform. To determine whether a distribution is drawn from $\mathcal{D}_1$ or $\mathcal{D}_2$, intuitively a testing algorithm has to find a point $i > 1$ with non-zero mass – and making a query on this point then gives away the type of distribution. However, since sampling queries will almost always return the very first element, finding such a $i > 1$ amounts to finding a needle in a haystack (without sampling) or to sampling many times (to get a non-trivial element) – and thus requires many queries. Before formalizing this intuition, we prove the first two items of the above claims:

Distance to monotonicity By moving all elements of S at the beginning of the support (points $2, \dots, |S|+1$), the distribution would be monotone; so in particular

$$\mathrm{d}_{\mathrm{TV}}(\mathbf{p}_i, \textsc{Monotone}) \leq \frac{1}{2} \cdot 2\,|S| \cdot \frac{\gamma_n}{|S|} = 2\gamma_n = \frac{2}{\log n}, \qquad i \in \{1, 2\}$$

Difference of entropy By their definition, for any two $\mathbf{p}_1$, $\mathbf{p}_2$, we have

$$|H(\mathbf{p}_1) - H(\mathbf{p}_2)| = \left| \sum_{i=2}^{n} \mathbf{p}_1(i) \log \mathbf{p}_1(i) - \sum_{i=2}^{n} \mathbf{p}_2(i) \log \mathbf{p}_2(i) \right| = \gamma_n \log K_n = \frac{1}{4}.$$

We now turn to the main item, the indistinguishability:

Telling $\mathbf{p}_1$ and $\mathbf{p}_2$ apart Assume we have an algorithm $\mathcal{T}$, which can estimate entropy of distributions that are $O(1/\log n)$-close to monotone up to an additive $1/3$ making $q(n) = o(\log n)$ queries; we claim that $\mathcal{T}$ cannot be correct with probability $2/3$. As argued before, we can further assume without loss of generality that $\mathcal{T}$ makes exactly $2q$ queries, q sampling queries and q evaluation ones; and that for any SAMP query, it gets "for free" the result of an evaluation query on the sample. Finally, and as the sampling queries are by definition non-adaptive, this also allows us to assume that $\mathcal{T}$ starts by making its q SAMP queries.

Let B_1 be the event that one of the q first queries results in sampling an element $i > 1$ (that is, B_1 is the event that the "hogging element" fails its role). Clearly, B_1 has same probability no matter with of the two families the unknown distribution belongs to, and

$$\Pr[\,B_1\,] = 1 - (1-\gamma_n)^q = 1 - 2^{q \log(1 - 1/\log n)} \leq 1 - 2^{-2q/\log n} = O(q/\log n) = o(1) \tag{4.82}$$

so with probability $1 - o(1)$, $\bar{B_1}$ holds. We further condition on this: i.e., the testing algorithm only saw the first element (which does not convey any information) after the sampling stage.

The situation is now as follows: unless one of its queries hits one of the relevant points in the uniform set S (call this event B_2), the algorithm will see in both case the same thing – a sequence of points with probability zero. But by construction, in both cases, the probability over the (uniform) choice of the support S to hit a relevant point with one query is either $\ell_n/(n-1) = \log n/(n-1)$ or $K_n\ell_n/(n-1) = n^{1/4}\log n/(n-1)$;

so that the probability of finding such a point in n queries is at most

$$\Pr[\, B_2 \,] \leq 1 - \left(1 - \frac{K_n \ell_n}{n-1}\right)^q = O\left(\frac{q \log n}{n^{3/4}}\right) = o(1) \tag{4.83}$$

Conditioning on $\bar{B}_1 \cup \bar{B}_2$, we get that $\mathcal{T}$ sees exactly the same transcript if the distribution is drawn from $\mathcal{D}_1$ or $\mathcal{D}_2$; so overall, with probability $1 - o(1)$ it cannot distinguish between the two cases – contradicting the assumption. □

4.2.3.4 Upper bound: exponential speedup for cumulative dual oracles

We now establish the positive result in the case of algorithms given cumulative dual query access. Note that Batu et al. [23] already consider the problem of getting a (multiplicative) estimate of the entropy of $\mathbf{p}$, under the assumption that the distribution is monotone; and describe (both in the evaluation-only and sample-only models) $\mathrm{polylog}(n)$-query algorithms for this task, which work by recursively splitting the domain in a suitable fashion to get a partition into near uniform and negligible intervals.

The main insight here (in addition to the mere fact that we allow ourself a stronger type of access to $\mathbf{p}$) is to use, instead of an *ad hoc* partition of the domain, a specific one tailored for monotone distributions, introduced by Birgé [32] – and which crucially *does not depend on the distribution itself*. Namely, for a given parameter ε we will rely on the oblivious (Birgé) decomposition $\mathcal{I}_\varepsilon$ from Definition 1.4.4, and the *flattened distribution* $\Phi_\varepsilon(\mathbf{p})$ of $\mathbf{p}$ with relation to this partition, as defined in Section 1.4:

$$\forall k \in [\ell], \forall i \in I_k, \quad \Phi_\varepsilon(\mathbf{p})(i) = \frac{\mathbf{p}(I_k)}{|I_k|}$$

We insist that while $\Phi_\varepsilon(\mathbf{p})$ (obviously) depends on $\mathbf{p}$, the partition $\mathcal{I}_\varepsilon$ itself does not; in particular, it can be computed prior to getting any sample or information about $\mathbf{p}$. Before proceeding further, we recall one of the main properties of this "Birgé flattening":

Corollary 1.4.6. *Suppose $\mathbf{p}$ is ε-close to monotone, and let $\alpha > 0$. Then $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \Phi_\alpha(\mathbf{p})) \leq 2\varepsilon + \alpha$. Furthermore, $\Phi_\alpha(\mathbf{p})$ is also ε-close to monotone.*

Finally, we shall also need the following well-known result relating total variation distance and difference of entropies (see e.g. [181], Eq. (4)):

Fact 4.2.20 (Total variation and Entropy)**.** *Let $\mathbf{p}_1$, $\mathbf{p}_2$ be two distributions on $[n]$ such that $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}_1, \mathbf{p}_2) \leq \alpha$, for $\alpha \in [0,1]$. Then $|H(\mathbf{p}_1) - H(\mathbf{p}_2)| \leq \alpha \log(n-1) + h_2(\alpha) \leq \alpha \log \frac{n}{\alpha} + (1-\alpha) \log \frac{1}{1-\alpha}$, where h_2 is the binary entropy function.*[20]

High-level idea Suppose we use the oblivious decomposition from Definition 1.4.4, with small parameter α (to be determined later), to reduce the domain into $\ell = o(n)$ intervals. Then, we can set out to approximate

[20] That is, $h_2(p) = -p \log p - (1-p) \log(1-p)$ is the entropy of a Bernoulli random variable with parameter p.

the entropy of the induced *flat* distribution – that we can efficiently simulate from the cumulative dual oracles, roughly reducing the complexity parameter from n to ℓ; it only remains to use the previous approach, slightly adapted, on this flat distribution. Of course, we have to be careful not to incur too much a loss at each step, where we first approximate $H(D)$ by $H(\bar{\mathbf{p}})$, and then specify our cutoff threshold to only consider significant contributions to $H(\bar{\mathbf{p}})$.

Details Consider the Birgé decomposition of $[n]$ into $\ell = \Theta(\log(n\alpha)/\alpha)$ intervals (for α to be defined shortly). Theorem 1.4.5 ensures the corresponding (unknown) flattened distribution $\bar{\mathbf{p}}$ is α-close to $\mathbf{p}$; which, by the fact above, implies that

$$|H(\bar{\mathbf{p}}) - H(D)| \leq \alpha \left(\log \frac{n}{\alpha} + 2\right) \tag{4.84}$$

Taking $\alpha \stackrel{\text{def}}{=} \Theta(\Delta/\log n)$, the right-hand-side is at most $\Delta/2$; so that it is now sufficient to estimate $H(\bar{\mathbf{p}})$ to $\pm\Delta/2$, where both sampling and evaluation access to $\bar{\mathbf{p}}$ can easily be simulated from the $\mathsf{CEVAL}_{\mathbf{p}}$ and $\mathsf{SAMP}_{\mathbf{p}}$ oracles. But although $\bar{\mathbf{p}}$ is a distribution on $[n]$, its "actual" support is morally only the $\ell = \tilde{\Theta}\big(\log^2 n/\Delta\big)$. Indeed, we may write the entropy of $\bar{\mathbf{p}}$ as

$$H(\bar{\mathbf{p}}) = \sum_{k=1}^{\ell} \sum_{x \in I_k} \bar{\mathbf{p}}(x) \log \frac{1}{\bar{\mathbf{p}}(x)} = \sum_{k=1}^{\ell} \sum_{x \in I_k} \frac{\mathbf{p}(I_k)}{|I_k|} \log \frac{|I_k|}{\mathbf{p}(I_k)} = \sum_{k=1}^{\ell} \mathbf{p}(I_k) \log \frac{|I_k|}{\mathbf{p}(I_k)} = \mathbb{E}_{k \sim \bar{\mathbf{p}}} \left[\log \frac{1}{\mathbf{p}_k}\right]$$

where $d_k = \frac{\mathbf{p}(I_k)}{|I_k|} \approx (1+\alpha)^{-k}\mathbf{p}(I_k)$.

As in the previous section, we can then define a cutoff threshold τ (for d_k) and only estimate $\mathbb{E}_{k \sim \bar{\mathbf{p}}} \left[\log \frac{1}{\mathbf{p}_k} \mathbb{1}_{\{\mathbf{p}_k \geq \tau\}}\right]$, for this purpose, we need $\ell \cdot \tau \log 1/\tau$ to be at most $\Delta/4$, i.e.

$$\tau \stackrel{\text{def}}{=} \Theta\left(\frac{\Delta/\ell}{\log \Delta/\ell}\right) = \tilde{\Theta}\left(\frac{\Delta^2}{\log^2 n}\right)$$

and to get with high probability a $\Delta/4$-approximation, it is as before sufficient to make $m = O\big(\Delta^2/\log^2(1/\tau)\big) = \tilde{O}\left(\frac{\log^2 \frac{\log n}{\Delta}}{\Delta^2}\right)$ queries.

Theorem 4.2.21. *In the cumulative dual access model, there exists an algorithm for monotone distributions estimating the entropy up to an additive Δ, with sample complexity $\tilde{O}\left(\log^2 \frac{\log n}{\Delta}/\Delta^2\right)$.*

Remark 4.2.22. We remark that the above result and algorithm (after some minor changes in the constants) still applies if $\mathbf{p}$ is only guaranteed to be $O(1/\log n)$-*close* to monotone; indeed, as stated in Corollary 1.4.6, the oblivious decomposition is (crucially) robust, and $\bar{\mathbf{p}}$ will still be $O(\varepsilon)$-close to $\mathbf{p}$.

4.2.3.5 Additive estimation of support size

We now turn to the task of estimating the effective support size of the distribution: given the promise that $\mathbf{p}$ puts on every element of the domain either no weight or at least some minimum probability mass $1/n > 0$, the goal is to output a good estimate (up to $\pm\varepsilon n$) of the number of elements in the latter situation.

Theorem 4.2.23. *In the dual access model, there exists an algorithm* ESTIMATE-SUPPORT *that, on input a threshold $n \in \mathbb{N}^*$ and a parameter $\varepsilon > 0$, and given access to a distribution $\mathbf{p}$ (over an arbitrary set) satisfying*

$$\min_{x \in \mathrm{supp}(\mathbf{p})} \mathbf{p}(x) \geq \frac{1}{n}$$

estimates the support size $|\mathrm{supp}(\mathbf{p})|$ up to an additive εn, with query complexity $O\big(\frac{1}{\varepsilon^2}\big)$.

Proof. Write $k \stackrel{\text{def}}{=} |\mathrm{supp}(\mathbf{p})|$. We describe ESTIMATE-SUPPORT which outputs (w.p. at least 2/3) an estimate as required:

If $\varepsilon > \frac{2}{\sqrt{n \ln 3n}}$: The algorithm will draw $m = \lceil \frac{4}{\varepsilon^2} \rceil$ samples $x_1, \ldots, x_m$ from $\mathbf{p}$, query their probability mass $\mathbf{p}(x_i)$, and output $\hat{k} = \lceil Y \rceil$, where

$$Y \stackrel{\text{def}}{=} \frac{1}{m} \sum_{i=1}^{m} \frac{\mathbb{1}_{\{\mathbf{p}(x_i) \geq \frac{1}{n}\}}}{\mathbf{p}(x_i)}$$

If $\varepsilon \leq \frac{2}{\sqrt{n \ln 3n}}$: in this case, ESTIMATE-SUPPORT just draws $m = n \ln 3n = O\big(\frac{1}{\varepsilon^2}\big)$ samples $x_1, \ldots, x_m$ from $\mathbf{p}$, and returns the number $\hat{k}$ of distincts elements it got (no query access is needed in this case).

Analysis In the first (and interesting) case, let ϕ be the function defined over the coset of $\mathbf{p}$ by $\phi(x) = \frac{1}{\mathbf{p}(x)} \cdot \mathbb{1}_{\{\mathbf{p}(x) \geq \frac{1}{n}\}}$, so that $\mathbb{E}_{x \sim \mathbf{p}}[\phi(x)] = \sum_{x : \mathbf{p}(x) > \frac{1}{n}} \mathbf{p}(x) \cdot \frac{1}{\mathbf{p}(x)} = \big|\{\, x : \mathbf{p}(x) > \frac{1}{n} \,\}\big| = k$; and as the r.v. $\phi(x_1), \ldots, \phi(x_m)$ are i.i.d. and taking value in $[0, n]$, an additive Chernoff bound yields

$$\Pr\Big[\, |Y - k| > \frac{\varepsilon n}{2} \,\Big] \leq 2e^{-\frac{\varepsilon^2 m}{2}} < \frac{1}{3}$$

Conditioned on this not happening, $k + \frac{\varepsilon}{2} n \leq Y \leq \hat{k} \leq Y + 1 \leq k + \frac{\varepsilon}{2} n + 1 \leq k + \varepsilon n$ (as $\varepsilon > \frac{2}{n}$), and $\hat{k}$ is as stated.

Turning now to the second case, observe first that the promise on $\mathbf{p}$ implies that $1 \leq k \leq n$. It is sufficient to bound the probability that an element of the support is *never* seen during the m draws – let F denote this event. By a union bound,

$$\Pr[\, F \,] \leq k \cdot \left(1 - \frac{1}{n}\right)^m \leq n e^{n \ln(3n) \ln(1 - \frac{1}{n})} \leq n e^{-\ln 3n} = \frac{1}{3}$$

so w.p. at least $2/3$, every element of the support is drawn, and ESTIMATE-SUPPORT returns (exactly) k. □

4.2.3.6 Lower bound

In this subsection, we show that the upper bound of Theorem 4.2.23 is tight.

Theorem 4.2.24. *In the dual access model, ε-additively estimating support size requires query complexity $\Omega\big(\frac{1}{\varepsilon^2}\big)$.*

Proof. Without loss of generality, suppose n is even, and let $k = \frac{n}{2}$. For any $p \in [0,1]$, consider the following process Φ_p, which yields a random distribution $\mathbf{p}_p$ on $[n]$ (See Fig. 4.2):

- draw k i.i.d. random variables $X_1, \ldots, X_k \sim \mathrm{Bern}(p)$;
- for $i \in [k]$, set $\mathbf{p}(i) = \frac{1}{n}(1 + X_i)$ and $\mathbf{p}(n-i) = \frac{1}{n}(1 - X_i)$

Note that by construction $\mathbf{p}(i) + \mathbf{p}(n-i) = \frac{2}{n}$ for all $i \in [k]$.

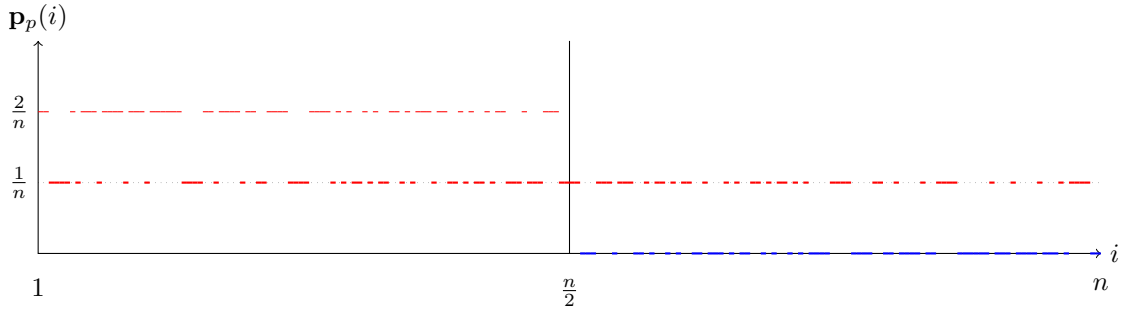

Figure 4.2: Lower bound for support size estimation in the dual model: An instance of distribution $\mathbf{p}_p$ with $p = 4/10$.

Define now, for any $\varepsilon \in (0, 1/6)$, the families of distributions $\mathcal{D}^+$ and $\mathcal{D}^-$ induced the above construction, taking p to be respectively $p^+ \stackrel{\text{def}}{=} \frac{1}{2}$ and $p^- \stackrel{\text{def}}{=} \frac{1}{2} - 6\varepsilon$. Hereafter, by $\mathbf{p}^+$ (resp. $\mathbf{p}^-$), we refer to a distribution from $\mathcal{D}^+$ (resp. $\mathcal{D}^-$) generated randomly as above (we assume further, without loss of generality, that $n \gg 1/\varepsilon^2$):

$$\mathbb{E}\big[\mathrm{supp}\big(\mathbf{p}^+\big)\big] = n - kp^+ = n\left(1 - \frac{p^+}{2}\right) = \frac{3}{4}n$$
$$\mathbb{E}\big[\mathrm{supp}\big(\mathbf{p}^-\big)\big] = n - kp^- = n\left(1 - \frac{p^-}{2}\right) = \left(\frac{3}{4} + 3\varepsilon\right)n$$

and, with an additive Chernoff bound,

$$\Pr\left[\mathrm{supp}\big(\mathbf{p}^+\big) \geq \frac{3}{4}n + \frac{\varepsilon}{2}n\right] \leq e^{-\frac{\varepsilon^2 n}{2}} < \frac{1}{100}$$
$$\Pr\left[\mathrm{supp}\big(\mathbf{p}^-\big) \leq \frac{3}{4}n + \frac{5\varepsilon}{2}n\right] \leq e^{-\frac{\varepsilon^2 n}{2}} < \frac{1}{100}$$

We hereafter condition on these events E^+ and E^- every time we consider a given $\mathbf{p}^+$ or $\mathbf{p}^-$, and set for convenience $s^+ \stackrel{\text{def}}{=} \frac{3}{4}(n + 2\varepsilon)$, $s^- \stackrel{\text{def}}{=} \frac{3}{4}(n + 10\varepsilon)$.

Reduction We shall once again reduce the problem of distinguishing between (a) a fair coin and (b) an $(\frac{1}{2} - 6\varepsilon)$-biased coin to the problem of approximating the support size: suppose by contradiction we have a tester $\mathcal{T}$ for the latter problem, making $q = o\big(\frac{1}{\varepsilon^2}\big)$ queries on input ε.

Given parameter $\varepsilon \in (0, 1/100)$ and $\mathsf{SAMP}_{\mathrm{coin}}$ access to i.i.d. coin tosses coming from one of those two situations ($(p^+ = \frac{1}{2}$, or $p^- = \frac{1}{2} - 6\varepsilon)$, define a distinguisher $\mathcal{A}$ as follows:

- after picking an even integer $n \gg 1/\varepsilon^2$, $\mathcal{A}$ will maintain a set $C \subseteq [n] \times \{0, \frac{1}{n}, \frac{2}{n}\}$ (initially empty), and run $\mathcal{T}$ as a subroutine with parameter ε;
- EVAL: when $\mathcal{T}$ makes an evaluation query on a point $i \in [n]$
  - if i has already been committed to (there is a pair (i, d_i) in C), it returns d_i;
  - otherwise, it asks for a sample b from $\mathsf{SAMP}_{\text{coin}}$, and sets

$$d_i = \begin{cases} \frac{1}{n} & \text{if } b = 0 \\ \frac{2}{n} & \text{if } b = 1 \text{ and } i \in [k] \\ 0 & \text{if } b = 1 \text{ and } i \in [n] \setminus [k] \end{cases}$$

    before adding (i, d_i) and $(n - i, \frac{2}{n} - d_i)$ to C and returning d_i.
- SAMP: when $\mathcal{T}$ makes an sampling query, $\mathcal{A}$ draws u.a.r. $i \sim [k]$, and then proceeds as in the EVAL case to get d_i and d_{n-i} (that is, if they are not in C, it first generates them from a $\mathsf{SAMP}_{\text{coin}}$ query and commits to them); and then, it returns i w.p. $(nd_i)/2$, and $n - i$ w.p. $(nd_{n-i})/2$.

It is easy to see that the process above exactly simulates dual access to a distribution $\mathbf{p}$ generated either according to Φ_{p^+} or Φ_{p^-} – in particular, this is true of the sampling queries because each pair $(i, n - i)$ has same total mass $\frac{2}{n}$ under any such distribution, so drawing from $\mathbf{p}$ is equivalent to drawing uniformly $i \in [k]$, and then returning at random i or $n - i$ according to the conditional distribution of $\mathbf{p}$ on $\{i, n - i\}$.

Furthermore, the number of queries to $\mathsf{SAMP}_{\text{coin}}$ is at most the number of queries made by $\mathcal{T}$ to $\mathcal{A}$, that is $o\big(\frac{1}{\varepsilon^2}\big)$. Conditioning on E^+ (or E^-, depending on whether we are in case (a) or (b)), the distribution $\mathbf{p}$ has support size at most s^+ (resp. at least s^-). As the estimate $\hat{s}$ that $\mathcal{T}$ will output will, with probability at least $2/3$, be εn-close to the real support size, and as $s^- - s^+ = 2\varepsilon n$, $\mathcal{A}$ will distinguish between cases (a) and (b) with probability at least $2/3 - 2/100 > 6/10$ – contradicting the fact that $\Omega\big(1/\varepsilon^2\big)$ samples are required to distinguish between a fair and a $(\frac{1}{2} - 6\varepsilon)$-biased coin with this probability. □

Chapter 5

Correcting Properties of Distributions: Changing the Goal

"For the Snark's a peculiar creature, that won't
Be caught in a commonplace way.
Do all that you know, and try all that you don't:
Not a chance must be wasted to-day!"

Lewis Carroll, *The Hunting of the Snark*

5.1 Introduction

Data consisting of samples from distributions is notorious for reliability issues: Sample data can be greatly affected by noise, calibration problems or other faults in the sample recording process; portions of data may be lost; extraneous samples may be erroneously recorded. Such noise may be completely random, or may have some underlying structure. To give a sense of the range of difficulties one might have with sample data, we mention some examples: A sensor network which tracks traffic data may have dead sensors which transmit no data at all, or other sensors that are defective and transmit arbitrary numbers. Sample data from surveys may suffer from response rates that are correlated with location or socioeconomic factors. Sample data from species distribution models are prone togeographic location errors [110].

Statisticians have grappled with defining a methodology for working with distributions in the presence of noise by *correcting* the samples. If, for example, you know that the uncorrupted distribution is Gaussian, then it would be natural to correct the samples of the distribution to the nearest Gaussian. The challenge in defining this methodology is: how do you correct the samples if you do not know much about the original uncorrupted distribution? To analyze distributions with noise in a principled way, approaches have included *imputation* [125, 160, 157] for the case of missing or incomplete data, and *outlier detection and removal* [109, 18, 115] to handle "extreme points" deviating significantly from the underlying distribution. More generally, the question of coping with the *sampling bias* inherent to many strategies (such as opportunity sampling) used in studying rare events or species, or with inaccuracies in the reported data, is a key challenge in many of the natural and social sciences (see e.g. [163, 161, 139]). While these problems are usually dealt with drawing on additional knowledge or by using specific modeling assumptions, no general procedure is known that addresses them in a systematic fashion.

In this work, we propose a methodology which is based on using *known structural properties* of the distribution to design *sampling correctors* which "correct" the sample data. While assuming these structural

properties is in itself a type of modeling, it is in general much weaker than postulating a strict form of the data (e.g., that it follows a linear model perturbed by Gaussian noise). Examples of structural properties which might be used to correct samples include the property of being bimodal, a mixture of several Gaussians, a mixture of piecewise-polynomial distributions, or an independent joint distribution. Within this methodology, the main question is: how best can one output samples of a distribution in which on one hand, the structural properties are restored, and on the other hand, the corrected distribution is close to the original distribution? We show that this task is intimately connected to distribution learning tasks, but we also give instances in which such tasks can be performed strictly more efficiently.

5.1.1 Our model

We introduce two (related) notions of algorithms to correct distributions: *sampling correctors* and *sampling improvers*. Although the precise definitions are deferred to Section 5.2, we describe and state informally what we mean by these. In what follows, Ω is a finite domain, $\mathcal{P}$ is any fixed property of distributions, i.e., a subset of distributions, over Ω and distances between distributions are measured according to their *total variation distance*.

A *sampling corrector* for $\mathcal{P}$ is a randomized algorithm which gets samples from a distribution $\mathbf{p}$ guaranteed to be ε-close to having property $\mathcal{P}$, and outputs a sample from a "corrected distribution" $\tilde{\mathbf{p}}$ which, with high probability, (a) *has* the property; and (b) is still close to the original distribution $\mathbf{p}$ (i.e., within distance ε_1). The *sample complexity* of such a corrector is the number of samples it needs to obtain from $\mathbf{p}$ in order to output one from $\tilde{\mathbf{p}}$.

To make things concrete, we give a simple example of correcting independence of distributions over a product space $[n] \times [m]$. For each pair of samples (x, y) and (x', y') from a distribution $\mathbf{p}$ which is ε-close to independent, output *one* sample (x, y'). As x and y' are independent, the resulting distribution clearly has the property; and it can be shown that if $\mathbf{p}$ was indeed ε-close to independent, then the distribution of (x, y') will indeed be 3ε-close to $\mathbf{p}$ [158]. (Whether this sample complexity can be reduced further to $q < 2$, even on average, is an open question.)

Note that in some settings it may be too much to ask for complete correction (or may even not be the most desirable option). For this reason, we also consider the weaker notion of *sampling improvers*, which is similar to a sampling corrector but is only required to transform the distribution into a new distribution which is *closer* to having the property $\mathcal{P}$.

One naive way to solve these problems, the "learning approach," is to approximate the probability mass function of $\mathbf{p}$, and find a candidate $\tilde{\mathbf{p}} \in \mathcal{P}$. Since we assume we have a complete description of $\tilde{\mathbf{p}}$, we can then output samples according to $\tilde{\mathbf{p}}$ without further access to $\mathbf{p}$. In general, such an approach can be very inefficient in terms of time complexity. However, if there is an *efficient* agnostic proper learning algorithm[1] for $\mathcal{P}$, we show that this approach can lead to efficient sampling correctors. For example, we use such an approach to give sampling correctors for the class of monotone distributions.

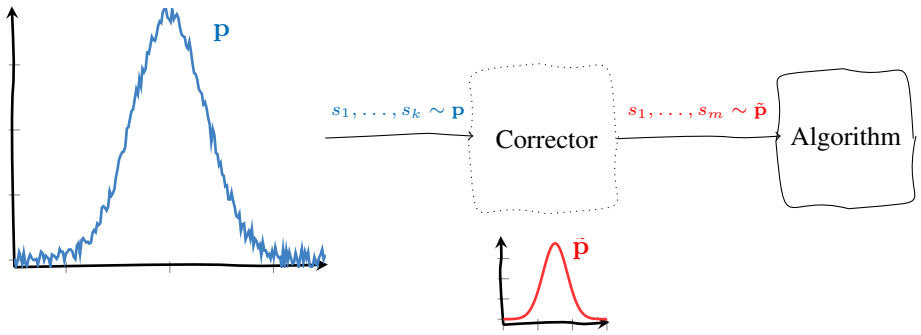

Figure 5.1: A sampling corrector acts as a filter being an imperfect source of data $\mathbf{p}$, which is only *close* to having a prespecified property $\mathcal{P}$, and an algorithm which requires data from a source with this property.

In our model, we wish to optimize the following two parameters of our correcting algorithms: The first parameter is the number of samples of $\mathbf{p}$ needed to output samples of $\tilde{\mathbf{p}}$. The second parameter is the number of *additional* truly random bits needed for outputting samples of $\tilde{\mathbf{p}}$. Note that in the above learning approach, the dependence on each of these parameters could be quite large. Although these parameters are not independent of each other (if $\mathbf{p}$ is of high enough entropy, then it can be used to simulate truly random bits), they can be thought of as complementary, as one typically will aim at a tradeoff between the two. Furthermore, a parsimonious use of extra random bits may be crucial for some applications, while in others the correction of the data itself is the key factor; for this reason, we track each of the parameters separately. For any property $\mathcal{P}$, the main question is whether one can achieve improved complexity in terms of these parameters over the use of the naive (agnostic) learning approach for $\mathcal{P}$.

5.1.2 Our results

Throughout this paper, we will focus on two particular properties of interest, namely *uniformity* and *monotonicity*. The first one, arguably one of the most natural and illustrative properties to be considered, is nonetheless deeply challenging in the setting of randomness scarcity. As for the second, not only does it provide insight in the workings of sampling correctors as well as non-trivial connections and algorithmic results, but is also one of the most-studied classes of distributions in the statistics and probability literature, with a body of work covering several decades (see e.g. [107, 32, 19, 63], or [74] for a detailed list of references). Moreover, recent work on distribution testing [74, 51] shows strong connections between monotonicity and a wide range of other properties, such as for instance log-concavity, Monotone Hazard Risk and Poisson Binomial Distributions. This gives evidence that the study of monotone distributions may have direct implications for correction of many of these "shape-constrained properties."

[1]Recall that a *learning algorithm* for a class of distributions $\mathcal{C}$ is an algorithm which gets independent samples from an unknown distribution $\mathbf{p} \in \mathcal{C}$; and on input ε must, with high probability, output a hypothesis which is ε-close to $\mathbf{p}$ in total variation distance. If the hypotheses the algorithm produces are guaranteed to belong to $\mathcal{C}$ as well, we call it a *proper learning algorithm*. Finally, if the – not-necessarily proper – algorithm is able to learn distributions that are only *close* to $\mathcal{C}$, returning a hypothesis at a distance at most $\mathrm{OPT} + \varepsilon$ from $\mathbf{p}$ – where OPT is the distance from $\mathbf{p}$ to the class, it is said to be *agnostic*. For a formal definition of these concepts, the reader is referred to Section 1.2 and Section 5.4.2.

Sampling correctors, learning algorithms and property testing algorithms We begin by showing implications of the existence of sampling correctors for the existence of various types of learning and property testing algorithms in other models. We first show in Theorem 5.4.1 that learning algorithms for a distribution class imply sampling correctors for distributions in this class (under *any* property to correct) with the same sample complexity, though not necessarily the same running time dependency. However, when efficient agnostic proper learning algorithms for a distribution class exist, we show that there are efficient sampling correctors for the same class. In [32, 55] efficient algorithms for agnostic learning of concise representations for several families of distributions are given, including distributions that are monotone, k-histograms, Poisson binomial, and sums of k independent random variables. Not all of these algorithms are proper.

Next, we show in Theorem 5.4.4 that the existence of (a) an efficient learning algorithm, as e.g. in [117, 56, 64, 73], and (b) an efficient sampling corrector for a class of distributions implies an efficient *agnostic* learning algorithm for the same class of distributions. It is well known that agnostic learning can be much harder than non-agnostic learning, as in the latter the algorithm is able to leverage structural properties of the class $\mathcal{C}$. Thus, by the above result we also get that any agnostic learning lower bounds can be used to obtain sampling corrector lower bounds.

Our third result in this section, Theorem 5.4.7, shows that an efficient property tester, an efficient distance estimator (which computes an additive estimate of the distance between two distributions) and an efficient sampling corrector for a distribution class imply a tolerant property tester with complexity equal to the complexity of correcting the number of samples required to run both the tester and estimator.[2] As tolerant property testing can be much more difficult than property testing [104, 20, 138, 167], this gives a general purpose way of getting both upper bounds on tolerant property testing and lower bounds on sampling correctors.

We describe how these results can be employed in Section 5.4, where we give specific applications in achieving improved property testers for various properties.

Is sampling correction easier than learning? We next turn to the question of whether there are natural examples of sampling correctors whose query complexity is asymptotically smaller than that of distribution learning algorithms for the same class. While the sample complexity of learning monotone distributions is known to be $\Omega(\log n)$ [32] (this lower bound on the sample and query complexity holds even when the algorithm is allowed both to make queries to the cumulative distribution function as well as to access samples of the distribution), we present in Section 5.5.2 an oblivious sampling corrector for monotone distributions whose sample complexity is $O(1)$ and that corrects error that is smaller than $\varepsilon \leq O\big(1/\log^2 n\big)$. This is done by first implicitly approximating the distribution by a "histogram" on only a small number of intervals, using ingredients from [32]. This (very close) approximation can then be combined, still in an oblivious way, with a carefully chosen slowly decreasing distribution, so that the resulting mixture is not only guaranteed to be

[2] Recall that the difference between testing and tolerant testing lies in that the former asks to distinguish whether an unknown distribution *has* a property, or is far from it, while the latter requires to decide whether the distribution is *close* to the property versus far from it. (See Section 1.2 for the rigorous definition.)

monotone, but also close to the original distribution.

It is open whether there exist sampling correctors for monotone distributions with sample complexity $o\big((\log n)/\varepsilon^3\big)$ that can correct arbitrary error $\varepsilon \in (0,1)$, thus beating the sample complexity of the "learning approach." (We note however that a logarithmic dependence on n is inherent when $\varepsilon = \omega(1/\log n)$, as pointed out to us by Paul Valiant [173].)

Assuming a stronger type of access to the unknown distribution – namely, query access to its cumulative distribution function (cdf) as in [23, 50], we describe in Section 5.5.3 a sampling corrector for monotonicity with (expected) query complexity $O\big(\sqrt{\log n}\big)$ which works for arbitrary $\varepsilon \in (0,1)$. At a high-level, our algorithm combines the "succinct histogram" technique mentioned above with a two-level bucketing approach to correct the distribution first at a very coarse level only (on "superbuckets"), and defer the finer corrections (within a given superbucket) to be made on-the-go at query time. The challenge in this last part is that one must ensure that all of these disjoint local corrections are consistent with each other – and crucially, *with all future sample corrections.* To achieve this, we use a "boundary correction" subroutine which fixes potential violations between two neighboring superbuckets by evening out the boundary differences. To make it possible, we use rejection sampling to allocate adaptively an extra "budget" to each superbucket that this subroutine can use for corrections.

Restricted error models Since many of the sampling correction problems are difficult to solve in general, we suggest error models for which more efficient sampling correction algorithms may exist. A first class of error models, which we refer to as *missing data errors*, is introduced in Section 5.6 and defined as follows – given a distribution over $[n]$, all samples in some interval $[i,j]$ for $1 < i < j < n$ are deleted. Such errors could correspond to samples from a sensor network where one of the sensors ran out of power; emails mistakenly deleted by a spam filter; or samples from a study in which some of the paperwork got lost. Whenever the input distribution $\mathbf{p}$, whose distance from monotonicity is $\varepsilon \in (0,1)$, falls under this model, we give a sampling improver that is able to find a distribution both ε_2-close to monotone and $O(\varepsilon)$-close to the original using $\tilde{O}\big(1/\varepsilon_2^3\big)$ samples. The improver works in two stages. In the "preprocessing stage," we detect the location of the missing interval (when the missing weight is sufficiently large) and then estimate its missing weight, using a "learning through testing" approach from [63] to keep the sample complexity under control. In the second stage, we give a procedure by which the algorithm can use its knowledge of the estimated missing interval to correct the distribution by rejection sampling.

Randomness Scarcity We then consider the case where only a limited amount of randomness (other than the input distribution) is available, and optimizing its use, possibly at the cost of worse parameters and/or sample complexity of our sampling improvers, is crucial. This captures situations where generating the random bits the algorithm use is either expensive[3] (as in the case of physical implementations relying on devices, such as Geiger counters or Zener diodes) or undesirable (e.g., when we want the output distribution to be a

[3] On this topic, see for instance the discussion in [124, 116], and references therein.

deterministic function of the input data, for the sake of reproducibility or parallelization). We focus on this setting in Section 5.7, and provide sampling correctors and improvers for uniformity that use samples *only* from the input distribution. For example, we give a sampling improver that, given access to distribution ε-close to uniform, grants access to a distribution ε_2-close to uniform distribution and has *constant* sample complexity $O_{\varepsilon,\varepsilon_2}(1)$. We achieve this by exploiting the fact that the uniform distribution is not only an absorbing element for convolution in Abelian groups, but also an *attractive fixed point* with high convergence rate. That is, by convolving a distribution with itself (i.e., summing independent samples modulo the order of the group) one gets very quickly close to uniform. Combining this idea with a different type of improvement (based on a von Neumann-type "trick") allows us to obtain an essentially optimal tradeoff between closeness to uniform and to the original distribution.

5.1.3 Open problems

Correcting vs. Learning A main direction of interest would be to obtain more examples of properties for which correcting is strictly more efficient than (agnostic or non-agnostic) learning. Such examples would be insightful even if they are more efficient only in terms of the number of samples required from the original distribution, without considering the additional randomness requirements for generating the distribution. More specifically, one may ask whether there exists a sampling corrector for monotonicity of distributions (i.e., one that beats the learning bound from Lemma 5.5.1) for all $\epsilon < 1$ which uses at most $o\big((\log n)/\varepsilon^3\big)$ samples from the original distribution per sample output of the corrected distribution. Other properties of interest, among many, include log-concavity of distributions, having a piecewise-constant density (i.e., being a k-histogram for some fixed value k), or being a Poisson Binomial Distribution.

The power of additional queries Following the line of work pursued in [54, 48, 50] (in the setting of distribution testing), it is natural in many situations to consider additional types of queries to the input distribution: e.g., either *conditional queries* (getting a sample conditioned on a specific subset of the domain) or *cumulative queries* (granting query access to the cumulative distribution function, besides the usual sampling). By providing algorithms with this extended access to the underlying probability distribution, can one obtain faster sampling correctors for specific properties, as we do in Section 5.5.3 in the case of monotonicity?

Confidence boosting Suppose that there exists, for some property $\mathcal{P}$, a sampling improver $\mathcal{A}$ that only guarantees a success probability[4] of $2/3$. Using $\mathcal{A}$ as a black-box, can one design a sampling improver $\mathcal{A}'$ which succeeds with probability $1-\delta$, for any δ?

More precisely, let $\mathcal{A}$ be a batch improver for $\mathcal{P}$ which, when queried, makes $q(\varepsilon_1,\varepsilon_2)$ queries and provides

[4]We note that the case of interest here is of batch sampling improvers: indeed, in order to generate a single draw, a sampling improver acts in a non-trivial way only if the parameter ε is greater than its failure probability δ. If not, a draw from the original distribution already satisfies the requirements.

$t \geq 1$ samples, with success probability at least $2/3$. Having black-box access to $\mathcal{A}$, can we obtain a batch improver $\mathcal{A}'$ which on input $\delta > 0$ provides $t' \geq 1$ samples, with success probability at least $1 - \delta$? If so, what is the best t' one can achieve, and what is the minimum query complexity of $\mathcal{A}'$ one can get (as a function of $q(\cdot, \cdot)$, t' and δ)?

This is known for property testing (by running the testing algorithm independently $O(\log(1/\delta))$ times and taking the majority vote), as well as for learning (again, by running the learning algorithm many times, and then doing hypothesis testing, e.g. *à la* [66, Theorem 19]). However, these approaches do not straightforwardly generalize to sampling improvers or correctors, respectively because the output is not a single bit, and as we only obtain a sequence of samples (instead of an actual, fully-specified hypothesis distribution).

5.1.4 Previous work

Dealing with noisy or incomplete datasets has been a challenge in Statistics and data sciences, and many methods have been proposed to handle them. One of the most widely used, *multiple imputation* (one of many variants of the general paradigm of *imputation*) was first introduced by Rubin [153] and consists of the creation of several complete datasets from an incomplete one. Specifically, one first obtains these new datasets by filling in the missing values randomly according to a maximum likelihood distribution computed from the observations and a modeling assumption made on the data. The parameters of this model are then updated using the new datasets and the ML distribution is computed again. This resembles the Expectation-Maximization (EM) algorithm, which can also be used for similar problems, as e.g. in [76]. After a few iterations, one can get both accurate parameter estimates and the right distribution to sample data from. Assuming the assumptions chosen to model the data did indeed reflect its true distribution, and that the number of these new datasets was large enough, this can be shown to yield statistically accurate and unbiased results [160, 125].

From a Theoretical Computer Science perspective, the problem of local correction of data has received much attention in the contexts of self-correcting programs, locally correctable codes, and local filters for graphs and functions over $[n]^d$ (some examples include [36, 179, 7, 159, 29, 118]). To the best of our knowledge, this is the first work to address the correction of data from distributions. (We observe that Chakraborty et al. consider in [53] a different question, although of a similar distributional flavor: namely, given query access to a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ which is close to a k-junta f^*, they show how to approximately generate uniform PAC-style samples of the form $\langle x, g^*(x)\rangle$ where $x \in \{0,1\}^k$ and g^* is the function underlying f^*. They then describe how to apply this "noisy sampler" primitive to test whether a function is close to being a junta.)

In this work, we show that the problem of estimating distances between distributions is related. There has been much work on this topic, but we note the following result: [74] show how to estimate the total variation distance between k-modal probability distributions.[5] The authors give a reduction of their problem into one

[5] A probability distribution $\mathbf{p}$ is *k-modal* if there exists a partition of $[n]$ in k intervals such that $\mathbf{p}$ is monotone (increasing or decreasing) on each.

with logarithmic size, using a result by Birgé on monotone distributions [32]. In particular, one can partition the domain $\Omega = [n]$ into $\log n/\varepsilon$ intervals in a oblivious way, such that the "flattening" of any monotone distribution according to that interval is $O(\varepsilon)$-close to the original one. We use similar ideas in order to obtain some of the results in the present paper.

It is instructive to compare the goal of our model of distribution sampling correctors to that of extractors: in spite of many similarities, the two have essential differences and the results are in many cases incomparable. We defer this discussion to Section 5.7.1.

5.2 Our model: definitions

In this section, we state the precise definitions of sampling correctors, improvers and batch sampling improvers. To get an intuition, the reader may think for instance of the parameter ε_1 below as being 2ε, and the error probability δ as $1/3$. Although all definitions are presented in terms of the total variation distance, analogous definitions in terms of other distances can also be made.

Definition 5.2.1 (Sampling Corrector)**.** Fix a given property $\mathcal{P}$ of distributions on Ω. An $(\varepsilon, \varepsilon_1)$-*sampling corrector for* $\mathcal{P}$ is a randomized algorithm which is given parameters $\varepsilon, \varepsilon_1 \in (0, 1]$ such that $\varepsilon_1 \geq \varepsilon$ and $\delta \in [0, 1]$, as well as sampling access to a distribution $\mathbf{p}$. Under the promise that $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathcal{P}) \leq \varepsilon$, the algorithm must provide, with probability at least $1 - \delta$ over the samples it draws and its internal randomness, sampling access to a distribution $\tilde{\mathbf{p}}$ such that

(i) $\tilde{\mathbf{p}}$ is close to $\mathbf{p}$: $\mathrm{d}_{\mathrm{TV}}(\tilde{\mathbf{p}}, \mathbf{p}) \leq \varepsilon_1$;

(ii) $\tilde{\mathbf{p}}$ has the property: $\tilde{\mathbf{p}} \in \mathcal{P}$.

In other terms, with high probability the algorithm will simulate exactly a sampling oracle for $\tilde{\mathbf{p}}$. The query complexity $q = q(\varepsilon, \varepsilon_1, \delta, \Omega)$ of the algorithm is the number of samples from $\mathbf{p}$ it takes per query in the worst case.

One can define a more general notion, which allows the algorithm to only get "closer" to the desired property, and convert some type of access ORACLE_1 into some other type of access ORACLE_2 (e.g., from sampling to evaluation access):

Definition 5.2.2 (Sampling Improver (general definition))**.** Fix a given property $\mathcal{P}$ over distributions on Ω. A *sampling improver for* $\mathcal{P}$ (from ORACLE_1 to ORACLE_2) is a randomized algorithm which, given parameter $\varepsilon \in (0, 1]$ and ORACLE_1 access to a distribution $\mathbf{p}$ with the promise that $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathcal{P}) \leq \varepsilon$ as well as parameters $\varepsilon_1, \varepsilon_2 \in [0, 1]$ satisfying $\varepsilon_1 + \varepsilon_2 \geq \varepsilon$, provides, with probability at least $1 - \delta$ over the answers from ORACLE_1 and its internal randomness, ORACLE_2 access to a distribution $\tilde{\mathbf{p}}$ such that

$$\mathrm{d}_{\mathrm{TV}}(\tilde{\mathbf{p}}, \mathbf{p}) \leq \varepsilon_1 \qquad \text{(Close to } \mathbf{p}\text{)}$$

$$\mathrm{d}_{\mathrm{TV}}(\tilde{\mathbf{p}}, \mathcal{P}) \leq \varepsilon_2 \qquad \text{(Close to } \mathcal{P}\text{)}$$

In other terms, with high probability the algorithm will simulate exactly ORACLE_2 access to $\tilde{\mathbf{p}}$. The query complexity $q = q(\varepsilon, \varepsilon_1, \varepsilon_2, \delta, \Omega)$ of the algorithm is the number of queries it makes to ORACLE_1 in the worst case.

Finally, one may ask for such an improver to provide *many* samples from the (same) improved distribution,[6] where "many" is a number committed in advance. We refer to such an algorithm as a *batch sampling improver* (or, similarly, batch sampling corrector):

Definition 5.2.3 (Batch Sampling Improver)**.** For $\mathcal{P}$, $\mathbf{p}$, ε, $\varepsilon_1, \varepsilon_2 \in [0, 1]$ as above, and parameter $m \in \mathbb{N}$, a *batch sampling improver for* $\mathcal{P}$ (from ORACLE_1 to ORACLE_2) is a sampling improver which provides, with probability at least $1 - \delta$, ORACLE_2 access to $\tilde{\mathbf{p}}$ for as many as m queries, in between which it is allowed to maintain some internal state ensuring consistency. The query complexity of the algorithm is now allowed to depend on m as well.
Note that, in particular, when providing sampling access to $\tilde{\mathbf{p}}$ the batch improver must guarantee independence of the m samples. When ε_2 is set to 0 in the above definition, we will refer to the algorithm as a *batch sampling corrector*.

Remark 5.2.4 (On parameters of interest.). We observe that the regime of interest of our correctors and improvers is when the number of corrected samples to output is at least of the order $\Omega(1/\varepsilon)$. Indeed, if fewer samples are required, then the assumption that the distribution $\mathbf{p}$ be ε-close to having the property implies that – with high probability – a small number of samples from $\mathbf{p}$ will be indistinguishable from the closest distribution having the property. (So that, intuitively, they are already "as good as it gets," and need not be corrected.)

Remark 5.2.5 (On testing lower bounds). A similar observation holds for properties $\mathcal{P}$ that are known to be *hard to test*, that is for which some lower bound of $q(n, \varepsilon)$ samples holds to decide whether a given distribution satisfies $\mathcal{P}$, or is ε-far from it. In light of such a lower bound, one may wonder whether there is something to be gained in correcting $m < q(n, \varepsilon)$ samples, instead of simply using m samples from the original distribution altogether. However, such a result only states that there exists *some* worst-case instance $\mathbf{p}^*$ that is at distance ε from the property $\mathcal{P}$, yet requires this many samples to be distinguished from it: so that any algorithm relying on samples from distributions satisfying $\mathcal{P}$ could be fed $q(n, \varepsilon) - 1$ samples from this particular $\mathbf{p}^*$ without complaining. Yet, for "typical" distributions that are ε-close to $\mathcal{P}$, far fewer samples are required to reveal their deviation from it: for many, as few as $O(1/\varepsilon)$ suffice. Thus, an algorithm that expects to get say $q(n, \varepsilon)^{.99}$ samples from a honest-to-goodness distribution from $\mathcal{P}$, but instead is provided with samples from one that is merely ε-close to it, may break down very quickly. Our corrector, in this very regime of $o(q(n, \varepsilon))$ samples, guarantees this will not happen.

[6] Indeed, observe that as sampling correctors and improvers are randomized algorithms with access to their "own" coins, there is no guarantee that fixing the input distribution $\mathbf{p}$ would lead to the same output distribution $\tilde{\mathbf{p}}$. This is particularly important when providing other types of access (e.g., evaluation queries) to $\tilde{\mathbf{p}}$ than only sampling.

We conclude this section by introducing a relaxation of the notion of sampling corrector, where instead of asking the unknown distribution be close to the class it is corrected for we instead decouple the two. For instance, one may require the unknown distribution to be close to a Binomial distribution, but only correct it to be unimodal. This leads to the following definition of a *non-proper corrector*:

Definition 5.2.6 (Non-Proper Sampling Corrector)**.** Fix two given properties $\mathcal{P}$, $\mathcal{P}'$ of distributions on Ω. An $(\varepsilon, \varepsilon_1)$*-non-proper sampling corrector for* $\mathcal{P}'$ *assuming* $\mathcal{P}$ is a randomized algorithm which is given parameters $\varepsilon, \varepsilon_1 \in (0, 1]$ such that $\varepsilon_1 \geq \varepsilon$ and $\delta \in [0, 1]$, as well as sampling access to a distribution $\mathbf{p}$. Under the promise that $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathcal{P}) \leq \varepsilon$, the algorithm must provide, with probability at least $1 - \delta$ over the samples it draws and its internal randomness, sampling access to a distribution $\tilde{\mathbf{p}}$ such that

(i) $\tilde{\mathbf{p}}$ is close to $\mathbf{p}$: $\mathrm{d}_{\mathrm{TV}}(\tilde{\mathbf{p}}, \mathbf{p}) \leq \varepsilon_1$;

(ii) $\tilde{\mathbf{p}}$ has the (target) property: $\tilde{\mathbf{p}} \in \mathcal{P}'$.

In other terms, with high probability the algorithm will simulate exactly a sampling oracle for $\tilde{\mathbf{p}}$. The query complexity $q = q(\varepsilon, \varepsilon_1, \delta, \Omega)$ of the algorithm is the number of samples from $\mathbf{p}$ it takes per query in the worst case.

Note that if there exists $\mathbf{p}$ close to $\mathcal{P}$ such that every $\mathbf{p}' \in \mathcal{P}'$ is far from $\mathbf{p}$, this may not be achievable. Hence, the above definition requires that some relation between $\mathcal{P}$ and $\mathcal{P}'$ hold: for instance, that any neighborhood of a distribution from $\mathcal{P}$ intersects $\mathcal{P}'$. Similarly, we extend this definition to non-proper improvers and batch improvers.

5.3 A warmup: non-proper correcting of histograms

To illustrate these ideas, we start with a toy example: non-proper correcting of *regular histograms*. Recall that a distribution $\mathbf{p}$ over $[n]$ is said to be a k*-histogram* if its probability mass function is piecewise-constant with at most k "pieces:" that is, if there exists a partition $\mathcal{I} = (I_1, \ldots, I_k)$ of $[n]$ into k intervals such that $\mathbf{p}$ is constant on each I_j.

Letting $\mathcal{H}_k$ denote the class of all k-histograms over $[n]$, we start with the following question: given samples from a distribution close to $\mathcal{H}_k$, can we efficiently provide sample access to a corrected distribution $\tilde{\mathbf{p}} \in \mathcal{H}_\ell$, for some $\ell = \ell(k, \varepsilon)$? I.e., is there a non-proper corrector for $\mathcal{H}_\ell$ assuming $\mathcal{H}_k$?

In this short section, we show how to design such a corrector, under some additional assumption on the min-entropy of the unknown distribution to correct. Namely, we will require the following definition: given some constant $c \geq 1$, we say that a distribution $\mathbf{p}$ is *c-regular* if $\mathbf{p}(i) \leq \frac{c}{n}$ for all $i \in [n]$, i.e. $\|\mathbf{p}\|_\infty \leq \frac{c}{n}$.

Proposition 5.3.1 (Correcting regular histograms)**.** *Fix any constant $c > 0$. For any ε, $\varepsilon_1 \geq 4\varepsilon$ and $\varepsilon_2 = 0$ as in the definition, there exists $\ell = O(k/\varepsilon)$ and a non-proper sampling corrector for $\mathcal{H}_\ell$ assuming $\mathcal{H}_k$ with sample complexity $O(1)$, under the assumption that the unknown distribution is c-regular.*

Proof. The algorithm works as follows: setting $K \stackrel{\text{def}}{=} \frac{ck}{\varepsilon}$, it first divides the domain into $K \leq L \leq K + 1$

intervals $I_1, \ldots, I_L$ of size less than or equal to $\lfloor \frac{n}{K} \rfloor$. Then, the corrected distribution is the "flattening" $\bar{\mathbf{p}}$ of $\mathbf{p}$ on these intervals: to output a sample from the L-histogram $\bar{\mathbf{p}}$, the algorithm draws a sample $s \sim \mathbf{p}$, checks which of the I_i's this sample s belongs to, and then outputs s' drawn uniformly from this interval. The sample complexity is clearly constant, as outputting one sample of $\bar{\mathbf{p}}$ only requires one from $\mathbf{p}$; and being an L-histogram, $\bar{\mathbf{p}} \in \mathcal{H}_\ell$ for $\ell \leq \frac{ck}{\varepsilon} + 1$.

We now turn to proving that $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \bar{\mathbf{p}}) \leq 4\varepsilon$. Denote by H the closest k-histogram to $\mathbf{p}$, i.e. $H \in \mathcal{H}_k$ such that $\alpha \stackrel{\text{def}}{=} \mathrm{d}_{\mathrm{TV}}(\mathbf{p}, H) = \mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathcal{H}_k)$; and let B be the union of the (at most k) intervals among $I_1, \ldots, I_L$ where H is not constant. Since $\mathbf{p}$ is c-regular, we do have $\mathbf{p}(B) \leq k \cdot \frac{c}{n} \cdot \frac{n}{K} = \varepsilon$. Then, since H is α-close to $\mathbf{p}$ we get $H(B) \leq \varepsilon + \alpha$.

Now, let $\bar{\mathbf{p}}$ (resp. $\bar{H}$) be the L-histogram obtained by "flattening" $\mathbf{p}$ (resp. H) on $I_1, \ldots, I_L$. By the data processing inequality (Fact 1.4.2), we obtain

$$\mathrm{d}_{\mathrm{TV}}\big(\bar{\mathbf{p}}, \bar{H}\big) \leq \mathrm{d}_{\mathrm{TV}}(\mathbf{p}, H).$$

Therefore, by the triangle inequality,

$$\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \bar{\mathbf{p}}) \leq \mathrm{d}_{\mathrm{TV}}(\mathbf{p}, H) + \mathrm{d}_{\mathrm{TV}}\big(H, \bar{H}\big) + \mathrm{d}_{\mathrm{TV}}\big(\bar{H}, \bar{\mathbf{p}}\big) \leq 2\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, H) + \mathrm{d}_{\mathrm{TV}}\big(H, \bar{H}\big).$$

Furthermore, as H and $\bar{H}$ can only differ on B, and since the flattening operation preserve the probability weight on each interval of $\mathcal{I}$, we obtain

$$\mathrm{d}_{\mathrm{TV}}\big(H, \bar{H}\big) = \frac{1}{2}\|H - \bar{H}\|_1 = \frac{1}{2}\sum_{i \in B} \big|H(i) - \bar{H}(i)\big| \leq \frac{1}{2}\left(H(B) + \bar{H}(B)\right) = H(B) \leq \varepsilon + \alpha$$

which, once plugged back in the previous expression, yields

$$\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \bar{\mathbf{p}}) \leq 2\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, H) + \varepsilon + \alpha = 3\alpha + \varepsilon \leq 4\varepsilon$$

since $\alpha \leq \varepsilon$ by assumption. □

5.4 Connections to learning and testing

In this section, we draw connections between sampling improvers and other areas, namely testing and learning. These connections shed light on the relation between our model and these other lines of work, and provide a way to derive new algorithms and impossibility results for both testing or learning problems. (For the formal definition of the testing and learning notions used in this section, the reader is referred to Section 1.2 and the relevant subsections.)

5.4.1 From learning to correcting

As a first observation, it is not difficult to see that, under the assumption that the unknown distribution $\mathbf{p}$ belongs to some specific class $\mathcal{C}$, correcting (or improving) a property $\mathcal{P}$ requires at most as many samples as learning the class $\mathcal{C}$; that is, *learning (a class of distributions) is at least as hard as correcting (distributions of this class).* Here, $\mathcal{P}$ and $\mathcal{C}$ need not be related.

Indeed, assuming there exists a learning algorithm $\mathcal{L}$ for $\mathcal{C}$, it then suffices to run $\mathcal{L}$ on the unknown distribution $\mathbf{p} \in \mathcal{C}$ to learn (with high probability) a hypothesis $\hat{\mathbf{p}}$ such that $\mathbf{p}$ and $\hat{\mathbf{p}}$ are at most at distance $\frac{\varepsilon_1-\varepsilon}{2}$. In particular, $\hat{\mathbf{p}}$ is at most $\frac{\varepsilon_1+\varepsilon}{2}$-far from $\mathcal{P}$. One can then (e.g., by exhaustive search) find a distribution $\tilde{\mathbf{p}}$ in $\mathcal{P}$ which is closest to $\hat{\mathbf{p}}$ (and therefore at most ε_1-far from $\mathbf{p}$), and use it to produce as many "corrected samples" as wanted:

Theorem 5.4.1. *Let $\mathcal{C}$ a class of probability distributions over Ω. Suppose there exists a learning algorithm $\mathcal{L}$ for $\mathcal{C}$ with sample complexity $q_{\mathcal{L}}$. Then, for any property $\mathcal{P}$ of distributions, there exists a (not-necessarily computationally efficient) sampling corrector for $\mathcal{P}$ with sample complexity $q(\varepsilon, \varepsilon_1, \delta) = q_{\mathcal{L}}\left(\frac{\varepsilon_1-\varepsilon}{2}, \delta\right)$, under the promise that $\mathbf{p} \in \mathcal{C}$.*

Furthermore, if the (efficient) learning algorithm $\mathcal{L}$ has the additional guarantee that its hypothesis class is a subset of $\mathcal{P}$ (i.e., the hypotheses it produces always belong to $\mathcal{P}$) and that the hypotheses it contains allow efficient generation of samples, then we immediately obtain a computationally efficient sampling corrector: indeed, in this case $\hat{\mathbf{p}} \in \mathcal{P}$ already. Furthermore, as mentioned in the introduction, when efficient agnostic proper learning algorithms for distribution classes exist, then there are efficient sampling correctors for the same classes. It is however worth pointing out that this correcting-by-learning approach is quite inefficient with regard to the amount of extra randomness needed: indeed, every sample generated from $\tilde{\mathbf{p}}$ requires fresh new random bits.

To illustrate this theorem, we give two easy corollaries. The first follows from Chan et al., who showed in [56] that monotone hazard risk distributions can be learned to accuracy ε using $\tilde{O}\big(\log n/\varepsilon^4\big)$ samples; moreover, the hypothesis obtained is a $O\big(\log(n/\varepsilon)/\varepsilon^2\big)$-histogram.

Corollary 5.4.2. *Let $\mathcal{C}$ be the class of monotone hazard risk distributions over $[n]$, and $\mathcal{P}$ be the property of being a histogram with (at most) $\sqrt{n}$ pieces. Then, under the promise that $\mathbf{p} \in \mathcal{C}$ and as long as $\varepsilon = \tilde{\Omega}(1/\sqrt{n})$, there is a sampling corrector for $\mathcal{P}$ with sample complexity $\tilde{O}\Big(\frac{\log n}{(\varepsilon_1-\varepsilon)^4}\Big)$.*

Our next example however demonstrates that this learning approach is not always optimal:

Corollary 5.4.3. *Let $\mathcal{C}$ be the class of monotone distributions over $[n]$, and $\mathcal{P}$ be the property of being a histogram with (at most) $\sqrt{n}$ pieces. Then, under the promise that $\mathbf{p} \in \mathcal{C}$ and as long as $\varepsilon = \tilde{\Omega}(1/\sqrt{n})$, there is a sampling corrector for $\mathcal{P}$ with sample complexity $O\Big(\frac{\log n}{(\varepsilon_1-\varepsilon)^3}\Big)$.*

Indeed, for learning monotone distributions $\Theta\big(\log n/\varepsilon^3\big)$ samples are known to be necessary and sufficient [32]. Yet, one can also correct the distribution by simulating samples directly from its flattening on the corresponding

Birgé decomposition (as per Definition 1.4.4); and every sample from this correction-by-simulation costs exactly *one* sample from the original distribution.

5.4.2 From correcting to agnostic learning

Let $\mathcal{C}$ and $\mathcal{H}$ be two classes of probability distributions over Ω. Recall that a *(semi-)agnostic learner for* $\mathcal{C}$ (using hypothesis class $\mathcal{H}$) is a learning algorithm $\mathcal{A}$ which, given sample access to an arbitrary distribution $\mathbf{p}$ and parameter ε, outputs a hypothesis $\hat{\mathbf{p}} \in \mathcal{H}$ such that, with high probability, $\hat{\mathbf{p}}$ does "as well as the best approximation from $\mathcal{C}$:"

$$\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \hat{\mathbf{p}}) \leq c \cdot \mathrm{OPT}_{\mathcal{C},\mathbf{p}} + O(\varepsilon)$$

where $\mathrm{OPT}_{\mathcal{C},\mathbf{p}} \stackrel{\mathrm{def}}{=} \inf_{\mathbf{p}_{\mathcal{C}} \in \mathcal{C}} \mathrm{d}_{\mathrm{TV}}(\mathbf{p}_{\mathcal{C}}, \mathbf{p})$ and $c \geq 1$ is some absolute constant (if $c = 1$, the learner is said to be agnostic).

We first describe how to combine a (non-agnostic) learning algorithm with a sampling corrector in order to obtain an agnostic learner, under the strong assumption that a (rough) estimate of OPT is known. Then, we explain how to get rid of this extra requirement, using machinery from the distribution learning literature (namely, an efficient *hypothesis selection* procedure).

Theorem 5.4.4. *Let $\mathcal{C}$ be as above. Suppose there exists a learning algorithm $\mathcal{L}$ for $\mathcal{C}$ with sample complexity $q_{\mathcal{L}}$, and a batch sampling corrector $\mathcal{A}$ for $\mathcal{C}$ with sample complexity $q_{\mathcal{A}}$. Suppose further that a constant-factor estimate $\widehat{\mathrm{OPT}}$ of $\mathrm{OPT}_{\mathcal{C},\mathbf{p}}$ is known (up to a multiplicative c).*

Then, there exists a semi-agnostic learner for $\mathcal{C}$ with sample complexity $q(\varepsilon, \delta) = q_{\mathcal{A}}(\widehat{\mathrm{OPT}}, \widehat{\mathrm{OPT}} + \varepsilon, q_{\mathcal{L}}(\varepsilon, \frac{\delta}{2}), \frac{\delta}{2})$ (where the constant in front of $\mathrm{OPT}_{\mathcal{C},\mathbf{p}}$ is c).

Proof. Let c be the constant such $\mathrm{OPT}_{\mathcal{C},\mathbf{p}} \leq \widehat{\mathrm{OPT}} \leq c \cdot \mathrm{OPT}_{\mathcal{C},\mathbf{p}}$. The agnostic learner $\mathcal{L}'$ for $\mathcal{P}$, on input $\varepsilon \in (0, 1]$, works as follows:

- Run $\mathcal{A}$ on $\mathbf{p}$ with parameters $(\widehat{\mathrm{OPT}}, \widehat{\mathrm{OPT}} + \varepsilon, \frac{\delta}{2})$ to get $q_{\mathcal{L}}(\varepsilon, \frac{\delta}{2})$ samples distributed according to some distribution $\tilde{\mathbf{p}}$.
- Run $\mathcal{L}$ on these samples, with parameters $\varepsilon, \frac{\delta}{2}$, and output its hypothesis $\hat{\mathbf{p}}$.

We hereafter condition on both algorithms succeeding (which, by a union bound, happens with probability at least $1 - \delta$). Since $\mathbf{p}$ is $\widehat{\mathrm{OPT}}$-close to $\mathcal{C}$, and therefore by correctness of the sampling corrector we have both $\tilde{\mathbf{p}} \in \mathcal{C}$ and $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \tilde{\mathbf{p}}) \leq \widehat{\mathrm{OPT}} + \varepsilon$. Hence, the output $\hat{\mathbf{p}}$ of the learning algorithm satisfies $\mathrm{d}_{\mathrm{TV}}(\tilde{\mathbf{p}}, \hat{\mathbf{p}}) \leq \varepsilon$, which implies

$$\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \hat{\mathbf{p}}) \leq \widehat{\mathrm{OPT}} + 2\varepsilon \leq c \cdot \mathrm{OPT}_{\mathcal{C},\mathbf{p}} + 2\varepsilon \tag{5.1}$$

for some absolute constant c, as claimed (using the assumption on $\widehat{\mathrm{OPT}}$). □

It is worth noting that in the case the learning algorithm is *proper* (meaning the hypotheses it outputs belong to the target class $\mathcal{C}$: that is, $\mathcal{H} \subseteq \mathcal{C}$), then so is the agnostic learner obtained with Theorem 5.4.4.

This turns out to be a very strong guarantee: specifically, getting (computationally efficient) proper agnostic learning algorithms remains a challenge for many classes of interest – see e.g. [64], which mentions efficient proper learning of Poisson Binomial Distributions as an open problem.

We stress that the above can be viewed as a *generic* framework to obtain efficient agnostic learning results from known efficient learning algorithms. For the sake of illustration, let us consider the simple case of Binomial distributions: it is known, for instance as a consequence of the aforementioned results on PBDs, that learning such distributions can be performed with $\tilde{O}\big(1/\varepsilon^2\big)$ samples (and that $\Omega\big(1/\varepsilon^2\big)$ are required). Our theorem then provides a simple way to obtain agnostic learning of Binomial distributions with sample complexity $\tilde{O}\big(1/\varepsilon^2\big)$: namely, by designing an efficient sampling corrector for this class with sample complexity $\mathrm{poly}(\log \frac{1}{\varepsilon}, \log \frac{1}{\varepsilon_1})$.

Corollary 5.4.5. *Suppose there exists a batch sampling corrector $\mathcal{A}$ for the class $\mathcal{B}$ of Binomial distributions over $[n]$, with sample complexity $q_{\mathcal{A}}(\varepsilon, \varepsilon_1, m, \delta) = \mathrm{polylog}(\frac{1}{\varepsilon}, \frac{1}{\varepsilon_1}, m, \frac{1}{\delta})$. Then, there exists a semi-agnostic learner for $\mathcal{B}$, which, given access to an unknown distribution $\mathbf{p}$ promised to be ε-close to some Binomial distribution, takes $\tilde{O}\big(\frac{1}{\varepsilon^2}\big)$ samples from $\mathbf{p}$ and outputs a distribution $\hat{B} \in \mathcal{B}$ such that*

$$\mathrm{d}_{\mathrm{TV}}\Big(\mathbf{p}, \hat{B}\Big) \leq 3\varepsilon$$

with probability at least $2/3$.

To the best of our knowledge, an agnostic learning algorithm for the class of Binomial distributions with sample complexity $\tilde{O}\big(1/\varepsilon^2\big)$ is not explicitly known, although the results of [55] do imply a $\tilde{O}\big(1/\varepsilon^3\big)$ upper bound and a modification of [64] (to make their algorithm agnostic) seems to yield one. The above suggests an approach which would lead to the (essentially optimal) sample complexity. (Since publication of our work, we have learned that [4] provides such a result unconditionally.)

5.4.2.1 Removing the assumption on knowing $\widehat{\mathrm{OPT}}$

In the absence of such an estimate $\widehat{\mathrm{OPT}}$ within a constant factor of $\mathrm{OPT}_{\mathcal{C},\mathbf{p}}$ given as input, one can apply the following strategy, inspired of [57, Theorem 6]. In the first stage, we try to repeatly "guess" a good $\widehat{\mathrm{OPT}}$, and run the agnostic learner of Theorem 5.4.4 with this value to obtain a hypothesis. After this stage, we have generated a succinct list $\mathcal{H}$ of hypotheses, one for each $\widehat{\mathrm{OPT}}$ that we tried: the second stage is then to run a hypothesis selection procedure to pick the best $h \in \mathcal{H}$: as long as one of the guesses was good, this h will be an accurate hypothesis.

More precisely, suppose we run the agnostic learner of Theorem 5.4.4 a total of $\log(1/\varepsilon)$ times, setting at the k^{th} iteration $\widehat{\mathrm{OPT}}_k \stackrel{\text{def}}{=} 2^k\varepsilon$ and $\delta' \stackrel{\text{def}}{=} \delta/(2\log(1/\varepsilon))$. For the first k such that $2^{k-1}\varepsilon \leq \mathrm{OPT}_{\mathcal{C},\mathbf{p}} < 2^k\varepsilon$, $\widehat{\mathrm{OPT}}_k$ is in $[\mathrm{OPT}_{\mathcal{C},\mathbf{p}}, 2 \cdot \mathrm{OPT}_{\mathcal{C},\mathbf{p}}]$. Therefore, by a union bound on all runs of the learner at least one of the hypotheses $\hat{\mathbf{p}}_k$ will have the agnostic learning guarantee we want to achieve; i.e. will satisfy (5.1), with $c = 2$.

Conditioned on this being the case, it remains to determine *which* hypothesis achieves the guarantee of

being $(2\text{OPT} + O(\varepsilon))$-close to the distribution $\mathbf{p}$. This is where we apply a hypothesis selection algorithm – a variant of the similar "tournament" procedures from [77, 66, 5] – to our $N = \log(1/\varepsilon)$ candidates, with accuracy parameter ε and failure probability $\delta/2$. This algorithm has the following guarantee:

Proposition 5.4.6 ([119])**.** *There exists a procedure* TOURNAMENT *that, given sample access to an unknown distribution* $\mathbf{p}$ *and both sample and evaluation access to* N *hypotheses* $H_1, \dots, H_N$*, has the following behavior.* TOURNAMENT *makes a total of* $\tilde{O}\big(\log(N/\delta)/\varepsilon^2\big)$ *queries to* $\mathbf{p}, H_1, \dots, H_N$*, runs in time* $O\big(N \log(N/\delta)/\varepsilon^2\big)$*, and outputs a hypothesis* H_i *such that, with probability at least* $1 - \delta$*,*

$$\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, H_i) \leq 9.1 \min_{j \in [N]} \mathrm{d}_{\mathrm{TV}}(\mathbf{p}, H_j) + O(\varepsilon).$$

Summary Using this result in the approach outlined above, we get with probability at least $1 - \delta$, we will obtain a hypothesis $\hat{\mathbf{p}}_{k^*}$ doing "almost as well as the best $\mathbf{p}_k$"; that is,

$$\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \hat{\mathbf{p}}_{k^*}) \leq 18.2 \cdot \text{OPT}_{\mathcal{C},\mathbf{p}} + O(\varepsilon)$$

The overall sample complexity is

$$\sum_{k=1}^{\log(1/\varepsilon)} q_{\mathcal{A}}\left(2^k\varepsilon, (2^k+1)\varepsilon, q_{\mathcal{L}}\left(\varepsilon, \frac{\delta}{4\log(1/\varepsilon)}\right), \frac{\delta}{4\log(1/\varepsilon)}\right) + \tilde{O}\left(\frac{1}{\varepsilon^2}\log\frac{1}{\delta}\right)$$

where the first term comes from the $\log(1/\varepsilon)$ runs of the learner from Theorem 5.4.4, and the second is the overhead due to the hypothesis selection tournament.

5.4.3 From correcting to tolerant testing

We observe that the existence of sampling correctors for a given property $\mathcal{P}$, along with an efficient distance estimation procedure, allows one to convert any distribution testing algorithm into a tolerant distribution testing algorithm. This is similar to the connection between "local reconstructors" and tolerant testing of graphs described in [39, Theorem 3.1] and [41, Theorem 3.1]. That is, if a property $\mathcal{P}$ has both a distance estimator and a sampling corrector, then one can perform *tolerant* testing of $\mathcal{P}$ in the time required to generate enough corrected samples for both the estimator and a (non-tolerant) tester.

We first state our theorem in all generality, before instantiating it in several corollaries. For the sake of clarity, the reader may wish to focus on these on a first pass.

Theorem 5.4.7. *Let* $\mathcal{C}$ *be a class of distributions, and* $\mathcal{P} \subseteq \mathcal{C}$ *a property. Suppose there exists an* $(\varepsilon, \varepsilon_1)$*-batch sampling corrector* $\mathcal{A}$ *for* $\mathcal{P}$ *with complexity* $q_{\mathcal{A}}$*, and a distance estimator* $\mathcal{E}$ *for* $\mathcal{C}$ *with complexity* $q_{\mathcal{E}}$ *– that is, given sample access to* $\mathbf{p}_1, \mathbf{p}_2 \in \mathcal{C}$ *and parameters* ε*,* δ*,* $\mathcal{E}$ *draws* $q_{\mathcal{E}}(\varepsilon, \delta)$ *samples from* $\mathbf{p}_1, \mathbf{p}_2$ *and outputs a value* $\hat{d}$ *such that* $\left|\hat{d} - \mathrm{d}_{\mathrm{TV}}(\mathbf{p}_1, \mathbf{p}_2)\right| \leq \varepsilon$ *with probability at least* $1 - \delta$*.*

Then, from any property tester $\mathcal{T}$ *for* $\mathcal{P}$ *with sample complexity* $q_{\mathcal{T}}$*, one can get a tolerant tester* $\mathcal{T}'$ *with*

query complexity $q(\varepsilon', \varepsilon, \delta) = q_{\mathcal{A}}\Big(\varepsilon', \Theta(\varepsilon), q_{\mathcal{E}}(\frac{\varepsilon-\varepsilon'}{4}, \frac{\delta}{3}) + q_{\mathcal{T}}(\frac{\varepsilon-\varepsilon'}{4}, \frac{\delta}{3}), \frac{\delta}{3}\Big)$.

Proof. The tolerant tester $\mathcal{T}'$ for $\mathcal{P}$, on input $0 \leq \varepsilon' < \varepsilon \leq 1$, works as follows, setting $\beta \stackrel{\text{def}}{=} \frac{\varepsilon-\varepsilon'}{4}$ and $\varepsilon_1 \stackrel{\text{def}}{=} \varepsilon' + \beta$:

- Run $\mathcal{A}$ on $\mathbf{p}$ with parameters $(\varepsilon', \varepsilon_1, \delta/3)$ to get $q_{\mathcal{E}}(\beta, \delta/3) + q_{\mathcal{T}}(\beta, \delta/3)$ samples distributed according to some distribution $\tilde{\mathbf{p}}$. Using these samples:
  1. Estimate $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \tilde{\mathbf{p}})$ to within an additive β, and reject if this estimate is more than $\varepsilon_1 + \beta = \frac{\varepsilon+\varepsilon'}{2}$;
  2. Otherwise, run $\mathcal{T}$ on $\tilde{\mathbf{p}}$ with parameter β and accept if and only if $\mathcal{T}$ outputs accept.

We hereafter condition on all 3 algorithms succeeding (which, by a union bound, happens with probability at least $1 - \delta$).

If $\mathbf{p}$ is ε'-close to $\mathcal{P}$, then the corrector ensures that $\tilde{\mathbf{p}}$ is ε_1-close to $\mathbf{p}$, so the estimate of $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \tilde{\mathbf{p}})$ is at most $\varepsilon_1 + \beta$: Step 1 thus passes, and as $\tilde{\mathbf{p}} \in \mathcal{P}$ the tester outputs accept in Step 2.

On the other hand, if $\mathbf{p}$ is ε-far from $\mathcal{P}$, then either (a) $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \tilde{\mathbf{p}}) > \varepsilon_1 + 2\beta$ (in which case we output reject in Step 1, since the estimate exceeds $\varepsilon_1 + \beta$), or (b) $\mathrm{d}_{\mathrm{TV}}(\tilde{\mathbf{p}}, \mathcal{P}) > \varepsilon - (\varepsilon_1 + 2\beta) = \beta$, in which case $\mathcal{T}$ outputs reject in Step 2. □

Remark 5.4.8. Only asking that the distance estimation procedure $\mathcal{E}$ be specific to the class $\mathcal{C}$ is not innocent; indeed, it is known ([171]) that for *general* distributions, distance estimation has sample complexity $n^{1-o(1)}$. However, the task becomes significantly easier for certain classes of distributions: and for instance can be performed with only $\tilde{O}(k \log n)$ samples, if the distributions are guaranteed to be k-modal [74]. This observation can be leveraged in cases when one knows that the distribution has a specific property, but does not quite satisfy a second property: e.g. is known to be k-modal but not known to be, say, log-concave.

The reduction above can be useful both as a black-box way to derive upper bounds for tolerant testing, as well as to prove lower bounds for either testing or distance estimation. For the first use, we give two applications of our theorem to provide tolerant monotonicity testers for k-modal distributions. The first is a conditional result, showing that the existence of good monotonicity correctors yield tolerant testers. The second, while unconditional, only guarantees a weaker form of tolerance (guaranteeing acceptance only of distributions that are very close to monotone); and relies on a corrector we describe in Section 5.5.2. As we detail shortly after stating these two results, even this weak tolerance improves upon the one provided by currently known testing algorithms.

Corollary 5.4.9. *Suppose there exists an $(\varepsilon, \varepsilon_1)$-batch sampling corrector for monotonicity with complexity q. Then, for any $k = O(\log n / \log\log n)$, there exists an algorithm that distinguishes whether a k-modal distribution is (a) ε-close to monotone or (b) 5ε-far from monotone with success probability $2/3$, and sample complexity*

$$q\left(\varepsilon, 2\varepsilon, C\frac{k \log n}{\varepsilon^4 \log\log n}, \frac{1}{9}\right)$$

where C is an absolute constant.

Proof. We combine the distance estimator of [74] with the monotonicity tester of [63, Section 3.4], which both apply to the class of k-modal distributions. As their respective sample complexity is, for distance parameter α and failure probability δ, $O\left(\left(\frac{k^2}{\alpha^4} + \frac{k \log n}{\alpha^4 \log(k \log n)}\right) \log \frac{1}{\delta}\right)$ and $O\left(\frac{k}{\alpha^2} \log \frac{1}{\delta}\right)$, the choice of parameters ($\delta = 1/3$, ε and 5ε) and the assumption on k yield

$$O\left(\frac{k}{\varepsilon^2}\right) + O\left(\frac{k^2}{\varepsilon^4} + \frac{k \log n}{\varepsilon^4 \log(k \log n)}\right) = O\left(\frac{k \log n}{\varepsilon^4 \log(k \log n)}\right)$$

and we obtain by Theorem 5.4.7 a tolerant tester with sample complexity $q\left(\varepsilon, 2\varepsilon, O\left(\frac{k \log n}{\varepsilon^4 \log(k \log n)}\right), \frac{1}{9}\right)$, as claimed. □

Another application of this theorem, but this time taking advantage of a result from Section 5.5.1, allows us to derive an *explicit* tolerant tester for monotonicity of k-modal distributions:

Corollary 5.4.10. *For any $k \geq 1$, there exists an algorithm that distinguishes whether a k-modal distribution is (a) $O\left(\varepsilon^3 / \log^2 n\right)$-close to monotone or (b) ε-far from monotone with success probability $2/3$, and sample complexity*

$$O\left(\frac{1}{\varepsilon^4} \frac{k \log n}{\log(k \log n)} + \frac{k^2}{\varepsilon^4}\right).$$

In particular, for $k = O(\log n / \log \log n)$ this yields a (weakly) tolerant tester with sample complexity $O\left(\frac{1}{\varepsilon^4} \frac{k \log n}{\log \log n}\right)$.

Proof. We again use the distance estimator of [74] and the monotonicity tester of [63], which both apply to the class of k-modal distributions, this time with the monotonicity corrector we describe in Corollary 5.5.5, which works for any ε_1 and $\varepsilon = O\left(\varepsilon_1^3 / \log^2 n\right)$ and has constant-rate sample complexity (that is, it takes $O(q)$ samples from the original distribution to output q samples). Similarly to Corollary 5.4.9, the sample complexity is a straightforward application of Theorem 5.4.7. □

Note that, to the best of our knowledge, no tolerant tester for monotonicity of k-modal distributions was previously known, though using the (regular) $O\left(k/\varepsilon^2\right)$-sample tester of [63] and standard arguments, one can achieve a weak tolerance on the order of $O\left(\varepsilon^2/k\right)$. While the sample complexity obtained in Corollary 5.4.10 is worse by a $\mathrm{polylog}(n)$ factor, it has better tolerance for $k = \Omega(\log^2 n/\varepsilon)$.

5.5 Sample complexity of correcting monotonicity

In this section, we focus on the sample complexity aspect of correcting, considering the specific example of monotonicity correction. As a first result, we show in Section 5.5.1 how to design a simple batch corrector for monotonicity which, after a preprocessing step costing logarithmically many samples, is able to answer an arbitrary number of queries. This corrector follows the "learning approach" described in Section 5.4.1, and in particular provides a very efficient way to amortize the cost of making many queries to a corrected distribution.

A natural question is then whether one can "beat" this approach, and correct the distribution without

approximating it as a whole beforehand. Section 5.5.2 answers it by the affirmative: namely, we show that one can correct distributions that are guaranteed to be $(1/\log^2 n)$-close to monotone in a completely *oblivious* fashion, with a non-adaptive approach that does not require to learn anything about the distribution.

Finally, we give in Section 5.5.3 a corrector for monotonicity with no restriction on the range of parameters, but assuming a stronger type of query access to the original distribution. Specifically, our algorithm leverages the ability to make *cdf queries* to the distribution $\mathbf{p}$, in order to generate independent samples from a corrected $\tilde{\mathbf{p}}$. This sampling corrector also outperforms the one from Section 5.5.1, making only $O\big(\sqrt{\log n}\big)$ queries per sample on expectation.

A parenthesis: non-proper correcting We note that it is easy to obtain a non-proper corrector for $k(n,\varepsilon)$-histograms assuming monotonicity with *constant* sample complexity, for $k(n,\varepsilon) = \Theta\Big(\frac{\log n}{\varepsilon}\Big)$. Indeed, this follows from the oblivious Birgé decomposition (see Definition 1.4.4) we shall be using many times through this section, which ensures that "flattening" a monotone distribution yields a $k(n,\varepsilon)$-histogram that remains close to the original distribution.

5.5.1 A natural approach: correcting by learning

Our first corrector works in a straightforward fashion: it *learns* a good approximation of the distribution to correct, which is also concisely represented. It then uses this approximation to build a sufficiently good monotone distribution M' "offline," by searching for the closest monotone distribution, which in this case can be achieved via linear programming. Any query made to the corrector is then answered according to the latter distribution, at no additional cost.

Lemma 5.5.1 (Correcting by learning)**.** *Fix any constant $c > 0$. For any ε, $\varepsilon_1 \geq (3+c)\varepsilon$ and $\varepsilon_2 = 0$ as in the definition, any type of oracle* ORACLE *and any number of queries m, there exists a sampling corrector for monotonicity from sampling to* ORACLE *with sample complexity $O\big(\log n/\varepsilon^3\big)$.*

Proof. Consider the Birgé decomposition $\mathcal{I}_\alpha = (I_1, \dots, I_\ell)$ with parameter $\alpha \stackrel{\text{def}}{=} \frac{c\varepsilon}{3}$ which partitions the domain $[n]$ into $O\Big(\frac{\log n}{\varepsilon}\Big)$ intervals. By Corollary 1.4.6 and the learning result of [32], we can learn with $O\Big(\frac{\log n}{\varepsilon^3}\Big)$ samples a $O\Big(\frac{\log n}{\varepsilon}\Big)$-histogram $\bar{\mathbf{p}}$ such that:

$$\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \bar{\mathbf{p}}) \leq 2\varepsilon + \alpha. \tag{5.2}$$

Also, let M be the closest monotone distribution to $\mathbf{p}$. From Eq. (1.4), we get the following: letting $\mathcal{M}$ denote the set of monotone distributions,

$$\mathrm{d}_{\mathrm{TV}}(\bar{\mathbf{p}}, \mathcal{M}) = \mathrm{d}_{\mathrm{TV}}(\Phi_\alpha(\mathbf{p}), \mathcal{M}) \leq \mathrm{d}_{\mathrm{TV}}(\Phi_\alpha(\mathbf{p}), \Phi_\alpha(M)) \leq \mathrm{d}_{\mathrm{TV}}(\mathbf{p}, M) \leq \varepsilon \tag{5.3}$$

where the first inequality follows from the fact that $\Phi_\varepsilon(M)$ is monotone. Thus, $\bar{\mathbf{p}}$ is ε-close to monotone,

which implies that $\bar{\mathbf{p}}'$ is $(\varepsilon+\alpha)$-close to monotone. Furthermore, it is easy to see that, without loss of generality, one can assume the closest monotone distribution $\bar{\mathbf{p}}'$ to be piecewise constant with relation to the same partition (e.g., using again Eq. (1.4)). It is therefore sufficient to find such a piecewise constant distribution: to do so, consider the following linear program which finds exactly this: a monotone M', closest to $\bar{\mathbf{p}}'$ and piecewise constant on $\mathcal{I}_\alpha$:

$$\begin{aligned}
\text{minimize } & \sum_{j=1}^{\ell} \left| x_j - \frac{\bar{\mathbf{p}}'(I_j)}{|I_j|} \right| \cdot |I_j| \\
\text{subject to } & 1 \geq x_1 \geq x_2 \geq \cdots \geq x_l \geq 0 \\
& \sum_{j=1}^{\ell} x_j \, |I_j| = 1
\end{aligned}$$

This linear program has $O\left(\frac{\log n}{\varepsilon}\right)$ variables and so it can be solved in time $\mathrm{poly}(\log n, \frac{1}{\varepsilon})$.

After finding a solution $(x_j)_{j\in[\ell]}$ to this linear program,[7] we define the distribution $M' : [n] \to [0,1]$ as follows: $M'(i) = x_{\mathrm{ind}(i)}$, where $\mathrm{ind}(i)$ is the index of the interval of $\mathcal{I}_\alpha$ which i belongs to. This implies that

$$\mathrm{d}_{\mathrm{TV}}(\bar{\mathbf{p}}', M') \leq \varepsilon + \alpha$$

and by the triangle inequality we finally get:

$$\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, M^*) \leq \mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \bar{\mathbf{p}}) + \mathrm{d}_{\mathrm{TV}}(\bar{\mathbf{p}}, \bar{\mathbf{p}}') + \mathrm{d}_{\mathrm{TV}}(\bar{\mathbf{p}}', M') \leq 3\varepsilon + 3\alpha = (3+c)\varepsilon.$$

□

5.5.2 Oblivious correcting of distributions which are very close to monotone

We now turn to our second monotonicity corrector, which achieves constant sample complexity for distributions already $(1/\log^2 n)$-close to monotone. Note that this is a very strong assumption, as if one draws less than $\log^2 n$ samples one does not expect to see any difference between such a distribution $\mathbf{p}$ and its closest monotone distribution. Still, our construction actually yields a stronger guarantee: namely, given *evaluation (query)* access to $\mathbf{p}$, it can answer evaluation queries to the corrected distribution as well. See Remark 5.5.6 for a more detailed statement.

The high-level idea is to treat the distribution as a k-histogram on the Birgé decomposition (for $k = O(\log n)$), thus "implicitly approximating" it; and to correct this histogram by adding a certain amount of probability weight to every interval, so that each gets slightly more than the next one. By choosing these quantities carefully, this ensures that *any* violation of monotonicity gets corrected in the process, without ever having to find out *where* they actually occur.

[7] To see why a good solution always exists, consider the closest monotone distribution to $\bar{\mathbf{p}}$, and apply Φ_α to it. This distribution satisfies all the constraints.

We start by stating the general correcting approach for general k-histograms satisfying a certain property (namely, the ratio between two consecutive intervals is constant).

Lemma 5.5.2. *Let $\mathcal{I} = (I_1, \ldots, I_k)$ be a decomposition of $[n]$ in consecutive intervals such that $|I_{j+1}| / |I_j| = 1 + c$ for all j, and $\mathbf{p}$ be a k-histogram distribution on $\mathcal{I}$ that is ε-close to monotone. Then, there is a monotone distribution $\tilde{\mathbf{p}}$ which can be sampled from in constant time given oracle access to $\mathbf{p}$, such that $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \tilde{\mathbf{p}}) = O(\varepsilon k^2)$. Further, $\tilde{\mathbf{p}}$ is also a k-histogram distribution on $\mathcal{I}$.*

Proof. We will argue that no interval can have significantly more total weight than the previous one, as it would otherwise contradict the bound on the closeness to monotonicity. This bound on the "jump" between two consecutive intervals enables us to define a new distribution $\hat{\mathbf{p}}$ which is a mixture of $\mathbf{p}$ with an arithmetically decreasing k-histogram (which only depends on ε and k); it can be shown that for the proper choice of parameters, $\hat{\mathbf{p}}$ is now monotone.

We start with the following claim, which leverages the distance to monotonicity in order to give a bound on the total violation between two consecutive intervals of the partition:

Claim 5.5.3. *Let $\mathbf{p}$ be a k-histogram distribution on $\mathcal{I}$ that is ε-close to monotone. Then, for any $j \in \{1, \ldots, k-1\}$,*

$$\mathbf{p}(I_{j+1}) \leq (1+c)\mathbf{p}(I_j) + \varepsilon(2+c). \tag{5.4}$$

Proof. First, observe that without loss of generality, one can assume the monotone distribution closest to $\mathbf{p}$ to be a k-histogram on $\mathcal{I}$ as well (e.g., by a direct application of Fact 1.4.2 to the flattening on $\mathcal{I}$ of the monotone distribution closest to $\mathbf{p}$). Assume there exists an index $j \in \{1, \ldots, k-1\}$ contradicting (5.4); then,

$$\frac{\mathbf{p}(I_{j+1})}{|I_{j+1}|} > (1+c)\frac{\mathbf{p}(I_j)}{|I_j|} \cdot \frac{|I_j|}{|I_{j+1}|} + \varepsilon\frac{2+c}{|I_{j+1}|} = \frac{\mathbf{p}(I_j)}{|I_j|} + \varepsilon\frac{2+c}{|I_{j+1}|}.$$

But any monotone distribution M which is a k-histogram on $\mathcal{I}$ must satisfy $\frac{M(I_{j+1})}{|I_{j+1}|} \leq \frac{M(I_j)}{|I_j|}$; so that at least $\varepsilon(2+c)$ total weight has to be "redistributed" to fix this violation. Indeed, it is not hard to see[8] that the minimum amount of probability weight to "move" in order to do so is at least what is needed to uniformize $\mathbf{p}$ on I_j and I_{j+1}. This latter process yields a distribution $\mathbf{p}'$ which puts weight $(\mathbf{p}(I_j) + \mathbf{p}(I_{j+1}))/((2+c)\,|I_j|)$ on each element of $I_j \cup I_{j+1}$, and the total variation distance between $\mathbf{p}$ and $\mathbf{p}'$ (a lower bound on its distance to monotonicity) is then

$$\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{p}') = \frac{\mathbf{p}(I_{j+1}) + \mathbf{p}(I_j)}{2+c} - \mathbf{p}(I_j) = \frac{\mathbf{p}(I_{j+1}) - (1+c)\mathbf{p}(I_j)}{2+c} > \frac{\varepsilon(2+c)}{2+c} = \varepsilon$$

which is a contradiction. □

This suggests immediately the following correcting scheme: to output samples according to $\tilde{\mathbf{p}}$, k-histogram

[8] E.g., by writing the ℓ_1 cost as the sum of the weight added/removed from "outside" the two buckets and the weight moved between the two buckets in order to satisfy the monotonicity condition, and minimizing this function.

on $\mathcal{I}$ defined by

$$\begin{aligned}\tilde{\mathbf{p}}(I_k) &= \lambda\left(\mathbf{p}(I_k)\right)\\ \tilde{\mathbf{p}}(I_{k-1}) &= \lambda\left(\mathbf{p}(I_{k-1}) + (2+c)\varepsilon\right)\\ &\vdots\\ \tilde{\mathbf{p}}(I_{k-j}) &= \lambda\left(\mathbf{p}(I_{k-j}) + j(2+c)\varepsilon\right)\end{aligned}$$

that is

$$\tilde{\mathbf{p}}(I_j) = \lambda\left(\mathbf{p}(I_j) + \varepsilon\sum_{i=j}^{k-1}\left(1+\frac{|I_{j+1}|}{|I_j|}\right)\right) \qquad 1 \le j \le k$$

where the normalizing factor is $\lambda \stackrel{\text{def}}{=} \left(1+\varepsilon(2+c)\frac{k(k-1)}{2}\right)^{-1}$. As, by Claim 5.5.3, adding weight decreasing by $(2+c)\varepsilon$ at each step fixes any pair of adjacent intervals whose average weights are not monotone, $\tilde{\mathbf{p}}/\lambda$ is a non-increasing non-negative function. The normalization by λ preserving the monotonicity, $\tilde{\mathbf{p}}$ is indeed a monotone distribution, as claimed.

It only remains to bound $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \tilde{\mathbf{p}})$:

$$\begin{aligned}2\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \tilde{\mathbf{p}}) &= \sum_{j=1}^{k}|\mathbf{p}(I_j) - \tilde{\mathbf{p}}(I_j)| = \sum_{j=1}^{k}\left|(1-\lambda)\mathbf{p}(I_j) - \lambda\varepsilon\sum_{i=j}^{k-1}(2+c)\right|\\ &\le (1-\lambda)\sum_{j=1}^{k}\mathbf{p}(I_j) + \lambda\varepsilon\sum_{j=1}^{k}\sum_{i=j}^{k-1}(2+c) = 1 - \frac{1-\varepsilon(2+c)\frac{k(k-1)}{2}}{1+\varepsilon(2+c)\frac{k(k-1)}{2}}.\end{aligned}$$

Finally, note that $\tilde{\mathbf{p}}$ is a mixture of $\mathbf{p}$ (with weight λ) and an explicit arithmetically non-increasing distribution; sampling from $\tilde{\mathbf{p}}$ is thus straightforward, and needs at most one sample from $\mathbf{p}$ for each draw. □

Remark 5.5.4. The above scheme can be easily adapted to the case where the ratio between consecutive intervals is not always the same, but is instead $|I_{j+1}|/|I_j| = 1 + c_j$ for some known $c_j \in [C_1, C_2]$; the result then depends on the ratio $C_2/C_1 = \Theta(1)$ as well.

As a direct corollary, this describes how to correct distributions which are promised to be (very) close to monotone, in a completely *oblivious* fashion: that is, the behavior of the corrector does not depend on what the input distribution is; furthermore, the probability of failure is null (i.e., $\delta = 0$).

Corollary 5.5.5 (Oblivious correcting of monotonicity)**.** *For every $\varepsilon' \in (0,1)$, there exists an (oblivious) sampling corrector for monotonicity, with parameters $\varepsilon = O\left(\varepsilon'^3/\log^2 n\right)$, $\varepsilon_1 = \varepsilon'$ and sample complexity $O(1)$.*

Proof. We will apply Lemma 5.5.2 for $k = O(\log n/\varepsilon')$ and $\mathcal{I}$ being the corresponding Birgé decomposition

(with parameter $\varepsilon'/2$). The idea is then to work with the "flattening" $\bar{\mathbf{p}}$ of $\mathbf{p}$: since $\mathbf{p}$ is ε-close to monotone, it is also $(\varepsilon'/2)$-close, and $\bar{\mathbf{p}}$ is both $(\varepsilon'/2)$-close to $\mathbf{p}$ and ε-close to monotone. Applying the correcting scheme with our value of k and c set to ε', the corrected distribution $\tilde{\mathbf{p}}$ is monotone, and

$$\mathrm{d}_{\mathrm{TV}}(\bar{\mathbf{p}}, \tilde{\mathbf{p}}) \leq 1 - \frac{1 - \varepsilon(2+\varepsilon')\frac{k(k-1)}{2}}{1 + \varepsilon(2+\varepsilon')\frac{k(k-1)}{2}} \leq \frac{\varepsilon'}{2}$$

where the last inequality derives from the fact that $k^2\varepsilon = O(\varepsilon')$. This in turn implies by a triangle inequality that $\tilde{\mathbf{p}}$ is ε'-close to $\mathbf{p}$. Finally, observe that, as stated in the lemma, $\tilde{\mathbf{p}}$ can be easily simulated given access to $\mathbf{p}$, using either 0 or 1 draw: indeed, $\tilde{\mathbf{p}}$ is a mixture with known weights of an explicit distribution and $\bar{\mathbf{p}}$, and access to the latter can be obtained from $\mathbf{p}$. □

Remark 5.5.6. An interesting feature of the above construction is that does not *only* yields a $O(1)$-query corrector from sampling to sampling: it similarly implies a corrector from ORACLE to ORACLE with query complexity $O(1)$, for ORACLE being (for instance) an evaluation or Cumulative Dual oracle (cf. Section 4.2). This follows from the fact that the corrected distribution $\tilde{\mathbf{p}}$ is of the form $\tilde{\mathbf{p}} = \lambda\mathbf{p} + (1-\lambda)P$, where both λ and P are fully known.

5.5.3 Correcting with Cumulative Dual access

In this section we prove the following result, which shows that correcting monotonicity with $o(\log n)$ queries (on expectation) is possible when one allows a stronger type of access to the original distribution. In particular, recall that in the Cumulative Dual model (as defined in Section 4.2) the algorithm is allowed to make, in addition to the usual draws from the distribution, *evaluation queries* to its cumulative distribution function.[9]

Theorem 5.5.7. *For any $\varepsilon \in (0,1]$, any number of queries m and $\varepsilon_1 = O(\varepsilon)$ as in the definition, there exists a sampling corrector for monotonicity from Cumulative Dual to* SAMP *with* expected *sample complexity* $O\left(\sqrt{m \log n/\varepsilon}\right)$.

In particular, since learning distributions in the Cumulative Dual model is easily seen to have query complexity $\Theta(\log n/\varepsilon)$ (e.g., by considering the lower bound instance of [32]), the above corrector beats the "learning approach" as long as $m = o(\log n/\varepsilon)$.

Remark 5.5.8. One may look at this ability to correct up to $o(\log n/\varepsilon)$ samples cautiously, as it is well-known that the lower bound for testing monotonicity of distributions is $\Omega\left(\sqrt{n}/\varepsilon^2\right)$ already [19]. However, this lower bound only establishes a *worst-case* indistinguishability: as pointed out in Remark 5.2.5, for many "typical" distributions that are ε-close to monotone, as few as $O(1/\varepsilon)$ samples would be sufficient to detect the discrepancy from monotone (and compromise the correctness of any algorithm relying on these uncorrected samples).

[9] We remark that our algorithm will in fact only use this latter type of access, and will not rely on its ability to draw samples from $\mathbf{p}$.

5.5.3.1 Overview and discussion

A natural idea would be to first group the elements into consecutive intervals (the "buckets"), and correct this distribution (now a histogram over these buckets) at two levels. That is, start by correcting it optimally at a coarse level (the "superbuckets," each of them being a group of consecutive buckets); then, every time a sample has to be generated, draw a superbucket from this coarse distribution and correct at a finer level *inside this superbucket*, before outputting a sample from the corrected local distribution (i.e. conditional on the superbucket that was drawn and corrected). While this approach seems tantalizing, the main difficulty with it lies in the possible boundary violations between superbuckets: that is, even if the average weights of the superbuckets are non-increasing, and the distribution over buckets is non-decreasing inside each superbucket, it might still be the case that there are local violations between adjacent superbuckets. (I.e., the boundaries are bad.) A simple illustration is the sequence $\langle .5, .1, .3, .1 \rangle$, where the first "superbucket" is $(.5, .1)$ and the second $(.3, .1)$. The average weight is decreasing, and the sequence is locally decreasing inside each superbucket; yet overall the sequence is not monotone.

Thus, we have to consider 3 kinds of violations:

(i) global superbucket violations: the average weight of the superbuckets is not monotone.

(ii) local bucket violations: the distribution of the buckets inside some superbucket is not monotone.

(iii) superbucket boundary violations: the probability of the last bucket of a superbucket is lower than the probability of the first bucket of the next superbucket.

The ideas underlying our sampling corrector (which is granted both sampling and cumulative query access to the distribution, as defined in the Cumulative Dual access model) are quite simple: after reducing *via* standard techniques the problem to that of correcting a *histogram* supported of logarithmically many intervals (the "Birgé decomposition"), we group these ℓ intervals in K "superbuckets," each containing L consecutive intervals from that histogram ("buckets"). (As a guiding remark, our overall goal is to output samples from a corrected distribution using $o(\ell)$ queries, as otherwise we would already use enough queries to actually learn the distribution.) This two-level approach will allow us to keep most of the corrected distribution implicit, only figuring out (and paying queries for that) the portions from which we will ending up outputting samples.

By performing K queries, we can exactly learn the coarse distribution on superbuckets, and correct it for monotonicity (optimally, e.g. by a linear program ensuring the average weights of the superbuckets are monotone), solving the issues of type (i). In order to fix the boundary violations (iii) on-the-go, the idea is to allocate to each superbucket an extra *budget* of probability weight that *can* be used for these boundary corrections. Importantly, if this budget is not entirely used the sampling process restarts from the beginning with a probability corresponding with the remaining budget. This effectively ends up simulating a distribution where each superbucket was assigned an extra weight matching exactly what was needed for the correction, *without having to figure out all these quantities beforehand* (as this would cost too many queries).

Essentially, each superbucket is selected according to its "potential weight," that includes both the actual

probability weight it has and the extra budget it is allowed to use for corrections. Whenever a superbucket S_i is selected this way, we first perform optimal local corrections of type (ii) both on it and the previous superbucket S_{i-1} making a cdf query at every boundary point between buckets in order to get the weights of all $2L$ buckets they contain, and then computing the optimal fix: at this point, the distribution is monotone inside S_i (and inside S_{i-1}). After this, we turn to the possible boundary violations of type (iii) between S_{i-1} and S_i, by "pouring" some of the weight from S_i's budget to fill "valleys" in the last part of S_{i-1}. Once this water-filling has ended,[10] we may not have used all of S_i's budget (but as we shall see we make sure we never run out of it): the remaining portion is thus redistributed to the whole distribution by restarting the sampling process from the beginning with the corresponding probability. Note that as soon as we know the weights of all $2L$ Birgé buckets, no more cdf queries are needed to proceed.

5.5.3.2 Preliminary steps (preprocessing)

First step: reducing to $\mathbf{p}$ to a histogram. Given cdfsamp access (i.e., granting both SAMP and cumulative distribution function (cdf) query access) to an unknown distribution $\mathbf{p}$ over $[n]$ which is ε-close to monotone, we can simulate cdfsamp access to its Birgé flattening $\mathbf{p}^{(1)} \stackrel{\text{def}}{=} \Phi_\varepsilon(\mathbf{p})$, also ε-close to monotone and 3ε-close to $\mathbf{p}$ (by Corollary 1.4.6). For this reason, we hereafter work with $\mathbf{p}^{(1)}$ instead of $\mathbf{p}$, as it has the advantage of being an ℓ-histogram for $\ell = O(\log n/\varepsilon)$. Because of this first reduction, it becomes sufficient to perform cdf queries on the buckets (and not the individual elements of $[n]$), which altogether entirely define $\mathbf{p}^{(1)}$.

Second step: global correcting of the superbuckets. By making K cdf queries, we can figure out exactly the quantities $\mathbf{p}^{(1)}(S_1), \ldots, \mathbf{p}^{(1)}(S_K)$. By running a linear program, we can re-weight them to obtain a distribution $\mathbf{p}^{(2)}$ such that (a) the averages $\frac{\mathbf{p}^{(2)}(S_j)}{|S_j|}$ are non-increasing; (b) the conditional distributions of $\mathbf{p}^{(1)}$ and $\mathbf{p}^{(2)}$ on each superbucket are identical ($\mathbf{p}^{(2)}{}_{S_j} = \mathbf{p}^{(1)}{}_{S_j}$ for all $j \in [K]$); and (c) $\sum_j \left|\mathbf{p}^{(2)}(S_j) - \mathbf{p}^{(1)}(S_j)\right|$ is minimized.

Third step: allocating budgets to superbuckets. For reasons that will become clear in the subsequent, "water-filling" step, we want to give each superbucket S_j a budget b_j of "extra weight" *added to its first bucket* $S_{j,1}$ that can be used for local corrections when needed – if it uses only part of this budget during the local correction, it will need to "give back" the surplus. To do so, define $\mathbf{p}^{(3)}$ as the distribution such that

- $\mathbf{p}^{(3)}(S_j) = \lambda^{(3)}(\mathbf{p}^{(2)}(S_j) + b_j)$, $j \in [K]$ (where $b_j \stackrel{\text{def}}{=} \mathbf{p}^{(2)}(S_j)/(1+\varepsilon)$ for $j \in [K]$; and $\lambda^{(3)} \stackrel{\text{def}}{=} (1 + \sum_j b_j)^{-1}$ is a normalization factor). Note that $\sum_j b_j = 1/(1+\varepsilon) \in [1/2, 1]$, so that $\lambda^{(3)} \in [1, 2]$.
- The conditional distribution on $S_j \setminus S_{j,1}$ satisfy $\mathbf{p}^{(3)}{}_{S_j \setminus S_{j,1}} = \mathbf{p}^{(2)}{}_{S_j \setminus S_{j,1}}$ for all $j \in [K]$.

[10] We borrow this graphic analogy with the process of pouring water from [7], which employs it in a different context (in order to bound the running time of an algorithm by a potential-based argument.).

That is, $\mathbf{p}^{(3)}$ is a version of $\mathbf{p}^{(2)}$ where each superbucket is re-weighted, but "locally" looks the same inside each superbucket *except for the first bucket of each superbucket, that received the additional "budget weight."* Observe that since the size $|S_j|$ of the superbuckets is multiplicatively increasing by an $(1+\varepsilon)$ factor (as a consequence of Birgé bucketing), the averages $\mathbf{p}^{(3)}(S_j)/|S_j|$ will remain non-increasing. That is, the average changes by less for "big" values of j's than for small values, as the budget is spread over more elements.

Remark 5.5.9. $\mathbf{p}^{(3)}$ is uniquely determined by ε, n and $\mathbf{p}$, and can be explicitly computed using K cdf queries.

5.5.3.3 Sampling steps (correcting while sampling)

Before going further, we describe a procedure that will be necessary for our fourth step, as it will be the core subroutine allowing us to perform local corrections *between superbuckets*.

Water-filling Partition each superbucket S_i into range H_i, M_i and L_i where (assuming the buckets in S_i are monotone):

- $m_i = \mathbf{p}^{(3)}(S_i)/|S_i|$ is the initial value of the average value of superbucket S_i [this does not change throughout the procedure]
- H_i are the (leftmost) elements whose value is greater than m_i [these elements may move to M_i or stay in H_i]
- M_i are the (middle) elements whose value is equal to m_i [these elements stay in M_i]
- L_i are the (rightmost) elements whose value is less than m_i [these elements may move to M_i or stay in L_i]
- $\min_i$ is the minimum probability value in superbucket S_i [this updates throughout the procedure]
- $\max_i$ is the maximum probability value in superbucket S_i [this updates throughout the procedure]

Let $e_i \stackrel{\text{def}}{=} \sum_{x\in H_i}(p(x)-m_i)$ to be the *surplus* (so that if $e_i = 0$ then $H_i = \emptyset$ and the superbucket is said to be *dry*) and $d_i \stackrel{\text{def}}{=} \sum_{x\in L_i}(m_i - p(x))$ to be the *deficit* (if $d_i = 0$ then $L_i = \emptyset$ and the superbucket is said to be *full*).

Algorithm 32 Procedure water-fill

1: take an infinitesimal amount ∂p from the top of the max, leftmost buckets of H_{i+1}, in superbucket S_{i+1} (this would be from the first bucket and any other buckets that have the same probability)
2: pour ∂p into superbucket S_i (this would land in the min, rightmost buckets of L_i, in superbucket S_i and spread to the left, to buckets that have the same probability, just like water)

Algorithm 33 Procedure front-fill

while the surplus e_{i+1} is greater than the extra budget b_{i+1} allocated in Section 5.5.3.2 **do**
  take an infinitesimal amount ∂p from the top of the max, leftmost elements of H_{i+1}, in superbucket S_{i+1} (this would be from the first bucket and any other buckets that have the same probability)
  pour ∂p into the very first bucket of the domain, $S_{1,1}$.
end while
return the total amount f_i of weight poured into $S_{1,1}$.

Algorithm 34 Procedure water-boundary-correction

Require: Superbucket index $j = i + 1$, with initial weight $\mathbf{p}^{(3)}(S_{i+1})$.

move weight from the surplus of H_{i+1} into L_i using water-fill until:
  (a) $\max_{i+1} \leq \min_i$; or
  (b) $L_i = \emptyset$ (S_i is full) – i.e. $\min_i = m_i$; or
  (c) $H_{i+1} = \emptyset$ (S_{i+1} is dry) – this can only happen if $e_{i+1} < d_i$ ▷ This should never happen because of the "budget allocation" step.

Note that the distribution might not yet be monotone on $S_i \cup S_{i+1}$, if one of the last two conditions is reached first. If this is the case, then do further correction:
  (a) if $L_i = \emptyset$ then do front-fill until $\max_{i+1} \leq \min_i$ (this will happen before $H_{i+1} = \emptyset$)
  (b) if $H_{i+1} = \emptyset$ then abort and **return** fail ▷ This should never happen because of the "budget allocation" step.

return the list $B_1, \ldots, B_s$ of buckets in $T_i \stackrel{\text{def}}{=} L_i \cup S_{i+1}$, along with the weights $w_1, \ldots, w_s$ they have from w after the redistribution and the portion ε_i of the budget that was not used and the portion f_i that was moved by front-fill (so that $\lambda^{(3)}\varepsilon_i + f_i + \sum_{t=1}^{s} w_t = \mathbf{p}^{(3)}(S_{i+1})$).

Sampling procedure Recall that we now start and work with $\mathbf{p}^{(3)}$, as obtained in Section 5.5.3.2.

- Draw a superbucket S_{i+1} according to the distribution $\mathbf{p}^{(3)}(S_1)$, $\mathbf{p}^{(3)}(S_K)$ on $[K]$.
- If $S_{i+1} \neq S_1$ (we did not land in the first superbucket):
  - Obtain (*via* cdf queries, if they were not previously known) the $2L$ values $\mathbf{p}^{(3)}(S_{i,j})$ $\mathbf{p}^{(3)}(S_{i+1,j})$ ($j \in [L]$) of the buckets in superbuckets S_i, S_{i+1}.
  - Correct them (separately for each of the two superbuckets) optimally for monotonicity, e.g. via linear programming (if that was not done in a prior stage of sample generation), ignoring the extra budget b_i and b_{i+1} on the first bucket of each superbucket. Compute H_i, M_i, L_i and $H_{i+1}, M_{i+1}, L_{i+1}$.
  - Call water-boundary-correction on $(i + 1)$ using the extra budget only if S_{i+1} becomes dry and not counting it while trying to satisfy condition (a).[11]
- If $S_{i+1} = S_1$ (we landed in the first superbucket), we proceed similarly as per the steps above, except for the water-boundary-correction. That is, we only correct locally S_1 for monotonicity.

[11] At this point, the "new" distribution $\mathbf{p}^{(4)}$ (which is at least partly implicit, as only known at a very coarse level over superbuckets and locally for some buckets inside $S_i \cup S_{i+1}$) obtained is monotone over the superbuckets (water-boundary-correction does not violate the invariant that the distribution over superbuckets is monotone), is monotone inside both S_i and S_{i+1}, and furthermore is monotone over $S_i \cup S_{i+1}$. Even more important, the fact that $\min_i \geq \max_{i+1}$ will ensure applying the same process in the future, e.g. to S_{i+2}, will remain consistent with regard to monotonicity.

- During the execution of water-boundary-correction, the water-filling procedure may have used some of the initial "allocated budget" b_{i+1} to pour into L_i. Let $\varepsilon_i \in [0, b_{i+1}]$ be the amount of the budget remaining (not used).
  - with probability $p_i \stackrel{\text{def}}{=} \lambda^{(3)}\varepsilon_i/\mathbf{p}^{(3)}(S_{i+1})$, restart the sampling process from the beginning (this is the "budget redistribution step," which ensures the correction only uses *what it needs* for each superbucket).
  - with probability $q_i \stackrel{\text{def}}{=} f_i/\mathbf{p}^{(3)}(S_{i+1})$, where f_i is the weight moved by the procedure front-fill, output from the very first bucket of the domain.
  - with the remaining probability, output a sample from the new (conditional) distribution on the buckets in $T_i \stackrel{\text{def}}{=} L_i \cup S_{i+1}$. This is the conditional distribution defined on T_i by the weights $w_1, \ldots, w_s$, as returned by water-boundary-correction.

Note that the distribution we output from if we initially select the superbucket S_{i+1}, is supported on $L_i \cup S_{i+1}$. Moreover, conditioning on $M_{i+1} \cup L_{i+1}$ we get exactly the conditional distribution $\mathbf{p}^{(3)}_{M_{i+1}\cup L_{i+1}}$. (This ensures that from each bucket there is a unique superbucket that has to be picked initially for the bucket's weight to be modified.) Observe that as defined above, buckets from $L_i \subseteq S_i$ can be outputted from either because superbucket S_i was picked, or because S_{i+1} was drawn and some of its weight was reassigned to L_i by water-boundary-correction. The probability of outputting any bucket in L_i is then the sum of the probabilities of the two types of events.

5.5.3.4 Analysis

The first observation is that the distribution of any sample output by the sampling process described above is not only consistent, but completely determined by n, ε and $\mathbf{p}$:

Claim 5.5.10. *The process described in Section 5.5.3.2 and 5.5.3.3 uniquely defines a distribution $\tilde{\mathbf{p}}$, which is a function of $\mathbf{p}$, n and $\varepsilon \in (0, 1)$ only.*

Claim 5.5.11. *The expected number of queries necessary to output m samples from $\tilde{\mathbf{p}}$ is upper bounded by $K + 4mL\varepsilon$.*

Proof. The number of queries for the preliminary stage is K; after this, generating a sample requires X queries, where X is a random variable satisfying

$$X \leq 2L + RX'$$

where X, X' are independent and identically distributed, and R is a Bernoulli random variable independent of X' and with parameter Δ (itself a random variable depending on X: Δ takes value p_i when the first draw selects superbucket $i + 1$), corresponding to the probability of restarting the sampling process from the

beginning. It follows that

$$\mathbb{E}[X] \leq 2L + \mathbb{E}[R]\,\mathbb{E}[X] = 2L + \mathbb{E}[\Delta]\,\mathbb{E}[X]\,.$$

Using the fact that $\mathbb{E}[\Delta] = \sum_{i\in[K]} \mathbf{p}^{(3)}(S_{i+1})p_i = \sum_{i\in[K]} \mathbf{p}^{(3)}(S_{i+1})\frac{\lambda^{(3)}\varepsilon_i}{\mathbf{p}^{(3)}(S_{i+1})} \leq \lambda^{(3)}\sum_{i\in[K]} b_i \in [1/3, 1/2]$ and rearranging, we get $\mathbb{E}[X] \leq 4L$. □

Lemma 5.5.12. *If $\mathbf{p}$ is a distribution on $[n]$ satisfying $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathcal{M}) \leq \varepsilon$, then the distribution $\tilde{\mathbf{p}}$ defined above is monotone.*

Proof. Observe that as the average weights of the superbuckets in $\mathbf{p}^{(2)}$ are non-increasing, the definition of $\mathbf{p}^{(3)}$ along with the fact that the lengths of the superbuckets are (multiplicatively) increasing implies that the average weights of the superbuckets in $\mathbf{p}^{(3)}$ are also non-increasing. In more detail, fix $1 \leq i \leq K-1$; we have

$$\frac{\mathbf{p}^{(2)}(S_i)}{|S_i|} \geq \frac{\mathbf{p}^{(2)}(S_{i+1})}{(1+\varepsilon)\,|S_i|}$$

using the fact that $|S_j| = (1+\varepsilon)\,|S_{j-1}|$. From there, we get that

$$(1+\varepsilon)\mathbf{p}^{(2)}(S_i) \geq \mathbf{p}^{(2)}(S_{i+1})$$

or equivalently

$$\frac{b_i + \mathbf{p}^{(2)}(S_i)}{|S_i|} = \frac{\mathbf{p}^{(2)}(S_i) + (1+\varepsilon)\mathbf{p}^{(2)}(S_i)}{(1+\varepsilon)\,|S_i|} \geq \frac{\mathbf{p}^{(2)}(S_{i+1}) + (1+\varepsilon)\mathbf{p}^{(2)}(S_{i+1})}{(1+\varepsilon)^2\,|S_i|} = \frac{b_{i+1} + \mathbf{p}^{(2)}(S_{i+1})}{|S_{i+1}|}$$

showing that before renormalization (and therefore after as well) the average weights of the superbuckets in $\mathbf{p}^{(3)}$ are indeed non-increasing. Rephrased, this means that the sequence of m_i's, for $i \in [K]$, is monotone. Moreover, notice that by construction the distribution $\tilde{\mathbf{p}}$ is monotone within each superbucket: indeed, it is explicitly made so one superbucket at a time, in the third step of the sampling procedure. After a superbucket has been made monotone this way, it only be changed by water-filling which by design can never introduce new violations: the weight is always moved "to the left," with the values m_i's acting as boundary conditions to stop the waterfilling process and prevent new violations, or moved to the first element of the domain.

It only remains to argue that monotonicity is not violated at the boundary of two consecutive superbuckets. But since the water-boundary-correction, if it does not abort, guarantees that the distribution is monotone between consecutive buckets as well (as $m_{i+1} \leq \max_{i+1} \leq \min_i \leq m_i$), it it sufficient to show that water-boundary-correction never returns fail. This is ensured by the "budget allocation" step, which by providing H_{i+1} with up to an additional b_{i+1} to spread into L_i guarantees it will become dry. Indeed, if this happened then it would mean that correcting this particular violation (before the budget allocation, which only affects the first elements of the superbuckets) in $\mathbf{p}^{(2)}$ required to move more than b_{i+1} weight, contradicting the fact that the average weights of the superbuckets in $\mathbf{p}^{(2)}$ were non-increasing. In more detail, the maximum amount of weight to "pour" in order to fill L_i is in the case where H_{i+1} is empty (i.e., the distribution on S_{i+1} is already

uniform) but L_i is (almost) all of S_i (i.e., all the weight in S_i is in the first bucket). To correct this with our waterfilling procedure, one would have to pour $|S_i| \cdot \frac{\mathbf{p}^{(2)}(S_{i+1})}{|S_{i+1}|} = \frac{\mathbf{p}^{(2)}(S_{i+1})}{1+\varepsilon}$ weight in L_i, which is exactly our choice of value for b_{i+1}. □

Lemma 5.5.13. *If* $\mathbf{p}$ *is a distribution on* $[n]$ *satisfying* $\mathrm{d_{TV}}(\mathbf{p}, \mathcal{M}) \leq \varepsilon$*, then* $\mathrm{d_{TV}}(\mathbf{p}, \tilde{\mathbf{p}}) = O(\varepsilon)$.

Proof. We will bound separately the distances $\mathbf{p}$ to $\mathbf{p}^{(1)}$, $\mathbf{p}^{(1)}$ to $\mathbf{p}^{(2)}$ and $\mathbf{p}^{(2)}$ to $\tilde{\mathbf{p}}$, and conclude by the triangle inequality.

- First of all, the distance $\mathrm{d_{TV}}\left(\mathbf{p}, \mathbf{p}^{(1)}\right)$ is at most 3ε, by properties of the Birgé decomposition (and as $\mathrm{d_{TV}}(\mathbf{p}, \mathcal{M}) \leq \varepsilon$).
- We now turn to $\mathrm{d_{TV}}\left(\mathbf{p}^{(1)}, \mathbf{p}^{(2)}\right)$, showing that it is at most ε: in order to do so, we introduce $\mathbf{p}'$, the piecewise-constant distribution obtained by "flattening" $\mathbf{p}^{(1)}$ on each of the K superbuckets (so that $\mathbf{p}'(S_j) = \mathbf{p}^{(1)}(S_j)$ for all j). It is not hard to see, e.g. by the data processing inequality for total variation distance, that $\mathbf{p}'$ is also ε-close to monotone, and additionally that the closest monotone distribution M' can also be assumed to be constant on each superbucket.

  Consider now the transformation that re-weights in $\mathbf{p}'$ each superbucket S_j by a factor $\alpha_j > 0$ to obtain M'; it is straightforward to see from Section 5.5.3.2 that this transformation maps $\mathbf{p}^{(1)}$ to $\mathbf{p}^{(2)}$. Therefore,

$$\begin{aligned}
2\mathrm{d_{TV}}\left(\mathbf{p}^{(1)}, \mathbf{p}^{(2)}\right) &= \sum_{j\in[K]} \sum_{x\in S_j} \left|\mathbf{p}^{(1)}(x) - \mathbf{p}^{(2)}(x)\right| = \sum_{j\in[K]} \sum_{x\in S_j} \left|\mathbf{p}^{(1)}(x) - \alpha_j \mathbf{p}^{(1)}(x)\right| \\
&= \sum_{j\in[K]} \sum_{x\in S_j} \mathbf{p}^{(1)}(x) \cdot |1-\alpha_j| = \sum_{j\in[K]} \mathbf{p}^{(1)}(S_j) \cdot |1-\alpha_j| \\
&= \sum_{j\in[K]} \sum_{x\in S_j} \mathbf{p}^{(1)}(x) \cdot |1-\alpha_j| = \sum_{j\in[K]} \mathbf{p}^{(1)}(S_j) \cdot \left|1 - \frac{M'(S_j)}{\mathbf{p}'(S_j)}\right| \\
&= \sum_{j\in[K]} |\mathbf{p}'(S_j) - M'(S_j)| = 2\mathrm{d_{TV}}(\mathbf{p}', M') \leq 2\varepsilon.
\end{aligned}$$

- To bound $\mathrm{d_{TV}}\left(\mathbf{p}^{(2)}, \tilde{\mathbf{p}}\right)$, first consider the distribution $\mathbf{p}''$ obtained by correcting optimally $\mathbf{p}^{(2)}$ for monotonicity *inside each superbucket separately*. That is, $\mathbf{p}''$ is the distribution satisfying monotonicity on each S_j (separately) and $\mathbf{p}''(S_j) = \mathbf{p}^{(2)}(S_j)$ for each $j \in [K]$; and minimizing

$$\sum_{j\in[K]} \sum_{i\in[L]} \left|\mathbf{p}''(S_{j,i}) - \mathbf{p}^{(2)}(S_{j,i})\right|$$

  (or, equivalently, minimizing $\sum_{i\in[L]} \left|\mathbf{p}''(S_{j,i}) - \mathbf{p}^{(2)}(S_{j,i})\right|$ for all $j \in [K]$). The first step is to prove that $\mathbf{p}''$ is close to $\mathbf{p}^{(2)}$: recall first that by the triangle inequality, our previous argument implies that $\mathbf{p}^{(2)}$ is (2ε)-close to monotone. Therefore, the (related) optimization problem asking to find a non-negative function P that minimizes the same objective, but under the different constraints "P is monotone on $[n]$ and $P([n]) = \mathbf{p}^{(2)}([n])$" has a solution P whose total variation distance from $\mathbf{p}^{(2)}$ is at most 2ε.

But P can be used to obtain P', solution to the original problem, by re-weighting each superbucket S_j the following way:

$$P'(x) \stackrel{\text{def}}{=} P(x) \cdot \frac{\mathbf{p}^{(2)}(S_j)}{P(S_j)}, \quad x \in S_j.$$

Clearly, P' satisfies the constraints of the first optimization problem; moreover,

$$\begin{aligned}
2\mathrm{d}_{\mathrm{TV}}\Big(P', \mathbf{p}^{(2)}\Big) &= \sum_{j\in[K]} \sum_{x\in S_j} \Big|P'(x) - \mathbf{p}^{(2)}(x)\Big| = \sum_{j\in[K]} \sum_{x\in S_j} \left|P(x)\frac{\mathbf{p}^{(2)}(S_j)}{P(S_j)} - \mathbf{p}^{(2)}(x)\right| \\
&\leq \sum_{j\in[K]} \sum_{x\in S_j} \Big|P(x) - \mathbf{p}^{(2)}(x)\Big| + \sum_{j\in[K]} \sum_{x\in S_j} P(x) \left|\frac{\mathbf{p}^{(2)}(S_j)}{P(S_j)} - 1\right| \\
&= 2\mathrm{d}_{\mathrm{TV}}\Big(P, \mathbf{p}^{(2)}\Big) + \sum_{j\in[K]} \Big|\mathbf{p}^{(2)}(S_j) - P(S_j)\Big| \leq 4\mathrm{d}_{\mathrm{TV}}\Big(P, \mathbf{p}^{(2)}\Big) \\
&\leq 8\varepsilon,
\end{aligned}$$

where we used the fact that $\sum_{j\in[K]} \big|\mathbf{p}^{(2)}(S_j) - P(S_j)\big| = \sum_{j\in[K]} \Big|\sum_{x\in S_j} \big(\mathbf{p}^{(2)}(x) - P(x)\big)\Big| \leq \sum_{j\in[K]} \sum_{x\in S_j} \big|\mathbf{p}^{(2)}(x) - P(x)\big|$. As $\mathrm{d}_{\mathrm{TV}}\big(P', \mathbf{p}^{(2)}\big)$ is an upperbound on the optimal value of the optimization problem, we get $\mathrm{d}_{\mathrm{TV}}\big(\mathbf{p}'', \mathbf{p}^{(2)}\big) \leq 4\varepsilon$.

The next and last step is to bound $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}'', \tilde{\mathbf{p}})$, and show that it is $O(\varepsilon)$ as well. To see why this will allow us to conclude, note that $\mathbf{p}''$ is the intermediate distribution that the sampling process we follow would define, it there was neither extra budget allocated nor water-boundary-correction. Put differently, $\tilde{\mathbf{p}}$ is derived from $\mathbf{p}''$ by adding the "right amount of extra budget $b'_j \in [0, b_j]$" to S_j, then pouring it to S_{j-1} by waterfilling and front-filling; and normalizing afterwards by $(1 + \sum_{j\in[K]} b'_j)^{-1}$. Writing $\tilde{\mathbf{p}}''$ for the result of the transformation above before the last renormalization step, we can bound $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}'', \tilde{\mathbf{p}})$ by

$$\begin{aligned}
2\mathrm{d}_{\mathrm{TV}}(\mathbf{p}'', \tilde{\mathbf{p}}) &= \|\mathbf{p}'' - \tilde{\mathbf{p}}\|_1 \leq \|\mathbf{p}'' - \tilde{\mathbf{p}}''\|_1 + \|\tilde{\mathbf{p}}'' - \tilde{\mathbf{p}}\|_1 \\
&\leq \sum_{j\in[K]} b'_j + \sum_{j\in[K]} f_j + \sum_{x\in[n]} \left|\Big(1 + \sum_{j\in[K]} b'_j\Big)\tilde{\mathbf{p}}(x) - \tilde{\mathbf{p}}(x)\right| \\
&\leq \sum_{j\in[K]} b'_j + \sum_{j\in[K]} f_j + |\Big(1 + \sum_{j\in[K]} b'_j\Big) - 1| = 2\sum_{j\in[K]} b'_j + \sum_{j\in[K]} f_j
\end{aligned}$$

where $f_j \geq 0$ is defined as the amount of weight moved from H_j to the first element of the domain during the execution of water-boundary-correction, if front-fill is called, and the bound on $\|\mathbf{p}'' - \tilde{\mathbf{p}}''\|_1$ comes from the fact that $\tilde{\mathbf{p}}''$ pointwise dominates $\mathbf{p}''$, and has a total additional $\sum_{j\in[K]} b'_j$ weight.

It then suffices to bound the quantities $\sum_{j\in[K]} f_j$ and $\sum_{j\in[K]} b'_j$, using for this the fact that by the triangle inequality $\mathbf{p}''$ is itself (6ε)-close to monotone. The at most K intervals where $\mathbf{p}''$ violates monotonicity (which are fixed by using the b'_j's) are disjoint, and centered at the boundaries between consecutive superbuckets: i.e., each of them is in a interval $V_j \subseteq L_{j-1} \cup H_j \subsetneq S_{j-1} \cup S_j$. Because of

this disjointness, each transformation of $\mathbf{p}''$ into a monotone distribution must add weight in $V_j \cap L_{j-1}$ or subtract some from $V_j \cap H_j$ to remove the corresponding violation. By definition of b'_j (as minimum amount of additional weight to bring to L_{j-1} when spreading weight from H_j to L_{j-1}), this implies that any such transformation has to "pay" at least $b'_j/2$ (in total variation distance) to fix violation V_j. From the bound on $\mathrm{d_{TV}}(\mathbf{p}'', \mathcal{M})$, we then get $\sum_{j\in[K]} b'_j \leq 12\varepsilon$. A similar argument shows than $\sum_{j\in[K]} f_j \leq 12\varepsilon$ as well, which in turn yields $\mathrm{d_{TV}}(\mathbf{p}'', \tilde{\mathbf{p}}) \leq 18\varepsilon$.

- Putting these bounds together, we obtain

$$\begin{aligned}\mathrm{d_{TV}}(\mathbf{p}, \tilde{\mathbf{p}}) &\leq \mathrm{d_{TV}}\left(\mathbf{p}, \mathbf{p}^{(1)}\right) + \mathrm{d_{TV}}\left(\mathbf{p}^{(1)}, \mathbf{p}^{(2)}\right) + \mathrm{d_{TV}}\left(\mathbf{p}^{(2)}, \mathbf{p}''\right) + \mathrm{d_{TV}}(\mathbf{p}'', \tilde{\mathbf{p}}) \\ &\leq 3\varepsilon + \varepsilon + 4\varepsilon + 18\varepsilon = 26\varepsilon.\end{aligned}$$

□

We are finally in position of proving the main result of the section:

Proof of Theorem 5.5.7. The theorem follows from Claim 5.5.10, Claim 5.5.11, Lemma 5.5.12 and Lemma 5.5.13, setting $K = mL = \sqrt{m\ell}$ (where $\ell = O(\log n/\varepsilon)$ as defined in the Birgé decomposition). □

5.6 Constrained Error Models

In the previous sections, no assumption was made on the form of the error, only on the amount. In this section, we suggest a model of errors capturing the deletion of a whole "chunk" of the distribution. We refer to this model as the *missing data model*, where we assume that some ε probability is removed by taking out all the weight of an arbitrary interval $[i, j]$ for $1 \leq i < j \leq n$ and redistributing it on the rest of the domain as per rejection sampling.[12] We show that one can design sampling improvers for monotone distributions with arbitrarily large amounts of error. Hereafter, $\mathbf{p}$ will denote the original (monotone) distribution (before the deletion error occured), and $\mathbf{p}' = \mathbf{p}^{i,j}$ the resulting (faulty) one, to which the sampling improver has access. Our sampling improver follows what could be called the "learning-just-enough" approach: instead of attempting to approximate the *whole* unaltered original distribution, it only tries to learn the values of i, j; and then generates samples "on-the-fly." At a high level, the algorithm works by (i) detecting the location of the missing interval (drawing a large (but still independent of n) number of samples), then (ii) estimating the weight of this interval under the original, unaltered distribution; and finally (iii) filling this gap uniformly by moving the right amount of probability weight from the end of the domain. To perform the first stage, we shall follow a paradigm first appeared in [63], and utilize testing as a subroutine to detect "when enough learning has been done."

[12] That is, if $\mathbf{p}$ was the original distribution, the faulty one $\mathbf{p}^{(i,j)}$ is formally defined as $(1+\varepsilon)\mathbb{1}_{[n]\setminus[i,j]} \cdot \mathbf{p} - \varepsilon \cdot \mathbf{u}_{[i,j]}$, where $\varepsilon = \mathbf{p}([i, j])$.

Theorem 5.6.1. *For the class of distributions following the "missing data" error model, there exists a batch sampling improver* MISSING-DATA-IMPROVER*, that, on input ε, q, δ and α, achieves parameters $\varepsilon_1 = O(\varepsilon)$ and any $\varepsilon_2 < \varepsilon$; and has sample complexity $\tilde{O}\left(\frac{1}{\varepsilon_2^3} \log \frac{1}{\delta}\right)$ independent of ε.*

The detailed proof of our approach, as well as the description of MISSING-DATA-IMPROVER, are given in the next subsection.

5.6.1 Proof of Theorem 5.6.1

Before describing further the way to implement our 3-stage approach, we will need the following lemmata. The first examines the influence of adding or removing probability weight ε from a distribution, as it is the case in the missing data model:

Lemma 5.6.2. *Let $\mathbf{p}$ be a distribution over $[n]$ and $\varepsilon > 0$. Suppose $\mathbf{p}' \stackrel{\text{def}}{=} (1+\varepsilon)\mathbf{p} - \varepsilon\mathbf{p}_1$, for some distribution $\mathbf{p}_1$. Then $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{p}') \leq \varepsilon$.*

The proof follows from a simple application of the triangle inequality to the ℓ_1 distance between $\mathbf{p}$ and $\mathbf{p}'$. We note that the same bound applies if $\mathbf{p}' = (1-\varepsilon)\mathbf{p} + \varepsilon\mathbf{p}_1$.

The next two lemmata show that the distance to monotonicity of distributions falling into this error model can be bounded in terms of the probability weight right after the missing interval.

Lemma 5.6.3. *Let $\mathbf{p}$ be a monotone distribution and $\mathbf{p}' = \mathbf{p}^{(i,j)}$ be the faulty distribution. If $\mathbf{p}([j+1, 2j-i+1]) > \varepsilon$, then $\mathbf{p}'$ is $\varepsilon/2$-far from monotone.*

Proof. Let $L \stackrel{\text{def}}{=} j - i$ be the length of the interval where the deletion occurred. Since the interval $[j+1, 2j-i+1]$ has the same length as $[i, j]$ and weight $p > \varepsilon$, the average weight of an element is at least $\frac{\varepsilon}{L}$. Every monotone distribution M should also be monotone on the interval $[i, 2j-i+1]$: therefore, one must have $M([i,j]) \geq M([j+1, 2j-i+1])$. Let $q \stackrel{\text{def}}{=} M([i,j])$. As $\mathbf{p}'([i,j]) = 0$, we get that $2\mathrm{d}_{\mathrm{TV}}(\mathbf{p}', \tilde{\mathbf{p}}) \geq q$. On one hand, if $q < p$ then at least $q - p$ weight must have been "removed" from $[i, 2j-i+1]$ to achieve monotonicity, and altogether $2\mathrm{d}_{\mathrm{TV}}(\mathbf{p}', M) \geq q + (p-q) = p$. On the other hand, if $q \geq p$ we directly get $2\mathrm{d}_{\mathrm{TV}}(\mathbf{p}', M) \geq q \geq p$. In both cases,

$$\mathrm{d}_{\mathrm{TV}}(\mathbf{p}', M) \geq p/2 \geq \varepsilon/2$$

and $\mathbf{p}'$ is $\varepsilon/2$-far from monotone. □

Lemma 5.6.4. *Let $\mathbf{p}$ be a monotone distribution and $\mathbf{p}' = \mathbf{p}^{(i,j)}$ as above. If $\mathbf{p}'([j+1, 2j-i+1]) < \varepsilon/2$, then $\mathbf{p}'$ is ε-close to monotone.*

Proof. We will constructively define a monotone distribution M which will be ε-close to $\mathbf{p}'$. Let $p \stackrel{\text{def}}{=} \mathbf{p}'([j+1, 2j-i+1]) < \varepsilon/2$. According to the missing data model, $\mathbf{p}'$ should be monotone on the intervals

$[1, i-1]$ and $[j+1, n]$. In particular, the probability weight of the last element of $[j+1, 2j-i+1]$ should be below the average weight of the interval, i.e. for all $k \geq 2j-i+1$ one has $\mathbf{p}'(k) \leq \mathbf{p}'(2j-i+1) < \frac{p}{j-i+1}$.

So, if we let the distribution M (that we are constructing) be uniform on the interval $[j+1, 2j-i+1]$ and have also total weight p there, monotonicity will not be violated at the right endpoint of the interval; and the ℓ_1 distance between $\mathbf{p}'$ and M in that interval will be at most $2p$. "Taking" another p probability weight from the very end of the domain and moving it to the interval $[i, j]$ (where it is then uniformly spread) to finish the construction of M adds at most another $2p$ to the ℓ_1 distance. Therefore, $2\mathrm{d}_{\mathrm{TV}}(\mathbf{p}', M) \leq 2p + 2p < 2\varepsilon$; and M is monotone as claimed. □

The sampling improver is described in Algorithm 35.

Algorithm 35 MISSING-DATA-IMPROVER

Require: $\varepsilon, \varepsilon_2 < \varepsilon$, $\delta \in (0,1)$ and $q \geq 1$, sample access to $\mathbf{p}'$.

Start ▷ PREPROCESSING
  Draw $m \stackrel{\text{def}}{=} \tilde{\Theta}\left(\frac{1}{\varepsilon_2^3} \log \frac{1}{\delta}\right)$ samples from $\mathbf{p}' = \mathbf{p}^{i,j}$.
  Run the algorithm of Lemma 5.6.5 on them to get an estimate (a, b) of the unknown (i, j) or the value close.
  Run the algorithm of Lemma 5.6.6 on them to get an estimate γ of $\mathbf{p}'([b+1, 2b-a+1])$, and values c, γ' such that $|\mathbf{p}'([c, n]) - \gamma'| \leq \varepsilon_2^{3/2}$.
End
Start ▷ GENERATING
  for i from 1 to q **do**
    Draw s_i from $\mathbf{p}'$.
    if the second step of PREPROCESSING returned close, or $\gamma < 5\varepsilon_2^{3/2}$ **then**
      return s_i ▷ The distribution is already ε_2-close to monotone; do not change it.
    end if
    if $s_i \in [c, n]$ **then** ▷ Move γ weight from the end to $[a, b]$
      With probability γ/γ', **return** a uniform sample from $[a, b]$
      Otherwise, **return** s_i
    else if $s_i \in [b+1, 2b-a+1]$ **then**
      return a uniform sample from $[b+1, 2b-a+1]$
    else
      return s_i ▷ Do not change the part of $\mathbf{p}'$ that need not be changed.
    end if
  end for
End

Implementing (i): detecting the gap

Lemma 5.6.5 (Lemma (i))**.** *There exists an algorithm that, on input $\alpha \in (0, 1/3)$ and $\delta \in (0,1)$, takes $\tilde{O}\left(\frac{1}{\alpha^6} \log \frac{1}{\delta}\right)$ samples from $\mathbf{p}' = \mathbf{p}^{i,j}$ and outputs either two elements $a, b \in [n]$ or close such that the following holds. With probability at least $1-\delta$,*

- *if it outputs elements a, b, then (a) $[i, j] \subseteq [a, b]$ and (b) $\mathbf{p}'([a, b]) \leq 3\alpha^2$;*
- *if it outputs close, then $\mathbf{p}'$ is α^2-close to monotone.*

Proof. Inspired by techniques from [63], we first partition the domain into $t = O\big(1/\alpha^2\big)$ intervals $I_1, \ldots, I_t$ of roughly equal weight as follows. By taking $O\big(\frac{1}{\alpha^6}\log\frac{1}{\delta}\big)$ samples, the DKW inequality ensures that with probability at least $1-\delta/2$ we obtain an approximation $\hat{\mathbf{p}}$ of $\mathbf{p}'$, close up to $\alpha^3/5$ in Kolmogorov distance. We hereafter assume this holds. For our partitioning to succeed, we first have to take care of the "big elements," which by assumption on $\mathbf{p}'$ (which originates from a monotone distribution) must all be at the beginning. In more detail, let

$$r \stackrel{\text{def}}{=} \max\left\{\, x \in [n] \,:\, \hat{\mathbf{p}}(x) \geq \frac{4\alpha^3}{5} \,\right\}$$

and $B \stackrel{\text{def}}{=} \{1, \ldots, r\}$ be the set of potentially big elements. Note that if $\mathbf{p}'(x) \geq \alpha^3$, then necessarily $x \in B$. This leaves us with two cases, depending on whether the "missing data interval" is amidst the big elements, or in the tail part of the support.

- If $[i,j] \subseteq B$: it is then straightforward to *exactly* find i, j, and output them as a, b. Indeed all elements $x \in B$ have, by monotonicity, either $\mathbf{p}'(x) \geq \mathbf{p}'(r) \geq \frac{3\alpha^3}{5}$, or $\mathbf{p}'(x) = 0$ (the latter if and only if $x \in [i,j]$). Thus, one can distinguish between $x \in [i,j]$ (for which $\hat{\mathbf{p}}(x) \leq \alpha^3/5$) and $x \notin [i,j]$ (in which case $\hat{\mathbf{p}}(x) \geq 2\alpha^3/5$).
- If $[i,j] \not\subseteq B$: then, as $r \notin [i,j]$ (since $\mathbf{p}'(r) > 0$), it must be the case that $[i,j] \subseteq \bar{B} = \{r+1, \ldots, n\}$. Moreover, every point $x \in \bar{B}$ is "light:" $\mathbf{p}'(x) < \alpha^3$ and $\hat{\mathbf{p}}(x) < \frac{4\alpha^3}{5}$. We iteratively define $I_1, \ldots, I_t \subseteq \bar{B}$, where $I_i = [r_i + 1, r_{i+1}]$: $r_1 \stackrel{\text{def}}{=} r+1$, $r_{t+1} \stackrel{\text{def}}{=} n$, and for $1 \leq i \leq t-1$

$$r_{i+1} \stackrel{\text{def}}{=} \min\left\{\, s > r_i \,:\, \hat{\mathbf{p}}([r_i+1, x]) \geq \alpha^2 \,\right\}.$$

  This guarantees that, for all $i \in [t]$, $\mathbf{p}'(I_i) \in [\alpha^2 - \frac{2\alpha^3}{5}, \alpha^2 + \frac{4\alpha^3}{5} + \frac{2\alpha^3}{5}] \subset [\alpha^2 - \frac{3\alpha^3}{2}, \alpha^2 + \frac{3\alpha^3}{2}]$. (And in turn that $t = O\big(1/\alpha^2\big)$ as claimed.) Observing that the definition of the missing data error model implies $\mathbf{p}'$ is 2-modal, we can now use the monotonicity tester of [63, Section 3.4]. This algorithm takes only $O\big(\frac{k}{\varepsilon^2}\log\frac{1}{\delta}\big)$ samples (crucially, no dependence on n) to distinguish with probability at least $1-\delta$ whether a k-modal distribution is monotone versus ε-far from it.

  We iteratively apply this tester with parameters $k = 2$, $\varepsilon = \alpha^2/4$ and $\delta' = O(\delta/t)$, to each of the at most t prefixes of the form $P_\ell \stackrel{\text{def}}{=} \cup_{i=1}^{\ell} I_i$; a union bound ensures that with probability at least $1-\delta/2$ all tests are correct. Conditioning on this, we are able to detect the first interval I_{ℓ^*} which either contains or falls after j (if no such interval is found, then the input distribution is already α^2-close to monotone and we output close). In more detail, suppose first no run of the tester rejects (so that close is outputted). Then, by Lemma 5.6.3, we must have $\mathbf{p}([j+1, 2j-i+1]) \leq 2 \cdot \alpha^2/4 = \alpha^2/2$, and Lemma 5.6.4 guarantees $\mathbf{p}'$ is then α^2-close to monotone.

  Suppose now that it rejects on some prefix P_{ℓ^*} (and accepted for all $\ell < \ell^*$). As $\mathbf{p}'$ is non-increasing on $[1, j]$, we must have $[i,j] \subset P_{\ell^*}$. Moreover, the tester will by Lemma 5.6.3 reject as soon as an interval $[j+1, s] \subseteq [j+1, 2j-i+1]$ of weight $\alpha^2/2$ is added to the current prefix. This implies, as each I_ℓ has weight at least $\alpha^2/2$, that $[i,j] \subseteq I_{\ell^*-1} \cup \ell^* = [a,b]$.

Finally, observe that the above can be performed with $O\big(\frac{1}{\alpha^2} \cdot \frac{1}{\alpha^4} \cdot \log t\big) = \tilde{O}\big(\frac{1}{\alpha^6} \log \frac{1}{\delta}\big)$ samples, as claimed (where the first $1/\alpha^2$ factor comes from doing rejection sampling to run the tester with domain P_ℓ only, which by construction is guaranteed to have weight $\Omega\big(1/\alpha^2\big)$). The overall probability of failure is at most $\delta/2 + \delta/2 = \delta$, as claimed.

□

Implementing (ii): estimating the missing weight Conditioning on the output a, b of Lemma 5.6.5 being correct, the next lemma explains how to get a good estimate of the total weight we should *put back* in $[a, b]$ in order to fix the deletion error.

Lemma 5.6.6. *Given $\mathbf{p}'$, α as above, $\delta \in (0,1)$ and a, b such that $[i, j] \subseteq [a, b]$ and $\mathbf{p}'([a, b]) \leq 3\alpha^2$, there exists an algorithm which takes $O\big(\frac{1}{\alpha^6} \log \frac{1}{\delta}\big)$ samples from $\mathbf{p}'$ and outputs values γ, γ' and c such that the following holds with probability at least $1 - \delta$:*

(i) $|\mathbf{p}'([b+1, 2b-a+1]) - \gamma| \leq \alpha^3$;

(ii) $|\mathbf{p}'([c, n]) - \gamma'| \leq \alpha^3$ *and* $\gamma' \geq \gamma$;

(iii) $\mathbf{p}'([c, n]) \geq \mathbf{p}'([b+1, 2b-a+1]) - 2\alpha^3$ *and* $\mathbf{p}'([c+1, n]) < \mathbf{p}'([b+1, 2b-a+1]) + 2\alpha^3$;

(iv) $\gamma \leq 2\varepsilon + 4\alpha^3$.

Proof. Again by invoking the DKW inequality, we can obtain (with probability at least $1-\delta$) an approximation $\hat{\mathbf{p}}$ of $\mathbf{p}'$, close up to $\alpha^3/2$ in Kolmogorov distance. This provides us with an estimate γ of $\mathbf{p}'([b+1, 2b-a+1])$ satisfying the first item (as, for any interval $[r, s]$, $\hat{\mathbf{p}}([r, s])$ is within an additive $\alpha^3/2$ of $\mathbf{p}'([r, s])$). Then, setting

$$c \stackrel{\text{def}}{=} \max\{\, x \in [n] \,:\, \hat{\mathbf{p}}([x, n]) \geq \gamma \,\}$$

and $\gamma' \stackrel{\text{def}}{=} \hat{\mathbf{p}}([c, n])$, items (ii) and (iii) follow. The last bound of (iv) derives from an argument identical as of Lemma 5.6.3 and the promise that $\mathbf{p}'$ is ε-close to monotone: indeed, one must then have $\mathbf{p}'([b+1, 2b-a+1]) \leq \mathbf{p}'([a, b]) + 2\varepsilon \leq 2\varepsilon + 3\alpha^2$, which with (i) concludes the argument. □

To finish the proof of Theorem 5.6.1, we apply the above lemmata with $\alpha \stackrel{\text{def}}{=} \Theta\big(\sqrt{\varepsilon_2}\big)$; and need to show that the algorithm generates samples from a distribution that is $\varepsilon_2 = O\big(\alpha^2\big)$-close to monotone. This is done by bounding the error encountered (due to approximation errors) in the following parts of the algorithm: when estimating the weight γ of an interval of equal length adjacent to the interval $[a, b]$, uniformizing its weight on that interval, and estimating the last γ-quantile of the distribution, in order to move the weight needed to fill the gap from there. If we could have perfect estimates of the gap ($[a, b] = [i, j]$), the missing weight γ and the point c such that $\mathbf{p}'([c, n]) = \gamma$, the corrected distribution would be monotone, as the probability mass function in both the gap and the next interval would be at the same "level" (that is, $\frac{\gamma}{b-a+1}$).

By choice of m, with probability at least $1 - \delta$ the two subroutines of the PREPROCESSING stage (from Lemma 5.6.5 and Lemma 5.6.6) behave as expected. We hereafter condition on this being the case.

For convenience, we write $I = [a, b]$, $J = [b + 1, 2b - a + 1]$ and $K = [c, n]$, where a, b, c and γ, γ' are the outcome of the preprocessing phase.

If the test in Line 9 passes If the preprocessing stage returned either close, or a value $\gamma < 5\varepsilon_2^{3/2} = 5\alpha^3$, then we claim that $\mathbf{p}'$ is already $O(\alpha^2)$-close to monotone. The first case is by correctness of Lemma 5.6.5; as for the second, observe that it implies $\mathbf{p}'(J) < 6\alpha^3$. Thus, "putting back" (from the tail of the support) weight at most $6\alpha^3$ in $[i, j]$ would be sufficient to correct the violation of monotonicity; which yields an $O(\alpha^3)$ upperbound on the distance of $\mathbf{p}'$ to monotone.

Otherwise This implies in particular that $\gamma \geq 5\alpha^3$, and thus $\mathbf{p}'(J) \geq 4\alpha^3$. By Lemma 5.6.6 (iii), it is then also the case that $\mathbf{p}'(K) \geq 2\alpha^3$. Then, denoting by $\tilde{\mathbf{p}}$ the corrected distribution, we have

$$\tilde{\mathbf{p}}(x) = \begin{cases} \mathbf{p}'(x) + \frac{\gamma}{\gamma'} \cdot \frac{\mathbf{p}'(K)}{|I|} & \text{if } x \in I \\ \frac{\mathbf{p}'(J)}{|J|} & \text{if } x \in J \\ \mathbf{p}'(x) \cdot (1 - \frac{\gamma}{\gamma'}) & \text{if } x \in K \\ \mathbf{p}'(x) & \text{otherwise.} \end{cases}$$

Distance to $\mathbf{p}'$ From the expression above, we get that

$$2\mathrm{d}_{\mathrm{TV}}(\tilde{\mathbf{p}}, \mathbf{p}') \leq \frac{\gamma}{\gamma'}\mathbf{p}'(K) + 2\mathbf{p}'(J) + \frac{\gamma}{\gamma'}\mathbf{p}'(K) = 2\left(\frac{\gamma}{\gamma'}\mathbf{p}'(K) + \mathbf{p}'(J)\right).$$

From Lemma 5.6.6, we also know that $\mathbf{p}'(J) \leq \gamma + \alpha^3$, $\mathbf{p}'(K) \leq \gamma' + \alpha^3$ and $\gamma/\gamma' \leq 1$, so that

$$\mathrm{d}_{\mathrm{TV}}(\tilde{\mathbf{p}}, \mathbf{p}') \leq \frac{\gamma}{\gamma'}(\gamma' + \alpha^3) + \gamma + \alpha^3 \leq 2(\gamma + \alpha^3) \leq 4\varepsilon + 10\alpha^3 = O(\varepsilon).$$

(Where, for the last inequality, we used Lemma 5.6.6 (iv); and finally the fact that $\varepsilon_2 \leq \varepsilon$).

Distance to monotone Consider the distributions M defined as

$$M(x) = \begin{cases} \mathbf{p}'(x) + \frac{\mathbf{p}'(J)}{|I|} & \text{if } x \in I \\ \frac{\mathbf{p}'(J)}{|J|} & \text{if } x \in J \\ \mathbf{p}'(x) \cdot \left(1 - \frac{\mathbf{p}'(J)}{\mathbf{p}'(K)}\right) & \text{if } x \in K \\ \mathbf{p}'(x) & \text{otherwise.} \end{cases}$$

We first claim that M is $O(\alpha^2)$-close to monotone. Indeed, M is monotone on $[a, n]$ by construction (and as $\mathbf{p}'$ was monotone on $[b, n]$). The only possible violations of monotonicity are on $[1, b]$, due to the approximation of (i, j) by (a, b) – that is, it is possible for the interval $[a, i]$ to now have too much weight, with $M(a - 1) < M(a)$. But as we have $\mathbf{p}'([a, b]) \leq 3\alpha^2$, the total extra weight of this "violating bump" is

$O(\alpha^2)$.

Moreover, the distance between M and $\tilde{\mathbf{p}}$ can be upperbounded by their difference on J and K:

$$2\mathrm{d}_{\mathrm{TV}}(\tilde{\mathbf{p}}, M) \leq 2\left|\mathbf{p}'(J) - \frac{\gamma}{\gamma'}\mathbf{p}'(K)\right| \leq 2\alpha^3 \frac{1+\frac{\alpha^3}{\mathbf{p}'(K)}}{1-\frac{\alpha^3}{\mathbf{p}'(K)}} \leq 6\alpha^3$$

where we used the fact that $\frac{\gamma}{\gamma'} \in \left[\frac{\mathbf{p}'(J)-\alpha^3}{\mathbf{p}'(K)+\alpha^3}, \frac{\mathbf{p}'(J)+\alpha^3}{\mathbf{p}'(K)-\alpha^3}\right]$, and that $\mathbf{p}'(K) \geq 2\alpha^3$. By the triangle inequality, $\tilde{\mathbf{p}}$ is then itself $O(\alpha^2)$-close to monotone. This concludes the proof of Theorem 5.6.1. □

5.7 Focusing on randomness scarcity

5.7.1 Correcting uniformity

In order to illustrate the challenges and main aspects of this section, we shall focus on what is arguably the most natural property of interest, "being uniform" (i.e. $\mathcal{P} = \{\mathbf{u}_n\}$). As a first observation, we note that when one is interested in correcting uniformity on an arbitrary domain Ω, allowing arbitrary amounts of additional randomness makes the task almost trivial: by using roughly $\log|\Omega|$ random bits per query, it is possible to interpolate arbitrarily between $\mathbf{p}$ and the uniform distribution. One can naturally ask whether the same can be achieved *while using no – or very little – additional randomness besides the draws from the sampling oracle itself.* As we show below, this is possible, at the price of a slightly worse query complexity. We hereafter focus once again on the case $\Omega = [n]$, and give constructions which achieve different trade-offs between the level of correction (of $\mathbf{p}$ to uniform), the fidelity to the original data (closeness to $\mathbf{p}$) and the sample complexity. We then show how to combine these constructions to achieve reasonable performance in terms of all the above parameters. In Section 5.7.1.1, we turn to the related problem of correcting uniformity on an (unknown) subgroup of the domain, and extend our results to this setting. Finally, we discuss the differences and relations with extractors in Section 5.7.2.

High-level ideas The first algorithm we describe (Theorem 5.7.1) is a sampling corrector based on a "von Neumann-type" approach: by seeing very crudely the distribution $\mathbf{p}$ as a distribution over two points (the first and second half of the support $[n]$), one can leverage the closeness of $\mathbf{p}$ to uniform to obtain with overwhelming probability a sequence of uniform random bits; and use them to generate a uniform element of $[n]$. The drawback of this approach lies in the number of samples required from $\mathbf{p}$: namely, $\tilde{\Theta}(\log n)$.

The second approach we consider relies on viewing $[n]$ as the Abelian group $\mathbb{Z}_n$, and leverages crucial properties of the convolution of distributions. Using a robust version of the fact that the uniform distribution is the absorbing element for this operation, we are able to argue that taking a *constant* number of samples from $\mathbf{p}$ and outputting their sum obeys a distribution $\tilde{\mathbf{p}}$ exponentially closer to uniform (Theorem 5.7.2). This result, however efficient in terms of getting closer to uniform, does not guarantee anything non-trivial about the distance $\tilde{\mathbf{p}}$ to the input distribution $\mathbf{p}$. More precisely, starting from $\mathbf{p}$ which is at a distance ε from uniform, it

is possible to end up with $\tilde{\mathbf{p}}$ at a distance ε' from uniform, but $\varepsilon + \Omega(\varepsilon')$ from $\mathbf{p}$ (see Claim 5.8.4 for more details). In other terms, this improver does get us closer to uniform, but somehow can *overshoot* in the process, getting too far from the input distribution.

The third improver we describe (in Theorem 5.7.3) yields slightly different parameters: it essentially enables one to get "midway" between $\mathbf{p}$ and the uniform distribution, and to sample from a distribution $\tilde{\mathbf{p}}$ (almost) $(\varepsilon/2)$-close to *both* the input and the uniform distributions. It achieves so by combining both previous ideas: using $\mathbf{p}$ to generate a (roughly) unbiased coin toss, and deciding based on the outcome whether to output a sample from $\mathbf{p}$ or from the improver of Theorem 5.7.2.

Finally, by "bootstrapping" the hybrid approach described above, one can provide sampling access to an improved $\hat{\mathbf{p}}$ both arbitrarily close to uniform *and* (almost) optimally close to the original distribution $\mathbf{p}$ (up to an additive $O(\varepsilon^3)$), as described in Theorem 5.7.4. Note that this is at a price of an extra $\log(1/\varepsilon_2)$ factor in the sample complexity, compared to Theorem 5.7.2: in a sense, the price of "staying faithful to the input data."

Theorem 5.7.1 (von Neumann Sampling Corrector)**.** *For any $\varepsilon < 0.49$ (and $\varepsilon_1 = \varepsilon$) as in the definition, there exists a sampling corrector for uniformity with query complexity $O(\log n(\log\log n + \log(1/\delta)))$ (where δ is the probability of failure per sample).*

Proof. Let $\mathbf{p}$ be a distribution over $[n]$ such that $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{u}) \leq \varepsilon < 1/2 - c$ for some absolute constant $c < 1/2$ (e.g., $c = 0.49$), and let S_0, S_1 denote respectively the sets $\{1, \ldots, n/2\}$ and $\{n/2 + 1, \ldots, n\}$. The high-level idea is to see a draw from $\mathbf{p}$ as a (biased) coin toss, depending on whether the sample lands in S_0 or S_1; by applying von Neumann's method, we then can retrieve a truly uniform bit at a time (with high probability). Repeating this $\log n$ times will yield a uniform draw from $[n]$. More precisely, it is immediate by definition of the total variation distance that $|\mathbf{p}(S_0) - \mathbf{p}(S_1)| \leq 2\varepsilon$, so in particular (setting $p \stackrel{\text{def}}{=} \mathbf{p}(S_0)$) we have access to a Bernoulli random variable with parameter $p \in \left[\frac{1}{2} - \varepsilon, \frac{1}{2} + \varepsilon\right]$.

To generate *one* uniform random bit (with probability of failure at most $\delta' = \delta/\log n$), it is sufficient to take in the worst case $m \stackrel{\text{def}}{=} \left\lceil (\log \frac{1}{1-c})^{-1} \log \frac{2}{\delta'} \right\rceil$ samples, and stop as soon as a sequence S_0S_1 or S_1S_0 is seen (giving respectively a bit 0 or 1). If it does not happen, then the corrector VN–IMPROVER$_n$ outputs fail; the probability of failure is therefore

$$\Pr[\,\text{VN–IMPROVER}_n \text{ outputs fail}\,] = p^m + (1-p)^m \leq 2 \cdot (1-c)^m \leq \delta' = \frac{\delta}{\log n}.$$

By a union bound over the $\log n$ bits to extract, VN–IMPROVER$_n$ indeed outputs a uniform random number $s \in [n]$ with probability at least $1 - \delta$, using at most $m \log n = O\left(\log n \log \frac{\log n}{\delta}\right)$ samples—and, in expectation, only $O((\log n)/p) = O(\log n)$. □

As previously mentioned, we hereafter work modulo n, equating $[n]$ to the Abelian group $(\mathbb{Z}_n, +)$. This convenient (and equivalent) view will allow us to use properties of convolutions of distributions over Abelian

groups,[13] in particular the fact that the uniform distribution on $\mathbb{Z}_n$ is (roughly speaking) an attractive fixed point for this operation. In particular, taking $\mathbf{p}$ to be the (unknown) distribution promised to be ε-close to uniform, Fact 5.8.3 guarantees that by drawing two independent samples $x, y \sim \mathbf{p}$ and computing $z = x + y \mod n$, the distribution of z is $(2\varepsilon^2)$-close to the uniform distribution on $[\![n]\!]$. This key observation is the basis for our next result:

Theorem 5.7.2 (Convolution Improver)**.** *For any $\varepsilon < \frac{1}{\sqrt{2}}$, ε_2 and $\varepsilon_1 = \varepsilon + \varepsilon_2$ as in the definition, there exists a sampling improver for uniformity with query complexity $O\left(\frac{\log \frac{1}{\varepsilon_2}}{\log \frac{1}{\varepsilon}}\right)$.*

Proof. Extending by induction the observation above to a sum of finitely many independent samples, we get that by drawing $k \stackrel{\text{def}}{=} \frac{\log \frac{1}{\varepsilon_2} - 1}{\log \frac{1}{\varepsilon} - 1}$ independent elements $s_1, \ldots, s_k$ from $\mathbf{p}$ and computing

$$s = \left(\sum_{\ell=1}^{k} s_\ell \mod n \right) + 1 \in [n]$$

the distribution $\tilde{\mathbf{p}}$ of s is $(\frac{1}{2}(2\varepsilon)^k)$-close to uniform; and by choice of k, $(\frac{1}{2}(2\varepsilon)^k) = \varepsilon_2$. As $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \tilde{\mathbf{p}}) \leq \mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{u}) + \mathrm{d}_{\mathrm{TV}}(\mathbf{u}, \tilde{\mathbf{p}}) \leq \varepsilon + \varepsilon_2$, the vacuous bound on the distance between $\mathbf{p}$ and $\tilde{\mathbf{p}}$ is as stated. □

This triggers a natural question: namely, can this "vacuous bound" be improved? That is, setting $\varepsilon \stackrel{\text{def}}{=} \mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{u})$ and $\mathbf{p}^{(k)} \stackrel{\text{def}}{=} \mathbf{p} * \cdots * \mathbf{p}$ (k-fold convolution), what can be said about $\mathrm{d}_{\mathrm{TV}}\left(\mathbf{p}, \mathbf{p}^{(k)}\right)$ as a function of ε and k? Trivially, the triangle inequality asserts that

$$\varepsilon - 2^{k-1}\varepsilon^k \leq \mathrm{d}_{\mathrm{TV}}\left(\mathbf{p}, \mathbf{p}^{(k)}\right) \leq \varepsilon + 2^{k-1}\varepsilon^k \ ;$$

but can the right-hand side be tightened further? For instance, one might hope to achieve ε. Unfortunately, this is not the case: even for $k = 2$, one cannot get better than $\varepsilon + \Omega(\varepsilon^2)$ as an upper bound. Indeed, one can show that for $\varepsilon \in (0, \frac{1}{2})$, there exists a distribution $\mathbf{p}$ on $\mathbb{Z}_n$ such that $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{u}) = \varepsilon$, yet $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{p} * \mathbf{p}) = \varepsilon + \frac{3}{4}\varepsilon^2 + O(\varepsilon^3)$ (see Claim 5.8.4 in the appendix).

Theorem 5.7.3 (Hybrid Improver)**.** *For any $\varepsilon \leq \frac{1}{2}$, $\varepsilon_1 = \frac{\varepsilon}{2} + 2\varepsilon^3 + \varepsilon'$ and $\varepsilon_2 = \frac{\varepsilon}{2} + \varepsilon'$, there exists a sampling improver for uniformity with query complexity $O\left(\frac{\log \frac{1}{\varepsilon'}}{\log \frac{1}{\varepsilon}}\right)$.*

Proof. Let $\mathbf{p}$ be a distribution over $[n]$ such that $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{u}) = \varepsilon$, and write d_0 (resp. d_1) for $\mathbf{p}(\{1, \ldots, n/2\})$ (resp. $\mathbf{p}(\{n/2 + 1, \ldots, n\})$). By definition, $|d_0 - d_1| \leq 2\varepsilon$. Define the Bernoulli random variable X by taking two independent samples s_1, s_2 from $\mathbf{p}$, and setting X to 0 if both land in the same half of the support (both in $\{1, \ldots, n/2\}$, or both in $\{n/2 + 1, \ldots, n\}$). It follows that $p_0 \stackrel{\text{def}}{=} \Pr[X = 0] = d_0^2 + d_1^2$ and $p_1 \stackrel{\text{def}}{=} \Pr[X = 1] = 2d_0d_1$, i.e. $0 \leq p_0 - p_1 = (d_1 - d_2)^2 \leq 4\varepsilon^2$. In other terms, $X \sim \mathrm{Bern}(p_0)$ with $\frac{1}{2} \leq p_0 \leq \frac{1}{2} + 2\varepsilon^2$.

[13] For more detail on this topic, the reader is referred to Section 5.8.

Consider now the distribution

$$\tilde{\mathbf{p}} \stackrel{\text{def}}{=} (1-p_0)\mathbf{p} + p_0\mathbf{p}^{(k)}$$

where $\mathbf{p}^{(k)} = \overbrace{\mathbf{p} * \ldots * \mathbf{p}}^{k \text{ times}}$ as in Theorem 5.7.2. Observe than getting a sample from $\tilde{\mathbf{p}}$ only requires at most $k+2$ queries[14] to the oracle for $\mathbf{p}$. Moreover,

$$\mathrm{d}_{\mathrm{TV}}(\tilde{\mathbf{p}}, \mathbf{u}) \leq (1-p_0)\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{u}) + p_0\mathrm{d}_{\mathrm{TV}}\left(\mathbf{p}^{(k)}, \mathbf{u}\right) \leq (1-p_0)\varepsilon + p_0 2^{k-1}\varepsilon^k \leq \frac{\varepsilon}{2} + \left(\frac{1}{4} + \varepsilon^2\right)(2\varepsilon)^k \leq \frac{\varepsilon}{2} + \frac{1}{2}(2\varepsilon)^k$$

while

$$\mathrm{d}_{\mathrm{TV}}(\tilde{\mathbf{p}}, \mathbf{p}) \leq p_0\mathrm{d}_{\mathrm{TV}}\left(\mathbf{p}^{(k)}, \mathbf{p}\right) \leq p_0\left(\varepsilon + 2^{k-1}\varepsilon^k\right) \leq \left(\frac{1}{2} + 2\varepsilon^2\right)\left(\varepsilon + 2^{k-1}\varepsilon^k\right) \leq \frac{\varepsilon}{2} + 2\varepsilon^3 + \frac{1}{2}(2\varepsilon)^k$$

(recalling for the rightmost step of each inequality that $\varepsilon \leq \frac{1}{2}$). Taking $k = 3$, one obtains, with a sample complexity at most 5, a distribution $\tilde{\mathbf{p}}$ satisfying

$$\mathrm{d}_{\mathrm{TV}}(\tilde{\mathbf{p}}, \mathbf{u}) \leq \frac{\varepsilon}{2} + 4\varepsilon^3, \qquad \mathrm{d}_{\mathrm{TV}}(\tilde{\mathbf{p}}, \mathbf{p}) \leq \frac{\varepsilon}{2} + 6\varepsilon^3 \ .$$

(Note that assuming $\varepsilon < 1/4$, one can get the more convenient – yet looser – bounds $\mathrm{d}_{\mathrm{TV}}(\tilde{\mathbf{p}}, \mathbf{u}) \leq \frac{21}{32}\varepsilon < \frac{2\varepsilon}{3}$, $\mathrm{d}_{\mathrm{TV}}(\tilde{\mathbf{p}}, \mathbf{p}) \leq \frac{97\varepsilon}{128} < \frac{4\varepsilon}{5}$.) □

Theorem 5.7.4 (Bootstrapping Improver)**.** *For any $\varepsilon \leq \frac{1}{2}$, $0 < \varepsilon_2 < \varepsilon$ and $\varepsilon_1 = \varepsilon - \varepsilon_2 + O\left(\varepsilon^3\right)$, there exists a sampling improver for uniformity with query complexity $O\left(\frac{\log^2 \frac{1}{\varepsilon_2}}{\log \frac{1}{\varepsilon}}\right)$.*

Proof. We show how to obtain such a guarantee – note that the constant 27 in the $O\left(\varepsilon^3\right)$ is not tight, and can be reduced at the price of a more cumbersome analysis. Let $\alpha > 0$ be a parameter (to be determined later) satisfying $\alpha < \varepsilon^2$, and k be the number of bootstrapping steps – i.e., the number of time one recursively apply the construction of Theorem 5.7.3 with α. We write $\mathbf{p}_j$ for the distribution obtained after the j^{th} recursive step, so that $\mathbf{p}_0 = \mathbf{p}$ and $\hat{\mathbf{p}} = \mathbf{p}_k$; and let u_j (resp. v_j) denote an upper bound on $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}_j, \mathbf{u})$ (resp. $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}_j, \mathbf{p})$). Note that by the guarantee of Theorem 5.7.3 and applying a triangle inequality for v_j, one gets the following recurrence relations for $(u_j)_{0 \leq j \leq k}$ and $(v_j)_{0 \leq j \leq k}$:

$$\begin{aligned} u_0 &= \varepsilon, & u_{j+1} &= \frac{1}{2}u_j + \alpha \\ v_0 &= 0, & v_{j+1} &= \left(\frac{1}{2}u_j + 2u_j^3 + \alpha\right) + v_j \end{aligned}$$

Solving this recurrence for u_k gives

$$u_k = \frac{\varepsilon}{2^k} + 2\left(1 - \frac{1}{2^k}\right)\alpha < \frac{\varepsilon}{2^k} + 2\alpha \tag{5.5}$$

[14] More precisely, 3 with probability $1 - p_0$, and $k+2$ with probability p_0, for an expected number $(k-1)p_0 + 3 \simeq k/2$.

while one gets an upper bound on v_k by writing

$$\begin{aligned} v_k = v_k - v_0 &= \sum_{j=0}^{k-1} (v_{j+1} - v_j) = k\alpha + \frac{1}{2}\sum_{j=0}^{k-1} u_j + 2\sum_{j=0}^{k-1} u_j^3 \\ &= 2k\alpha + \left(1 - \frac{1}{2^k}\right)\varepsilon - 2\left(1 - \frac{1}{2^k}\right)\alpha + 2\sum_{j=0}^{k-1} u_j^3 \\ &< \left(1 - \frac{1}{2^k}\right)\varepsilon + 2\underbrace{\left(k - 1 + \frac{1}{2^k}\right)\alpha}_{\leq k\alpha} + \left(3\varepsilon^3 + 16\varepsilon^2\alpha + 48\varepsilon\alpha^2 + 16k\alpha^3\right) \end{aligned}$$

where we used the expression (5.5) for u_j. Since $\alpha < \varepsilon^2 \leq \frac{1}{4}$, we can bound the rightmost terms as $16k\alpha^3 \leq k\alpha$, $48\varepsilon\alpha^2 < 48\varepsilon^5$ and $16\varepsilon^2\alpha < 16\varepsilon^4$, so that

$$v_k < \left(1 - \frac{1}{2^k}\right)\varepsilon + 3k\alpha + 3\varepsilon^3 + 16\varepsilon^4 + 48\varepsilon^5 < \left(1 - \frac{1}{2^k}\right)\varepsilon + 3k\alpha + 23\varepsilon^3 \tag{5.6}$$

It remains to choose k and α; to get $u_k \leq \varepsilon_2$, set $k \stackrel{\text{def}}{=} \left\lceil \log \frac{\varepsilon}{\varepsilon_2(1-\varepsilon^2)} \right\rceil \leq \log \frac{4\varepsilon}{3\varepsilon_2} + 1$ and $\alpha \stackrel{\text{def}}{=} \frac{1}{2}\varepsilon_2\varepsilon^2$, so that $\frac{\varepsilon}{2^k} \leq (1-\varepsilon^2)\varepsilon_2$ and $2\alpha \leq \varepsilon^2\varepsilon_2$. Plugging these values in (5.6),

$$v_k \underset{(\varepsilon_2 < \varepsilon)}{<} \left(1 - \frac{\varepsilon_2(1-\varepsilon^2)}{\varepsilon}\right)\varepsilon + \frac{3}{2}k\varepsilon_2\varepsilon^2 + 23\varepsilon^3 = \varepsilon - \varepsilon_2 + \frac{3}{2}k\varepsilon_2\varepsilon^2 + 24\varepsilon^3 < \varepsilon - \varepsilon_2 + 27\varepsilon^3$$

where the last inequality comes from the fact that $\frac{3}{2}k\frac{\varepsilon_2}{\varepsilon} \leq \frac{3}{2}\log\frac{8\varepsilon}{3\varepsilon_2} \cdot \frac{\varepsilon_2}{\varepsilon} \leq 3$. Therefore, we have $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}_k, \mathbf{u}) \leq \varepsilon_2$, $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}_k, \mathbf{p}) \leq \varepsilon - \varepsilon_2 + 27\varepsilon^3$ as claimed. We turn to the number m of queries made along those k steps; from Theorem 5.7.3, this is at most

$$m \leq \sum_{j=0}^{k-1} \left\lceil \frac{\log\frac{1}{\alpha} - 1}{\log\frac{1}{u_j} - 1} \right\rceil \leq k \cdot \left\lceil \frac{\log\frac{1}{\alpha} - 1}{\log\frac{1}{\varepsilon} - 1} \right\rceil = O\left(\frac{\log^2\frac{1}{\varepsilon_2}}{\log\frac{1}{\varepsilon}}\right)$$

which concludes the proof. □

Note that in all four cases, as our improvers do not use any randomness of their own, they always output according to the same improved distribution: that is, after fixing the parameters $\varepsilon, \varepsilon_2$ and the unknown distribution $\mathbf{p}$, then $\hat{\mathbf{p}}$ is uniquely determined, even across independent calls to the improver.

5.7.1.1 Correcting uniformity on a subgroup

Outline It is easy to observe that all the results above still hold when replacing $\mathbb{Z}_n$ by any finite Abelian group G. Thus, a natural question to turn to is whether one can generalize these results to the case where the unknown distribution is close to the uniform distribution on an arbitrary, unknown, *subgroup* H of the domain G.

To do so, a first observation is that if H were known, and if furthermore a constant (expected) fraction of the samples were to fall within it, then one could directly apply our previous results by conditioning samples on

being in H, using rejection sampling. The results of this section show how to achieve this "identification" of the subgroup with only a $\log(1/\varepsilon)$ overhead in the sample complexity. At a high-level, the idea is to take a few samples, and argue that their greatest common divisor will (with high probability) be a generator of the subgroup.

Details Let G be a finite cyclic Abelian group of order n, and $H \subseteq G$ a subgroup of order m. We denote by $\mathbf{u}_H$ the uniform distribution on this subgroup. Moreover, for a distribution $\mathbf{p}$ over G, we write $\mathbf{p}_H$ for the conditional distribution it induces on H, that is

$$\forall x \in G, \quad \mathbf{p}_H(x) = \frac{\mathbf{p}(x)}{\mathbf{p}(H)} \mathbb{1}_H(x)$$

which is defined as long as $\mathbf{p}$ puts non-zero weight on H. The following lemma shows that if $\mathbf{p}$ is close to $\mathbf{u}_H$, then so is $\mathbf{p}_H$:

Lemma 5.7.5. *Assume* $\mathrm{d_{TV}}(\mathbf{p}, \mathbf{u}_H) < 1$. *Then* $\mathrm{d_{TV}}(\mathbf{p}_H, \mathbf{u}_H) \leq \mathrm{d_{TV}}(\mathbf{p}, \mathbf{u}_H)$.

Proof. First, observe that the assumption implies $\mathbf{p}_H$ is well-defined: indeed, as $\mathrm{d_{TV}}(\mathbf{p}, \mathbf{u}_H) = \sup_{S \subseteq G} (\mathbf{u}_H(S) - \mathbf{p}(S))$, taking $S = H$ yields $1 > \mathrm{d_{TV}}(\mathbf{p}, \mathbf{u}_H) \geq \mathbf{u}_H(H) - \mathbf{p}(H) = 1 - \mathbf{p}(H)$, and thus $\mathbf{p}(H) > 0$.

Rewriting the definition of $\mathrm{d_{TV}}(\mathbf{p}_H, \mathbf{u}_H)$, one gets $\mathrm{d_{TV}}(\mathbf{p}, \mathbf{u}_H) = \frac{1}{2}\left(\sum_{x \in H} \left|\mathbf{p}(x) - \frac{1}{|H|}\right| + \sum_{x \notin H} \mathbf{p}(x)\right)$; so that

$$\begin{aligned}
2\mathrm{d_{TV}}(\mathbf{p}_H, \mathbf{u}_H) = \sum_{x \in H} \left|\mathbf{p}_H(x) - \frac{1}{|H|}\right| &\leq \sum_{x \in H} |\mathbf{p}_H(x) - \mathbf{p}(x)| + \sum_{x \in H} \left|\mathbf{p}(x) - \frac{1}{|H|}\right| \\
&= \sum_{x \in H} |\mathbf{p}_H(x) - \mathbf{p}(x)| + \left(2\mathrm{d_{TV}}(\mathbf{p}, \mathbf{u}_H) - \sum_{x \notin H} \mathbf{p}(x)\right) \\
&= \sum_{x \in H} \mathbf{p}(x) \left|\frac{1}{\mathbf{p}(H)} - 1\right| + 2\mathrm{d_{TV}}(\mathbf{p}, \mathbf{u}_H) - (1 - \mathbf{p}(H)) \\
&= \mathbf{p}(H) \left|\frac{1}{\mathbf{p}(H)} - 1\right| + 2\mathrm{d_{TV}}(\mathbf{p}, \mathbf{u}_H) - (1 - \mathbf{p}(H)) \\
&= |1 - \mathbf{p}(H)| + 2\mathrm{d_{TV}}(\mathbf{p}, \mathbf{u}_H) - (1 - \mathbf{p}(H)) \\
&= 2\mathrm{d_{TV}}(\mathbf{p}, \mathbf{u}_H).
\end{aligned}$$

□

Let $\mathbf{p}$ be a distribution on G promised to be ε-close to the uniform distribution $\mathbf{u}_H$ on some unknown subgroup H, for $\varepsilon < \frac{1}{2} - c$. For the sake of presentation, we hereafter without loss of generality identify G to $\mathbb{Z}_n$. Let h be the generator of H with smallest absolute values (when seen as an integer), so that $H = \{0, h, 2h, 3h, \ldots, (m-1)h\}$.

Observe that $\mathbf{p}(H) > 1 - 2\varepsilon$, as $2\mathrm{d_{TV}}(\mathbf{p}, \mathbf{u}_H) = \sum_{x \in H} \left|\mathbf{p}(x) - \frac{1}{|H|}\right| + \mathbf{p}(H^c)$; therefore, *if* H were known one could efficiently simulate sample access to $\mathbf{p}_H$ via rejection sampling, with only a constant factor overhead

(in expectation) per sample. It would then become possible, as hinted in the foregoing discussion, to correct uniformity on $\mathbf{p}_H$ (which is ε-close to $\mathbf{u}_H$ by Lemma 5.7.5) via one of the previous algorithms for Abelian groups. The question remains to show how to find H; or, equivalently, h.

Algorithm 36 Algorithm FIND-GENERATOR-SUBGROUP

Require: $\varepsilon \in (0, \frac{1}{2} - c]$, SAMP_D with $\mathbf{p}$ ε-close to uniform on some subgroup $H \subseteq \mathbb{Z}_n$
Ensure: Outputs a generator $\hat{h}$ of H with probability $1 - \tilde{O}(\varepsilon)$
  Draw k independent samples $s_1, \ldots, s_k$ from $\mathbf{p}$, for $k \stackrel{\text{def}}{=} O\big(\log \frac{1}{\varepsilon}\big)$
  Compute $\hat{h} = \gcd(s_1, \ldots, s_k)$
  return $\hat{h}$

Lemma 5.7.6. *Let G, H be as before. There exists an algorithm (Algorithm 36) which, given $\varepsilon < 0.49$ as well as sample access to some distribution $\mathbf{p}$ over G, makes $O\big(\log \frac{1}{\varepsilon}\big)$ calls to the oracle and returns an element of G. Further, if $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{u}_H) \leq \varepsilon$, then with probability at least $1 - \tilde{O}(\varepsilon)$ its output is a generator of H.*

Proof. In order to argue correctness of the algorithm, we will need the following well-known facts:

Fact 5.7.7. *Fix any $k \geq 1$, and let $p_{n,k}$ be the probability that k independent numbers drawn uniformly at random from $[n]$ be relatively prime. Then $p_{n,k} \xrightarrow[n\to\infty]{} \frac{1}{\zeta(k)}$ (where ζ is the Riemann zeta function).*

Fact 5.7.8. *One has $\zeta(x) \underset{x\to\infty}{=} 1 + \frac{1}{2^x} + o\big(\frac{1}{2^x}\big)$; and in particular $\frac{1}{\zeta(k)} \underset{k\to\infty}{=} 1 - \frac{1}{2^k} + o\big(\frac{1}{2^k}\big)$.*

With this in hand, let $k \stackrel{\text{def}}{=} O\big(\log \frac{1}{\varepsilon}\big)$, chosen so that $\varepsilon k = \Theta\big(\frac{1}{2^k}\big) = \Theta\Big(\frac{\varepsilon}{\log \frac{1}{\varepsilon}}\Big)$. We break the analysis of our subgroup-finding algorithm in two cases:

Case 1: $|H| = \Theta(1)$ This is the easy case: if H only contains constantly many elements (m is a constant of n and ε), then after taking k samples $s_1, \ldots, s_k \sim \mathbf{p}$, we have

- $s_1, \ldots, s_k \in H$ (event E_1) with probability at least $(1-\varepsilon)^k = 1 - O(k\varepsilon)$;
- the probability that there exists an element of H not hit by any of the s_i's is at most, by a union bound,

$$\sum_{x \in H} (1 - \mathbf{p}(x))^k \leq m \left(1 - \frac{1}{m} + \varepsilon\right)^k = 2^{-\Omega(k)}$$

  for ε sufficiently small ($\varepsilon \ll \frac{1}{m}$). Let E_2 be the event each element of H appears amongst the samples.

Overall, with probability $1 - O(k\varepsilon)$ (conditioning on E_1 and E_2), our set of samples is exactly H, and $\gcd(s_1, \ldots, s_k) = \gcd(H) = h$.

Case 2: $|H| = \omega(1)$ This amounts to saying that $h = o(n)$. In this case, taking again k samples $s_1, \ldots, s_k \sim \mathbf{p}$ and denoting by $\hat{h}$ their greatest common divisor:

- $s_1, \ldots, s_k \in H$ (event E_1) with probability at least $(1-\varepsilon)^k = 1 - O(k\varepsilon)$ as before;
- conditioned on E_1, note that if the s_i's were *uniformly* distributed in H, then the probability that $\hat{h} = h$

would be exactly $p_{\frac{n}{h},k}$ – as $\gcd(ha, \dots, hb) = h$ if and only if $\gcd(a, \dots, b) = 1$, i.e. if $a, \dots, b$ are relatively prime. In this ideal scenario, therefore, we would have

$$\Pr[\gcd(s_1, \dots, s_k) = h \mid E_1] = p_{\frac{n}{h},k} \xrightarrow[n\to\infty]{} \frac{1}{\zeta(k)} = 1 - O\left(\frac{1}{2^k}\right)$$

by Fact 5.7.7 and our assumption $h = o(n)$.

To adapt this result to our case – where $s_1, \dots, s_k \sim \mathbf{p}_H$ (as we conditioned on E_1), it is sufficient to observe that by the Data Processing Inequality for total variation distance,

$$\left| \Pr_{s_1,\dots,s_k\sim\mathbf{p}_H}[\gcd(s_1, \dots, s_k) = h] - \Pr_{s_1,\dots,s_k\sim\mathbf{u}_H}[\gcd(s_1, \dots, s_k) = h] \right| \le \mathrm{d}_{\mathrm{TV}}\left(\mathbf{p}_H^{\otimes k}, \mathbf{u}_H^{\otimes k}\right) \le k\varepsilon$$

so that in our case

$$\Pr[\gcd(s_1, \dots, s_k) = h \mid E_1] \ge p_{\frac{n}{h},k} - k\varepsilon \xrightarrow[n\to\infty]{} \frac{1}{\zeta(k)} - k\varepsilon = 1 - O\left(\frac{1}{2^k}\right) = 1 - O\left(\frac{\varepsilon}{\log\frac{1}{\varepsilon}}\right) \tag{5.7}$$

In either case, with probability at least $1 - \tilde{O}(\varepsilon)$, we find a generator h of H, acting as a (succinct) representation of H which allows us to perform rejection sampling. □

This directly implies the theorem below: any sampling improver for uniformity on a group directly yields an improver for uniformity on an unknown *subgroup*, with essentially the same complexity.

Theorem 5.7.9. *Suppose we have an $(\varepsilon, \varepsilon_1, \varepsilon_2)$-sampling improver for uniformity over Abelian finite cyclic groups, with query complexity $q(\varepsilon, \varepsilon_1, \varepsilon_2)$. Then there exists an $(\varepsilon, \varepsilon_1, \varepsilon_2)$-sampling improver for uniformity on subgroups, with query complexity*

$$O\left(\log\frac{1}{\varepsilon} + q(\varepsilon, \varepsilon_1, \varepsilon_2)\log q(\varepsilon, \varepsilon_1, \varepsilon_2)\right)$$

Proof. Proof is straightforward (rejection sampling over the subgroup, once identified: constant probability of hitting it, so by trying at most $O(\log q)$ draws per samples before outputting fail, one can provide a sample from $\mathbf{p}_H$ to the original algorithm with probability $1 - 1/10q$, for each of the (at most) q queries). □

5.7.2 Comparison with randomness extractors

In the randomness extractor model, one is provided with a source of imperfect random bits (and sometimes an additional source of completely random bits), and the goal is to output as many random bits as possible that are close to uniformly distributed. In the distribution corrector model, one is provided with a distribution that is *close to having* a property $\mathcal{P}$, and the goal is to have the ability to generate a *similar* distribution that *has* property $\mathcal{P}$.

One could therefore view extractors as sampling improvers for the property of uniformity of distributions

(i.e. $\mathcal{P} = \{\mathbf{u}_n\}$): indeed, both sampling correctors and extractors attempt to minimize the use of extra randomness. However, there are significant differences between the two settings. A first difference is that randomness extractors assume a lower bound on the min-entropy[15] of the input distribution, whereas sampling improvers assume the distribution to be ε-close to uniform in total variation distance. Note that the two assumptions are not comparable.[16]Secondly, in both the extractor and sampling improver models, since the entropy of the output distribution should be larger, one would either need more random bits from the weak random source or additional uniform random bits. Our sampling improvers do not use any extra random bits, which is also the case in deterministic extractors, but not in other extractor constructions. However, unlike the extractor model, in the sampling improver model, there is no bound on the number of independent samples one can take from the original distribution. Tight bounds and impossibility results are known for both general and deterministic extractors [166, 145], in particular in terms of the amount of additional randomness required. Because of the aforementioned differences in both the assumptions on and access to the input distribution, these lower bounds do not apply to our setting – which explains why our sampling improvers avoid this need for extra random bits.

5.7.3 Monotone distributions and randomness scarcity

In this section, we describe how to utilize a (close-to-monotone) input distribution to obtain the uniform random samples some of our previous correctors and improvers need. This is at the price of a $\tilde{O}(\log n)$-sample overhead per draw, and follows the same general approach as in Theorem 5.7.1. We observe that even if this *seems* to defeat the goal (as, with this many samples, one could even *learn* the distribution, as stated in Lemma 5.5.1), this is not actually the case: indeed, the procedure below is meant as a subroutine for these very same correctors, emancipating them from the need for truly additional randomness – which they would require otherwise, e.g. to generate samples from the corrected or learnt distribution.

Lemma 5.7.10 (Randomness from (almost) monotone)**.** *There exists a procedure which, which, given $\varepsilon \in [0, 1/3)$ and $\delta > 0$, as well as sample access to a distribution $\mathbf{p}$ guaranteed to be ε-close to monotone, either returns*

- *"point mass," if $\mathbf{p}$ is ε-close to the point distribution[17] on the first element;*
- *or a uniform random sample from $[n]$;*

with probability of failure at most δ. The procedure makes $O\left(\frac{1}{\varepsilon^2}\log\frac{1}{\delta}\right)$ samples from $\mathbf{p}$ in the first case, and $O\left(\frac{\log n}{\varepsilon}\log\frac{\log n}{\delta}\right)$ in the second.

[15]The min-entropy of a distribution $\mathbf{p}$ is defined as $H_\infty(\mathbf{p}) = \log\frac{1}{\max_i \mathbf{p}(i)}$.

[16]For example, the min-entropy of a distribution which is ε-close to uniform can range from $\log(1/\varepsilon)$ to $\Omega(\log n)$. Conversely, the distance to uniformity of a distribution which has high min-entropy can also vary significantly: there exist distributions with min-entropy $\Omega(\log n)$ but which are respectively $\Omega(1)$-far from and $O(1/n)$-close to uniform.

[17]The Dirac distribution δ_1 defined by $\delta_1(1) = 1$, which can be trivially sampled from without the use for any randomness by always outputting 1.

Proof. By taking $O\big(\log(1/\delta)/\varepsilon^2\big)$ samples, the algorithm starts by approximating by $\hat{F}$ the cdf F of the distribution up to an additive $\frac{\varepsilon}{4}$ in ℓ_∞. Then, defining

$$m \stackrel{\text{def}}{=} \min\left\{\, i \in [n] :\ \hat{F}(i) \geq 1 - \frac{\varepsilon}{2} \,\right\}$$

so that $F(m) \geq 1 - \frac{3\varepsilon}{4}$. According to the value of m, we consider two cases:

- If $m = 1$, then $\mathbf{p}$ is ε-close to the (monotone) distribution δ_1 which has all weight on the first element; this means we have effectively *learnt* the distribution, and can thereafter consider, for all purposes, δ_1 in lieu of $\mathbf{p}$.
- If $m > 1$, then $\mathbf{p}(1) < 1 - \frac{\varepsilon}{4}$ and we can partition the domain in two sets $S_0 \stackrel{\text{def}}{=} \{1, \ldots, k\}$ and $S_1 \stackrel{\text{def}}{=} \{k, \ldots, n\}$, by setting

$$k \stackrel{\text{def}}{=} \min\left\{\, i \in [n] :\ \hat{F}(i) < 1 - \frac{\varepsilon}{2} \,\right\}$$

By our previous check we know that this quantity is well-defined. Further, this implies that $\mathbf{p}(S_0) < 1 - \frac{\varepsilon}{4}$ and $\mathbf{p}(S_1) > \frac{\varepsilon}{4}$ (the actual values being known up to $\pm\frac{\varepsilon}{4}$). $\mathbf{p}$ being ε-close to monotone, it also must be the case that

$$\mathbf{p}(S_0) \geq \frac{k}{k+1}\left(1 - \frac{3\varepsilon}{4}\right) - 2\varepsilon \geq \frac{1}{2}\left(1 - \frac{3\varepsilon}{4}\right) - 2\varepsilon \geq \frac{1}{2} - \frac{19\varepsilon}{8}$$

since $\hat{F}(k+1) \geq 1 - \frac{\varepsilon}{2}$ implies $\mathbf{p}(\{1, \ldots, k\}) + \mathbf{p}(k) = \mathbf{p}(\{1, \ldots, k+1\}) \geq 1 - \frac{3\varepsilon}{4}$. By setting $p \stackrel{\text{def}}{=} \mathbf{p}(S_0)$, this means we have access to a Bernoulli random variable with parameter $p \in \left[\frac{1}{2} - 3\varepsilon, 1 - \frac{\varepsilon}{4}\right]$. As in the proof of Theorem 5.7.1 (the constant c being replaced by $\min\big(\frac{\varepsilon}{4}, \frac{1}{2} - 3\varepsilon\big)$), one can then leverage this to output with probability at least $1 - \delta$ a uniform random number $s \in [n]$ using $O\Big(\frac{\log n}{\varepsilon} \log \frac{\log n}{\delta}\Big)$ samples—and $O\Big(\frac{\log n}{\varepsilon}\Big)$ in expectation. □

5.8 On convolutions of distributions over an Abelian finite cyclic group

Definition 5.8.1. For any two probability distributions $\mathbf{p}_1, \mathbf{p}_2$ over a finite group G (not necessarily Abelian), the *convolution* of $\mathbf{p}_1$ and $\mathbf{p}_2$, denoted $\mathbf{p}_1 * \mathbf{p}_2$, is the distribution on G defined by

$$\mathbf{p}_1 * \mathbf{p}_2(x) = \sum_{g \in G} \mathbf{p}_1(xg^{-1})\mathbf{p}_2(g)$$

In particular, if G is Abelian, $\mathbf{p}_1 * \mathbf{p}_2 = \mathbf{p}_2 * \mathbf{p}_1$.

Fact 5.8.2. *The convolution satisfies the following properties:*

(i) it is associative:

$$\forall \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \quad \mathbf{p}_1 * (\mathbf{p}_2 * \mathbf{p}_3) = (\mathbf{p}_1 * \mathbf{p}_2) * \mathbf{p}_3 = \mathbf{p}_1 * \mathbf{p}_2 * \mathbf{p}_3 \tag{5.8}$$

(ii) it has a (unique) absorbing element, the uniform distribution $\mathbf{u}(G)$*:*

$$\forall \mathbf{p}, \quad \mathbf{u}(G) * \mathbf{p} = \mathbf{u}(G) \tag{5.9}$$

(iii) it can only decrease the total variation:

$$\forall \mathbf{p}_1, \mathbf{p}_2, \mathbf{p}_3, \quad \mathrm{d}_{\mathrm{TV}}(\mathbf{p}_1 * \mathbf{p}_2, \mathbf{p}_1 * \mathbf{p}_3) \leq \mathrm{d}_{\mathrm{TV}}(\mathbf{p}_2, \mathbf{p}_3) \tag{5.10}$$

For more on convolutions of distributions over finite groups, see for instance [78] or [24].

Fact 5.8.3 ([129])**.** *Let* G *be a finite Abelian group, and* $\mathbf{p}_1$*,* $\mathbf{p}_2$ *two probability distributions over* G*. Then, the convolution of* $\mathbf{p}_1$ *and* $\mathbf{p}_2$ *satisfies*

$$\mathrm{d}_{\mathrm{TV}}(\mathbf{u}(G), \mathbf{p}_1 * \mathbf{p}_2) \leq 2\mathrm{d}_{\mathrm{TV}}(\mathbf{u}(G), \mathbf{p}_1)\mathrm{d}_{\mathrm{TV}}(\mathbf{u}(G), \mathbf{p}_2) \tag{5.11}$$

where $\mathbf{u}(G)$ *denotes the uniform distribution on* G*. Furthermore, this bound is tight.*

Claim 5.8.4. *For* $\varepsilon \in (0, \frac{1}{2})$*, there exists a distribution* $\mathbf{p}$ *on* $\mathbb{Z}_n$ *such that* $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{u}) = \varepsilon$*, yet* $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{p} * \mathbf{p}) = \varepsilon + \frac{3}{4}\varepsilon^2 + O(\varepsilon^3) > \varepsilon$*.*

Proof. Inspired by—and following—a question on MathOverflow ([128]). Setting $\delta = 1 - \varepsilon > 1/2$, and taking $\mathbf{p}_A$ to be uniform on a subset A of $\mathbb{Z}_n$ with $|A| = \delta n$, one gets that $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}_A, \mathbf{u}) = \varepsilon$, and yet

$$\mathrm{d}_{\mathrm{TV}}(\mathbf{p}_A, \mathbf{p}_A * \mathbf{p}_A) = \frac{1}{2}\|\mathbf{p}_A - \mathbf{p}_A * \mathbf{p}_A\|_1 = \frac{1}{2}\sum_{g \in G}\left|\frac{\mathbb{1}_A(g)}{|A|} - \frac{r(g)}{|A|^2}\right| = 1 - \frac{1}{|A|^2}\sum_{a \in A} r(a)$$

where $r(g)$ is the number of representations of g as a sum of two elements of A (as one can show that $\mathbf{p}_A * \mathbf{p}_A(g) = |A|^{-2} r(g)$). Fix A to be the interval of length δn centered around $n/2$, that is $A = \{\ell, \ldots, L\}$ with

$$\ell \stackrel{\text{def}}{=} \frac{1-\delta}{2}n, \qquad L \stackrel{\text{def}}{=} \frac{1+\delta}{2}n$$

Computing the quantity $\sum_{a \in A} r(a)$ amounts to counting the number of pairs $(a, b) \in A \times A$ whose sum (modulo n) lies in A. For convenience, define $k = (1-\delta)n$ and $K = \delta n$:

- for $k \leq a \leq K$, $a + b$ ranges from $\ell + a \leq L$ to $L + a \geq \ell + n$, so that modulo n exactly $|A| - |A^c| = (2\delta - 1)n$ elements of A are reached (each of them exactly once);
- for $\ell \leq a < k$, $a + b$ ranges from $\ell + a < L$ to $L + a < \ell + n$, so that the elements of A not obtained are those in the interval $\{\ell, \ell + a - 1\}$ – there are a of them – and again the others are obtained exactly once;
- for $K < a \leq L$, $a + b$ ranges from $L < \ell + a \leq n$ to $L + a \leq K + n$, so that the elements of A not obtained are those in the interval $\{L + a - n + 1, L\}$ – there are $n - a$ of them – and as before the

others are hit exactly once.

It follows that

$$\begin{aligned}
\sum_{a\in A} r(a) &= (K-k+1)(2\delta-1)n + \sum_{a=\ell}^{k-1}(\delta n - a) + \sum_{a=K+1}^{L}(\delta n - (n-a)) \\
&= (2\delta-1)(2\delta-1)n^2 + \sum_{a=\ell}^{k-1}(\delta n - a) + \sum_{a=K+1}^{L}(a-(1-\delta)n) \qquad \text{(the 2 sums are equal)} \\
&= (2\delta-1)^2 n^2 + 2\cdot\frac{n^2}{8}(7\delta-3)(1-\delta) + O(n) \\
&= \frac{1}{4}\left(4(4\delta^2-4\delta+1) + (7\delta-3)(1-\delta)\right)n^2 + O(n) = \frac{1}{4}\left(9\delta^2-6\delta+1\right)n^2 + O(n) \\
&= \left(1-3\varepsilon+\frac{9}{4}\varepsilon^2\right)n^2 + O(n)
\end{aligned}$$

and thus

$$\mathrm{d}_{\mathrm{TV}}(\mathbf{p}_A, \mathbf{p}_A * \mathbf{p}_A) = 1 - \frac{1}{|A|^2}\sum_{a\in A} r(a) = 1 - \frac{1-3\varepsilon+\frac{9}{4}\varepsilon^2}{(1-\varepsilon)^2} + O\left(\frac{1}{n}\right) = \varepsilon + \frac{3}{4}\varepsilon^2 + O\left(\varepsilon^3\right)$$

(as $\varepsilon = \omega(1/\sqrt[3]{n})$). □

Conclusion

For the Snark *was* a Boojum, you see.

Lewis Carroll, *The Hunting of the Snark*

In this dissertation, we have pursued a three-pronged approach towards a better understanding of distribution testing and what lies beyond. First, placing ourselves in the standard setting of distribution testing, we advocated for a paradigm shift: by, instead of tackling each new distribution testing problem in an *ad hoc* way, developing *general* tools and algorithms that can be used for any of these problems. We contributed to that shift by providing two widely applicable algorithmic approaches – one based on shape constraints, the other on properties of the Fourier transform, as well as two lower bound frameworks – one based on reductions between distribution testing questions, and the other from communication complexity.

Second, we departed from this standard "sample-only" setting, which – albeit the most natural and conservative – fails to capture many situations of interest, and can be for those significantly *too* conservative. We introduced two incomparable generalizations of this setting, respectively the *conditional* and *extended* oracle access models; and explored the power and limitations of testing algorithms in these new models, for a wide range of fundamental questions.

Finally, we went beyond distribution testing and described a new algorithmic primitive, that of a sampling *corrector*. We studied some of the applications of this new notion, and investigated its relation to the fields of distribution testing and learning. Of an exploratory nature, our work opens the door to an entirely new research direction, which we believe will lead to new insights and applications in learning theory.

Open questions and future work

In spite of the length of this dissertation and our best efforts, the results we obtained here leave many promising questions unanswered, of which we list a few below.

Instance-optimality, communication complexity, and interpolation theory In Section 3.2, we instantiated our communication complexity methodology to obtain an "instance-optimal" lower bound on the problem of *identity testing*. This lower bound allowed us to establish a connection between distribution testing and the seemingly unrelated field of interpolation theory from functional analysis, leading to new insights on a result of Valiant and Valiant [169]. Two questions immediately come to mind:

Question 5.8.5. *Can one leverage this methodology to obtain lower bounds on* closeness testing[18] *via a reduction from communication complexity?*

Question 5.8.6. *What other connections between distribution testing and interpolation theory can be made? As a concrete example, is there a analogous characterization of the sample complexity of* tolerant *identity testing in terms of the K-functional between some ℓ_p and ℓ_q spaces?*

As an aside, we remark that defining what "instance-optimality" should mean in the case of identity testing (or even in the case of testing a given property $\mathcal{P}$ exhibiting some obvious structure) is rather intuitive: namely, the parameter should now be a functional of the (known) reference distribution $\mathbf{p}$, instead of the (also known) domain size n. Defining instance-optimality for *closeness* testing, however, is less straightforward: indeed, there is no longer any reference distribution, as both "players" $\mathbf{p}, \mathbf{q}$ are unknown. This leads to our next question:

Question 5.8.7. *How to define "instance-optimality" for closeness testing of two distributions in a meaningful and robust way? Does such a notion inherently require* adaptivity *from the testing algorithms?*

(We note that Diakonikolas and Kane do study this question in [80]; it is not entirely obvious to us, however, that the notion of instance-optimality they rely on is the "right" one.)

Coding Theory The results of Section 3.2 crucially hinged on the use of "good" codes, with quite specific requirements – which conveniently happened to exist. In a recent work with Tom Gur [47] on property (not distribution) testing, we established an "adaptivity hierarchy theorem" for property testing: there too, several crucial arguments were contingent on the existence of error-correcting codes satisfying a plethora of unlikely conditions. There too, such codes turned out to be waiting for us in the literature, and the proofs went through.

Question 5.8.8. *Can we find more applications of coding theory in property (and specifically distribution) testing, or even two-way connections between testing and error-correcting codes?*

Sampling correction In Chapter 5, we introduced the notions of sampling correctors and improvers, and studied some of their applications. We believe investigating further this new paradigm and its interplay with other areas of computational learning to be a fruitful research direction; specifically, we ask the following two questions.

Question 5.8.9. *Is there a sampling corrector (or even improver) for* independence *of probability distributions over $[n]^d$ with (amortized) rate $r < 1/d$, in the sub-learning regime? That is, is there a sampling corrector which, on average, requires strictly fewer than d samples from a close-to-product distribution $\mathbf{p}$ on $[n]^d$ to produce one sample from a corrected product distribution $\mathbf{p}'$ (and does not do so by first learning the distribution $\mathbf{p}$)?*

[18] Recall that closeness testing problem asks to distinguish $\mathbf{p} = \mathbf{q}$ from $\mathrm{d}_{\mathrm{TV}}(\mathbf{p}, \mathbf{q}) > \varepsilon$, where *both* $\mathbf{p}, \mathbf{q} \in \Delta([n])$ are unknown.

The second leans towards more applied considerations; we consider it of significant practical interest:

Question 5.8.10. *Can one revisit the existing literature on data imputation under the viewpoint of sampling correction, and leverage results in the latter to obtain new methods to systematically and rigorously handle missing data?*

Lower bounds for conditional sampling One punchline from Chapter 4 is that proving lower bounds in the conditional sampling model is *hard*. Although the reduction technique from Section 3.2, or the concept of "adaptive core tester" from [54] (see also [1]) can be used to obtain such results, we have staggeringly few methods to argue about what adaptivity allows the testing algorithms to do.

Question 5.8.11. *Can we develop a general information-theoretic characterization of what an algorithm learns by interacting with a conditional sampling (*COND*) oracle? Further, can we exploit this characterization to obtain a general lower bound technique in the conditional sampling setting?*

Distribution testing beyond the discrete setting In contrast with the situation in distribution *learning*, there is no clear notion of how to generalize distribution testing to *continuous* distributions. Indeed, the stringency of the total variation metric implies that, for a naïve extension from the discrete to the continuous case, the sample complexity of most testing questions immediately becomes infinite. One workaround would be to restrict the class of probability distributions, asking that the unknown distribution $\mathbf{p}$ be "smooth enough" (instead of arbitrary). This solution, however, strikes us as lacking in generality; instead, we believe changing the *metric* to be a more elegant path.

Question 5.8.12. *Let $\Delta([0,1])$ be the set of continuous probability distributions on $[0,1]$,* without *smoothness assumptions. What is the "right" notion of metric to consider for distribution testing over $\Delta([0,1])$?*

We note that a natural and promising idea is the Earth mover's distance (also known as Wasserstein), e.g. with regard to L_1.[19] We would welcome a general theory of distribution testing of continuous distributions in Earth mover's distance with open arms, and great interest.

[19]The use of Earth mover's distance (EMD) in distribution testing *was* considered in [12]; however, the authors rely on discretization of the domain and use total variation as a proxy for testing in EMD, which strikes us as somehow sidestepping the question.

Bibliographic Note

Most of the contents of this dissertation have appeared somewhere in some form.

Chapter 2 is based on two papers: the first, "Testing Shape Restrictions of Discrete Distributions," is joint work with Ilias Diakonikolas, Themis Gouleakis, and Ronitt Rubinfeld, and appeared in the Proceedings of the 33rd International Symposium on Theoretical Aspects of Computer Science [51] as well as in the corresponding special issue of *Theory of Computing Systems* [52]. The second, "Fourier-Based Testing for Families of Distributions," is joint work with Ilias Diakonikolas and Alistair Stewart, and is currently under submission; a technical report is available at [45].

Chapter 3 is again based on two papers: the first is [51]. The second, "Distribution Testing Lower Bounds *via* Reductions from Communication Complexity," is joint work with Eric Blais and Tom Gur, and appeared in the Proceedings of the 32nd Computational Complexity Conference [34].

Chapter 4 once more contains research from two papers. The first, "Testing probability distributions using conditional samples," is joint work with Dana Ron and Rocco Servedio, and appeared in the SIAM Journal on Computing [49]. The second, "Testing Probability Distributions Underlying Aggregated Data," is joint work with Ronitt Rubinfeld, and appeared in the Proceedings of the 41st International Colloquium on Automata, Languages and Programming [50].

Finally, Chapter 5 is based on the paper "Sampling Correctors," which is joint work with Themis Gouleakis and Ronitt Rubinfeld, and appeared in the Proceedings of the 7th Innovations in Theoretical Computer Science [46].

Bibliography

[1] Jayadev Acharya, Clément L. Canonne, and Gautam Kamath. "A Chasm Between Identity and Equivalence Testing with Conditional Queries." In: *APPROX-RANDOM*. Vol. 40. LIPIcs. 2015, pp. 449–466.

[2] Jayadev Acharya and Constantinos Daskalakis. "Testing Poisson Binomial Distributions." In: *Proceedings of SODA*. 2015. Chap. 122, pp. 1829–1840.

[3] Jayadev Acharya, Constantinos Daskalakis, and Gautam C. Kamath. "Optimal Testing for Properties of Distributions." In: *Advances in Neural Information Processing Systems 28*. Ed. by C. Cortes, N.D. Lawrence, D.D. Lee, M. Sugiyama, R. Garnett, and R. Garnett. Curran Associates, Inc., 2015, pp. 3577–3598. URL: http://papers.nips.cc/paper/5839-optimal-testing-for-properties-of-distributions.pdf.

[4] Jayadev Acharya, Ilias Diakonikolas, Jerry Zheng Li, and Ludwig Schmidt. "Sample-Optimal Density Estimation in Nearly-Linear Time." In: *Proceedings of SODA*. SIAM, 2017, pp. 1278–1289.

[5] Jayadev Acharya, Ashkan Jafarpour, Alon Orlitsky, and Ananda T. Suresh. "Sorting with adversarial comparators and application to density estimation." In: *Information Theory (ISIT), 2014 IEEE International Symposium on*. June 2014, pp. 1682–1686. DOI: 10.1109/ISIT.2014.6875120.

[6] José A. Adell and Pedro Jodra. "Exact Kolmogorov and total variation distances between some familiar discrete distributions." In: *Journal of Inequalities and Applications* 2006.1 (2006), p. 64307. ISSN: 1029-242X. DOI: 10.1155/JIA/2006/64307.

[7] Nir Ailon, Bernard Chazelle, Seshadhri Comandur, and Ding Liu. "Property-Preserving Data Reconstruction." In: *Algorithmica* 51.2 (2008), pp. 160–182. ISSN: 0178-4617. DOI: 10.1007/s00453-007-9075-9. URL: http://dx.doi.org/10.1007/s00453-007-9075-9.

[8] Maryam Aliakbarpour, Eric Blais, and Ronitt Rubinfeld. "Learning and Testing Junta Distributions." In: *Proceedings of COLT*. Vol. 49. JMLR Workshop and Conference Proceedings. JMLR.org, 2016, pp. 19–46.

[9] Mark Y. An. *Log-concave probability distributions: theory and statistical testing*. Tech. rep. Centre for Labour Market and Social Research, Denmark, 1996. URL: http://EconPapers.repec.org/RePEc:fth:clmsre:96-01.

[10] Sanjeev Arora and Subhash Khot. "Fitting algebraic curves to noisy data." In: *Journal of Computer and System Sciences* 67.2 (2003). Special Issue on STOC 2002, pp. 325 –340.

[11] Sergey V. Astashkin. “Rademacher functions in symmetric spaces.” In: *Journal of Mathematical Sciences* 169.6 (Sept. 2010), pp. 725–886. DOI: 10.1007/s10958-010-0074-z. URL: http://dx.doi.org/10.1007/s10958-010-0074-z.

[12] Khanh Do Ba, Huy L. Nguyen, Huy N. Nguyen, and Ronitt Rubinfeld. “Sublinear Time Algorithms for Earth Mover’s Distance.” In: *Theory of Computing Systems* 48.2 (2011), pp. 428–442.

[13] Mark Bagnoli and Theodore C. Bergstrom. “Log-concave probability and its applications.” English. In: *Economic Theory* 26.2 (2005), pp. 445–469. ISSN: 0938-2259. DOI: 10.1007/s00199-004-0514-4. URL: http://dx.doi.org/10.1007/s00199-004-0514-4.

[14] Ziv Bar-Yossef. “The Complexity of Massive Data Set Computations.” Adviser: Christos Papadimitriou. Available at http://webee.technion.ac.il/people/zivby/index_files/Page1489.html. PhD thesis. UC Berkeley, 2002.

[15] Andrew D. Barbour. “Stein’s method and Poisson process convergence.” In: *J. Appl. Probab.* Special Vol. 25A (1988). A celebration of applied probability, pp. 175–184. ISSN: 0021-9002.

[16] Andrew D. Barbour, Lars Holst, and Svante Janson. *Poisson approximation*. Vol. 2. Oxford Studies in Probability. Oxford Science Publications. The Clarendon Press, Oxford University Press, New York, 1992, pp. x+277. ISBN: 0-19-852235-5.

[17] Richard E. Barlow, Bartholomew D.J, J.M Bremner, and H.D Brunk. *Statistical Inference Under Order Restrictions: The Theory and Application of Isotonic Regression*. Wiley Series in Probability and Mathematical Statistics. London, New York: J. Wiley, 1972. ISBN: 0-471-04970-0.

[18] Vic Barnett. “The study of outliers: purpose and model.” In: *Applied Statistics* (1978), pp. 242–250.

[19] Tuğkan Batu, Ravi Kumar, and Ronitt Rubinfeld. “Sublinear algorithms for testing monotone and unimodal distributions.” In: *Proceedings of STOC*. New York, NY, USA: ACM, 2004, pp. 381–390. ISBN: 1-58113-852-0. DOI: 10.1145/1007352.1007414. URL: http://doi.acm.org/10.1145/1007352.1007414.

[20] Tuğkan Batu, Lance Fortnow, Ronitt Rubinfeld, Warren D. Smith, and Patrick White. “Testing Closeness of Discrete Distributions.” In: *Journal of the ACM* 60.1 (2013). This is a long version of [22]., 4:1–4:25.

[21] Tuğkan Batu, Eldar Fischer, Lance Fortnow, Ravi Kumar, Ronitt Rubinfeld, and Patrick White. “Testing random variables for independence and identity.” In: *Proceedings of FOCS*. 2001, pp. 442–451.

[22] Tuğkan Batu, Lance Fortnow, Ronitt Rubinfeld, Warren D. Smith, and Patrick White. “Testing that distributions are close.” In: *Proceedings of FOCS*. 2000, pp. 189–197.

[23] Tuğkan Batu, Sanjoy Dasgupta, Ravi Kumar, and Ronitt Rubinfeld. “The complexity of approximating the entropy.” In: *SIAM Journal on Computing* 35.1 (2005), pp. 132–150.

[24] Michael Ben-Or, Don Coppersmith, Michael Luby, and Ronitt Rubinfeld. "Non-abelian homomorphism testing, and distributions close to their self-convolutions." In: *Random Struct. Algorithms* 32.1 (2008), pp. 49–70.

[25] Colin Bennett and Robert C. Sharpley. *Interpolation of Operators*. Pure and Applied Mathematics. Elsevier Science, 1988. ISBN: 9780080874487. URL: https://books.google.com/books?id=HpqF9zjZWMMC.

[26] Vidmantas Bentkus. "On the dependence of the Berry-Esseen bound on dimension." In: *J. Statist. Plann. Inference* 113.2 (2003), pp. 385–402. ISSN: 0378-3758. DOI: 10.1016/S0378-3758(02)00094-0. URL: http://dx.doi.org/10.1016/S0378-3758(02)00094-0.

[27] Aditya Bhaskara, Devendra Desai, and Srikanth Srinivasan. "Optimal Hitting Sets for Combinatorial Shapes." In: *15th International Workshop, APPROX 2012, and 16th International Workshop, RANDOM 2012*. 2012, pp. 423–434.

[28] Arnab Bhattacharyya and Yuichi Yoshida. *Property Testing*. Forthcoming, 2017. URL: https://propertytestingbook.wordpress.com/.

[29] Arnab Bhattacharyya, Elena Grigorescu, Madhav Jha, Kyomin Jung, Sofya Raskhodnikova, and David P. Woodruff. "Lower Bounds for Local Monotonicity Reconstruction from Transitive-Closure Spanners." In: *SIAM Journal on Discrete Mathematics* 26.2 (2012), pp. 618–646. DOI: 10.1137/100808186. eprint: http://dx.doi.org/10.1137/100808186. URL: http://dx.doi.org/10.1137/100808186.

[30] Arnab Bhattacharyya, Eldar Fischer, Ronitt Rubinfeld, and Paul Valiant. "Testing monotonicity of distributions over general partial orders." In: *Proceedings of ITCS*. 2011, pp. 239–252.

[31] Rishiraj Bhattacharyya and Sourav Chakraborty. "Property Testing of Joint Distributions using Conditional Samples." In: *ArXiV* abs/1702.01454 (2017).

[32] Lucien Birgé. "On the Risk of Histograms for Estimating Decreasing Densities." In: *The Annals of Statistics* 15.3 (1987), pp. 1013–1022. ISSN: 00905364. URL: http://www.jstor.org/stable/2241812.

[33] Eric Blais, Joshua Brody, and Kevin Matulef. "Property Testing Lower Bounds via Communication Complexity." In: *Computational Complexity* 21.2 (2012), pp. 311–358. DOI: 10.1007/s00037-012-0040-x. URL: http://dx.doi.org/10.1007/s00037-012-0040-x.

[34] Eric Blais, Clément L. Canonne, and Tom Gur. "Distribution Testing Lower Bounds via Reductions from Communication Complexity." In: *Computational Complexity Conference*. Vol. 79. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017, 28:1–28:40.

[35] Avrim Blum, Katrina Ligett, and Aaron Roth. "A learning theory approach to noninteractive database privacy." In: *Journal of the ACM* 60.2 (2013), 12:1–12:25. DOI: 10.1145/2450142.2450148. URL: http://doi.acm.org/10.1145/2450142.2450148.

[36] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. "Self-testing/Correcting with Applications to Numerical Problems." In: *Proceedings of the Twenty-second Annual ACM Symposium on Theory of*

Computing. STOC '90. New York, NY, USA: ACM, 1990, pp. 73–83. ISBN: 0-89791-361-2. DOI: 10.1145/100216.100225. URL: http://doi.acm.org/10.1145/100216.100225.

[37] Christian Borgs, Jennifer Chayes, Nicole Immorlica, Adam Tauman Kalai, Vahab Mirrokni, and Christos Papadimitriou. "The myth of the Folk theorem." In: *Games Econom. Behav.* 70.1 (2010), pp. 34–43. ISSN: 0899-8256. DOI: 10.1016/j.geb.2009.04.016. URL: http://dx.doi.org/10.1016/j.geb.2009.04.016.

[38] Stéphane Boucheron, Gábor Lugosi, and Pascal Massart. *Concentration Inequalities: A Nonasymptotic Theory of Independence*. OUP Oxford, 2013.

[39] Zvika Brakerski. "Local Property Restoring." Manuscript. 2008.

[40] Joshua Brody, Kevin Matulef, and Chenggang Wu. "Lower Bounds for Testing Computability by Small Width OBDDs." In: *TAMC*. Vol. 6648. Lecture Notes in Computer Science. Springer, 2011, pp. 320–331.

[41] Andrea Campagna, Alan Guo, and Ronitt Rubinfeld. "Local reconstructors and tolerant testers for connectivity and diameter." In: *CoRR* abs/1208.2956 (2012).

[42] Clément L. Canonne. "A Survey on Distribution Testing: your Data is Big. But is it Blue?" In: *Electronic Colloquium on Computational Complexity (ECCC)* 22 (Apr. 2015), p. 63.

[43] Clément L. Canonne. "Are Few Bins Enough: Testing Histogram Distributions." In: *Proceedings of PODS*. Association for Computing Machinery (ACM), 2016. DOI: 10.1145/2902251.2902274. URL: http://dx.doi.org/10.1145/2902251.2902274.

[44] Clément L. Canonne. "Big Data on the Rise? Testing Monotonicity of Distributions." In: *Proceedings of ICALP*. Springer, 2015, pp. 294–305. DOI: 10.1007/978-3-662-47672-7_24. URL: http://dx.doi.org/10.1007/978-3-662-47672-7_24.

[45] Clément L. Canonne, Ilias Diakonikolas, and Alistair Stewart. "Fourier-Based Testing for Families of Distributions." In: *Electronic Colloquium on Computational Complexity (ECCC)* 24 (2017), p. 75. URL: http://eccc.hpi-web.de/report/2017/075.

[46] Clément L. Canonne, Themis Gouleakis, and Ronitt Rubinfeld. "Sampling Correctors." In: *Proceedings of ITCS*. ACM, 2016, pp. 93–102.

[47] Clément L. Canonne and Tom Gur. "An Adaptivity Hierarchy Theorem for Property Testing." In: *Computational Complexity Conference*. Vol. 79. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017, 27:1–27:25.

[48] Clément L. Canonne, Dana Ron, and Rocco A. Servedio. "Testing equivalence between distributions using conditional samples." In: *Proceedings of SODA*. Portland, Oregon: Society for Industrial and Applied Mathematics (SIAM), 2014, pp. 1174–1192. ISBN: 978-1-611973-38-9. URL: http://dl.acm.org/citation.cfm?id=2634074.2634161.

[49] Clément L. Canonne, Dana Ron, and Rocco A. Servedio. "Testing probability distributions using conditional samples." In: *SIAM Journal on Computing* 44.3 (2015), pp. 540–616. DOI: 10.1137/130945508.

[50] Clément L. Canonne and Ronitt Rubinfeld. "Testing Probability Distributions Underlying Aggregated Data." In: *Proceedings of ICALP*. 2014, pp. 283–295.

[51] Clément L. Canonne, Ilias Diakonikolas, Themis Gouleakis, and Ronitt Rubinfeld. "Testing Shape Restrictions of Discrete Distributions." In: *Proceedings of STACS*. See also [52] (full version). 2016. DOI: 10.4230/LIPIcs.STACS.2016.25. URL: https://doi.org/10.4230/LIPIcs.STACS.2016.25.

[52] Clément L. Canonne, Ilias Diakonikolas, Themis Gouleakis, and Ronitt Rubinfeld. "Testing Shape Restrictions of Discrete Distributions." In: *Theory of Computing Systems* (2017), pp. 1–59. DOI: 10.1007/s00224-017-9785-6. URL: http://dx.doi.org/10.1007/s00224-017-9785-6.

[53] Sourav Chakraborty, David García-Soriano, and Arie Matsliah. "Efficient Sample Extractors for Juntas with Applications." In: *Automata, Languages and Programming - 38th International Colloquium, ICALP 2011, Zurich, Switzerland, July 4-8, 2011, Proceedings, Part I*. Ed. by Luca Aceto, Monika Henzinger, and Jiri Sgall. Vol. 6755. Lecture Notes in Computer Science. Springer, 2011, pp. 545–556.

[54] Sourav Chakraborty, Eldar Fischer, Yonatan Goldhirsh, and Arie Matsliah. "On the Power of Conditional Samples in Distribution Testing." In: *Proceedings of ITCS*. Berkeley, California, USA: ACM, 2013, pp. 561–580. ISBN: 978-1-4503-1859-4. DOI: 10.1145/2422436.2422497.

[55] Siu-on Chan, Ilias Diakonikolas, Rocco A. Servedio, and Xiaorui Sun. "Efficient density estimation via piecewise polynomial approximation." In: *Proceedings of STOC*. ACM, 2014, pp. 604–613.

[56] Siu-on Chan, Ilias Diakonikolas, Rocco A. Servedio, and Xiaorui Sun. "Learning mixtures of structured distributions over discrete domains." In: *Proceedings of SODA*. 2013, pp. 1380–1394.

[57] Siu-on Chan, Ilias Diakonikolas, Rocco A. Servedio, and Sun. Xiaorui. "Near-Optimal Density Estimation in Near-Linear Time Using Variable-Width Histograms." In: *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*. 2014, pp. 1844–1852.

[58] Siu-on Chan, Ilias Diakonikolas, Gregory Valiant, and Paul Valiant. "Optimal Algorithms for Testing Closeness of Discrete Distributions." In: *Proceedings of SODA*. 2014, pp. 1193–1203.

[59] Louis H. Y. Chen, Larry Goldstein, and Qi-Man Shao. *Normal approximation by Stein's method*. Probability and its Applications (New York). Springer, Heidelberg, 2011, pp. xii+405. ISBN: 978-3-642-15006-7. DOI: 10.1007/978-3-642-15007-4. URL: http://dx.doi.org/10.1007/978-3-642-15007-4.

[60] Sean X. Chen and Jun S. Liu. "Statistical applications of the Poisson-Binomial and conditional Bernoulli distributions." In: *Statistica Sinica* 7.4 (1997). URL: http://www3.stat.sinica.edu.tw/statistica/j7n4/j7n44/j7n44.htm.

[61] Yu Cheng, Ilias Diakonikolas, and Alistair Stewart. "Playing Anonymous Games Using Simple Strategies." In: *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*. Proceedings of SODA. Barcelona, Spain: Society for Industrial and Applied Mathematics, 2017, pp. 616–631. URL: http://dl.acm.org/citation.cfm?id=3039686.3039726.

[62] Herman Chernoff. "A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations." In: *The Annals of Mathematical Statistics* 23 (1952), pp. 493–507.

[63] Constantinos Daskalakis, Ilias Diakonikolas, and Rocco A. Servedio. "Learning k-Modal Distributions via Testing." In: *Theory of Computing* 10.20 (2014), pp. 535–570. DOI: 10.4086/toc.2014.v010a020.

[64] Constantinos Daskalakis, Ilias Diakonikolas, and Rocco A. Servedio. "Learning Poisson Binomial Distributions." In: *Proceedings of STOC*. STOC '12. New York, New York, USA: ACM, 2012, pp. 709–728. ISBN: 978-1-4503-1245-5.

[65] Constantinos Daskalakis, Ilias Diakonikolas, and Rocco A. Servedio. "Learning Poisson Binomial Distributions." In: *Algorithmica* 72.1 (2015), pp. 316–357.

[66] Constantinos Daskalakis and Gautam Kamath. "Faster and Sample Near-Optimal Algorithms for Proper Learning Mixtures of Gaussians." In: *Proceedings of The 27th Conference on Learning Theory, Barcelona, Spain, June 13-15, 2014*. COLT '14. 2014, pp. 1183–1213.

[67] Constantinos Daskalakis, Gautam Kamath, and Christos Tzamos. "On the Structure, Covering, and Learning of Poisson Multinomial Distributions." In: *Proceedings of FOCS*. 2015.

[68] Constantinos Daskalakis and Christos H Papadimitriou. "Approximate Nash equilibria in anonymous games." In: *Journal of Economic Theory* (2014).

[69] Constantinos Daskalakis and Christos H Papadimitriou. "Computing Equilibria in Anonymous Games." In: *Proceedings of FOCS*. 2007, pp. 83–93.

[70] Constantinos Daskalakis and Christos H Papadimitriou. "Discretized Multinomial Distributions and Nash Equilibria in Anonymous Games." In: *Proceedings of FOCS*. 2008, pp. 25–34.

[71] Constantinos Daskalakis and Christos H Papadimitriou. "On Oblivious PTAS's for Nash Equilibrium." In: *Proceedings of STOC*. 2009, pp. 75–84.

[72] Constantinos Daskalakis, Anindya De, Gautam Kamath, and Christos Tzamos. "A Size-Free CLT for Poisson Multinomials and its Applications." In: *Proceedings of STOC*. 2016.

[73] Constantinos Daskalakis, Ilias Diakonikolas, Ryan O'Donnell, Rocco A. Servedio, and Li-Yang Tan. "Learning Sums of Independent Integer Random Variables." In: *Proceedings of FOCS*. IEEE Computer Society, 2013, pp. 217–226.

[74] Constantinos Daskalakis, Ilias Diakonikolas, Rocco A. Servedio, Gregory Valiant, and Paul Valiant. "Testing k-modal Distributions: Optimal Algorithms via Reductions." In: *Proceedings of SODA*. New Orleans, Louisiana: Society for Industrial and Applied Mathematics (SIAM), 2013, pp. 1833–1852.

ISBN: 978-1-611972-51-1. URL: http://dl.acm.org/citation.cfm?id=2627817.2627948.

[75] Anindya De. "Beyond the Central Limit Theorem: asymptotic expansions and pseudorandomness for combinatorial sums." In: *Proceedings of FOCS*. 2015.

[76] Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. "Maximum likelihood from incomplete data via the EM algorithm." In: *J. Roy. Statist. Soc. Ser. B* 39.1 (1977). With discussion, pp. 1–38. ISSN: 0035-9246. URL: http://www.jstor.org/stable/2984875.

[77] Luc Devroye and Gábor Lugosi. *Combinatorial Methods in Density Estimation*. Springer Series in Statistics. Springer New York, 2001. ISBN: 9780387951171. URL: http://books.google.com/books?id=jvT-sUt1HZYC.

[78] Persi Diaconis. "Chapter 2: Basics of Representations and Characters." In: *Group representations in probability and statistics*. Vol. Volume 11. Lecture Notes–Monograph Series. Hayward, CA: Institute of Mathematical Statistics, 1988, pp. 5–16. DOI: 10.1214/lnms/1215467411. URL: http://dx.doi.org/10.1214/lnms/1215467411.

[79] Ilias Diakonikolas. "Learning Structured Distributions." In: *Handbook of Big Data*. CRC Press, 2016.

[80] Ilias Diakonikolas and Daniel M. Kane. "A New Approach for Testing Properties of Discrete Distributions." In: *Proceedings of FOCS*. IEEE Computer Society, 2016.

[81] Ilias Diakonikolas, Daniel M. Kane, and Vladimir Nikishkin. "Optimal Algorithms and Lower Bounds for Testing Closeness of Structured Distributions." In: *Proceedings of FOCS*. 2015.

[82] Ilias Diakonikolas, Daniel M. Kane, and Vladimir Nikishkin. "Testing Identity of Structured Distributions." In: *Proceedings of SODA*. 2015.

[83] Ilias Diakonikolas, Daniel M. Kane, and Alistair Stewart. "Efficient Robust Proper Learning of Log-concave Distributions." In: *CoRR* abs/1606.03077 (2016). URL: http://arxiv.org/abs/1606.03077.

[84] Ilias Diakonikolas, Daniel M. Kane, and Alistair Stewart. "Nearly Optimal Learning and Sparse Covers for Sums of Independent Integer Random Variables." In: *CoRR* abs/1505.00662 (2015).

[85] Ilias Diakonikolas, Daniel M. Kane, and Alistair Stewart. "Optimal Learning via the Fourier Transform for Sums of Independent Integer Random Variables." In: *Proceedings of COLT*. Vol. 49. JMLR Workshop and Conference Proceedings. Full version in [84]. JMLR.org, 2016, pp. 831–849.

[86] Ilias Diakonikolas, Daniel M. Kane, and Alistair Stewart. "Properly Learning Poisson Binomial Distributions in Almost Polynomial Time." In: *Proceedings of COLT*. Full version available at arXiv:1511.04066. 2016, pp. 850–878.

[87] Ilias Diakonikolas, Daniel M. Kane, and Alistair Stewart. "The Fourier Transform of Poisson Multinomial Distributions and its Algorithmic Applications." In: *Proceedings of STOC*. Full version available at arXiv:1511.03592. 2016.

[88] Irit Dinur and Kobbi Nissim. "Revealing information while preserving privacy." In: *Proceedings of PODS*. ACM. 2003, pp. 202–210.

[89] Devdatt P. Dubhashi and Alessandro Panconesi. *Concentration of measure for the analysis of randomized algorithms*. Cambridge: Cambridge University Press, 2009.

[90] Lutz Dümbgen and Kaspar Rufibach. "logcondens: Computations Related to Univariate Log-Concave Density Estimation." In: *J. Statist. Software* 39.6 (2011).

[91] Aryeh Dvoretzky, Jack Kiefer, and Jacob Wolfowitz. "Asymptotic Minimax Character of the Sample Distribution Function and of the Classical Multinomial Estimator." In: *The Annals of Mathematical Statistics* 27.3 (1956), pp. 642–669.

[92] Cynthia Dwork. "Differential privacy: A survey of results." In: *Theory and Applications of Models of Computation*. Vol. 4978. Springer, 2008, pp. 1–19. ISBN: 978-3-540-79227-7.

[93] Cynthia Dwork and Kobbi Nissim. "Privacy-preserving datamining on vertically partitioned databases." In: *Advances in Cryptology–CRYPTO 2004*. Springer. 2004, pp. 528–544.

[94] Moein Falahatgar, Ashkan Jafarpour, Alon Orlitsky, Venkatadheeraj Pichapathi, and Ananda Theertha Suresh. "Faster Algorithms for Testing under Conditional Sampling." In: *Proceedings of COLT*. JMLR Proceedings. 2015, pp. 607–636.

[95] Eldar Fischer. "The Art of Uninformed Decisions: A primer to property testing." In: *Bulletin of the EATCS* 75 (2001), p. 97.

[96] Eldar Fischer, Oded Lachish, and Yadu Vasudev. "Improving and extending the testing of distributions for shape-restricted properties." In: *ArXiV* abs/1609.06736 (2016).

[97] Eldar Fischer, Oded Lachish, and Yadu Vasudev. "Improving and Extending the Testing of Distributions for Shape-Restricted Properties." In: LIPIcs 66 (2017), 31:1–31:14.

[98] Sara A. van de Geer. *Empirical Processes in M-estimation*. Vol. 6. Cambridge University Press, 2000.

[99] Paul W. Goldberg and Stefano Turchetta. "Query Complexity of Approximate Equilibria in Anonymous Games." In: *WINE*. Vol. 9470. Lecture Notes in Computer Science. Springer, 2015, pp. 357–369.

[100] Oded Goldreich. *Introduction to Property Testing*. Forthcoming, 2017. URL: http://www.wisdom.weizmann.ac.il/~oded/pt-intro.html.

[101] Oded Goldreich, ed. *Property Testing: Current Research and Surveys*. LNCS 6390. Springer, 2010.

[102] Oded Goldreich. "The uniform distribution is complete with respect to testing identity to a fixed distribution." In: *Electronic Colloquium on Computational Complexity (ECCC)* 23 (2016), p. 15.

[103] Oded Goldreich, Shafi Goldwasser, and Dana Ron. "Property Testing and Its Connection to Learning and Approximation." In: *Journal of the ACM* 45.4 (July 1998), pp. 653–750.

[104] Oded Goldreich and Dana Ron. *On Testing Expansion in Bounded-Degree Graphs*. Tech. rep. TR00-020. Electronic Colloquium on Computational Complexity (ECCC), 2000.

[105] Parikshit Gopalan, Daniel M. Kane, and Raghu Meka. "Pseudorandomness via the discrete Fourier transform." In: *Proceedings of FOCS*. 2015.

[106] Parikshit Gopalan, Raghu Meka, Omer Reingold, and David Zuckerman. "Pseudorandom generators for combinatorial shapes." In: *SIAM Journal on Computing* 42.3 (2013), pp. 1051–1076. ISSN: 0097-5397. DOI: 10.1137/110854990. URL: http://dx.doi.org/10.1137/110854990.

[107] Ulf Grenander. "On the theory of mortality measurement." In: *Scandinavian Actuarial Journal* 1956.1 (1956), pp. 70–96.

[108] Sudipto Guha, Andrew McGregor, and Suresh Venkatasubramanian. "Streaming and Sublinear Approximation of Entropy and Information Distances." In: *Proceedings of SODA*. Miami, Florida: Society for Industrial and Applied Mathematics (SIAM), 2006, pp. 733–742. ISBN: 0-89871-605-5. URL: http://dl.acm.org/citation.cfm?id=1109557.1109637.

[109] Douglas M Hawkins. *Identification of outliers*. Vol. 11. Springer, 1980.

[110] Trevor J. Hefley, David M. Baasch, Andrew J. Tyre, and Erin E. Blankenship. "Correction of location errors for presence-only species distribution models." In: *Methods in Ecology and Evolution* 5.3 (2014), pp. 207–214. ISSN: 2041-210X. DOI: 10.1111/2041-210X.12144. URL: http://dx.doi.org/10.1111/2041-210X.12144.

[111] Paweł Hitczenko and Stanisław Kwapień. "On the Rademacher series." In: *Probability in Banach spaces, 9 (Sandjberg, 1993)*. Vol. 35. Progr. Probab. Birkhäuser Boston, Boston, MA, 1994, pp. 31–36.

[112] Wassily Hoeffding. "Probability inequalities for sums of bounded random variables." In: *Journal of the American Statistical Association* 58 (1963), pp. 13–30.

[113] Tord Holmstedt. "Interpolation of Quasi-Normed Spaces." In: *Mathematica Scandinavica* 26.0 (1970), pp. 177–199. ISSN: 1903-1807. URL: http://www.mscand.dk/article/view/10976.

[114] Philip Hougaard. "Survival models for heterogeneous populations derived from stable distributions." In: *Biometrika* 73 (1986), pp. 397–96.

[115] Boris Iglewicz and David Caster Hoaglin. *How to detect and handle outliers*. Vol. 16. Asq Press, 1993.

[116] Russell Impagliazzo and David Zuckerman. "How to Recycle Random Bits." In: *Proceedings of the 30th Annual Symposium on Foundations of Computer Science*. SFCS '89. Washington, DC, USA: IEEE Computer Society, 1989, pp. 248–253. ISBN: 0-8186-1982-1. DOI: 10.1109/SFCS.1989.63486. URL: http://dx.doi.org/10.1109/SFCS.1989.63486.

[117] Piotr Indyk, Reut Levi, and Ronitt Rubinfeld. "Approximating and Testing K-histogram Distributions in Sub-linear Time." In: *Proceedings of the 31st Symposium on Principles of Database Systems*. Proceedings of PODS. Scottsdale, Arizona, USA: ACM, 2012, pp. 15–22. ISBN: 978-1-4503-1248-6.

DOI: 10.1145/2213556.2213561. URL: http://doi.acm.org/10.1145/2213556.2213561.

[118] Madhav Jha and Sofya Raskhodnikova. "Testing and Reconstruction of Lipschitz Functions with Applications to Data Privacy." In: *Proceedings of FOCS*. 2011, pp. 433–442. DOI: 10.1109/FOCS.2011.13.

[119] Gautam Kamath. Private communication. 2015.

[120] Michael Kearns, Yishay Mansour, Dana Ron, Ronitt Rubinfeld, Robert E. Schapire, and Linda Sellie. "On the Learnability of Discrete Distributions." In: *Proceedings of the Twenty-sixth Annual ACM Symposium on Theory of Computing*. STOC '94. Montreal, Quebec, Canada: ACM, 1994, pp. 273–282. ISBN: 0-89791-663-8. DOI: 10.1145/195058.195155. URL: http://doi.acm.org/10.1145/195058.195155.

[121] J. Keilson and H. Gerber. "Some Results for Discrete Unimodality." In: *Journal of the American Statistical Association* 66.334 (1971), pp. 386–389. ISSN: 01621459. URL: http://www.jstor.org/stable/2283941.

[122] Arlene K. H. Kim and Richard J. Samworth. "Global rates of convergence in log-concave density estimation." In: *Ann. Statist.* 44.6 (Dec. 2016), pp. 2756–2779. URL: http://arxiv.org/abs/1404.2298.

[123] Julius Kruopis. "Precision of approximation of the generalized binomial distribution by convolutions of Poisson measures." In: *Lithuanian Mathematical Journal* 26.1 (1986), pp. 37–49.

[124] Eyal Kushilevitz and Adi Rosén. "A Randomness-Rounds Tradeoff in Private Computation." English. In: *Advances in Cryptology — CRYPTO '94*. Ed. by YvoG. Desmedt. Vol. 839. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1994, pp. 397–410. ISBN: 978-3-540-58333-2. DOI: 10.1007/3-540-48658-5_36. URL: http://dx.doi.org/10.1007/3-540-48658-5_36.

[125] Roderick J. A. Little and Donald B. Rubin. *Statistical Analysis with Missing Data*. Wiley Series in Probability and Statistics. Second edition. John Wiley & Sons, Inc., 2002. ISBN: 9780471183860.

[126] Wei-Liem Loh. "Stein's method and multinomial approximation." In: *Ann. Appl. Probab.* 2.3 (1992), pp. 536–554. ISSN: 1050-5164. URL: http://links.jstor.org/sici?sici=1050-5164(199208)2:3<536:SMAMA>2.0.CO;2-6&origin=MSN.

[127] Shang-Keng Ma. "Calculation of entropy from data of motion." In: *Journal of Statistical Physics* 26.2 (1981), pp. 221–240. ISSN: 0022-4715. DOI: 10.1007/BF01013169. URL: http://dx.doi.org/10.1007/BF01013169.

[128] S Maciej. *Anticoncentration of the convolution of two characteristic functions*. MathOverflow. http://mathoverflow.net/q/148973 (version: 2013-11-16). 2013. eprint: http://mathoverflow.net/q/148973. URL: http://mathoverflow.net/q/148973.

[129] S Maciej. *Convergence rate of the convolution of almost uniform measures on* $\mathbb{Z}_p$. MathOverflow. http://mathoverflow.net/q/148779 (version: 2013-11-13). 2013. eprint: http://mathoverflow.net/q/148779. URL: http://mathoverflow.net/q/148779.

[130] Benoit Mandelbrot. "New Methods in Statistical Economics." In: *Journal of Political Economy* 71.5 (1963), pp. 421–440.

[131] Pascal Massart. "The Tight Constant in the Dvoretzky–Kiefer–Wolfowitz Inequality." In: *The Annals of Probability* 18.3 (July 1990), pp. 1269–1283. DOI: 10.1214/aop/1176990746. URL: http://dx.doi.org/10.1214/aop/1176990746.

[132] Pascal Massart and Jean Picard. "Concentration inequalities and model selection." In: Lecture Notes in Mathematics. 33, Saint-Flour, Cantal: Springer, 2007. ISBN: 978-3-540-48497-4. URL: http://opac.inria.fr/record=b1122538.

[133] Stephen J. Montgomery-Smith. "The distribution of Rademacher sums." In: *Proceedings of the American Mathematical Society* 109.2 (1990), pp. 517–522.

[134] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. New York, NY: Cambridge University Press, 1995.

[135] Ilan Newman. "Property Testing of Massively Parametrized Problems – A Survey." In: *Property Testing*. Vol. 6390. Lecture Notes in Computer Science. Springer, 2010, pp. 142–157.

[136] Ilan Newman and Mario Szegedy. "Public vs. private coin flips in one round communication games." In: *Proceedings of STOC*. ACM. 1996, pp. 561–570.

[137] Jerzy Neyman. "On the Two Different Aspects of the Representative Method: The Method of Stratified Sampling and the Method of Purposive Selection." In: *Journal of the Royal Statistical Society* 97.4 (1934), pp. 558–625.

[138] Liam Paninski. "A Coincidence-Based Test for Uniformity Given Very Sparsely Sampled Discrete Data." In: *IEEE Transactions on Information Theory* 54.10 (2008), pp. 4750–4755.

[139] Stefano Panzeri, Cesare Magri, and Ludovico Carraro. "Sampling bias." In: *Scholarpedia* 3.9 (2008). revision #91742, p. 4258.

[140] Michal Parnas, Dana Ron, and Ronitt Rubinfeld. "Tolerant property testing and distance approximation." In: *Journal of Computer and System Sciences* 72.6 (2006), pp. 1012–1042.

[141] Jaak Peetre. *A theory of interpolation of normed spaces*. Notas de Matemática, No. 39. Instituto de Matemática Pura e Aplicada, Conselho Nacional de Pesquisas, Rio de Janeiro, 1968, pp. iii+86.

[142] Siméon Denis Poisson. *Recherches sur la probabilité des jugements en matière criminelle et en matière civile: précédées des règles générales du calcul des probabilités*. Bachelier, 1837. URL: http://books.google.fr/books?id=uB8OAAAAQAAJ.

[143] David Pollard. *Asymptopia*. Manuscript. 2003. URL: http://www.stat.yale.edu/~pollard/Books/Asymptopia/ (visited on 11/08/2016).

[144] Ernst L. Presman. "Approximation of binomial distributions by infinitely divisible ones." In: *Theory Probab. Appl.* 28 (1983), pp. 393–403.

[145] Jaikumar Radhakrishnan and Amnon Ta-Shma. "Bounds For Dispersers, Extractors, And Depth-Two Superconcentrators." In: *SIAM Journal on Discrete Mathematics* 13 (2000), p. 2000.

[146] Sofya Raskhodnikova, Dana Ron, Amir Shpilka, and Adam Smith. "Strong lower bounds for approximating distributions support size and the distinct elements problem." In: *SIAM Journal on Computing* 39.3 (2009), pp. 813–842.

[147] Leo Reyzin. *Extractors and the leftover hash lemma*. http://www.cs.bu.edu/~reyzin/teaching/s11cs937/notes-leo-1.pdf. Lecture notes. Mar. 2011.

[148] L. Bruce Richmond and Jeffrey O. Shallit. "Counting Abelian Squares." In: *ArXiv e-prints* (July 2008). arXiv: 0807.5028 [math.CO].

[149] Dana Ron. "Algorithmic and Analysis Techniques in Property Testing." In: *Foundations and Trends in Theoretical Computer Science* 5 (2 2010), pp. 73–205.

[150] Dana Ron. "Property Testing: A Learning Theory Perspective." In: *Foundations and Trends in Machine Learning* 1.3 (2008), pp. 307–402.

[151] Bero Roos. "Closeness of convolutions of probability measures." In: *Bernoulli* 16.1 (2010), pp. 23–50. ISSN: 1350-7265. DOI: 10.3150/08-BEJ171. URL: http://dx.doi.org/10.3150/08-BEJ171.

[152] Bero Roos. "On the rate of multivariate Poisson convergence." In: *J. Multivariate Anal.* 69.1 (1999), pp. 120–134. ISSN: 0047-259X. DOI: 10.1006/jmva.1998.1789. URL: http://dx.doi.org/10.1006/jmva.1998.1789.

[153] Donald B. Rubin. *Multiple imputation for nonresponse in surveys*. John Wiley & Sons, 1987.

[154] Ronitt Rubinfeld. "Taming big probability distributions." In: *XRDS: Crossroads, The ACM Magazine for Students* 19.1 (2012), p. 24. DOI: 10.1145/2331042.2331052. URL: http://dx.doi.org/10.1145/2331042.2331052.

[155] Ronitt Rubinfeld and Rocco A. Servedio. "Testing monotone high-dimensional distributions." In: *Random Structures and Algorithms* 34.1 (Jan. 2009), pp. 24–44. ISSN: 1042-9832. DOI: 10.1002/rsa.v34:1.

[156] Ronitt Rubinfeld and Madhu Sudan. "Robust Characterization of Polynomials with Applications to Program Testing." In: *SIAM Journal on Computing* 25.2 (1996), pp. 252–271.

[157] Maytal Saar-Tsechansky and Foster Provost. "Handling Missing Values when Applying Classification Models." In: *J. Mach. Learn. Res.* 8 (Dec. 2007), pp. 1623–1657. ISSN: 1532-4435. URL: http://dl.acm.org/citation.cfm?id=1314498.1314553.

[158] Amit Sahai and Salil Vadhan. "Manipulating Statistical Difference." In: *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*. American Mathematical Society, 1998, pp. 251–270.

[159] Michael Saks and Comandur Seshadhri. "Local Monotonicity Reconstruction." In: *SIAM Journal on Computing* 39.7 (2010), pp. 2897–2926. DOI: 10.1137/080728561. eprint: http://dx.doi.org/10.1137/080728561. URL: http://dx.doi.org/10.1137/080728561.

[160] Joseph L. Schafer. *Analysis of incomplete multivariate data*. CRC press, 1997.

[161] Paul D. Senese and John A. Vasquez. "A unified explanation of territorial conflict: Testing the impact of sampling bias, 1919–1992." In: *International Studies Quarterly* 47.2 (2003), pp. 275–298.

[162] Debasis Sengupta and Asok K. Nanda. "Log-concave and concave distributions in reliability." In: *Naval Research Logistics (NRL)* 46.4 (1999), pp. 419–433. ISSN: 1520-6750. DOI: 10.1002/(SICI)1520-6750(199906)46:4<419::AID-NAV5>3.0.CO;2-B. URL: http://dx.doi.org/10.1002/(SICI)1520-6750(199906)46:4<419::AID-NAV5>3.0.CO;2-B.

[163] Philip W. Signor and Jere H. Lipps. "Sampling bias, gradual extinction patterns and catastrophes in the fossil record." In: *Geological Society of America Special Papers* 190 (1982), pp. 291–296.

[164] Mervyn J. Silvapulle and Pranab K. Sen. *Constrained Statistical Inference*. John Wiley & Sons, Inc., 2001, pp. i–xvii. ISBN: 9781118165614. DOI: 10.1002/9781118165614.fmatter. URL: http://dx.doi.org/10.1002/9781118165614.fmatter.

[165] Constantino Tsallis, Silvio V. F. Levy, André M. C. Souza, and Roger Maynard. "Statistical-Mechanical Foundation of the Ubiquity of Lévy Distributions in Nature." In: *Phys. Rev. Lett.* 75 (20 Nov. 1995), pp. 3589–3593. DOI: 10.1103/PhysRevLett.75.3589. URL: http://link.aps.org/doi/10.1103/PhysRevLett.75.3589.

[166] Salil P. Vadhan. *Pseudorandomness*. Vol. 7. Now Publishers Inc., 2012, pp. 1–336. ISBN: 1601985940, 9781601985941. DOI: 10.1561/0400000010. URL: http://dx.doi.org/10.1561/0400000010.

[167] Gregory Valiant and Paul Valiant. "A CLT and tight lower bounds for estimating entropy." In: *Electronic Colloquium on Computational Complexity (ECCC)* 17 (2010), p. 179.

[168] Gregory Valiant and Paul Valiant. "An Automatic Inequality Prover and Instance Optimal Identity Testing." In: *Proceedings of FOCS*. 2014.

[169] Gregory Valiant and Paul Valiant. "An Automatic Inequality Prover and Instance Optimal Identity Testing." In: *SIAM Journal on Computing* 46.1 (2017). Journal version of [168]., pp. 429–455.

[170] Gregory Valiant and Paul Valiant. "Estimating the unseen: A sublinear-sample canonical estimator of distributions." In: *Electronic Colloquium on Computational Complexity (ECCC)* 17 (2010), p. 180.

[171] Gregory Valiant and Paul Valiant. "Estimating the Unseen: An $n/\log n$-sample Estimator for Entropy and Support Size, Shown Optimal via New CLTs." In: *Proceedings of STOC*. 2011, pp. 685–694.

[172] Gregory Valiant and Paul Valiant. "The Power of Linear Estimators." In: *Proceedings of FOCS*. See also [167] and [170]. Oct. 2011, pp. 403–412. DOI: 10.1109/FOCS.2011.81.

[173] Paul Valiant. Private communication. May 2015.

[174] Paul Valiant. "Testing symmetric properties of distributions." In: *SIAM Journal on Computing* 40.6 (2011), pp. 1927–1968.

[175] Guenther Walther. "Inference and Modeling with Log-concave Distributions." In: *Statistical Science* 24.3 (2009), pp. 319–327. URL: http://projecteuclid.org/DPubS?service=UI&version=1.0&verb=Display&handle=euclid.ss/1270041258.

[176] Wikipedia. *Hypergeometric distribution — Wikipedia, The Free Encyclopedia*. [Online; accessed 28-May-2016]. 2016. URL: https://en.wikipedia.org/w/index.php?title=Hypergeometric_distribution&oldid=702419153#Multivariate_hypergeometric_distribution.

[177] Wikipedia contributors. "Stratified Sampling." http://en.wikipedia.org/wiki/Stratified_sampling. accessed July 1, 2013.

[178] Wing Hung Wong and Xiaotong Shen. "Probability inequalities for likelihood ratios and convergence rates of sieve MLEs." In: *Ann. Statist.* 23.2 (1995), pp. 339–362. ISSN: 0090-5364. DOI: 10.1214/aos/1176324524. URL: http://dx.doi.org/10.1214/aos/1176324524.

[179] Sergey Yekhanin. *Locally decodable codes*. Now Publishers Inc., 2010. URL: http://research.microsoft.com/apps/pubs/default.aspx?id=141304.

[180] Bin Yu. "Assouad, Fano, and Le Cam." In: *Festschrift for Lucien Le Cam*. Springer, 1997, pp. 423–435. DOI: 10.1007/978-1-4612-1880-7_29. URL: http://dx.doi.org/10.1007/978-1-4612-1880-7_29.

[181] Zhengmin Zhang. "Estimating Mutual Information Via Kolmogorov Distance." In: *IEEE Transactions on Information Theory* 53.9 (2007), pp. 3280–3282. ISSN: 0018-9448. DOI: 10.1109/TIT.2007.903122.

Deferred proofs

We here give the proof of Lemma 1.4.8, restated below:

Lemma 1.4.8 (Adapted from [82, Theorem 11])**.** *There exists an algorithm* CHECK-SMALL-ℓ_2 *which, given parameters $\varepsilon, \delta \in (0,1)$ and $c \cdot \sqrt{|I|}/\varepsilon^2 \log(1/\delta)$ independent samples from a distribution $\mathbf{p}$ over I (for some absolute constant $c > 0$), outputs either* yes *or* no*, and satisfies the following.*

- *If $\|\mathbf{p} - \mathbf{u}_I\|_2 > \varepsilon/\sqrt{|I|}$, then the algorithm outputs* no *with probability at least $1 - \delta$;*
- *If $\|\mathbf{p} - \mathbf{u}_I\|_2 \le \varepsilon/2\sqrt{|I|}$, then the algorithm outputs* yes *with probability at least $1 - \delta$.*

Proof. We first describe an algorithm that distinguishes between $\|\mathbf{p} - \mathbf{u}\|_2^2 \ge \varepsilon^2/n$ and $\|\mathbf{p} - \mathbf{u}\|_2^2 < \varepsilon^2/(2n)$ with probability at least $2/3$, using $C \cdot \frac{\sqrt{n}}{\varepsilon^2}$ samples. Boosting the success probability to $1 - \delta$ at the price of a multiplicative $\log \frac{1}{\delta}$ factor can then be achieved by standard techniques.

Similarly as in the proof of Theorem 11 (whose algorithm we use, but with a threshold $\tau \stackrel{\text{def}}{=} \frac{3}{4}\frac{m^2\varepsilon^2}{n}$ instead of $\frac{4m}{\sqrt{n}}$), define the quantities

$$Z_k \stackrel{\text{def}}{=} \left(X_k - \frac{m}{n}\right)^2 - X_k, \qquad k \in [n]$$

and $Z \stackrel{\text{def}}{=} \sum_{k=1}^n Z_k$, where the X_k's (and thus the Z_k's) are independent by Poissonization, and $X_k \sim \mathrm{Poisson}(m\mathbf{p}(k))$. It is not hard to see that $\mathbb{E}Z_k = \Delta_k^2$, where $\Delta_k \stackrel{\text{def}}{=} (\frac{1}{n} - \mathbf{p}(k))$, so that $\mathbb{E}Z = m^2\|\mathbf{p} - \mathbf{u}\|_2^2$. Furthermore, we also get

$$\mathrm{Var}\, Z_k = 2m^2\left(\frac{1}{n} - \Delta_k\right)^2 + 4m^3\left(\frac{1}{n} - \Delta_k\right)\Delta_k$$

so that

$$\mathrm{Var}\, Z = 2m^2\left(\sum_{k=1}^n \Delta_k^2 + \frac{1}{n} - 2m\sum_{k=1}^n \Delta_k^3\right) \tag{12}$$

(after expanding and since $\sum_{k=1}^n \Delta_k = 0$).

Soundness Almost *straight from [82], but the threshold has changed.* Assume $\Delta^2 \stackrel{\text{def}}{=} \|\mathbf{p} - \mathbf{u}\|_2^2 \ge \varepsilon^2/n$; we will show that $\Pr[\, Z < \tau\,] \le 1/3$. By Chebyshev's inequality, it is sufficient to show that $\tau \le \mathbb{E}Z - \sqrt{3}\sqrt{\mathrm{Var}\, Z}$, as

$$\Pr\left[\, \mathbb{E}Z - Z > \sqrt{3}\sqrt{\mathrm{Var}\, Z}\,\right] \le 1/3\,.$$

As $\tau < \frac{3}{4}\mathbb{E}Z$, arguing that $\sqrt{3}\sqrt{\mathrm{Var}\, Z} \le \frac{1}{4}\mathbb{E}Z$ is enough, i.e. that $48\,\mathrm{Var}\, Z \le (\mathbb{E}Z)^2$. From (12), this is equivalent to showing

$$\Delta^2 + \frac{1}{n} - 2m\sum_{k=1}^n \Delta_k^3 \le \frac{m^2\Delta^4}{96}\,.$$

We bound the LHS term by term.

- As $\Delta^2 \ge \frac{\varepsilon^2}{n}$, we get $m^2\Delta^2 \ge \frac{C^2}{\varepsilon^2}$, and thus $\frac{m^2\Delta^4}{288} \ge \frac{C^2}{288\varepsilon^2}\Delta^2 \ge \Delta^2$ (as $C \ge 17$ and $\varepsilon \le 1$).
- Similarly, $\frac{m^2\Delta^4}{288} \ge \frac{C^2}{288\varepsilon^2} \cdot \frac{\varepsilon^2}{n} \ge \frac{1}{n}$.

- Finally, recalling that[20]
$$\sum_{k=1}^{n} |\Delta_k|^3 \le \left(\sum_{k=1}^{n} |\Delta_k|^2\right)^{3/2} = \Delta^3$$
we get that $\left|2m\sum_{k=1}^{n}|\Delta_k|^3\right| \le 2m\Delta^3 = \frac{m^2\Delta^4}{288}\cdot\frac{2\cdot 288}{m\Delta} \le \frac{m^2\Delta^4}{288}$, using the fact that $\frac{m\Delta}{2\cdot 288} \ge \frac{C}{576\varepsilon} \ge 1$ (by choice of $C \ge 576$).

Overall, the LHS is at most $3\cdot\frac{m^2\Delta^4}{288} = \frac{m^2\Delta^4}{96}$, as claimed.

Completeness Assume $\Delta^2 = \|\mathbf{p}-\mathbf{u}\|_2^2 < \varepsilon^2/(4n)$. We need to show that $\Pr[\,Z \ge \tau\,] \le 1/3$. Chebyshev's inequality implies
$$\Pr\Big[\,Z - \mathbb{E}Z > \sqrt{3}\sqrt{\operatorname{Var} Z}\,\Big] \le 1/3$$
and therefore it is sufficient to show that
$$\tau \ge \mathbb{E}Z + \sqrt{3}\sqrt{\operatorname{Var} Z}$$
Recalling the expressions of $\mathbb{E}Z$ and $\operatorname{Var} Z$ from (12), this is tantamount to showing
$$\frac{3}{4}\frac{m^2\varepsilon^2}{n} \ge m^2\Delta^2 + \sqrt{6}m\sqrt{\Delta^2 + \frac{1}{n} - 2m\sum_{k=1}^{n}\Delta_k^3}$$
or equivalently
$$\frac{3}{4}\frac{m}{\sqrt{n}}\varepsilon^2 \ge m\sqrt{n}\Delta^2 + \sqrt{6}\sqrt{1 + n\Delta^2 - 2nm\sum_{k=1}^{n}\Delta_k^3}\,.$$
Since $\sqrt{1+n\Delta^2 - 2nm\sum_{k=1}^{n}\Delta_k^3} \le \sqrt{1+n\Delta^2} \le \sqrt{1+\varepsilon^2/4} \le \sqrt{5/4}$, we get that the second term is at most $\sqrt{30/4} < 3$. All that remains is to show that $m\sqrt{n}\Delta^2 \ge 3m\frac{\varepsilon^2}{4\sqrt{n}} - 3$. But as $\Delta^2 < \varepsilon^2/(4n)$, $m\sqrt{n}\Delta^2 \le m\frac{\varepsilon^2}{4\sqrt{n}}$; and our choice of $m \ge C\cdot\frac{\sqrt{n}}{\varepsilon^2}$ for some absolute constant $C \ge 6$ ensures this holds. □

[20] For any sequence $x = (x_1,\dots,x_n) \in \mathbb{R}^n$, $p > 0 \mapsto \|x\|_p$ is non-increasing. In particular, for $0 < p \le q < \infty$,
$$\left(\sum_i |x_i|^q\right)^{1/q} = \|x\|_q \le \|x\|_p = \left(\sum_i |x_i|^p\right)^{1/p}.$$
To see why, one can easily prove that if $\|x\|_p = 1$, then $\|x\|_q^q \le 1$ (bounding each term $|x_i|^q \le |x_i|^p$), and therefore $\|x\|_q \le 1 = \|x\|_p$. Next, for the general case, apply this to $y = x/\|x\|_p$, which has unit ℓ_p norm, and conclude by homogeneity of the norm.