New Methods in Sublinear Computation for High Dimensional Problems

Erik Waingarten

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

COLUMBIA UNIVERSITY

2020

ABSTRACT

New Methods in Sublinear Computation for High Dimensional Problems

Erik Waingarten

We study two classes of problems within sublinear algorithms: data structures for approximate nearest neighbor search, and property testing of Boolean functions. We develop algorithmic and analytical tools for proving upper and lower bounds on the complexity of these problems, and obtain the following results:

- We give data structures for approximate nearest neighbor search achieving state-of-the-art approximations for various high-dimensional normed spaces. For example, our data structure for *arbitrary* normed spaces over $\mathbb{R}^d$ answers queries in sublinear time while using nearly linear space and achieves approximation which is sub-polynomial in the dimension.
- We prove query complexity lower bounds for property testing of three fundamental properties: k-juntas, monotonicity, and unateness. Our lower bounds for non-adaptive junta testing and adaptive unateness testing are nearly optimal, and the lower bound for adaptive monotonicity testing is the best that is currently known.
- We give an algorithm for testing unateness with nearly optimal query complexity. The algorithm is crucially adaptive and based on a novel analysis of binary search over long paths of the hypercube.

Contents

List of Figures

Acknowledgments

First and foremost, I thank my advisors, Xi Chen and Rocco Servedio. The one thing I knew coming to Columbia as a first-year was that I wanted to work with Rocco and Xi. Looking back, it's hard for me to envision any alternative which could be as amazing as the past five years were. I remember countless meetings in the fifth floor offices of CSB, discussing problems, conjectures, and balancing both rocks and parameters. From these discussions I learned many interesting research problems and adopted the research philosophy they implicitly carry; namely, that research is and should be a lot of fun.

I also thank Alex Andoni wholeheartedly. I was extremely lucky to have taken Alex's class on algorithms for massive datasets my first semester, and even luckier that he invited me to join a research meeting with him, Ilya Razenshteyn, and Aleksandar "Sasho" Nikolov. Since then, I have been completely captivated by problems from that class, and I still remember that meeting at Max Soha with Ilya and Sasho, where we asked the questions which are now half of this thesis! This takes me to Ilya Razenshteyn, who has been an invaluable mentor and friend. My internship at Microsoft Research, where he supervised me, was filled with research discussions, espressos at Vivace, and beautiful hikes – the least I could do was show him how to grill a proper steak.

I thank all of my co-authors and collaborators: Alex Andoni, Omri Ben-Eliezer, Clement Canonne, Xi Chen, Rajesh Jayaram, Gautam Kamath, Thijs Laarhoven, Amit Levi, Shoham Letzter, Jerry Li, Assaf Naor, Huy Nguyen, Sasho Nikolov, Ramesh Pallavoor, Sofya Raskhodnikova, Ilya Razenshteyn, Rocco Servedio, Li-Yang Tan, and Jinyu Xie. I would especially like to thank Li-Yang Tan for regularly hosting me at Stanford, Prasad Raghavendra for hosting me at Berkeley, and Amit Levi for multiple productive visits. Being part of Columbia's theory group has been wonderful, and I thank everyone in the group for the spurious discussions, lunches, and coffee breaks that made my time there so pleasant.

I cannot understate the love and support I have received (and continue to receive) from my family: my parents Jose Luis and Elizabeth, and siblings Chiara and Yannick. My father, who first taught me what an algorithm is, has been helping me foster my mathematical curiosity since day one. For this, and everything, I cannot thank them enough. Finally, none of this would be possible without my partner and best friend Vitoria – she has been on this incredible journey with me from the beginning, and having her has made these years superb.

Berele – en realidad, esta tesis marca mi segundo titulo. Mi primer “doctorado” lo recibí con vos, mientras charlabamos sobre la vida, la economia, y la matematica (entre varias otras cosas) los mediodías de verano en el Tequendama. Te la dedico a vos. Gracias.

Bibliographic Note

The contents of this thesis have appeared in various papers in some form.

The results in Chapter 1 gives an overview of the results in approximate nearest neighbor search, stating results from papers [12, 18, 14, 16] with co-authors Alexandr Andoni, Thijs Laarhoven, Assaf Naor, Huy L. Nguyen, Aleksandar Nikolov and Ilya Razenshteyn. Chapter 2 gives the formal proofs from [12]. Chapter 3 gives the formal proofs from [14]. Chapter 4 gives the formal proofs from [16]. We refer the reader to the articles [90, 75] about [12, 14, 16] geared toward a general audience.

The results in Chapter 5 are based on the paper [56], which is co-authored with Xi Chen, Rocco A. Servedio, Li-Yang Tan, and Jinyu Xie. Chapter 6 is based on the paper [51], co-authored with Xi Chen and Jinyu Xie. Chapter 7 is based on the paper [50], co-authored with Xi Chen.

Introduction

This thesis is on *sublinear algorithms*, a paradigm in algorithm design addressing new challenges faced by increasing data. The starting point is that modern demands on various computational tasks make "efficient" algorithms unsuitable. The hardness is not necessarily due to the computational complexity of the problem in the classical sense. In fact, most of the problems considered in this thesis have linear time algorithms. Rather, we intend to run these algorithms on massive inputs. In this context, even linear time is unacceptable.

Sublinear algorithms address this setting. The goal is to solve problems with time and/or space which is *significantly* smaller than the size of the input. This focus on ultra-efficiency requires new problem formulations, which usually allow approximate solutions and a small probability of error. The area has had a profound impact in computer science: these include sketching and streaming algorithms in computer networking and databases [3, 109, 21], fast algorithms for linear algebra in machine learning [100, 136], and testing of local codes in hardness of approximation [38, 125]. This thesis covers topics in two areas of sublinear algorithms: approximate nearest neighbor search and property testing of Boolean functions.

Approximate Nearest Neighbors The problem of similarity search, also known as *nearest neighbor search*, is formalized by considering a *distance function* defining a *metric space* on the objects which measures dissimilarity. Then, one designs a *data structure* supporting searches for close objects in that metric space. The nearest neighbor search problem is both an indispensable algorithmic primitive and a hallmark tool in modern data analysis with applications in diverse fields such as machine learning, robotics, and biology [128, 127]. As the number of objects grows and metric spaces become more complex, straightforward solutions become computationally intractable. The *curse of dimensionality* exemplifies this phenomenon: data structures for nearest neighbor search in high dimensional spaces

either have time complexities scaling linearly with the number of objects, or have space complexities scaling exponentially with the dimension of the underlying space.

Major advances were made by allowing approximations, and currently, many techniques exist for designing efficient data structures with good approximations in important special cases. These techniques include Locality-Sensitive Hashing [79, 47, 60], metric embeddings [81, 80, 27], and sketching [93, 7, 15]. The resulting data structures find *approximately* nearest neighbors with *sublinear* time and *polynomial* space complexities. (See the recent survey [9], as well as the thesis [5, 120].)

Property Testing Property testing, introduced in [125, 68], deals with approximate decision making under extreme computational constraints. The algorithmic problems ask to determine, given access to an unknown large object, whether or not it has a particular property. The objects may be functions, graphs, or even probability distributions. Ideally, algorithms answer these questions by inspecting the object in very few locations.

An analytic theme is "local-to-global" reasoning: the idea that global properties may be deduced by making few local (and random) observations. One well-known instance is random sampling for estimating statistics, and while sampling is a central tool in property testing, the techniques stretch way beyond that. New methods of aggregation and consistency of local observations, developed within the context of property testing, offer new perspectives on classic mathematical concepts. These include, for example, linearity of functions, sortedness of total and partial orders, and triangle freeness of large graphs (see [123, 122, 67] for a survey on property testing and the recent textbook [66] for more examples).

Summary of Contributions This thesis is divided in two parts. The first discusses approximate nearest neighbor search, where the motivating question is: what properties of a metric space enable the design of efficient data structures with good approximations? The central concept introduced is the *cutting modulus* of a metric space, a quantity which will determine (in part) the complexity of data structures. We design necessary algorithmic primitives and develop analytical tools to understand the cutting modulus of a metric space, and give new data structures for large classes of metric spaces which were previously not

well understood. The material presented appeared in the following works:

- [12]: "Approximate near neighbors for general symmetric norms", with Alexandr Andoni, Huy L. Nguyen, Aleksandar Nikolov, and Ilya Razenshteyn, appeared in STOC 2017.
- [14]: "Data-dependent hashing via non-linear spectral gaps", with Alexandr Andoni, Assaf Naor, Aleksandar Nikolov, and Ilya Razenshteyn, appeared in STOC 2018.
- [16]: "Hölder homeomorphisms and approximate nearest neighbors", with Alexandr Andoni, Assaf Naor, Aleksandar Nikolov, and Ilya Razenshteyn, appeared in FOCS 2018.

The second part discusses topics in property testing of Boolean functions, and in particular, the well-studied properties of juntas, monotonicity, and unateness. We prove new upper and lower bounds on the query complexity of certain aspects of these testing tasks. A theme is understanding the *power of adaptivity*, i.e., how the query complexity changes when considering adaptive and non-adaptive algorithms. We give a nearly optimal lower bound for testing juntas non-adaptively, a matching upper and lower bound on testing unateness, and a new lower bound for testing monotonicity. The material presented appeared in the following works:

- [56]: "Settling the query complexity of non-adaptive junta testing", with Xi Chen, Rocco A. Servedio, Li-Yang Tan, and Jinyu Xie, appeared in CCC 2017 and Journal of the ACM.
- [51]: "Beyond Talagrand functions: new lower bounds for testing monotonicity and unateness", with Xi Chen and Jinyu Xie, appeared in STOC 2017.
- [50]: "Testing unateness nearly optimally", with Xi Chen, appeared in STOC 2019.

We give a more detailed overview next, where we set up some necessary context and highlight the main results. The following section is on approximate nearest neighbor search, and the following is on property testing of Boolean functions.

Approximate Near Neighbor Search

Definition 1 (ANN Problem)**.** *Let (X, d_X) be a metric space, $r \geq 0$ and $c \geq 1$. The* (r, cr)-Approximate Near Neighbor Search *problem asks to preprocess a set of n points $P \subset X$ into a data structure which supports* near-neighbor queries*: given a query point $q \in X$ such that there is a data point $p \in P$ with $d_X(q, p) \leq r$, return a point $\widehat{p} \in P$ with $d_X(q, \widehat{p}) \leq cr$.*

We refer to $c \geq 1$ as the *approximation* and $r \geq 0$ as the *distance threshold*; both parameters are known during the preprocessing. We frequently abbreviate the (r, cr)-Approximate Near Neighbor Problem as c-ANN when a data structure exists for an arbitrary distance threshold r. The computational task is to produce a data structure D which optimizes the *space* the data structure occupies and the *time* it takes to answer a single query. The data structures will be *randomized*: the preprocessing step and the query step may use randomness, and a query must succeed with probability at least $2/3$ over the randomness in the preprocessing and query step.

The exact version of the problem (when $c = 1$) has been intensely studied since the early work of Minsky and Papert (see [106] Section 12.7), and is considered one of the fundamental problems in computational geometry [119]. When the metric space consists of vectors in $\mathbb{R}^d$ with some norm[1] specifying distances, the best known data structures have runtime $O(dn)$ (amounting to a scan of the dataset for each query), or have exponential space dependence with respect to the dimension, for instance, $n^{O(d)}$. In fact, this "curse of dimensionality" seems to be an intrinsic barrier [134, 2, 135].

The works of Indyk and Motwani [79], as well as Kushilevitz, Ostrovsky and Rabani [93] obtained sublinear query time and polynomial space data structures for small approximations over ℓ_1^d and ℓ_2^d. For example, [79] design a data structure for c-ANN over ℓ_1^d with query time $O(dn^{1/c} \log(n))$ and space $O(nd + n^{1+1/c})$. These works ushered a new class of ANN algorithms side-stepping the "curse of dimensionality" in high-dimensional normed spaces,

[1] A normed space is a metric space specified by a vector space V and a map, or norm, $\|\cdot\| \colon V \to \mathbb{R}^{\geq 0}$, which satisfies some additional *norm axioms*: 1) $\|x\| \geq 0$ and is 0 only if $x = 0$, 2) $\|\alpha x\| = |\alpha| \|x\|$ for all $\alpha \in \mathbb{R}$ and $x \in V$, and 3) $\|x + y\| \leq \|x\| + \|y\|$ for all $x, y \in V$. The distance between two vectors $x, y \in V$ is given by $\|x - y\|$.

and pioneered the use of *randomized space partitions* for data structure design. These data structures provide a prototypical example of the theorems we are after. Formally, we want the following guarantee.

Definition 2 (c-ANN Data Structure)**.** *For a metric space (X, d_X), there exists a data structure for c-ANN over X if the following holds. For any $n \in \mathbb{N}$ and $r \geq 0$, there exists $s = s(n) \in \mathbb{N}$ (denoting the* space complexity*), $t = t(n) \in \mathbb{N}$ (denoting the* time complexity*), $t_{\mathrm{cp}} = t_{\mathrm{cp}}(n) \in \mathbb{N}$ (denoting the* cell-probe complexity*), and $w = O(\log(n))$ (denoting the* word size*), as well as two (randomized) functions:*

- *A preprocessing function,* $\texttt{Preprocess}\colon X^n \to (\{0,1\}^w)^s$*, taking as input n points in X, and outputting a data structure $D \in (\{0,1\}^w)^s$ of s words of w bits each.*
- *A querying algorithm* $\texttt{Query}\colon X \times (\{0,1\}^w)^s \to X$*, which given as input a query $q \in X$, access to w-bit words from a data structure D, outputs a point $p \in X$. The algorithm probes at most t_{cp} words from D and computing $\texttt{Query}(\cdot,\cdot)$ takes at most t time.*[2]

For any dataset $P = (p_1, \ldots, p_n) \in X^n$, and any $q \in X$ where $d_X(p_i, q) \leq r$ for some $i \in [n]$, we denote the random variable given by the output of the data structure on query q,

$$\boldsymbol{p} = \texttt{Query}(q, \texttt{Preprocess}(P)) \in P.$$

The output point $\boldsymbol{p}$ should satisfy

$$\mathbf{Pr}\left[d_X(\boldsymbol{p}, q) \leq cr\right] \geq \frac{2}{3}.$$

We frequently consider growing families of metric spaces, where $\log|X|$ is useful measure of complexity: points in X are encoded using $O(\log|X|)$ bits, and we assume that evaluating distances between two points $a, b \in X$ takes $\mathrm{polylog}|X|$ time. Infinite metric spaces are handled via an (natural) encoding of a finite subset of the metric space. For example, points in $\mathbb{R}^d$ are encoded by d coordinates of w bits each, where $w = O(\log n)$.

[2]Notice that $t_{\mathrm{cp}} \leq t$, since the running time of the algorithm is at least the number of times it probes the data structure.

The induced metric space is on $\{-n^{O(1)}, \ldots, n^{O(1)}\}^d$, and since the querying algorithm reads w-bit words, computing distances tends to take $O(d)$ or $\mathrm{poly}(d)$ time.[3]

Theorem 1 (Indyk and Motwani's ℓ_1 Data Structure [79])**.** *For any $c \geq 1$, there exists a data structure for c-ANN over ℓ_1^d with space complexity $O(nd + n^{1+1/c})$ and time complexity $O(dn^{1/c} \log(n))$.*

For the rest of this thesis, a data structure is *efficient* if the time and space complexity are akin to Theorem 1, i.e., the query time is sublinear in the number of data points n and polynomial in the underlying dimension d (or in $\log |X|$ in the case of a finite metric (X, d_X)), and the space is polynomial in the dimension and the number of data points.[4] Theorem 1 remains highly efficient even for approximation factors c which are close to 1. This is a major accomplishment of [79, 93], and we will encounter metric spaces for which the best approximation c must be at least some fixed constant (larger than 1) or fixed factor which grows with d, or n. For example, we will, at times, heavily rely on the following theorem of Indyk [78], obtaining a $O(\log \log d)$-ANN data structure for ℓ_∞^d.

Theorem 2 (Indyk's ℓ_∞ Data Structure [78])**.** *For any $\rho > 0$, there exists a data structure for $(4\lceil \log_{1+\rho} \log(4d) \rceil + 1)$-ANN over ℓ_∞^d with space complexity $O(dn^{1+\rho})$ and time complexity $O(d \log n)$.*

A rich theory for ANN algorithms for Hamming/Manhattan (ℓ_1) and Euclidean (ℓ_2) metrics developed following the works of [79, 93]. The new data structures were based on *hashing* — in particular, Locality-Sensitive Hashing (LSH) [6] and its data-dependent counterparts [13, 10, 17]. However, the landscape quickly blurs when considering metric spaces beyond ℓ_1, ℓ_2 and ℓ_∞, as well as some other notable examples. Each ANN data structure tends to be tailored to the specific metric space, and there lacks a general recipe for dealing with new metric spaces, either from the upper bounds or lower bounds perspective. This state of affairs motivates the following broad question.

[3] We work in the Word RAM model, so we assume that bit-wise operations on words, such as arithmetic, may be done in constant time.

[4] Another interesting parameter to optimizing is the time complexity of *preprocessing*. This quantity is not discussed in this thesis, and for the most part, the preprocessing time for data structures presented is inefficient.

Problem 1. *For a given approximation $c > 1$, which metric spaces admit efficient c-ANN algorithms?*

The exploration is after a "simple" characterization of metric spaces admitting efficient c-ANN data structures. We want a theory which specifies, for each metric space and approximation factor $c > 1$, how to build a data structure for that metric space, and a proof that the approximation cannot be improved. Problem 1 is motivated from several complementary directions. The first is the ubiquity of similarity search data structures in practice, and the need for data structures in important metric spaces where ANN is poorly understood (e.g. the Earth Mover's Distance (EMD), the edit distance, generalized versions of the Hamming distance[5], etc). Oftentimes, the data structures used for ANN require modern mathematical machinery and are highly non-intuitive; such circumstances necessitate a thorough and rigorous approach to study the performance of these data structures. The second, perhaps more theoretical motivation, aims to understand ANN at a fundamental level. The goal is to identify the geometric properties governing the hardness of ANN, and to develop algorithmic primitives for manipulating high dimensional data in sublinear time.

Our Contributions in Approximate Nearest Neighbor Search

In the first part of this thesis, we develop algorithmic and analytical tools for addressing Problem 1. The first chapter gives a broad tour of results in approximate nearest neighbor search, and subsequent chapters formalize the notions developed there. We explore the "embeddings approach" as a powerful (but also limited) paradigm for data structure design. For a given metric space (X, d_X), the goal is to design an embedding $f\colon X \to Y$, where (Y, d_Y) is another metric space, such that distances are roughly preserved, i.e., $d_X(x, y) \approx d_Y(f(x), f(y))$, and (Y, d_Y) is "algorithmically tractable," in the sense that there exists an efficient ANN data structure over Y. Then, a data structure for ANN over X proceeds by first mapping all points according to f, and then utilizing the data structure over Y. The tradeoff is between the extent distances in Y approximate distances in X under f and the tractability of the space (Y, d_Y). We use this approach to design a data structure for any

[5]E.g., a metric of interest in applications is (X^d, ρ_{X^d}), where X is a metric itself, with the distance between vectors $x, y \in X^d$ defined as $\rho_{X^d}(x, y) = \sum_{i=1}^{d} d_X(x_i, y_i)$.

symmetric normed space[6] with $\mathrm{poly}(\log\log n)$ approximation [12] (Chapter 1, Section 1.1).

Theorem 3 (ANN for Symmetric Norms)**.** *Let $X = (\mathbb{R}^d, \|\cdot\|_X)$ be any symmetric normed space. There exists a data structure for $\mathrm{poly}(\log\log n)$-ANN over X using space complexity $\mathrm{poly}(d) \cdot n^{1+o(1)}$ and time complexity $\mathrm{poly}(d) \cdot n^{o(1)}$.*

The conceptual significance of Theorem 3 is its generality and modularity. The resulting data structure was the first to improve on a $O(\sqrt{d})$-approximation[7] for a large class of d-dimensional normed spaces. Consequently, Theorem 3 (in [12]) initiated the research direction aimed at answering Problem 1 for high dimensional normed spaces.

In Chapter 1, Section 1.2, we briefly discuss the list-of-points model, which is a useful abstraction for hashing-based data structures [18], and prove a lower bound for the complexity of data structures over the shortest path metric on a constant degree expander. The implication of this lower bound, namely, that ANN over the shortest path metric on expanders is computationally hard, will be clear shortly. Then, we introduce the *cutting modulus* of a metric space, the metric invariant which will play a prominent role in this thesis. A more thorough discussion of the cutting modulus appears in Chapter 1, Section 1.3; however, we present an informal definition here. The subsequent paragraphs and Figure 0.1 give a useful consequence.

Definition 3 (The Cutting Modulus – Simplified)**.** *Let (X, d_X) be a finite metric space and $\varepsilon \in (0,1)$. Then, X has cutting modulus $\Xi(X,\varepsilon) \geq 0$, if for any $r \geq 0$ and any d-regular*[8] *graph $G = (V, E)$ with vertices $V \subset X$ and edges supported on points within distance r,*[9] *at least one of the following two scenarios hold.*

---

[6] Symmetric normed spaces are specified by norms $\|\cdot\|_X : \mathbb{R}^d \to \mathbb{R}_{\geq 0}$, which in addition to satisfying norm axions, are invariant to permutations of coordinates and sign flips, i.e., in other words, for any $x \in \mathbb{R}^d$, and any bijection $\pi : [d] \to [d]$, letting $x_\pi \in \mathbb{R}^d$ be the vector with $(x_\pi)_i = |x_{\pi(i)}|$, we have $\|x\|_X = \|x_\pi\|_X$.

[7] A data structure achieving approximation $O(\sqrt{d})$ for any normed space over $\mathbb{R}^d$ follows from an application of John's theorem to embed into ℓ_2^d [82, 40] , and then applying the data structure of [79].

[8] We note that this is a simplification of the definition. In particular, we will not assume G is regular, and we will also consider the weighted graph version. We refer the reader to Chapter 1 Section 1.4 for the formal definition.

[9] In other words, $(a,b) \in E$ only if $d_X(a,b) \leq r$

1. ***Cluster scenario**: Half of the vertices of the graph G are clustered within a set of diameter at most $\Xi(X, \varepsilon) \cdot r$. In other words, there exists $B \subset X$ with $\mathrm{diam}_X(B) \leq \Xi(X, \varepsilon) \cdot r$ with $|B \cap V| \geq |V|/2$.*

2. ***Cut scenario**: There exists a subset $S \subset V$ of conductance at most ε,*

$$\Phi_G(S) = \frac{|E(S, V \setminus S)|}{d \cdot \min\{|S|, |V \setminus S|\}} \leq \varepsilon. \tag{1}$$

It is useful to think of $G = (V, E)$ as encoding near-neighbor relationships among all points to be considered by an ANN data structure; specifically, the dataset P will be a subset of $V \subset X$, and the query $q \in V$ will be connected by an edge in E to a near-neighbor in P. Intuitively, Definition 3 aids in building a data structure by recursively partitioning the metric space X. For example, consider the "cluster scenario" first, where there exists a set $B \subset X$ of diameter $\mathrm{diam}_X(B) \leq r \cdot \Xi(X, \varepsilon)$ with $|B \cap V| \geq |V|/2$, so at least half of vertices in G lie in a cluster of diameter $r \cdot \Xi(X, \varepsilon)$. It is conceivable that a data structure would choose one vertex in G to represent all vertices within the cluster (up to approximation $r \cdot \Xi(X, \varepsilon)$), since such a representative would summarize at least half of the graph G. On the other hand, if no such set B exists, we are in the "cut scenario", and there exists a low-conductance cut $S \subset V$ of G. Suppose, furthermore, that $S \subset V$ is a balanced separator.[10] The data structure may then partition G according to S and recurse on both sides; if the query-and-neighbor pair (q, p) were sampled by a uniformly random edge in E, the probability that the data structure errs from partitioning G according to S is small, i.e., bounded by $O(\varepsilon)$.

Another useful example to consider Definition 3 is the shortest path metric on an expander graph. Specifically, we let $G = (V, E)$ be a 3-regular expander graph, and we consider the metric space (V, d_G) where $d_G(x, y)$ is the length of the shortest path between vertex x and vertex y in G. Since G is an expander graph, there are no low-conductance cuts; formally, every set $S \subset V$ satisfies $\Phi_G(S) = \Omega(1)$. Furthermore, by virtue of G being 3-regular, the number of vertices within any set of diameter D is at most 3^D; this means that

[10] We have that $\Phi_G(S) \leq \varepsilon$ and the denominator of the left-hand side of (1) is at least $d|V|/3$, i.e., $|V|/3 \leq |S| \leq 2|V|/3$.

no set of diameter less than $\log_3(|V|/2)$ contains half of the vertices in G. We have thus shown that for a small constant $\varepsilon > 0$, $\Xi((V, d_G), \varepsilon) \geq \Omega(\log(|V|))$.[11]

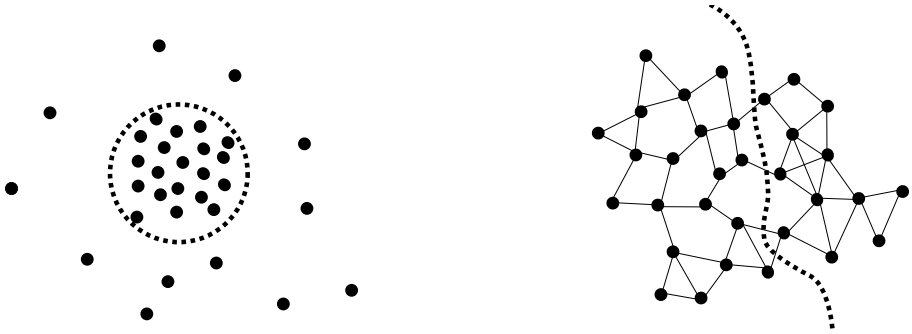

Figure 0.1: The "cluster-or-cut" scenario for metric spaces displayed. See Definition 3 and the subsequent paragraphs. On the left-hand side, the "cluster scenario" is displayed, where the vertices of a graph $G = (V, E)$ satisfy that at least half lie in a cluster of radius less than $r \cdot \Xi(X, \varepsilon)$. On the other hand, the right-hand side shows a graph $G = (V, E)$ without any clusters, and as a result, the cutting modulus implies that a low-conductance cut of the graph exists.

As shortest path metrics on expander graphs have high cutting modulus, upper bounds on the cutting modulus of a metric space (X, d_X) measure the "non-expander" nature of (X, d_X). Our use for the cutting modulus is the following theorem, which gives a (cell-probe) data structure for any metric space (X, d_X) whose approximation is given by the cutting modulus. Conceptually, the theorem shows that *there exists a data structure for ANN over (X, d_X) whenever the metric space (X, d_X) is unlike a shortest path metric on an expander graph*. We refer to Chapter 1 Section 1.2 and Section 1.4 for a more elaborate discussion on this point.

Theorem 4 (ANN Cell-Probe Data Structure from Cutting Modulus)**.** *For any metric space (X, d_X) and any $\varepsilon \in (0, 1)$, there exists a data structure for $O(\Xi(X, \varepsilon))$-ANN over X with space complexity $\mathrm{poly}(\log |X|) \cdot n^{1+O(\varepsilon)}$ and cell-probe complexity $\mathrm{poly}(\log |X|) \cdot n^{O(\varepsilon)}$.*

The introduction of the cutting modulus allows for leveraging a growing body of work on metric spectral gaps for data structure design [105, 113, 111]. Metric spectral gaps were originally developed to prove nonembeddability theorems, and here, we use these techniques for upper bounding the cutting modulus, and in turn, designing data structures.

[11] This bound is maximal, since the diameter of a constant degree expander is $O(\log |V|)$.

For example, if $X = (\mathbb{R}^d, \|\cdot\|_X)$ is a normed space, Theorem 1 in [111] and Cheeger's inequality [48, 4] will imply an upper bound of $\Xi(X, \varepsilon) \lesssim \log d/\varepsilon^2$, and hence, Theorem 4 gives a cell-probe data structure for $O(\log d)$-ANN over any d-dimensional normed space. In addition, the cutting modulus sheds new light on sufficient conditions for (data-dependent) locality-sensitive hashing, as well as new randomized space partitions for classical normed spaces (see Chapter 1, Section 1.4). Given the applicability of locality-sensitive hashing to other algorithmic problems (beyond ANN), we suspect the cutting modulus could play a central role there.

Beyond the black-box application of Theorem 4, this approach yields time-efficient algorithms (as opposed to merely cell-probe efficiency) by augmenting Definition 3. In particular, the goal is to bound the computational complexity of low conductance cuts S found in the cut scenario.[12]

The theorems we present next may be considered the "crown jewels" of our exploration into Problem 1, as these constitute significant quantitative improvements to prior known ANN data structures. (See Chapter 1, Section 1.4 and Section 1.5 for a high level discussion on how this theorem is obtained.)

Theorem 5 (ANN Data Structure for Any Norm)**.** *There exists a data structure for $d^{o(1)}$-ANN over X, where $X = (\mathbb{R}^d, \|\cdot\|_X)$ is any d-dimensional normed space with space complexity $\mathrm{poly}(d) \cdot n^{1+o(1)}$ and time complexity $\mathrm{poly}(d) \cdot n^{o(1)}$.*

Prior to this work, the best approximation factor achievable in such generality was a $O(\sqrt{d})$-ANN data structure, which readily follows from the "embeddings approach:" the embedding of any d-dimensional normed space over $\mathbb{R}^d$ into ℓ_2^d is given by John's theorem proceeded by known data structures for ℓ_2^d (see Chapter 1, Section 1.1 for a more thorough discussion). Furthermore, a study of the cutting modulus of specific metric spaces gives new ANN data structures for classic normed spaces improving on the state-of-the-art. For the classic ℓ_p^d normed spaces, previous data structures achieved approximation $O(2^p)$ [114, 27]

[12] In particular, in the cut scenario of Definition 3, the set S should be accompanied by an algorithm which given a point $x \in V$, determines whether $x \in S$ or $x \notin S$ in time polynomial in the representation of x (which is logarithmic in the size of the graph). See Chapter 1, Section 1.4 for more discussion.

or $O(\log \log d)$ [5]. For the Schatten-p normed spaces, no (non-trivial) data structures were known.[13]

Theorem 6 (ANN Data Structure for ℓ_p Norms)**.** *For any $\varepsilon \in (0,1)$ and $p \in [1,\infty)$, there exists a data structure for $O(p/\varepsilon)$-ANN over ℓ_p^d with space complexity $\mathrm{poly}(d) \cdot n^{1+\varepsilon}$ and time complexity $\mathrm{poly}(d) \cdot n^{\varepsilon}$.*

Theorem 7 (ANN Data Structure for Schatten-p Norms)**.** *For any $\varepsilon \in (0,1)$ and $p \in [1,\infty)$, there exists a data structure for $O(p/\varepsilon)$-ANN over S_p^d [14] with space complexity $\mathrm{poly}(d) \cdot n^{1+\varepsilon}$ and time complexity $\mathrm{poly}(d) \cdot n^{\varepsilon}$.*

Property Testing of Boolean Functions

The second part of this thesis studies property testing of Boolean functions. The goal is to study highly efficient randomized algorithms which are given black-box query access to a Boolean function $f\colon \{-1,1\}^n \to \{-1,1\}$,[15] and *approximately* decide whether f has a particular property $\mathcal{P}$. In general, the property $\mathcal{P}$ is a collection of Boolean functions, namely, those which have the property. The approximation refers to the fact that algorithms may mistakenly decide f has property $\mathcal{P}$ when f is very "close" to having the property. Formally, we will consider the following definition. We refer the reader to Chapter 1 of the textbook [66] for thorough justification of the definition below.

Definition 4 (Property Testing for $\mathcal{P}$ [68])**.** *Fix $n \in \mathbb{N}$, $\varepsilon \in (0,1)$, and a set $\mathcal{P} \subset \{f : \{-1,1\}^n \to \{-1,1\}\}$. The problem of* testing $\mathcal{P}$ *asks to design a randomized algorithm which receives black-box query access to an unknown Boolean function $f\colon \{-1,1\}^n \to \{-1,1\}$, and outputs* `accept` *or* `reject`*. If $f \in \mathcal{P}$, the algorithm outputs* `accept`

[13] The trivial data structure for Schatten-p norms interprets the matrix as a vector in ℓ_2, and obtains approximation $d^{|1/2-1/p|}$.

[14] For $p \in [1,\infty)$, the Schatten-p normed space $S_p^d = (\mathbb{R}^{d\times d}, \|\cdot\|_{S_p})$ is defined over the space of $d \times d$ matrices of real numbers, where for a matrix $x \in \mathbb{R}^{d\times d}$, $\|x\|_{S_p} = (\sum_{i=1}^{d} \sigma_i(x)^p)^{1/p}$, where $\sigma_1(x), \ldots, \sigma_d(x) \in \mathbb{R}$ are the singular values of x. We note S_1 is also known as the "nuclear norm" or "trace norm" and S_∞ as the "operator norm."

[15] In particular, algorithms interact with an oracle for the function $f\colon \{-1,1\}^n \to \{-1,1\}$. A query specifies an input $x \in \{-1,1\}^n$, and the oracle outputs the value $f(x)$.

with probability at least $2/3$*, and if* $\min_{g \in \mathcal{P}} \mathbf{Pr}_{\boldsymbol{x} \sim \{-1,1\}^n}[f(\boldsymbol{x}) \neq g(\boldsymbol{x})] \geq \varepsilon$*, the algorithm outputs* `reject` *with probability at least* $2/3$*.*

The set $\mathcal{P}$ is called the *property* and $\varepsilon \in (0,1)$ is called the *proximity parameter*. We say f is ε-far from $\mathcal{P}$ when $\min_{g \in \mathcal{P}} \mathbf{Pr}_{\boldsymbol{x} \sim \{-1,1\}^n}[f(\boldsymbol{x}) \neq g(\boldsymbol{x})] \geq \varepsilon$. Algorithms testing $\mathcal{P}$ (primarily) focus on minimizing the query complexity, which perhaps surprisingly, is oftentimes a universal constant (depending only on ε) or small function of n and ε.[16] From a purely structural perspective, efficient algorithms testing $\mathcal{P}$ give query-efficient and randomized characterizations of $\mathcal{P}$ which are robust to small changes to the function (in particular, this is the view spearheaded in [125]). The contrapositive of Definition 4 implies that if an algorithm testing $\mathcal{P}$ outputs `accept` on an input f with probability at least $2/3$, then f must be "close" to $\mathcal{P}$; i.e., there exists $g \in \mathcal{P}$ with $\mathbf{Pr}_{\boldsymbol{x} \sim \{-1,1\}^n}[f(\boldsymbol{x}) = g(\boldsymbol{x})] \geq 1 - \varepsilon$.[17]

From the computational perspective, Definition 4 and its various manifestations offers a model for approximate decision-making in sublinear time. Our aim, in this thesis, is to develop some of the algorithmic tools needed for these analyzes. Property testing provides a space for the development of these techniques, as well as for a systematic evaluation of their relative power. We give a short example of an important question in the analysis of testing algorithms, commonly referred to as the *power of adaptivity*, which plays a prominent role in this thesis and property testing research more broadly (see [69, 71, 122, 42] for a short list).

One common stipulation is that testing algorithms be *non-adaptive*: these algorithms query the unknown function f at inputs which are independent of responses to previous queries.[18] They are more desirable to some extent; non-adaptive algorithms are maximally

[16]The trivial algorithm (aside from querying the entire input) queries the unknown function f at $O(\log|\mathcal{P}|/\varepsilon)$ uniformly random inputs. The algorithm scans through all $g \in \mathcal{P}$ and determines whether f agrees on all randomly chosen inputs with some function in $\mathcal{P}$. If it does, the algorithm outputs `accept` , to indicate its belief that $f \in \mathcal{P}$; otherwise, it outputs `reject` . Notice that since the total number of functions $f\colon \{-1,1\}^n \to \{-1,1\}$ is 2^{2^n}, $\log|\mathcal{P}|$ could be as large as 2^n.

[17]The probability that two functions disagree on a uniformly randomly input defines a metric over the space of functions $f\colon \{-1,1\}^n \to \{-1,1\}$. We write $\operatorname{dist}(f,g) = \mathbf{Pr}_{\boldsymbol{x} \sim \{-1,1\}^n}[f(\boldsymbol{x}) \neq g(\boldsymbol{x})]$ and the distance to a property $\mathcal{P}$ as $\min_{g \in \mathcal{P}} \operatorname{dist}(f,g)$.

[18]Without loss of generality, we may assume a q-query non-adaptive algorithm generates a set of queries $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_q) \in (\{-1,1\}^n)^q$ drawn from some distribution $\mathcal{D}$, and outputs `accept` or `reject` according to

parallelizable, simpler to implement, and simpler to analyze. Yet, certain scenarios require "adaptivity," i.e., dependence on responses to previous queries, to achieve query-optimal algorithms.

For example, consider the task of finding a *bi-chromatic edge* of a Boolean function $f\colon \{-1,1\}^n \to \{-1,1\}$.[19] In all testing problems considered here, a single bi-chromatic edge of f will convey important information about f, and finding (special) bi-chromatic edges of f will often be the core of the analysis.

One well-known technique, which plays a crucial role in algorithm design more generally, is *random sampling*. Let the *total influence* of f, denoted I_f, be the number of bi-chromatic edges of f divided by 2^n. Since the total number of edges in $\{-1,1\}^n$ is $n2^{n-1}$, an algorithm which samples edges of $\{-1,1\}^n$ uniformly at random and queries both endpoints will find a bi-chromatic edge of f after roughly $O(n/I_f)$ attempts. Note that this sampling algorithm is non-adaptive, since the algorithm may (randomly) choose the $O(n/I_f)$ queries in advance.

On the other hand, there is another algorithmic technique which may dramatically improve on the "non-adaptive edge sampling" algorithm. An (improved) algorithm may proceed by sampling two inputs $\boldsymbol{x}, \boldsymbol{y} \sim \{-1,1\}^n$ uniformly at random, and executing the following sub-routine, which we informally term "binary search" (see also, Figure 0.2).

---

the *response vector* $(f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_q)) \in \{-1,1\}^q$.

[19] An *edge* of the hypercube $(x,y) \in \{-1,1\}^n \times \{-1,1\}^n$ is a pair of points which differ on a single coordinate. In other words, there exists exactly one index $i \in [n]$ such that $x_i \neq y_i$ and $x_j = y_j$ for all $j \in [n] \setminus \{i\}$. The edge (x,y) is *bi-chromatic* under the Boolean function $f\colon \{-1,1\}^n \to \{-1,1\}$ if $f(x) \neq f(y)$. For $x \in \{-1,1\}^n$ and $i \in [n]$, the vector $x^{(i)} \in \{-1,1\}^n$ is given by negating the i-th bit of x. Hence, we $(x, x^{(i)})$ is an edge, and we will say the edge $(x, x^{(i)})$ is in *direction* i.

Informal Description of Subroutine $\texttt{BinarySearch}(f, x, y)$.

Input: Query access to $f\colon \{-1,1\}^n \to \{-1,1\}$, and two points $x, y \in \{-1,1\}^n$.
Output: A bi-chromatic edge of f along a direction in $\{i \in [n] : x_i \neq y_i\}$, or "fail."

1. Query $f(x)$ and $f(y)$ and return "fail" if $f(x) = f(y)$. Otherwise, $f(x) \neq f(y)$, and we proceed to the next step.
2. If (x, y) is an edge of $\{-1,1\}^n$, output (x, y).
3. Otherwise, let $x = a_0, \ldots, a_\ell = y$ be a (uniformly random) shortest path from x to y along edges in $\{-1,1\}^n$. Let $z = a_{\lfloor \ell/2 \rfloor}$ and query $f(z)$.
4. If $f(z) = f(x)$, output $\texttt{BinarySearch}(f, z, y)$. Otherwise, output $\texttt{BinarySearch}(f, x, z)$.

The binary search sub-routine is adaptive, since the value of $f(z)$ determines whether $\texttt{BinarySearch}(f, x, z)$ or $\texttt{BinarySearch}(f, y, z)$ is executed. If, upon sampling $\boldsymbol{x}, \boldsymbol{y} \sim \{-1,1\}^n$, the algorithm finds $f(\boldsymbol{x}) \neq f(\boldsymbol{y})$, then $\texttt{BinarySearch}(f, \boldsymbol{x}, \boldsymbol{y})$ will output a bi-chromatic edge after $O(\log n)$ additional queries. In particular, the query complexity of the adaptive algorithm is $O(1/\mathbf{Var}(f) + \log n)$,[20] which is never larger than $O(n/I_f)$ and may be significantly smaller.

Unsurprisingly, binary search sub-routines are extremely useful for property testing (for example, [33, 89, 53, 28, 99, 45, 50]). However, basic questions about "binary search" beyond (trivial) observations become extremely hard to answer. The difficulty seems to stem from the richness in structure of Boolean functions, and the adaptive nature of binary search. In this thesis, we study property testing algorithms for three well-studied properties: k-juntas, monotonicity, and unateness. We explore the power of adaptivity in the context of these three properties, and in doing so, prove new upper and lower bounds on the query complexity of testing algorithms.

[20] We denote the *variance* of the Boolean function, as $\mathbf{Var}(f) = \mathbf{E}_{\boldsymbol{x},\boldsymbol{y}\sim\{-1,1\}^n}[(f(\boldsymbol{x}) - f(\boldsymbol{y}))^2]$, and notice that $\mathbf{Var}(f)$ is proportional (since the functions are Boolean) to the probability that two random inputs are unequal.

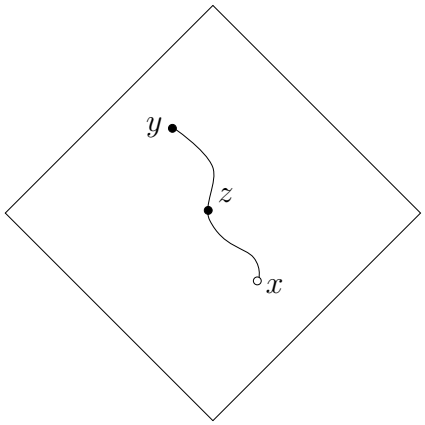

Figure 0.2: Pictorial representation of one step of the binary search strategy for finding bi-chromatic edges. The hypercube $\{-1,1\}^n$ is represented as a diamond, corresponding to the shape of the Hasse diagram of the partial order on $\{-1,1\}^n$; the bottom-most point is the all -1's point, and the top point is the all 1's point. Points x and y are given with $f(x) = 0$ and $f(y) = 1$ and a particular path represents flipping variables where x and y differ one at a time. Finally z lies in the middle of a shortest path between x and y; in this case, $f(z) = 1$, so $\texttt{BinarySearch}(f, x, y)$ recurses down $\texttt{BinarySearch}(f, x, z)$.

k-Juntas

Definition 5 (k-Juntas)**.** *For $n, k \in \mathbb{N}$, a Boolean function $f\colon \{-1,1\}^n \to \{-1,1\}$ is a k-junta if there exists k indices $i_1, \ldots, i_k \in [n]$ and a function $g\colon \{-1,1\}^k \to \{-1,1\}$ where $f(x_1, \ldots, x_n) = g(x_{i_1}, \ldots, x_{i_k})$ for every $x \in \{-1,1\}^n$.*

The study of computational aspects of k-juntas is especially motivated by the *feature selection* problem in machine learning (see e.g. [36, 37, 76, 73, 98, 46]). The n bits of an input $x \in \{-1,1\}^n$ specify the presence or absence of certain features.[21] A k-junta $f\colon \{-1,1\}^n \to \{-1,1\}$ specifies a concept where, when $n \gg k$, contains a large number (in particular, at least $n - k$) of irrelevant features/variables [36, 37, 108, 132]. In practice, filtering out irrelevant features can increase accuracy of certain learned models, and assist in interpretability. Definition 5 specifies a notion of relevance and irrelevance for Boolean functions on $\{-1,1\}^n$: a variable $i \in [n]$ is relevant if there exists a bi-chromatic edge $(x, x^{(i)})$.

The main result of Fischer, Kindler, Ron, Safra, and Samorodnitsky [64], who first

[21] For example, we interpret $x \in \{-1,1\}^n$ with $x_i = 1$ as x containing attribute i and $x_i = -1$ as x lacking attribute i.

studied k-juntas from the property testing perspective, is a $\tilde{O}(k^2)/\varepsilon$-query testing algorithm, showing that the complexity of testing is independent of the size of the input f. Since [64], subsequent works sought the optimal dependence on k and ε. Chockler and Gutfreund [57] showed that any tester for k-juntas requires $\Omega(k)$ queries (for a constant ε), and that result was (recently) improved to $\Omega(k \log k)$ by Saglam [126]. We highlight two works of Blais [32, 33] which gave new algorithms for testing k-juntas in the adaptive and non-adaptive settings.

Theorem 8 (Blais's Adaptive Junta Tester [33])**.** *There is an $O(k \log k + k/\varepsilon)$-query, adaptive algorithm with the following property: Given $\varepsilon > 0$ and query access to an unknown Boolean function $f\colon \{-1,1\}^n \to \{-1,1\}$, it always accepts when f is a k-junta and rejects with probability at least $2/3$ when f is ε-far from a k-junta.*

Theorem 9 (Blais's Non-adaptive Junta Tester [32])**.** *There is an $\tilde{O}(k^{3/2})/\varepsilon$-query, non-adaptive algorithm with the following property: Given $\varepsilon > 0$ and query access to an unknown Boolean function $f\colon \{-1,1\}^n \to \{-1,1\}$, it accepts with probability at least $2/3$ when f is a k-junta and rejects with probability at least $2/3$ when f is ε-far from k-junta.*

We note that Theorem 8 is optimal.[22] The algorithm behind Theorem 8 is intuitive; it proceeds by randomly partitioning the variables $[n]$ into $\mathrm{poly}(k/\varepsilon)$ parts, and finding relevant parts with a binary search strategy. The novelty of [33] lies in the analysis. The existence of a non-adaptive subroutine which would close the query complexity gap between Theorem 8 and Theorem 9 remained open for some time.

Our Contributions to Junta Testing We show a $\tilde{\Omega}(k^{3/2}/\varepsilon)$ lower bound for non-adaptive algorithms, matching the algorithm of [32] and settling the question up to poly-logarithmic factors [56].

Theorem 10 (Non-adaptive Lower Bound for Testing k-Juntas)**.** *Let $\alpha \in (0.5, 1)$ be a fixed constant, and let $k = k(n)\colon \mathbb{N} \to \mathbb{N}$ and $\varepsilon = \varepsilon(n)\colon \mathbb{N} \to \mathbb{R}_{\geq 0}$ be any functions where $k(n) \leq \alpha n$ and $2^{-n} \leq \varepsilon(n) \leq 1/6$ for all sufficiently large n. Then, any non-adaptive*

[22] While the title to Blais's paper, published in 2009, was "Testing juntas nearly optimally", the recent improvement of the lower bound by Saglam [126] shows that it is optimal up to constant factors.

algorithm testing whether a Boolean function $f\colon \{-1,1\}^n \to \{-1,1\}$ is a k-junta or ε-far from a k-junta makes $\tilde{\Omega}(k^{3/2}/\varepsilon)$ queries.

The technique for proving Theorem 10 was suprisingly general, and in particular, also used for proving the query-complexity lower bounds for monotonicity and unateness testing that we will see in the next sections.[23] The main idea is to design two distributions over Boolean functions (one supported on functions with the property, and one supported on functions far from the property) by sampling a random partition of $\{-1,1\}^n$ and placing a "hardness gadget" in each part. The proof of the lower bound then proceeds in two steps. First, we show that the random partition imposes serious restrictions on testing algorithms. For example, we will show that queries falling within the same part must be close in Hamming distance.[24] Then, we prove that algorithms with these restrictions cannot determine whether or not the function came from one distribution or the other, i.e., whether the function has the property or is ε-far from it. The challenge is designing the random partition and hardness gadgets giving rise to two distributions; at the same time, we want the random partition to place the most serious restrictions on algorithms, and the hardness gadget to require many queries before it reveals useful information to the algorithm.

Monotonicity

Definition 6 (Monotonicity)**.** *For $n \in \mathbb{N}$, a Boolean function $f\colon \{-1,1\}^n \to \{-1,1\}$ is monotone if every pair $x, y \in \{-1,1\}^n$ where $x \preceq y$ satisfies $f(x) \leq f(y)$.*[25]

One may interpret a Boolean function $f\colon \{-1,1\}^n \to \{-1,1\}$ as a subset $A_f = \{x \in \{-1,1\}^n : f(x) = 1\} \subset \{-1,1\}^n$. A monotone Boolean function is one where for any $x \in A_f$, any shortest path between the all 1's point and x is contained in A_f (see the left-hand side of Figure 0.3). Studying monotone Boolean functions from the property testing perspective was first considered by Goldreich, Goldwasser, Lehman, Ron

[23]The same technique for proving lower bounds is also implicit in [28, 96, 117] as well.

[24]For non-adaptive algorithms, the argument may by considering two queries $x_1, x_2 \in \{-1,1\}^n$ which are far in Hamming distance and showing that the random partition places x_1 and x_2 in different parts with high probability.

[25]The symbol $\preceq$ specifies the natural poset on $\{-1,1\}^n$, $x \preceq y$ if and only if $x_i \leq y_i$ for all $i \in [n]$.

and Samorodnitsky [70], and since has been intensely studied (see Chapter 4 of [66] for an overview and references therein).

Monotonicity testing provides the context for studying structural properties of the partial order $\{-1,1\}^n$, and this perspective has uncovered a surprising amount of combinatorial structure. For example, the analysis of the "edge tester" [70, 61], which led to the first $O(n/\varepsilon)$-query algorithm, is the perfect example of a *directed* isoperimetric result.[26] Formally, it states that the distance from f to monotonicity, scaled by 2^n, is a lower bound for the number of hypercube edges (x, y) where $x \preceq y$ and $f(x) > f(y)$.[27] We will refer to such edges as decreasing edges, or *violating* edges, as these form a certificate that the function f is non-monotone. Chakrabarty and Seshadhri [43] gave the first improvement to the "edge tester," cementing the analogy to (undirected) isoperimetric inequalities with an algorithm of query complexity $\tilde{O}(n^{7/8}/\varepsilon^{3/2})$; the algorithm was improved to $\tilde{O}(n^{5/6}/\varepsilon^4)$ by Chen, Servedio and Tan [49], and most recently to $\tilde{O}(\sqrt{n}/\varepsilon^2)$ by Khot, Minzer and Safra [88]. This latest development of [88] gave a directed analogue of an inequality of Talagrand [131], and proved the following elegant theorem, giving a "path tester", which readily implies the $\tilde{O}(\sqrt{n}/\varepsilon^2)$-query upper bound.

Theorem 11 (Khot, Minzer, and Safra's Analysis of the Path Tester [88])**.** *For $n \in \mathbb{N}$, let $\mathcal{D}$ be the distribution, supported on pairs of points $(\boldsymbol{x}, \boldsymbol{y})$ in $\{-1,1\}^n$ given by the following procedure:*

1. *Sample $\boldsymbol{x} \sim \{-1,1\}^n$ uniformly at random, and sample $t \sim \{0, \ldots, \lceil \log_2 n \rceil / 2\}$ uniformly at random.*
2. *Sample $\boldsymbol{y}$ by picking a random subset $\mathbf{S} \subset [n]$ of 2^t variables, and let*

$$\boldsymbol{y}_i = \begin{cases} 1 & i \in S \\ \boldsymbol{x}_i & i \notin S \end{cases}.$$

[26] As far as we can tell, the connection is made in hindsight and attributed to the work of Chakrabarty and Seshadhri [43] in [88].

[27] The first algorithm proceeds by sampling $O(n/\varepsilon)$ uniformly random edges of the hypercube and querying the end points. Since the total number of edges is $n2^n$, and $\varepsilon 2^n$ are decreasing when f is ε-far from monotone, $O(n/\varepsilon)$ queries suffice for an algorithm which succeeds with high constant probability. Furthermore, this algorithm is non-adaptive and makes one-sided error.

Then, if f is ε-far from monotone,

$$\Pr_{(\boldsymbol{x},\boldsymbol{y})\sim\mathcal{D}}[f(\boldsymbol{x}) > f(\boldsymbol{y})] \geq \tilde{\Omega}(\varepsilon^2/\sqrt{n}).$$

The corresponding "path tester" is non-adaptive and always outputs `accept` when f is monotone. With these restrictions, Theorem 11 matches a query-complexity lower bound of $\Omega(\sqrt{n})$ (for constant ε) [63]. The techniques developed in two papers [49, 54] showed $\Omega(n^{1/2-\delta})$-queries are needed for non-adaptive algorithms, for any constant $\delta > 0$. Whether adaptive algorithms may improve on the query complexity of the path tester was (and still is) an enticing open question. An approach is to appeal to the (extremely effective and efficient) binary search procedure. Analytically, we want to answer the following question: when f is ε-far from monotone, is there a lower bound on the probability that (some variation of) binary search outputs a violating edge? Until the work of Belovs and Blais [34] not even a super-logarithmic query-complexity lower bound on adaptive algorithms was known.

Theorem 12 (Belov and Blais's Adaptive Monotonicity Testing Lower Bound)**.** *Let $\varepsilon > 0$ be a fixed constant. Any algorithm which tests whether an unknown Boolean function $f\colon \{-1,1\}^n \to \{-1,1\}$ is monotone or ε-far from monotone makes $\Omega(n^{1/4})$ queries.*

Our Contributions to Monotonicity Testing Here, we improve the adaptive lower bound to $\tilde{\Omega}(n^{1/3})$ and the non-adaptive lower bound to $\tilde{\Omega}(\sqrt{n})$ [51].

Theorem 13 (Adaptive Monotonicity Testing Lower Bound)**.** *There exists a fixed constant $\varepsilon > 0$ such that any algorithm which tests whether an unknown Boolean function $f\colon \{-1,1\}^n \to \{-1,1\}$ is monotone or ε-far from monotone makes $\tilde{\Omega}(n^{1/3})$ queries.*

Theorem 14 (Non-adaptive Monotonicity Testing Lower Bound)**.** *There exists a fixed constant $\varepsilon > 0$ such that any non-adaptive algorithm which tests whether an unknown Boolean function $f\colon \{-1,1\}^n \to \{-1,1\}$ is monotone or ε-far from monotone makes $\tilde{\Omega}(\sqrt{n})$ queries.*

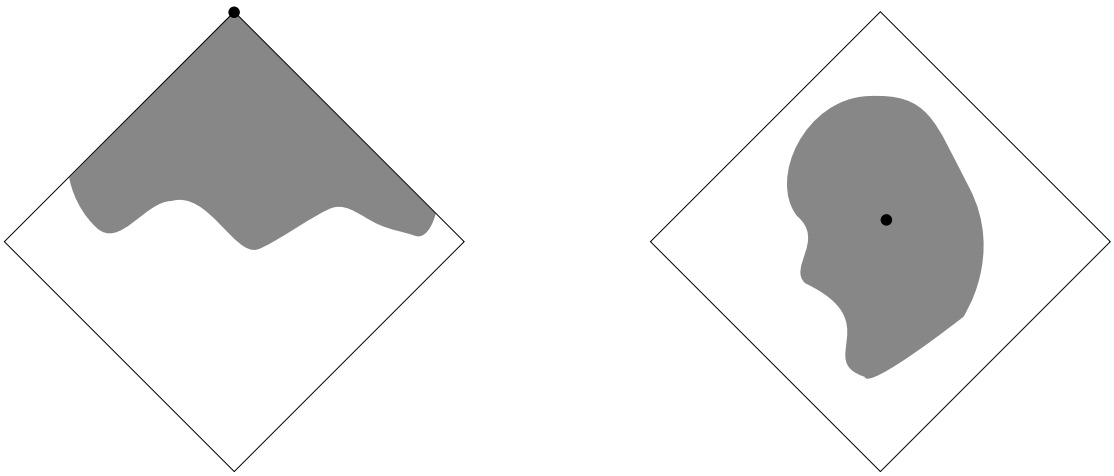

Figure 0.3: Pictorial representation of a monotone Boolean function (on the left-hand side), and a unate Boolean function (on the right-hand side). The hypercube $\{-1,1\}^n$ is represented as a diamond, corresponding to the shape of the Hasse diagram of the partial order on $\{-1,1\}^n$; the bottom-most point is the all -1's point, and the top point is the all 1's point. The functions have the regions $A_f = \{x \in \{-1,1\}^n : f(x) = 1\}$ as shaded gray. When $f\colon \{-1,1\}^n \to \{-1,1\}$ is monotone, any shortest path between any point in A_f and the all 1's point along edges of $\{-1,1\}^n$ is fully contained in A_f. When f is unate, there exists a point $r \in \{-1,1\}^n$ such that $f(x \cdot r)$ is monotone, i.e., any shortest path between any point in A_f and r is fully contained in A_f.

Unateness

The problem of testing unateness, first studied in [70], has also attracted significant attention [89, 44, 24, 23]. A Boolean function f is unate if every variable is either monotone non-decreasing, or monotone non-increasing. Equivalently, a Boolean function $f\colon \{-1,1\}^n \to \{-1,1\}$, specifying the set $A_f = \{x \in \{-1,1\}^n : f(x) = 1\} \subset \{-1,1\}^n$,[28] is unate iff there exists a point $r \in A_f$ such that every shortest path between r and a point $x \in A_f$ is contained in A_f (see the right-hand side of Figure 0.3); formally, we have the following definition.

Definition 7 (Unateness)**.** *For $n \in \mathbb{N}$, a Boolean function $f\colon \{-1,1\}^n \to \{-1,1\}$ is unate if there exists $r \in \{-1,1\}^n$ so that $g\colon \{-1,1\}^n \to \{-1,1\}$ given by $g(x) = f(x \cdot r)$ is monotone.*

Similarly to the case of monotonicity testing, there is a non-adaptive "edge tester" with

[28] We are implicitly assuming here that A_f is non-empty.

query complexity $\tilde{O}(n/\varepsilon)$ [89, 43, 24, 23].[29] These non-adaptive algorithms turned out to be optimal [51, 22] for non-adaptive algorithms which always accept unate functions. Using the techniques developed for proving Theorem 11, we gave an $\tilde{O}(n^{3/4}/\varepsilon^2)$-query adaptive algorithm in [52].

Our Contributions to Unateness Testing Here, we improve on the $\tilde{O}(n^{3/4}/\varepsilon^2)$-query tester. The main result is an (nearly optimal) algorithm which tests unateness with query complexity $\tilde{O}(n^{2/3}/\varepsilon^2)$. The main conceptual contribution is an adaptive preprocessing stage which efficiently identifies a subset of variables containing many violations to unateness. The preprocessing stage is based on, unsurprisingly, a novel analysis of a binary search procedure. Whether an analogous preprocessing stage, making $o(\sqrt{n})$-queries, exists for monotonicity testing is a tantalizing open problem. Such a procedure has the potential to close the gap between the $\tilde{O}(\sqrt{n})$-query upper bound of [88] and the $\tilde{\Omega}(n^{1/3})$-query lower bound of Theorem 13 [51].

Theorem 15 (Adaptive Unateness Tester)**.** *There is an $\tilde{O}(n^{2/3}/\varepsilon^2)$-query, adaptive algorithm with the following property: Given $\varepsilon > 0$ and query access to an unknown Boolean function $f\colon \{-1,1\}^n \to \{-1,1\}$, it always accepts when f is unate, and rejects with probability at least $2/3$ when f is ε-far from unate.*

Theorem 15 has optimal dependence on n up to poly-logarithmic factors, as we show a matching lower bound.

Theorem 16 (Adaptive Unateness Testing Lower Bound)**.** *There exists a fixed constant $\varepsilon > 0$ such that any algorithm which tests whether an unknown Boolean function $f\colon \{-1,1\}^n \to \{-1,1\}$ is unate or ε-far from unate makes $\tilde{\Omega}(n^{2/3})$ queries.*

[29] We note that the unateness edge tester from [23] is more involved. It utilizes Levin's work investment strategy to carefully choose how many random edges to sample in each direction. A simple edge tester, which only samples uniformly random edges, is analyzed in [70] and makes $O(n^{3/2}/\varepsilon)$ queries.

Part I

Approximate Nearest Neighbor Search in General Metric Spaces

Chapter 1

Overview of the Results

The results in approximate nearest neighbor search presented in this thesis [12, 14, 16] build on each other, and hence, we use this chapter to give a broad overview of these results. While some technical arguments are presented, the purpose here is not to present formal proofs. Later chapters elaborate on the arguments and formal proofs are given.

1.1 The Embedding's Approach and ANN for Symmetric Norms

The first direction toward Problem 1 is understanding the power and limitations of metric embeddings. These essentially function as *reductions* between metric spaces. To design a data structure for a desired metric space, one identifies a target metric space which is both algorithmically tractable and sufficiently expressive; i.e., the target space admits an efficient data structure (for instance ℓ_1^d as in Theorem 1), and there is an embedding from the desired metric space into the target space which approximately preserves distances. The resulting data structure proceeds by first applying the embedding, and then using the data structure in the target metric space. The following definition, formally stating what a (bi-Lipschitz) embedding is, as well as its use in the subsequent theorem encapsulates the "embeddings approach."

Definition 8 (Bi-Lipschitz Metric Embedding)**.** *Let (X, d_X) and (Y, d_Y) be two metric spaces. For $c \geq 1$, a function $f : X \to Y$ is an embedding of X into Y with distortion c if there exists a threshold $\tau > 0$ such that for every $a, b \in X$,*

$$d_X(a, b) \leq \tau \cdot d_Y(f(a), f(b)) \leq c \cdot d_X(a, b).$$

Theorem 17 (The Embeddings Approach)**.** *Let (X, d_X) and (Y, d_Y) be two metric spaces. Suppose that:*

- *$f\colon X \to Y$ is an embedding of X into Y of distortion c_0, and there exists an algorithm $\texttt{ALG}_f$ which given as input a point $a \in X$, outputs $f(a)$ in time $T(f)$.*
- *For every $n \in \mathbb{N}$ and $c > 1$, there exists a data structure for c-ANN over Y with query time $q(Y, n, c)$ and space $s(Y, n, c)$.*

Then, there for every $n \in \mathbb{N}$ there exists a data structure for $(c_0 c)$-ANN over X with query time $T(f) + q(Y, n, c)$ and space $s(Y, n, c)$.

This very elegant and versatile approach already gives a wealth of ANN data structures over various metric spaces. First and foremost, it implies efficient $c(1+\varepsilon)$-ANN data structures for ℓ_p^d for $p \in [1, 2]$ [83].[1] Common target metric spaces for this approach (which are substituted into (Y, d_Y) in Theorem 17) are ℓ_1^d, ℓ_2^d [79, 93], and ℓ_∞^d [78]. Successes include a $\mathrm{poly}(\log\log d)$-approximation for the Ulam metric [8], a $O(\log d)$-approximation for EMD [47, 81], a $2^{\widetilde{O}(\sqrt{\log d})}$-approximation for edit distance [116], and a $\mathrm{poly}(\log d)$-approximation for Frechét distance [77]. Generally, combining Theorem 17 with John's theorem (stated next) gives an ANN data structure for *any d-dimensional normed space* with approximation $O(\sqrt{d})$.

Theorem 18 (John's Theorem [82, 25])**.** *Let $(\mathbb{R}^d, \|\cdot\|_X)$ be any d-dimensional normed space. There exists a linear map*[2] *$f\colon \mathbb{R}^d \to \mathbb{R}^d$ such that for every $a \in \mathbb{R}^d$,*

$$\|a\|_X \le \|f(a)\|_2 \le \sqrt{d} \cdot \|a\|_X.$$

One may further weaken the notion of embeddings to allow for randomization. The idea is to relax the condition that all pairwise distances are approximately preserved, and instead require that the probability of the near neighbor be preserved is high enough for the reduction

[1] The reduction to Theorem 1 proceeds by defining a distribution over linear maps $\mathbb{R}^d \to \mathbb{R}^{O(d\log(1/\varepsilon)/\varepsilon^2)}$ such that a random map from this distribution is an embedding of ℓ_p to ℓ_1 with distortion $(1+\varepsilon)$ with high probability. For $\ell_2 \to \ell_1$, matrix multiplication by a scaled Gaussian matrix gives the desired embedding.

[2] Notice that since the function is linear and $d_X(a, b) = \|a - b\|_X$ in a normed space, an upper and lower bound on the norm of each vector implies Definition 8. Furthermore, computing f is done via a matrix-vector multiplication.

to ANN over the target space to succeed. In fact, the following definition implies the *nearest neighbor preserving embeddings* of [80] and will give $O(\log \log d)$-approximation for all ℓ_p-norms [5].

Definition 9. *Let (X, d_X) and (Y, d_Y) be two metric spaces. For $r \geq 0$, $c \geq 1$, and $\alpha, \beta > 0$, a distribution $\mathcal{D}$ supported on functions $f \colon X \to Y$ (α, β)-preserves approximate near neighbors*[3]*, if there exists thresholds $0 < \tau_1 < \tau_2$ such that for every $a, b \in X$:*

- *If $d_X(a, b) \leq r$, then $\mathbf{Pr}_{\boldsymbol{f} \sim \mathcal{D}}[d_Y(\boldsymbol{f}(a), \boldsymbol{f}(b)) \leq \tau_1] \geq \alpha$.*
- *If $d_X(a, b) \geq cr$, then $\mathbf{Pr}_{\boldsymbol{f} \sim \mathcal{D}}[d_Y(\boldsymbol{f}(a), \boldsymbol{f}(b)) \leq \tau_2] \leq \beta$.*

Given a data structure for (τ_2/τ_1)-ANN over (Y, d_Y), a c-ANN data structure for (X, d_X) proceeds by, independently for $O(1/(\alpha - \beta n))$ iterations, sampling $\boldsymbol{f} \sim \mathcal{D}$ and applying the (τ_2/τ_1)-ANN data structure for (Y, d_Y) to the data set $\boldsymbol{f}(P) = (\boldsymbol{f}(p_1), \ldots, \boldsymbol{f}(p_n))$. For a query $q \in X$ with near-neighbor $p_i \in P$, following a union bound, at least one of the $O(1/(\alpha - \beta n))$ iterations of $\boldsymbol{f} \sim \mathcal{D}$ will satisfy $d_Y(\boldsymbol{f}(q), \boldsymbol{f}(p_i)) \leq \tau_1$ and every $p_j \in P$ where $d_X(q, p_j) \geq cr$ will satisfy $d_Y(\boldsymbol{f}(q), \boldsymbol{f}(p_j)) \geq \tau_2$ with high constant probability. Once this happens, the corresponding (τ_2/τ_1)-ANN data structure over (Y, d_Y) will find the near neighbor with high constant probability.

I will now sketch how to use randomized embeddings of this form, combined with known results from the literature (specifically, [78, 77, 5]), to obtain a $\mathrm{poly}(\log \log n)$-ANN data structure for any symmetric norm. A formal description of the embedding, as well as the algorithm, is presented in Chapter 2.

Definition 10 (Symmetric Norm)**.** *A normed space $X = (\mathbb{R}^d, \|\cdot\|_X)$ is* symmetric *if $\|\cdot\|_X \colon \mathbb{R}^d \to \mathbb{R}^{\geq 0}$ is invariant under permutation and sign-flips of coordinates. In other words, if $x \in \mathbb{R}^d$ and $x^* \in \mathbb{R}^d$ is given by sorting the coordinates of $|x|$ in non-increasing value, then $\|x\|_X = \|x^*\|_X$.*

The key is to identify a target metric space which is algorithmically tractable (for the doubly-logarithmic approximation we are aiming for) and expressive enough to admit

[3] and in particular, will be used in conjunction with an ANN algorithm for (Y, d_Y) to obtain an ANN algorithm for (X, d_X).

embeddings for all d-dimensional symmetric norms with a small constant distortion. We state one of the main theorems, whose proof appears in Chapter 2.

Theorem 19 (Universal Target for Symmetric Norms)**.** *Fix $d \in \mathbb{N}$ and $\delta \in (0, 1/2)$. There exist $t = t(d, \delta) \leq d^{O(\log(1/\delta)/\delta)}$ and a normed space $Y = (\mathbb{R}^t, \|\cdot\|_Y)$ such that any symmetric norm $X = (\mathbb{R}^d, \|\cdot\|_X)$ embeds into Y via a linear map $f \colon \mathbb{R}^d \to \mathbb{R}^t$ with distortion $1 + \delta$.*

The normed space Y is given by an iterated product space of top-k norms; we will view $x = (x_{i,k})_{i \in [t_1], k \in [d]} \in (\mathbb{R}^d)^{t_1 d}$, so that the parameter t in Theorem 19 is given by $t_1 d^2$, then

$$\|x\|_Y = \max_{i \in [t_1]} \left(\sum_{j=k}^{d} \|x_{i,k}\|_{T(k)} \right),$$

and $T(k) = (\mathbb{R}^d, \|\cdot\|_{T(k)})$ denotes the top-k norm.[4] We abbreviate $Y = \ell_\infty^{t_1}(\ell_1^d(T^{(k)}))$. Conceptually, the existence of such a low-dimensional universal space incuring only constant distortion is surprising. Classical results in convex geometry and metric embeddings, such as John's theorem [82] (cited in Theorem 18) and Frechét's embedding technique [101] show that ℓ_2 and ℓ_∞ admit embeddings of general normed spaces; however, the embeddings either incur distortion which is polynomial in the dimension or suffer an exponential blowup in the dimension of the target space.

The new data structure for the normed space Y uses a combination of novel and known techniques based on randomized embeddings, the data structure of Theorem 2 [78], as well as the data structure for ℓ_p-product spaces [77, 8, 5].

Theorem 20 (Theorem 1 of [77] and Theorem 5.1.2 of [5])**.** *Let $k \in \mathbb{N}$, and suppose $\{(X_i, d_{X_i})\}_{i \in [k]}$ is a collection of metric spaces where each admits a c-ANN data structure using space at most $s(n) \geq n$ and query time at most $q(n)$ for some $c \geq 1$. For any $\delta > 0$, $\varepsilon \in (0, 1)$, and $p \in [1, \infty)$, there exists a data structure for $O(c\varepsilon^{-1} \log_{1+\delta} \log n)$-ANN*

[4] The top-k norm of a point $x \in \mathbb{R}^d$ is given by sum of the highest (in absolute value) k coordinates of x; i.e., $\|x\|_{T(k)} = \sum_{j=1}^{k} x_j^*$ (see Definition 10).

over the ℓ_p-product of X's[5] $(\times_{i=1}^{k} X_i, d_{\ell_p^k(X_i)})$ *using space*[6] $O(ks(n)n^{\delta+\varepsilon^p})$*, and query time* $O(n^{\varepsilon^p}) \cdot O(q(n)\log(n) + k\log(n)\max_{j\in[k]}\log|X_k|)$.

The final step is to give a randomized embedding, in the sense of Definition 9, from $(\mathbb{R}^d, \|\cdot\|_{T(k)})$ to ℓ_∞^d which (α, β)-preserves approximate near neighbors with $\beta n \ll \alpha$, for arbitrary $k \in [d]$. Combining the embedding with Theorem 2 gives a data structure for $O(\log\log d)$-ANN for top-k norms. We will then successive apply Theorem 20 to give a $\mathrm{poly}(\log\log n)$-ANN data structure for Y, and thus, for any symmetric norm.

Lemma 1.1.1 (Randomized Embedding of Top-k into ℓ_∞ [12])**.** *Fix $k, d \in \mathbb{N}$, and let $r \geq 0$, $c \geq 1$, and $\varepsilon \in (0, 1/2)$. There exists a distribution $\mathcal{D}$ supported on $\mathbb{R}^{>0}$ such that letting $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_d \sim \mathcal{D}$, the linear map $\boldsymbol{f} = f_{\boldsymbol{u}_1,\ldots,\boldsymbol{u}_d} \colon \mathbb{R}^d \to \mathbb{R}^d$ given by*

$$(x_1, x_2, \ldots, x_d) \overset{\boldsymbol{f}}{\mapsto} \left(\frac{x_1}{\boldsymbol{u}_1}, \frac{x_2}{\boldsymbol{u}_2}, \ldots, \frac{x_d}{\boldsymbol{u}_d}\right)$$

is a randomized embedding between $(\mathbb{R}^d, \|\cdot\|_{T^{(k)}})$ and ℓ_∞^d which $(1/n^\varepsilon, 1/n^2)$-preserves approximate near neighbors with thresholds $\tau_1 = r$ and $\tau_2 = 3cr \cdot \varepsilon$.

The proof of Lemma 1.1.1 follows from a more general construction of randomized embeddings from any d-dimensional Orlicz norm into ℓ_∞^d. Specifically, for a convex function $G \colon \mathbb{R}^{\geq 0} \to \mathbb{R}^{\geq 0}$ with $G(0) = 0$ defining the Orlicz norm $(\mathbb{R}^d, \|\cdot\|_G)$[7], the distributions $\mathcal{D}$ over $\mathbb{R}^{>0}$, defining the scalings in the randomized embedding, is one where $\mathbf{Pr}_{\boldsymbol{u}\sim\mathcal{D}}[|t/\boldsymbol{u}| \leq \tau_1] = \alpha^{-G(|t|/r)}$. Hence, the condition that $\|\boldsymbol{f}(x)\|_\infty \leq \tau_1$ translates (by independence of the draws to $\mathcal{D}$) to a product of probabilities, where each term in the product is $\mathbf{Pr}_{\boldsymbol{u}_i\sim\mathcal{D}}[|x_i/\boldsymbol{u}| \leq \tau_1]$. The definition of $\mathcal{D}$ then implies, when $\|x\|_G \leq r$, that the product is at least α.

---

[5]Given a collection of metric spaces $\{(X_i, d_{X_i})\}_{i\in[k]}$ and $p \in [1, \infty]$, the ℓ_p-product of X's the metric space defined over the cartesian product $\times_{i=1}^k X_i$, where distances between points $(a_1, \ldots, a_k), (b_1, \ldots, b_k)$ is given by $d_{\ell_p^k(X_i)}(a, b) = (\sum_{i=1}^k d_{X_i}(a_i, b_i)^p)^{1/p}$.

[6]We remark that [77], which first obtained this Theorem for $p = \infty$, states a weaker bound in the space complexity. [8] noticed a better analysis gave the corresponding space bound. See also, Appendix A of [12].

[7]The Orlicz norm over $\mathbb{R}^d$ specified by the function G is given by $\|x\|_G = \inf\{r \in (0, \infty) : \sum_{i=1}^d G(|x_i|/r) \leq 1\}$.

Theorem 21 (Symmetric Norm Data Structure [12])**.** *Fix $d \in \mathbb{N}$ and $\varepsilon \in (0, 1/2)$, and let $X = (\mathbb{R}^d, \|\cdot\|_X)$ be a symmetric norm. There exists a data structure for $O(\log^2 \log(n) \log\log d/\varepsilon^5)$-ANN over X using space $d^{O(1)} \cdot O(n^{1+\varepsilon})$ and query time $d^{O(1)} \cdot O(n^{\varepsilon})$.*

We remark that the "embeddings approach" outlined in this section has its limits. Indyk suggested the proof of the following theorem (formally proven in Chapter 2), which shows that one cannot hope for a universal target space (in the sense of Theorem 19) for general norms with a polynomial blowup in the dimension improving on Theorem 18.

Theorem 22 ([12])**.** *Fix $\varepsilon \in (0, 1/2)$ and $C > 1$. Suppose that for any $d \in \mathbb{N}$, there exists $d' \in \mathbb{N}$ and a normed space $Y = (\mathbb{R}^{d'}, \|\cdot\|_Y)$ such that for any normed space $X = (\mathbb{R}^d, \|\cdot\|_X)$, there exists a distribution $\mathcal{D}$ supported on linear maps $\mathbb{R}^d \to \mathbb{R}^{d'}$ where every $x \in \mathbb{R}^d$ satisfies*

$$\Pr_{\boldsymbol{f} \sim \mathcal{D}} \left[\|x\|_X \le \|\boldsymbol{f}(x)\|_Y \le C \cdot d^{1/2-\varepsilon} \|x\|_X \right] \ge \frac{2}{3}.$$

Then, $d' = \exp\left(\Omega(d^{2\varepsilon})\right)$.

1.2 A Lower Bound in the List-of-Points Model

This section explores Problem 1 from a lower bounds perspective. We construct a class of metric spaces, the shortest path metrics on constant-degree expander graphs, which, for large classes of data structures, do not admit small approximations for ANN. The restriction to "large classes of data structures" is necessary; current techniques for proving (general) data structure lower bounds are far-from distinguishing the nuances between metric spaces.[8] Rather, we formalize the *list-of-points* framework. This algorithmic framework encapsulates *hashing-based* approaches pioneered in [79], and the lower bounds proved will inform the upper bounds to come.

[8]The best cell-probe lower bounds for ANN appear in [118]. Here, the lower bounds become trivial once the query time exceeds $O(\log n)$.

Definition 11. *Let $G = (V, E)$ be a connected, undirected graph. The* shortest path metric *on G is the metric space (V, d_G) given by letting $d_G(u, v)$ be the length of the shortest path between u and v in G.*

We consider an arbitrary family $\mathcal{G} = \{G_N = ([N], E_N)\}_{N \in \mathbb{N}}$ of 3-regular expanders, i.e., there exists constants $\eta \in (0, 1)$ and $N_0 \in \mathbb{N}$ such that all $N > N_0$ satisfy $\lambda_2(\mathcal{L}_{G_N}) \geq \eta$.[9] We consider the family $\mathcal{M} = \{([N], d_{G_N})\}_{N \in \mathbb{N}}$ of shortest path metrics on $\mathcal{G}$. For $N \in \mathbb{N}$, we seek efficient data structures for ANN over $([N], d_{G_N})$ when n, the number of dataset points, satisfies

$$\omega(\log n) \leq \log N \leq n^{o(1)}, \tag{1.1}$$

since this choice of parameters aligns well with the notion of efficiency for Definition 2.[10] Here, we want to prove a *curse of dimensionality* for metric spaces in $\mathcal{M}$: for any $\delta \in (0, 1)$ there are infinitely many $N \in \mathbb{N}$, such that any data structure for $(\log N)^{1-\delta}$-ANN over $([N], d_{G_N})$ for n dataset points must have time complexity $\Omega(n)$ or space complexity scaling exponentially in $(\log N)^{\delta}$. The approximation factor of $(\log N)^{1-\delta}$ for any $\delta \in (0, 1)$ would be best possible, due to the embedding theorem of Bourgain [39].

Theorem 23 (Bourgain's Embedding [39, 97])**.** *For any metric space (X, d_X), there exists an embedding $f \colon X \to \ell_2^{O(\log N)}$ with distortion $O(\log N)$.*

The above theorem, together with the "embeddings approach" of Theorem 17, lays a path toward data structures for $O(\log N)$-ANN over $([N], d_{G_N})$. Time efficiency does *not necessarily* follow because the function f may take $N^{\Omega(1)}$ time to compute.[11] However, Theorem 23 does imply an efficient cell-probe data structure, and leaves us hopeless for unconditional lower bounds except for dramatic advancements in complexity theory.

[9] Here, I am letting $\lambda_2(\mathcal{L}_{G_N})$ be the second smallest eigenvalue of the normalized Laplacian of the graph G_N.

[10] Recall that the metric analogue of dimension will be $\log |X|$ for a metric space (X, d_X).

[11] In particular, the time complexity of computing f will crucially depend on the oracle access granted to the metric space $([N], d_{G_N})$. In the *black-box model*, where a metric space (X, d_X) is accessed only via black-box evaluations of $d(\cdot, \cdot)$, computing even a single coordinate of f may be too costly. See [91] for a more thorough treatment of the black-box model for ANN.

The hardness of ANN over metric spaces alike $\mathcal{M}$ is unsurprising. A growing body of work, establishing non-embedability results for finite metric spaces, utilize the shortest path on constant degree expanders as the hard instance. These lower bounds preclude efficient data structures for ANN which proceed via low-distortion embeddings into low-dimensional tractable spaces. For example, the work of London, Linial, and Rabinovich [97] shows that embedding $([N], d_{G_N})$ into ℓ_1 requires $\Omega(\log N)$ distortion, matching the upper bound on the distortion from Theorem 23[12] (see also Section 15.5 in [102]). More generally, there are strong lower bounds on the required distortion[13] for embedding $\mathcal{M}$ into arbitrary low-dimensional normed spaces [110, 111]. The arguments are formalized with the notion of *non-linear spectral gaps*, first studied by [105], we give a proof of the subsequent theorem in Chapter 3.

Definition 12 (Non-Linear Spectral Gaps)**.** *Let (X, d_X) be a metric space and for $n \in \mathbb{N}$, μ be a symmetric probability measure supported on $[n] \times [n]$ with marginal distribution ρ. For $p > 0$, the inverse of the* non-linear spectral gap *$\gamma(\mu, d_X^p)$ is the infimum over $\gamma > 0$ such that for any $u_1, \ldots, u_n \in X$,*

$$\mathop{\mathbf{E}}_{i,j\sim\rho}\left[d_X(u_i, u_j)^p\right] \leq \gamma \mathop{\mathbf{E}}_{(i,j)\sim\mu}\left[d_X(u_i, u_j)^p\right].$$

Theorem 24. *Let $d \in \mathbb{N}$ and consider any normed space $X = (\mathbb{R}^d, \|\cdot\|_X)$. For $n \in \mathbb{N}$, and a symmetric probability measure μ over $[n] \times [n]$ with marginal ρ,*

$$\gamma(\mu, \|\cdot\|_X^2) \lesssim \left(\frac{\log d}{\lambda_2(\mathcal{L}_\mu)}\right)^2,$$

where $\lambda_2(\mathcal{L}_\mu)$ is the second smallest eigenvalue of the normalized Laplacian matrix $\mathcal{L}_\mu = I_n - \mathrm{diag}(\rho)^{-1/2}\mu\mathrm{diag}(\rho)^{-1/2}$.

For the metric space $([N], d_{G_N}) \in \mathcal{M}$, consider the distribution μ which samples a

[12]Even though the upper bound in Theorem 23 holds for embedding into ℓ_2, the embedding into ℓ_1 follows from concatenating Theorem 23 with the $O(1)$-distortion embedding of $\ell_2 \to \ell_1$.

[13]even average distortion

uniformly random edge of G_N, and notice that since G_N is 3-regular, $\gamma(\mu, d^2_{G_N}) \gtrsim \log^2 N$.[14] If for a normed space $X = (\mathbb{R}^d, \|\cdot\|_X)$, there existed an embedding $f\colon ([N], d_{G_N}) \to (\mathbb{R}^d, \|\cdot\|_X)$ with distortion c, then $\gamma(\mu, \|\cdot\|^2_X) \gtrsim \log^2 N/c^2$; however, Theorem 24 implies $\log^2 N/c^2 \lesssim \log^2 d$, since $\lambda_2(\mu) = \Omega(1)$, and thus $c = \Omega(\log N/\log d)$. This latter result gives a curse of dimensionality for data structures which proceed via embeddings into an arbitrary d-dimensional normed space: any $(\log N)^{1-\delta}$-ANN data structure implied by an embedding into a normed space with Theorem 17 will have time and space complexity exponential in $(\log N)^{\delta}$. (As well will see, Theorem 24 will be useful for designing data structures as well!)

Definition 13 (List-of-Points Framework [18])**.** *For a metric space (X, d_X), $n \in \mathbb{N}$ and $r \geq 0$, a data structure for c-ANN over X in the* list-of-points *framework is specified as follows:*

- *There are two parameters $\ell = \ell(n) \in \mathbb{N}$ and a distribution $\mathcal{H}$ supported on a list of ℓ pairs of subsets of X, i.e., a sample $\{(\mathbf{A}_1, \mathbf{B}_1), \ldots, (\mathbf{A}_\ell, \mathbf{B}_\ell)\} \sim \mathcal{H}$ satisfies $\mathbf{A}_i, \mathbf{B}_i \subset X$.*
- *The function* `Preprocess`*$\colon X^n \times \{0,1\}^m \to (\{0,1\}^w)^s$, given $P \in X^n$ and random bits, constructs $D \in (\{0,1\}^w)^s$ by:*
  1. *Using the random bits to sample $\{(\mathbf{A}_1, \mathbf{B}_1), \ldots, (\mathbf{A}_\ell, \mathbf{B}_\ell)\} \sim \mathcal{H}$,*
  2. *D is divided into three parts; the first part encodes the lists $\{(\mathbf{A}_1, \mathbf{B}_1), \ldots (\mathbf{A}_\ell, \mathbf{B}_\ell)\}$;*[15] *the second part uses $n \cdot \lceil \log |X|/w \rceil$ words to store the dataset P in order; the third part is divided into ℓ blocks. For $i \in [\ell]$, we denote D_i as the ith block.*
  3. *For each $i \in [\ell]$, D_i stores indices of points from P inside $\mathbf{A}_i$, using at most one word to store each index.*[16]
- *The function* `Query`*$\colon X \times \{0,1\}^m \to X$, when given a query $q \in X$ proceeds by:*

[14] In particular, letting $n = N$ and $u_i = i$ in Definition 12, we notice that $d_{G_N}(\boldsymbol{i}, \boldsymbol{j}) = 1$ for every $(\boldsymbol{i}, \boldsymbol{j}) \sim \mu$ and $d_{G_N}(\boldsymbol{i}, \boldsymbol{j})^2 \gtrsim \log^2 N$ with probability approaching 1 as $N \to \infty$ (1.2), which lower bounds $\gamma(\mu, d^2_{G_N}) \gtrsim \log^2 N$.

[15] for example, we may succinctly encode this by storing the random bits

[16] Recall that $w = \Theta(\log n)$, so that one word is enough to an element of $[n]$.

1. *Reading the encoding of the list* $\{(\mathbf{A}_1, \mathbf{B}_1), \ldots, (\mathbf{A}_\ell, \mathbf{B}_\ell)\}$ *and forming the set* $I(q) = \{i \in [\ell] : q \in \mathbf{B}_i\}$.[17]
2. *For each* $i \in I(q)$*, the algorithm scans* D_i,[18] *and computes, for each index* $j \in D_i$*,* $d_X(q, p_j)$*. If the algorithm finds a point* p_j *where* $d_X(q, p_j) \leq cr$*, it outputs* p_j*.*

Definition 13 captures data structures based on randomized space decompositions. A draw of $\{(\mathbf{A}_1, \mathbf{B}_1), \ldots, (\mathbf{A}_\ell, \mathbf{B}_\ell)\} \sim \mathcal{H}$ gives two such decompositions of X; where $\mathbf{A} = \{\mathbf{A}_1, \ldots, \mathbf{A}_\ell\}$ governs where dataset points are placed, and $\mathbf{B} = \{\mathbf{B}_1, \ldots, \mathbf{B}_\ell\}$ governs where queries search. The setting of *hashing*, á la Locality-Sensitive Hashing (LSH) [79], corresponds to the case $\mathbf{A}_i = \mathbf{B}_i$ for every $i \in [\ell]$. For general data structures, Definition 13 has a glaring weakness: the space decompositions used are completely independent of the dataset. This exempts the preprocessing and query phases from using the dataset P to guide the execution, a major detriment to efficient data structures.[19] Nevertheless, the list-of-points framework will provide a useful abstraction for studying necessary properties of randomized space decompositions for ANN data structures. A formalization of hashing-based data structures, allowing for adaptivity of the hash functions, appears in [11], where the goal there is understanding limits of the data-dependent LSH approach for ℓ_2 [13, 10].

We now define the *robust expansion* of a metric space, a central quantity of a metric space which will dictate the quality of space decompositions. This quantity characterizes the time and space complexity of list-of-points data structures. The robust expansion was first identified in [118] and used to establish cell-probe lower bounds for ANN under various metric spaces, most notably for ℓ_1, ℓ_2, and ℓ_∞ [58, 18, 85].

---

[17] We will ignore subtle issues of encoding of the list $\{(\mathbf{A}_1, \mathbf{B}_1), \ldots, (\mathbf{A}_\ell, \mathbf{B}_\ell)\}$, as well as the time complexity of computing $I(q)$ since this definition will be useful for lower bounds.

[18] there is a subtle issue in finding D_i. In particular, since the blocks may have varying sizes (because storing a different number of points from P), the algorithm will not know where in D the block D_i lies. We note this issue may be safely ignored for lower bounds, and assume the algorithm knows where blocks D_i begin.

[19] For example, when $X = \mathbb{R}$, a data structure which sorts the dataset P and performs a binary search is not captured by Definition 13.

Definition 14 (Robust Expansion [118])**.** *Fix a metric space (X, d_X) and a distribution μ supported on pairs points $X \times X$ with marginal distributions ρ_1 and ρ_2. For $\gamma > 0$ and $\alpha > 0$,*

$$\Phi_\mu(\alpha, \gamma) = \inf \left\{ \frac{\rho_2(B)}{\rho_1(A)} : A, B \subset X, \rho_1(A) \leq \alpha \text{ and } \mathop{\mathbf{Pr}}_{(\boldsymbol{p},\boldsymbol{q})\sim\mu}[\boldsymbol{q} \in B | \boldsymbol{p} \in A] \geq \gamma \right\}.$$

The following lemma formalizes the connection between Definition 14 and lower bounds for data structures in the list-of-points model. It analyzes an appropriate distribution over dataset-query pairs and applies Yao's principle. The lower bound on the space and time complexity follow by a lower bound on the expected time and expected space complexity of the assumed list-of-points data structure with respect to the distribution over inputs. The following lemma appears in [18]. It is a simple exercise in computing the time and space complexity of list-of-points dat structures.

Lemma 1.2.1. *Let (X, d_X) be a metric space, $r \geq 0$ and $n \in \mathbb{N}$. Suppose that $\mathcal{H}$ and ℓ specify a list-of-points data structure for c-ANN over (X, d_X) with distance threshold r and n dataset points. Let μ be a distribution supported on pairs of points $X \times X$ with distance at most r, and let ρ_1 and ρ_2 be the marginal distributions of μ, where*

$$\mathop{\mathbf{Pr}}_{\boldsymbol{p}_1,\boldsymbol{p}_2\sim\rho_1}[d_X(\boldsymbol{p}_1, \boldsymbol{p}_2) < cr] = o\left(\frac{1}{n}\right).$$

Then, there exists $\{(A_i, B_i)\}_{i=1}^{\ell}$ in the support of $\mathcal{H}$ such that for every $i \in [\ell]$, $\alpha_i = \rho_1(A_i)$, $\gamma_i = \mathbf{Pr}_{(\boldsymbol{p},\boldsymbol{q})\sim\mu}[\boldsymbol{q} \in B_i \mid \boldsymbol{p} \in A_i]$ satisfying $\sum_{i=1}^{\ell} \alpha_i\gamma_i \geq \frac{1}{2}$, and

$$s(n) \geq \ell + n\sum_{i=1}^{\ell} \alpha_i, \qquad q_{\mathrm{cp}}(n) \geq \sum_{i=1}^{\ell} \Phi_\mu(\alpha_i, \gamma_i)\alpha_i \cdot (1 + n\alpha_i).$$

Given Lemma 1.2.1, it remains to compute the robust expansion of metric spaces in $\mathcal{M}$ for an appropriate distribution μ, and apply the lower bound. Specifically, for any $\delta \in (0, 1)$ and $N > N_0$, let $r = (\log N)^\delta/10$. The distribution μ supported on pairs of points in $([N], d_{G_N})$ at distance at most r is given by letting $(\boldsymbol{p}, \boldsymbol{q}) \sim \mu$ where $\boldsymbol{p} \sim [N]$ uniformly and $\boldsymbol{q}$ is the result of a random walk from $\boldsymbol{p}$ of length r in G_N. Then, since both marginal

distributions of μ are equal and uniform, and G_N is 3-regular,

$$\mathop{\mathbf{Pr}}_{\boldsymbol{p}_1,\boldsymbol{p}_2\sim\rho_1}\left[d_{G_N}(\boldsymbol{p}_1,\boldsymbol{p}_2) < cr\right] < \frac{3^{cr}}{N} < \frac{1}{\sqrt{N}}, \tag{1.2}$$

and since n satisfies (1.1), we may apply Lemma 1.2.1. Finally, an application of the expander mixing lemma shows that for all $\alpha, \gamma \in (0,1)$,

$$\gamma \leq \Phi_\mu(\alpha,\gamma)\cdot\alpha + (1-\lambda_2(\mathcal{L}_{G_N}))^r\sqrt{\Phi_\mu(\alpha,\gamma)}.$$

Deducing the curse of dimensionality is now straight-foward: we apply Lemma 1.2.1 to obtain the sequence of points $\{(\alpha_i,\gamma_i)\}_{i\in[\ell]}$ and consider the subset $L \subset [\ell]$ of indices satisfying $\Phi_\mu(\alpha_i,\gamma_i)\alpha_i \geq \gamma_i/2$. First, consider the case $\sum_{i\in L}\alpha_i\gamma_i \geq 1/4$, and notice that substituting this (along with the definition of L) into the expression for $q_{\mathrm{cp}}(n)$ in Lemma 1.2.1 implies $q_{\mathrm{cp}}(n) = \Omega(n)$. On the other hand, if $\sum_{i\in L}\alpha_i\gamma_i < 1/4$, we must have $\sum_{i\notin L}\alpha_i\gamma_i \geq 1/4$. In this case and since $N > N_0$, every $i \notin L$ satisfies $\gamma_i/2 \leq (1-\eta)^r\sqrt{\Phi_\mu(\alpha_i,\gamma_i)}$. This implies

$$\frac{1}{8}\left(\frac{1}{1-\eta}\right)^r \leq \sum_{i\notin L}\alpha_i\sqrt{\Phi_\mu(\alpha_i,\gamma_i)} \leq \ell,$$

because $\Phi_\mu(\alpha_i,\gamma_i)\alpha_i \leq 1$ and $\ell - |L| \leq \ell$. Seeing as $1-\eta \in (0,1)$ is a fixed constant and the expression of $s(n)$ in Lemma 1.2.1 depends linearly on ℓ, we obtain space complexity exponential in $r = (\log N)^\delta/10$.

1.3 The Cutting Modulus and Random Partitions

In Section 1.2, we showed that the shortest path metric on expander graphs, $\mathcal{M} = \{[N], d_{G_N}\}_{N\in\mathbb{N}}$, suffers the curse of dimensionality in the list-of-points framework; i.e., any sublinear time data structure for ANN incurs exponential dependence in the dimension of the space.[20] Utilizing $\mathcal{M}$ as an initial lower bound instance, Theorem 17 can derive lower bounds for an arbitrary metric space (X, d_X). Informally, if for some $\delta \in (0,1)$, there exists

[20]Recall we use $\log|X|$ as a proxy for dimension of a metric space (X, d_X)

a function $f\colon [N] \to X$ embedding $([N], d_{G_N})$ into (X, d_X) with distortion $(\log N)^{1-\delta}$; then, for any $\delta' < \delta$, we obtain a lower bound for data structures for $(\log N)^{\delta'}$-ANN over X. Hence, a prerequisite for efficient data structures for ANN over X (in the list-of-points framework) is that *shortest path metrics on expander graphs do not embed into* X. This section is based on the work in [14] (and the formal proofs are given in Chapter 3), where we provide a partial converse to the above statement: for any metric space (X, d_X) where expanders do not embed,[21] we construct an ANN data structure over X.[22] The resulting data structure will be in the cell-probe model, i.e., the data structure is efficient in its space and cell-probe complexity, but not in its time complexity.

Definition 15 (The Cutting Modulus [14])**.** *Let (X, d_X) be a finite metric space and $\varepsilon \in (0, 1)$. The cutting modulus $\Xi(X, \varepsilon)$ is the infimum of $\Xi > 0$ such that the following holds. For all $r \geq 0$, and all symmetric probability measures μ supported on pairs of points $X \times X$ at distance at most r in X with marginal distribution ρ, either (1) there exists a set $B \subset X$ with $\rho(B) \geq 1/2$ and $\operatorname{diam}_X(B) \leq r \cdot \Xi$, or (2) there exists a set $S \subset X$ with conductance*

$$\Phi_\mu(S) = \frac{\mu(S, X \setminus S)}{\min\{\rho(S), 1 - \rho(S)\}} \leq \varepsilon.$$

The case of infinite metric spaces is handled by considering all finite subsets. Specifically, the cutting modulus of a (infinite) metric space (Y, d_Y), denoted $\Xi(Y, \varepsilon)$, is the infimum over all $\Xi > 0$ and $N \in \mathbb{N}$ such that all metric spaces (Y', d_Y) induced from Y by considering a subset $Y' \subset Y$ of size N has $\Xi(Y', \varepsilon) \leq \Xi$.

The metric spaces $\mathcal{M}$ defined in Section 1.2 have cutting modulus $\Xi(([N], d_{G_N}), \varepsilon) \gtrsim \log N$ when $\varepsilon < \eta/2$ and $N \geq N_0$. The proof follows from considering $r = 1$ and μ to be the uniform distribution over pairs of points in G_N connected by an edge (which implies ρ is uniform since the graphs are regular). By virtue of the graphs G_N having degree 3, any subset of diameter significantly smaller than $\log N$ will not contain any constant fraction of the graph (the computation is similar to (1.2)). On the other hand, $\lambda_2(\mathcal{L}_{G_N}) \geq \eta$ for

[21] The precise notion of nonembeddability is captured by the *cutting modulus*, which we define in Definition 15.

[22] Furthermore, we note that the data structure will not be in the list-of-points framework.

$N \geq N_0$, which implies, by the (easy direction of the) Cheeger inequalities [48, 4], that any subset has conductance at least $\eta/2$.

More generally, upper bounds on the cutting modulus may follow from an application of Cheeger's inequality to an upper bound on the non-linear spectral gap (Definition 12), and we may utilize various techniques exist for upper-bounding non-linear spectral gaps of metric spaces [103, 105, 110, 111]. The machinery (including Theorem 24) were initially developed for proving nonembeddability of expander metrics; however, these automatically give upper bounds on the cutting modulus. We sketch how Theorem 24 imples $\Xi(X, \varepsilon) \lesssim \log d/\varepsilon^2$: if μ is a symmetric probability distribution supported on pairs of points $X \times X$ at distance at most r in X with marginal ρ, and if $\rho(B) \leq 1/2$ for all $\operatorname{diam}_X(B) \lesssim r \cdot \log d/\varepsilon^2$, then, $\gamma(\mu, \|\cdot\|_X^2) \gtrsim \log^2 d/\varepsilon^4$, and by Theorem 24, $\lambda_2(\mathcal{L}_\mu) \lesssim \varepsilon^2$, so that we may apply Cheeger's inequality to obtain a low-conductance cut $S \subset X$ of μ [48, 4].

Theorem 25 is stated next and places the cutting modulus at the focus of our study of Problem 1.[23]

Theorem 25 (Cell-Probe Data Structure for $(\Xi(X, \varepsilon) + 1)$-ANN [14])**.** *Let (X, d_X) be a metric space of size N and $\varepsilon \in (0, 1)$. There exists a data structure for $(\Xi(X, \varepsilon) + 1)$-ANN over X with space complexity $n^{1+O(\varepsilon)} \log N(\log(1/\varepsilon) + \log\log N)$ and cell-probe complexity $n^{O(\varepsilon)} \log N(\log(1/\varepsilon) + \log\log N)$.*

At a high level, the data structure behind the proof of Theorem 25 builds a (data-dependent) randomized decision tree. The data structure begins with a set P of n dataset points in X and continues the following process while more than one dataset point is considered. The algorithm checks whether there is a cluster in P containing a constant fraction of points with diameter $r \cdot \Xi(X, \varepsilon)$. If such a cluster exists, any dataset point inside the cluster is a $(\Xi(X, \varepsilon) + 1)r$-approximate near neighbor to any query point within distance r from any point in the cluster. The data structure stores one point inside the cluster and makes a child node, recursing on remaining dataset points outside the cluster. If no such cluster exists, Theorem 26, relying on condition (2) of Definition 15, will imply existence of

[23] We know the cutting modulus in conjunction with Theorem 25 cannot be the conclusion of Problem 1. For example, $\Xi(\ell_\infty^d, \varepsilon) = \Theta(\log d/\varepsilon)$, while Theorem 2 gives an $O(\log\log d)$-ANN data structure [78].

a randomized space partition supported on subsets with small conductance (with respect to some symmetric probability measures) which split the dataset roughly evenly. The data structure samples a set $\mathbf{S}$ from the randomized space partition and makes two child nodes: one node recurses on $P \cap \mathbf{S}$ and the other recurses on $P \cap (X \setminus \mathbf{S})$.

The approach of taking a "ball" or "cut" most resembles the data structures for ANN over ℓ_1/ℓ_2 of [10] and ℓ_∞^d of Indyk [78] (proving Theorem 2). At a high level, these follow by designing randomized space partitions which performs well under a fixed dataset assuming the dataset satisfies a well-separatedness condition. In this case, the absence of large clusters of small diameter serves as this condition. The randomized space partitions depend on the dataset, and hence termed "data-dependent LSH."[24] The next subsection is devoted to one such "ball" or "cut" theorem for general metric spaces, analogous to Lemma 1 of [78]. The quality of the randomized space partitions depends on the cutting modulus of the metric space, and following that subsection, we show how the theorem is used to prove Theorem 25.

1.3.1 Data-dependent randomized space partitions from bounds on the cutting modulus

Theorem 26 ([14])**.** *Let (X, d_X) be a metric space of size N, $\varepsilon \in (0, 1/8)$, and $r \geq 0$. There exists a collection $\mathcal{F}$ of at most $N^{O(\log(1/\varepsilon)+\log\log N)}$ subsets of X such that the following holds. If, for $1 < m \leq N$ and $x_1, \ldots, x_m \in X$, all $B \subset X$ with $\operatorname{diam}_X(B) \leq r \cdot \Xi(X, \varepsilon)$ satisfy $|B \cap \{x_1, \ldots, x_m\}| \leq m/8$, there exists a distribution $\mathcal{H}$ supported on $\mathcal{F}$ satisfying:*

1. *A subset $\mathbf{S} \sim \mathcal{H}$ satisfies $|\mathbf{S} \cap \{x_1, \ldots, x_m\}| \in [m/8, 7m/8]$ with probability 1.*
2. *If $y, z \in X$ satisfy $d_X(y, z) \leq r$, then*

$$\Pr_{\mathbf{S}\sim\mathcal{H}}[|\{y, z\} \cap \mathbf{S}| = 1] \leq 20\varepsilon.$$

There are three crucial components of the randomized space partition defined in

[24] In contrast, a (data-independent) LSH is a randomized space partition which performs well under any dataset. For example, the data structures for ANN over ℓ_1 of [79] and [93] (proving of Theorem 1) fall in this category.

Theorem 26. The first is the balancedness of parts with respect to a dataset of points $x_1, \ldots, x_m \in X$, as guaranteed by (1). The second is that the probability of separating any two points within distance r is small, as guaranteed by (2). In the language of Locality-Sensitive Hashing (LSH) [79], (1) is analogous to an upper bound of $7/8$ on p_2 and (2) is analogous to a lower bound of $1 - 20\varepsilon$ on p_1. Subsequently, Theorem 26 foreshadows an ANN data structure with time complexity $n^{\log(1/p_1)/\log(1/p_2)} = n^{O(\varepsilon)}$. The third crucial component is the fact that samples from the distribution $\mathcal{H}$ may be encoded in at most $O(\log N(\log(1/\varepsilon) + \log\log N))$ bits. This is a consequence of the collection $\mathcal{F}$ being small and fixed in advanced.

Roughly speaking, the cutting modulus asserts that for every symmetric distribution μ over pairs of points in X at distance at most r, where the marginal distribution ρ contains no large clusters of diameter $r \cdot \Xi(X, \varepsilon)$, there exists a set S with conductance at most ε. After an iterative argument taking unions of low-conductance cuts, the set S can be assumed to be roughly balanced, i.e., $\min\{\rho(S), 1 - \rho(S)\} \geq 1/3$ and $\mu(S, X \setminus S) \leq 2\varepsilon$.[25] If points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$ from X are *sampled* from ρ, $|S \cap \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m\}| \in [m/8, 7m/8]$ with very high probability over randomness in generating the points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$. Furthermore, since $\mu(S, X \setminus S)$ is exactly the probability that a random pair $(\boldsymbol{y}, \boldsymbol{z}) \sim \mu$ is separated by S, i.e., $|\{\boldsymbol{y}, \boldsymbol{z}\} \cap S| = 1$, if the query and near-neighbor pair $(\boldsymbol{y}, \boldsymbol{z})$ are sampled from μ, $\boldsymbol{y}$ and $\boldsymbol{z}$ lie in the same side of S with probability at least $1 - 2\varepsilon$ over randomness in generating the query and near-neighbor pair. As a result, the cutting modulus gives, for each distribution μ with marginal ρ, a *single* set $S \subset X$ satisfying conditions akin to (1) and (2) assuming the points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m$ and $\boldsymbol{y}, \boldsymbol{z}$ are sampled from ρ and μ, and the probability of success is with respect to the randomness in generating the points.

On the other hand, Theorem 26 considers a fixed set of points $x_1, \ldots, x_m$ and y, z, and asks for a distribution $\mathcal{H}$ over subsets of X performing well for $x_1, \ldots, x_m$ and y, z, where the probability of success is with respect to a sample $\mathbf{S} \sim \mathcal{H}$. At a high level, Theorem 26 generates, from a solution for any distributional input, a distribution over (few) solutions for any worst-case input. The proof of Theorem 26 proceeds in two steps. We first argue

[25] For example, one may use the reduction which converts a low-conductance cut into a balanced separator (Section 3.3 of [95]).

Theorem 26 is implied by a lemma we state next. Then, we prove the next lemma assuming an upper bound on the cutting modulus.

Lemma 1.3.1. *Let (X, d_X) be a metric space of size N, $\varepsilon \in (0, 1/8)$, and $r \geq 0$. Consider $s \in \mathbb{N}$, and suppose that for every distribution ρ supported on points in X satisfying all $B \subset X$ with $\operatorname{diam}_X(B) \leq r \cdot \Xi(X, \varepsilon)$ have $\rho(B) \leq 1/7$, there exists a distribution $\mathcal{H}_\rho$ supported on at most s subsets of X where the following hold:*

1. *A subset $\mathbf{S} \sim \mathcal{H}_\rho$ satisfies $\rho(\mathbf{S}) \in [1/7, 6/7]$ with probability 1.*
2. *If $y, z \in X$ satisfy $d_X(y, z) \leq r$, then $|\{y, z\} \cap \mathbf{S}| = 1$ with probability at most 20ε over $\mathbf{S} \sim \mathcal{H}_\rho$.*

Then, the conclusion of Theorem 26 hold with $\log |\mathcal{F}| \lesssim \log N \cdot \log s$.

The implication from Lemma 1.3.1 to Theorem 26 can be understood in the context of online learning. An oracle has an unknown and fixed dataset $P \subset X$ without large clusters, and we want a (deterministic) algorithm which builds a distribution $\mathcal{H}$ satisfying the conditions of Theorem 26 by querying the oracle. The algorithm proceeds in a sequence of iterations. In each iteration, the algorithm updates a "proposed distribution", ρ supported on X, which over the course of the execution, approaches the uniform distribution over P, which we denote ψ_P. The notion of approaches is with respect to Kullback-Liebler (KL) divergence, $\mathrm{RE}(\cdot\|\cdot)$, and the updates are according to the multiplicative weights update rule (see, [74, 20]).

Claim 1.3.2. *Let X be a fixed set, ρ be any distribution supported on X, and $\eta \in (0, 1)$ be a fixed constant. For any set $S \subset X$, there exists two distributions ρ_S^+ and ρ_S^- supported on X such that for any unknown distribution ψ supported on X,*

- *If $\rho(S) - \psi(S) \geq \eta$, then $\mathrm{RE}(\rho_S^-\|\psi) \leq \mathrm{RE}(\rho\|\psi) - \eta^2/4$.*
- *If $\rho(S) - \psi(S) \leq -\eta$, then $\mathrm{RE}(\rho_S^+\|\psi) \leq \mathrm{RE}(\rho\|\psi) - \eta^2/4$.*

Initially, the algorithm sets ρ to the uniform over X, and continually proceeds as follows. Suppose there exists a set $B \subset X$ with $\operatorname{diam}_X(B) \leq r \cdot \Xi(X, \varepsilon)$ satisfying $\rho(B) > 1/7$; since the $\psi_P(B) < 1/8$, the algorithm updates ρ to ρ_S^- according to Claim 1.3.2, thereby decreasing $\mathrm{RE}(\rho\|\psi_P)$ by at least a small constant. Otherwise, using the assumption of

Lemma 1.3.1, the algorithm designs a candidate distribution $\mathcal{H}_\rho$ over subsets of X, and measures the quality of $\mathcal{H}_p$ with respect to ψ_P by querying the oracle.[26] In particular, if (1) of Theorem 26 is not satisfied, and there is a subset S in the support of $\mathcal{H}_\rho$ where $\psi_P(S) \notin [1/8, 7/8]$, the oracle reports the set S and whether $\psi_P(S)$ is larger than $7/8$ or smaller than $1/8$ using $\lceil \log_2 s \rceil + 1$ bits. Again, the algorithm updates ρ by to either ρ_S^+ or ρ_S^- according to Claim 1.3.2, which decreases $\mathrm{RE}(\rho \| \psi_P)$ by at least a constant. To complete the argument, notice that $\mathrm{RE}(\rho \| \psi_P) \in [0, \log N]$, implying that the algorithm terminates with a distribution $\mathcal{H}_\rho$ satisfying the conditions of Theorem 26 after $O(\log N)$ iterations. The upper bound on $|\mathcal{F}|$ follows from the fact that $\mathcal{H}_\rho$ only depends on $O(\log N \log s)$ bits communicated by the oracle.

The second step, summarized in the lemma below, implies the conditions of Lemma 1.3.1 with $s = O(\log(N)/\varepsilon^2)$ and completes the proof of Theorem 26.

Lemma 1.3.3. *Let (X, d_X) be a metric space of size N, $\varepsilon \in (0, 1/8)$, $r \geq 0$, and $s = O(\log(N)/\varepsilon^2)$. Suppose ρ is a distribution supported on points in X such that all $B \subset X$ with $\mathrm{diam}_X(B) \leq r \cdot \Xi(X, \varepsilon)$ satisfy $\rho(B) \leq 1/7$. There exists a distribution $\mathcal{H}_\rho$ supported on at most s subsets of X where the following holds:*

1. *A subset $\mathbf{S} \sim \mathcal{H}_\rho$ satisfies $\rho(\mathbf{S}) \in [1/7, 6/7]$ with probability 1.*
2. *If $y, z \in X$ satisfy $d_X(y, z) \leq r$, then $|\{y, z\} \cap \mathbf{S}| = 1$ with probability at most 20ε over $\mathbf{S} \sim \mathcal{H}_\rho$.*

The proof of Lemma 1.3.3, without the sparsity condition on the distributions $\mathcal{H}_\rho$, follows from von Neumann's minimax theorem. Specifically, let $C_r \subset X \times X$ be the set of pairs of points $(y, z) \in X \times X$ at distance at most r, i.e., $d_X(y, z) \leq r$, and consider the compact and convex set $\mathcal{D} \subset \mathbb{R}^{|C_r|}$ of symmetric probability distributions supported on pairs in C_r. Furthermore, denote $\mathcal{S}_\rho$ as the collection of subsets X which are roughly balanced with respect to ρ, i.e., $S \in \mathcal{S}_\rho$ satisfies $\rho(S) \in [1/7, 6/7]$, and consider the compact, and convex set $\mathcal{K} \subset \mathbb{R}^{|\mathcal{S}_\rho|}$ of probability distributions supported on subsets in $\mathcal{S}_\rho$.

[26] Notice that the distribution $\mathcal{H}_\rho$ satisfies condition (2) of Theorem 26, as this is the same as condition (2) of Lemma 1.3.1. Hence, only (1) is to be checked.

Then, Lemma 1.3.3 is exactly saying

$$\min_{\substack{\mathcal{H}\in\mathcal{K}\\ \mathcal{H}\ s\text{-sparse}}} \max_{\mu\in\mathcal{D}} \sum_{(y,z)\in C_r} \sum_{S\in\mathcal{S}_\rho} \mathbf{1}\left\{|\{y,z\}\cap S|=1\right\}\cdot\mu(y,z)\cdot\mathcal{H}(S)\le 20\varepsilon. \tag{1.3}$$

Removing the sparsity condition on $\mathcal{H}$ and applying von Neumann's minimax theorem, a proof of Lemma 1.3.3 without a bound on s follows from showing that

$$\max_{\mu\in\mathcal{D}} \min_{\mathcal{H}\in\mathcal{K}} \sum_{(y,z)\in C_r} \sum_{S\in\mathcal{S}_\rho} \mathbf{1}\left\{|\{y,z\}\cap S|=1\right\}\cdot\mu(y,z)\cdot\mathcal{H}(S)\le 20\varepsilon. \tag{1.4}$$

Toward this (relaxed) goal, consider any distribution $\mu\in\mathcal{D}$, and let $\tilde{\mu}\in\mathcal{D}$ be the following mixture: a sample $(\boldsymbol{y},\boldsymbol{z})\sim\tilde{\mu}$ is generated by letting $(\boldsymbol{y},\boldsymbol{z})=(\boldsymbol{y}',\boldsymbol{z}')\sim\mu$ with probability $1/8$, and $(\boldsymbol{y},\boldsymbol{z})=(\boldsymbol{x},\boldsymbol{x})$ where $\boldsymbol{x}\sim\rho$ with probability $7/8$. We now apply Definition 15 to $\tilde{\mu}$. By writing the marginal distribution $\tilde{\rho}$ of $\tilde{\mu}$, we have $\tilde{\rho}(B)\le 1/2$ for all $B\subset X$ with $\mathrm{diam}_X(B)\le r\cdot\Xi(X,\varepsilon)$, since $\tilde{\rho}(B)\le 7/8\cdot\rho(B)+1/8<7/8\cdot 1/7+1/8$. Hence, let $S\subset X$ have $\tilde{\mu}(S,X\setminus S)\le 2\varepsilon$ and $\tilde{\rho}(S)\in[1/3,2/3]$. Using again the fact that $\tilde{\mu}$ is defined as a mixture of μ and twice ρ, we have $\mu(S,X\setminus S)\le 16\varepsilon$ and $\rho(S)\in[5/21,16/21]$. Letting $\mathcal{H}$ be the distribution which is only supported on the set S proves the left-hand side of (1.4) is at most $16\varepsilon<20\varepsilon$.

The fact that there exists an s-sparse distribution $\mathcal{H}\in\mathcal{K}$ satisfying (1.3), follows, again, from the multiplicative weights update algorithm. The argument proceeds as in Section 3.2 of [20] on approximately solving zero-sum games [65]. We consider a payoff matrix indexed by $C_r\times\mathcal{S}_\rho$. For $(y,z)\in C_r$ and $S\in\mathcal{S}_\rho$, the entry in the (y,z)-th row and S-th column is $\mathbf{1}\{|\{y,z\}\cap S|=1\}$. As shown in the previous paragraph, for any "mixed strategy" of the row player, i.e., a distribution $\mu\in\mathcal{D}$, there exists a "pure strategy" of the column player, i.e., a set $S\in\mathcal{S}_\rho$, with expected payoff less than 16ε. Hence, the goal is to find a "mixed strategy" for the column player, which is approximately optimal up to additive 4ε against any "pure strategy" of the row player; in addition, the mixed strategy should be s-sparse. Theorem 3.1 of [20] shows exactly this: there exists a 4ε-approximate mixed strategy for the column player given by the average of $O(\log(|C_r|)/\varepsilon^2)$ pure strategies for the column player. Since $|C_r|\le N^2$, this completes Lemma 1.3.3, and hence, Theorem 26.

1.3.2 Proof of Theorem 25

Assuming Theorem 26, completing the sketch of Theorem 25 is straight-forward. The depth of the randomized decision tree is at most $\log_{8/7} n$: either there is a cluster of more than $1/8$-fraction of dataset points within a set of diameter $r \cdot \Xi(X, \varepsilon)$, and we store one point inside the cluster, or the data structure samples a set $\mathbf{S} \sim \mathcal{H}$ and recurses on dataset points falling inside and outside the set $\mathbf{S}$, decreasing the dataset by at least a $1/8$-fraction.

The querying algorithm, upon receiving a query $q \in X$, begins at the root and walks down the tree. If a node contained a cluster of dataset points, and the point stored as part of the cluster is within distance $r(\Xi(X, \varepsilon) + 1)$ from q, the data structure returns the point. Otherwise, the querying algorithm reads the encoding of the set $\mathbf{S} \sim \mathcal{H}$ and checks whether $q \in \mathbf{S}$. If so, the algorithm proceeds with the child node corresponding to $\mathbf{S}$; otherwise, the algorithm proceeds with the child node corresponding to $X \setminus \mathbf{S}$.

Suppose $p \in P$ satisfies $d_X(q, p) \leq r$, and consider the event, over the randomness in samples to the distributions $\mathcal{H}$ along the path in the decision tree containing p (of depth at most $\log_{8/7} n$), that whenever a set $\mathbf{S} \sim \mathcal{H}$ is sampled, $|\{p, q\} \cap \mathbf{S}| \neq 1$. Notice this occurs with probability at least $(1 - 20\varepsilon)^{\log_{8/7} n} = n^{-\log_{8/7}(1/(1-20\varepsilon))} \geq n^{-1000\varepsilon}$. When the event does occur, two outcomes may occur. The first is that query algorithm encounters a cluster node with a point within distance $r(\Xi(X, \varepsilon) + 1)$ of the query and returns successfully. The second outcome is when the first outcome does not occur. In this case, p does not belong to a cluster (otherwise, q is within $r(\Xi(X, \varepsilon) + 1)$ from a cluster point); since p and q were never split by a set $\mathbf{S}$, they lie in the same leaf node and the query algorithm returns p. Repeating the randomized decision tree $O(n^{1000\varepsilon})$ times to increase the probability of success implies Theorem 25.

Remark 27 (A remark on cell-probe complexity)**.** This section's primary focus was ANN data structures with efficient space and cell-probe complexities. Time complexity was completely ignored. The key concern is that a root-to-leaf path in the randomized decision tree built may take $|X|^{\Omega(1)}$ time to traverse. In particular, the data structure consists of $n^{O(\varepsilon)}$ decision trees and for a query $q \in X$, a root-to-leaf path on the tree is determined

by the inclusion or exclusion of q in a set $\mathbf{S} \subset X$ sampled from $\mathcal{H}$.[27] The set $\mathbf{S}$ may be large (for example, $|\mathbf{S}| \in [|X|^{0.01}, |X| - |X|^{0.01}]$), and a priori have no structure. On a query $q \in X$, the data structure walks down the decision tree and must repeatedly performing the following: at the current node (which does not contain a cluster), the data structure reads the encoding of the set $\mathbf{S}$ sampled for that node, decodes the set $\mathbf{S}$, and scans $\mathbf{S}$ to determine if $q \in \mathbf{S}$. Notice that the cell-probe complexity is determined by the size of the encoding of $\mathbf{S}$, and the time complexity is determined by the time to decode $\mathbf{S}$ and $|\mathbf{S}|$.

1.4 Upper Bounds on the Cutting Modulus

This section presents upper bounds on the cutting modulus of various metric spaces. Claims in this section are formalized in Chapter 4. Combined with Theorem 25, these result in efficient cell-probe ANN data structures. Emphasis is given to bounds which also yield time efficient data structures. As briefly alluded to in the last section, we augment the notion of cutting modulus to mind time complexity. We begin with bounding the cutting modulus of ℓ_1^d. This is the simplest such bound, and elucidates approaches to come.

1.4.1 The cutting modulus of ℓ_1^d

Fix $N \in \mathbb{N}$ and consider any finite subset $X \subset \ell_1^d$. Let μ be any symmetric probability distribution supported on pairs in $X \times X$ and let ρ be the marginal distribution of μ. Consider the quantity

$$\mathsf{R}(\mu, \|\cdot\|_1) = \frac{\sum_{x \in X} \sum_{y \in X} \mu(x,y) \|x - y\|_1}{\sum_{x \in X} \sum_{y \in X} \rho(x)\rho(y) \|x - y\|_1} = \frac{\mathop{\mathbf{E}}_{(\boldsymbol{x},\boldsymbol{y}) \sim \mu} [\|\boldsymbol{x} - \boldsymbol{y}\|_1]}{\mathop{\mathbf{E}}_{\boldsymbol{x},\boldsymbol{y} \sim \rho} [\|\boldsymbol{x} - \boldsymbol{y}\|_1]}, \tag{1.5}$$

Suppose that, for $r \geq 0$, μ and ρ satisfy the conditions of Definition 15. In other words, μ is supported on pairs of points at distance at most r, and $\rho(B) \leq 1/2$ for every $B \subset X$ with

[27] Technically, there exists the (easy) case when there was a large cluster of dataset points with small diameter. In this case, the data structure performs computes $d_X(q, p)$ for some $p \in P$ and either returns p, or proceeds to the child node.

$\mathrm{diam}_{\ell_1^d}(B) \le 4r/\varepsilon$.[28] Under these assumptions, $\mathsf{R}(\mu, \|\cdot\|_1) \le \varepsilon$; specifically, we upper bound the numerator of the right-hand side of (1.5) by r, and lower bound the denominator of the right-hand of side of (1.5) by r/ε. The numerator is at most r because every $(\boldsymbol{x}, \boldsymbol{y}) \sim \mu$ has $\|\boldsymbol{x} - \boldsymbol{y}\|_1 \le r$. For bounding the denominator, notice that for every $x \in X$, the set $B = X \cap B_{\ell_1^d}(x, 2r/\varepsilon)$ satisfies $\rho(B) \le 1/2$ because $\mathrm{diam}_{\ell_1^d}(X) \le 4r/\varepsilon$. This implies $\mathbf{E}_{\boldsymbol{y}\sim\rho}[\|x - \boldsymbol{y}\|_1] \ge r/\varepsilon$ and gives a lower bound on the denominator.

Lemma 1.4.1. *Let $d \in \mathbb{N}$ and μ be any symmetric probability distribution supported on finitely many pairs of points in $\mathbb{R}^d$. Let ρ be the marginal of μ and suppose it has non-zero support on at least two points. There exists a distribution $\mathcal{S}$ supported on subsets of $\mathbb{R}^d$ satisfying*

$$\frac{\mathop{\mathbf{E}}_{\mathbf{S}\sim\mathcal{S}}\left[\mu(\mathbf{S}, \overline{\mathbf{S}})\right]}{\mathop{\mathbf{E}}_{\mathbf{S}\sim\mathcal{S}}\left[\rho(\mathbf{S})(1-\rho(\mathbf{S}))\right]} \le \mathsf{R}(\mu, \|\cdot\|_1). \tag{1.6}$$

Since $\mathsf{R}(\mu, \|\cdot\|_1) \ge 1/\varepsilon$,[29] Lemma 1.4.1 implies there exists a set $S \subset \mathbb{R}^d$ in the support of $\mathcal{S}$ with $\Phi_\mu(S) \le \varepsilon$.[30] The proof of Lemma 1.4.1 proceeds by noticing that, since $\|x - y\|_1 = \sum_{k\in[d]} |x_k - y_k|$, there exists $k \in [d]$ where[31]

$$\frac{\mathop{\mathbf{E}}_{(\boldsymbol{x},\boldsymbol{y})\sim\mu}[|\boldsymbol{x}_k - \boldsymbol{y}_k|]}{\mathop{\mathbf{E}}_{\boldsymbol{x},\boldsymbol{y}\sim\rho}[|\boldsymbol{x}_k - \boldsymbol{y}_k|]} \le \frac{\mathop{\mathbf{E}}_{(\boldsymbol{x},\boldsymbol{y})\sim\mu}[\|\boldsymbol{x} - \boldsymbol{y}\|_1]}{\mathop{\mathbf{E}}_{\boldsymbol{x},\boldsymbol{y}\sim\rho}[\|\boldsymbol{x} - \boldsymbol{y}\|_1]} = \mathsf{R}(\mu, \|\cdot\|_1). \tag{1.7}$$

Let $\alpha = \inf\{x_k \in \mathbb{R} : x \in \mathrm{supp}(\rho)\}$ and $\beta = \sup\{x_k \in \mathbb{R} : x \in \mathrm{supp}(\rho)\}$ be the smallest and largest values on the k-th coordinate of a sample of ρ, and consider the distribution $\mathcal{S}$ supported on sets $\mathbf{S} \subset \mathbb{R}^d$ given by letting $\mathbf{S} = \left\{x \in \mathbb{R}^d : x_k \le \boldsymbol{\tau}\right\}$ where $\boldsymbol{\tau} \sim [\alpha, \beta]$ is

[28] Here, the assumption on $\mathrm{diam}_{\ell_1^d}(B) \le 4r/\varepsilon$ is to upper bound $\Xi(\ell_1^d, \varepsilon) \le 4/\varepsilon$.

[29] Notice this implies ρ has non-zero supported on at least two different points, since otherwise, the denominator of (1.5) is 0.

[30] In particular, (1.6) implies, by re-arranging terms, that $\mathbf{E}_{\mathbf{S}\sim\mathcal{S}}\left[\mu(\mathbf{S}, \overline{\mathbf{S}}) - \varepsilon\rho(\mathbf{S})(1-\rho(\mathbf{S}))\right] \le 0$. Hence, there exists a set $S \subset \mathbb{R}^d$ in the support of $\mathcal{S}$ where $\mu(S, \overline{S}) - \varepsilon\rho(S)(1-\rho(S)) \le 0$, and the bound on $\Phi_\mu(S)$ follows since $\rho(S)(1-\rho(S)) \le \min\{\rho(S), 1-\rho(S)\}$. Generally, for any $m \in \mathbb{N}$ and collection $\{(a_i, b_i)\}_{i\in[m]}$ of pairs of real numbers, $(\sum_i a_i)/(\sum_i b_i) = \gamma$ implies there exists some $i \in [m]$ where $b_i \ne 0$ and $a_i/b_i \le \gamma$.

[31] The argument is analogous to footnote 30.

drawn uniformly at random.[32] Then, we simply notice the left-hand side of (1.6) is exactly the left-hand side of (1.7). Formally,

$$\mathop{\mathbf{E}}_{\mathbf{S}\sim\mathcal{S}}\left[\mu(\mathbf{S},\overline{\mathbf{S}})\right] = \sum_{x\in X}\sum_{y\in Y}\mu(x,y)\mathop{\mathbf{Pr}}_{\mathbf{S}\sim\mathcal{S}}\left[|\{x,y\}\cap\mathbf{S}|=1\right] = \mathop{\mathbf{E}}_{(\boldsymbol{x},\boldsymbol{y})\sim\mu}\left[\frac{|\boldsymbol{x}_k-\boldsymbol{y}_k|}{\beta-\alpha}\right].$$
$$\mathop{\mathbf{E}}_{\mathbf{S}\sim\mathcal{S}}\left[\rho(\mathbf{S})(1-\rho(\mathbf{S}))\right] = \mathop{\mathbf{E}}_{\in Y}\rho(x)\rho(y)\mathop{\mathbf{Pr}}_{\mathbf{S}\sim\mathcal{S}}\left[|\{x,y\}\cap\mathbf{S}|=1\right] = \mathop{\mathbf{E}}_{\boldsymbol{x},\boldsymbol{y}\sim\rho}\left[\frac{|\boldsymbol{x}_k-\boldsymbol{y}_k|}{\beta-\alpha}\right].$$

Remark 28 (Time Complexity Bounds for ℓ_1^d)**.** Even though the above-mentioned argument bounding $\Xi(X,\varepsilon)\le 4/\varepsilon$, combined with Theorem 25, gives a cell-probe data structure, we may additionally conclude in this case that $q(n)=O(q_{\mathrm{cp}}(n))$. The argument proceeds by noticing that Lemma 1.4.1 finds, given any distribution μ on pairs of points in ℓ_1^d, a low-conductance cut $S\subset\mathbb{R}^d$ given by an axis-aligned threshold. In other words, a set S is specified by a coordinate $k\in[d]$ and a threshold $\tau\in\mathbb{R}$; then, $x\in S$ if and only if $x_k\le\tau$. Following the argument in Subsection 1.3.2 and Remark 27, a walk down the decision tree must repeatedly determine whether the query q lies in a set $\mathbf{S}$ sampled from $\mathcal{H}$. In this case, the set $\mathbf{S}$ is given by the union of axis-aligned thresholds, i.e., axis-aligned boxes (with possibly infinite sides).[33] Hence, sets $\mathbf{S}$ in $\mathcal{H}$, or their complements, are encoded by a list of at most d triples $(k,\tau_-,\tau_+)\in[d]\times(\mathbb{R}\cup\{-\infty,\infty\})^2$, and determining whether $q\in\mathbf{S}$ takes linear time in the encoding of $\mathbf{S}$, implying $q(n)=O(q_{\mathrm{cp}}(n))$.

1.4.2 Bounding the Cutting Modulus via Holder Maps of Unit Spheres

For any metric space (X,d_X), the natural approach to upper bound $\Xi(X,\varepsilon)$ follows, yet again, from embedding X into ℓ_1^d and applying Lemma 1.4.1. This is the approach we take in this section when (X,d_X) is a normed space, albeit with a notion of embedding less stringent than Definition 8. Consider any symmetric probability distribution μ supported on finitely many pairs in X and let ρ be its marginal distribution. For any $r\in(0,\infty)$, we

[32] Notice that by definition of α and β, $\rho(\mathbf{S})\in(0,1)$ with probability 1.

[33] Even though, Lemma 1.4.1 gives a low conductance set, $\rho(S)$ may be arbitrarily small. Recall that a set $\mathbf{S}$ in $\mathcal{H}$ must have $\rho(S)\in[1/3,2/3]$, so we may apply Lemma 1.4.1 multiple times. See also, Footnote 25.

consider the quantity[34]

$$\mathsf{R}(\mu, d_X^r) = \frac{\mathop{\mathbf{E}}_{(\boldsymbol{x},\boldsymbol{y})\sim\mu}[d_X(\boldsymbol{x},\boldsymbol{y})^r]}{\mathop{\mathbf{E}}_{\boldsymbol{x},\boldsymbol{y}\sim\rho}[d_X(\boldsymbol{x},\boldsymbol{y})^r]}, \tag{1.8}$$

which we term the *non-linear Rayleigh quotient*. When (X, d_X) is the metric on the real line, $(\mathbb{R}, |\cdot|)$, and $r = 2$, (1.8) is exactly the Rayleigh quotient of the normalized Laplacian of the (weighted) graph defined according to μ. Specifically, let $U \subset \mathbb{R}$ be the support of ρ, and since (1.8) is invariant under translation, assume points in U satisfy $\sum_{x\in U} \rho(x)x = 0$.[35] Consider the weighted graph $G = (V, E, w)$ where $V = U$, $(x, y) \in E$ iff $\mu(x, y) \neq 0$, and $w(x, y) = \mu(x, y)$. The normalized Laplacian $\mathcal{L}_G$ is the $|U| \times |U|$ matrix, where rows and columns are indexed by $x \in U$, and the entry $(\mathcal{L}_G)_{x,y}$ is $-\mu(x, y)/\sqrt{\rho(x)\rho(y)}$ if $x \neq y$, and $1 - \mu(x, x)/\rho(x)$ if $x = y$. Then, the vector $z \in \mathbb{R}^{|U|}$ where $z_x = \sqrt{\rho(x)} \cdot x$ satisfies $\langle z, z\rangle = (1/2)\sum_{x,y\in U} \rho(x)\rho(y)|x - y|^2$ and $\langle z, \mathcal{L}_G z\rangle = (1/2)\sum_{x,y\in U} \mu(x, y)|x - y|^2$.

For a metric space (Y, d_Y) and $p \in (0, \infty)$, we consider a function $f\colon X \to Y$,[36] and write

$$\mathsf{R}(f(\mu), d_Y^r) = \frac{\mathop{\mathbf{E}}_{(\boldsymbol{x},\boldsymbol{y})\sim\mu}[d_Y(f(\boldsymbol{x}), f(\boldsymbol{y}))^r]}{\mathop{\mathbf{E}}_{\boldsymbol{x},\boldsymbol{y}\sim\rho}[d_Y(f(\boldsymbol{x}), f(\boldsymbol{y}))^r]}.$$

The goal of this section is to design, given a d-dimensional normed space $X = (\mathbb{R}^d, \|\cdot\|_X)$ and a symmetric probability distribution μ over finitely many pairs of points in $\mathbb{R}^d$, function $f\colon \mathbb{R}^d \to \mathbb{R}^d$ such that for $\Psi\colon \mathbb{R}^{\geq 0} \to \mathbb{R}^{\geq 0}$ with $\Psi(\varepsilon) \to 0$ as $\varepsilon \to 0$,[37]

$$\mathsf{R}(f(\mu), \|\cdot\|_1) \leq \Psi\left(\mathsf{R}(\mu, \|\cdot\|_X^r)\right). \tag{1.9}$$

Combining (1.9) with Lemma 1.4.1 immediately implies $\Xi(X, \varepsilon) \leq 4(1/\Psi^{-1}(\varepsilon))^{1/r}$, where $\Psi^{-1}(\varepsilon) = \sup\{\varepsilon_0 \geq 0 : \forall \varepsilon' \leq \varepsilon_0, \Psi(\varepsilon') \leq \varepsilon\}$. Then, Theorem 25 gives an $(4/\Psi^{-1}(\varepsilon)^{1/r} +$

[34]Equation (1.5) corresponds exactly to the case $(X, d_X) = \ell_1^d$ and $r = 1$.

[35]Formally, let $t = \sum_{x\in U} \rho(x)x$ and $U' = \{x - t : x \in U\}$. Then, the ratio in (1.8) remains unchanged when using the distribution μ' given by $(\boldsymbol{x} - t, \boldsymbol{y} - t)$ where $(\boldsymbol{x}, \boldsymbol{y}) \sim \mu$ with marginal ρ' instead of μ. Furthermore, $\sum_{x\in U'} \rho'(x)x = 0$.

[36]For example, f could be an embedding as in Definition 8, or as we will see, a slightly weaker object.

[37]In the discussion, it is implicit that f does not map all points in $\mathrm{supp}(\rho)$ to a single point and make the denominator of (1.8) be 0.

1)-ANN cell-probe data structure over X. We note that comparison inequalities analogous to (1.9) between general metric spaces are studied in [113]. There, comparisons are made among nonlinear spectral gap (recall Definition 12), which may be equivalently stated as the supremum over all $f\colon X \to X$ of $1/\mathsf{R}(f(\mu), d_X^r)$. While relating the nonlinear spectral gap of a metric space (X, d_X) and that of ℓ_1 gives upper bounds on $\Xi(X, \varepsilon)$ as well, maintaining the function f in (1.9) allows for a discussion on computational complexity. In particular, upper bounds on the time complexity of ANN data structures obtained via this method will follow from restrictions imposed on the computational complexity of f.

Remark 29. *Section 1.3 describes the cutting modulus and suggests an approach for ANN data structures in order to address limitations of the embeddings approach from Section 1.1. It seems peculiar that we will upper bound $\Xi(X, \varepsilon)$ using metric embeddings into ℓ_1, and that this will overcome limitations of the embeddings approach for ANN data structures. The key is that the embeddings necessary are substantially weaker than those prescribed in Section 1.1. In particular, the inequality (1.9) allows the embedding f to preserve distances on average over μ and ρ. This relaxed guarantee is what this section will exploit.*

Definition 16 (Holder Homeomorphism between Spheres)**.** *Let $X = (\mathbb{R}^d, \|\cdot\|_X)$ and $Y = (\mathbb{R}^{d'}, \|\cdot\|)$ be two normed spaces. For $c \geq 0$ and $\alpha \in (0, 1]$, function $\phi\colon S(X) \to S(Y)$ is an α-Holder homeomorphism with Holder constant c if it is bijective and for every $a, b \in S(X)$,*

$$\|\phi(a) - \phi(b)\|_Y \leq c \cdot \|a - b\|_X^{\alpha}.$$

Lemma 1.4.2 (Radial Extensions of Holder Homeomorphism between Spheres)**.** *Let $X = (\mathbb{R}^d, \|\cdot\|_X)$ and $Y = (\mathbb{R}^{d'}, \|\cdot\|_Y)$ be two normed spaces, and for $c \geq 0$ and $\alpha \in (0, 1]$, let $\phi\colon S(X) \to S(Y)$ be an α-Holder homeomorphism with constant c. Then, for $r \geq 1$, the radial extension $f_{\phi,r}\colon \mathbb{R}^d \to \mathbb{R}^{d'}$ given by*

$$f_{\phi,r}(x) = \|x\|_X^r \cdot \phi(x/\|x\|_X) \tag{1.10}$$

satisfies $\|f_{\phi,r}(x)\|_Y = \|x\|_X^r$ *for every* $x \in \mathbb{R}^d$*, and for every* $x, y \in \mathbb{R}^d$*,*

$$\|f_{\phi,r}(x) - f_{\phi,r}(y)\|_Y \leq \left(2^\alpha c + 2^{1-\alpha} r\right) \|x - y\|_X^\alpha \left(\|x\|_X^{r-\alpha} + \|y\|_X^{r-\alpha}\right). \tag{1.11}$$

A weakness of using the radial extensions of Holder homeomorphism as embeddings is that the quality of (4.2) degrades as norms of vectors grow. However, we will still derive (1.9) when f is a translation of $f_{\phi,r}$, since distances will be preserved on average. To garner intuition, consider a distribution μ supported on finitely many pairs in $\mathbb{R}^d \times \mathbb{R}^d$, and suppose that after a translation, the marginal distribution ρ is roughly centered after applying $f_{\phi,r}$; formally, assume[38]

$$\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\rho}[f_{\phi,r}(\boldsymbol{x})] = 0.$$

Under this condition,

$$\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\rho}[\|f_{\phi,r}(\boldsymbol{x})\|_Y] \leq \mathop{\mathbf{E}}_{\boldsymbol{x},\boldsymbol{y}\sim\rho}[\|f_{\phi,r}(\boldsymbol{x}) - f_{\phi,r}(\boldsymbol{y})\|_Y] \leq 2 \mathop{\mathbf{E}}_{\boldsymbol{x}\sim\rho}[\|f_{\phi,r}(\boldsymbol{x})\|_Y] \tag{1.12}$$

by the triangle inequality. We use (4.1) in Lemma 1.4.2 to lower bound the denominator of $\mathsf{R}(f_{\phi,r}(\mu), \|\cdot\|_Y)$, and (4.2) to upper bound the numerator of $\mathsf{R}(f_{\phi,r}(\mu), \|\cdot\|_Y)$. In particular, (4.1) and (1.12) imply the denominator of $\mathsf{R}(f_{\phi,r}(\mu), \|\cdot\|_Y)$ is up to a constant factor at least $\mathbf{E}_{\boldsymbol{x}\sim\rho}[\|x\|_X^r] \geq 2^{-r-1}\, \mathbf{E}_{\boldsymbol{x},\boldsymbol{y}\sim\rho}[\|x - y\|_X^r]$. Suppose that after applying $f_{\phi,r}$ to points in $(\boldsymbol{x}, \boldsymbol{y}) \sim \mu$, the numerator of $\mathsf{R}(f_{\phi,r}(\mu), \|\cdot\|_Y)$ increases substantially in terms of the numerator of $\mathsf{R}(\mu, \|\cdot\|_X)$. This occurs when the upper bound on $\|f_{\phi,r}(\boldsymbol{x}) - f_{\phi,r}(\boldsymbol{y})\|_Y$ in terms of $\|\boldsymbol{x} - \boldsymbol{y}\|_X$ in (4.2) degrades substantially, which only occurs when $\|\boldsymbol{x}\|_X$ and $\|\boldsymbol{y}\|_X$ are large. However, this means $\mathbf{E}_{\boldsymbol{x}\sim\rho}[\|f_{\phi,r}(\boldsymbol{x})\|_Y]$, and because of (1.12), the denominator of $\mathsf{R}(f_{\phi,r}(\mu), \|\cdot\|_Y)$ is large. The next theorem closely follows an argument of [103], formalizing the intuition with a relaxed assumption of the centering of ρ, as well as for bounding $\mathsf{R}(f_{\phi,r}(\mu), \|\cdot\|_Y^s)$.

Theorem 30 (Matousek's Extrapolation [103, 16])**.** *Let* $X = (\mathbb{R}^d, \|\cdot\|_X)$ *and* $Y = (\mathbb{R}^{d'}, \|\cdot$

[38]We seek a vector $z \in \mathbb{R}^d$ which depends on ρ and satisfies $\mathbf{E}_{\boldsymbol{x}\sim\rho}[f_{\phi,r}(\boldsymbol{x} - z)] = 0$. If such a vector exists, it suffices to prove (1.9) under the centering assumption since $\mathsf{R}(\mu, \|\cdot\|_X)$ is invariant to translations. The existence of such a vector is a subtle matter which will be discussed shortly.

$\|_Y)$ *be any normed spaces, and suppose that for* $\alpha \in (0,1]$ *and* $c \geq 0$*, there exists an* α*-Holder homeomorphism* $\phi\colon S(X) \to S(Y)$ *with constant* c*. Let* μ *be any symmetric probability distribution supported on finitely many pairs of points in* $X \times X$*, and let* ρ *be the marginal of* μ*. Suppose that, for* $r > \alpha$ *and* $s \geq 0$*, the distribution* ρ *under* $f_{\phi,r}$ *is approximately centered, i.e.,*

$$\left\| \mathop{\mathbf{E}}_{\boldsymbol{x}\sim\rho}[f_{\phi,r}(\boldsymbol{x})] \right\|_Y \leq \left(\frac{1}{2^{s+1}} \cdot \mathop{\mathbf{E}}_{\boldsymbol{x}\sim\rho}[\|f_{\phi,r}(\boldsymbol{x})\|_Y^s] \right)^{1/s}, \tag{1.13}$$

then,

$$\mathsf{R}(f_{\phi,r}(\mu), \|\cdot\|_Y^s) \leq 2^{2s+2}\,(c+r)^s \cdot \mathsf{R}(\mu, \|\cdot\|_X^{sr})^{\alpha/r}. \tag{1.14}$$

For certain important cases, Theorem 30 obtains a bound on the cutting modulus better than by using the best bi-Lipschitz embeddings. Furthermore, we may employ techniques for constructing Holder homeomorphism between spheres of normed spaces (for example, see Chapter 9 of [29]) to obtain new ANN data structures. For a particularly elegant example of when Theorem 30 and Theorem 25 shine, consider ℓ_p spaces when $p \geq 2$. Here, bi-Lipschitz embeddings into ℓ_1 incur distortion $d^{1/2-1/p}$,[39] resulting in ANN data structure via Theorem 17 with polynomial approximation. On the other hand, a well-known map, the Mazur map ([104], and Section 9.1 of [29]), between the $S(\ell_p^d) \to S(\ell_1^d)$ given by[40]

$$(x_1, \ldots, x_d) \xmapsto{\phi} (\operatorname{sign}(x_1) \cdot |x_1|^p, \ldots, \operatorname{sign}(x_d) \cdot |x_d|^p), \tag{1.15}$$

is 1-Holder with constant at most $2p$. Consider setting $r = p$ and $s = 1$, and notice that $f_{\phi,r}\colon \mathbb{R}^d \to \mathbb{R}^d$ acts coordinate-wise as specified in (1.15). In order to find a translation $z \in \mathbb{R}^d$ which centers the dataset, we let $z_k \in \mathbb{R}$ where $\mathbf{E}_{\boldsymbol{x}\sim\rho}[\operatorname{sign}(\boldsymbol{x}_k - z_k)|\boldsymbol{x}_k - z_k|^p] = 0$ for every $k \in [d]$.[41] Therefore, we may assume (4.3) and apply Theorem 30, with $s = \alpha = 1$

[39]The identity map embeds $\ell_p \to \ell_2$ with distortion $d^{1/2-1/p}$, and matrix multiplication by Gaussian random variables embeds $\ell_2 \to \ell_1$ with small constant distortion. See Footnote 1.

[40]More generally, a Mazur map exists from any $\ell_p \to \ell_q$ norm, by analogously mapping $(x_1, \ldots, x_d) \to (\operatorname{sign}(x_1)|x_1|^{p/q}, \ldots, \operatorname{sign}(x_d), |x_d|^{p/q})$.

[41]To see that such a point $z_k \in \mathbb{R}$ exists, notice that $\mathbf{E}_{\boldsymbol{x}\sim\rho}[\operatorname{sign}(\boldsymbol{x}_k - z_k)|\boldsymbol{x}_k - z_k|^p]$ is continuous as a

and $r = p$, to obtain the bound $\mathsf{R}(f_{\phi,r}(\mu), \|\cdot\|_1) \le 24p \cdot \mathsf{R}(\mu, \|\cdot\|_p^p)^{1/p}$ and hence, show that the cutting modulus $\Xi(\ell_p^d, \varepsilon) \lesssim p/\varepsilon$.[42] Finally, we show the last ingredient, which shows a center for the distribution ρ always exists.

Theorem 31 (A Good Translation Always Exists [16])**.** *Let $X = (\mathbb{R}^d, \|\cdot\|_X)$ and $Y = (\mathbb{R}^{d'}, \|\cdot\|_Y)$ be any normed spaces. Suppose that for $\alpha \in (0, 1]$ and $c \ge 0$, the function $\phi\colon S(X) \to S(Y)$ is an α-Holder homeomorphism with constant c. For any probability distribution ρ supported on finitely many points in $\mathbb{R}^d$ and any $r > 1$, there exists a point $z \in \mathbb{R}^d$ such that*

$$\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\rho}[f_{\phi,r}(\boldsymbol{x} - z)] = 0. \tag{1.16}$$

Furthermore, if, for $R \in \mathbb{R}$, ρ is supported in $B_X(0, R)$, the point z may be chosen to satisfy $\|z\|_X \le 8^{r/\alpha} R$.

A similar approach, utilizing a properly generalized Mazur map, obtains ANN data structures for the Schatten-p norms with approximation $O(p)$. These are normed spaces defined on $\mathbb{R}^{d\times d}$, where vectors x are interpreted as matrices, and the Schatten-p norm is given by $\|x\|_{S_p} = (\sum_{i=1}^d |\sigma_i(x)|^p)^{1/p}$, where $\sigma_1(x), \ldots, \sigma_d(x)$ are the d singular values of x. The non-commutative Mazur map [121] maps $S(S_p) \to S(S_q)$ by mapping $x \mapsto U\Lambda^{p/q}V$, where $U\Lambda V$ is the singular value decomposition of x, and bounds on α and c analogous to that of $S(\ell_p) \to S(\ell_q)$ are attained for any $p, q \in [1, \infty)$. Recall that Schatten-2, i.e., the Frobenius norm, is equivalent to $\ell_2^{d^2}$, and we have established (1.9) when the right-hand side is $\mathsf{R}(\mu, \|\cdot\|_2^2)$. One would apply Theorem 46 to obtain a centering $z \in \mathbb{R}^d$, and then apply Theorem 30 with $X = S_p$, for some $p \in [1, \infty)$, $Y = S_2$, and $s = 2$.

---

function of z_k, and at some moment changes sign as z_k increases.

[42] We note that with a different approach, [114, 27] utilize the Mazur map as an embedding to design a data structure for $O(2^p)$-ANN over ℓ_p norms. The challenge with that approach is that distances between large vectors are poorly preserved; to overcome this, [114, 27] reduce the *c-bounded*-ANN problem, where dataset points lie in a bounded ball of radius $O(c)$.

1.5 Time Efficient Algorithms for Any Norm

Special care is need when defining the exact computational model for ANN data structures over arbitrary normed spaces $X = (\mathbb{R}^d, \|\cdot\|_X)$. For example, computing $\|x\|_X$ should not be the bottleneck, and instances considered should be numerically stable, i.e., discretizing coordinates to within w-bit precision should not incur massive changes to distances.[43] We consider the following assumptions:

1. We design data structures for $(1, c)$-ANN over X, i.e., the distance threshold is 1; the assumption is without loss of generality, since otherwise, we may rescale all points.
2. We assume the unit ball of the normed space X, B_X, satisfies $B_{\ell_2} \subset B_X \subset \sqrt{d}B_{\ell_2}$. Equivalently, this corresponds to the assumption that every $x \in \mathbb{R}^d$ satisfies $\|x\|_2/\sqrt{d} \leq \|x\|_X \leq \|x\|_2$. By John's theorem, one may apply a linear transformation (increasing running time by $O(d^2)$ times bit complexity of the linear transformation).
3. For simplicity, we consider the case all dataset points $P \subset \mathbb{R}^d$, as well as the query $q \in \mathbb{R}^d$, lie within a Euclidean ball of radius $\mathrm{poly}(d)$ from $0 \in \mathbb{R}^d$. This assumption is also without loss of generality, although some algorithmic preprocessing is required: we first build the data structure for ANN over ℓ_1 in Theorem 1, which uses a randomly chosen LSH function and utilize it to hash dataset points into buckets; for each non-empty bucket, we recenter the points to enforce a bounded diameter.
4. We assume access to an oracle which computes $\|x\|_X$ for any $x \in \mathbb{R}^d$.

We record the final theorem of this section, which shows that an algorithm which computes the radial extension of an α-Holder homeomorphism into ℓ_2 efficiently suffices for *time-efficient* ANN data structures.

Theorem 32 (Time Efficient Data Structures from Holder Homeomorphisms [16])**.** *Let $X = (\mathbb{R}^d, \|\cdot\|_X)$ be any normed space, and suppose that for $\alpha \in (0, 1]$, $c \geq 0$, and $d' \in \mathbb{N}$, there exists an α-Holder homeomorphism $\phi\colon S(X) \to S(\ell_2^{d'})$ with constant c. Suppose, that for $r \geq 1$, there exists an algorithm $\mathtt{ALG}_{\phi,r}$ which receives as input a vector $x \in \mathbb{R}^d$ and an*

[43] Recall that w is the word complexity of data structures, which will commonly be set to $O(\log n)$.

error parameter $\delta \in (0,1)$*, and outputs in time* $\mathrm{poly}(d, d', 1/\delta)$ *a vector* $y \in \mathbb{R}^{d'}$ *such that*

$$\|f_{\phi,r}(x) - y\|_2 \leq \delta.$$

Then, for any $\varepsilon \in (0,1)$*, there exists a data structure for ANN over* X *with space complexity* $\mathrm{poly}(d, d')n^{1+O(\varepsilon)} \log(1/\varepsilon)$ *and time complexity* $\mathrm{poly}(d, d')n^{O(\varepsilon)} \log(1/\varepsilon)$*, which achieves approximation* $(O((c+r)/\varepsilon))^{1/\alpha}$*.*

At a high level, the proof of Theorem 32 designs an algorithm by combining the bound of $\Xi(X, \varepsilon) \leq (O((c+r)/\varepsilon))^{1/\alpha}$ from Theorem 30 and Theorem 46, as well as the ideas behind Theorem 25. Similarly to Remark 28, the query algorithm behind Theorem 25 needs to walk down a decision tree, where the child node in the walk depends on whether the query point $q \in \mathbb{R}^d$ lies in a set $\mathbf{S} \subset \mathbb{R}^d$, where $\mathbf{S}$ was sampled from a distribution $\mathcal{H}$ made up of unions of (possibly infinite) low-conductance cuts guaranteed to exists from the bound on $\Xi(X, \varepsilon)$. For each $S \in \mathrm{supp}(\mathcal{H})$, there was some symmetric probability distribution μ supported on pairs of points $X \times X$ within distance at most 1 whose marginal ρ has no small-diameter dense clusters, and the set S was a balanced separator with respect to μ (i.e., $\mu(S, X \setminus S) \lesssim \varepsilon$, and $\min\{\rho(S), 1 - \rho(S)\} \geq 1/3$).

We make two observations when applying Theorem 30 and Theorem 46 to μ with marginal ρ. The first is that if ρ does not contain small-diameter clusters, the low-conductance cut S is specified by the inverse image of an axis aligned threshold after a translation of $f_{\phi,r}$. In other words, Theorem 30 and Theorem 46 imply that if we translate all points in μ according to some centering (as specified by Theorem 46), and then we apply $f_{\phi,r}$ to every point, we obtain a distribution μ' in $\ell_2^{d'}$ where $\mathsf{R}(\mu', \|\cdot\|_2^2)$ is small.[44] The claim is that there exists an axis aligned threshold in $\mathbb{R}^{d'}$ which has low conductance with respect to μ.[45] As a result, in order to determine whether the query q belongs to S or

[44] That occurs when we apply Theorem 30 with $s = 2$.

[45] Technically, we showed in Section 1.4.1 how to obtain axis aligned thresholds of low conductance given an upper bound on $\mathsf{R}(\mu', \|\cdot\|_1)$, while we have an upper bound on $\mathsf{R}(\mu', \|\cdot\|_2^2)$. The thing to notice is that we may apply Theorem 30 and Theorem 46 with the Mazur map $M_{2,1} \colon S(\ell_2^{d'}) \to S(\ell_1^{d'})$ (see Equation 1.15 and Section 4.1.1), and parameters $r = 2$, $s = 1$, $\alpha = 1$; the result is that if we translate the points in μ' and apply $M_{2,1}$ to every point, we obtain a distribution μ'' where $\mathsf{R}(\mu'', \|\cdot\|_1)$ is small, hence, giving an axis aligned threshold of low-conductance. Finally, notice that the Mazur map acts coordinate-wise and monotonically, i.e.,

not, one needs to translate q, apply $f_{\phi,r}$, and check the appropriate coordinate, which may be done efficiently given an algorithm for computing $f_{\phi,r}$. We note that one (minor) issue which adds some complication is that the algorithm $\texttt{ALG}_{\phi,r}$ does not compute $f_{\phi,r}$ exactly, but rather, outputs a vector y which within distance δ of $f_{\phi,r}$; hence, we must ensure that Theorem 30 and Theorem 46 are robust to perturbations on the order of $1/\mathrm{poly}(d^r, d')$.

The second observation is to address the following potential worry: the balanced separator may be the union of (possibly infinite) low-conductance cuts, and if each cut used a different translation, the time required to check whether a query lies in this union would be unbounded. We now use the fact Theorem 46 gave a translation where (1.16) is 0, and that (4.3) allowed some slack. The good news is that we may re-use translations. Specifically, consider any subset S where $\rho(S)$ is very small, and let μ' be the distribution given by $(\boldsymbol{x}, \boldsymbol{y}) \sim \mu$ conditioned on $\boldsymbol{x}, \boldsymbol{y} \notin S$, and let ρ' be the marginal of μ'. Since all points are bounded within a ball of radius R (in $\|\cdot\|_X$), we always have $\|f_{\phi,r}(\boldsymbol{x})\|_2 \leq R^r$, which means that

$$\left\| \mathop{\mathbf{E}}_{\boldsymbol{x}\sim\rho} [f_{\phi,r}(\boldsymbol{x})] - \mathop{\mathbf{E}}_{\boldsymbol{x}\sim\rho'} [f_{\phi,r}(\boldsymbol{x})] \right\|_2 \lesssim R^r \rho(S).$$

In particular, if $\rho(S)$ is $1/\mathrm{poly}(d^r)$, for sufficiently large polynomial, the left-hand side of (4.3) is small, while the right-hand side of (4.3) is large when there are no low-diameter dense clusters. Therefore, we may invoke Theorem 46 only after the having removed $1/\mathrm{poly}(d^r)$-fraction of points. When the same translation is used, the union of axis aligned thresholds is an axis aligned box (with possibly infinite sides). In summary, the resulting balanced separators are the union of polynomially many axis aligned boxes obtained after applying translations and $f_{\phi,r}$.

We state the final ingredient for applying Theorem 32 in order to obtain ANN data structures for any norm. In particular, we establish efficiently computable Holder homeomorphisms between unit spheres of normed spaces.

Theorem 33 (Efficient Holder Homeomorphism between Perturbations of Spheres [59, 29, 16])**.** *Let $X = (\mathbb{C}^d, \|\cdot\|_X)$ be any normed space, and fix $\alpha, \beta, \gamma \in (0, 1/2]$. There*

$M_{2,1}(x)_i \leq M_{2,1}(y)_i$ if and only if $x_i \leq y_i$. Thus, the inverse image of an axis aligned threshold under the transformation $M_{2,1}$ is also an axis aligned threshold.

exists two normed spaces $Y = (\mathbb{C}^d, \|\cdot\|_Y)$ and $Z = (\mathbb{C}^d, \|\cdot\|_Z)$, as well as an α-Holder homeomorphism $\phi\colon S(Y) \to S(Z)$ with constant $O(1/\sqrt{\gamma\beta})$ such that the following three items hold:

- *The normed space Y is a small perturbation of X: $B_Y \subset B_X \subset d^{\alpha+\beta(1-2\alpha)/2} B_Y$. Furthermore, there exists an algorithm which given as input a vector $x \in \mathbb{R}^d$ and $\delta > 0$, as well as oracle access to $\|\cdot\|_X$, outputs $\widehat{\eta} \in \mathbb{R}_{\geq 0}$ satisfying $\widehat{\eta} \in (\|x\|_Y - \delta, \|x\|_Y + \delta)$ in time $\mathrm{poly}(d, 1/\delta)$.*
- *The normed space Z is a small perturbation of ℓ_2^d: $B_{\ell_2^d} \subset B_Z \subset d^{\gamma(1-2\alpha)/2} B_{\ell_2^d}$.*
- *There exists an algorithm which given as input a vector $x \in \mathbb{R}^d$ and $\delta > 0$ such that $\|x\|_Y \in (1-\delta, 1+\delta)$, as well as oracle access to $\|\cdot\|_X$, outputs in time $\mathrm{poly}(d, 1/\delta)$, a vector $y \in \mathbb{R}^d$ satisfying*

$$\|y - \phi(x/\|x\|_Y)\|_Z \leq \delta.$$

We now apply Theorem 33 with

$$\alpha = \beta = \frac{1}{\sqrt{\log d}} \qquad \text{and} \qquad \gamma = \frac{1}{\log d}.$$

Let μ be any symmetric probability distribution supported on finitely many pairs $\mathbb{R}^d \times \mathbb{R}^d$. First, notice that by Theorem 33,

$$\mathsf{R}(\mu, \|\cdot\|_Y^2) \leq d^{2\alpha+\beta(1-2\alpha)} \cdot \mathsf{R}(\mu, \|\cdot\|_X^2). \tag{1.17}$$

Applying Theorem 30 with ϕ from Theorem 33, $r = 1$, and $s = 2$, we have that there exists a translation z, giving the distribution μ' such that

$$\mathsf{R}(f_{\phi,r}(\mu'), \|\cdot\|_Z^2) \lesssim 1/(\gamma\beta) \cdot \mathsf{R}(\mu, \|\cdot\|_Y^2)^\alpha. \tag{1.18}$$

Finally, once again, we use the fact that Z is a perturbation of $(\mathbb{C}^d, \|\cdot\|_2)$ to write

$$\mathsf{R}(f_{\phi,r}(\mu'), \|\cdot\|_2^2) \lesssim d^{\gamma(1-2\alpha)} \cdot \mathsf{R}(f_{\phi,r}(\mu'), \|\cdot\|_Z^2). \tag{1.19}$$

Combining (1.17), (1.18), and (1.19), as well as the parameter settings,

$$\begin{aligned}\mathsf{R}(f_{\phi,r}(\mu'), \|\cdot\|_2^2) &\lesssim d^{\gamma(1-2\alpha)} \cdot 1/(\gamma\beta) \cdot d^{2\alpha^2+\beta\alpha(1-2\alpha)} \mathsf{R}(\mu, \|\cdot\|_X^2)^\alpha \\ &= \mathrm{poly}(\log d) \cdot \mathsf{R}(\mu, \|\cdot\|_X^2)^{1/\sqrt{\log d}}.\end{aligned}$$

Chapter 2

ANN for General Symmetric Norms

In this chapter, we formally prove Theorem 21, Theorem 22, and Theorem 22, as stated in Section 1.1 in Chapter 1. We re-state the definition as well as the main theorem for the algorithm for convenience.

Definition 17. *A normed space $X = (\mathbb{R}^d, \|\cdot\|_X)$ is* symmetric *if $\|\cdot\|_X \colon \mathbb{R}^d \to \mathbb{R}^{\geq 0}$ is invariant under permutation and sign-flips of coordinates. In other words, if $x \in \mathbb{R}^d$ and $x^* \in \mathbb{R}^d$ is given by sorting the coordinates of $|x|$ in non-increasing value, satisfy $\|x\|_X = \|x^*\|_X$.*

Recall the main result is a $\mathrm{poly}(\log\log n)$-ANN data structure for an arbitrary symmetric norm. Formally, we prove the following theorem, which we instantiate with $d = n^{o(1)}$ so as to disregard polynomial dependencies in the dimension.

Theorem 34. *Fix $n \in \mathbb{N}$ and $d = n^{o(1)}$. For every d-dimensional symmetric norm $\|\cdot\|$, there exists a data structure for ANN over $\|\cdot\|$ for n-point datasets with approximation $(\log\log n)^{O(1)}$ space $n^{1+o(1)}$, and query time $n^{o(1)}$.*

We re-state two theorems which we use extensively in this chapter: Theorem 2, which gives a data structure for ℓ_∞, as well as Theorem 20, which shows how to combine many data structures for product spaces.

Theorem 35 ([78])**.** *For any $\rho \geq 0$, there exists a data structure for $(4\lceil \log_{1+\rho} \log(4d)\rceil + 1)$-ANN over ℓ_∞^d with space complexity $O(dn^{1+\rho})$ and time complexity $O(d \log n)$.*

Theorem 36 (Theorem 1 of [77] and Theorem 5.1.2 of [5])**.** *Let $k \in \mathbb{N}$, and suppose $\{(X_i, d_{X_i})\}_{i\in[k]}$ is a collection of metric spaces where each admits a c-ANN data structure using space at most $s(n) \geq n$ and query time at most $q(n)$ for some $c \geq 1$. For any $\delta > 0$,*

$\varepsilon \in (0,1)$, *and* $p \in [1,\infty)$, *there exists a data structure for* $O(c\varepsilon^{-1}\log_{1+\delta}\log n)$*-ANN over the* ℓ_p*-product of* X*'s*[1] $(\times_{i=1}^{k} X_i, d_{\ell_p^k(X_i)})$ *using space*[2] $O(ks(n)n^{\delta+\varepsilon^p})$*, and query time* $O(n^{\varepsilon^p}) \cdot O(q(n)\log(n) + k\log(n)\max_{j\in[k]}\log|X_k|)$.

2.1 An algorithm for Orlicz norms

Before showing a data structure for general symmetric norms, we give an algorithm for general Orlicz norms. We then show how to apply these ideas to top-k norms. This restricted setting has a simple analysis and illustrates one of the main techniques used. A similar approach was used in prior work to construct randomized embeddings of ℓ_p norms into ℓ_∞, and solve the ANN search problem; here we show that these techniques are in fact applicable in much greater generality.

Definition 18 (Orlicz Norms)**.** *Let* $G\colon \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ *be a non-zero convex function with* $G(0) = 0$*. For* $d \in \mathbb{N}$*, the Orlicz norm* $G = (\mathbb{R}^d, \|\cdot\|_G)$ *is defined by letting its unit ball* $B_G \subset \mathbb{R}^d$ *consist of all vectors* $x \in \mathbb{R}^d$ *with* $\sum_{i=1}^{d} G(|x_i|) \leq 1$.

Lemma 2.1.1. *Let* $\|\cdot\|_G$ *be an Orlicz norm. For every* $D, \alpha > 1$ *and every* $\mu \in (0, 1/2)$ *there exists a randomized linear map* $\boldsymbol{f}\colon \mathbb{R}^d \to \mathbb{R}^d$ *such that for every* $x \in \mathbb{R}^d$*:*

- *if* $\|x\|_G \leq 1$*, then* $\mathbf{Pr}_{\boldsymbol{f}}\left[\left\|\boldsymbol{f}(x)\right\|_\infty \leq 1\right] \geq \mu$*;*
- *if* $\|x\|_G > \alpha D$*, then* $\mathbf{Pr}_{\boldsymbol{f}}\left[\left\|\boldsymbol{f}(x)\right\|_\infty > D\right] \geq 1 - \mu^\alpha$.

Proof. Let the distribution $\mathcal{D}$ over $\mathbb{R}_+$ have the following CDF $F : \mathbb{R}_+ \to [0,1]$:

$$F(t) = \mathop{\mathbf{Pr}}_{u\sim\mathcal{D}}[u \leq t] = 1 - \mu^{G(t)}.$$

[1] Given a collection of metric spaces $\{(X_i, d_{X_i})\}_{i\in[k]}$ and $p \in [1,\infty]$, the ℓ_p-product of X's the metric space defined over the cartesian product $\times_{i=1}^{k} X_i$, where distances between points $(a_1,\ldots,a_k), (b_1,\ldots,b_k)$ is given by $d_{\ell_p^k(X_i)}(a,b) = (\sum_{i=1}^{k} d_{X_i}(a_i,b_i)^p)^{1/p}$.

[2] We remark that [77], which first obtained this Theorem for $p = \infty$, states a weaker bound in the space complexity. [8] noticed a better analysis gave the corresponding space bound. See also, Appendix A of [12].

Consider the following randomized linear map $\boldsymbol{f} : \mathbb{R}^d \to \mathbb{R}^d$:

$$(x_1, x_2, \ldots, x_d) \overset{\boldsymbol{f}}{\mapsto} \left(\frac{x_1}{\boldsymbol{u}_1}, \frac{x_2}{\boldsymbol{u}_2}, \ldots, \frac{x_d}{\boldsymbol{u}_d} \right)$$

where $\boldsymbol{u}_1, \ldots, \boldsymbol{u}_d \sim \mathcal{D}$ are i.i.d. samples from $\mathcal{D}$. Suppose that $\|x\|_G \leq 1$. Then, $\sum_{i=1}^d G(|x_i|) \leq 1$. This, in turn, implies:

$$\mathop{\mathbf{Pr}}_{\boldsymbol{f}} \left[\|\boldsymbol{f}(x)\|_\infty \leq 1 \right] = \prod_{i=1}^d \mathop{\mathbf{Pr}}_{\boldsymbol{u}_i \sim \mathcal{D}} \left[\left| \frac{x_i}{\boldsymbol{u}_i} \right| \leq 1 \right] = \prod_{i=1}^d \mu^{G(|x_i|)} = \mu^{\sum_{i=1}^d G(|x_i|)} \geq \mu.$$

Now suppose that $\|x\|_G > \alpha D$. This, together with the convexity of $G(\cdot)$, implies:

$$\sum_{i=1}^d G\left(\frac{|x_i|}{D} \right) \geq (1-\alpha) G(0) + \alpha \cdot \sum_{i=1}^d G\left(\frac{|x_i|}{\alpha D} \right) \geq \alpha.$$

Thus, we have:

$$\mathop{\mathbf{Pr}}_{\boldsymbol{f}} \left[\|\boldsymbol{f}(x)\|_\infty \leq D \right] = \prod_{i=1}^d \mathop{\mathbf{Pr}}_{\boldsymbol{u}_i \sim \mathcal{D}} \left[\left| \frac{x_i}{\boldsymbol{u}_i} \right| \leq D \right] = \prod_{i=1}^d \mu^{G(|x_i|/D)} = \mu^{\sum_{i=1}^d G(|x_i|/D)} \leq \mu^\alpha.$$

□

Theorem 37 (ANN for Orlicz norms)**.** *For every d-dimensional Orlicz norm $\|\cdot\|_G$ and every $\varepsilon \in (0, 1/2)$, there exists a data structure for ANN over $\|\cdot\|_G$, which achieves approximation $O\left(\frac{\log\log d}{\varepsilon^2}\right)$ using space $O\left(dn^{1+\varepsilon}\right)$ and query time $O\left(dn^{\varepsilon}\right)$.*

Proof. Let $P \subset \mathbb{R}^d$ be a dataset of n points. Consider the data structure which does the following:

1. For all $1 \leq i \leq n^\varepsilon$, we independently apply the randomized linear map $\boldsymbol{f}$ from Lemma 2.1.1 with parameters $\mu = n^{-\varepsilon}$, $D = O\left(\frac{\log\log d}{\varepsilon}\right)$, and $\alpha = \frac{2}{\varepsilon}$. We define

$$\mathbf{P}_i = \{\boldsymbol{f}_i(x) \mid x \in P\}$$

   to be the image of the dataset under $\boldsymbol{f}_i$, where $\boldsymbol{f}_i$ is the i-th independent copy of $\boldsymbol{f}$.

2. For each $1 \leq i \leq n^\varepsilon$, we use Theorem 2 to build a data structure for ANN over ℓ_∞

with approximation D for dataset $\mathbf{P}_i$. We refer to the i-th data structure as T_i.

Each T_i occupies space $O(dn^{1+\varepsilon})$ and achieves approximation D with query time $O(d \log n)$. To answer a query $q \in \mathbb{R}^d$, we query T_i with $\boldsymbol{f}_i(q)$ for each $i \in [n^\varepsilon]$. Let x_i be the point returned by T_i, and let $p_i \in P$ be the pre-image of x_i under $\boldsymbol{f}_i$, so that $\boldsymbol{f}_i(p_i) = x_i$. If for some T_i, the point returned satisfies $\|p_i - q\|_G \leq \alpha D$, then we return p_i.

- If there exists some $p \in P$ with $\|p - q\|_G \leq 1$, then by Lemma 2.1.1, with probability $1 - (1 - n^{-\varepsilon})^{n^\varepsilon} \geq \frac{3}{5}$, some $\boldsymbol{f}_i$ has $\|\boldsymbol{f}_i(p - q)\|_\infty \leq 1$. Since $\boldsymbol{f}_i$ is linear, $\|\boldsymbol{f}_i(p) - \boldsymbol{f}_i(q)\|_\infty \leq 1$ as well.
- Let $i \in [n^\varepsilon]$ be an index where some $p \in P$ with $\|p - q\|_G \leq 1$ has $\|\boldsymbol{f}_i(p) - \boldsymbol{f}_i(q)\|_\infty \leq 1$. Every other $p' \in P$ with $\|p' - q\|_G \geq \alpha D$ satisfies

$$\mathbf{Pr}\Big[\|\boldsymbol{f}_i(p') - \boldsymbol{f}_i(q)\|_\infty \leq D\Big] \leq \frac{1}{n^2}.$$

  A union bound over at most n points with distance greater than αD to q shows that except with probability at most $\frac{1}{n}$, T_i returns some $p_i \in P$ with $\|p_i - q\|_G \leq \alpha D$. Thus, the total probability of success of the data structure is at least $\frac{3}{5} - \frac{1}{n}$.

The total query time is $O\left(dn^\varepsilon \cdot \log n\right)$ and the total space used is $O\left(dn^{1+2\varepsilon}\right)$. This data structure achieves approximation $\alpha D = O\left(\frac{\log \log d}{\varepsilon^2}\right)$. Decreasing ε by a constant factor, we get the desired guarantees. □

Remark. The construction of the randomized embedding in Lemma 2.1.1 and the data structure from Theorem 37 work in a somewhat more general setting, rather than just for Orlicz norms. For a fixed norm $\|\cdot\|$, we can build a randomized map $\boldsymbol{f}\colon \mathbb{R}^d \to \mathbb{R}^d$ with the guarantees of Lemma 2.1.1 if there exists a non-decreasing $G\colon \mathbb{R}_+ \to \mathbb{R}_+$ where $G(0) = 0$, $G(t) \to \infty$ as $t \to \infty$, and for every $x \in \mathbb{R}^d$:

- if $\|x\| \leq 1$, then $\sum_{i=1}^d G(|x_i|) \leq 1$, and
- if $\|x\| \geq \alpha D$, then $\sum_{i=1}^d G\left(\frac{|x_i|}{D}\right) \geq \alpha$.

The data structure itself just requires the existence of a randomized linear map satisfying the conditions of Lemma 2.1.1.

We now describe how to obtain a data structure for ANN for any top-k norm, $(\mathbb{R}^d, \|\cdot\|_{T(k)})$, where $\|x\|_{T(k)}$ computes the sum of the highest (in magnitude) k coordinates of x. In particular, for any $x \in \mathbb{R}^d$, let $x^* \in \mathbb{R}^d$ be the vector given by a non-increasing re-arrangement of coordinates.[3] Then, $\|x\|_{T(k)} = \sum_{i=1}^{k} |x_i^*|$.

Lemma 2.1.2 (Randomized Embedding of Top-k into ℓ_∞)**.** *Fix any $k \in [d]$. For every $D, \alpha > 1$ and every $\mu \in (0, 1/2)$, there exists a randomized linear map $\boldsymbol{f}\colon \mathbb{R}^d \to \mathbb{R}^d$ such that for every $x \in \mathbb{R}^d$:*

- *if $\|x\|_{T(k)} \le 1$, then $\mathbf{Pr}_{\boldsymbol{f}}\left[\|\boldsymbol{f}(x)\|_\infty \le 1\right] \ge \mu$;*
- *if $\|x\|_{T(k)} > \alpha D$, then $\mathbf{Pr}_{\boldsymbol{f}}\left[\|\boldsymbol{f}(x)\|_\infty > D\right] \ge 1 - \mu^{\alpha-1}$.*

Proof. We define $G\colon \mathbb{R}_+ \to \mathbb{R}_+$ where for every $x \in \mathbb{R}^d$,

$$G(t) = t \cdot \chi_{[\frac{1}{k}, \infty)}(t)$$

If $\|x\|_{T(k)} \le 1$, there are at most k coordinates where $|x_i| \ge \frac{1}{k}$. Therefore, $\sum_{i=1}^{d} G(|x_i|) \le \|x\|_{T(k)} \le 1$. If $\|x\|_{T(k)} \ge \alpha D$, then $\sum_{i=1}^{k} |x_i^*| \ge \alpha D$. Therefore, $\sum_{i=1}^{d} G\left(\frac{|x_i^*|}{D}\right) \ge \sum_{i=1}^{k} G\left(\frac{|x_i^*|}{D}\right) \ge \alpha - 1$. The proof now follows in the same way as Lemma 2.1.1. □

Lemma 2.1.2 gives us a data structure for any top-k norm with approximation $O(\log \log d)$ applying Theorem 37.

2.2 Embedding symmetric norms into product spaces

We construct an embedding of general symmetric norms into product spaces of top-k norms. To state the main result of this section, we need the following definition.

Definition 19. *For any $c_1, \ldots, c_d \ge 0$, let $\ell_1^d(T^{(c)}) \subset \mathbb{R}^{d^2}$ be the space given by the seminorm $\|\cdot\|_{T,1}^{(c)} \cdot \mathbb{R}^{d^2} \to \mathbb{R}$ where for $x = (x_1, \ldots, x_d) \in \mathbb{R}^{d^2}$ and $x_1, \ldots, x_d \in \mathbb{R}^d$:*

$$\|x\|_{T,1}^{(c)} = \sum_{k=1}^{d} c_k \|x_k\|_{T(k)}.$$

[3] In other words, for $x \in \mathbb{R}^d$, the vector $x^* \in \mathbb{R}^d$ is such that $x_1^* \ge \cdots \ge x_d^*$ and there exists a bijection $\pi\colon [d] \to [d]$ where for all $i \in [d]$, $x_i = x_{\pi(i)}^*$.

Theorem 38 (Re-statement of Theorem 19)**.** *For any constant $\delta \in (0, 1/2)$, any symmetric norm $\|\cdot\|_X \colon \mathbb{R}^d \to \mathbb{R}$ can be embedded linearly with distortion $1+\delta$ into $\ell_\infty^t(\ell_1^d(T^{(c)}))$ where $t = d^{O(\log(1/\delta)\delta^{-1})}$. In particular, there exists $c \in \mathbb{R}_+^{t\times d}$ such that for every $x \in \mathbb{R}^d$,*

$$(1-\delta)\|x\|_X \leq \max_{i\in[t]} \left(\sum_{k=1}^{d} c_{i,k}\|x\|_{T(k)} \right) \leq (1+\delta)\|x\|_X. \tag{2.1}$$

The vectors in $\ell_\infty^t(\ell_1^d(T^{(c)})) \subset \mathbb{R}^{td^2}$ can be broken up into td blocks of d coordinates each. The embedding referenced above will simply map $x \in \mathbb{R}^d$ into $\mathbb{R}^{td^2}$ by making each of the td many blocks equal to a copy of x. The non-trivial part of the above theorem is setting the constants $c_{i,k}$ for $i \in [t]$ and $k \in [d]$ so (2.1) holds. Before going on to give the proof of Theorem 38, we establish some definitions and propositions which will be used in the proof. For the subsequent sections, let $\beta \in (1, 2)$ be considered a constant close to 1.

Definition 20 (Levels and Level Vectors)**.** *For any fixed vector $x \in \mathbb{R}^d$ and any $k \in \mathbb{Z}$, we define* level k with respect to x *as $B_k(x) = \{i \in [d] \mid \beta^{-k-1} < |x_i| \leq \beta^{-k}\}$. Additionally, we let $b_k(x) = |B_k(x)|$ be the* size of level k with respect to x*. The* level vector *of x, $V(x) \in \mathbb{R}^d$ is given by*

$$V(x) = (\underbrace{\beta^k, \ldots, \beta^k}_{b_{-k}(x) \text{ times}}, \underbrace{\beta^{k-1}, \ldots, \beta^{k-1}}_{b_{-k+1}(x) \text{ times}}, \ldots, \underbrace{\beta^{-k}, \ldots, \beta^{-k}}_{b_k(x) \text{ times}}, 0, \ldots 0)$$

where k is some integer such that all non-zero coordinates lie in some level between $-k$ and k. We say the i-th level vector $V_i(x) \in \mathbb{R}^d$ *is given by*

$$V_i(x) = (\underbrace{\beta^{-i}, \ldots, \beta^{-i}}_{b_i(x) \text{ times}}, 0, \ldots, 0).$$

The notation used for level vectors appears in [35]; however, we refer to level k as the coordinates of x lying in $(\beta^{-k-1}, \beta^{-k}]$; whereas [35] refers to level k as the coordinates of x lying in $[\beta^{k-1}, \beta^k)$.

Definition 21. *Fix some $\tau > 0$. For any vector $x \in \mathbb{R}^d$, let $C(x) \in \mathbb{R}^d$ be the vector where*

each $i \in [d]$ *sets*

$$C(x)_i = \begin{cases} x_i & |x_i| \geq \tau \\ 0 & |x_i| < \tau \end{cases}.$$

Proposition 2.2.1 (Proposition 3.4 in [35])**.** *Let* $\|\cdot\|_X$ *be any symmetric norm and* $x \in \mathbb{R}^d$ *be any vector. Then*

$$\frac{1}{\beta}\|V(x)\|_X \leq \|x\|_X \leq \|V(x)\|_X.$$

Proposition 2.2.2. *Let* $\|\cdot\|_X$ *be any symmetric norm. For any vector* $x \in \mathbb{R}^d$*,*

$$\|x\|_X - \tau d \leq \|C(x)\|_X \leq \|x\|_X.$$

Proof. Note that x weakly majorizes $C(x)$, so $\|C(x)\|_X \leq \|x\|_X$. For the other direction, let $v = x - C(x)$. Then v has all coordinates with absolute value at most τ, so $\tau d\xi^{(1)}$ weakly majorizes v. Therefore,

$$\|x\|_X \leq \|C(x)\|_X + \|v\|_X \leq \|C(x)\|_X + \tau d.$$

□

Intuitively, the above two propositions say that up to multiplicative loss β and additive loss τd in the norm of the vector, we may assume that all coordinates are exactly β^j for $j \geq \log_\beta(\tau)$. Thus, if $x \in \mathbb{R}^d$, then

$$\|x\|_X - \tau d \leq \|V(C(x))\|_X \leq \beta\|x\|_X.$$

If additionally, we let $\tau = \frac{\beta}{d^2}$, so when $\|x\|_X \leq 1$ there are at most $2\log_\beta d$ non-empty levels in $V(C(x))$.

Definition 22 (Rounded counts vector)**.** *Fix any level vector* $x \in \mathbb{R}^d$*. The* rounded counts vector *of* x*,* $W(x) \in \mathbb{R}^d$ *is given by* y *where the* $y \in \mathbb{R}^d$ *is constructed using the following procedure:*

1: *Initialize* $y = (0, \ldots, 0) \in \mathbb{R}^d$ *and* $c = d$*.*

2: ***for*** $k = -\infty, \ldots, 2\log_\beta(d) - 1$ ***do***

3: ***if*** $b_k(x) \geq 0$ ***then***

4: *Let* $j \in \mathbb{Z}_+$ *be the integer where* $\beta^{j-1} < b_k(x) \leq \beta^j$.

5: ***if*** $c \geq \lfloor \beta^j \rfloor$ ***then***

6: *Set the first* $\lfloor \beta^j \rfloor$ *zero-coordinates of* y *with* β^{-k}. *Update* $c \leftarrow c - \lfloor \beta^{-k} \rfloor$.

7: ***end if***

8: ***end if***

9: ***end for***

10: *Return* y

Intuitively, $W(x)$ represents the level vector of x where we ignore coordinates smaller than $\frac{\beta}{d^2}$, and additionally, we round the counts of coordinates to powers of β.

Lemma 2.2.3. *For every vector* $x \in \mathbb{R}^d$ *and any symmetric norm* $\|\cdot\|_X$,

$$\|x\|_X - \tau d \leq \|W(V(C(x)))\|_X \leq \beta^2 \|x\|_X.$$

Proof. The bound $\|x\|_X - \tau d \leq \|W(V(C(x)))\|_X$ follows by combining Proposition 2.2.1 and Proposition 2.2.2, as well as the monotonicity of norms. The bound $\|W(V(C(x)))\|_X \leq \beta^2 \|x\|_X$ follows from Proposition 2.2.1, Proposition 2.2.2, as well as Lemma 3.5 from [35]. □

In order to simplify notation, we let $R \colon \mathbb{R}^d \to \mathbb{R}^d$ given by $R(x) = W(V(C(x)))$.

Definition 23. *Let the set* $\mathcal{L} \subset \mathbb{R}^d_+$ *be given by*

$$\mathcal{L} = \{y \in \mathbb{R}^d_+ \mid y_1 \geq \ldots y_d \geq 0\}.$$

Additionally, for an arbitrary symmetric norm $\|\cdot\|_X$ *with dual norm* $\|\cdot\|_{X^*}$, *we let the set* $\mathcal{R} \subset \mathcal{L}$ *be given by*

$$\mathcal{R} = \{R(y) \in \mathbb{R}^d_+ \mid y \in \mathcal{L} \cap B_{X^*}\}.$$

Definition 24. *Fix a vector* $y \in \mathcal{L} \setminus \{0\}$ *(*y *has non-negative, non-increasing coordinates). Let the* maximal seminorm *with respect to* y, $\|\cdot\|_y \colon \mathbb{R}^d \to \mathbb{R}$ *be the seminorm where for*

every $x \in \mathbb{R}^d$*,*

$$\|x\|_y = \langle |x^*|, y \rangle.$$

We first show there exists some setting of $c \in \mathbb{R}^d$ such that we may compute $\|x\|_y$ as $\ell_1^d(T^{(c)})$.

Lemma 2.2.4. *For every vector* $y \in \mathcal{L} \setminus \{0\}$*, there exists* $c_1, \ldots, c_d \geq 0$ *where for all* $x \in \mathbb{R}^d$*,*

$$\|x\|_y = \|x\|_{T,1}^{(c)}.$$

Proof. For $k \in [d]$, we let $c_k = y_k - y_{k+1}$, where $y_{d+1} = 0$.

$$\langle |x^*|, y \rangle = \sum_{i=1}^{d} |x_i^*| y_i = \sum_{i=1}^{d} |x_i^*| \left(\sum_{k=i}^{d} c_k \right) = \sum_{k=1}^{d} c_k \left(\sum_{i=1}^{k} |x_i^*| \right) = \sum_{k=1}^{d} c_k \|x\|_{T(k)}$$

□

Given Lemma 2.2.4, it suffices to show that for an arbitrary symmetric norm $\|\cdot\|_X$, we may compute $\|x\|_X$ (with some distortion) as a maximum over many maximal seminorms. In the following lemma, we show that taking the maximum over maximal norms from $\mathcal{R}$ suffices, but gives sub-optimal parameters. We then improve the parameters to prove Theorem 38.

Lemma 2.2.5. *Let* $\|\cdot\|_X$ *be an arbitrary symmetric norm and let* $\|\cdot\|_{X^*}$ *be its dual norm. Then for any* $\|x\|_X \leq 1$*,*

$$\|x\|_X - \tau d \leq \max_{y \in \mathcal{R}} \|x\|_y \leq \beta^2 \|x\|_X.$$

Proof. Without loss of generality, we rescale the norm so that $\|e_1\|_{X^*} = 1$, where e_1 is the first standard basis vector. Consider any $x \in \mathbb{R}^d$ with $\|x\|_X \leq 1$. Then since $\|\cdot\|_X$ is symmetric, we may assume without loss of generality that all coordinates of x are non-negative and in non-increasing order. Thus for each $y \in \mathcal{L} \cap \{0\}$, we have $\|x\|_y = \langle x, y \rangle$.

The lower bound simply follows from the fact that $R(z)$, other than coordinates less than

τ, is monotonically above z, and all coordinates in x are non-negative. More specifically,

$$\|x\|_X = \sup_{z \in \mathcal{L} \cap B_{X^*}} \langle x, z \rangle \leq \sup_{z \in \mathcal{L} \cap B_{X^*}} \langle x, R(z) \rangle + \tau d = \max_{y \in \mathcal{R}} \langle x, y \rangle + \tau d,$$

where τd comes from the fact that because $\|x\|_X \leq 1$, every coordinate of x is at most 1. On the other hand, we have

$$\max_{y \in \mathcal{R}} \langle x, y \rangle = \beta^2 \max_{y \in \mathcal{R}} \langle x, \frac{y}{\beta^2} \rangle \leq \beta^2 \sup_{z \in B_{X^*}} \langle x, z \rangle = \beta^2 \|x\|_X,$$

where we used the fact that $\|y/\beta^2\|_{X^*} \leq 1$ by Lemma 2.2.3. □

Given Lemma 2.2.5, it follows that we may linearly embed X into $\ell_\infty^t(\ell_1^d(T^{(c)}))$ where $t = |\mathcal{R}|$, with distortion $\beta^2/(1 - \tau d) \leq \beta^3$ (where we used the fact $\tau = \beta/d$ and that $1 + \beta/d \leq \beta$ for a large enough d). The embedding follows by copying the vector x into the t spaces $\ell_1^d(T^{(c)})$ corresponding to each vector $y \in \mathcal{R}$ given in Lemma 2.2.4. The one caveat is that this embedding requires t copies of $\ell_1^d(T^{(c)})$, and t is as large as $(\log_\beta d + 1)^{2 \log_\beta d} = d^{O(\log \log d)}$. This is because there are at most $2 \log_\beta d$ many levels, and each contains has number of coordinates being some value in $\{\beta^i\}_{i=0}^{\log_\beta d}$. Thus, our algorithm becomes inefficient once $d \geq 2^{\omega\left(\frac{\log n}{\log \log n}\right)}$.

In order to avoid this problem, we will make the embedding more efficient by showing that we do not need all of $\mathcal{R}$, but rather a fine net of $\mathcal{R}$. In addition, our net will be of polynomial size in the dimension, which gives an efficient algorithm for all $\omega(\log n) \leq d \leq n^{o(1)}$. We first show that it suffices to consider fine nets of $\mathcal{R}$, and then build a fine net of $\mathcal{R}$ of size $\mathrm{poly}(d)$.

Lemma 2.2.6. *Fix an $\gamma \in (0, 1/2)$. Let $\|\cdot\|_X$ be an arbitrary symmetric norm and $\|\cdot\|_{X^*}$ be its dual norm. If N is a γ-net of $\mathcal{R}$ with respect to distance given by $\|\cdot\|_{X^*}$, then*

$$(1 - \gamma - \tau d)\|x\|_X \leq \max_{y \in N} \|x\|_y \leq (\beta^2 + \gamma)\|x\|_X$$

Proof. Since the embedding we build is linear, it suffices to show that every vector $x \in \mathbb{R}^d$

with $\|x\|_X = 1$ has

$$1 - \gamma - \tau d \leq \max_{y \in N} \|x\|_y \leq (\beta^2 + \gamma).$$

Consider a fixed vector $x \in \mathbb{R}^d$ with $\|x\|_X = 1$. Additionally, we may assume the coordinates of x are non-negative and in non-increasing order. We simply follow the computation:

$$\|x\|_X = \||x^*|\|_X \leq \max_{y \in \mathcal{R}} \langle |x^*|, y\rangle + \tau d = \max_{y \in N} (\langle |x^*|, y\rangle + \langle |x^*|, v\rangle) + \tau d \leq \max_{y \in N} \langle |x^*|, y\rangle + \gamma \|x\|_X + \tau d,$$

where we used Lemma 2.2.5 and the fact that $\|v\|_{X^*} \leq \gamma$ in a γ-net of $\mathcal{R}$ with respect to the distance given by $\|\cdot\|_{X^*}$. On the other hand,

$$\max_{y \in N} \|x\|_y = \max_{y \in \mathcal{R}} (\langle |x^*|, y\rangle + \langle |x^*|, v\rangle) \leq \max_{y \in \mathcal{R}} \|x\|_y + \gamma \|x\|_X \leq (\beta^2 + \gamma)\|x\|_X,$$

where again, we used Lemma 2.2.5 and the fact that $\|v\|_{X^*} \leq \gamma$. □

Finally, we conclude the theorem by providing a γ-net for $\mathcal{R}$ of size $d^{O(\log(1/\gamma)\gamma^{-1})}$.

Lemma 2.2.7. *Fix any symmetric space X with dual X^*. There exists an $8(\beta - 1)$-net of size $d^{O(\log(1/(\beta-1))/\log \beta)}$ for $\mathcal{R}$ with respect to distances given by $\|\cdot\|_{X^*}$.*

We defer the proof of Lemma 2.2.7 to the next section. The proof of Theorem 38 follows by combining Lemma 2.2.4, Lemma 2.2.6, and Lemma 2.2.7. In particular, given a $((\beta - 1)/8)$-net of $\mathcal{R}$, we get an embedding with distortion at most $(\beta^2 + 8(\beta - 1))(1 + (8(\beta - 1) + \tau d)^2)$ from Lemma 2.2.6. We let $\tau = \beta/d^2$ and $\beta = \sqrt{1 + \delta/100}$ to get the desired linear embedding with distortion $1 + \delta$. We now proceed to proving Lemma 2.2.7, which gives the desired upper bound on the size of the net.

2.2.1 Proof of Lemma 2.2.7: bounding the net size

We now give an upper bound on the size of a fine net of $\mathcal{R}$. We proceed by constructing a further simplification of $R(x)$. Intuitively we show that one can ignore the higher levels if there are fewer coordinates in the higher levels than some lower level.

Lemma 2.2.8. *Let $\|\cdot\|_X$ be a symmetric norm. Consider any nonnegative vector $x \in \mathbb{R}^d_+$ as well as two indices $u, v \in [d]$. Let $y \in \mathbb{R}^d_+$ be the vector with:*

$$y_k = \begin{cases} x_k & k \in [d] \setminus \{u, v\} \\ x_u + x_v & k = u \\ 0 & k = v \end{cases}.$$

Then $\|y\|_X \geq \|x\|_X$.

Proof. Consider the vector $z \in B_{X^*}$ where $\langle x, z\rangle = \|x\|_X$. Now, we let $z' \in \mathbb{R}^d$ be given by

$$z'_k = \begin{cases} z_k & k \in [d] \setminus \{u, v\} \\ \max\{z_u, z_v\} & k = u \\ \min\{z_u, z_v\} & k = v \end{cases}$$

Note that z' is a permutation of z, so $z' \in B_{X^*}$. Now,

$$\langle y, z'\rangle = (x_u + x_v)\max\{z_u, z_v\} + \sum_{k\in[d]\setminus\{u,v\}} x_k z_k \geq \sum_{k\in[d]} x_k z_k = \langle x, z\rangle = \|x\|_X.$$

□

Definition 25. *Consider a vector $x \in \mathcal{R}$. We define the* simplified rounded vector $S(x)$ *as the vector returned by the following procedure.*

Initialize $z = x$
for $k = 0, 1, \ldots, 2\log_\beta(d) - 1$ ***do***
    if $b_k(z) \leq \max_{j<k+3\log_\beta(\beta-1)} b_j(z)$ ***then***
        Set all coordinates of z of value β^{-k} to 0 i.e. set $b_k(z) = 0$.
    end if
end for
Sort the coordinates of z in non-increasing order and return z.

Next we show that the simplified rounded vector is close to the rounded counts vector.

Lemma 2.2.9. *Let $\|\cdot\|_X$ be a symmetric norm and let $x \in \mathcal{R}$. Then $\|S(x) - x\|_X \leq 2(\beta-1)\|x\|_X$.*

Proof. Consider some $k \in [2\log_\beta d - 1]$ and let $C_k(x) \subset [d]$ be set of coordinates where x is at level k and does not equal $S(x) = z$, i.e.,

$$C_k(x) = \{i \in [d] \mid x_i = \beta^{-k} \text{ and } x_i \neq z_i\}.$$

Additionally, for each $k \in [2\log_\beta d - 1]$, let $T_k \subset [d]$ be the coordinates at level k in x which trigger line 3 of $S(x)$, and thus become 0's in z (we do not need to consider the case $k = 0$ since line 3 never triggers, in fact, we do not need to consider $k \in [-3\log_\beta(b-1)]$ either). In other words,

$$T_k(x) = \{i \in [d] \mid x_i = \beta^{-k} \text{ and at iteration } k \text{ of } S(x), b_k(z) \leq \max_{j<k+3\log_\beta(\beta-1)} b_j(z)\}.$$

Note that $T_1(x), \ldots, T_{2\log_\beta d-1}(x)$ are all disjoint, and $|C_k(x)| \leq \sum_{j\in[k]} |T_j(x)|$, since whenever we zero out coordinates in levels less than or equal to k, $S(x)$ will shift the coordinates when we sort, causing $x_i \neq z_i$. Thus, we may consider an injection $s_k \colon C_k(x) \to \bigcup_{j\in[k]} T_j(x)$, which charges coordinates in $C_k(x)$ to coordinates which were zeroed out in line 3 of $S(x)$.

Additionally, for each $j \in [2\log_\beta d - 1]$ where $T_j(x) \neq \emptyset$, we let q_j be the integer between 0 and $j + 3\log_\beta(\beta-1)$ which triggered line 3 of $S(x)$ at $k = j$. More specifically, $0 \leq q_j \leq j + 3\log_\beta(\beta-1)$ is the integer for which $b_j(x) \leq b_{q_j}(x)$.

Finally, for each $j \in [2\log_\beta d - 1]$ where $T_j(x) \neq \emptyset$, we let $g_j \colon T_j(x) \to B_{q_j}(x)$ (recall that $B_{q_j}(x) \subset [d]$ are the indices of x at level q_j) be an arbitrary injection. Such an injection exists because $b_j(x) \leq b_{q_j}(x)$. We may consider the mapping $F \colon \bigcup_{k\in[2\log_\beta d-1]} C_k(x) \to [d]$ where

$$F(i) = g_j(s_k(i)) \qquad \text{where } k \text{ and } j \text{ are such that } i \in C_k(x) \text{ and } s_k(i) \in T_j(x).$$

See Figure 2.2.1 for an example of a coordinate being mapped by F. Let y be the vector where we "aggregate" coordinates of $\bigcup_{k\in[2\log_\beta(d)-1]} C_k(x)$ of x according to the map F

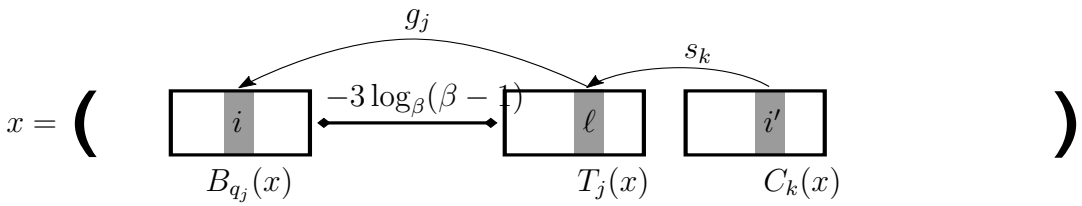

Figure 2.1: Example mapping of particular levels of x with F showing aggregation of coordinates. The coordinate $i' \in C_k(x)$ belongs to level k and will be non-zero in $x - S(x)$. In particular, coordinate $i' \in C_k(x)$ is mapped via s_k to coordinate $\ell \in T_j(x)$, which is β^{-j} in x and 0 in $S(x)$. Then coordinate ℓ is mapped to $i \in B_{q_j}(x)$, where q_j is the level below j with $b_{q_j} \geq b_j$. Thus, the composed map F sends i' to i.

according to Lemma 2.2.8. In particular, we define $y \in \mathbb{R}^d$ where for $i \in [d]$, we let

$$y_i = \sum_{i' \in F^{-1}(i)} x_{i'}.$$

Note that for each $i \in [d]$, $0 \leq (x - z)_i \leq x_i$, and $\bigcup_{k \in [2\log_\beta(d)-1]} C_k(x) \subset [d]$ are the non-zero coordinates of $x - z$. Thus, from Lemma 2.2.8, we conclude that $\|x - z\|_X \leq \|y\|_X$. We now turn to upper-bounding $\|y\|_X$.

Fix some $i \in [d]$ where $x_i = \beta^{-j}$. Then

$$y_i = \sum_{k > j - 3\log_\beta(\beta-1)} \left(\sum_{k' \geq k} x_{s_{k'}^{-1}(g_k^{-1}(i))} \right),$$

where we interpret $x_{s_{k'}^{-1}(g_k^{-1}(i))}$ as 0 when $g_k^{-1}(i) = \emptyset$, or $s_{k'}^{-1}(g_k^{-1}(i)) = \emptyset$. Note that whenever $x_{s_{k'}^{-1}(g_k^{-1}(i))} \neq 0$, $x_{s_{k'}^{-1}(g_k^{-1}(i))} = \beta^{-k'}$. Thus,

$$y_i \leq \sum_{k > j - 3\log_\beta(\beta-1)} \left(\sum_{k' \geq k} \beta^{-k'} \right) \leq \sum_{k > j - 3\log_\beta(\beta-1)} \frac{\beta^{1-k}}{\beta - 1} \leq \frac{\beta^{1-j+3\log_\beta(\beta-1)}}{(\beta-1)^2} \leq \beta(\beta-1) \cdot \beta^{-j}.$$

Recall that $x_i = \beta^{-j}$, so $\beta(\beta-1)x$ weakly majorizes y, and thus

$$\|x - z\|_X \leq \|y\|_X \leq \beta(\beta-1) \cdot \|x\|_X.$$

Hence, when $\beta \leq 2$, we have $\|x - z\|_X \leq 2(\beta-1)\|x\|_X$. □

Proof of Lemma 2.2.7. We now prove the theorem by showing that the set

$$N = \{S(x) \in \mathbb{R}^d \mid x \in \mathcal{R}\}$$

is a γ-net of $\mathcal{R}$, and we give an upper bound on the size. By Lemma 2.2.9, $\|x - S(x)\|_X \leq 2(\beta-1)\|x\|_X \leq 8(\beta-1)$. So it suffices to give an upper bound on the size of N.

We bound from above the number of net points by an encoding argument. Let $z = S(x)$ and let

$$t_k = \frac{b_k(z)}{\max_{j<k+3\log_\beta(\beta-1)} b_j(z)}.$$

Let $k^* \in \{0, \ldots, 2\log_\beta d - 1\}$ be the smallest level k with non-zero $b_k(z)$. For all $j > k^* - 3\log_\beta(\beta-1)$, we either have $t_j(z) = 0$ or $t_j(z) \geq 1$. Additionally, z has d coordinates, so

$$\prod_{j=k^*-3\log_\beta(\beta-1)}^{2\log_\beta d-1} \max(t_j, 1) \leq d^{-3\log_\beta(\beta-1)},$$

since terms of the product "cancel" except for at most $-3\log_\beta(\beta-1)$, which are each at most d.

We will encode $z \in N$ in three steps. In the first step, we use $2\log_\beta d - 1$ bits in order to encode whether $b_k(z) = 0$ or $b_k(z) > 0$. In the second step, we then encode the values $b_{k^*+j}(z)$ for $j \in \{0, \ldots, 3\log_\beta(1/(\beta-1))\}$. Finally, we go through $j > k^* + 3\log_\beta(1/(\beta-1))$, and encode t_i using a prefix-free code, where writing t_i uses at most $O(\log\max(t_i, 1))$ many bits. Thus, in total, the number of bits we use is

$$\begin{aligned}
&O\left(\log_\beta d + \log d \log_\beta\left(\frac{1}{\beta-1}\right) + \sum_{j=k^*-3\log_\beta(\beta-1)}^{2\log_\beta d-1} \log\max(t_j, 1)\right) \\
&\quad = O\left(\frac{\log d \cdot \log\left(\frac{1}{\beta-1}\right)}{\log\beta} + \log\left(\prod_{j=k^*-3\log_\beta(\beta-1)}^{2\log_\beta d-1} \max(t_j, 1)\right)\right) \\
&\quad = O\left(\frac{\log d \cdot \log\left(\frac{1}{\beta-1}\right)}{\log\beta}\right).
\end{aligned}$$

Thus, we obtain N is a $8(\beta-1)$-net, and the size of N is $d^{O(\log(1/(\beta-1))/\log\beta)}$. □

2.3 Proof of Theorem 34

We now prove our main result, Theorem 34. The algorithm here achieves approximation

$$O\left(\frac{\log^2 \log n \cdot \log \log d}{\varepsilon^5}\right).$$

We proceed by giving an algorithm for $\ell_\infty^t(\ell_1^d(T^{(c)}))$ using Theorem 2, Theorem 5.1.2 from [5], and Theorem 36.

Lemma 2.3.1. *Fix some $c_1, \ldots, c_d \geq 0$. Let $\ell_\infty^d(T^{(c)})$ be the space with $\|\cdot\|_{T,\infty}^{(c)} \colon \mathbb{R}^{d^2} \to \mathbb{R}$ seminorm where for every $x = (x_1, \ldots, x_d) \in \mathbb{R}^{d^2}$,*

$$\|x\|_{T,\infty}^{(c)} = \max_{k \in [d]} c_k \|x_k\|_{T(k)}.$$

For every $\varepsilon \in (0, 1/2)$, there exists a data structure for ANN over $\|\cdot\|_{T,\infty}^{(c)}$ which achieves approximation $O(\log \log n \cdot \log \log d/\varepsilon^3)$ using space $O\left(d^2 \cdot n^{1+\varepsilon}\right)$ and query time $O\left(d^2 \cdot n^{\varepsilon}\right)$.

Proof. Given the randomized embedding from Lemma 2.1.2, we can build a data structure for $c_k\|\cdot\|_{T(k)}$ achieving approximation $O(\log \log d/\varepsilon^2)$ using space $O(d^2 n^{1+\varepsilon/2})$ and query time $O(d^2 n^{\varepsilon/2})$. This data structure works in the same way as in the proof of Theorem 37. We handle the constant c_k by rescaling the norm, and since the embeddings are linear, it does not affect the correctness of the data structure. Then we apply Theorem 2. □

Lemma 2.3.2. *Fix some $c_1, \ldots, c_d \geq 0$. Let $\ell_1^d(T^{(c)})$ be the space with $\|\cdot\|_{T,1}^{(c)} \colon \mathbb{R}^{d^2} \to \mathbb{R}$ seminorm where $x = (x_1, \ldots, x_m) \in \mathbb{R}^{d^2}$,*

$$\|x\|_{T,1}^{(c)} = \sum_{k=1}^{d} c_k \|x_k\|_{T(k)}.$$

For every $\varepsilon \in (0, 1/2)$, there exists a data structure for ANN over $\|\cdot\|_{T,1}^{(c)}$ which achieves approximation $O(\log \log n \cdot \log \log d/\varepsilon^4)$ using space $O(d^2 \cdot n^{1+\varepsilon})$ and query time $O(d^2 \cdot n^{\varepsilon})$.

Proof. The proof follows from Theorem 5.1.2 in [5] and Lemma 2.3.1. □

Finally, we are combine the above results to get an improved algorithm for general symmetric norms.

Theorem 39. *For every d-dimensional symmetric norm $\|\cdot\|_X$ and every $\varepsilon \in (0, 1/2)$, there exists a data structure for ANN over $\|\cdot\|_X$ which achieves approximation $O(\log^2 \log n \log\log d/\varepsilon^5)$ using space $d^{O(1)} \cdot O(n^{1+\varepsilon})$ and query time $d^{O(1)} \cdot O(n^{\varepsilon})$.*

Proof. Given Theorem 38, we embed $\|\cdot\|_X$ into $\ell_\infty^{\mathrm{poly}(d)}(\ell_1^d(T^{(c)}))$ with approximation $(1 \pm \frac{1}{10})$. The result from Lemma 2.3.2 allows we to apply Theorem 36 to obtain the desired data structure. □

Theorem 39 implies our main result Theorem 34.

2.4 A lower bound for linear embeddings of general symmetric norms

We show that these techniques do not extend to general norms. In particular, we show there does not exist a *universal* norm U for which any norm embeds (possibly randomized) with constant distortion, unless the blow-up in dimension is exponential. Hence the result from below applies to cases of $U = \ell_\infty$ as well as an (low-dimensional) product spaces.

Theorem 40. *For any $\varepsilon > 0$, let U be a d'-dimensional normed space such that for any d-dimensional normed space X, there exists a distribution $\mathcal{D}$ supported on linear embeddings $f : \mathbb{R}^d \to \mathbb{R}^{d'}$ where for every $x \in \mathbb{R}^d$,*

$$\|x\|_X \leq \|f(x)\|_U \leq D\|x\|_U$$

holds with probability at least $2/3$ over the draw of $f \sim \mathcal{D}$, for $D = O(d^{1/2-\varepsilon})$. Then $d' = \exp(\Omega(d^{2\varepsilon}))$.

The following proof presented was suggested by Piotr Indyk. We will prove the above theorem by showing that if there exists a universal normed space U satisfying the conditions of Theorem 40 above, then two parties, call them Alice and Bob, can use the embeddings

to solve the communication problem INDEX with only a few bits. Let U be a proposed d'-dimensional normed space satisfying the conditions of Theorem 40. By the John's theorem [25], we may apply a linear transform so that:

$$B_{\ell_2} \subset B_U \subset \sqrt{d'} B_{\ell_2}$$

Lemma 2.4.1. *For any $\varepsilon > 0$, there exists a set of $\exp(\Omega(d^{2\varepsilon}))$ many points on the unit sphere S^{d-1} such that pairwise inner-products are at most $1/d^{1/2-\varepsilon}$. In fact, these points may consist of points whose coordinates are $\pm 1/\sqrt{d}$.*

Proof. Consider picking two random points $x, y \in S^{d-1}$ where each entry is $\pm\frac{1}{\sqrt{d}}$. Then by Bernstein's inequality,

$$\mathbf{Pr}_{x,y}\left[|\langle x, y\rangle| \geq \frac{1}{d^{1/2-\varepsilon}}\right] \leq 2\exp\left(-\Omega(d^{2\varepsilon})\right)$$

We may pick $\exp(\Omega(d^{2\varepsilon}))$ random points and union bound over the probability that some pair has large inner product. □

Fix $\varepsilon > 0$ and $C = d^{1/2-\varepsilon}$, and let P be set a set of unit vectors with pairwise inner-product at most $\frac{1}{C}$ of size $\exp(\Omega(d^{2\varepsilon}))$. For each $a \in \{0,1\}^P$ consider the following norm:

$$\|x\|_a = C \cdot \max_{y \in P : a_y = 1} |\langle x, y\rangle|.$$

Assume there exists a randomized linear embedding $f\colon \mathbb{R}^d \to \mathbb{R}^{d'}$ with the following guarantees:

- For every $x \in \mathbb{R}^d$, $\|x\|_a \leq \|f(x)\|_U \leq D\|x\|_a$ with probability at least $\frac{2}{3}$.

Note the embedding f can be described by M, a $d' \times d$ matrix of real numbers. Additionally, we consider rounding each entry of M by to the nearest integer multiple of $\frac{1}{\mathrm{poly}(d)}$ to obtain M'. For each $x \in S^{d-1}$, $\|(M - M')x\|_U \leq \|(M - M')x\|_2 \leq \frac{1}{\mathrm{poly}(d)}$. Thus, we may assume each entry of M is an integer multiple of $\frac{1}{\mathrm{poly}(d)}$, and lose $(1 \pm \frac{1}{\mathrm{poly}(d)})$ factor in the distortion of the embedding for vectors in B_2.

We now show that the existence of the randomized embedding implies a one-way randomized protocol for the communication problem INDEX. We first describe the problem. In an instance of INDEX:

- Alice receives a string $a \in \{0,1\}^n$.
- Bob receives an index $i \in [n]$.
- Alice communicates with Bob so that he can output a_i.

Theorem 41 ([92]). *The randomized one-way communication complexity of* INDEX *is* $\Omega(n)$.

We give a protocol for INDEX:

1. Suppose Alice has input $a \in \{0,1\}^P$. She will generate the norm $\|\cdot\|_a$ described above. Note that $f \sim \mathcal{D}$ has that for each $x \in \mathbb{R}^d$, the embedding preserves the norm of x up to D with probability $\frac{2}{3}$. In particular, if Bob's input is $i \in |P|$, corresponding to point y, then an embedding $f \in \mathcal{D}$, which we represent as a $d' \times d$ matrix M, satisfies:

   $$\|y\|_a \leq \|My\|_U \leq D\|y\|_a$$

   with probability $\frac{2}{3}$. In particular, with probability $\frac{2}{3}$:

   - If $a_i = 0$, then $\|y\|_a \leq 1$, which implies $\|My\|_U \leq D$.
   - If $a_i = 1$, then $\|y\|_a \geq C$, which implies $\|My\|_U \geq C$.

   Alice computes the set $P_c \subset P$ of vectors which satisfy the above property (i.e. the embedding M preserves increases the norm by at most a factor D).

2. Alice finds a subset $B \subset P_c$ of linearly independent vectors such that every $x \in P_c$ we have $x \in \mathrm{span}(B)$. Note that $|B| \leq d$ and for all $x \in B$, $\|Mx\|_2 \leq \sqrt{d'}\|Mx\|_U \leq C \cdot D \cdot \sqrt{d'}$. Therefore, each $Mx \in \mathbb{R}^{d'}$ can be written with $\tilde{O}(d')$ bits. So Alice sends the set B, as well as Mx for each $x \in B$ using $\tilde{O}(dd')$ bits.

3. In order for Bob to decode a_i, he first checks whether $y \in \mathrm{span}(B)$, and if not, he

guesses. If $y \in \mathrm{span}(B)$, which happens with probability $\frac{2}{3}$, then Bob writes

$$y = \sum_{b_i \in B} c_i b_i$$

and $My = \sum_{b_i \in B} c_i M b_i$. If $\|My\|_U \leq D$, then $a_i = 0$ and if $\|My\|_U \geq C$ then $a_i = 1$. Thus, if $D < \frac{C}{2}$, Bob can recover a_i with probability $\frac{2}{3}$.

Alice communicates $\tilde{O}(dd')$ bits, and Bob is able to recover a_i with probability $\frac{2}{3}$. By Theorem 41, $dd' \geq \tilde{\Omega}(|P|)$, which in turn implies $d' \geq \exp(\Omega(d^{2\varepsilon}))$.

Chapter 3

ANN via the Cutting Modulus

We now provide the formal arguments for Section 1.3 in Chapter 1, where the main algorithmic contribution is to show that small cutting modulus (Definition 15) yields a new ANN bit-probe data structure with small approximation, $O(\Xi(X, \varepsilon))$. We recall that a data structure in the bit-probe model is specified by a preprocessing procedure that encodes the input as a string of bits, which is the data structure, and a query procedure that inspects some of the bits in the data structure to answer a given query.

Definition 26 (Re-statement of the Cutting Modulus [14])**.** *Let (X, d_X) be a finite metric space and $\varepsilon \in (0, 1)$. The cutting modulus $\Xi(X, \varepsilon)$ is the infimum of $\Xi > 0$ such that the following holds. For all $r \geq 0$, and all symmetric probability measures μ supported on pairs of points $X \times X$ at distance at most r in X with marginal distribution ρ, either (1) there exists a set $B \subset X$ with $\rho(B) \geq 1/2$ and $\operatorname{diam}_X(B) \leq r \cdot \Xi$, or (2) there exists a set $S \subset X$ with conductance*

$$\Phi_\mu(S) = \frac{\mu(S, X \setminus S)}{\min\{\rho(S), 1 - \rho(S)\}} \leq \varepsilon.$$

Theorem 42 (Re-statement of Cell-Probe Data Structure for $(\Xi(X, \varepsilon) + 1)$-ANN [14])**.** *Let (X, d_X) be a metric space of size N and $\varepsilon \in (0, 1)$. There exists a data structure for $(\Xi(X, \varepsilon) + 1)$-ANN over X with space complexity $n^{1+O(\varepsilon)} \log N(\log(1/\varepsilon) + \log\log N)$ and cell-probe complexity $n^{O(\varepsilon)} \log N(\log(1/\varepsilon) + \log\log N)$.*

We do not bound the *time* of the query procedure, only the number of bits read from the preprocessed memory. To simply store the dataset requires $\Theta(n \cdot \log N)$ bits, so the space overhead of the data structure is essentially $n^{O(\varepsilon)}$, up to lower order terms. Finally, the query procedure is deterministic and the randomness is only used in the preprocessing step.

The main tool used to prove Theorem 42 is a new randomized partitioning procedure for a metric space (X, d_X) with guarantees similar to those of data-dependent LSH [13, 10, 18]. Crucially, the quality of the resulting partitions is directly governed by the cutting modulus $\Xi(X, \varepsilon)$. They are also tractable, in the sense that, for any dataset $P \subseteq X$, a partition of X can be specified by a small number of bits. The precise guarantees are given in the following theorem.

Theorem 43 (Re-statement of the Main Partitioning Theorem[14])**.** *Let (X, d_X) be a metric space of size N, $\varepsilon \in (0, 1/8)$, and $r \geq 0$. There exists a collection $\mathcal{F}$ of at most $N^{O(\log(1/\varepsilon)+\log\log N)}$ subsets of X such that the following holds. If, for $1 < m \leq N$ and $x_1, \ldots, x_m \in X$, all $B \subset X$ with $\operatorname{diam}_X(B) \leq r \cdot \Xi(X, \varepsilon)$ satisfy $|B \cap \{x_1, \ldots, x_m\}| \leq m/8$, there exists a distribution $\mathcal{H}$ supported on $\mathcal{F}$ satisfying:*

1. *A subset $\mathbf{S} \sim \mathcal{H}$ satisfies $|\mathbf{S} \cap \{x_1, \ldots, x_m\}| \in [m/8, 7m/8]$ with probability 1.*
2. *If $y, z \in X$ satisfy $d_X(y, z) \leq r$, then*

$$\Pr_{\mathbf{S}\sim\mathcal{H}}\left[|\{y, z\} \cap \mathbf{S}| = 1\right] \leq 20\varepsilon.$$

3.1 Partitioning general metrics

The proof of Theorem 43 consists of three parts, which are described in the next three sections. We first set up some notation. For a finite set U, let $\Delta(U)$ be the set of all *probability measures* over U, and let $\Gamma(U)$ be the set of all *symmetric* probability measures over $U \times U$. For $\rho \in \Delta(U)$ and $u \in U$, we write $\rho(u) = \rho(\{u\})$, and similarly, for $\mu \in \Gamma(U)$ and $u, v \in U$, we write $\mu(u, v) = \mu(\{(u, v)\})$. For $\mu \in \Gamma(U)$, recall that we denote the *marginal* probability measure by $\rho_\mu \in \Delta(U)$, given by $\rho_\mu(u) = \sum_{v\in U} \mu(u, v)$.

Let (X, d_X) be a *finite* metric space. For $r \geq 0$, we denote by $\Gamma_r(X) \subseteq \Gamma(X)$ the set of all symmetric probability measures over $X \times X$ supported on the pairs of points (x, y) with $d_X(x, y) \leq r$. We use $B_X(x, r) = \{y \in X : d_X(x, y) \leq r\}$ to denote the ball of the radius r centered at $x \in X$.

3.1.1 Cutting Modulus and Partitioning

Definition 27 (Dispersed measures)**.** *A measure $\rho \in \Delta(X)$ is* (R, β)-dispersed *for $R > 0$ and $0 \leq \beta \leq 1$ if every $B \subset X$ with $\mathrm{diam}_X(B) \leq R$ has $\rho(B) \leq \beta$. The subset $\Delta_{R,\beta}(X) \subset \Delta(X)$ is the set of all (R, β)-dispersed probability measures over X.*

With this definition, we can restate Definition 26 of the cutting modulus $\Xi(X, \varepsilon)$ as the infimum over all $\Xi > 0$ for which we have that, for every $r > 0$ and every $\mu \in \Gamma_r(X)$ with $\rho_\mu \in \Delta_{\Xi \cdot r, 1/2}(X)$, there exists $S \subset X$ such that $\Phi_\mu(S) \leq \varepsilon$. The next two definitions deal with notions of randomly partitioning a metric space.

Definition 28 (Balanced Cuts)**.** *Let U be a finite set, $S \subseteq U$ and $\rho \in \Delta(U)$. Fix $0 \leq \beta \leq 1/2$. We say that S is* β-balanced *with respect to ρ, if $\beta \leq \rho(S) \leq 1 - \beta$. Moreover, we say that a probability distribution $\mathcal{D}$ over 2^U is β-balanced with respect to ρ if all sets in the support of $\mathcal{D}$ are β-balanced.*

Definition 29 (Separating Distribution)**.** *We say that a distribution $\mathcal{D}$ over subsets of X is* (s, δ)-separating *for $s > 0$ and $0 \leq \delta \leq 1$, if for every $x, y \in X$ with $0 < d_X(x, y) \leq s$, one has: $\mathbf{Pr}_{S \sim \mathcal{D}}[\left|\{x, y\} \cap S\right| = 1] \leq \delta$.*

Theorem 43, stated more concisely now, guarantees that, for every $\rho \in \Delta_{\Xi(X,\varepsilon) \cdot r, 1/8}(X)$, there exists a $(r, 20\varepsilon)$-separating distribution $\mathcal{D}$ supported on $\mathcal{F}$, all of which are $1/8$-balanced with respect to ρ.

3.1.1.1 Finding *balanced* sparse cuts

Lemma 3.1.1. *Let $\mu \in \Gamma_r(X)$ satisfy $\rho_\mu \in \Delta_{R,1/4}(X)$. Then, there exists a subset $S \subseteq X$ which is $1/4$-balanced with respect to ρ_μ and satisfies $\mu(S \times (X \setminus S)) \leq \varepsilon/2$.*

We note that $\mu \in \Gamma_r(X)$ satisfying $\rho_\mu \in \Delta_{R,1/4}(X)$ must also satisfy $\rho_\mu \in \Delta_{R,1/2}(X)$, so by the definition of the cutting modulus, there exists a subset $S \subset X$ with $\Phi_\mu(S) \leq \varepsilon$. Since X is a finite metric space, the proof of Lemma 3.1.1 will follow from applying the following claim.

Claim 3.1.2. *Let $\mu \in \Gamma_r(X)$ satisfy $\rho_\mu \in \Delta_{R,1/4}(X)$ and suppose $S \subset X$ satisfies $\Phi_\mu(S) \leq \varepsilon$ and $\rho_\mu(S) < 1/4$. Then, there exists a non-empty subset $S' \subset X \setminus S$ with $\rho_\mu(S') \leq 1/2$ satisfying $\Phi_\mu(S \cup S') \leq \varepsilon$.*

Proof. Consider the measure $\tilde{\mu} \in \Gamma_r(X)$ given by conditioning μ on $(X \setminus S) \times (X \setminus S)$. We will show that $\rho_{\tilde{\mu}} \in \Delta_{R,1/2}(X)$, which by the definition of the cutting modulus, implies there exists a set $S' \subset X$ with $\rho_{\tilde{\mu}}(S') \leq 1/2$ and $\Phi_\mu(S') \leq \varepsilon$. Then, we will show that $S \cup S'$ satisfies $\Phi_\mu(S \cup S') \leq \varepsilon$.

First, we note that for any subset $T \subset X$,

$$\rho_{\tilde{\mu}}(T) = \frac{\rho_\mu(T) - \mu(T \times S) - \mu((T \cap S) \times (X \setminus S))}{1 - \rho_\mu(S) - \mu(S \times (X \setminus S)) - \mu((X \setminus S) \times S)}. \tag{3.1}$$

Since $\rho_\mu(S) < 1/4$ and $\Phi_\mu(S) \leq \varepsilon$ for $\varepsilon < 1/8$, we have $\rho_{\tilde{\mu}}(T) \leq 2\rho_\mu(T)$. In particular, this implies that if $x \in X$ and we consider the ball $T = B_X(x, R)$, then $\rho_{\tilde{\mu}}(T) \leq 2\rho_\mu(T) \leq 1/2$ since $\rho_\mu \in \Delta_{R,1/4}$. We conclude $\rho_{\tilde{\mu}} \in \Delta_{R,1/2}(X)$, and let $S' \subset X$ be a set satisfying $\rho_{\tilde{\mu}}(S) \leq 1/2$ and $\Phi_{\tilde{\mu}}(S) \leq \varepsilon$ guaranteed to exists since $R = \Xi(X, \varepsilon) \cdot r$. Without loss of generality, we may assume $S' \subset X \setminus S$, since $\rho_{\tilde{\mu}}(x) = 0$ for all $x \in S$. Applying (3.1) to the set S' gives $\rho_\mu(S) \leq 1/2$. Finally, since $\Phi_\mu(S) \leq \varepsilon$ and $\Phi_{\tilde{\mu}}(S') \leq \varepsilon$, we have:

$$\mu((S \cup S') \times (X \setminus (S \cup S'))) \leq \varepsilon\rho_\mu(S) + \varepsilon\rho_{\tilde{\mu}}(S')\mu((X \setminus S) \times (X \setminus S)) \leq \varepsilon(\rho_\mu(S) + \rho_\mu(S')),$$

which finished the claim. □

Proof of Lemma 3.1.1. Let $\mu \in \Gamma_r(X)$ satisfy $\rho_\mu \in \Delta_{R,1/4}(X)$, and let $\mathcal{S}$ be the collection of subsets $S \subset X$ with $\Phi_\mu(S) \leq \varepsilon$ and $\rho_\mu(S) < 3/4$, and let $S^* \in \mathcal{S}$ be the set of maximum cardinality (which is well-defined since X is finite). Then, if $\rho_\mu(S^*) < 1/4$, letting S' be the non-empty set obtained from Claim 3.1.2, we have $S^* \cup S' \in \mathcal{S}$, which contradicts the fact S^* has maximum cardinality among all sets in $\mathcal{S}$. Thus, $\Phi_\mu(S^*) \leq \varepsilon$ and S^* is $1/4$-balanced with respect to ρ_μ, which implies $\mu(S^* \times (X \setminus S^*)) \leq \varepsilon/2$. □

Lemma 3.1.3. *Suppose that $\nu \in \Delta_{R,1/7}(X)$ and $\mu \in \Gamma_r(X)$. Then, there exists a subset $S \subseteq X$, which is $1/7$-balanced with respect to ν and $\mu\Big(S \times (X \setminus S)\Big) \leq 8\varepsilon$.*

Proof. Consider the probability measure $\tilde{\mu}$ over $X \times X$ given by

$$\tilde{\mu}(x_1, x_2) = \begin{cases} \frac{1}{8} \cdot \mu(x_1, x_2) & \text{if } x_1 \neq x_2; \\ \frac{1}{7} \cdot \mu(x_1, x_1) + \frac{7}{8} \cdot \nu(x_1) & \text{otherwise.} \end{cases}$$

Note $\tilde{\mu} \in \Gamma_r(X)$, and for every $S \subseteq X$, $\rho_{\tilde{\mu}}(S) = \rho_\mu(S)/8 + 7\nu(S)/8$ and $\tilde{\mu}(S \times (X \setminus S)) = \mu(S \times (X \setminus S))/8$. Furthermore, any ball of radius R has measure at most $1/7$ in ν and at most 1 in μ, which gives $\tilde{\mu} \in \Delta_{R,1/4}(X)$. We now apply Lemma 3.1.1 to $\tilde{\mu}$ to obtain a set $S \subset X$ which is $1/4$-balanced with respect to $\rho_{\tilde{\mu}}$ and has $\tilde{\mu}(S \times (X \setminus S)) \leq \varepsilon/2$. Then, by the properties of $\rho_{\tilde{\mu}}$ and $\tilde{\mu}$ above, S' must be $1/7$-balanced with respect to ν and $\mu(S \times (X \setminus S)) \leq 4\varepsilon$. □

3.1.2 The Multiplicative Weights Update Method

One of the crucial ingredients in our arguments is a version of the Multiplicative Weights Update algorithm [20]. Before we state the relevant result, let us recall the notion of *relative entropy*. Suppose that U is a finite set, and let ψ, ν be two probability measures over U. We define the relative entropy $\mathrm{RE}(\psi \| \nu)$ as

$$\mathrm{RE}(\psi \| \nu) = \sum_{u \in U} \psi(u) \cdot \ln \frac{\psi(u)}{\nu(u)},$$

where $\psi(u) \cdot \ln \frac{\psi(u)}{\nu(u)} = 0$ when $\psi(u) = 0$, and $\psi(u) \cdot \ln \frac{\psi(u)}{\nu(u)} = \infty$ if $\nu(u) = 0$ and $\psi(u) > 0$.

The following lemma is implicit in the proof of Theorem 2.4. in [20], and is also used in [74]. We include a proof for completeness.

Lemma 3.1.4. *Let U be a finite set. Let $\nu \in \Delta(U)$, and $0 \leq \eta \leq 1$. Then, for every $S \subseteq U$, there exist two probability measures $\nu^+, \nu^- \in \Delta(U)$, with the same support as ν, such that for every $\psi \in \Delta(U)$:*

$$\mathrm{RE}(\psi \| \nu^+) - \mathrm{RE}(\psi \| \nu) \leq \eta(\nu(S) - \psi(S)) + \eta^2$$
$$\mathrm{RE}(\psi \| \nu^-) - \mathrm{RE}(\psi \| \nu) \leq \eta(\psi(S) - \nu(S)) + \eta^2.$$

Proof. We define the measures as follows: for any $u \in U$, we set

$$\nu^+(u) = \frac{\nu(u)e^{\eta S(x)}}{\sum_{w\in U}\nu(w)e^{\eta S(w)}}, \qquad \nu^-(u) = \frac{\nu(u)e^{-\eta S(x)}}{\sum_{w\in U}\nu(w)e^{-\eta S(w)}},$$

where $S(u)$ equals 1 if $u \in S$, and 0 otherwise. We prove the lemma for $\nu^+(u)$; the proof for $\nu^-(u)$ is analogous.

By the definition of relative entropy, we have

$$\begin{aligned}\mathrm{RE}(\psi\|\nu^+(u)) - \mathrm{RE}(\psi\|\nu) &= \sum_{u\in U}\psi(u)\ln\frac{\nu(u)}{\nu^+(u)} = \sum_{u\in U}\psi(u)\ln\frac{\sum_{w\in U}\nu(w)e^{\eta S(w)}}{e^{\eta S(u)}} \\ &= -\eta\psi(S) + \ln\sum_{w\in U}\nu(w)e^{\eta S(w)} \le -\eta\psi(S) + \ln\sum_{w\in U}\nu(w)(1+\eta S(w)+\eta^2 S(w) \\ &= -\eta\psi(S) + \ln(1+\eta\nu(S)+\eta^2\nu(S)) \le \eta(\nu(S)-\psi(S)) + \eta^2.\end{aligned}$$

The first inequality above follows from $e^z \le 1+z+z^2$ for all $|z| \le 1$. The second inequality follows from $\ln(1+z) \le z$. □

We will use the following immediate corollary of Lemma 3.1.4.

Corollary 3.1.5. *Let U be a finite set. Let $\nu \in \Delta(U)$, and $0 \le \delta \le 1$. Then, for every $S \subseteq U$, there exist two probability measures $\nu^+, \nu^- \in \Delta(U)$ such that for every $\psi \in \Delta(U)$:*

- *If $\psi(S) \ge \nu(S) + \delta$, then $\mathrm{RE}(\psi\|\nu^+) \le \mathrm{RE}(\psi\|\nu) - \frac{\delta^2}{4}$;*
- *If $\psi(S) \le \nu(S) - \delta$, then $\mathrm{RE}(\psi\|\nu^-) \le \mathrm{RE}(\psi\|\nu) - \frac{\delta^2}{4}$.*

Proof. Follows from Lemma 3.1.4 by setting $\eta = \frac{\delta}{2}$.

□

3.1.2.1 Weaker version of Theorem 43

Using Lemma 3.1.3, we now show a weaker version of Theorem 43, which will later be used to show Theorem 43.

Lemma 3.1.6. *For every $\nu \in \Delta_{R,1/7}(X)$, there exists a $(r, 9\varepsilon)$-separating distribution $\mathcal{D}$ supported on $O(\varepsilon^{-2}\log N)$ subsets of X, all of which are $1/7$-balanced with respect to ν.*

The key difference between Theorem 43 and Lemma 3.1.6 is that in Lemma 3.1.6 we do not require the distribution $\mathcal{D}$ to be supported on a collection $\mathcal{F}$ of subsets of X that is both of small size ($|\mathcal{F}| = N^{O(\log N)}$) and is *independent of the measure ν*. This property is crucial for the ANN application, since it allows to encode a subset sampled from $\mathcal{D}$ using merely $O((\log N)(\log \varepsilon^{-1} + \log\log N)) \ll N$ bits. Nevertheless, Lemma 3.1.6 is a useful step towards Theorem 43.

Proof of Lemma 3.1.6. To give some intuition, let us first show the existence of $\mathcal{D}$ supported on possibly *all* the 2^N subsets of X. By von Neumann's minimax theorem,

$$\min_{\mathcal{D}} \max_{x_1,x_2} \mathop{\mathbf{Pr}}_{S\sim\mathcal{D}}\Big[\Big|\{x_1,x_2\}\cap S\Big| = 1\Big] = \max_{\mu}\min_{S} \mu(S\times(X\setminus S)),$$

where $\mathcal{D}$ ranges over distributions supported on $1/7$-balanced sets with respect to ν, x_1, x_2 range over pairs of points in X such that $0 < d_X(x_1,x_2) \le r$, μ ranges over $\Gamma_r(X)$, and S ranges over $1/7$-balanced sets. Then, if there is no distribution $\mathcal{D}$ as in the statement in the lemma, there must exist a distribution $\mu \in \Gamma_r(X)$ such that for every set $S \subseteq X$ that is $1/7$-balanced with respect to ν, one has $\mu\Big(S\times(X\setminus S)\Big) > 8\varepsilon$. But this directly contradicts Lemma 3.1.3. In order to also give a bound on the support size of $\mathcal{D}$, we will use a constructive proof of the minimax theorem, based on the multiplicative weights update method. The argument follows along the lines of [65] (see also [20]).

For any $x_1, x_2 \in X$ such that $0 < d_X(x_1,x_2) \le r$, define ψ_{x_1,x_1} to be the measure that assigns $\frac{1}{2}$ to (x_1,x_2) and to (x_2,x_1), and 0 to all other pairs of points. Let $\mu_0 \in \Gamma_r(X)$ be the uniform probability measure on $\{(x_1,x_2) : 0 < d_X(x_1,x_2) \le r\}$. Let $0 \le \eta \le 1$ be a real parameter, and T be an integer parameter, both to be determined soon. Suppose that we have defined measures $\mu_0, \mu_1, \ldots, \mu_{t-1}$, where $1 \le t \le T$ is an integer. Let $S_t \subset X$ be a $1/7$-balanced set with respect to ν such that $\mu_{t-1}\Big(S_t\times(X\setminus S_t)\Big) \le 8\varepsilon$, which is guaranteed to exist by Lemma 3.1.3. By Lemma 3.1.4, there exists a measure $\mu_t = \mu^+ \in \Gamma_r(X)$ such that, for any distinct x_1 and x_2 satisfying $d_X(x_1,x_2) \le r$,

$$\mathrm{RE}(\psi_{x_1,x_2}\|\mu_t) \le \mathrm{RE}(\psi_{x_1,x_2}\|\mu_{t-1}) + 8\eta\varepsilon - \eta\psi_{x_1,x_2}(S_t\times(X\setminus S_t)) + \eta^2.$$

Adding these inequalities over $t \in \{1, \ldots, T\}$ gives that, for all x_1, x_2 as above,

$$\mathrm{RE}(\psi_{x_1,x_2} \| \mu_T) \leq \mathrm{RE}(\psi_{x_1,x_2} \| \mu_0) + 8T\eta\varepsilon - |\{t : |\{x_1, x_2\} \cap S_t| = 1\}| + T\eta^2. \quad (3.2)$$

Let us set $T \geq \frac{4\log(N^2)}{\varepsilon^2}$ and $\eta = \sqrt{\frac{\log(N^2)}{T}}$ and, and define $\mathcal{D}$ to be distribution of S_t when t is sampled uniformly at random from $[T]$. An easy calculation shows that $D(\psi_{x_1,x_2} \| \mu_0) < \log(N^2)$, and, by the non-negativity of relative entropy, $\mathrm{RE}(\psi_{x_1,x_2} \| \mu_T) \geq 0$. Using these facts, and rearranging inequality (3.2), we get

$$\mathop{\mathbf{Pr}}_{S \sim \mathcal{D}}\Big[\Big|\{x_1, x_2\} \cap S\Big| = 1\Big] < \frac{\log(N^2)}{T\eta} + 8\varepsilon + \eta \leq 9\varepsilon,$$

as we wanted to prove. □

3.1.2.2 Proof of Theorem 43

We are now ready to prove Theorem 43 using Lemma 3.1.6 as a subroutine.

Lemma 3.1.7. *There exists two functions* $\mathtt{Enc}\colon \Delta_{R,1/8}(X) \to \{0,1\}^\ell$ *and* $\mathtt{Dec}\colon \{0,1\}^\ell \to \Delta_{R,1/7}(X)$ *with* $\ell = O((\log N)(\log \varepsilon^{-1} + \log\log N))$ *such that for any* $\psi \in \Delta_{R,1/8}(X)$*, letting* $\nu = \mathtt{Dec}(\mathtt{Enc}(\psi))$ *and* $\mathcal{D}$ *be the* $(r, 9\varepsilon)$*-separating distribution which is* $1/7$*-balanced with respect to* ν *given by Lemma 3.1.6,* $\mathcal{D}$ *is* $1/8$*-balanced with respect to* ψ*.*

Proof. For the remainder of the proof, given a distribution $\nu \in \Delta_{R,1/7}$, denote $\mathcal{D}_\nu$ as the $(r, 9\varepsilon)$-separating distribution which is $1/7$-balanced with respect to ν, as given by Lemma 3.1.6. Let us fix some $T > 4 \cdot 10^4 \cdot \log N$. For any $\psi \in \Delta_{R,1/7}(X)$, we consider the sequence $\mathcal{T}(\psi) = ((\nu_i, S_i) \in \Delta(X) \times 2^X : 0 \leq i \leq T)$ defined recursively by letting $\nu_0 \in \Delta(X)$ be the uniform distribution and $S_0 = \emptyset$, and when $i \geq 1$:

- If $\nu_{i-1} \notin \Delta_{R,1/7}(X)$, then we let S_i be a ball $B_X(x, R)$ for $x \in X$ with $\nu_{i-1}(S_i) \geq 1/7$. If there are many such balls, we choose one in some fixed way, e.g. by imposing an order on X, and picking the ball whose center comes first in the order. Since $\psi \in \Delta_{R,1/8}(X)$, we have $\psi(S_i) < 1/8$, so we let $\nu_i = \nu_{i-1}^-$ be the distribution obtained from applying Corollary 3.1.5 with set S_i to ν_{i-1}, ψ, and $\delta = 1/100$.

- If $\nu_{i-1} \in \Delta_{R,1/7}(X)$, we consider the distribution $\mathcal{D}_{\nu_{i-1}}$. If every $S \in \mathrm{supp}(\mathcal{D}_{\nu_{i-1}})$ is $1/8$-balanced with respect to ψ, we let $\nu_i = \nu_{i-1}$ and $S_i = \emptyset$. Otherwise, we let S_i be a set in $\mathrm{supp}(\mathcal{D}_{\nu_{i-1}})$ where ψ is not $1/8$-balanced. Therefore, $|\psi(S_i) - \nu_{i-1}(S_i)| \geq 1/100$, so let ν_i be the distribution obtained from applying Corollary 3.1.5 with S_{i-1} to ν_{i-1}, ψ, and $\delta = 1/100$. In particular, if $\psi(S_i) \geq \nu_{i-1}(S_i) + 1/100$, we let $\nu_i = \nu_{i-1}^+$, and if $\psi(S_i) \leq \nu_{i-1}(S_i) - 1/100$, we let $\nu_i = \nu_{i-1}^-$.

We make two crucial observations. The first observation is that for all $i \geq 1$ whenever $\nu_{i-1} \neq \nu_i$, $D(\psi\|\nu_i) \leq \mathrm{RE}(\psi\|\nu_{i-1}) - \delta^2/4$ (for $\delta = 1/100$) by Corollary 3.1.5. Recall that relative entropy is always non-negative, so $0 \leq \mathrm{RE}(\psi\|\nu_i)$. Moreover, since ν_0 is uniform, $\mathrm{RE}(\psi\|\nu_0) = \log N - H(\psi) \leq \log N$, where $H(\psi)$ is the Shannon entropy of ψ. It follows that, for some large $i \leq T$, we have $\nu_T = \ldots = \nu_i = \nu_{i-1}$ and equivalently, $\nu_T \in \Delta_{R,1/7}(X)$ and $\mathcal{D}_{\nu_T}$ is $1/8$-balanced with respect to ψ. The second observation is that for all $i \geq 1$, ν_i only depends on S_i, ν_{i-1}, and on whether $\psi(S_i) > \nu_{i-1}(S_i)$, and this information can be encoded concisely. Next we describe the precise encoding. Let us fix some $\ell' = O(\log \varepsilon^{-1} + \log\log N)$ such that $2^{\ell'} - 1$ is an upper bound on the support size of $\mathcal{D}_{\nu_i}$, and, for each $i \geq 0$, fix a bijection f_i from the support of $\mathcal{D}_{\nu_i}$ to $\{0,1\}^{\ell'} \setminus \{0\}$. For example, we can fix an ordering on X, and, for a set S in the support of $\mathcal{D}_{\nu_i}$, let $f_i(S)$ be its rank in the lexicographic ordering of $\mathrm{supp}(\mathcal{D}_{\nu_i})$. We define $\sigma_i \in \{0,1\}^{\ell'+1}$ for each $i \geq 1$ as follows:

- If $\nu_{i-1} \notin \Delta_{R,1/7}(X)$, or if $\nu_{i-1} \in \Delta_{R,1/7}(X)$ and every set in the support of $\mathcal{D}_{\nu_{i-1}}$ is $1/8$ balanced, we set $\sigma_i = 0$.
- If $\nu_i \in \Delta_{R,1/7}$, then we set $\sigma_i = f_{i-1}(S_i) \circ 0$ in case $\psi(S_i) > \nu_{i-1}(S_i)$, and we set $\sigma_i = f_{i-1}(S_i) \circ 1$ in case $\psi(S_i) < \nu_{i-1}(S_i)$. Here $\circ$ denotes concatenation.

Then, $\mathtt{Enc}(\psi)$ equals the concatenation $\sigma = \sigma_1 \circ \ldots \circ \sigma_T$. Clearly, $\ell = T \cdot (\ell' + 1) = O((\log N)(\log \varepsilon^{-1} + \log\log N))$ as claimed. The value of the decoding function $\mathtt{Dec}(\sigma)$, for $\sigma \in \{0,1\}^\ell$, can be computed by decoding the sequence $\mathcal{T}(\psi)$, and returning $\nu = \nu_T$. In particular, we set ν_0 to be the uniform distribution on X, and use σ to simulate the recursive construction of $\mathcal{T}(\psi)$ above. For that purpose, we divide σ into $\sigma_1, \ldots, \sigma_T$, each a block of $\ell' + 1$ bits. If $\sigma_i = 0$ and $\nu_{i-1} \notin \Delta_{R,1/7}(X)$, we set S_i to be a ball $B_X(x, R)$

such that $\nu_{i-1}(S_i) \geq 1/7$, and set $\nu_i = \nu_{i-1}^-$. (If there are multiple such balls, we break ties in the same way we did in the construction of $\mathcal{T}$.) If $\sigma = 0$ and $\nu_{i-1} \in \Delta_{R,1/7}(X)$, then we set $S_i = \emptyset$ and $\nu_i = \nu_{i-1}$. If $\sigma_i \neq 0$, then we let σ_i' be the first ℓ' bits of σ_i, and set $S_i = f_i^{-1}(\sigma_i')$. If the last bit of σ_i is 0, we set $\nu_i = \nu_{i-1}^+$, and if the last bit of σ_i is 1, we set $\nu_i = \nu_{i-1}^-$. It is straightforward to verify that the sequence $(\nu_i, S_i)_{i=0}^T$ constructed in this way equals $\mathcal{T}(\psi)$. Then, $\mathtt{Dec}(\sigma) = \nu_T$ as promised, and, as observed above, we must have that $\nu_T \in \Delta_{R,1/7}(X)$ and $\mathcal{D}_{\nu_T}$ is $1/8$-balanced with respect to ψ.

□

Proof of Theorem 43. For any $\sigma \in \{0,1\}^\ell$, where $\ell = O((\log N)(\log \varepsilon^{-1} + \log\log N))$ is as in Lemma 3.1.7, consider the distribution $\nu = \mathtt{Dec}(\sigma) \in \Delta_{R,1/7}(X)$, and let $\mathcal{D}_\sigma$ be the $1/7$-balanced, $(r, 9\varepsilon)$-separating distribution supported on $O(\varepsilon^{-2}\log N)$ subsets of X, guaranteed to exist by Lemma 3.1.6. We let $\mathcal{F} = \bigcup_{\sigma\in\{0,1\}^\ell} \mathrm{supp}(\mathcal{D}_\sigma)$. The cardinality of $\mathcal{F}$ is bounded by $O(2^\ell \varepsilon^{-2}\log N) = 2^{O((\log N)(\log\varepsilon^{-1}+\log\log N))}$. By Lemma 3.1.7, for every $\psi \in \Delta_{R,1/8}(X)$, there exists some $(r, 9\varepsilon)$-separating distribution $\mathcal{D}_\sigma$ (namely, the one for $\sigma = \mathtt{Enc}(\psi)$) which is $1/8$-balanced with respect to ψ. By construction, $\mathcal{D}_\sigma$ is supported on $O(\varepsilon^{-2}\log N)$ of the sets in $\mathcal{F}$, as required. □

3.2 Cell-probe ANN data structure

Proof. We first describe a "building block": the data structure with success probability $n^{-O(\varepsilon)}$. The final data structure will simply consist of $n^{O(\varepsilon)}$ such independent data structures, and the query algorithm will query each of them. The building block is simply a randomized *decision tree*, which we build and query recursively.

During recursion, we will have the following mutually exclusive cases, which cover all the possibilities.

- **Case** 1**.** $P = \emptyset$.
- **Case** 2**.** $|P| = 1$.
- **Case** 3**.** $|P| > 1$ and there is a point $x_0 \in X$ such that $|P \cap B_X(x_0, r\Xi(X,\varepsilon)/2)| > |P|/8$.

- **Case** 4. $|P| > 1$ and for every $x_0 \in X$, one has $|P \cap B_X(x_0, r\Xi(X,\varepsilon)/2)| \le |P|/8$.

Preprocessing algorithm During the preprocessing, we do the following depending on the case.

- **Case** 1**.** We do nothing.
- **Case** 2**.** We store the only element $p_0 \in P$.
- **Case** 3**.** We store $x_0 \in X$ and an arbitrary point $p_0 \in P \cap B_X(x_0, r\Xi(X,\varepsilon)/2)$ and call the preprocessing procedure recursively for the set $P \setminus B_X(x_0, r\Xi(X,\varepsilon)/2)$.
- **Case** 4**.** Let ψ be the uniform distribution over P. By the assumption, $\psi \in \Delta_{r\Xi(X,\varepsilon),1/8}(X)$, thus we can apply Theorem 43 to it. Theorem 43 gives a distribution $\mathcal{D}$ over subsets of X. We sample $S \sim \mathcal{D}$, and store the description of S using $O(\log N(\log \varepsilon^{-1} + \log\log N))$ bits. Finally, we call the preprocessing procedure recursively for the sets $P \cap S$ and $P \setminus S$.

Query algorithm The query algorithm proceeds recursively descending down the tree built during the preprocessing. Let $q \in X$ be a query point.

- **Case** 1**.** We return $\perp$.
- **Case** 2**.** If $d_X(p_0, q) \le (\Xi(X,\varepsilon)+1)r$, we return p_0, otherwise, we return $\perp$.
- **Case** 3**.** If $q \in B_X(x_0, (\Xi(X,\varepsilon)+1)r)$, then we return the point $p_0 \in P \cap B_X(x_0, r\Xi(X,\varepsilon)/2)$ we stored, otherwise, we query the data structure for $P \setminus B_X(x_0, r\Xi(X,\varepsilon)/2)$.
- **Case** 4**.** If $q \in S$ (to check this, we reconstruct S from the succinct $O(\log N(\log \varepsilon^{-1} + \log\log N))$-bit description that we stored), then we query the data structure for $P \cap S$, otherwise, we query the data structure for $P \setminus S$.

Now let us analyze the above algorithm.

Claim 3.2.1. *The depth of the built decision tree is* $O(\log n)$.

Proof. Consider the preprocessing procedure. For Case 3, $|P \cap B_X(x_0, r\Xi(X,\varepsilon)/2)| > |P|/8$, thus, we recurse on the dataset of size $(1 - \Omega(1)) \cdot |P|$. The same happens for Case 4, since the set S is $1/8$-balanced with respect to the uniform distribution over P. □

Claim 3.2.2. *The total number of nodes in the built decision tree is* $O(n)$.

Proof. This readily follows from the fact that we always recurse on at most two (possibly empty) sets of points, which are always disjoint. □

Claim 3.2.3. *One can store each tree node using* $O(\log N(\log \varepsilon^{-1} + \log\log N))$ *bits.*

Proof. In a tree node, we need to store $O(1)$ pointers, $O(1)$ points from X, and at most one subset $S \subseteq X$. Pointers require $O(\log n) = O(\log N)$ bits, points require $O(\log N)$ bits, and subsets can be encoded using $O(\log N(\log \varepsilon^{-1} + \log\log N))$ bits. □

Claim 3.2.4. *The space required to store the decision tree is* $O(n \cdot (\log N(\log \varepsilon^{-1} + \log\log N)))$ *bits.*

Proof. This is an easy corollary of Claim 3.2.2 and Claim 3.2.3. □

Claim 3.2.5. *The query procedure inspects* $O(\log n \cdot (\log N(\log \varepsilon^{-1} + \log\log N)))$ *bits of the memory.*

Proof. Similar to Claim 3.2.4, this is an immediate corollary of Claim 3.2.1 and Claim 3.2.3. □

Claim 3.2.6. *For a fixed query point* $q \in X$*, which is within* r *from some* $p^* \in P$*, the probability of returning a point* $p \in P$ *with* $d_X(q,p) \le (\Xi(X,\varepsilon)+1)r$ *is at least* $n^{-O(\varepsilon)}$.

Proof. The desired probability is lower bounded by the probability that q and p^* are not separated in the decision tree in the "Case 4" nodes. This probability is at least $(1-9\varepsilon)^{O(\log n)} = n^{-O(\varepsilon)}$, since by Claim 3.2.1 the depth of the tree is $O(\log n)$, and the distribution $\mathcal{D}$ in any "Case 4" tree node is $(r, 9\varepsilon)$-separating. □

To get the success probability 0.9, we build $n^{O(\varepsilon)}$ decision trees, and when answering a query, we query all of them. The desired bounds can be easily obtained from the above claims for a single tree.

□

3.3 An Inefficient Upper Bound on the Cutting Modulus of any Normed Space

Here, we prove Theorem 24 from Section 1.2 in Chapter 1, which as discussed, immediately gives an upper bound on the cutting modulus of any d-dimensional normed space.

Definition 30 (Re-statement of Non-Linear Spectral Gaps)**.** *Let (X, d_X) be a metric space and for $m \in \mathbb{N}$, $\mu \in \Gamma([m])$. For $p > 0$, the inverse of the* non-linear spectral gap $\gamma(\mu, d_X^p)$ *is the infimum over $\gamma > 0$ such that for any $u_1, \ldots, u_n \in X$,*

$$\mathop{\mathbf{E}}_{\boldsymbol{i},\boldsymbol{j}\sim\rho_\mu}\left[d_X(u_{\boldsymbol{i}}, u_{\boldsymbol{j}})^p\right] \leq \gamma \mathop{\mathbf{E}}_{(\boldsymbol{i},\boldsymbol{j})\sim\mu}\left[d_X(u_{\boldsymbol{i}}, u_{\boldsymbol{j}})^p\right].$$

Theorem 44. *Let $d \in \mathbb{N}$ and consider any normed space $X = (\mathbb{R}^d, \|\cdot\|_X)$. For $m \in \mathbb{N}$ and $\mu \in \Gamma([m])$ over $[m] \times [m]$,*

$$\gamma(\mu, \|\cdot\|_X^2) \lesssim \left(\frac{\log d}{\lambda_2(\mathcal{L}_\mu)}\right)^2,$$

where $\lambda_2(\mathcal{L}_\mu)$ is the second smallest eigenvalue of the normalized Laplacian matrix $\mathcal{L}_\mu = I_n - D_\mu^{-1/2} G_\mu D_\mu^{-1/2}$, where $D_\mu = \mathrm{diag}(\rho_\mu)$, and G_μ is the $m \times m$ symmetric matrix with $(G_\mu)_{i,j} = \mu(i,j)$.

We obtain the following estimate on the cutting modulus of X:

$$\Xi(X, \varepsilon) \lesssim \frac{1 + \log c_2(X)}{\varepsilon^2} \lesssim \frac{1 + \log d}{\varepsilon^2},$$

where the second step follows from the bound $c_2(X) \leq \sqrt{d}$, which, in turn, is an immediate corollary of John's theorem. The remainder of this section is devoted to the proof of Theorem 44.

Let $V_\mu \subset (\mathbb{R}^d)^m$ be the following codimension-1 subspace:

$$V_\mu = \left\{(v_1, v_2, \ldots, v_m) \in (\mathbb{R}^d)^m \;\middle|\; \sum_{i=1}^m \sqrt{\rho_\mu(i)} \cdot v_i = 0\right\}.$$

We denote by $V_\mu^X = (V_\mu, \|\cdot\|_{V_\mu^X})$ the normed space where for $\boldsymbol{v} = (v_1, v_2, \ldots, v_m) \in V_\mu$, the norm is given by $\|\boldsymbol{v}\|_{V_\mu^X} = (\sum_{i=1}^m \|v_i\|_X^2)^{1/2}$. Denote by $\mathcal{A}_\mu \colon V_\mu \to V_\mu$ the following linear map:

$$(\mathcal{A}_\mu \boldsymbol{v})_i = \sum_{j=1}^m \frac{\mu(i,j)}{\sqrt{\rho_\mu(i)\rho_\mu(j)}} \cdot v_j.$$

Putting it differently, $\mathcal{A}_\mu$ acts on a size-m tuple of d-dimensional vectors the same way as $A_\mu = D_\mu^{-1/2} G_\mu D_\mu^{-1/2}$ acts on a size-m tuple of scalars. It is immediate to check that the image of $\mathcal{A}_\mu$ indeed lies in V_μ; this follows from the fact that $(\rho_\mu(1)^{1/2}, \rho_\mu(2)^{1/2}, \ldots, \rho_\mu(m)^{1/2})$ is an eigenvector of A_μ. Let $\mathcal{I} \colon V_\mu \to V_\mu$ be the identity map.

We show that Theorem 44 readily follows from the following lemma.

Lemma 3.3.1. *One has:* $\left\|(\mathcal{I} - \mathcal{A}_\mu)^{-1}\right\|_{V_\mu^X \to V_\mu^X} \lesssim (1 + \log c_2(X))/\lambda_2(\mathcal{L}_\mu)$.

Proof of "Lemma 3.3.1 ⇒ Theorem 44". Indeed, an immediate reformulation of Lemma 3.3.1 is that for every $v_1, v_2, \ldots, v_m \in \mathbb{R}^d$ such that $\sum_{i=1}^m \rho_\mu(i)^{1/2} \cdot v_i = 0$,

$$\sum_{i=1}^m \|v_i\|_X^2 \lesssim \left(\frac{1 + \log c_2(X)}{\lambda_2(\mathcal{L}_\mu)}\right)^2 \sum_{i=1}^m \left\| v_i - \sum_{j=1}^m \frac{\mu(i,j) \cdot v_j}{\sqrt{\rho_\mu(i)\rho_\mu(j)}} \right\|_X^2. \tag{3.3}$$

Our goal is to show that for every $u_1, u_2, \ldots, u_m \in \mathbb{R}^d$,

$$\sum_{i,j=1}^m \rho_\mu(i)\rho_\mu(j) \cdot \|u_i - u_j\|_X^2 \lesssim \left(\frac{1 + \log c_2(X)}{\lambda_2(\mathcal{L}_\mu)}\right)^2 \sum_{i,j=1}^m \mu(i,j) \cdot \|u_i - u_j\|_X^2. \tag{3.4}$$

Since (3.4) contains only pairwise differences of u_i's, we can assume that $\sum_{i=1}^m \rho_\mu(i) \cdot u_i = 0$. We set $v_1, \ldots, v_m \in \mathbb{R}^d$ by letting $v_i = \rho_\mu(i)^{1/2} \cdot u_i$, so that $\sum_{i=1}^m \rho_\mu(i)^{1/2} \cdot v_i = 0$ and, thus, (3.3) applies. On the one hand,

$$\begin{aligned}
\sum_{i,j=1}^m \rho_\mu(i)\rho_\mu(j)\|u_i - u_j\|_X^2 &\le \sum_{i,j=1}^m \rho_\mu(i)\rho_\mu(j)\Big(\|u_i\|_X + \|u_j\|_X\Big)^2 \\
&\le 2\sum_{i,j=1}^m \rho_\mu(i)\rho_\mu(j)\Big(\|u_i\|_X^2 + \|u_j\|_X^2\Big) \\
&= 4\sum_{i=1}^m \rho_\mu(i)\|u_i\|_X^2 = 4\sum_{i=1}^m \|v_i\|_X^2.
\end{aligned} \tag{3.5}$$

On the other hand,

$$\sum_{i=1}^m \left\| v_i - \sum_{j=1}^m \frac{\mu(i,j)\cdot v_j}{\sqrt{\rho_\mu(i)\rho_\mu(j)}} \right\|_X^2 = \sum_{i=1}^m \left\| \sum_{j=1}^m \frac{\mu(i,j)}{\sqrt{\rho_\mu(i)}} \left(\frac{v_i}{\sqrt{\rho_\mu(i)}} - \frac{v_j}{\sqrt{\rho_\mu(j)}} \right) \right\|_X^2$$
$$= \sum_{i=1}^m \left\| \sum_{j=1}^m \frac{\mu(i,j)}{\sqrt{\rho_\mu(i)}} \cdot (u_i - u_j) \right\|_X^2 \le \sum_{i=1}^m \left(\sum_{j=1}^m \frac{\mu(i,j)}{\sqrt{\rho_\mu(i)}} \cdot \left\| u_i - u_j \right\|_X \right)^2$$
$$\le \sum_{i,j=1}^m \mu(i,j) \cdot \left\| u_i - u_j \right\|_X^2, \tag{3.6}$$

where the third step is due to the triangle inequality, and the fourth step is due to Jensen's inequality. Combining (3.3), (3.5) and (3.6), we obtain (3.4). □

We now proceed to the proof of Lemma 3.3.1. Let $H = (\mathbb{R}^d, \|\cdot\|_H)$ be a Hilbert space such that for every $v \in \mathbb{R}^d$, one has $\|v\|_H \le \|v\|_X \le c_2(X)\cdot\|v\|_H$, whose existence follows from John's theorem. We define the normed space $V_\mu^H = (V_\mu, \|\cdot\|_{V_\mu^H})$ similarly to V_μ^X: the norm $\|\boldsymbol{v}\|_{V_\mu^H}$ for $\boldsymbol{v} = (v_1, v_2, \dots, v_m) \in V_\mu$ is given by $\|\boldsymbol{v}\|_{V_\mu^H} = (\sum_{i=1}^m \|v_i\|_H^2)^{1/2}$. Clearly, for every $\boldsymbol{v} \in V$,

$$\|\boldsymbol{v}\|_{V_\mu^H} \le \|\boldsymbol{v}\|_{V_\mu^X} \le c_2(X)\cdot\|\boldsymbol{v}\|_{V_\mu^H}. \tag{3.7}$$

Finally, we define $\widetilde{A}_\mu = (A_\mu + I)/2$ and $\widetilde{\mathcal{A}}_\mu = (\mathcal{A}_\mu + \mathcal{I})/2$. Let us observe that $\mathcal{I} - \widetilde{\mathcal{A}}_\mu = (\mathcal{I} - \mathcal{A}_\mu)/2$, so $\|(\mathcal{I}-\mathcal{A}_\mu)^{-1}\|_{V_\mu^X \to V_\mu^X} \le \|(\mathcal{I}-\widetilde{\mathcal{A}}_\mu)^{-1}\|_{V_\mu^X \to V_\mu^X}/2$, thus it is enough to show that

$$\left\|(\mathcal{I}-\widetilde{\mathcal{A}}_\mu)^{-1}\right\|_{V_\mu^X \to V_\mu^X} \lesssim \frac{1+\log c_2(X)}{\lambda_2(L_\mu)}. \tag{3.8}$$

One can see that (3.8) is an immediate corollary of the following three statements together with (3.7). Let us note that Lemma 3.3.4 is the place where the logarithmic dependence on the distortion shows up.

Claim 3.3.2. *One has* $\|\widetilde{\mathcal{A}}_\mu\|_{V_\mu^X \to V_\mu^X} \le 1$.

Claim 3.3.3. *One has* $\|\widetilde{\mathcal{A}}_\mu\|_{V_\mu^H \to V_\mu^H} \le 1 - \lambda_2(\mathcal{L}_\mu)/2$.

Lemma 3.3.4. *Let* $\|\cdot\|_P$ *and* $\|\cdot\|_Q$ *be two norms on* $\mathbb{R}^{d'}$ *such that for some* $\Phi \ge 1$ *and for every* $u \in \mathbb{R}^{d'}$ *one has* $\|u\|_Q \le \|u\|_P \le \Phi\cdot\|u\|_Q$. *Suppose that* $T\colon \mathbb{R}^{d'} \to \mathbb{R}^{d'}$ *is a*

linear map such that $\|T\|_{P\to P} \le 1$ *and* $\|T\|_{Q\to Q} \le 1-\varepsilon$ *for some* $0 < \varepsilon < 1$. *Then,* $\|(I-T)^{-1}\|_{P\to P} \lesssim \frac{1+\log\Phi}{\varepsilon}$.

Proof of Claim 3.3.2. For every $\boldsymbol{v} = (v_1, v_2, \ldots, v_m) \in V_\mu^X$ one has

$$\|\mathcal{A}_\mu \boldsymbol{v}\|_{V_\mu^X}^2 = \sum_{i=1}^m \|(\mathcal{A}_\mu \boldsymbol{v})_i\|_X^2 = \sum_{i=1}^m \left\| \sum_{j=1}^m \frac{\mu(i,j)\cdot v_j}{\sqrt{\rho_\mu(i)\rho_\mu(j)}} \right\|_X^2 \le \sum_{i=1}^m \left(\sum_{j=1}^m \frac{\mu(i,j)}{\rho_\mu(i)} \left\| \sqrt{\frac{\rho_\mu(i)}{\rho_\mu(j)}} \cdot v_j \right\|_X \right)^2$$
$$\le \sum_{i=1}^m \sum_{j=1}^m \frac{\mu(i,j)}{\rho_\mu(i)} \left\| \sqrt{\frac{\rho_\mu(i)}{\rho_\mu(j)}} \cdot v_j \right\|_X^2 = \sum_{i=1}^m \sum_{j=1}^m \frac{\mu(i,j)}{\rho_\mu(j)} \|v_j\|_X^2 = \sum_{j=1}^m \|v_j\|_X^2 = \|\boldsymbol{v}\|_{V_\mu^X}^2,$$

where the third step is by the triangle inequality, and the fourth step is by Jensen's inequality. Hence, $\|\mathcal{A}_\mu\|_{V_\mu^X \to V_\mu^X} \le 1$. But this implies that $\|\widetilde{\mathcal{A}}_\mu\|_{V_\mu^X \to V_\mu^X} \le 1$ as well. □

Proof of Claim 3.3.3. Let us first observe that for every $u \in \mathbb{R}^m$ such that $\sum_{i=1}^m (\rho_\mu(i))^{1/2} \cdot u_i = 0$, one has

$$\|\widetilde{A}_\mu u\|_2 \le \left(1 - \frac{\lambda_2(\mathcal{L}_\mu)}{2}\right) \cdot \|u\|_2, \tag{3.9}$$

since $\widetilde{A}_\mu$ is positive semidefinite, its largest eigenvalue is 1, the corresponding eigenvector is $((\rho_\mu(i))^{1/2})_{i=1}^m$, and the second largest eigenvalue is $1 - \lambda_2(\mathcal{L}_\mu)/2$.

The desired inequality reduces to (3.9) as follows. Since H is a Hilbert space, there exists an orthogonal (not necessarily orthonormal) basis $e_1, e_2, \ldots, e_d \in \mathbb{R}^d$ such that for every $u \in \mathbb{R}^d$, one has $\|u\|_H^2 = \sum_{i=1}^d \langle u, e_i\rangle^2$. For $1 \le i \le d$ and $\boldsymbol{v} = (v_1, v_2, \ldots, v_m) \in V_\mu^H$, define $\pi_i(\boldsymbol{v}) = (\langle v_1, e_i\rangle, \langle v_2, e_i\rangle, \ldots, \langle v_m, e_i\rangle) \in \mathbb{R}^m$. Then, $\|\boldsymbol{v}\|_{V_\mu^H}^2 = \sum_{i=1}^d \|\pi_i(\boldsymbol{v})\|_2^2$. One has:

$$\|\widetilde{\mathcal{A}}_\mu \boldsymbol{v}\|_{V_\mu^H}^2 = \sum_{i=1}^d \|\pi_i(\widetilde{\mathcal{A}}_\mu \boldsymbol{v})\|_2^2 = \sum_{i=1}^d \|\widetilde{A}_\mu \pi_i(\boldsymbol{v})\|_2^2 \le \left(1 - \frac{\lambda_2(\mathcal{L}_\mu)}{2}\right)^2 \sum_{i=1}^d \|\pi_i(\boldsymbol{v})\|_2^2$$
$$= \left(1 - \frac{\lambda_2(\mathcal{L}_\mu)}{2}\right)^2 \|\boldsymbol{v}\|_{V_\mu^H}^2.$$

□

Proof of Lemma 3.3.4. For every $k \ge 1$, one has

$$\|T^k\|_{P\to P} \le \Phi \cdot \|T^k\|_{Q\to Q} \le \Phi \cdot (1-\varepsilon)^k.$$

Thus, we can choose $k^* \lesssim \frac{1+\log\Phi}{\varepsilon}$ such that $\|T^{k^*}\|_{P\to P} \leq 1/2$. Finally, we have:

$$\|(I-T)^{-1}\|_{P\to P} \leq \sum_{k=0}^{\infty} \|T^k\|_{P\to P} \leq k^* \cdot \sum_{i=0}^{\infty} \|T^{ik^*}\|_{P\to P} \leq k^* \cdot \sum_{i=0}^{\infty} (1/2)^i = 2k^* \lesssim \frac{1+\log\Phi}{\varepsilon}$$

as desired. □

Chapter 4

Constructive Bounds on the Cutting Modulus of Any Norm

In this chapter, we present the formal arguments to claims made in Section 1.4 Chapter 1. We give various upper bounds on the cutting modulus (recall, Definition 15) which yield time efficient ANN algorithms.

Definition 31 (Non-linear Rayleigh Quotient)**.** *Let (X, d_X) be a metric space, and let μ be any symmetric probability distribution supported on finitely many pairs of X, and let ρ be its marginal distribution. For any $r \in (0, \infty)$, the* non-linear Rayleigh quotient *is given by*

$$\mathsf{R}(\mu, d_X^r) = \frac{\mathop{\mathbf{E}}_{(\boldsymbol{x},\boldsymbol{y})\sim\mu}[d_X(\boldsymbol{x},\boldsymbol{y})^r]}{\mathop{\mathbf{E}}_{\boldsymbol{x},\boldsymbol{y}\sim\rho}[d_X(\boldsymbol{x},\boldsymbol{y})^r]}.$$

The main lemma, which illustrates the approach for bounding the cutting modulus is summarized in the following lemma.

Lemma 4.0.1 (Comparison of Rayleigh Quotient to ℓ_1)**.** *Let $d \in \mathbb{N}$ and let $X = (\mathbb{R}^d, \|\cdot\|_X)$ be a normed space. Suppose that for any symmetric probability distribution μ over finitely many pairs of points in $\mathbb{R}^d$, there exists $f\colon \mathbb{R}^d \to \mathbb{R}^d$ such that for $\Psi\colon \mathbb{R}^{\geq 0} \to \mathbb{R}^{\geq 0}$ with $\Psi(\varepsilon) \to 0$ as $\varepsilon \to 0$ and $r > 0$,*

$$\mathsf{R}(f(\mu), \|\cdot\|_1) \leq \Psi(\mathsf{R}(\mu, \|\cdot\|_X^r)).$$

Then, $\Xi(X, \varepsilon) \leq 4(1/\Psi^{-1}(\varepsilon))^{1/r}$, where $\Psi^{-1}(\varepsilon) = \sup\{\varepsilon_0 \geq 0 : \forall \varepsilon' < \varepsilon_0, \Psi(\varepsilon') \leq \varepsilon\}$.

Proof. Let μ be any symmetric probability distribution supported on finitely many pairs of points in $\mathbb{R}^d$ within distance at most 1, and let ρ be its marginal distribution. Suppose that for any set $B \subset \mathbb{R}^d$ with $\operatorname{diam}_X(B) \leq R$, we have $\rho(B) \leq 1/2$, where we set

$R = 4(1/\Psi^{-1}(\varepsilon))^{1/r}$. Then, we notice that

$$\mathop{\mathbf{E}}_{(\boldsymbol{x},\boldsymbol{y})\sim\mu}[\|\boldsymbol{x}-\boldsymbol{y}\|_X^r] \le 1 \qquad \text{and} \qquad \mathop{\mathbf{E}}_{\boldsymbol{x},\boldsymbol{y}\sim\rho}[\|\boldsymbol{x}-\boldsymbol{y}\|_X^r] \ge \frac{1}{2}\cdot\left(\frac{R}{2}\right)^r,$$

where in the second inequality, we are using the fact that for any $\boldsymbol{x}$ sampled from ρ, $B_X(\boldsymbol{x}, R/2)$ has diameter at most R, and hence with probability at least $1/2$ over the draw of $\boldsymbol{y} \sim \rho$, $\|\boldsymbol{x}-\boldsymbol{y}\|_X \ge R/2$. Hence, $\mathsf{R}(\mu, \|\cdot\|_X^r) \le 2(2/R)^r$, which implies that $\mathsf{R}(f(\mu), \|\cdot\|_1) \le \Psi(2(2/R)^r) \le \varepsilon$. By Lemma 1.4.1, there exists a cut $S \subset \mathbb{R}^d$ of conductance $\Phi_\mu(S) \le \varepsilon$. □

4.1 Relating Rayleigh Quotients with Holder Homeomorphisms

Definition 32 (Holder Homeomorphism between Spheres)**.** *Let $X = (\mathbb{R}^d, \|\cdot\|_X)$ and $Y = (\mathbb{R}^{d'}, \|\cdot\|)$ be two normed spaces. For $c \ge 0$ and $\alpha \in (0,1]$, function $\phi\colon S(X) \to S(Y)$ is an α-Holder homeomorphism with Holder constant c if it is bijective and for every $a, b \in S(X)$,*

$$\|\phi(a) - \phi(b)\|_Y \le c \cdot \|a-b\|_X^\alpha.$$

Lemma 4.1.1 (Radial Extensions of Holder Homeomorphism between Spheres)**.** *Let $X = (\mathbb{R}^d, \|\cdot\|_X)$ and $Y = (\mathbb{R}^{d'}, \|\cdot\|_Y)$ be two normed spaces, and for $c \ge 0$ and $\alpha \in (0,1]$, let $\phi\colon S(X) \to S(Y)$ be an α-Holder homeomorphism with constant c. Then, for $r \ge 1$, the radial extension $f_{\phi,r}\colon \mathbb{R}^d \to \mathbb{R}^{d'}$ given by*

$$f_{\phi,r}(x) = \|x\|_X^r \cdot \phi(x/\|x\|_X) \tag{4.1}$$

satisfies $\|f_{\phi,r}(x)\|_Y = \|x\|_X^r$ for every $x \in \mathbb{R}^d$, and for every $x, y \in \mathbb{R}^d$,

$$\|f_{\phi,r}(x) - f_{\phi,r}(y)\|_Y \le \left(2^\alpha c + 2^{1-\alpha} r\right)\|x-y\|_X^\alpha \left(\|x\|_X^{r-\alpha} + \|y\|_X^{r-\alpha}\right). \tag{4.2}$$

Proof. The first claim, that $\|f_{\phi,r}(x)\|_Y = \|x\|_X^r$ is trivial, as

$$\|f_{\phi,r}(x)\|_Y = \|x\|_X^r \|\phi(x/\|x\|_X)\|_Y = \|x\|_X^r,$$

since $\phi\colon S(X) \to S(Y)$. For the second claim, let $x, y \in \mathbb{R}^d$, and let $a = x/\|x\|_X$, $b = y/\|y\|_X$, and $t = \|y\|_X/\|x\|_X$, which we assume without loss of generality, that $t \leq 1$.

$$\begin{aligned}\frac{\|f_{\phi,r}(x) - f_{\phi,r}(y)\|_Y}{\|x\|_X^r} &\leq \|\phi(a) - \phi(b)\|_Y + \|\phi(b) - t^r\phi(b)\|_Y = c\|a-b\|_X^\alpha + (1-t)\cdot r \\ &\leq c\|a-b\|_X^\alpha + r\left(\|a - tb\|_X\right) \leq \|a-tb\|_X^\alpha \left(2^\alpha c + r\|a-tb\|_X^{1-\alpha}\right) \\ &\leq \left(2^\alpha c + 2^{1-\alpha} r\right)\|a-tb\|_X^\alpha.\end{aligned}$$

where I used the fact that

$$\|a-b\|_X \leq \|a-tb\|_X + \|tb-b\|_X = \|a-tb\|_X + (1-t) = \|a-tb\|_X + \|a\|_X - t\|b\|_Y \leq 2\|a-tb\|_X.$$

Therefore, we conclude

$$\|f_{\phi,r}(x) - f_{\phi,r}(y)\|_Y \leq \left(2^\alpha c + 2^{1-\alpha} r\right)\|x-y\|_X^\alpha \cdot \|x\|_X^{r-\alpha},$$

and the remaining term $(2^\alpha c + 2^{1-\alpha} r)\,\|x-y\|_X^\alpha \cdot \|y\|_X^{r-\alpha}$ is added in case $\|y\|_X \geq \|x\|_X$. $\square$

Theorem 45 (Matousek's Extrapolation [103, 16])**.** *Let $X = (\mathbb{R}^d, \|\cdot\|_X)$ and $Y = (\mathbb{R}^{d'}, \|\cdot\|_Y)$ be any normed spaces, and suppose that for $\alpha \in (0, 1]$ and $c \geq 0$, there exists an α-Holder homeomorphism $\phi\colon S(X) \to S(Y)$ with constant c. Let μ be any symmetric probability distribution supported on finitely many pairs of points in X, and let ρ be the marginal of μ. Suppose that, for $r > \alpha$ and $s \geq 1$, the distribution ρ under $f_{\phi,r}$ is approximately centered, i.e.,*

$$\left\| \mathop{\mathbf{E}}_{\boldsymbol{x}\sim\rho}[f_{\phi,r}(\boldsymbol{x})] \right\|_Y \leq \left(\frac{1}{2^{s+1}} \cdot \mathop{\mathbf{E}}_{\boldsymbol{x}\sim\rho}\left[\|f_{\phi,r}(\boldsymbol{x})\|_Y^s\right] \right)^{1/s}, \tag{4.3}$$

then,

$$\mathsf{R}(f_{\phi,r}(\mu), \|\cdot\|_Y^s) \leq 2^{2s+2}\,(c+r)^s \cdot \mathsf{R}(\mu, \|\cdot\|_X^{sr})^{\alpha/r}. \tag{4.4}$$

Proof. We upper-bound the left-hand side of (4.4) by upper bounding and lower bounding the following quantities, respectively,

$$\mathop{\mathbf{E}}_{(\boldsymbol{x},\boldsymbol{y})\sim\mu}\left[\|f_{\phi,r}(\boldsymbol{x}) - f_{\phi,r}(\boldsymbol{y})\|_Y\right] \qquad \text{and} \qquad \mathop{\mathbf{E}}_{\boldsymbol{x},\boldsymbol{y}\sim\rho}\left[\|f_{\phi,r}(\boldsymbol{x}) - f_{\phi,r}(\boldsymbol{y})\|_Y\right].$$

First, the upper bound follows from Lemma 4.1.1:

$$\begin{aligned}\mathop{\mathbf{E}}_{(\boldsymbol{x},\boldsymbol{y})\sim\mu}\left[\|f_{\phi,r}(\boldsymbol{x}) - f_{\phi,r}(\boldsymbol{y})\|_Y^s\right] &\leq \left(2^\alpha c + 2^{1-\alpha} r\right)^s \mathop{\mathbf{E}}_{(\boldsymbol{x},\boldsymbol{y})\sim\mu}\left[\|\boldsymbol{x}-\boldsymbol{y}\|_X^{s\alpha}\left(\|\boldsymbol{x}\|_X^{sr-s\alpha} + \|\boldsymbol{y}\|_X^{sr-s\alpha}\right)\right] \\ &\leq 2\left(2^\alpha c + 2^{1-\alpha} r\right)^s \left(\mathop{\mathbf{E}}_{(\boldsymbol{x},\boldsymbol{y})\sim\mu}\left[\|\boldsymbol{x}-\boldsymbol{y}\|_X^{sr}\right]\right)^{\alpha/r} \left(\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\rho}\left[\|\boldsymbol{x}\|_X^{sr}\right]\right)^{(r-\alpha)/r},\end{aligned}$$

by applying Holder's inequality with conjugates r/α and $r/(r-\alpha)$. On the other hand, we use (4.3) and the fact $\|f_{\phi,r}(x)\|_Y = \|x\|_X^r$ for the lower bound. First, notice that

$$\begin{aligned}\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\rho}\left[\|f_{\phi,r}(\boldsymbol{x})\|_Y^s\right] &\leq 2^s \mathop{\mathbf{E}}_{\boldsymbol{x}\sim\rho}\left[\left\|f_{\phi,r}(\boldsymbol{x}) - \mathop{\mathbf{E}}_{\boldsymbol{y}\sim\rho}\left[f_{\phi,r}(\boldsymbol{y})\right]\right\|_Y^s\right] + 2^s \left\|\mathop{\mathbf{E}}_{\boldsymbol{y}\sim\rho}\left[f_{\phi,r}(\boldsymbol{y})\right]\right\|_Y^s \\ &\leq 2^s \mathop{\mathbf{E}}_{\boldsymbol{x},\boldsymbol{y}\sim\rho}\left[\|f_{\phi,r}(\boldsymbol{x}) - f_{\phi,r}(\boldsymbol{y})\|_Y^s\right] + \frac{1}{2}\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\rho}\left[\|f_{\phi,r}(\boldsymbol{x})\|_Y^s\right],\end{aligned}$$

where we used the triangle inequality and Jensen inequality (since $s \geq 1$). This implies

$$\mathop{\mathbf{E}}_{\boldsymbol{x},\boldsymbol{y}\sim\rho}\left[\|f_{\phi,r}(\boldsymbol{x}) - f_{\phi,r}(\boldsymbol{y})\|_Y^s\right] \geq \frac{1}{2^{s+1}}\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\rho}\left[\|f_{\phi,r}(\boldsymbol{x})\|_Y^s\right] = \frac{1}{2^{s+1}}\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\rho}\left[\|\boldsymbol{x}\|_X^{sr}\right].$$

Combining the upper and lower bounds, we have

$$\begin{aligned}\mathsf{R}(f(\mu), \|\cdot\|_Y^s) &\leq 2\left(2^\alpha c + 2^{1-\alpha} r\right)^s \cdot 2^{s+1} \cdot \frac{\left(\mathop{\mathbf{E}}_{(\boldsymbol{x},\boldsymbol{y})\sim\mu}\left[\|\boldsymbol{x}-\boldsymbol{y}\|_X^{sr}\right]\right)^{\alpha/r}}{\left(\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\rho}\left[\|f_{\phi,r}(\boldsymbol{x})\|_X^{sr}\right]\right)^{\alpha/r}} \\ &\leq 2^{2s+2}\cdot(c+r)^s \cdot \mathsf{R}(\mu, \|\cdot\|_X^{sr})^{\alpha/r}.\end{aligned}$$

□

4.1.1 ℓ_p Spaces

Definition 33 (Mazur Map into ℓ_1)**.** *Let $d \in \mathbb{N}$ and $p \geq 1$. We let $M_{p,1} \colon \mathbb{R}^d \to \mathbb{R}^d$ be the map which takes the vector $x \in \mathbb{R}^d$ to*

$$(x_1, \ldots, x_d) \overset{M_{p,1}}{\longmapsto} (\operatorname{sign}(x_1) \cdot |x_1|^p, \operatorname{sign}(x_2) \cdot |x_2|^p, \ldots, \operatorname{sign}(x_d) \cdot |x_d|^p)$$

Lemma 4.1.2. *Fix $d \in \mathbb{N}$ and $p \geq 1$. Then $M_{p,1} \colon S(\ell_p^d) \to S(\ell_1^d)$ is a 1-Holder homeomorphism of constant at most $2p$.*

Proof. First, notice that $M_{p,1}$ is a bijection, and that for any $x \in S(\ell_p^d)$,

$$\|M_{p,1}(x)\|_1 = \sum_{i=1}^{d} |x_i|^p = 1.$$

For the 1-Holder estimate, consider $x, y \in S(\ell_p^d)$, we upper bound each coordinate by taking its linear approximation, and then using Holder's inequality with conjugates p and $p/(p-1)$. Specifically, we have

$$\begin{aligned}
\sum_{i=1}^{d} |\operatorname{sign}(x_i)|x_i|^p - \operatorname{sign}(y_i)|y_i|^p| &\leq \sum_{i=1}^{d} p|x_i - y_i| \max\{|x_i|, |y_i|\}^{p-1} \\
&\leq p \left(\sum_{i=1}^{d} |x_i - y_i|^p \right)^{1/p} \left(\sum_{i=1}^{d} \max\{|x_i|, |y_i|\}^p \right)^{(p-1)/p} \\
&\leq 2^{(p-1)/p} p \|x - y\|_p,
\end{aligned}$$

where we notice that $\sum_{i=1}^{d} \max\{|x_i|, |y_i|\}^p \leq 2$ since $x, y \in S(\ell_p^d)$. □

Lemma 4.1.3. *Fix $d \in \mathbb{N}$ and $p \geq 1$. Let μ be any symmetric probability distribution supported on finitely many pairs of points in $\mathbb{R}^d$. There exists a point $z \in \mathbb{R}^d$ and a function $f \colon \mathbb{R}^d \to \mathbb{R}^d$ such that*

$$f(x) = M_{p,1}(x - z),$$

and we have

$$\mathsf{R}(f(\mu), \|\cdot\|_1) \leq 24p \cdot \mathsf{R}(\mu, \|\cdot\|_p^p)^{1/p}.$$

Proof. Letting ρ be the marginal distribution of μ, we first notice that there exists $z \in \mathbb{R}^d$ such that

$$\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\rho}[M_{p,1}(\boldsymbol{x} - z)] = 0,$$

which follows from the fact that $M_{p,1}$ acts coordinate-wise, and that each $i \in [d]$ has $\mathbf{E}_{\boldsymbol{x}\sim\rho}[M_{p,1}(\boldsymbol{x} - z)_i]$ goes from $-\infty$ to ∞ as z_i goes from $-\infty$ to ∞. We therefore may consider the distribution $\tilde{\mu}$ given by $(\boldsymbol{x} - z, \boldsymbol{y} - z)$ for $(\boldsymbol{x}, \boldsymbol{y}) \sim \mu$, whose marginal $\tilde{\rho}$ satisfies (4.3) with $f_{\phi,r} \colon \mathbb{R}^d \to \mathbb{R}^d$ for $\phi = M_{p,1}$ and $r = p$. In particular, notice that $f_{\phi,r} = M_{p,1}$ when viewed as a map $\mathbb{R}^d \times \mathbb{R}^d$. Thus, we apply Theorem 45 with $s = 1$, $c = 2p$ and $\alpha = 1$ to obtain

$$\mathsf{R}(f(\mu), \|\cdot\|_1) = \mathsf{R}(f_{\phi,r}(\tilde{\mu}), \|\cdot\|_1) \leq 24p \cdot \mathsf{R}(\tilde{\mu}, \|\cdot\|_p^p)^{1/p} = 24p \cdot \mathsf{R}(\mu, \|\cdot\|_p^p)^{1/p}.$$

□

Corollary 4.1.4. *For any $d \in \mathbb{N}$ and any $p \in \mathbb{N}$, $\Xi(\ell_p^d, \varepsilon) = O(p/\varepsilon)$.*

Proof. We apply Lemma 4.1.3 and Lemma 4.0.1. □

By utilizing Lemma 4.1.3, it suffices to relate Rayleigh quotients in a metric space to that of ℓ_2. Since one may then apply $M_{2,1}$ in order to relate to Rayleigh quotients in ℓ_1, and hence obtain sparse cuts.

4.1.2 A Good Translation Always Exists

Theorem 46 (A Good Translation Always Exists [16])**.** *Let $X = (\mathbb{R}^d, \|\cdot\|_X)$ and $Y = (\mathbb{R}^{d'}, \|\cdot\|_Y)$ be any normed spaces. Suppose that for $\alpha \in (0, 1]$ and $c \geq 0$, the function $\phi \colon S(X) \to S(Y)$ is an α-Holder homeomorphism with constant c. For any probability distribution ρ supported on finitely many points in $\mathbb{R}^d$ and any $r > 1$, there exists a point $z \in \mathbb{R}^d$ such that*

$$\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\rho}[f_{\phi,r}(\boldsymbol{x} - z)] = 0.$$

Furthermore, if, for $R \in \mathbb{R}$, ρ is supported in $B_X(0, R)$, the point z satisfies $\|z\|_X \leq 8^{r/\alpha} R$.

Proof. Consider the continuous function $h\colon (\mathbb{R}^d, \|\cdot\|_Y) \to (\mathbb{R}^d, \|\cdot\|_Y)$ which maps a vector $u \in \mathbb{R}^d$ by

$$h(u) = \mathop{\mathbf{E}}_{\boldsymbol{x}\sim\rho}\left[f_{\phi,r}(\boldsymbol{x} + f_{\phi,r}^{-1}(u))\right].$$

We will prove that h is a surjection, so that there exists $u \in \mathbb{R}^d$ where $h(u) = 0$. Then, we let $z = -f_{\phi,r}^{-1}(u)$, and we obtain the desired center. Notice that for a fixed setting of ρ, as u becomes a vector with very large norm, $h(u)$ tends to be very close to u, i.e.,

$$\begin{aligned}
\|h(u) - u\|_Y &\leq \mathop{\mathbf{E}}_{\boldsymbol{x}\sim\rho}\left[\left\|f_{\phi,r}(\boldsymbol{x} + f_{\phi,r}^{-1}(u)) - f_{\phi,r}(f_{\phi,r}^{-1}(u))\right\|_Y\right] \\
&\leq \mathop{\mathbf{E}}_{\boldsymbol{x}\sim\rho}\left[\|\boldsymbol{x}\|_X^{\alpha}\left(\|\boldsymbol{x} + f_{\phi,r}^{-1}(u)\|_X^{r-\alpha} + \|f_{\phi,r}^{-1}(u)\|_X^{r-\alpha}\right)\right] \\
&\leq 2^{r-\alpha}\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\rho}\left[\|\boldsymbol{x}\|_X^{r}\right] + \mathop{\mathbf{E}}_{\boldsymbol{x}\sim\rho}\left[\|\boldsymbol{x}\|_X^{\alpha}\right]\cdot 2^{r-\alpha+1}\cdot\|f_{\phi,r}^{-1}(u)\|_X^{r-\alpha} \\
&= 2^{r-\alpha}\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\rho}\left[\|\boldsymbol{x}\|_X^{r}\right] + 2^{r-\alpha+1}\mathop{\mathbf{E}}_{\boldsymbol{x}\sim\rho}\left[\|\boldsymbol{x}\|_X^{\alpha}\right]\cdot\|u\|_Y^{1-\alpha/r} = o(\|u\|_Y), \qquad (4.5)
\end{aligned}$$

Claim 4.1.5. *Let $h\colon \mathbb{R}^d \to \mathbb{R}^d$ be any continuous function such that for some norm $\|\cdot\|$, $\|h(u) - u\| \leq o(\|u\|)$. Then, h is surjective.*

Consider the basis vector $e_{d+1} \in S^d$ and let $\tau\colon S^{d-1} \setminus \{e_{d+1}\} \to \mathbb{R}^d$ be the homeomorphism given by stereographic projection, and consider the map $\tilde{h}\colon S^d \to S^d$ given by

$$\tilde{h}(v) = \begin{cases} \tau^{-1}(h(\tau(v))) & v \neq e_{d+1} \\ e_{d+1} & v = e_{d+1} \end{cases},$$

and notice that since $h(u) \to \infty$ as $u \to \infty$, $\tilde{h}$ is continuous. We verify that $\tilde{h}$ is homotopic to the identity map by showing that the map $\tilde{F}\colon S^d \times [0,1] \to S^d$ given by

$$\tilde{F}(v,t) = \begin{cases} e_{d+1} & v = e_{d+1} \\ \tau^{-1}(t\tau(v) + (1-t)h(\tau(v))) & v \neq e_{d+1} \end{cases}$$

is continuous in $S^d \times [0,1]$. Notice that τ is a homeomorphism, and $(u,t) \mapsto tu + (1-t)h(u)$ is continuous over $\mathbb{R}^d \times [0,1]$, it suffices to show that $\tilde{F}(v,t)$ is continuous at $e_{d+1} \times [0,1]$.

In order to do this, it suffices to show that for every $t \in [0,1]$ and for large enough radii $r \in \mathbb{R}_{\geq 0}$, every $u \in \mathbb{R}^d$ with $\|u\| \geq r$ has $\|tu + (1-t)h(u)\| \geq \delta(r)$, for $\delta \to \infty$ as $r \to \infty$. In particular,

$$\|tu + (1-t)h(u)\| \geq \|u\| - (1-t)\|h(u) - u\| \geq (1 - o(1))\|u\|.$$

Hence, $\tilde{h}$ is surjective, and so is h (see Exercise 5 in page 91 of [19]).

When $h(u) = 0$, then by (4.5), if all $\boldsymbol{x} \sim \rho$ have $\|\boldsymbol{x}\|_X \leq R$,

$$\|u\|_Y \leq \max\left\{2^{r-\alpha+1}R^r, 2^{r-\alpha+2}R^\alpha \|u\|_Y^{1-\alpha/r}\right\}.$$

In the case the maximum is achieved by the left-most term, we have $\|u\|_Y = \|z\|_X^r \leq 2^{r-\alpha+1}R^r$, so that $\|z\|_X \leq 2R$. In the case the maximum is achieved by the right-most term, then $\|u\|_Y^{\alpha/r} \leq 2^{r-\alpha+2}R^\alpha$, so that $\|z\|_X \leq 8^{r/\alpha}R$. □

4.1.3 Schatten-p spaces

Definition 34 (Schatten-p normed space)**.** *For $d \in \mathbb{N}$ and $p \in [1,\infty)$, the normed space $S_p = (\mathbb{R}^{d\times d}, \|\cdot\|_{S_p})$ is defined over $d \times d$ matrices of real numbers, and the $\|x\|_{S_p}^p = \sum_{i=1}^d |\sigma_i(x)|^p$, where $\sigma_i(x)$ is the i-th singular value of x. We note that S_2 is isomorphic (via identity map) to $\ell_2^{d\times d}$.*

Definition 35 (Non-commutative Mazur Map into S_2)**.** *Let $d \in \mathbb{N}$ and $p \in [1,\infty)$.We let $M_{p,2}^\circ \colon \mathbb{R}^{d\times d} \to \mathbb{R}^{d\times d}$ be the map, which given a matrix $x \in \mathbb{R}^{d\times d}$ where $x = U\Sigma V$ is the singular value decomposition of x, sets*

$$M_{p,2}^\circ(x) = U\Sigma^{p/2}V. \tag{4.6}$$

Similarly to the case of ℓ_p, Holder estimates on the map $M_{p,2}^\circ$ give a relation between Rayleigh quotients in S_p and S_2, and since $S_2 = \ell_2$, we obtain a Rayleigh quotient relation with ℓ_1. The argument relies on the following estimates.

Lemma 4.1.6 (Holder estimates for $M_{p,2}^{\circ}$ [121])**.** *For any $p \in [1, \infty)$, there exists a constant $c_p \in \mathbb{R}_{\geq 0}$ such that for any $d \in \mathbb{N}$, $M_{p,2}^{\circ} \colon S(S_p) \to S(S_2)$ is $\min\{p/2, 1\}$-Holder with constant c_p.*

We may use Lemma 4.1.6, Theorem 45, and Theorem 46 to relate Rayleigh quotients in S_p to those in S_2 (and, hence ℓ_2 and ℓ_1).

Lemma 4.1.7. *Fix $d \in \mathbb{N}$ and $p \geq 1$ and $p \neq 2$. Let μ be any symmetric probability distribution supported on finitely many pairs of points in $\mathbb{R}^{d \times d}$. There exists two points $z_1, z_2 \in \mathbb{R}^{d \times d}$ and a function $f \colon \mathbb{R}^{d \times d} \to \mathbb{R}^{d \times d}$ such that*

$$f(x) = M_{2,1}\left(M_{p,2}^{\circ}(x - z_1) - z_2\right),$$

and

$$\mathsf{R}(f(\mu), \|\cdot\|_1) \lesssim p \cdot \mathsf{R}(\mu, \|\cdot\|_{S_p}^p)^{\min\{1/2, 1/p\}}.$$

Proof. Letting ρ be the marginal distribution of μ, we apply Theorem 46 with $X = S_p$, $Y = S_2$, and $\phi = M_{p,2}$ and $r = \max\{p/2, 1\} > \min\{p/2, 1\} = \alpha$ to find a point $z_1 \in \mathbb{R}^{d \times d}$ with

$$\mathop{\mathbf{E}}_{\boldsymbol{x} \sim \rho}\left[M_{p,2}^{\circ}(\boldsymbol{x} - z_1)\right] = 0,$$

where I used the fact that the radial extension of $M_{p,2}^{\circ}$ to $\mathbb{R}^{d \times d}$ applies (4.6). By applying Theorem 45, to the symmetric probability distribution $\tilde{\mu}$ given by $(\boldsymbol{x} - z_1, \boldsymbol{y} - z)$ with $(\boldsymbol{x}, \boldsymbol{y}) \sim \mu$, $s = 2$, $r = p/2$, and $\alpha = \min\{p/2, 1\}$, we have

$$\mathsf{R}(M_{p,2}^{\circ}(\tilde{\mu}), \|\cdot\|_{S_2}^2) \lesssim p^2 \cdot \mathsf{R}(\tilde{\mu}, \|\cdot\|_{S_p}^p)^{\min\{1, 2/p\}}.$$

Considering the symmetric probability distribution $\tilde{\tilde{\mu}}$ given by $(M_{p,2}^{\circ}(\boldsymbol{x}), M_{p,2}^{\circ}(\boldsymbol{y}))$ where $(\boldsymbol{x}, \boldsymbol{y}) \sim \tilde{\mu}$, we may apply Lemma 4.1.3 to obtain that there exists $z_2 \in \mathbb{R}^{d \times d}$ such that $f \colon \mathbb{R}^{d \times d} \to \mathbb{R}^{d \times d}$ given by $f(x) = M_{2,1}(x - z_2)$ satisfies

$$\mathsf{R}(f(\tilde{\tilde{\mu}}), \|\cdot\|_1) \leq 48 \cdot \mathsf{R}(\tilde{\tilde{\mu}}, \|\cdot\|_2^2)^{1/2}.$$

Combining both Rayleigh quotient relations, as well as noticing that Rayleigh quotients are invariant to translations, the function $F\colon \mathbb{R}^{d\times d} \to \mathbb{R}^{d\times d}$ given by $F(x) = M_{2,1}\left(M^{\circ}_{p,2}(x - z_1) - z_2\right)$ satisfies

$$\begin{aligned}\mathsf{R}(F(\mu), \|\cdot\|_1) = \mathsf{R}(f(\tilde{\tilde{\mu}}), \|\cdot\|_1) &\leq 48 \cdot \mathsf{R}(\tilde{\tilde{\mu}}, \|\cdot\|_2^2)^{1/2} = 48 \cdot \mathsf{R}(M^{\circ}_{p,2}(\tilde{\mu}), \|\cdot\|_{S_2}^2)^{1/2} \\ &\lesssim p \cdot \mathsf{R}(\tilde{\mu}, \|\cdot\|_{S_p}^p)^{\min\{1/2,1/p\}} = p \cdot \mathsf{R}(\mu, \|\cdot\|_{S_p}^p)^{\min\{1/2,1/p\}}.\end{aligned}$$

□

Corollary 4.1.8. *For any $d \in \mathbb{N}$ and any $p \in \mathbb{N}$, $\Xi(S_p^d, \varepsilon) \lesssim (p/\varepsilon)^{\max\{2/p,1\}}$.*

Proof. We apply Lemma 4.1.7 and Lemma 4.0.1. □

4.2 A Holder Homeomorphism Between Perturbations of Spheres

Below, the unit ball and unit sphere of a (complex[1]) normed space $X = (\mathbb{C}^d, \|\cdot\|_X)$ are denoted $B_X = \{x \in \mathbb{C}^d : \|x\|_X \leq 1\}$ and $S_X = \{x \in \mathbb{C}^d : \|x\|_X = 1\}$, respectively.

Theorem 47 (Existence of a Hölder homeomorphism between spheres of perturbed spaces)**.** *Let $X = (\mathbb{C}^d, \|\cdot\|_X)$ be a normed space and fix $\alpha, \beta, \gamma \in (0, \frac{1}{2}]$. Suppose that the inradius and outradius of B_X are $r > 0$ and $R > 0$, respectively, i.e., $rB_{\ell_2^d} \subset B_X \subset RB_{\ell_2^d}$. Then there are normed spaces $Y = (\mathbb{C}^d, \|\cdot\|_Y)$ and $Z = (\mathbb{C}^d, \|\cdot\|_Z)$, and a bijection $\phi\colon S_Y \to S_Z$, with the following properties.*

1. $r^{2\alpha+\beta(1-2\alpha)}B_Y \subset B_X \subset R^{2\alpha+\beta(1-2\alpha)}B_Y$.
2. $r^{\gamma(1-2\alpha)}B_{\ell_2^d} \subset B_Z \subset R^{\gamma(1-2\alpha)}B_{\ell_2^d}$.
3. $\|\phi(y_1) - \phi(y_2)\|_Z \lesssim \frac{1}{\sqrt{\beta\gamma}}\|y_1 - y_2\|_Y^{\alpha}$ *for all* $y_1, y_2 \in S_Y$.

[1]It is convenient and most natural to carry out the ensuing geometric and analytic considerations for normed spaces over the complex scalars $\mathbb{C}$, but all of their applications that we obtain here hold also for normed spaces over the real scalars $\mathbb{R}$ through a standard complexification procedure which is recalled in Section 4.3 below.

The parameters α, β, γ are chosen to be small, in which case the first two assertions of Theorem 47 mean that Y and Z are relatively small perturbations of X and ℓ_2^d, respectively. The last assertion of Theorem 47 state that ϕ is an α-Holder homeomorphism between the unit spheres of these perturbed spaces with constant $O(1/\sqrt{\beta\gamma})$. The tension is between the smallness of α, β, γ (thus, the extent to which the initial geometries of X and ℓ_2^d were deformed) and the quality of the continuity of ϕ; the parameters are eventually set to appropriately balance these competing features.

Theorem 47 is a finite-dimensional quantitative refinement in the spirit of [110] of the work of Daher [59] which is itself an extension of a landmark contribution of Odell and Schlumprecht [115] (in unpublished work, Kalton independently obtained the result of [59]; see [30, page 216] or the MathSciNet review of [59]). Our proof of Theorem 47 is an adaptation of the proof of the corresponding qualitative infinite-dimensional result that appears in [30, Chapter 9].

Theorem 48. *Let $0 < \varepsilon < 1$ and $X = (\mathbb{C}^d, \|\cdot\|_X)$ be a d-dimensional normed space. Then there exists a randomized data structure for c-ANN over X with the following guarantees:*

- *The approximation is $c \leq \exp\left(O\left(\sqrt{\log d} \cdot \max\left\{\sqrt{\log\log d}, \frac{\log(1/\varepsilon)}{\sqrt{\log\log d}}\right\}\right)\right)$;*
- *The query procedure takes $n^{\varepsilon} \cdot d^{O(1)}$* time*;*
- *The space used by the data structure is $n^{1+\varepsilon} \cdot d^{O(1)}$;*
- *The preprocessing time is $n^{O(1)} \cdot d^{O(d)}$.*

Both the preprocessing and query procedures access the norm through an oracle, which, given a vector $x \in \mathbb{C}^d$, computes $\|x\|_X$.

4.2.1 Algorithmic version of Theorem 47

For algorithmic applications, we would like to compute the mapping φ from Theorem 47 efficiently at any given input point in $\mathbb{C}^d$. The main ingredient in the construction of F is the notion of *complex interpolation* between normed spaces, which was introduced in [41]. For two d-dimensional normed spaces U and V, complex interpolation provides a one-parameter family of d-dimensional normed spaces $[U, V]_\theta$ indexed by $\theta \in [0, 1]$, such that $[U, V]_0 = U$, $[U, V]_1 = V$ and $[U, V]_\theta$ depends, in a certain sense, smoothly on θ. In particular, we need to

compute the norm of a vector in $[U, V]_\theta$ given suitable oracles for the norm computation in U and V. This is a non-trivial task since the norm in $[U, V]_\theta$ is defined as the minimum of a certain functional on an infinite-dimensional space of holomorphic functions. We show how to properly "discretize" this optimization problem using harmonic and complex analysis, and ultimately solve it using convex programming (more specifically, the "robust" ellipsoid method [94]). We expect that the resulting algorithmic version of complex interpolation will have further applications.

More specifically, for $x \in \mathbb{C}^d$ the interpolated norm $\|x\|_{[U,V]_\theta}$ is defined as follows. First, we consider the space $\mathcal{F}$ of functions $F\colon \overline{\mathfrak{S}} \to \mathbb{C}^d$, where $\overline{\mathfrak{S}} = \{z \in \mathbb{C} \mid 0 \le \mathrm{Re} z \le 1\}$ is a strip on the complex plane, such that:

- F is bounded and continuous;
- F is holomorphic on the interior of $\overline{\mathfrak{S}}$.

The norm $\|F\|_{\mathcal{F}}$ in the space $\mathcal{F}$ is defined as follows:

$$\|F\|_{\mathcal{F}} = \max\left\{\sup_{\mathrm{Re} z=0} \|F(z)\|_U, \sup_{\mathrm{Re} z=1} \|F(z)\|_V\right\}.$$

Finally, for $x \in \mathbb{C}^d$, we define:

$$\|x\|_{[U,V]_\theta} = \inf_{\substack{F \in \mathcal{F}: \\ F(\theta)=x}} \|F\|_{\mathcal{F}}. \tag{4.7}$$

A priori, it is not clear how to solve (4.7), since the space $\mathcal{F}$ is infinite-dimensional. However, we are able to show that one can search for an approximately optimal $F \in \mathcal{F}$ of the following form:

$$F(z) = e^{\varepsilon z^2} \cdot \sum_{|k| \le M} v_k e^{\frac{kz}{L}},$$

for a fixed $\varepsilon > 0$, M and L, and variables are $v_k \in \mathbb{C}^d$. This turns (4.7) into a finite-dimensional convex program, which we might hope to solve. However, in order for the optimization procedure to be efficient, one needs to upper bound M and the magnitudes of v_k. This can be done by taking an approximately optimal (in terms of (4.7)) function F, smoothing it by convolving with an appropriate Gaussian, and finally considering its Fourier

expansion, whose convergence we can control using the classical Fejér's theorem [86]. To bound the magnitudes of v_k, we need a statement similar to the Paley–Wiener theorem [86]. Finally, to address the issue that the norm in $\mathcal{F}$ is defined as a supremum over the infinite set (the boundary of the strip $\overline{\mathfrak{S}}$), we show how to discretize and truncate the boundary so that the maximum over the discretization is not too far from the true supremum. This is again possible due to the bounds on the magnitudes of ε, v_k and M we are able to show.

4.3 Preliminaries

Given two quantities $a, b > 0$, the notation $a \lesssim b$ and $b \gtrsim a$ means $a \leq Cb$ for some universal constant $C > 0$. In this work we use some tools from complex analysis. Denote $\mathfrak{S} = \{z \in \mathbb{C} \mid 0 < \mathrm{Re} z < 1\} \subset \mathbb{C}$ the unit open strip on the complex plane, let $\partial\mathfrak{S} = \{z \in \mathbb{C} \mid \mathrm{Re} z \in \{0, 1\}\}$ be its boundary, and, finally, let $\overline{\mathfrak{S}} = \mathfrak{S} \cup \partial\mathfrak{S}$ be the corresponding closed strip. Given a normed space X defined over a (real or complex) vector space V, the subset $B_X \subset V$ is the unit ball of X, i.e., $B_X = \{x \in V : \|x\|_X \leq 1\}$. For a measure space (Ω, μ) and a Banach space X we denote $L_p(\Omega, \mu, X)$ the Banach space of measurable functions $f \colon \Omega \to X$ such that

$$\int_\Omega \|f\|_X^p \, d\mu < +\infty;$$

we define the norm to be:

$$\|f\|_{L_p(\Omega,\mu,X)}^p = \int_\Omega \|f\|_X^p \, d\mu.$$

Sometimes, we omit Ω in the notation if it is clear from the context (or unimportant).

Computational model for general normed spaces

Throughout this work, we deal with computational aspects of ANN defined over general normed spaces, in particular $X = (\mathbb{R}^d, \|\cdot\|_X)$. We work with the standard computational models for convex sets over $\mathbb{R}^d$. In particular, we may assume the following about X:

- There exists an oracle which, given $x \in \mathbb{R}^d$, computes $\|x\|_X$;

- The unit ball of X satisfies $B_X \subset B_2 \subset \mathsf{d}B_X$ for $\mathsf{d} = \mathrm{poly}(d)$.

The second assumption is essentially without loss of generality. Indeed, if one assumes B_X is contained within the unit Euclidean ball and contains a small Euclidean ball of radius $r = \exp(-\mathrm{poly}(d))$, then, by the reductions of [72], we may design a separation oracle for B_X, and as noted in Section 1.1 of [84], this means we can transform B_X to be in a position such that the second assumptions holds.

The Poisson kernel for the strip $\mathfrak{S}$

For $w \in \mathfrak{S}$ and $z \in \partial\mathfrak{S}$, the Poisson kernel $P(w, z)$ for $\mathfrak{S}$ is defined as follows:

$$P(w,z) = \begin{cases} \frac{1}{2} \cdot \frac{\sin \pi u}{\cosh \pi(\tau - v) - \cos \pi u}, & w = u + iv \text{ and } z = i\tau, \\ \frac{1}{2} \cdot \frac{\sin \pi u}{\cosh \pi(\tau - v) + \cos \pi u}, & w = u + iv \text{ and } z = 1 + i\tau. \end{cases} \tag{4.8}$$

For every $w \in \mathfrak{S}$, and every $z \in \partial\mathfrak{S}$, one has $P(w, z) \geq 0$. In addition, for every $w \in \mathfrak{S}$,

$$\int_{\partial\mathfrak{S}} P(w,z)\, dz = 1,$$

which allows us to denote μ_w the measure on $\partial\mathfrak{S}$ with the density $P(w, \cdot)$. We refer the reader to [133] for further properties of the kernel $P(\cdot, \cdot)$.

For $\theta_1, \theta_2 \in (0, 1)$, we let

$$\Lambda(\theta_1, \theta_2) \stackrel{\text{def}}{=} \sqrt{\left(\frac{1}{\theta_1} + \frac{1}{1-\theta_1}\right)\left(\frac{1}{\theta_2} + \frac{1}{1-\theta_2}\right)}, \tag{4.9}$$

Claim 4.3.1. *For any $z \in \partial\mathfrak{S}$ and $\theta_1, \theta_2 \in (0, 1)$,*

$$\frac{P(\theta_1, z)}{P(\theta_2, z)} \lesssim \Lambda(\theta_1, \theta_2)^2.$$

Proof. First, consider the case $z = i\tau$ when $\tau \in \mathbb{R}$. Then by the first case of (4.8),

$$\begin{aligned}\frac{P(\theta_1, i\tau)}{P(\theta_2, i\tau)} &= \frac{\sin(\pi\theta_1)}{\cosh(\pi\tau) - \cos(\pi\theta_1)} \cdot \frac{\cosh(\pi\tau) - \cos(\pi\theta_2)}{\sin(\pi\theta_2)} \\ &\lesssim \theta_1 \left(\frac{1}{\theta_2} + \frac{1}{1-\theta_2}\right) \cdot \frac{\cosh(\pi\tau) - \cos(\pi\theta_2)}{\cosh(\pi\tau) - \cos(\pi\theta_1)} \\ &\lesssim \theta_1 \left(\frac{1}{\theta_2} + \frac{1}{1-\theta_2}\right) \cdot \left(1 + \frac{1}{\theta_1^2}\right),\end{aligned}$$

where in the first line, we use the fact that $\sin(\pi\theta) \approx \theta$ when $\theta \approx 0$ and $\sin(\pi\theta) \approx 1 - \theta$ when $\theta \approx 1$, and in the second line, we use the fact that $\cosh(\pi\tau) \geq 1$, and the fact that $1 - \cos(\pi\theta) \gtrsim \frac{1}{\theta^2}$. By the second case of (4.8), when $z = 1 + i\tau$ for $\tau \in \mathbb{R}$,

$$\begin{aligned}\frac{P(\theta_1, 1 + i\tau)}{P(\theta_2, 1 + i\tau)} &= \frac{\sin(\pi\theta_1)}{\cosh(\pi\tau) + \cos(\pi\theta_1)} \cdot \frac{\cosh(\pi\tau) + \cosh(\pi\theta_2)}{\sin(\pi\theta_2)} \\ &\lesssim (1 - \theta_1) \left(\frac{1}{\theta_2} + \frac{1}{1-\theta_2}\right) \cdot \left(1 + \frac{1}{(1-\theta_1)^2}\right),\end{aligned}$$

where we now use the fact that $1 + \cos(\pi\theta) \gtrsim \frac{1}{(1-\theta)^2}$. □

Harmonic and holomorphic functions on $\mathfrak{S}$

Lemma 4.3.2 ([133])**.** *Let $f \colon \overline{\mathfrak{S}} \to \mathbb{R}$ be a continuous function which is harmonic (as a function of two real variables) in $\mathfrak{S}$. Moreover, suppose that the integral*

$$\int_{\partial\mathfrak{S}} |f(z)| \, d\mu_w(z)$$

if finite for $j = 0, 1$ and some $w \in \mathfrak{S}$. Then for every $w \in \mathfrak{S}$, one has:

$$f(w) = \int_{\partial\mathfrak{S}} f(z) \, d\mu_w(z).$$

Corollary 4.3.3. *Let $f \colon \overline{\mathfrak{S}} \to \mathbb{C}^d$ be a continuous function which is holomorphic in $\mathfrak{S}$. Moreover, suppose that*

$$\int_{\partial\mathfrak{S}} \big\| f(z) \big\| \, d\mu_w(z) < \infty$$

for some $w \in \mathfrak{S}$. Then for every $w \in \mathfrak{S}$, one has:

$$f(w) = \int_{\partial \mathfrak{S}} f(z)\, d\mu_w(z).$$

Proof. This follows from Lemma 4.3.2 and the fact that the real and the imaginary part of a holomorphic function are harmonic. □

Complexification

Let $X = (\mathbb{R}^d, \|\cdot\|_X)$ be a normed space over the vector space $\mathbb{R}^d$. The *complexification* of X, denoted by $X^{\mathbb{C}}$, is a normed space over $\mathbb{C}^d$ defined as follows. Elements of $X^{\mathbb{C}}$ are formal sums $u+iv$ for $u, v \in X$. Given $u+iv, w+iy \in X^{\mathbb{C}}$, addition of $(u+iv)+(w+iy)$ is given by $(u+iv)+(w+iy) = (u+w)+i(v+y)$. Given $u+iv \in X^{\mathbb{C}}$ and $\alpha = p+iq \in \mathbb{C}$, scalar multiplication $\alpha(u+iv)$ is given by, $\alpha(u+iv) = (pu-qv)+i(pv+qu)$. Finally, the norm on $X^{\mathbb{C}}$ is defined as:

$$\|u+iv\|^2_{X^{\mathbb{C}}} = \frac{1}{\pi}\int_0^{2\pi} \|u\cos\varphi - v\sin\varphi\|^2_X \, d\varphi. \tag{4.10}$$

The space $X^{\mathbb{C}} = (\mathbb{C}^d, \|\cdot\|_{X^{\mathbb{C}}})$ defined above is indeed a complex normed space and, moreover, X embeds into $X^{\mathbb{C}}$ isometrically via the map $u \mapsto u + i\cdot 0$. In addition, consider the d-dimensional space $(\ell_2^d)^{\mathbb{C}} = (\mathbb{C}^d, \|\cdot\|_{(\ell_2^d)^{\mathbb{C}}})$ given by the complexification of the space $\ell_2^d = (\mathbb{R}^d, \|\cdot\|_2)$. We note that:

$$\|u+iv\|^2_{(\ell_2^d)^{\mathbb{C}}} = \frac{1}{\pi}\int_0^{2\pi} \|u\cos\phi - v\sin\phi\|_2^2 d\phi = \frac{1}{\pi}\int_0^{2\pi}\sum_{i=1}^{d}(u_i\cos\phi - v_i\sin\phi)^2 d\phi = \|u\|_2^2 + \|v\|_2^2,$$

which implies that $(\ell_2^d)^{\mathbb{C}}$ is isometric to $\ell_2^{2d} = (\mathbb{R}^{2d}, \|\cdot\|_2)$, where we consider splitting the real and imaginary parts of each coordinate, and interpreting these as real numbers. We note that by a simple calculation, if $W_0 = (\mathbb{R}^d, \|\cdot\|_{W_0})$ and $W_1 = (\mathbb{R}^d, \|\cdot\|_{W_1})$ are real normed spaces with $B_{W_0} \subset B_{W_1} \subset \mathsf{d}\cdot B_{W_0}$, then $B_{W_0{}^{\mathbb{C}}} \subset B_{W_1{}^{\mathbb{C}}} \subset \mathsf{d}\cdot B_{W_0{}^{\mathbb{C}}}$.

Complex interpolation between normed spaces

Let $W_0 = (\mathbb{C}^d, \|\cdot\|_{W_0})$ and $W_1 = (\mathbb{C}^d, \|\cdot\|_{W_1})$ be two d-dimensional complex normed spaces. We will now define a family of spaces $[W_0, W_1]_\theta = (\mathbb{C}^d, \|\cdot\|_{[W_0,W_1]_\theta})$ for $0 \leq \theta \leq 1$ that, in a sense we will make precise later, interpolate between W_0 and W_1. This definition appeared for the first time in [41], see also the book [31]. Let us first define an auxiliary (infinite-dimensional) normed space $\mathcal{F}$ as the space of bounded continuous functions $f\colon \overline{\mathfrak{S}} \to \mathbb{C}^d$, which are holomorphic in $\mathfrak{S}$. The norm on $\mathcal{F}$ is defined as follows:

$$\|f\|_{\mathcal{F}} = \max\left\{\sup_{\mathrm{Re}(z)=0} \|f(z)\|_{W_0}, \sup_{\mathrm{Re}(z)=1} \|f(z)\|_{W_1}\right\}.$$

Now we can define the interpolation norm $\|\cdot\|_{[W_0,W_1]_\theta}$ on $\mathbb{C}^d$ as follows:

$$\|x\|_{[W_0,W_1]_\theta} = \inf_{\substack{f\in\mathcal{F}:\\ f(\theta)=x}} \|f\|_{\mathcal{F}}. \tag{4.11}$$

The fact that $\|x\|_{[W_0,W_1]_\theta}$ is a norm is straightforward to check modulo the property "$\|x\|_{[W_0,W_1]_\theta} = 0$ implies $x = 0$". The latter is a consequence of the Hadamard three-lines theorem [129].

Fact 4.3.4. *For every $\theta \in [0,1]$, $[W_0, W_1]_\theta = [W_1, W_0]_{1-\theta}$.*

Fact 4.3.5 (Reiteration theorem)**.** *For every $0 \leq \theta_1 \leq \theta_2 \leq 1$ and $0 \leq \theta_3 \leq 1$, one has:*

$$\Big[[W_0, W_1]_{\theta_1}, [W_0, W_1]_{\theta_2}\Big]_{\theta_3} = [W_0, W_1]_{(1-\theta_3)\theta_1+\theta_3\theta_2}.$$

Below is arguably the most useful statement about complex interpolation.

Fact 4.3.6 ([41, 31])**.** *Let $W_0 = (\mathbb{C}^d, \|\cdot\|_{W_0})$ and $W_1 = (\mathbb{C}^d, \|\cdot\|_{W_1})$ be d-dimensional complex normed spaces, and let $U_0 = (\mathbb{C}^{d'}, \|\cdot\|_{U_0})$ and $U_1 = (\mathbb{C}^{d'}, \|\cdot\|_{U_1})$ be a couple of d'-dimensional ones. Suppose that $T\colon \mathbb{C}^d \to \mathbb{C}^{d'}$ be a linear map. Then, for every $0 \leq \theta \leq 1$, one has:*

$$\|T\|_{[W_0,W_1]_\theta \to [U_0,U_1]_\theta} \leq \|T\|^{1-\theta}_{W_0\to U_0} \cdot \|T\|^{\theta}_{W_1\to U_1}.$$

Corollary 4.3.7. *Let $W_0 = (\mathbb{C}^d, \|\cdot\|_{W_0})$ and $W_1 = (\mathbb{C}^d, \|\cdot\|_{W_1})$ be complex normed spaces such that for some $\mathsf{d}_1, \mathsf{d}_2 \geq 1$ and every $x \in \mathbb{C}^d$, the following holds:*

$$\frac{1}{\mathsf{d}_1} \cdot \|x\|_{W_1} \leq \|x\|_{W_0} \leq \mathsf{d}_2 \cdot \|x\|_{W_1}. \tag{4.12}$$

Then, for every $0 \leq \theta \leq 1$ and every $x \in \mathbb{C}^d$, one has:

$$\frac{1}{\mathsf{d}_1^{\theta}} \cdot \|x\|_{[W_0,W_1]_\theta} \leq \|x\|_{W_0} \leq \mathsf{d}_2^{\theta} \cdot \|x\|_{[W_0,W_1]_\theta} \qquad \text{and} \qquad \frac{1}{\mathsf{d}_1^{1-\theta}} \cdot \|x\|_{W_1} \leq \|x\|_{[W_0,W_1]_\theta} \leq \mathsf{d}_2^{1-\theta} \cdot \|x\|_{W_1}.$$

Proof. This follows from Fact 4.3.6 applied to the identity map. □

Fact 4.3.8 ([41, 31])**.** *Let $W_0 = (\mathbb{C}^d, \|\cdot\|_{W_0})$ and $W_1 = (\mathbb{C}^d, \|\cdot\|_{W_1})$ be complex normed spaces, and let ${W_0}^* = (\mathbb{C}^d, \|\cdot\|_{{W_0}^*})$ and ${W_1}^* = (\mathbb{C}^d, \|\cdot\|_{{W_1}^*})$ be the dual spaces, respectively. For any $\theta \in [0,1]$, the dual space to $[W_0, W_1]_\theta$, given by $[W_0, W_1]_\theta^* = (\mathbb{C}^d, \|\cdot\|_{[W_0,W_1]_\theta^*})$ is isometric to the space $[{W_0}^*, {W_1}^*]_\theta$.*

Uniform convexity

Let $W = (\mathbb{C}^d, \|\cdot\|_W)$ be a complex normed space. We give necessary definitions related to the notion of uniform convexity. For a thorough overview, see [26].

Definition 36. *For $2 \leq p \leq \infty$, the space W has modulus of convexity of power type p iff there exists $K \geq 1$ such that for every $x, y \in W$:*

$$\left(\|x\|_W^p + \frac{1}{K^p}\|y\|_W^p\right)^{1/p} \leq \left(\frac{\|x+y\|_W^p + \|x-y\|_W^p}{2}\right)^{1/p}.$$

Definition 37. *The infimum of such K is called the p-convexity constant of W and is denoted by $K_p(W)$.*

Claim 4.3.9. *One always has $K_\infty(W) = 1$, and for a Hilbert space, one has: $K_2(\ell_2^d) = 1$.*

Claim 4.3.10. *One has $K_p(W_0 \oplus_2 W_1) \lesssim \max\{K_p(W), K_p(W_1)\}$ for every $p \geq 2$.*

Proof. The claim follows from $W_0 \oplus_2 W_1$ being isomorphic to $W_0 \oplus_p W_1$ and the fact that $K_p(W_0 \oplus_p W_1) \leq \max\{K_p(W), K_p(W_1)\}$. □

Lemma 4.3.11 ([112]). *One has $K_p(L_2(\mu, W)) \lesssim K_p(W)$ for every $p \geq 2$.*

The following lemma shows how the p-convexity constant interacts with complex interpolation.

Lemma 4.3.12 ([110]). *For every $2 \leq p_1, p_2 \leq \infty$ and every $0 \leq \theta \leq 1$, one has:*

$$K_{\frac{p_1 p_2}{\theta p_1 + (1-\theta)p_2}}\Big([W_0, W_1]_\theta\Big) \leq K_{p_1}(W_0)^{1-\theta} K_{p_2}(W_1)^\theta.$$

The space $\mathcal{F}_2(\theta)$

Now we define another space related to $\mathcal{F}$. This definition appears in [41], see also [30]. First, for $0 < \theta < 1$, let us consider the normed space $\mathcal{G}(\theta)$ of continuous functions $f\colon \overline{\mathfrak{S}} \to \mathbb{C}^d$, which are holomorphic in $\mathfrak{S}$, and

$$\int\limits_{\partial\mathfrak{S}} \Big\|f(z)\Big\|^2 d\mu_\theta(z) < \infty.$$

The norm $\|f\|_{\mathcal{G}(\theta)}$ is defined as follows:

$$\|f\|^2_{\mathcal{G}(\theta)} = \int\limits_{\mathrm{Re}(z)=0} \Big\|f(z)\Big\|^2_{W_0} d\mu_\theta(z) + \int\limits_{\mathrm{Re}(z)=1} \Big\|f(z)\Big\|^2_{W_1} d\mu_\theta(z). \tag{4.13}$$

Clearly, $\mathcal{F} \subset \mathcal{G}(\theta)$. One may naturally view $\mathcal{G}(\theta)$ as a (not closed) subspace of $L_2(\{z \mid \mathrm{Re}(z) = 0\}, \mu_\theta, W_0) \oplus_2 L_2(\{z \mid \mathrm{Re}(z) = 1\}, \mu_\theta, W_1)$. Now we can define the space $\mathcal{F}_2(\theta)$ as the closure of $\mathcal{G}(\theta)$ (in particular, $\mathcal{G}(\theta)$ is dense in $\mathcal{F}_2(\theta)$). An element of $\mathcal{F}_2(\theta)$ can be identified with a function $f\colon \overline{\mathfrak{S}} \to \mathbb{C}^d$ defined *almost everywhere* on $\partial\mathfrak{S}$ and defined everywhere on $\mathfrak{S}$ such that:

- f restricted on $\{z \mid \mathrm{Re}(z) = 0\}$ belongs to $L_2(\{z \mid \mathrm{Re}(z) = 0\}, \mu_\theta, W_0)$;
- f restricted on $\{z \mid \mathrm{Re}(z) = 1\}$ belongs to $L_2(\{z \mid \mathrm{Re}(z) = 1\}, \mu_\theta, W_1)$;
- f is holomorphic in $\mathfrak{S}$;

In this representation, the norm is defined similar to (4.13):

$$\|f\|^2_{\mathcal{F}(\theta)} = \int\limits_{\mathrm{Re}(z)=0} \Big\|f(z)\Big\|^2_{W_0} d\mu_\theta(z) + \int\limits_{\mathrm{Re}(z)=1} \Big\|f(z)\Big\|^2_{W_1} d\mu_\theta(z).$$

Fact 4.3.13. *For every $f \in \mathcal{F}_2(\theta)$ and $w \in \mathfrak{S}$, one has:*

$$f(w) = \int_{\partial\mathfrak{S}} f(z)\, d\mu_w(z).$$

Proof. This identity is true for $\mathcal{G}(\theta)$ by Corollary 4.3.3. Hence it holds for $\mathcal{F}_2(\theta)$, since every element of $\mathcal{F}_2(\theta)$ is a limit of a sequence of elements of $\mathcal{G}(\theta)$, which converges pointwise in $\mathfrak{S}$ and in L_2 in $\partial\mathfrak{S}$. □

The following gives an alternative definition of an interpolated norm, which should be compared with the original definition (4.11).

Fact 4.3.14 ([30])**.** *For every $x \in \mathbb{C}^d$, one has:*

$$\|x\|_{[W_0,W_1]_\theta} = \inf_{\substack{f\in\mathcal{F}_2(\theta):\\ f(\theta)=x}} \|f\|_{\mathcal{F}_2(\theta)}. \tag{4.14}$$

Claim 4.3.15. *For every $p \geq 2$, one has:*

$$K_p(\mathcal{F}_2(\theta)) \lesssim \max\{K_p(W_0), K_p(W_1)\}.$$

Proof. One has:

$$\begin{aligned} K_p(\mathcal{F}_2(\theta)) &\leq K_p\Big(L_2(\{z \mid \mathrm{Re}(z)=0\}, \mu_\theta, W_0) \oplus_2 L_2(\{z \mid \mathrm{Re}(z)=1\}, \mu_\theta, W_1)\Big) \\ &\lesssim \max\Big\{K_p\Big(L_2(\{z \mid \mathrm{Re}(z)=0\}, \mu_\theta, W_0)\Big), K_p\Big(L_2(\{z \mid \mathrm{Re}(z)=1\}, \mu_\theta, W_1)\Big)\Big\} \\ &\lesssim \max\{K_p(W_0), K_p(W_1)\}, \end{aligned}$$

where the first step is due to $\mathcal{F}_2(\theta)$ being a subspace of $L_2(\{z \mid \mathrm{Re}(z)=0\}, \mu_\theta, W_0) \oplus_2 L_2(\{z \mid \mathrm{Re}(z)=1\}, \mu_\theta, W_1)$, the second step is due to Claim 4.3.10, and the third step is

due to Lemma 4.3.11. □

Lemma 4.3.16. *For $0 < \theta_1, \theta_2 < 1$, the spaces $\mathcal{F}_2(\theta_1)$ and $\mathcal{F}_2(\theta_2)$ are isomorphic via the identity map. More specifically, for every $f \in \mathcal{F}_2(\theta_1)$ one has:*

$$\|f\|_{\mathcal{F}_2(\theta_2)} \leq \Lambda(\theta_1, \theta_2) \cdot \|f\|_{\mathcal{F}_2(\theta_1)};$$

and, similarly, for every $f \in \mathcal{F}_2(\theta_2)$, one has:

$$\|f\|_{\mathcal{F}_2(\theta_1)} \leq \Lambda(\theta_1, \theta_2) \cdot \|f\|_{\mathcal{F}_2(\theta_2)}.$$

Proof. This easily follows from the definition of $\mathcal{F}_2(\theta)$ and Claim 4.3.1. □

4.4 Hölder homeomorphisms: an existential argument

In this section we show the proof of Theorem 47 making the exposition of the result from [59] in [30] quantitative. We make the construction of the map algorithmic in Section 4.5. Let $X = (\mathbb{C}^d, \|\cdot\|_X)$ be a normed space of interest. For a real normed space, one can consider its complexification, which contains the real version isometrically.

Let us first assume that $K_p(X) < \infty$ for some $2 \leq p < \infty$. We start with taking a closer look at Fact 4.3.14. Suppose that we interpolate between X and ℓ_2^d and moreover for some $0 < r < R$ one has:

$$rB_{\ell_2^d} \subseteq B_X \subseteq RB_{\ell_2^d}.$$

Let $\mathcal{F}_2(\theta)$ be defined with respect to X and ℓ_2^d.

Fact 4.4.1 ([30])**.** *For every $x \in \mathbb{C}^d$, in the optimization problem*

$$\inf_{\substack{F \in \mathcal{F}_2(\theta): \\ F(\theta) = x}} \|F\|_{\mathcal{F}_2(\theta)}$$

the minimum is attained on an element of $\mathcal{F}_2(\theta)$. Moreover, the minimizer is unique, and we denote it by $F^*_{\theta x} \in \mathcal{F}_2(\theta)$.

Below statement shows that minimizers $F^*_{\theta x}$ have very special structure.

Fact 4.4.2 ([30]). *Fix $x \in \mathbb{C}^d$ and $0 < \theta < 1$ and consider $F^*_{\theta x} \in \mathcal{F}_2(\theta)$. Then,*

- *For $z \in \mathbb{C}$ such that $\mathrm{Re}z = 0$, $\|F^*_{\theta x}(z)\|_X = \|x\|_{[X,\ell_2^d]_\theta}$ almost everywhere;*
- *For $z \in \mathbb{C}$ such that $\mathrm{Re}z = 1$, $\|F^*_{\theta x}(z)\|_{\ell_2^d} = \|x\|_{[X,\ell_2^d]_\theta}$ almost everywhere;*
- *For every $0 < \widetilde{\theta} < 1$, $\|F^*_{\theta x}(\widetilde{\theta})\|_{[X,\ell_2^d]_{\widetilde{\theta}}} = \|x\|_{[X,\ell_2^d]_\theta}$.*

Below lemma is the core of the overall argument.

Lemma 4.4.3 (a quantitative version of a statement from [30]). *For every $0 < \theta < 1$ and every $x_1, x_2 \in S_{[X,\ell_2^d]_\theta}$, one has:*

$$\left\|F^*_{\theta x_1} - F^*_{\theta x_2}\right\|_{\mathcal{F}_2(\theta)} \lesssim K_p(X) \cdot \|x_1 - x_2\|^{1/p}_{[X,\ell_2^d]_\theta}.$$

Proof. By Claim 4.3.15, one has:

$$K_p(\mathcal{F}_2(\theta)) \lesssim \max\{K_p(X), K_p(\ell_2^d)\} \lesssim K_p(X),$$

where the second step follows from $K_p(\ell_2^d) \lesssim 1$. Second, suppose that for $x_1, x_2 \in S_{[X,\ell_2^d]_\theta}$, one has $\|x_1 - x_2\|_{[X,\ell_2^d]_\theta} = \varepsilon > 0$. Then,

$$\left\|F^*_{\theta x_1} + F^*_{\theta x_2}\right\|_{\mathcal{F}_2(\theta)} \geq \|x_1 + x_2\|_{[X,\ell_2^d]_\theta} \geq 2 - \varepsilon, \tag{4.15}$$

where the first step follows from Fact 4.3.14, and the second step follows from x_1 and x_2 being unit and the triangle inequality. Now by the definition of $K_p(\mathcal{F}_2(\theta))$ (Definition 36) and the fact that the minimizers are unit, we have:

$$\left\|F^*_{\theta x_1} + F^*_{\theta x_2}\right\|^p_{\mathcal{F}_2(\theta)} + \frac{\left\|F^*_{\theta x_1} - F^*_{\theta x_2}\right\|^p_{\mathcal{F}_2(\theta)}}{K_p(\mathcal{F}_2(\theta))^p} \leq \frac{\|2F^*_{\theta x_1}\|^p_{\mathcal{F}_2(\theta)} + \|2F^*_{\theta x_2}\|^p_{\mathcal{F}_2(\theta)}}{2} = 2^p. \tag{4.16}$$

Combining (4.15) and (4.16), we get:

$$\left\|F^*_{\theta x_1} - F^*_{\theta x_2}\right\|^p_{\mathcal{F}_2(\theta)} \leq K_p(\mathcal{F}_2(\theta))^p \cdot (2^p - (2-\varepsilon)^p) \leq p2^{p-1} \cdot K_p(\mathcal{F}_2(\theta))^p \cdot \varepsilon.$$

Finally, we get:

$$\left\|F^*_{\theta x_1} - F^*_{\theta x_2}\right\|_{\mathcal{F}_2(\theta)} \lesssim K_p(\mathcal{F}_2(\theta)) \cdot \varepsilon^{1/p} \lesssim K_p(X) \cdot \varepsilon^{1/p} = K_p(X) \cdot \|x_1 - x_2\|^{1/p}_{[X,\ell_2^d]_\theta}$$

as desired. □

Fix $0 < \theta_1, \theta_2 < 1$. Define the map $U_{\theta_1\theta_2} \colon S_{[X,\ell_2^d]_{\theta_1}} \to S_{[X,\ell_2^d]_{\theta_2}}$ as follows:

$$x \mapsto F^*_{\theta_1 x}(\theta_2).$$

The map is well-defined, since by Fact 4.4.2, for every x with $\|x\|_{[X,\ell_2^d]_{\theta_1}}$, one has $\|F^*_{\theta_1,x}(\theta_2)\|_{[X,\ell_2^d]_{\theta_2}} = 1$. One also has: $U^{-1}_{\theta_1,\theta_2} = U_{\theta_2,\theta_1}$, since, again by Fact 4.4.2, for every $x \in \mathbb{C}^d$, one has:

$$F^*_{\theta_2 F^*_{\theta_1 x}(\theta_2)} = F^*_{\theta_1 x}.$$

In particular, U_{θ_1,θ_2} is a bijection between the unit spheres of $[X,\ell_2^d]_{\theta_1}$ and $[X,\ell_2^d]_{\theta_2}$.

Lemma 4.4.4 (a quantitative version of the statement from [30])**.** *For $x_1, x_2 \in S_{[X,\ell_2^d]_{\theta_1}}$, one has:*

$$\|U_{\theta_1\theta_2}(x_1) - U_{\theta_1\theta_2}(x_2)\|_{[X,\ell_2^d]_{\theta_2}} \lesssim \Lambda(\theta_1,\theta_2) \cdot K_p(X) \cdot \|x_1 - x_2\|^{1/p}_{[X,\ell_2^d]_{\theta_1}}.$$

Proof. One has:

$$\begin{aligned}
\|U_{\theta_1\theta_2}(x_1) - U_{\theta_1\theta_2}(x_2)\|_{[X,\ell_2^d]_{\theta_2}} &= \|F^*_{\theta_1 x_1}(\theta_2) - F^*_{\theta_1 x_2}(\theta_2)\|_{[X,\ell_2^d]_{\theta_2}} \\
&\leq \|F^*_{\theta_1 x_1} - F^*_{\theta_1 x_2}\|_{\mathcal{F}_2(\theta_2)} \\
&\leq \Lambda(\theta_1,\theta_2) \cdot \|F^*_{\theta_1 x_1} - F^*_{\theta_1 x_2}\|_{\mathcal{F}_2(\theta_1)} \\
&\lesssim \Lambda(\theta_1,\theta_2) \cdot K_p(X) \cdot \|x_1 - x_2\|^{1/p}_{[X,\ell_2^d]_{\theta_1}},
\end{aligned}$$

where the first step is by the definition of $U_{\theta_1\theta_2}$, the second step is due to Fact 4.3.14, the third step is due to Lemma 4.3.16, and the last step is due to Lemma 4.4.3. □

The below theorem summarizes the above discussion.

Theorem 49. *Let $X = (\mathbb{C}^d, \|\cdot\|_X)$ be a complex normed space such that $K_p(X) < \infty$ for some $2 \le p < \infty$ and for some $0 < r < R$, one has: $rB_{\ell_2^d} \subseteq B_X \subseteq RB_{\ell_2^d}$. Fix $0 < \beta, \gamma \le 1/2$. Then there exist two spaces $Y = (\mathbb{C}^d, \|\cdot\|_Y)$ and $Z = (\mathbb{C}^d, \|\cdot\|_Z)$ and a bijection $\varphi\colon S_Y \to S_Z$ such that:*

- *One has: $r^\beta B_Y \subseteq B_X \subseteq R^\beta B_Y$;*
- *One has: $r^\gamma B_{\ell_2^d} \subseteq B_Z \subseteq R^\gamma B_{\ell_2^d}$;*
- *for every $y_1, y_2 \in S_Y$, one has: $\|\varphi(y_1) - \varphi(y_2)\|_Z \lesssim \frac{K_p(X)}{\sqrt{\beta\gamma}} \cdot \|y_1 - y_2\|_Y^{1/p}$;*
- *for every $z_1, z_2 \in S_Z$, one has: $\|\varphi^{-1}(z_1) - \varphi^{-1}(z_2)\|_Y \lesssim \frac{K_p(X)}{\sqrt{\beta\gamma}} \cdot \|z_1 - z_2\|_Z^{1/p}$.*

Proof. We set Y and Z to be $[X, \ell_2^d]_\beta$ and $[X, \ell_2^d]_{1-\gamma}$, respectively. Finally, set φ to be $U_{\beta,1-\gamma}$. Then, the first two inequalities follow from Corollary 4.3.7. The third inequality follows from Lemma 4.4.4 combined with the estimate $\Lambda(\beta, \gamma) \lesssim \frac{1}{\sqrt{\beta\gamma}}$. The fourth inequality is shown similar to the third taking into account that $\varphi^{-1} = U_{1-\gamma,\beta}$. □

Now let us turn to the case when X is not necessarily p-convex.

Theorem 50 (Theorem 47, restated)**.** *Let $X = (\mathbb{C}^d, \|\cdot\|_X)$ be a complex normed space such that for some $0 < r < R$, one has: $rB_{\ell_2^d} \subseteq B_X \subseteq RB_{\ell_2^d}$. Fix $0 < \alpha, \beta, \gamma \le 1/2$. Then there exist two spaces $Y = (\mathbb{C}^d, \|\cdot\|_Y)$ and $Z = (\mathbb{C}^d, \|\cdot\|_Z)$ and a bijection $\varphi\colon S_Y \to S_Z$ such that:*

- *One has: $r^{2\alpha+\beta(1-2\alpha)} B_Y \subseteq B_X \subseteq R^{2\alpha+\beta(1-2\alpha)} B_Y$;*
- *One has: $r^{\gamma(1-2\alpha)} B_{\ell_2^d} \subseteq B_Z \subseteq R^{\gamma(1-2\alpha)} B_{\ell_2^d}$;*
- *for every $y_1, y_2 \in S_Y$, one has: $\|\varphi(y_1) - \varphi(y_2)\|_Z \lesssim \frac{1}{\sqrt{\beta\gamma}} \cdot \|y_1 - y_2\|_Y^\alpha$;*
- *for every $z_1, z_2 \in S_Z$, one has: $\|\varphi^{-1}(z_1) - \varphi^{-1}(z_2)\|_Y \lesssim \frac{1}{\sqrt{\beta\gamma}} \cdot \|z_1 - z_2\|_Z^\alpha$.*

Proof. Denote $A = [X, \ell_2^d]_{2\alpha}$. By Lemma 4.3.12, one has $K_{1/\alpha}(A) \le 1$. Let us now apply Theorem 49 to A, which yields two spaces $Y = [A, \ell_2^d]_\beta$ and $Z = [A, \ell_2^d]_\gamma$. By Fact 4.3.5, one has: $Y = [X, \ell_2^d]_{2\alpha+\beta(1-2\alpha)}$ and $Z = [X, \ell_2^d]_{2\alpha+(1-\gamma)(1-2\alpha)}$, which together with Corollary 4.3.7 yields the first two items. The third and fourth items follow from Theorem 49 applied to A. □

4.5 Computing approximate Hölder homeomorphisms

4.5.1 High-level overview

The goal of this section is to give polynomial time algorithms for computing various aspects of complex interpolation which completes the description of the approximate Hölder homeomorphism from Section **??**. The data structures for ANN over a real normed space $X = (\mathbb{R}^d, \|\cdot\|_X)$ will compute this map, so we will assume oracle access to computing $\|\cdot\|_X$. In particular, we will provide algorithms for the two tasks specified towards the end of Subsection **??**. We will consider a complex normed space $W = (\mathbb{C}^d, \|\cdot\|_W)$ and the Hilbert space $H = (\ell_2^d)^{\mathbb{C}}$, i.e., the complexification of ℓ_2^d, and assume

$$B_W \subset B_H \subset \mathsf{d} \cdot B_W, \tag{4.17}$$

(note that we may compute $\|x\|_H$ since it is isometrically isomorphic to ℓ_2^{2d} by separating the real and imaginary parts of each coordinate; see Subsection 4.3).

At a high level, our algorithm will express the algorithmic tasks from Subsection **??** as the optimums of convex programs, which we then solve using various tools from convex optimization. We first set up some notation. For a convex set $K \subset \mathbb{R}^m$, and a real number $\delta > 0$ we let $B(K, \delta) = \{y \in \mathbb{R}^m : x \in K \text{ and } \|x - y\|_2 \leq \delta\}$, and we abuse notation slightly by letting $B(y, \delta) = B(\{y\}, \delta)$ for $y \in \mathbb{R}^m$. Then, we let $B(K, -\delta) = \{y \in \mathbb{R}^m : B(y, \delta) \subset K\}$. We will frequently interpret convex sets $K \subset \mathbb{C}^m$ as convex sets of $\mathbb{R}^{2m}$, by separating the real and imaginary parts of the vectors in $\mathbb{C}^m$.

Definition 38 (Membership Oracle ($\texttt{MEM}(K)$) [94])**.** *For a convex set $K \subset \mathbb{R}^m$, given a vector $y \in \mathbb{R}^m$ and a real number $\delta > 0$, with probability $1 - \delta$, either assert that $y \in B(K, \delta)$ or assert $y \notin B(K, -\delta)$.*

The goal of this section is to solve the following algorithmic task, which we denote $\texttt{ApproxRep}(x, \theta, \varepsilon; W)$, where we will assume access to $\texttt{MEM}(B_W)$, thus, we will measure the complexity of the algorithm for $\texttt{ApproxRep}(x, \theta, \varepsilon; W)$ in the number of calls to $\texttt{MEM}(B_W)$, as well as the error parameter $\delta > 0$ in these calls. In Subsection 4.5.5, we show

how the subsequent algorithms are used in our applications to ANN.

Definition 39 (Algorithmic Task $\texttt{ApproxRep}(x, \theta, \varepsilon; W)$)**.** *For a complex normed space* $W = (\mathbb{C}^d, \|\cdot\|_W)$ *satisfying*

$$B_W \subset B_H \subset \mathsf{d} \cdot B_W,$$

we want to solve the following algorithmic task. Given access to $\texttt{MEM}(B_W)$*, a parameter* $\theta \in (0,1)$*, a vector* $x \in \mathbb{C}^d$*, and an approximation parameter* $\varepsilon > 0$*, output the representation of a function* $f : \overline{\mathfrak{S}} \to \mathbb{C}^d \in \mathcal{F}$ *(where* $\mathcal{F}$ *is defined with respect to the couple* (W, H)[2]*) such that:*

- $\|f(\theta) - x\|_{[W,H]_\theta} \leq \varepsilon \|x\|_{[W,H]_\theta}$*, and*
- $\|f\|_{\mathcal{F}} = \max\{\sup_{t\in\mathbb{R}} \|f(it)\|_W, \sup_{t\in\mathbb{R}} \|f(1+it)\|_H\} \leq (1+\varepsilon_1)\|x\|_{[W,H]_\theta}$.

The representation of the function should also have the property that we may compute a $(1 \pm \varepsilon)$*-multiplicative approximation to* $\|f\|_{\mathcal{F}}$ *in* $\mathrm{poly}(d/\varepsilon)$ *time, and for any* $\theta' \in (0,1)$ *we may compute* $f(\theta') \in \mathbb{C}^d$ *in* $\mathrm{poly}(d/\varepsilon)$ *time.*

In the following section, we will assume that ε is a small enough parameter, which will later be set to $\frac{1}{\mathrm{poly}(d)}$. Before proceeding to present an algorithm for $\texttt{ApproxRep}(x, \theta, \varepsilon; W)$, we give the following simple consequence of being able to solve $\texttt{ApproxRep}(x, \theta, \varepsilon; W)$, which solves the first algorithmic task from Subsection **??**.

Corollary 4.5.1. *For any* $x \in \mathbb{C}^d$*,* $\theta \in (0,1)$ *and* $\varepsilon > 0$*, we may obtain a* $(1 \pm \varepsilon)^2$*-multiplicative approximation to* $\|x\|_{[W,H]_\theta}$ *from a call to* $\texttt{ApproxRep}(x, \theta, \varepsilon; W)$.

Proof. Given $f \in \mathcal{F}$ as an output of $\texttt{ApproxRep}(x, \theta, \varepsilon; W)$, we note that

$$\|f\|_{\mathcal{F}} \geq \|f(\theta)\|_{[W,H]_\theta} \geq \|x\|_{[W,H]_\theta} - \|f(\theta) - x\|_{[W,H]_\theta} \geq (1-\varepsilon)\|x\|_{[W,H]_\theta},$$

where we first used the definition of interpolation, and then we utilized the triangle inequality, as well as the fact that $f(\theta)$ is close to x. The upper bound follows by definition of $\texttt{ApproxRep}(x, \theta, \varepsilon; W)$. Finally, we note that we may compute a $(1 \pm \varepsilon)$-approximation to $\|f\|_{\mathcal{F}}$ in $\mathrm{poly}(d/\varepsilon)$ time. □

[2] see Subsection 4.3 for a formal definition of $\mathcal{F}$

For simplicity, when working with a couple (W, H), where $W = (\mathbb{C}^d, \|\cdot\|_W)$ is a complex normed space and $H = (\ell_2^d)^{\mathbb{C}}$ is a Hilbert space satisfying (4.17) we denote $W_\theta = (\mathbb{C}^d, \|\cdot\|_\theta)$ as the complex normed space given by $W_\theta = [W, H]_\theta$ and $\|\cdot\|_\theta = \|\cdot\|_{[W,H]_\theta}$.

4.5.2 Discretization of $\mathcal{F}$

We will now give the discretization of $\mathcal{F}$ we optimize over. Specifically, we show a quantitative version of Lemma 4.2.2 from [31]. We will write the discretization of $\mathcal{F}$, where $\mathcal{F}$ is defined with respect to the complex normed spaces $W_0 = (\mathbb{C}^d, \|\cdot\|_{W_0})$ and $W_1 = (\mathbb{C}^d, \|\cdot\|_{W_1})$. In our applications of this discretization to computing $\texttt{ApproxRep}(x, \theta, \varepsilon; W)$, we will set $W_0 = W$ and $W_1 = H$. Assume that W_0 and W_1 are both close to the Hilbert space $H = (\ell_2^d)^{\mathbb{C}}$, i.e., every $x \in \mathbb{C}^d$ satisfies

$$\|x\|_{W_0} \le \|x\|_H \le \mathsf{d}\|x\|_{W_0} \qquad \text{and} \qquad \|x\|_{W_1} \le \|x\|_H \le \mathsf{d}\|x\|_{W_1}. \tag{4.18}$$

Recall from Subsection 4.3, the space $\mathcal{F}$, defined with respect to W_0 and W_1, is a space over bounded continuous functions $f\colon \overline{\mathfrak{S}} \to \mathbb{C}^d$ which are holomorphic in $\mathfrak{S}$. We will consider a $x \in \mathbb{C}^d$ such that $\|x\|_{[W_0,W_1]_\theta} = 1$.

Let us first introduce some notation. For $f \in \mathcal{F}$ and $\tau \in \mathbb{R}$, we denote:

$$B_{\mathcal{F}}(f, \tau) \stackrel{\text{def}}{=} \max\{\|f(i\tau)\|_{W_0}, \|f(1 + i\tau)\|_{W_1}\}, \qquad \text{and} \tag{4.19}$$

$$B_\infty(f, \tau) \stackrel{\text{def}}{=} \max_{0 \le u \le 1} \|f(u + i\tau)\|_{\ell_\infty}. \tag{4.20}$$

In addition, for each $f \in \mathcal{F}$, we may view $g_0\colon \mathbb{R} \to \mathbb{C}^d$ as $g_0(\tau) = f(i\tau)$ and $g_1(\tau) = f(1 + i\tau)$. Given these definitions, when the derivatives $\frac{dg_0}{d\tau}$ and $\frac{dg_1}{d\tau}$ exist, we denote:

$$D_\infty(f, \tau) \stackrel{\text{def}}{=} \max\left\{\left\|\frac{dg_0}{d\tau}(\tau)\right\|_{\ell_\infty}, \left\|\frac{dg_1}{d\tau}(\tau)\right\|_{\ell_\infty}\right\}.$$

The following lemma is a quantitative version of the classical Fejér's theorem [86].

Lemma 4.5.2. *Let $f\colon \mathbb{R} \to \mathbb{C}$ to be a differentiable $2\pi L$-periodic function. Consider its*

Fourier series:

$$f \mapsto \sum_{n\in\mathbb{Z}} a_n e^{\frac{inx}{L}},$$

where

$$a_n = \frac{1}{2\pi L} \cdot \int_{-\pi L}^{\pi L} f(x) e^{-\frac{inx}{L}}\, dx.$$

Then the Césaro partial sums

$$f_M(x) = \sum_{|n|\le M} \left(1 - \frac{n}{M+1}\right) a_n e^{\frac{inx}{L}}$$

satisfy $\|f_M - f\|_{\ell_\infty} \le \varepsilon$ *for* $0 < \varepsilon < 0.1$ *with*

$$M \lesssim \frac{\|f\|_{\ell_\infty} \cdot \|f'\|_{\ell_\infty}^2 \cdot L^2}{\varepsilon^3}.$$

Proof. We can assume, by rescaling, that $L = 1$. Then, one has:

$$f_M = f * F_M,$$

where

$$F_M = \frac{1}{M}\left(\frac{\sin\frac{Mx}{2}}{\sin\frac{x}{2}}\right)^2$$

is the Fejér's kernel. Thus, for every $\delta > 0$, one has:

$$\begin{aligned}
|f_M(x) - f(x)| &= \left|\frac{1}{2\pi}\int_{-\pi}^{\pi} \Big(f(x-t) - f(x)\Big) F_M(t)\, dt\right| \\
&\le \frac{1}{2\pi}\int_{-\pi}^{\pi} \Big|f(x-t) - f(x)\Big| F_M(t)\, dt \\
&= \frac{1}{2\pi}\int_{|t|\le\delta} \Big|f(x-t) - f(x)\Big| F_M(t)\, dt + \frac{1}{2\pi}\int_{\delta\le|t|\le\pi} \Big|f(x-t) - f(x)\Big| F_M(t)\, dt \\
&\lesssim \delta\|f'\|_{\ell_\infty} + \frac{\|f\|_{\ell_\infty}}{\delta^2 M},
\end{aligned}$$

where the first step follows from $\frac{1}{2\pi}\int_{-\pi}^{\pi} F_M(t)\, dt = 1$, the second step follows from $F_M(t) \ge$

0, and the fourth step follows from $\frac{1}{2\pi}\int_{-\pi}^{\pi} F_M(t)\,dt = 1$, and $F_M(t) \lesssim \frac{1}{\delta^2 M}$ for $\delta \le |t| \le \pi$. Substituting $\delta \sim \left(\frac{\|f\|_{\ell_\infty}}{M\|f'\|_{\ell_\infty}}\right)^{1/3}$, we get

$$|f_M(x) - f(x)| \lesssim \left(\frac{\|f\|_{\ell_\infty}\|f'\|_{\ell_\infty}^2}{M}\right)^{1/3},$$

which implies the desired bound. □

Claim 4.5.3. *For every $x \in \mathbb{C}^d$ with $\|x\|_{[W_0,W_1]_\theta} = 1$ and every $\varepsilon > 0$, there exists $f_x \in \mathcal{F}$ such that:*

- $f_x(\theta) = x$,
- *for every $\tau \in \mathbb{R}$, $B_{\mathcal{F}}(f_x, \tau) \le 1 + \varepsilon$ and $B_\infty(f_x, \tau) \lesssim \mathsf{d}$;*

Proof. By definition of $\|x\|_{[W_0,W_1]_\theta}$ from (4.11), let $f_x \in \mathcal{F}$ be the function with $f_x(\theta) = x$ and $\|f_x\|_{\mathcal{F}} \le \|x\|_{[W_0,W_1]_\theta} + \varepsilon \le 1 + \varepsilon$. In addition, since $\|f_x\|_{\mathcal{F}} = \sup_{\tau\in\mathbb{R}} B_{\mathcal{F}}(f_x, \tau)$, we obtain that $B_{\mathcal{F}}(f_x, \tau) \le 1 + \varepsilon$. Finally, we note that f_x is holomorphic on $\mathfrak{S}$, continuous on $\overline{\mathfrak{S}}$, and bounded, so by the Hadamard three-lines theorem, $B_\infty(f_x, \tau) \le \sup_{\substack{u\in(0,1)\\ \tau\in\mathbb{R}}} \|f(u + i\tau)\|_H \le \sup_{\tau\in\mathbb{R}} \max\{\|f_x(i\tau)\|_H, \|f_x(1 + i\tau)\|_H\} \le \mathsf{d}$. □

Claim 4.5.4. *For every $x \in \mathbb{C}^d$ with $\|x\|_{[W_0,W_1]_\theta} = 1$ and every $\varepsilon > 0$, there exists $f_x^{(2)} \in \mathcal{F}$ such that:*

- $\|f_x^{(2)}(\theta) - x\|_{\ell_\infty} \lesssim \varepsilon$, *and*
- *for every $\tau \in \mathbb{R}$, $B_{\mathcal{F}}(f_x^{(2)}, \tau) \le 1 + \varepsilon$, $B_\infty(f_x^{(2)}, \tau) \lesssim \mathsf{d}$, and $D_\infty(f_x^{(2)}, \tau) \lesssim \frac{\mathsf{d}^2\cdot\Lambda(\theta,\theta)}{\varepsilon}$.*

Proof. Let $f_x \in \mathcal{F}$ be from Claim 4.5.3 for the vector $x \in \mathbb{C}^d$ and ε. For $\sigma > 0$ let

$$f_x^{(2)}(z) \stackrel{\text{def}}{=} \mathop{\mathrm{E}}_{g\sim N(0,\sigma^2)} f_x(z + ig). \tag{4.21}$$

We note the bounds on $B_{\mathcal{F}}(f_x^{(2)}, \tau)$ and $B_\infty(f_x^{(2)}, \tau)$ are immediate from Jensen's inequality

and convexity of $\|\cdot\|\colon \mathbb{C}^d \to \mathbb{R}^{\geq 0}$. In order to bound $\|f_x^{(2)}(\theta) - x\|_{\ell_\infty}$, we have

$$\begin{aligned} f_x^{(2)}(\theta) - f_x(\theta) &\overset{(4.3.3)}{=} \int_{\partial\mathfrak{S}} \Big(f_x^{(2)}(z) - f_x(z)\Big)\, d\mu_\theta(z) \overset{(4.21)}{=} \int_{\partial\mathfrak{S}} \left(\mathop{\mathrm{E}}_{g\sim N(0,\sigma^2)} f_x(z+ig) - f_x(z)\right) d\mu_\theta(z) \\ &= \mathop{\mathrm{E}}_{g\sim N(0,\sigma^2)} \int_{\partial\mathfrak{S}} \left(f_x(z+ig) - f_x(z)\right)\, d\mu_\theta(z) \\ &= \mathop{\mathrm{E}}_{g\sim N(0,\sigma^2)} \int_{\partial\mathfrak{S}} f_x(z)\Big(P(\theta, z-ig) - P(\theta,z)\Big)\, dz, \end{aligned}$$

where we used the definition of $\mu_\theta(z)$ from Subsection 4.3 in the last line. Therefore,

$$\begin{aligned} \|f_x^{(2)}(\theta) - f_x(\theta)\|_{\ell_\infty} &\leq \left(\sup_{\tau\in\mathbb{R}} B_\infty(f_x,\tau)\right) \mathop{\mathrm{E}}_{g\sim N(0,\sigma^2)} \int_{\partial\mathfrak{S}} |P(\theta,z) - P(\theta,z-ig)|\, dz \\ &\lesssim \mathsf{d} \mathop{\mathrm{E}}_{g\sim N(0,\sigma^2)} \int_{\partial\mathfrak{S}} |P(\theta,z) - P(\theta,z-ig)|\, dz. \end{aligned} \tag{4.22}$$

Note that for $\theta \in (0,1)$, $P(\theta,\cdot)$ is symmetric around zero and unimodal, when $g \geq 0$, we have $P(\theta,z) \geq P(\theta,z-ig)$ when $\mathrm{Im}(z) \leq \frac{g}{2}$ and $P(\theta,z) \leq P(\theta,z-ig)$ when $\mathrm{Im}(z) \geq \frac{g}{2}$. So we have that when $g \geq 0$,

$$\begin{aligned} \int_{\partial\mathfrak{S}} |P(\theta,z) - P(\theta,z-ig)| dz &= \int\limits_{\substack{z\in\partial\mathfrak{S}\\ \mathrm{Im}(z)\leq\frac{g}{2}}} (P(\theta,z) - P(\theta,z-ig)) dz - \int\limits_{\substack{z\in\partial\mathfrak{S}\\ \mathrm{Im}(z)\geq\frac{g}{2}}} (P(\theta,z) - P(\theta,z-ig)) dz \\ &= \int\limits_{\substack{z\in\partial\mathfrak{S}\\ \mathrm{Im}(z)\leq\frac{g}{2}}} P(\theta,z) dz - \int\limits_{\substack{z\in\partial\mathfrak{S}\\ \mathrm{Im}(z)\leq-\frac{g}{2}}} P(\theta,z) dz - \int\limits_{\substack{z\in\partial\mathfrak{S}\\ \mathrm{Im}(z)\geq\frac{g}{2}}} P(\theta,z) dz + \int\limits_{\substack{z\in\partial\mathfrak{S}\\ \mathrm{Im}(z)\geq-\frac{g}{2}}} P(\theta,z) dz \\ &= 2 \int\limits_{\substack{z\in\partial\mathfrak{S}\\ |\mathrm{Im}(z)|\leq\frac{g}{2}}} P(\theta,z) dz. \end{aligned} \tag{4.23}$$

The case when $g \leq 0$ is symmetric, thus, we may combine (4.22) and (4.23) to conclude

$$\|f_x^{(2)}(\theta) - f_x(\theta)\|_{\ell_\infty} \lesssim \mathsf{d} \mathop{\mathrm{E}}_{g\sim N(0,\sigma^2)} \int\limits_{\substack{z\in\partial\mathfrak{S}\\ |\mathrm{Im}(z)|\leq\frac{|g|}{2}}} P(\theta,z) \lesssim \mathsf{d}\Lambda(\theta,\theta) \mathop{\mathrm{E}}_{g\sim N(0,\sigma^2)} |g| \lesssim \mathsf{d}\Lambda(\theta,\theta)\sigma \leq \varepsilon$$

when $\sigma \lesssim \frac{\varepsilon}{\mathsf{d}\cdot\Lambda(\theta,\theta)}$, where we used the fact that $P(\theta,z) \lesssim \Lambda(\theta,\theta)$ when $z \in \partial\mathfrak{S}$ and $\theta \in (0,1)$. Now let us upper bound $D_\infty(f_x^{(2)},\tau)$. Denote $p_\sigma(t)$ the p.d.f. of $N(0,\sigma^2)$. In addition, if we let $g_0, g_1, g_0^{(2)}, g_1^{(2)}\colon \mathbb{R} \to \mathbb{C}^d$ have $g_j(\tau) = f_x(j+i\tau)$ and $g_j^{(2)}(\tau) = f_x^{(2)}(j+i\tau)$ for

$j = 0, 1$, we have $g_j^{(2)} = g_j * p_\sigma(\tau)$. Thus, we have $\frac{dg_j^{(2)}}{d\tau}(\tau) = g_j * p'_\sigma(\tau)$,

$$\left\| \frac{d}{d\tau}[g_j^{(2)}(\tau)] \right\|_{\ell_\infty} = \|g_j * p'_\sigma\|_{\ell_\infty} \lesssim \frac{\mathsf{d}}{\sigma} \lesssim \frac{\mathsf{d}^2 \Lambda(\theta, \theta)}{\varepsilon}.$$

□

Claim 4.5.5. *For every $x \in \mathbb{C}^d$ with $\|x\|_{[W_0, W_1]_\theta} = 1$ and every $\varepsilon > 0$, there exists $f_x^{(3)} \in \mathcal{F}$ such that:*

- $\|f_x^{(3)}(\theta) - x\|_{\ell_\infty} \lesssim \varepsilon$,
- *for every $\tau \in \mathbb{R}$, $B_{\mathcal{F}}(f_x^{(3)}, \tau) \leq (1 + \varepsilon) \cdot e^{\frac{\varepsilon(1-\tau^2)}{\mathsf{d}}}$ and $B_\infty(f_x^{(3)}, \tau) \lesssim \mathsf{d} \cdot e^{-\frac{\varepsilon\tau^2}{\mathsf{d}}}$, and*
- *for every $\tau \in \mathbb{R}$, one has: $D_\infty(f_x^{(3)}, \tau) \lesssim \frac{\mathsf{d}^2 \cdot \Lambda(\theta,\theta)}{\varepsilon} \cdot e^{-\frac{\varepsilon\tau^2}{\mathsf{d}}} + \varepsilon|\tau| \cdot e^{-\frac{\varepsilon\tau^2}{\mathsf{d}}}$.*

Proof. We may consider $f_x^{(2)} \in \mathcal{F}$ from Claim 4.5.4 and set

$$f_x^{(3)}(z) = e^{\frac{\varepsilon z^2}{\mathsf{d}}} \cdot f_x^{(2)}(z).$$

All the desired properties are immediate to check. □

Claim 4.5.6. *For every $x \in \mathbb{C}^d$ with $\|x\|_{[W_0, W_1]_\theta} = 1$ and every $\varepsilon > 0$, there exists $f_x^{(4)} \in \mathcal{F}$ such that:*

- $\|f_x^{(4)}(\theta) - x\|_{\ell_\infty} \lesssim \varepsilon$,
- *for every $\tau \in \mathbb{R}$, $B_{\mathcal{F}}(f_x^{(4)}, \tau) \leq 1 + O(\varepsilon)$, $B_\infty(f_x^{(4)}, \tau) \lesssim \mathsf{d}$, $D_\infty(f_x^{(4)}, \tau) \lesssim \frac{\mathsf{d}^2 \cdot \Lambda(\theta,\theta)}{\varepsilon}$, and*
- *$f_x^{(4)}$ is $2\pi i L$-periodic for*
$$L \lesssim \sqrt{\frac{\mathsf{d} \cdot \log \frac{\mathsf{d}}{\varepsilon}}{\varepsilon}}.$$

Proof. We take $f_x^{(3)}$ from Claim 4.5.5, and set:

$$f_x^{(4)}(z) = \sum_{k \in \mathbb{Z}} f_x^{(3)}(z + 2\pi i L k).$$

All the desired properties are immediate to check. □

Denote for $n \in \mathbb{Z}$ the n-th Fourier coefficient:

$$a_n = \frac{1}{2\pi L} \int\limits_{-\pi L}^{\pi L} f_x^{(4)}(i\tau) e^{-\frac{in\tau}{L}} \, d\tau \in \mathbb{C}^d.$$

Claim 4.5.7. *For every $0 \leq \widetilde{\theta} \leq 1$, one has:*

$$\frac{1}{2\pi L} \int\limits_{-\pi L}^{\pi L} f_x^{(4)}(\widetilde{\theta} + i\tau) e^{-\frac{n(\widetilde{\theta}+i\tau)}{L}} \, d\tau = a_n.$$

Proof. First, let us show that the left-hand side, which we denote by $A(\widetilde{\theta})$, does not change when $\widetilde{\theta}$ is varied in $(0; 1)$. Consider the following contour on the complex plane for $k \geq 1$: $\theta_1 - i\pi Lk$ to $\theta_1 + i\pi Lk$ to $\theta_2 + i\pi Lk$ to $\theta_2 - i\pi Lk$ to $\theta_1 - i\pi Lk$. On the one hand, the integral of $f_x^{(4)}(z)e^{-\frac{nz}{L}}$ over it equals to zero. On the other hand, it is equal to $k \cdot (A(\theta_1) - A(\theta_2))$ plus a term that is independent of k. Since k is arbitrary, we get $A(\theta_1) = A(\theta_2)$. On the other hand, for $0 < \theta < 1$, we have $A(\theta) = A(0) = A(1) = a_n$, since the function $f_x^{(4)}(z)$ converges uniformly to the corresponding boundary value as the real part of z converges to zero or to one. □

Claim 4.5.8. *One has:*

- $\|a_n\|_{\ell_\infty} \leq \min\{1, e^{-n/L}\} \cdot \|f_x^{(4)}\|_{\ell_\infty} \lesssim \min\{1, e^{-n/L}\} \cdot \mathsf{d}$.
- *If we denote $f_x^{(5)} \in \mathcal{F}$ by*

$$f_x^{(5)}(z) = \sum_{|n| \leq M} \frac{M + 1 - |n|}{M + 1} \cdot a_n e^{\frac{nz}{L}},$$

then $\|f_x^{(5)}(z) - f_x^{(4)}(z)\|_{\ell_\infty} \leq \widetilde{\varepsilon}$ for

$$M \lesssim \frac{\mathsf{d}^5 \cdot L^2 \cdot \Lambda(\theta, \theta)^2}{\widetilde{\varepsilon}^3}.$$

Proof. The bound $\|a_n\|_{\ell_\infty} \leq \|f_x^{(4)}\|_{\ell_\infty} \lesssim \mathsf{d}$ is trivial. To show that $\|a_n\|_{\ell_\infty} \leq e^{-n/L} \cdot$

$\|f_x^{(4)}\|_{\ell_\infty}$, we just use Claim 4.5.7 with $\widetilde{\theta} = 1$:

$$a_n = \frac{1}{2\pi L} \int_{-\pi L}^{\pi L} f_x^{(4)}(1+i\tau) e^{-\frac{n(1+i\tau)}{L}}\, d\tau = e^{-n/L} \cdot \frac{1}{2\pi L} \int_{-\pi L}^{\pi L} f_x^{(4)}(1+i\tau) e^{-\frac{in\tau}{L}}\, d\tau.$$

$f_x^{(5)}(z)$ converges to $f_x^{(4)}(z)$ uniformly in ℓ_∞ for $\mathrm{Re} z = 0$ by Lemma 4.5.2. But due to Claim 4.5.7, it is also the case for $\mathrm{Re} z = 1$. We get the required bound on M from the conclusions of Lemma 4.5.2 and Claim 4.5.6. For $0 < \mathrm{Re} z < 1$, we simply use the Hadamard three-line theorem. □

Lemma 4.5.9. *For every $x \in \mathbb{C}^d$ with $\|x\|_{[W_0,W_1]_\theta} = 1$ and every $\varepsilon > 0$, there exists a function $\tilde{f}_x \in \mathcal{F}$ with*

$$\tilde{f}_x(z) = e^{\frac{z^2}{M}} \sum_{q \in \mathbb{Q}_M} v_q \cdot e^{qz},$$

where $\mathbb{Q}_M = \{\frac{s}{L} : |s| \le ML\}$ and $v_q \in \mathbb{C}^d$ for all $q \in \mathbb{Q}_M$ satisfying:

- $L \lesssim M = \mathrm{poly}(d/\varepsilon)$;
- $\|v_q\|_H \lesssim \min\{1, e^{-q}\} \cdot M$;
- $\|\tilde{f}_x\|_{\mathcal{F}} \le 1 + \frac{\varepsilon}{\mathsf{d}^2}$;
- $\|\tilde{f}_x(\theta) - x\|_{[W_0,W_1]_\theta} \le \frac{\varepsilon}{\mathsf{d}^2}$.

Proof. We take $f_x^{(5)}(z)$ from Claim 4.5.8 with $\widetilde{\varepsilon} \lesssim \mathrm{poly}(\varepsilon, \frac{1}{\mathsf{d}})$. This implies that $\|f_x^{(5)} - f_x^{(4)}\|_{\mathcal{F}} \lesssim \frac{\varepsilon}{\mathsf{d}^2}$ and $\|f_x^{(5)}(\theta) - x\|_{[W_0,W_1]_\theta} \le \frac{\varepsilon}{\mathsf{d}^2}$. Finally, setting $\tilde{f}_x(z) = e^{\frac{z^2}{M}} \cdot f_x^{(5)}(z)$, and a similar argument to that of Claim 4.5.5, the required properties hold. □

4.5.3 Convex program for $\texttt{ApproxRep}(x, \theta, \varepsilon; W)$

In this subsection, we present a convex program for solving $\texttt{ApproxRep}(x, \theta, \varepsilon; W)$ and argue that a good enough solution to this program can be a valid response for $\texttt{ApproxRep}(x, \theta, \varepsilon; W)$.

Given any vector $x \in \mathbb{C}^d$, we may compute the vector $y \in \mathbb{C}^d$ with $y = \frac{x}{\|x\|_H}$. We note that since $\|x\|_H \le \|x\|_\theta \le \mathsf{d}\|x\|_H$ from (4.17), we have $1 \le \|y\|_\theta \le \mathsf{d}$, and recall that $\mathsf{d} = \mathrm{poly}(d)$. Thus, we may assume that calls to $\texttt{ApproxRep}(x, \theta, \varepsilon; W)$ always have

$x \in \mathbb{C}^d$ satisfying

$$\|x\|_H = 1 \qquad \text{and} \qquad 1 \le \|x\|_\theta \le \mathsf{d}. \tag{4.24}$$

Given $x \in \mathbb{C}^d$ satisfying (4.24), we will define a convex program $\texttt{Rep}(x, \theta, \varepsilon; W)$ which takes a parameter $\varepsilon > 0$ and has size $\mathrm{poly}(d/\varepsilon)$ whose optimum will give a valid response for $\texttt{ApproxRep}(x, \theta, 10\varepsilon; W)$. Recall the parameter $M = \mathrm{poly}(d, \frac{1}{\varepsilon}, \Lambda(\theta, \theta))$, as well as the definition of the set $\mathbb{Q}_M \subset \mathbb{R}$ from Lemma 4.5.9, and let $N \lesssim \frac{M^4 \mathsf{d}}{\varepsilon}$ for a large enough constant.[3] For $j \in \{0, 1\}$, consider the subset:

$$\mathbb{D}_N^{(j)} \stackrel{\text{def}}{=} \left\{ j + \frac{is}{N} \in \partial\mathfrak{S} : s \in \mathbb{Z}, |s| \le MN \right\}.$$

Let the sequence of vectors $V = (v_q \in \mathbb{C}^d : q \in \mathbb{Q}_M)$ define a map $f_V \in \mathcal{F}$ given by:

$$f_V(z) \stackrel{\text{def}}{=} e^{\frac{z^2}{M}} \sum_{q \in \mathbb{Q}_M} v_q \cdot e^{qz}. \tag{4.25}$$

The convex program $\texttt{Rep}(x, \theta, \varepsilon; W)$ is given by:

$$\texttt{Rep}(x, \theta, \varepsilon; W) = \begin{cases} \min\limits_{\substack{V \in (\mathbb{C}^d)^{|\mathbb{Q}_M|} \\ \alpha \in \mathbb{R}^{\ge 0}}} \alpha & \\ \text{s.t} & \text{i.} \quad \forall z \in \mathbb{D}_N^{(0)}, \|f_V(z)\|_W \le \alpha + \varepsilon \\ & \text{ii.} \quad \forall z \in \mathbb{D}_N^{(1)}, \|f_V(z)\|_H \le \alpha + \varepsilon \\ & \text{iii.} \quad \forall q \in \mathbb{Q}_M, \|v_q\|_H \max\{e^q, 1\} \le 2M\mathsf{d} \\ & \text{iv.} \quad \|f_V(\theta) - x\|_H \le \frac{2\varepsilon}{\mathsf{d}} \end{cases}. \tag{4.26}$$

In the language of Grötschel, Lovász and Schrijver [72], we will argue that solving the weak optimization problem of $\texttt{Rep}(x, \theta, \varepsilon; W)$ satisfies the requirements of $\texttt{ApproxRep}(x, \theta, 8\varepsilon; W)$. After that, we address the problem of computing $\texttt{Rep}(x, \theta, \varepsilon; W)$.

Lemma 4.5.10. *For $x \in \mathbb{C}^d$ and $\varepsilon > 0$, and $\theta \in (0, 1)$ with $\Lambda(\theta, \theta) \le \mathrm{poly}(d)$, let*

[3] see (4.9) for the definition of $\Lambda(\theta_1, \theta_2)$, and note that $\Lambda(\theta, \theta) = \mathrm{poly}(d)$ for $\theta \in (\frac{1}{\mathrm{poly}(d)}, 1 - \frac{1}{\mathrm{poly}(d)})$.

$(V, \alpha) \in (\mathbb{C}^d)^{|\mathbb{Q}_M|} \times \mathbb{R}$ *be a feasible solution to* $\texttt{Rep}(x, \theta, \varepsilon; W)$*, where the optimum of* $\texttt{Rep}(x, \theta, \varepsilon; W)$ *is at least* $\alpha - 5\varepsilon$*. Then* $f_V \in \mathcal{F}$ *is a valid output of* $\texttt{ApproxRep}(x, \theta, 8\varepsilon; W)$.

Before giving the proof of Lemma 4.5.10, we give some discussion as well as a sequence of claims from which Lemma 4.5.10 will easily follow.

Consider the unit vector $a = \frac{x}{\|x\|_\theta}$,[4] and let $f_a \in \mathcal{F}$ be an *optimal representative* for a at θ in $\mathcal{F}$. In other words, $f_a(\theta) = a$ and $\|f_a\|_{\mathcal{F}} = 1$. Applying Lemma 4.5.9, there exists some $\tilde{f}_a \in \mathcal{F}$ such that:

- $\tilde{f}_a(z) = e^{\frac{z^2}{M}} \sum_{q \in \mathbb{Q}_N} v_q e^{qz}$, and for all $q \in \mathbb{Q}_M$, $v_q \in \mathbb{C}^d$ with $\|v_q\|_H \cdot \max\{e^q, 1\} \leq M$.
- $\|\tilde{f}_a\|_{\mathcal{F}} \leq 1 + \frac{\varepsilon}{\mathsf{d}^2}$ and $\|\tilde{f}_a(\theta) - a\|_H \leq \frac{\varepsilon}{\mathsf{d}^2}$.

Therefore, let the function $\tilde{f}_x \in \mathcal{F}$ be

$$\tilde{f}_x(z) \stackrel{\text{def}}{=} \|x\|_\theta \tilde{f}_a(z) \tag{4.27}$$

which satisfies,

$$\|\tilde{f}_x(\theta) - x\|_H \leq \|x\|_\theta \cdot \|\tilde{f}_a(\theta) - a\|_H \leq \|x\|_\theta \cdot \frac{\varepsilon}{\mathsf{d}^2} \leq \frac{\varepsilon}{\mathsf{d}}. \tag{4.28}$$

In addition, we have:

$$\|\tilde{f}_x(\theta) - x\|_\theta \leq \mathsf{d}\|\tilde{f}_x(\theta) - x\|_H \leq \frac{\varepsilon}{\mathsf{d}} \cdot \|x\|_\theta \tag{4.29}$$

$$\|x\|_\theta \|v_q\| \cdot \max\{e^q, 1\} \leq M\mathsf{d} \qquad \text{for all } q \in \mathbb{Q}_M, \tag{4.30}$$

$$\|\tilde{f}_x\|_{\mathcal{F}} \leq \|x\|_\theta \cdot \|\tilde{f}_a\|_{\mathcal{F}} \leq \left(1 + \frac{\varepsilon}{\mathsf{d}^2}\right) \|x\|_\theta. \tag{4.31}$$

We note that the above facts imply that if $(V, \alpha) \in (\mathbb{C}^d)^{|\mathbb{Q}_M|} \times \mathbb{R}$, where $V = (v_q : q \in \mathbb{Q}_M)$ and the vectors v_q define $\tilde{f}_x \in \mathcal{F}$ according to (4.25), and $\alpha = \|\tilde{f}_x\|_{\mathcal{F}}$, then (V, α) is a feasible solution for $\texttt{Rep}(x, \theta, \varepsilon; W)$.

[4] algorithmically, we do not have access to this vector

Claim 4.5.11. *Suppose $V = (v_q : q \in \mathbb{Q}_M), \alpha \in \mathbb{R}^d$ defines a feasible solution for* $\texttt{Rep}(x, \theta, \varepsilon; W)$*, and suppose $z = j + i\tau \in \partial\mathfrak{S}$ with $|\tau| \geq M$, then $\|f_V(z)\|_H \leq \frac{1}{2\mathsf{d}}$.*

Proof. We simply follow the computations, using the third constraint of (4.26):

$$\begin{aligned}\|f_V(z)\|_H &\leq e^{\frac{j^2-\tau^2}{M}} \left(\sum_{q \in \mathbb{Q}_M} \|v_q\|_H \cdot e^{qj} \right) \\ &\leq e^{-\frac{\tau^2-1}{M}} \left(|\mathbb{Q}_M| \cdot 2M\mathsf{d} \right) \leq \frac{1}{2\mathsf{d}},\end{aligned}$$

when $\tau \geq M \gg \sqrt{M \log(d) \log(M)}$, since $\mathsf{d} = \mathrm{poly}(d)$. □

Corollary 4.5.12. *Let $V = (v_q : q \in \mathbb{Q}_M), \alpha \in \mathbb{R}$ be a feasible solution for* $\texttt{Rep}(x, \theta, \varepsilon; W)$*. We have that:*

$$\|f_V\|_{\mathcal{F}} = \max \left\{ \sup_{|\tau| \leq M} \|f_V(i\tau)\|_W, \sup_{|\tau| \leq M} \|f_V(1 + i\tau)\|_H \right\}.$$

Proof. From the fourth constraint of (4.26), $\|f_V(\theta) - x\|_\theta \leq \mathsf{d}\|f_V(\theta) - x\|_H \leq 2\varepsilon$. Therefore, we have $\|f_V\|_{\mathcal{F}} \geq \|f_V(\theta)\|_\theta \geq \|x\|_\theta - 2\varepsilon \geq 1 - 2\varepsilon > \frac{1}{2}$ for small enough $\varepsilon < \frac{1}{4}$. Since by Claim 4.5.11, any $|\tau| \geq M$ satisfies $\|f_V(i\tau)\|_W \leq \mathsf{d}\|f_V(i\tau)\|_H \leq \frac{1}{2}$ and $\|f_V(1 + i\tau)\|_H \leq \frac{1}{2\mathsf{d}}$, we must have

$$\|f_V\|_{\mathcal{F}} = \max \left\{ \sup_{|\tau| \leq M} \|f_V(i\tau)\|_W, \sup_{|\tau| \leq M} \|f_V(1 + i\tau)\|_H \right\}.$$

□

Let us write $f_V \colon \overline{\mathfrak{S}} \to \mathbb{C}^d$ as $f_V = (g_1(z), \ldots, g_d(z))$, where $g_k \colon \overline{\mathfrak{S}} \to \mathbb{C}$ is given by the k-th coordinate of f_V. Then, taking derivatives, all $k \in [d]$ satisfy

$$g_k'(z) = e^{z^2/M} \sum_{q \in \mathbb{Q}_M} (v_q)_k e^{qz} \left(q + \frac{2z}{M} \right).$$

Thus, when (V, α) is feasible in $\texttt{Rep}(x, \theta, \varepsilon; W)$, the third constraint of (4.26) implies that

for each $z = j + i\tau \in \partial\mathfrak{S}$ with $|\tau| \le M$ and every $k \in [d]$,

$$\begin{aligned} |g_k'(z)| &\le e \cdot e^{-\tau^2/M} \sum_{q \in \mathbb{Q}_M} |(v_q)_k \cdot e^q| (q+2) \\ &\le e|\mathbb{Q}_M| \cdot 2M\mathsf{d} \cdot (M+2) \lesssim M^4\mathsf{d}. \end{aligned} \tag{4.32}$$

Claim 4.5.13. *For large enough $N \lesssim \frac{M^4\mathsf{d}^3}{\varepsilon} = \mathrm{poly}(d/\varepsilon)$, we have that any (V, α) which is feasible for $\mathtt{Rep}(x, \theta, \varepsilon; W)$ satisfies*

$$\|f_V\|_{\mathcal{F}} \le \max\left\{ \sup_{z \in \mathbb{D}_N^{(0)}} \|f_V(z)\|_W, \sup_{z \in \mathbb{D}_N^{(1)}} \|f_V(z)\|_H \right\} + \varepsilon.$$

Proof. By definition of $\mathbb{D}_N^{(j)}$, any $z \in \partial\mathfrak{S}$ with$|\mathrm{Im}(z)| \le M$ and $\mathrm{Re}(z) = j$, there exists some $z' \in \mathbb{D}_N^{(j)}$ with $|z - z'| \le \frac{1}{N}$. This implies that every $k \in [d]$ satisfies $|g_k(z) - g_k(z')| \lesssim |z - z'| \cdot M^4\mathsf{d} = \frac{M^4\mathsf{d}}{N}$ by (4.32). Thus,

$$\|f_V(z) - f_V(z')\|_H^2 \le \sum_{k=1}^{d} |g_k(z) - g_k(z')|^2 \le \frac{\varepsilon^2}{\mathsf{d}^2},$$

for a large enough setting of N. Therefore, we have $\|f_V(z) - f_V(z')\|_H \le \frac{\varepsilon}{\mathsf{d}}$, and that $\|f_V(z) - f_V(z')\|_W \le \varepsilon$, which completes the proof. □

Claim 4.5.14. *We have that*

$$\|x\|_\theta \le \mathtt{Rep}(x, \theta, \varepsilon; W) + 3\varepsilon.$$

Proof. Every vector (V, α) which is feasible in $\mathtt{Rep}(x, \theta, \varepsilon; W)$ defines a function $f_V \in \mathcal{F}$ by (4.25), which by Claim 4.5.13, satisfies

$$\|f_V\|_{\mathcal{F}} \le \max\left\{ \sup_{z \in \mathbb{D}_N^{(0)}} \|f_V(z)\|_W, \sup_{z \in \mathbb{D}_N^{(1)}} \|f_V(z)\|_H \right\} + \varepsilon.$$

Therefore, for all (V, α) which are feasible for $\mathtt{Rep}(x, \theta, \varepsilon; W)$, we have:

$$\|x\|_\theta \le \|f_V(\theta)\|_\theta + \|f_V(\theta) - x\|_\theta \le \|f_V\|_{\mathcal{F}} + \varepsilon \le \alpha + 3\varepsilon,$$

which implies $\|x\|_\theta \le \texttt{Rep}(x, \theta, \varepsilon; W) + 3\varepsilon$. □

Proof of Lemma 4.5.10. First, consider the solution (V, α) whose values of $(v_q : q \in \mathbb{Q}_M)$ define the function $\tilde{f}_x$ according to (4.25), and $\alpha = \|\tilde{f}_x\|_{\mathcal{F}}$. Then, (V, α) lies in the feasible set of $\texttt{Rep}(x, \theta, \varepsilon; W)$. In particular, (4.28) shows that v_q satisfies the last constraint of (4.26). In addition, since $\|x\|_\theta \le \mathsf{d}$, the third constraint of (4.26) is also satisfied. Therefore, we have that the optimal solution of $\texttt{Rep}(x, \theta, \varepsilon; W)$ satisfies:

$$\texttt{Rep}(x, \theta, \varepsilon; W) \le \|\tilde{f}_x\|_{\mathcal{F}} \le \left(1 + \frac{\varepsilon}{\mathsf{d}^2}\right) \|x\|_\theta.$$

If, in addition, (V, α) has $\alpha \le \texttt{Rep}(x, \theta, \varepsilon; W) + 5\varepsilon$, then by Claim 4.5.14 and $\|x\|_\theta \ge 1$,

$$(1 - 8\varepsilon)\|x\|_\theta \le \alpha \le \left(1 + \frac{\varepsilon}{\mathsf{d}^2} + \varepsilon\right) \|x\|_\theta,$$

which is a valid output of $\texttt{ApproxRep}(x, \theta, 8\varepsilon; W)$. □

4.5.4 Computing $\texttt{ApproxRep}(x, \theta, \varepsilon; W)$ with $\texttt{MEM}(B_W)$

In this section, we show how to compute an approximate optimum of $\texttt{Rep}(x, \theta, \varepsilon; X)$ described in (4.26). In particular, we will compute some value (V, α) which satisfies the conditions of Lemma 4.5.10. In other words, the solution (V, α) is feasible for $\texttt{Rep}(x, \theta, \varepsilon; W)$, and the optimal value of $\texttt{Rep}(x, \theta, \varepsilon; W)$ is at least $\alpha - 5\varepsilon$. If the algorithm runs in $\mathrm{poly}(d/\varepsilon)$ time, by Lemma 4.5.10, this would give us the required algorithm for $\texttt{ApproxRep}(x, \theta, 8\varepsilon; X)$.

Let $P \subset (\mathbb{C}^d)^{|\mathbb{Q}_M|} \times \mathbb{R}^{\ge 0}$ be the set:

$$P = \left\{ ((v_q \max\{e^q, 1\} : q \in \mathbb{Q}_M), \alpha) \in (\mathbb{C}^d)^{|\mathbb{Q}_M|} \times \mathbb{R}^{\ge 0} : \begin{array}{lll} \text{i. } \forall z \in \mathbb{D}_N^{(0)} & \|f_V(z)\|_W \le \alpha + \frac{\varepsilon}{2} \\ \text{ii. } \forall z \in \mathbb{D}_N^{(1)} & \|f_V(z)\|_H \le \alpha + \frac{\varepsilon}{2} \\ \text{iii. } \forall q \in \mathbb{Q}_M & \|v_q\|_H \max\{e^q, 1\} \le \frac{3M \cdot \mathsf{d}}{2} \\ \text{iv.} & \|f_V(\theta) - x\|_H \le \frac{3\varepsilon}{2\mathsf{d}} \\ \text{v.} & \sum_{q \in \mathbb{Q}_M} \|v_q\|_H^2 \max\{e^{2q}, 1\} + \alpha^2 \end{array} \right. \tag{4.33}$$

where $R = \mathrm{poly}(d/\varepsilon)$ is a large enough parameter. In addition, we view the set $P \subset (\mathbb{C}^d)^{|\mathbb{Q}_M|} \times \mathbb{R}$ as a subset of $(\mathbb{R}^{2d})^{|\mathbb{Q}_M|} \times \mathbb{R}$ by separating the real and imaginary parts of the vectors v_q for $q \in \mathbb{Q}_M$.

4.5.4.1 Properties of the set P

The following are a couple of useful facts showing that the convex set P is nice. In particular, we will have that P is convex and inscribed within a Euclidean ball of polynomial radius. Any solution *close* to P (by an inverse polynomial amount), gives a feasible solution to $\texttt{Rep}(x, \theta, \varepsilon; W)$. The *approximate* representative to x, given by $\tilde{f}_x$ in (4.27) lies well within P (by an inverse polynomial amount). Finally, there exists an explicit solution well within P (by an inverse polynomial amount).

For the remainder of the section, we let $\delta = \mathrm{poly}\left(\varepsilon, \frac{1}{d}\right)$ be the parameter:

$$\delta \stackrel{\text{def}}{=} \frac{\varepsilon}{2e \cdot |\mathbb{Q}_M| \cdot \mathsf{d}}.$$

Fact 4.5.15. *The set P is convex and is contained in a Euclidean ball of radius R.*

Lemma 4.5.16. *Let $(U, \alpha) \in B(P, \delta)$ where $U = (u_q : q \in \mathbb{Q}_M)$, and let $V = (v_q : q \in \mathbb{Q}_M)$ where $v_q = \frac{u_q}{\max\{e^q, 1\}}$. Then, (V, α) is a feasible solution to $\texttt{Rep}(x, \theta, \varepsilon; W)$.*

Proof. Consider $(U, \alpha) \in B(P, \delta)$ and let $(\widetilde{U}, \widetilde{\alpha}) \in P$ with $\|(U, \alpha) - (\widetilde{U}, \widetilde{\alpha})\|_2 \leq \delta$. We denote $\widetilde{U} = (\widetilde{u_q} : q \in \mathbb{Q}_M)$ and $\widetilde{V} = (\widetilde{v_q} : q \in \mathbb{Q}_M)$ where $\widetilde{v_q} = \frac{\widetilde{u_q}}{\max\{e^q, 1\}}$. Therefore, we have

$$\sum_{q \in \mathbb{Q}_M} \|u_q - \widetilde{u_q}\|_H^2 + |\alpha - \widetilde{\alpha}|^2 \leq \delta^2.$$

We will check that assuming constraints (i–iv) in (4.33), we can satisfy constraints (i–iv) in (4.26) for the point (V, α). All the subsequent checks proceed in the same fashion: we first use the triangle inequality to argue about $(\widetilde{U}, \widetilde{\alpha})$ and use (4.17) and the constraints (i–iv) in (4.33) of $(\widetilde{U}, \widetilde{\alpha})$ to deduce (V, α) is feasible for $\texttt{Rep}(x, \theta, \varepsilon; W)$.

iv. We simply follow the computations:

$$\begin{aligned}\|f_V(\theta) - x\|_H &\le \|f_{\widetilde{V}}(\theta) - x\|_H + \|f_V(\theta) - f_{\widetilde{V}}(\theta)\|_H \le \frac{3\varepsilon}{2\mathsf{d}} + e^{\theta^2/M} \sum_{q\in\mathbb{Q}_M} e^{q\theta}\|v_q - \widetilde{v_q}\|_H \\ &\le \frac{3\varepsilon}{2\mathsf{d}} + e^{\theta^2/M} \sum_{q\in\mathbb{Q}_M} \frac{e^{q\theta}}{\max\{e^q,1\}}\|u_q - \widetilde{u_q}\|_H \le \frac{3\varepsilon}{2\mathsf{d}} + e \sum_{q\in\mathbb{Q}_M} \|u_q - \widetilde{u_q}\|_H \\ &\le \frac{3\varepsilon}{2\mathsf{d}} + e|\mathbb{Q}_M|\delta \le \frac{2\varepsilon}{\mathsf{d}}.\end{aligned}$$

iii. For $q \in \mathbb{Q}_M$, we have:

$$\|v_q\|_H \max\{e^q,1\} \le \|\widetilde{v_q}\|_H \max\{e^q,1\} + \|u_q - \widetilde{u_q}\|_H \le \frac{3M\cdot\mathsf{d}}{2} + \delta \le 2M\cdot\mathsf{d}.$$

ii. For $z = 1 + i\tau \in \mathbb{D}_N^{(1)}$, we have:

$$\begin{aligned}\|f_V(z)\|_H &\le \|f_{\widetilde{V}}(z)\|_H + \|f_V(z) - f_{\widetilde{V}}(z)\|_H \le \alpha + \frac{\varepsilon}{2} + e^{-\frac{\tau^2-1}{M}} \sum_{q\in\mathbb{Q}_M} e^q\|\widetilde{v_q} - v_q\|_H \\ &\le \alpha + \frac{\varepsilon}{2} + e|\mathbb{Q}_M|\delta \le \alpha + \varepsilon.\end{aligned}$$

i. In addition, for $z = i\tau \in \mathbb{D}_N^{(1)}$, we similarly have,

$$\begin{aligned}\|f_V(z)\|_W &\le \|f_{\widetilde{V}}(z)\|_W + \mathsf{d}\|f_V(z) - f_{\widetilde{V}}(z)\|_H \le \alpha + \frac{\varepsilon}{2} + \mathsf{d}\cdot e^{-\frac{\tau^2}{M}} \sum_{q\in\mathbb{Q}_M} \|\widetilde{v_q} - v_q\|_H \\ &\le \alpha + \frac{\varepsilon}{2} + \mathsf{d}|\mathbb{Q}_M|\delta \le \alpha + \varepsilon.\end{aligned}$$

This completes the proof. □

Lemma 4.5.17. *We have $B((U,\alpha),\delta) \subset P$, where (U,α) is given by $U = (v_q \max\{e^q,1\} : q \in \mathbb{Q}_M)$ where the vectors $v_q \in \mathbb{C}^d$ define $\tilde{f}_x$ in (4.27) according to (4.25), and $\alpha = \|\tilde{f}_x\|_{\mathcal{F}}$. In other words, $(U,\alpha) \in B(P,-\delta)$.*

Proof. Consider the point (V,α) where the vectors v_q define the function $\tilde{f}_x \in \mathcal{F}$ from (4.27) and $\alpha = \|\tilde{f}_x\|_{\mathcal{F}}$. Let $U = (u_q : q \in \mathbb{Q}_M)$ where $u_q = v_q \max\{e^q,1\}$. We claim that $B((U,\alpha),\delta) \subset P$. In particular, consider any (U',α') with $U' = (u'_q : q \in \mathbb{Q}_M)$ where

$\|(U', \alpha') - (U, \alpha)\|_2 \leq \delta$, i.e.,

$$\sum_{q \in \mathbb{Q}_M} \|u'_q - u_q\|_H^2 + |\alpha' - \alpha|^2 \leq \delta^2.$$

With arguments very similar to those in the proof of Lemma 4.5.16 above, we may check that constraints (i–iv) of (4.33) have $(U', \alpha') \in P$ since v_q satisfy (4.29) and (4.30)[5]. It remains to check that constraint (v) is satisfied. Letting $v'_q = \frac{u'_q}{\max\{e^q, 1\}}$,

$$\begin{aligned}
\sum_{q \in \mathbb{Q}_M} \|v'_q\|_H^2 \max\{e^{2q}, 1\} + \alpha'^2 &\leq \sum_{q \in \mathbb{Q}_M} (\|u_q\|_H + \|u'_q - u_q\|_H)^2 + (\alpha + |\alpha' - \alpha|)^2 \\
&\leq 2 \left(\sum_{q \in \mathbb{Q}_M} \|u_q\|_H^2 + \alpha^2 \right) + 2 \left(\sum_{q \in \mathbb{Q}_M} \|u'_q - u_q\|_H^2 + |\alpha' - \alpha|^2 \right) \\
&\leq 2 \left(\sum_{q \in \mathbb{Q}_M} \|v_q\|_H^2 \max\{e^{2q}, 1\} + \alpha^2 \right) + 2\delta^2 \\
&\leq 2 \left(|\mathbb{Q}_M| M^2 \mathsf{d}^2 + \mathsf{d}^2 \right) + 2\delta^2 \leq R.
\end{aligned}$$

Therefore, we may conclude that $(U', \alpha') \in P$, which implies that $B((U, \alpha), \delta) \subset P$. □

Corollary 4.5.18. *We have that* $\min_{(V,\alpha) \in B(P, -\delta)} \alpha \leq \|\tilde{f}_x\|_{\mathcal{F}}$.

In addition, using the very similar arguments as in the proof of Lemma 4.5.16, one may deduce the following lemma.

Lemma 4.5.19. *Let* (U, α) *be the vector given by:*

$$u_q = \begin{cases} x \cdot e^{-\theta^2/M} & q = 0 \\ 0 & \textit{otherwise} \end{cases} \qquad \textit{and} \qquad \alpha = \mathsf{d}.$$

Then, $B((U, \alpha), \delta) \subset P$.

[5]Note that v_q here corresponds to $v_q \|x\|_\theta$ in (4.29) and (4.30), since in (4.29) and (4.30), v_q are the vectors which define $\tilde{f}_a$, and $\tilde{f}_x = \|x\|_\theta \tilde{f}_a$.

4.5.4.2 Optimizing over P

In this subsection, we show how to optimize over the convex set P defined in (4.33) using the tools from [94].

Definition 40 (Optimization Oracle ($\texttt{OPT}(K)$) [94])**.** *For a convex set $K \subset \mathbb{R}^m$, given a unit vector $c \in \mathbb{R}^m$ and a real number $\delta > 0$, with probability $1 - \delta$, the oracle either finds a vector $y \in \mathbb{R}^m$ such that $y \in B(K, \delta)$ and $c^\intercal x \leq c^\intercal y + \delta$ for all $B(K, -\delta)$, or asserts $B(K, -\delta)$ is empty.*

The next lemma, combined with Lemma 4.5.10 shows that in order to solve $\texttt{ApproxRep}(x, \theta, \varepsilon; W)$, it suffices to give an optimization oracle for P, $\texttt{OPT}(P)$.

Lemma 4.5.20. *Let (U, α) be the response to a call to $\texttt{OPT}(P)$ with vector $c = (0, \ldots, 0, -1)$ and $\delta = \frac{\varepsilon}{2e|\mathbb{Q}_M|\mathsf{d}}$, and let $V = (v_q : q \in \mathbb{Q}_M)$ be given by $v_q = \frac{u_q}{\max\{e^q, 1\}}$. Then, (V, α) is a feasible solution for $\texttt{Rep}(x, \theta, \varepsilon; W)$ and the optimum of $\texttt{Rep}(x, \theta, \varepsilon; W)$ is at least $\alpha - 5\varepsilon$.*

Proof. Note from Lemma 4.5.17 that $B(P, -\delta)$ is non-empty, so that $\texttt{OPT}(P)$ always returns a vector $(U, \alpha) \in (\mathbb{R}^{2d})^{|\mathbb{Q}_M|} \times \mathbb{R}$ such that $(U, \alpha) \in B(P, \delta)$ and $\alpha' \geq \alpha - \delta$ for all $(U', \alpha') \in B(P, -\delta)$. We note that by Lemma 4.5.16, (V, q) is feasible for $\texttt{Rep}(x, \theta, \varepsilon; W)$.

Additionally, let (U', α') where $U' = (v'_q \max\{e^q, 1\} : q \in \mathbb{Q}_M)$ where the vectors v'_q define $\tilde{f}_x$ from (4.27) according to (4.25), and $\alpha' = \|\tilde{f}_x\|_{\mathcal{F}}$. By Lemma 4.5.17, $(U', \alpha') \in B(P, -\delta)$, which implies $\alpha - \delta \leq \alpha' = \|\tilde{f}_x\|_{\mathcal{F}}$, and by (4.31), $\|\tilde{f}_x\|_{\mathcal{F}} \leq (1 + \frac{\varepsilon}{\mathsf{d}^2})\|x\|_\theta$. Finally, using Claim 4.5.14, and the fact that $\texttt{Rep}(x, \theta, \varepsilon; W) \leq 2\mathsf{d}$,

$$\alpha \leq \texttt{Rep}(x, \theta, \varepsilon; W) + \frac{2\varepsilon}{\mathsf{d}} + 4\varepsilon + \delta \leq \texttt{Rep}(x, \theta, \varepsilon; W) + 5\varepsilon,$$

which completes the proof. □

Given Lemma 4.5.20 and Lemma 4.5.10, in order to solve $\texttt{ApproxRep}(x, \theta, 8\varepsilon; W)$ it suffices to give an algorithm which solves $\texttt{OPT}(K)$ with a unit vector c and error δ in time $\mathrm{poly}(\frac{d}{\varepsilon}, \frac{1}{\delta})$. In order to do this, we will utilize the reduction from [94] which reduces the optimization problem to the separation problem.

Definition 41 (Separation Oracle ($\texttt{SEP}(K)$) [94]). *For a convex set $K \subset \mathbb{R}^m$, given a vector $y \in \mathbb{R}^m$ and a real number $\delta > 0$, with probability $1 - \delta$, the oracle either asserts $y \in B(K, \delta)$, or finds a unit vector $c \in \mathbb{R}^m$ such that $c^\intercal x \leq c^\intercal y + \delta$ for all $x \in B(K, -\delta)$.*

Theorem 51 (Theorem 15 from [94]). *Let $K \subset \mathbb{R}^m$ be a convex set satisfying $B(0, r) \subset K \subset B(0, 1)$, and let $\kappa = \frac{1}{r}$. For any $\delta \in (0, 1)$, with probability $1 - \delta$, one can compute $\texttt{OPT}(K)$ with a unit vector c and parameter δ with $O(m \log(\frac{m\kappa}{\delta}))$ calls to $\texttt{SEP}(K)$ with error parameter $\delta' = \text{poly}(m, \delta, \frac{1}{\kappa})$ and $\text{poly}(m, \log(\frac{\kappa}{\delta}))$ additional time.*

Given Theorem 51, we give an algorithm which solves $\texttt{OPT}(P)$ using calls to $\texttt{SEP}(P)$.

Lemma 4.5.21. *There exists an algorithm for $\texttt{OPT}(P)$ with a unit vector c and error parameter δ making $\text{poly}\left(\frac{d}{\varepsilon}, \log\left(\frac{1}{\delta}\right)\right)$ calls to $\texttt{SEP}(P)$ with error parameter $\delta' = \text{poly}\left(\frac{d}{\varepsilon}, \delta\right)$ and $\text{poly}\left(\frac{d}{\varepsilon}, \log\left(\frac{1}{\delta}\right)\right)$ additional time.*

Proof. Recall that by Lemma 4.5.19, there exists a vector $(U_0, \alpha_0) \in B(P, -\delta)$, which we may compute algorithmically. This implies that the set $P' = \{(U, \alpha) - (U_0, \alpha_0) : (U, \alpha) \in P\}$, has $B(0, \delta) \subset P' \subset B(0, 2R)$. This in turn, implies that the set $P_0 = \{\frac{1}{2R}(U, \alpha) : (U, \alpha) \in P'\}$ has $B(0, \frac{\delta}{2R}) \subset P_0 \subset B(0, 1)$. Suppose $(\widetilde{U}, \widetilde{\alpha})$ is the output of $\texttt{OPT}(P_0)$ with a unit vector c and error parameter $\frac{\delta}{2R}$. In addition, one may easily verify that $2R(\widetilde{U}, \widetilde{\alpha}) + (U_0, \alpha_0)$ is a valid output for $\texttt{OPT}(P)$ with unit vector c and error parameter δ.

Given Theorem 51, and the fact that $R = \text{poly}(d/\varepsilon)$, it suffices to show that one may implement $\texttt{SEP}(P_0)$ with $\texttt{SEP}(P)$. Consider a call to $\texttt{SEP}(P_0)$ with some point (U, α) and error parameter $\delta' > 0$. Let $(\widetilde{U}, \widetilde{\alpha}) = 2R(U, \alpha) + (U_0, \alpha_0)$. Then if $(\widetilde{U}, \widetilde{\alpha}) \in B(P, 2R\delta')$, then $(U, \alpha) \in B(P_0, \delta')$. If c is a unit vector where $c^\intercal \cdot (U', \alpha') \leq c^\intercal \cdot (\widetilde{U}, \widetilde{\alpha}) + 2R\delta'$ for all $(U', \alpha') \in B(P, -2R\delta')$, then, the vector c is a valid output of $\texttt{SEP}(P_0)$. □

Thus, given Lemma 4.5.21, it suffices to design an algorithm for $\texttt{SEP}(P)$ which runs in $\text{poly}\left(\frac{d}{\varepsilon}, \delta\right)$ time with error parameter δ.

Lemma 4.5.22. *There exists an oracle for $\texttt{SEP}(P)$ with error parameter δ which makes $\text{poly}\left(\frac{d}{\varepsilon}\right)$ calls to an oracle $\texttt{SEP}(B_W)$ with error parameter $\delta' = \text{poly}\left(\frac{d}{\varepsilon}, \frac{1}{\delta}\right)$ and takes $\text{poly}\left(\frac{d}{\varepsilon}\right)$ additional time.*

Proof. Consider some $(U, \alpha) \in (\mathbb{R}^{2d})^{|\mathbb{Q}_M|} \times \mathbb{R}$ with $U = (u_q : q \in \mathbb{Q}_M)$ which is an input to $\texttt{SEP}(P)$ with error parameter δ. Let $V = (v_q : q \in \mathbb{Q}_M)$ have $v_q = \frac{u_q}{\max\{e^q, 1\}}$. The algorithm proceeds as follows:

1. First, check whether constraint (v) in (4.33) is violated by computing $\|(U, \alpha)\|_2^2$ in $\text{poly}\left(\frac{d}{\varepsilon}\right)$ time since $H = \ell_2^{2d}$. If $\|(U, \alpha)\|_2^2 \leq R^2$, then continue. Otherwise, we output the vector $c = \frac{(U,\alpha)}{\|(U,\alpha)\|_2}$, which is a valid output of $\texttt{SEP}(P)$.
2. Second, we may think of U_M as the $2d \times |\mathbb{Q}_M|$ matrix, whose columns are the vectors $v_q \in \mathbb{R}^{2d}$. Thus, the constraints (ii–iv) may be written as $\|U_M\gamma\|_H \leq \lambda$ for some $\gamma \in \mathbb{R}^{|\mathbb{Q}_M|}$ which is a column vector. If none of the constraints (ii–iv) are violated, then we continue. If some constraint is violated, i.e., $\|U_M\gamma\|_H > \lambda$, let $b = U_M\gamma$, so that $\frac{b^\intercal U_M\gamma}{\|b\|_H} > \lambda$, but any $(U', \alpha') \in P$ with $\|U'_M\gamma\|_H \leq \lambda$ has $\frac{b^\intercal U'_M\gamma}{\|b\|_H} \leq \lambda$. Thus, we consider the vector $c = (\frac{\gamma_q b}{\|b\|_H} \in \mathbb{R}^{2d} : q \in \mathbb{Q}_M) \times (0) \in (\mathbb{R}^{2d})^{|\mathbb{Q}_M|} \times \mathbb{R}$, so that $c^\intercal \cdot (U, \alpha) = \frac{b^\intercal U_M\gamma}{\|b\|_H}$. Thus, $\frac{c}{\|c\|_2}$ is a valid output for $\texttt{SEP}(P)$.
3. Finally, we consider constraint (i), which may be written as $\|U_M\gamma\|_W \leq \lambda$ for some $\lambda \in \mathbb{R}$ with $\lambda \in (\frac{\varepsilon}{2}, 2R)$. For each constraint of type (i), we query the oracle $\texttt{SEP}(B_W)$ on the vector $y = \frac{U_M\gamma(1+\delta'\mathsf{d})}{\lambda}$ with error parameter δ' (which we specify later). If $\texttt{SEP}(B_W)$ asserts $y \in B(B_W, \delta')$, then $\|U_M\gamma\|_W \leq \lambda$. So if all oracle calls to $\texttt{SEP}(B_W)$ assert $y \in B(B_W, \delta')$, then since constraints (ii–v) are satisfied as well, we have $(U, \alpha) \in P$, so we may assert $(U, \alpha) \in B(P, \delta)$.

   Otherwise, suppose $\|U_M\gamma\|_W > \lambda$. Then, we have

$$\frac{\lambda}{\mathsf{d}} \leq \|U_M\gamma\|_H \leq \sum_{q \in \mathbb{Q}} |\gamma_q| \|u_q\|_H \leq \|\gamma\|_1 \cdot M \cdot \mathsf{d}.$$

   In this case, $\texttt{SEP}(B_W)$ outputs a unit vector $b \in \mathbb{R}^{2d}$ where $b^\intercal \widetilde{y} \leq b^\intercal y + \delta'$ for all $\widetilde{y} \in B(B_W, -\delta')$. So suppose $(\widetilde{U}, \widetilde{\alpha}) \in P$ and in particular, $\|\widetilde{U}_M\gamma\|_W \leq \lambda$. Then, letting $\widetilde{y} = \frac{\widetilde{U}_M\gamma(1-\delta'\mathsf{d})}{\lambda}$, we have $\widetilde{y} \in B(B_W, -\delta')$. Therefore,

$$b^\intercal \widetilde{U}_M\gamma \leq b^\intercal U_M\gamma + \delta'\lambda + \delta'\mathsf{d}\|b\|_2\|\widetilde{U}_M\gamma\|_H \leq b^\intercal U_M\gamma + 2\delta'\lambda\mathsf{d}.$$

   Similarly to step 2 above, we consider the vector $c = (\gamma_q b : q \in \mathbb{Q}_M) \times (0) \in$

$(\mathbb{R}^{2d})^{|\mathbb{Q}_M|} \times \mathbb{R}$ which satisfies $b^\intercal U_M \gamma = c^\intercal (U, \alpha)$ for all (U, α). Recall that since b is a unit vector in $\mathbb{R}^{2d}$, we have $\|c\|_2 = \|\gamma\|_2 \geq \frac{\|y\|_1}{2d|\mathbb{Q}_M|} \geq \frac{\varepsilon}{4d\mathsf{d}^2 M|\mathbb{Q}_M|}$, which in turn, implies that:

$$\frac{c^\intercal}{\|c\|_2} \cdot (\widetilde{U}, \widetilde{\alpha}) \leq \frac{c^\intercal}{\|c\|_2} \cdot (U, \alpha) + \frac{8d\lambda \mathsf{d}^3 M|\mathbb{Q}_M|}{\varepsilon} \cdot \delta' \leq \frac{c^\intercal}{\|c\|_2} \cdot (U, \alpha) + \delta,$$

when $\delta' = \dfrac{\delta\varepsilon}{8dR\mathsf{d}^3 M|\mathbb{Q}_M|}$.

□

Finally, we will use a reduction from [94], which asserts that one may implement $\texttt{SEP}(B_W)$ with $\texttt{MEM}(B_W)$.

Theorem 52 (Theorem 14 from [94])**.** *Let $K \subset \mathbb{R}^m$ be a convex body satisfying $B(0, r) \subset K \subset B(0, 1)$, and let $\kappa = \frac{1}{r}$. For any $\delta \in (0, 1)$, with probability $1 - \delta$, one can compute $\texttt{SEP}(K)$ with error parameter δ with $O\left(m \log\left(\frac{m\kappa}{\delta}\right)\right)$ calls to $\texttt{MEM}(K)$ with error parameter $\delta' = \mathrm{poly}(\delta, \frac{1}{\kappa}, \frac{1}{m})$ and $\mathrm{poly}(m, \log(\frac{\kappa}{\delta}))$ time.*

Lemma 4.5.23. *There exists an algorithm for $\texttt{SEP}(B_W)$ with parameter δ which makes $\mathrm{poly}\left(\frac{d}{\varepsilon}, \log\left(\frac{1}{\delta}\right)\right)$ calls to $\texttt{MEM}(B_W)$ with parameter $\delta' = \mathrm{poly}(\delta, \frac{\varepsilon}{d})$.*

Proof. We simply use Theorem 14 from [94], where we note that $B_W \subset B_2 \subset \mathsf{d}B_W$, which implies that $B(0, \frac{1}{\mathsf{d}}) \subset B_W \subset B(0, 1)$. □

4.5.5 Summary and instantiation for applications

From the discussion above, we may conclude the following theorem, whose proof simply follows by combining the reductions given in Lemma 4.5.10, Lemma 4.5.20, Lemma 4.5.21, Lemma 4.5.22, and Lemma 4.5.23.

Theorem 53. *There exists an algorithm which solves $\texttt{ApproxRep}(x, \theta, \varepsilon; W)$ with probability at least $1 - \mathrm{poly}(\varepsilon, \frac{1}{d})$ making $\mathrm{poly}(d, \frac{1}{\varepsilon})$ calls to $\texttt{MEM}(W)$ and using $\mathrm{poly}(d, \frac{1}{\varepsilon})$ additional time.*

In order to see how the above algorithm will be applied, recall from our discussion in Subsection 4.3, that in designing algorithms for ANN over X, we will assume oracle access to the real normed space $X = (\mathbb{R}^d, \|\cdot\|_X)$.

Lemma 4.5.24. *Let $\varepsilon < \frac{1}{10}$ and assume access to computing $\|\cdot\|_X$, where $X = (\mathbb{R}^d, \|\cdot\|_X)$ is a real normed space with*

$$B_X \subset B_2 \subset \mathsf{d} \cdot B_X,$$

for $\mathsf{d} = \mathrm{poly}(d)$. Given a vector $x \in \mathbb{C}^d$ with $\|x\|_{X^{\mathbb{C}}}$, one can compute a multiplicative $(1 \pm 4\varepsilon)$-approximation to $\|x\|_{X^{\mathbb{C}}}$ in time $\mathrm{poly}(d/\varepsilon)$.

Proof. Let $x = u + iv \in \mathbb{C}^d$. We note that we may compute $\|x\|_H$ for $H = (\ell_2^d)^{\mathbb{C}}$, so that the vector $y = \frac{x}{\|x\|_H}$ satisfies $1 \leq \|y\|_{X^{\mathbb{C}}} \leq \mathsf{d}$. Thus, we may assume without loss of generality that $1 \leq \|x\|_{X^{\mathbb{C}}} \leq \mathsf{d}$. For a parameter $P > 0$ (which we set briefly to $\mathrm{poly}(d/\varepsilon)$, consider the set:

$$\mathbb{D}_P^{(C)} = \left\{ \frac{k}{P} : 0 \leq k \leq 2\pi P \right\}.$$

Then, we note that by differentiation, we have that for all $\phi \in [0, 2\pi]$, if we let $\widetilde{\phi} \in \mathbb{D}_P^{(C)}$ be the smallest element greater than ϕ, when $P = \frac{2R}{\varepsilon}$,

$$\|u\cos\widetilde{\phi} + v\sin\widetilde{\phi}\|_X - \varepsilon \leq \|u\cos\phi + v\sin\phi\|_X \leq \|u\cos\widetilde{\phi} + v\sin\widetilde{\phi}\|_X + \varepsilon.$$

The value $\ell_{X^{\mathbb{C}}}(x) = \frac{1}{\pi \cdot |\mathbb{D}_P^{(C)}|} \sum_{\widetilde{\phi} \in \mathbb{D}_P^{(C)}} \|u\cos\widetilde{\phi} + v\sin\widetilde{\phi}\|_X^2$ may be computed in $\mathrm{poly}(\frac{d}{\varepsilon}, R)$ time with access to $\|\cdot\|_X$, and we have:

$$(1 - 2\varepsilon)\ell_{X^{\mathbb{C}}}(x) \leq \frac{1}{\pi}\int_0^{2\pi} \|u\cos\phi + v\sin\phi\|_X^2 d\phi \leq (1 + 2\varepsilon)\ell_{X^{\mathbb{C}}}(x) + \varepsilon^2.$$

Therefore, we have $(1 - 2\varepsilon)\ell_{X^{\mathbb{C}}}(x) \leq \|x\|_{X^{\mathbb{C}}}$ and $\|x\|_{X^{\mathbb{C}}} \leq (1 + 2\varepsilon)\ell_{X^{\mathbb{C}}}(x) + \varepsilon\|x\|_{X^{\mathbb{C}}}$, so $\|x\|_{X^{\mathbb{C}}} \leq (1 + 4\varepsilon)\ell_{X^{\mathbb{C}}}(x)$. □

Given Lemma 4.5.24, we may now design a membership oracle for $B_{X^{\mathbb{C}}}$.

Lemma 4.5.25. *Assume oracle access to a real normed space $X = (\mathbb{R}^d, \|\cdot\|_X)$ with*

$$B_X \subset B_2 \subset \mathsf{d} \cdot B_X.$$

Let $B_{X^{\mathbb{C}}} \subset \mathbb{R}^{2d}$ be the convex set given by the unit ball $B_{X^{\mathbb{C}}} \subset \mathbb{C}^d$. There exists an membership oracle $\texttt{MEM}(B_{X^{\mathbb{C}}})$ running in time $\mathrm{poly}(d, \frac{1}{\delta})$.

Proof. Given a vector $y \in \mathbb{R}^{2d}$, first compute $\|y\|_2$, and if $\|y\|_2 > 1$, assert that $y \notin B(B_{X^{\mathbb{C}}}, -\delta)$ since $y \notin B_2$ and $B_{X^{\mathbb{C}}} \subset B_2$. We interpret the vector $y \in \mathbb{C}^d$ and compute $\ell_{X^{\mathbb{C}}}(y) \in \mathbb{R}^{\geq 0}$ satisfying

$$\|y\|_{X^{\mathbb{C}}} \leq \ell_{X^{\mathbb{C}}}(y) \leq \left(1 + \frac{\delta}{2\|y\|_2}\right) \|y\|_{X^{\mathbb{C}}}. \tag{4.34}$$

Note that since $\|y\|_2 \leq 1$, the computing $\ell_{X^{\mathbb{C}}}(y)$ takes $\mathrm{poly}(d, \frac{1}{\delta})$ time by Lemma 4.5.24. If $\ell_{X^{\mathbb{C}}}(y) \leq 1$, then by (4.34), $\|y\|_{X^{\mathbb{C}}} \leq 1$, i.e, $y \in B_{X^{\mathbb{C}}}$. This means we may safely assert that $y \in B(X^{\mathbb{C}}, \delta)$. On the other hand, if $\ell_{X^{\mathbb{C}}}(y) > 1$, then by (4.34), $\|y\|_{X^{\mathbb{C}}} > \frac{1}{1+\frac{\delta}{2\|y\|_2}}$. Therefore, $\|y + \frac{\delta y}{\|y\|_2}\|_{X^{\mathbb{C}}} = (1 + \frac{\delta}{\|y\|_2})\|y\|_{X^{\mathbb{C}}} > 1$, which implies that $B(y, \delta) \notin B_{X^{\mathbb{C}}}$, i.e., we may safely assert $y \notin B(B_{X^{\mathbb{C}}}, -\delta)$. □

Looking ahead to Section **??** and Section **??**, as well as the discussion from Section **??**, starting from oracle access to the normed space $X = (\mathbb{R}^d, \|\cdot\|_X)$, we will consider the normed spaces

$$A = [X^{\mathbb{C}}, H]_\alpha \qquad \text{and} \qquad Y = [A, H]_\beta,$$

for some values of $\alpha, \beta \in (0, 1)$ with $\frac{1}{d} \leq \alpha, \beta \leq 1 - \frac{1}{d}$ (which implies $\Lambda(\alpha, \alpha)$ and $\Lambda(\beta, \beta)$ are at most $\mathrm{poly}(d)$). During the course of the algorithms in Section **??** and Section **??**, we will need to compute $\texttt{ApproxRep}(x, \beta, \varepsilon; A)$, which is needed for computing norms $\|x\|_Y$ as well as approximate representatives in $\mathcal{F}$ defined by the couple $[A, H]_\theta$ in the definition of Section **??**.

From Theorem 53, we consider the setting with $W = A$, so it suffices to construct an oracle $\texttt{MEM}(B_A)$ which runs in $\mathrm{poly}(d, \frac{1}{\delta})$ time. Note that we do not have oracle access to $\|\cdot\|_A$ (which would give $\texttt{MEM}(B_A)$). However, by Corollary 4.5.1, we may design $\texttt{MEM}(B_A)$

running in $\mathrm{poly}(d, \frac{1}{\delta})$ time by solving $\texttt{ApproxRep}(x, \alpha, \delta; X^{\mathbb{C}})$ again. Thus, we construct $\texttt{MEM}(B_A)$ by using Theorem 53, this time with $W = X^{\mathbb{C}}$, to solve $\texttt{ApproxRep}(x, \alpha, \delta; X^{\mathbb{C}})$ using $\mathrm{poly}(d, \frac{1}{\delta})$ time and oracle calls to $\texttt{MEM}(X^{\mathbb{C}})$. Finally, Lemma 4.5.25 shows how to solve $\texttt{MEM}(X^{\mathbb{C}})$ in $\mathrm{poly}(d, \frac{1}{\delta})$ time from oracle access to $\|\cdot\|_X$. We thus conclude the discussion below into the following corollary.

Corollary 4.5.26. *Assume oracle access to a real normed space $X = (\mathbb{R}^d, \|\cdot\|_X)$ with*

$$B_X \subset B_2 \subset \mathsf{d} \cdot B_X,$$

where $\mathsf{d} = \mathrm{poly}(d)$. Then, letting $H = (\ell_2^d)^{\mathbb{C}}$,

$$A = [X^{\mathbb{C}}, H] \qquad \textit{and} \qquad Y = [A, H],$$

there exists a $\mathrm{poly}(d/\varepsilon)$ time algorithm which for each $x \in \mathbb{C}^d$, computes a $(1 \pm \varepsilon)$-approximation to $\|x\|_Y$ with probability at least $1 - \varepsilon$.

Part II

Property Testing of Boolean Functions

Chapter 5

A Lower Bound for Non-Adaptive Junta Testing

The main result of this chapter is the following theorem:

Theorem 54 (Lower Bound for Testing k-Juntas Non-adaptively [55])**.** *Let $\alpha \in (0.5, 1)$ be an absolute constant. Let $k = k(n)\colon \mathbb{N} \to \mathbb{N}$ and $\varepsilon = \varepsilon(n)\colon \mathbb{N} \to \mathbb{R}_{>0}$ be two functions that satisfy $k(n) \leq \alpha n$ and $2^{-n} \leq \varepsilon(n) \leq 1/6$ for all sufficiently large n. Then any non-adaptive ε-tester for k-juntas must make $\widetilde{\Omega}(k^{3/2}/\varepsilon)$ many queries.*[1]

Together with the $\widetilde{O}(k^{3/2})/\varepsilon$ non-adaptive upper bound from [32], Theorem 54 settles the query complexity of non-adaptive junta testing up to poly-logarithmic factors.

5.1 High-level overview of our approach

Below we provide a high level overview of the proof of the lower bound given by Theorem 54. First, it is not difficult to show that Theorem 54 is a consequence of the following more specific lower bound for the case where $k = \alpha n$:[2]

Theorem 55. *Let $\alpha \in (0.5, 1)$ be an absolute constant. Let $k = k(n)\colon \mathbb{N} \to \mathbb{N}$ and $\varepsilon = \varepsilon(n)\colon \mathbb{N} \to \mathbb{R}_{>0}$ be two functions that satisfy $k(n) = \alpha n$ and $2^{-(2\alpha-1)n/2} \leq \varepsilon(n) \leq 1/6$ for sufficiently large n. Then any non-adaptive ε-tester for k-juntas must make $\widetilde{\Omega}(n^{3/2}/\varepsilon)$ many queries.*

We now provide a sketch of how Theorem 65 is proved. It may be convenient for the reader, on the first reading, to consider $\alpha = 3/4$ and to think of ε as being a small constant such as 0.01.

[1] The precise lower bound is $\Omega\left(\frac{k^{3/2}}{\varepsilon \log^3(k) \log^3(k/\varepsilon)}\right)$.

[2] See Section 5.6 for the proof that Theorem 65 implies Theorem 54.

Fix a sufficiently large n. Let $k = \alpha n$ and $\varepsilon = \varepsilon(n)$ with ε satisfying the condition in Theorem 65.

This lower bound proof consists of two components:

1. A reduction from a simple algorithmic task called Set-Size-Set-Queries (SSSQ for short), which we discuss informally later in this subsection and we define formally in Section 5.3. This reduction implies that the non-adaptive deterministic query complexity of distinguishing $\mathcal{D}_{\text{yes}}$ and $\mathcal{D}_{\text{no}}$ is at least as large as that of SSSQ.
2. A lower bound of $\widetilde{\Omega}(n^{3/2}/\varepsilon)$ for the query complexity of SSSQ.

Our yes-functions and no-functions have very similar structure to each other, but are constructed with slightly different parameter settings. The first step in drawing a random partition by choosing a uniform random subset $\mathbf{M}$ of $\Theta(n)$ "addressing" variables from $x_1, \ldots, x_n$. A random subset $\mathbf{A}$ of the complementary variables $\overline{\mathbf{M}}$ is also selected, and for each assignment to the variables in $\mathbf{M}$ (let us denote such an assignment by i), there is an independent random function $\boldsymbol{h}_i$ over a randomly selected subset $\mathbf{S}_i$ of the variables in $\mathbf{A}$. A random function from $\mathcal{D}_{\text{no}}$ is constructed in the same way, except that now the random subset $\mathbf{A}$ is chosen to be slightly larger than in the yes-case. This disparity in the size of $\mathbf{A}$ between the two cases causes random functions from $\mathcal{D}_{\text{yes}}$ to almost always be k-juntas and random functions from $\mathcal{D}_{\text{no}}$ to almost always be far from k-juntas.

An intuitive explanation of why this construction is amenable to a lower bound for non-adaptive algorithms is as follows. Intuitively, for an algorithm to determine that it is interacting with (say) a random no-function rather than a random yes-function, it must determine that the subset $\mathbf{A}$ is larger than it should be in the yes-case. Since the set $\mathbf{M}$ of $\Theta(n)$ many "addressing" variables is selected randomly, if a non-adaptive algorithm uses two query strings x, x' that differ in more than a few coordinates, it is very likely that the random set $\mathbf{M}$ will contain a variable where x and x' differ, and therefore, x and x' will correspond to two different random functions $\boldsymbol{h}_i, \boldsymbol{h}_{i'}$. Hence, every pair of query strings x, x' that correspond to the same $\boldsymbol{h}_i$ can differ only in a few coordinates with high probability. This phenomenon significantly limits the power of a non-adaptive algorithm distinguishing

$\mathcal{D}_{\text{yes}}$ and $\mathcal{D}_{\text{no}}$, and allows us to reduce from the algorithmic task SSSQ at the price of only a small quantitative cost in query complexity, see Section 5.4.

At a high level, the SSSQ task involves distinguishing whether or not a hidden set (corresponding to $\mathbf{A}$) is "large." An algorithm for this task can only access certain random bits, whose biases are determined by the hidden set and whose exact distribution is inspired by the exact definition of the random functions $\boldsymbol{h}_i$ over the random subsets $\mathbf{S}_i$. The SSSQ problem is much easier to work with compared to the original problem of distinguishing $\mathcal{D}_{\text{yes}}$ and $\mathcal{D}_{\text{no}}$. In particular, we give a reduction from an even simpler algorithmic task called Set-Size-Element-Queries (SSEQ for short) to SSSQ (see Section 5.5.1) and the query complexity lower bound for SSSQ follows directly from the lower bound for SSEQ presented in Section 5.5.2. We hope that the SSSQ problem and/or the SSEQ problem may find other applications in lower bounds for query algorithms.

Let us give a high-level description of the SSEQ task to provide some intuition for how we prove a query lower bound on it. Roughly speaking, in this task an oracle holds an unknown and random subset $\mathbf{A}$ of $[m]$ (here $m = \Theta(n)$) which is either "small" (size roughly $m/2$) or "large" (size roughly $m/2 + \Theta(\sqrt{n} \cdot \log n)$), and the task is to determine whether $\mathbf{A}$ is small or large. The algorithm may repeatedly query the oracle by providing it, at the j-th query, with an element $i_j \in [m]$; if $i_j \notin \mathbf{A}$ then the oracle responds "0" with probability 1, and if $i_j \in \mathbf{A}$ then the oracle responds "1" with probability $\varepsilon/\sqrt{n}$ and "0" otherwise. Intuitively, the only way for an algorithm to determine that the unknown set $\mathbf{A}$ is (say) large, is to determine that the fraction of elements of $[m]$ that belong to $\mathbf{A}$ is $1/2 + \Theta(\log n/\sqrt{n})$ rather than $1/2$; this in turn intuitively requires sampling $\Omega(n/\log^2 n)$ many random elements of $[m]$ and for each one ascertaining with high confidence whether or not it belongs to $\mathbf{A}$. But the nature of the oracle access described above for SSEQ is such that for any given $i \in [m]$, at least $\Omega(\sqrt{n}/\varepsilon)$ many repeated queries to the oracle on input i are required in order to reach even a modest level of confidence as to whether or not $i \in \mathbf{A}$. As alluded to earlier, the formal argument establishing our lower bound on the query complexity of SSEQ relies on an upper bound on the total variation distance between two Binomial distributions.

5.2 The $\mathcal{D}_{\text{yes}}$ and $\mathcal{D}_{\text{no}}$ distributions

Let $\alpha \in (0.5, 1)$ be an absolute constant. Let n be a sufficiently large integer, with $k = \alpha n$, and let ε be the distance parameter that satisfies

$$2^{-(2\alpha-1)n/2} \leq \varepsilon \leq 1/6. \tag{5.1}$$

In this section we describe a pair of probability distributions $\mathcal{D}_{\text{yes}}$ and $\mathcal{D}_{\text{no}}$ supported over Boolean functions $f\colon \{0,1\}^n \to \{0,1\}$. We then show that $\boldsymbol{f} \leftarrow \mathcal{D}_{\text{yes}}$ is a k-junta with probability $1 - o(1)$, and that $\boldsymbol{f} \leftarrow \mathcal{D}_{\text{no}}$ is ε-far from being a k-junta with probability $1 - o(1)$.

We start with some parameters settings. Define

$$\delta \stackrel{\text{def}}{=} 1 - \alpha \in (0, 0.5), \quad p \stackrel{\text{def}}{=} \frac{1}{2}, \qquad\qquad p_{\text{no}} \stackrel{\text{def}}{=} \frac{1}{2} + \frac{\log n}{\sqrt{n}},$$
$$m \stackrel{\text{def}}{=} 2\delta n + \delta\sqrt{n}\log n, \quad t \stackrel{\text{def}}{=} n - m = (2\alpha - 1)n - \delta\sqrt{n}\log n, \quad N \stackrel{\text{def}}{=} 2^t.$$

A function $\boldsymbol{f} \leftarrow \mathcal{D}_{\text{yes}}$ is drawn according to the following randomized procedure:

1. Sample a random subset $\mathbf{M} \subset [n]$ of size t. Let $\mathbf{\Gamma} = \Gamma_{\mathbf{M}} : \{0,1\}^n \to [N]$ be the function that maps $x \in \{0,1\}^n$ to the integer encoded by $x_{|\mathbf{M}}$ in binary plus one. Note that $|\overline{\mathbf{M}}| = n - t = m$.
2. Sample an $\mathbf{A} \subseteq \overline{\mathbf{M}}$ by including each element of $\overline{\mathbf{M}}$ in $\mathbf{A}$ independently with probability p.
3. Sample independently a sequence of N random subsets $\mathbf{S} = (\mathbf{S}_i \colon i \in [N])$ of $\mathbf{A}$ as follows: for each $i \in [N]$, each element of $\mathbf{A}$ is included in $\mathbf{S}_i$ independently with probability $\varepsilon/\sqrt{n}$. Next we sample a sequence of N functions $\mathbf{H} = (\boldsymbol{h}_i \colon i \in [N])$, by letting $\boldsymbol{h}_i \colon \{0,1\}^n \to \{0,1\}$ be a random function over the coordinates in $\mathbf{S}_i$, i.e., we sample an unbiased bit $\boldsymbol{z}_i(b)$ for each string $b \in \{0,1\}^{\mathbf{S}_i}$ independently and set $\boldsymbol{h}_i(x) = \boldsymbol{z}_i(x_{|\mathbf{S}_i})$.
4. Finally $\boldsymbol{f} = \boldsymbol{f}_{\mathbf{M},\mathbf{A},\mathbf{H}} \colon \{0,1\}^n \to \{0,1\}$ is defined using $\mathbf{M}$, $\mathbf{A}$ and $\mathbf{H}$ as follows

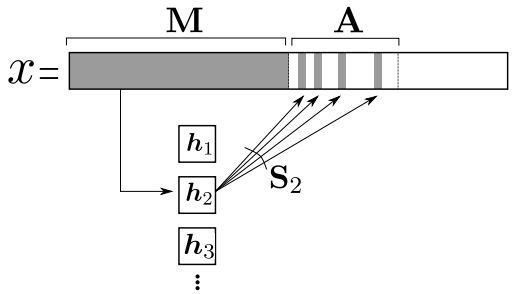

Figure 5.1: An example of how an input $x \in \{0,1\}^n$ is evaluated by $\boldsymbol{f} \sim \mathcal{D}_{\text{yes}}$ (or $\mathcal{D}_{\text{no}}$). The relevant variables of x are shaded gray. The output $\boldsymbol{f}(x)$ is computed in two steps. First, the input x is indexed into one of N functions $\boldsymbol{h}_1, \ldots, \boldsymbol{h}_N$ according to $\boldsymbol{\Gamma}_{\mathbf{M}}(x) = x_{|\mathbf{M}} + 1$. Second, letting $i = \boldsymbol{\Gamma}_{\mathbf{M}}(x)$, the output $\boldsymbol{f}(x)$ is equal to $\boldsymbol{h}_i(x)$, which depends on the values of $x_{|\mathbf{S}_i}$ for a subset $\mathbf{S}_i \subset \mathbf{A}$.

(note that we can skip $\mathbf{S}$ since the choice of $\mathbf{S}$ is included in the choice of $\mathbf{H}$):

$$\boldsymbol{f}(x) = \boldsymbol{h}_{\boldsymbol{\Gamma}_{\mathbf{M}}(x)}(x), \quad \text{for each } x \in \{0,1\}^n.$$

In words, an input x is assigned the value $\boldsymbol{f}(x)$ as follows: according to the coordinates of x in the set $\mathbf{M}$ (which intuitively should be thought of as unknown), one of the N functions $\boldsymbol{h}_i$ (each of which is, intuitively, a random function over an unknown subset $\mathbf{S}_i$ of coordinates) is selected and evaluated on x's coordinates in $\mathbf{S}_i$. For intuition, we note that both $\mathbf{M}$ and $\overline{\mathbf{M}}$ will always be of size $\Theta(n)$, the size of $\mathbf{A}$ will almost always be $\Theta(n)$, and for a given $i \in [N]$ the expected size of $\mathbf{S}_i$ will typically be $\Theta(\varepsilon\sqrt{n})$ (though the size of $\mathbf{S}_i$ may not be as highly concentrated as the other sets when ε is tiny).

A function $\boldsymbol{f} \leftarrow \mathcal{D}_{\text{no}}$ is generated using the same procedure except that $\mathbf{A}$ is a random subset of $\overline{\mathbf{M}}$ drawn by including each element of $\overline{\mathbf{M}}$ in $\mathbf{A}$ independently with probability p_{no} (instead of p). See Figure 5.1 for an example of how an input $x \in \{0,1\}^n$ is evaluated by $\boldsymbol{f} \sim \mathcal{D}_{\text{yes}}$ or $\mathcal{D}_{\text{no}}$.

5.2.1 Most functions drawn from $\mathcal{D}_{\text{yes}}$ are k-juntas

We first prove that $\boldsymbol{f} \leftarrow \mathcal{D}_{\text{yes}}$ is a k-junta with probability $1 - o(1)$.

Lemma 5.2.1. *A function $\boldsymbol{f} \leftarrow \mathcal{D}_{yes}$ is a k-junta with probability $1 - o(1)$.*

Proof. By the definition of $\mathcal{D}_{\text{yes}}$, all the relevant variables of $\boldsymbol{f} \sim \mathcal{D}_{\text{yes}}$ belong to $\mathbf{M} \cup \mathbf{A}$. Note that $|\mathbf{M}| = t$. On the other hand, the expected size of $\mathbf{A}$ is $\delta n + \delta\sqrt{n}\log n/2$. By a Chernoff bound, we have

$$|\mathbf{A}| \leq \delta n + \frac{\delta\sqrt{n}\log n}{2} + \frac{\delta\sqrt{n}\log n}{4} < \delta n + \delta\sqrt{n}\log n$$

with probability $1 - o(1)$. When this happens we have $|\mathbf{M} \cup \mathbf{A}| < \alpha n = k$. □

5.2.2 Most functions drawn from $\mathcal{D}_{\text{no}}$ are ε-far from k-juntas

Next we prove that $\boldsymbol{f} \leftarrow \mathcal{D}_{\text{no}}$ is ε-far from any k-junta with probability $1 - o(1)$. The details of the argument are somewhat technical so we start by giving some high-level intuition, which is relatively simple. Since $p_{\text{no}} = p + \log(n)/\sqrt{n}$, a typical outcome of $\mathbf{A}$ drawn from $\mathcal{D}_{\text{no}}$ is slightly larger than a typical outcome drawn from $\mathcal{D}_{\text{yes}}$, and this difference causes almost every outcome of $|\mathbf{M} \cup \mathbf{A}|$ in $\mathcal{D}_{\text{no}}$ (with $\mathbf{M} \cup \mathbf{A}$ being the set of relevant variables for $\boldsymbol{f} \leftarrow \mathcal{D}_{\text{no}}$) to be larger than k by at least $9\sqrt{n}$. As a result, the relevant variables of any k-junta must miss either (a) at least one variable from $\mathbf{M}$, or (b) at least $9\sqrt{n}$ variables from $\mathbf{A}$. Missing even a single variable from $\mathbf{M}$ causes the k-junta to be far from $\boldsymbol{f}$ (this is made precise in Claim 5.2.4 below). On the other hand, missing $9\sqrt{n}$ variables from $\mathbf{A}$ means that with probability $\Omega(\varepsilon)$, at least one variable is missing from a typical $\mathbf{S}_i$ (recall that these are random $(\varepsilon/\sqrt{n})$-dense subsets of $\mathbf{A}$). Because $\boldsymbol{h}_i$ is a random function over the variables in $\mathbf{S}_i$, missing even a single variable would lead to a constant fraction of error when $\boldsymbol{h}_i$ is the function determining the output of $\boldsymbol{f}$.

Lemma 5.2.2. *A function $\boldsymbol{f} \leftarrow \mathcal{D}_{no}$ is ε-far from being a k-junta with probability $1 - o(1)$.*

Proof. Fix any subset $M \subset [n]$ of size t, and consider $\boldsymbol{f} = \boldsymbol{f}_{M,\mathbf{A},\mathbf{H}}$ where $\mathbf{A}$ and $\mathbf{H}$ are sampled according to the procedure for $\mathcal{D}_{\text{no}}$. With probability $1 - o(1)$ over the choice of $\mathbf{A}$, we have

$$|\mathbf{A}| \geq p_{\text{no}} m - \frac{\delta\sqrt{n}\log n}{2} \geq \delta n + 2\delta\sqrt{n}\log n \quad \text{and} \quad |\mathbf{M} \cup \mathbf{A}| \geq k + \delta\sqrt{n}\log n. \tag{5.2}$$

Assume this is the case for the rest of the proof and fix any such set $A \subset \overline{M}$. It suffices to show that $\boldsymbol{f} = \boldsymbol{f}_{M,A,\mathbf{H}}$ is ε-far from any k-junta with probability $1 - o(1)$, where $\mathbf{H}$ is sampled according to the rest (steps 3 and 4) of the procedure for $\mathcal{D}_{\text{no}}$ (by sampling $\mathbf{S}_i$ from A and then $\boldsymbol{h}_i$ over $\mathbf{S}_i$).

The plan for the rest of the proof is the following. For each $V \subset M \cup A$ of size $9\sqrt{n}$, we use $\mathbf{E}_V$ to denote the size of the *maximum* set of vertex-disjoint, $\boldsymbol{f}$-bichromatic edges along directions in V only. We will prove the following claim:

Claim 5.2.3. *For each $V \subset M \cup A$ of size $9\sqrt{n}$, we have $\mathbf{E}_V \geq \varepsilon 2^n$ with probability $1 - \exp(-2^{\Omega(n)})$ over the choice of $\mathbf{H}$.*

Note that when $\mathbf{E}_V \geq \varepsilon 2^n$, we have $\operatorname{dist}(\boldsymbol{f}, g) \geq \varepsilon$ for every function g that does not depend on any variable in V. This is because, for every $\boldsymbol{f}$-bichromatic edge $(x, x^{(\ell)})$ along a coordinate $\ell \in V$, we must have $\boldsymbol{f}(x) \neq \boldsymbol{f}(x^{(\ell)})$ since the edge is bichromatic but $g(x) = g(x^{(\ell)})$ as g does not depend on the ℓth variable. As a result, $\boldsymbol{f}$ must disagree with g on at least $\varepsilon 2^n$ many points.

Assuming Claim 5.2.3 for now, we can apply a union bound over all

$$\binom{|M \cup A|}{9\sqrt{n}} \leq \binom{n}{9\sqrt{n}} \leq 2^{O(\sqrt{n}\log n)}$$

possible choices of $V \subset M \cup A$ to conclude that with probability $1 - o(1)$, $\boldsymbol{f} = \boldsymbol{f}_{M,A,\mathbf{H}}$ is ε-far from all functions that do not depend on at least $9\sqrt{n}$ variables in $M \cup A$. By (5.2), this set includes all k-juntas. This concludes the proof of the Lemma 5.2.2 modulo the proof of Claim 5.2.3. □

In the rest of the section, we prove Claim 5.2.3 for a fixed subset $V \subset M \cup A$ of size $9\sqrt{n}$. We start with the simpler case when $V \cap M$ is nonempty.

Claim 5.2.4. *If $V \cap M \neq \emptyset$, then $\mathbf{E}_V \geq 2^n/5$ with probability $1 - \exp(-2^{\Omega(n)})$ over the choice of $\mathbf{H}$.*

Proof. Fix an $\ell \in V \cap M$; we will argue that with probability $1 - \exp(-2^{\Omega(n)})$ there are at least $2^n/5$ $\boldsymbol{f}$-bichromatic edges along direction ℓ. This suffices since such edges are clearly vertex-disjoint.

Observe that since $\ell \in M$, every $x \in \{0,1\}^n$ has $\Gamma(x) \neq \Gamma(x^{(\ell)})$. For each $b \in \{0,1\}^M$, let X_b be the set of $x \in \{0,1\}^n$ with $x_{|M} = b$. We partition $\{0,1\}^n$ into 2^{t-1} pairs X_b and $X_{b^{(\ell)}}$, where b ranges over the 2^{t-1} strings in $\{0,1\}^M$ with $b_\ell = 0$. For each such pair, we use $\mathbf{D}_b$ to denote the number of $\boldsymbol{f}$-bichromatic edges between X_b and $X_{b^{(\ell)}}$. We are interested in lower bounding $\sum_b \mathbf{D}_b$.

We will apply Hoeffding's inequality. For this purpose we note that the $\mathbf{D}_b$'s are independent (since they depend on distinct $\boldsymbol{h}_i$'s), always lie between 0 and 2^m, and each one has expectation 2^{m-1}. The latter is because each edge $(x, x^{(\ell)})$ has $\boldsymbol{f}(x)$ and $\boldsymbol{f}(x^{(\ell)})$ drawn as two independent random bits, which is the case since $\Gamma(x) \neq \Gamma(x^{(\ell)})$. Thus, the expectation of $\sum_b \mathbf{D}_b$ is 2^{n-2}. By Hoeffding's inequality, we have

$$\mathbf{Pr}\left[\left|\sum \mathbf{D}_b - 2^{n-2}\right| \geq \frac{2^n}{20}\right] \leq 2 \cdot \exp\left(-\frac{2(2^n/20)^2}{2^{t-1} \cdot 2^{2m}}\right) = \exp\left(-2^{\Omega(n)}\right)$$

since $t = \Omega(n)$. This finishes the proof of the claim. □

Now we may assume that $V \subset A$ (and $|V| = 9\sqrt{n}$). We use $\mathbf{I}$ to denote the set of $i \in [N]$ such that $\mathbf{S}_i \cap V \neq \emptyset$. The following claim shows that $\mathbf{I}$ is large with extremely high probability:

Claim 5.2.5. *We have $|\mathbf{I}| \geq 4.4\varepsilon N$ with probability at least $1 - \exp(-2^{\Omega(n)})$ over the choice of $\mathbf{S}$.*

Proof. For each $i \in [N]$ we have (using $1 - x \leq e^{-x}$ for all x and $1 - x/2 \geq e^{-x}$ for $x \in [0, 1.5]$):

$$\mathbf{Pr}\left[i \in \mathbf{I}\right] = 1 - \left(1 - \frac{\varepsilon}{\sqrt{n}}\right)^{9\sqrt{n}} \geq 1 - e^{-9\varepsilon} \geq 4.5\varepsilon,$$

since $\varepsilon/\sqrt{n}$ is the probability of each element of A being included in $\mathbf{S}_i$ and $\varepsilon \leq 1/6$ so $9\varepsilon \leq 1.5$.

Using $\varepsilon \geq 2^{-(2\alpha-1)n/2}$ from (5.1), we have $\mathbf{E}[|\mathbf{I}|] \geq 4.5\varepsilon N = 2^{\Omega(n)}$. Since the $\mathbf{S}_i$'s are independent, a Chernoff bound implies that $|\mathbf{I}| \geq 4.4\varepsilon N$ with probability $1 - \exp(-2^{\Omega(n)})$. □

By Claim 5.2.5, we fix $S_1, \ldots, S_N$ to be any sequence of subsets of A that satisfy $|I| \geq 4.4\varepsilon N$ in the rest of the proof, and it suffices to show that over the random choices of $\boldsymbol{h}_1, \ldots, \boldsymbol{h}_N$ (where each $\boldsymbol{h}_i$ is chosen to be a random function over S_i), $\mathbf{E}_V \geq \varepsilon 2^n$ with probability at least $1 - \exp(-2^{\Omega(n)})$.

To this end we use $\rho(i)$ for each $i \in I$ to denote the first coordinate of S_i in V, and Z_i to denote the set of $x \in \{0,1\}^n$ with $\Gamma(x) = i$. Note that the Z_i's are disjoint. We further partition each Z_i into disjoint $Z_{i,b}$, $b \in \{0,1\}^{S_i}$, with $x \in Z_{i,b}$ iff $x \in Z_i$ and $x_{|S_i} = b$. For each $i \in I$ and $b \in \{0,1\}^{S_i}$ with $b_{\rho(i)} = 0$, we use $\mathbf{D}_{i,b}$ to denote the number of $\boldsymbol{f}$-bichromatic edges between $Z_{i,b}$ and $Z_{i,b^{(\rho(i))}}$ along the $\rho(i)$th direction. It is clear that such edges, over all i and b, are vertex-disjoint and thus,

$$\mathbf{E}_V \geq \sum_{i \in I} \sum_{\substack{b \in \{0,1\}^{S_i} \\ b_{\rho(i)}=0}} \mathbf{D}_{i,b}. \tag{5.3}$$

We will apply Hoeffding's inequality. Note that $\mathbf{D}_{i,b}$ is $2^{m-|S_i|}$ with probability $1/2$, and 0 with probability $1/2$. Thus, the expectation of the RHS of (5.3) is

$$\sum_{i \in I} 2^{|S_i|-1} \cdot 2^{m-|S_i|-1} = |I| \cdot 2^{m-2} \geq 1.1\varepsilon 2^n,$$

using $|I| \geq 4.4\varepsilon N$. Since all the $\mathbf{D}_{i,b}$'s are independent, by Hoeffding's inequality we have

$$\begin{aligned}\mathbf{Pr}\left[\left|\text{RHS of (5.3)} - |I| \cdot 2^{m-2}\right| \geq 0.01|I| \cdot 2^{m-2}\right] &\leq 2 \cdot \exp\left(-\frac{2(0.01|I| \cdot 2^{m-2})^2}{\sum_{i \in I} 2^{|S_i|-1} \cdot 2^{2(m-|S_i|)}}\right) \\ &\leq \exp\left(-2^{\Omega(n)}\right),\end{aligned}$$

since $|I| \geq \Omega(\varepsilon N) = 2^{\Omega(n)}$. Therefore, with probability $1 - \exp\left(2^{-\Omega(n)}\right)$, we have

$$\mathbf{E}_V \geq 0.99 \cdot |I| \cdot 2^{m-2} > \varepsilon 2^n.$$

This concludes the proof of Claim 5.2.3. □

5.3 The Set-Size-Set-Queries (SSSQ) Problem

We first introduce the Set-Size-Set-Queries (SSSQ for short) problem, which is an artificial problem that we use as a bridge to prove Theorem 65. We use the same parameters p, p_{no} and m from the definition of $\mathcal{D}_{\text{yes}}$ and $\mathcal{D}_{\text{no}}$, with n being sufficiently large (so $m = \Omega(n)$ is also sufficiently large).

We start by defining $\mathcal{A}_{\text{yes}}$ and $\mathcal{A}_{\text{no}}$, two distributions over subsets of $[m]$: $\mathbf{A} \sim \mathcal{A}_{\text{yes}}$ is drawn by independently including each element of $[m]$ with probability p and $\mathbf{A} \sim \mathcal{A}_{\text{no}}$ is drawn by independently including each element with probability q. In SSSQ, the algorithm needs to determine whether an unknown $A \subseteq [m]$ is drawn from $\mathcal{A}_{\text{yes}}$ or $\mathcal{A}_{\text{no}}$. (For intuition, to see that this task is reasonable, we observe here that a straightforward Chernoff bound shows that almost every outcome of $\mathbf{A} \sim \mathcal{A}_{\text{yes}}$ is larger than almost every outcome of $\mathbf{A} \sim \mathcal{A}_{\text{no}}$ by $\Omega(\sqrt{n} \log n)$.)

Let A be a subset of $[m]$ which is hidden in an oracle. An algorithm accesses A (in order to tell whether it is drawn from $\mathcal{A}_{\text{yes}}$ or $\mathcal{A}_{\text{no}}$) by interacting with the oracle in the following way: each time it calls the oracle, it does so by sending a subset of $[m]$ to the oracle. The oracle responds as follows: for each j in the subset, it returns a bit that is 0 if $j \notin A$, and is 1 with probability $\varepsilon/\sqrt{n}$ and 0 with probability $1 - \varepsilon/\sqrt{n}$ if $j \in A$. The cost of such an oracle call is the size of the subset provided to the oracle.

More formally, a deterministic and non-adaptive algorithm $\text{ALG} = (g, T)$ for SSSQ accesses the set A hidden in the oracle by submitting a list of queries $T = (T_1, \ldots, T_d)$, for some $d \geq 1$, where each $T_i \subseteq [m]$ is a set. (Thus, we call each T_i a *set query*, as part of the name SSSQ.)

- Given T, the oracle returns a list of random vectors $\boldsymbol{v} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_d)$, where $\boldsymbol{v}_i \in \{0,1\}^{T_i}$ and each bit $\boldsymbol{v}_{i,j}$ is independently distributed as follows: if $j \notin A$ then $\boldsymbol{v}_{i,j} = 0$; if $j \in A$ then

$$\boldsymbol{v}_{i,j} = \begin{cases} 1 & \text{with probability } \varepsilon/\sqrt{n} \\ 0 & \text{with probability } 1 - (\varepsilon/\sqrt{n}). \end{cases} \tag{5.4}$$

Note that the random vectors in $\boldsymbol{v}$ depend on both T and A.

- Given $\boldsymbol{v} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_d)$, ALG returns (deterministically) the value of $g(\boldsymbol{v}) \in \{\text{“yes”}, \text{“no”}\}$.

The performance of $\text{ALG} = (g, T)$ is measured by its *query complexity* and its *advantage*.

- The query complexity of ALG is defined as $\sum_{i=1}^{d} |T_i|$, the total size of all the set queries. On the other hand, the advantage of ALG is defined as

$$\Pr_{\mathbf{A} \sim \mathcal{A}_{\text{yes}}} \Big[\text{ALG}(\mathbf{A}) = \text{“yes”}\Big] - \Pr_{\mathbf{A} \sim \mathcal{A}_{\text{no}}} \Big[\text{ALG}(\mathbf{A}) = \text{“yes”}\Big].$$

Remark 56. *In the definition above, g is a deterministic map from all possible sequences of vectors returned by the oracle to “yes” or “no.” Considering only deterministic as opposed to randomized g is without loss of generality since given any query sequence T, the highest possible advantage can always be achieved by a deterministic map g.*

We prove the following lower bound for any deterministic, non-adaptive ALG in Section 5.5.

Lemma 5.3.1. *Any deterministic, non-adaptive* ALG *for* SSSQ *with advantage at least* $2/3$ *satisfies*

$$\sum_{i=1}^{d} |T_i| \geq \frac{n^{3/2}}{\varepsilon \cdot \log^3 n \cdot \log^2(n/\varepsilon)}.$$

5.4 Reducing from SSSQ to distinguishing $\mathcal{D}_{\text{yes}}$ and $\mathcal{D}_{\text{no}}$

In this section we reduce from SSSQ to the problem of distinguishing the pair of distributions $\mathcal{D}_{\text{yes}}$ and $\mathcal{D}_{\text{no}}$. More precisely, let $\text{ALG}^* = (h, X)$ be a deterministic and non-adaptive algorithm that makes $q \leq (n/\varepsilon)^2$ string queries[3] $X = (x_1, \ldots, x_q)$ to a hidden function f drawn from either $\mathcal{D}_{\text{yes}}$ or $\mathcal{D}_{\text{no}}$, applies the (deterministic) map h to return

[3] Any algorithm that makes more than this many queries already fits the $\widetilde{\Omega}(n^{3/2}/\varepsilon)$ lower bound we aim for.

$h(f(x_1), \ldots, f(x_q)) \in \{\text{“yes”}, \text{“no”}\}$, and satisfies

$$\Pr_{\boldsymbol{f} \sim \mathcal{D}_{\text{yes}}} \left[\textsc{Alg}^*(\boldsymbol{f}) = \text{“yes”} \right] - \Pr_{\boldsymbol{f} \sim \mathcal{D}_{\text{no}}} \left[\textsc{Alg}^*(\boldsymbol{f}) = \text{“yes”} \right] \geq 3/4. \tag{5.5}$$

We show how to define from $\textsc{Alg}^* = (h, X)$ an algorithm $\textsc{Alg} = (g, T)$ for the problem SSSQ with query complexity at most $\tau \cdot q$ and advantage $2/3$, where $\tau = c_\alpha \cdot 5 \log(n/\varepsilon)$ and

$$c_\alpha = -\frac{1}{\log(1.5 - \alpha)} > 0 \quad \text{with} \quad (1.5 - \alpha)^{c_\alpha} = 1/2$$

is a constant that depends on α. Given this reduction it follows from Lemma 5.3.1 that $q \geq \widetilde{\Omega}(n^{3/2}/\varepsilon)$. This finishes the proof of Theorem 65.

We first give a sketch of the reduction and some intuition behind the proof. Fixing a subset M of $[n]$ of size t, we use $X_1, \ldots, X_d$, for some $d \geq 1$, to denote a partition of the query strings $x_1, \ldots, x_q$ such that x_i and x_j belong to the same X_ℓ if and only if $(x_i)_{|M} = (x_j)_{|M}$. For each $\ell \in [d]$, we use T_ℓ to denote the set of indices $k \in \overline{M}$ such that $x_k \neq y_k$ for some strings $x, y \in X_\ell$. The first part of the proof shows that there exists an M such that (1) $\textsc{Alg}^*$ can distinguish $\mathcal{D}_{\text{yes}}$ and $\mathcal{D}_{\text{no}}$ conditioning on $\mathbf{M} = M$ (recall in both $\mathcal{D}_{\text{yes}}$ and $\mathcal{D}_{\text{no}}$, $\mathbf{M}$ is a subset of $[n]$ of size t drawn uniformly at random), and (2) $\|x - y\|_1 \leq \tau$ for all $\ell \in [d]$ and $x, y \in X_\ell$, which in turn implies that $\sum_{\ell \in [d]} |T_\ell| \leq \tau \cdot q$. Indeed we show that most draws from $\mathbf{M}$ satisfy both properties; the intuition behind (2) is that two query strings with a large Hamming distance would have different projections on $\mathbf{M}$ with high probability. Fixing such an M, we identify indices of $\overline{M}$ as those of $[m]$ in SSSQ (by picking an arbitrary bijection between them) and show that $T = (T_1, \ldots, T_d)$ can be used to obtain an algorithm $\textsc{Alg} = (g, T)$ for SSSQ, with query complexity at most $\tau \cdot q$, for some appropriate g. The intuition is that in SSSQ we receive intersections of T_ℓ with random subsets $\mathbf{S}_\ell$ drawn independently from the hidden subset $\mathbf{A}$, which can be used to simulate random functions $\boldsymbol{h}_\ell$ over $\mathbf{S}_\ell$ evaluated on strings in X_ℓ.

We start the reduction with some notation. For a fixed M of size t, we use $\mathcal{E}_{\text{yes}}(M)$ to denote the distribution of $\mathbf{A}$ and $\mathbf{H}$ sampled in the randomized procedure for $\mathcal{D}_{\text{yes}}$, conditioning on $\mathbf{M} = M$. We define $\mathcal{E}_{\text{no}}(M)$ similarly. Then conditioning on $\mathbf{M} = M$, $\boldsymbol{f} \sim \mathcal{D}_{\text{yes}}$ is distributed as $f_{M,\mathbf{A},\mathbf{H}}$ with $(\mathbf{A}, \mathbf{H}) \sim \mathcal{E}_{\text{yes}}(M)$ and $\boldsymbol{f} \sim \mathcal{D}_{\text{no}}$ is distributed as

$f_{M,\mathbf{A},\mathbf{H}}$ with $(\mathbf{A},\mathbf{H}) \sim \mathcal{E}_{\text{no}}(M)$. This allows us to rewrite (6.5) as

$$\frac{1}{\binom{n}{t}} \cdot \sum_{M:|M|=t} \left(\mathop{\mathbf{Pr}}_{(\mathbf{A},\mathbf{H})\sim\mathcal{E}_{\text{yes}}(M)} \left[\text{ALG}^*(\boldsymbol{f}_{M,\mathbf{A},\mathbf{H}}) = \text{“yes”}\right] - \mathop{\mathbf{Pr}}_{(\mathbf{A},\mathbf{H})\sim\mathcal{E}_{\text{no}}(M)} \left[\text{ALG}^*(\boldsymbol{f}_{M,\mathbf{A},\mathbf{H}}) = \text{“yes”}\right] \right) \geq \frac{3}{4}.$$

We say $M \subset [n]$ is *good* if any two queries x_i and x_j in X with Hamming distance $\|x_i - x_j\|_1 \geq \tau$ have different projections on M, i.e., $(x_i)_{|M} \neq (x_j)_{|M}$. We prove below that most M's are good.

Claim 5.4.1. $\mathbf{Pr}_{\mathbf{M}}\left[\mathbf{M} \textit{ is not good}\right] = o(1)$.

Proof. For each pair of strings x_i and x_j in X with Hamming distance at least τ, the probability of them having the same projection on $\mathbf{M}$ (drawn uniformly from all size-t subsets) is at most

$$\frac{\binom{n-\tau}{t}}{\binom{n}{t}} = \frac{(n-\tau-t+1)\cdots(n-t)}{(n-\tau+1)\cdots n} \leq \left(1-\frac{t}{n}\right)^{\tau} \leq \left(2(1-\alpha)+o(1)\right)^{\tau} < (1.5-\alpha)^{\tau} \leq O\left(\frac{\varepsilon}{n}\right)^{5},$$

by our choices of c_α and τ. The claim follows by a union bound over at most $q^2 \leq (n/\varepsilon)^4$ pairs. □

We can split the sum (6.5) into two sums: the sum over good M and the sum over bad M. By Claim 5.4.1 the contribution from the bad M is at most $o(1)$, and thus we have that

$$\frac{1}{\binom{n}{t}} \cdot \sum_{\text{good } M} \left(\mathop{\mathbf{Pr}}_{(\mathbf{A},\mathbf{H})\sim\mathcal{E}_{\text{yes}}(M)} \left[\text{ALG}^*(\boldsymbol{f}_{M,\mathbf{A},\mathbf{H}}) = \text{“yes”}\right] - \mathop{\mathbf{Pr}}_{(\mathbf{A},\mathbf{H})\sim\mathcal{E}_{\text{no}}(M)} \left[\text{ALG}^*(\boldsymbol{f}_{M,\mathbf{A},\mathbf{H}}) = \text{“yes”}\right] \right)$$

is at least $3/4 - o(1)$. Thus, there must exist a good set $M \subset [n]$ of size t with

$$\mathop{\mathbf{Pr}}_{(\mathbf{A},\mathbf{H})\sim\mathcal{E}_{\text{yes}}(M)} \left[\text{ALG}^*(\boldsymbol{f}_{M,\mathbf{A},\mathbf{H}}) = \text{“yes”}\right] - \mathop{\mathbf{Pr}}_{(\mathbf{A},\mathbf{H})\sim\mathcal{E}_{\text{no}}(M)} \left[\text{ALG}^*(\boldsymbol{f}_{M,\mathbf{A},\mathbf{H}}) = \text{“yes”}\right] \geq 2/3. \tag{5.6}$$

Fix such a good M. We use $\text{ALG}^* = (h, X)$ and M to define an algorithm $\text{ALG} = (g, T)$ for SSSQ as follows (note that the algorithm ALG below actually works over the universe $\overline{M}$ (of size m) instead of $[m]$ as in the original definition of SSSQ but this can be handled

by picking any bijection between $\overline{M}$ and $[m]$; accordingly $\mathbf{A} \sim \mathcal{A}_{\text{yes}}$ is drawn by including each element of $\overline{M}$ with probability p and $\mathbf{A} \sim \mathcal{A}_{\text{no}}$ is drawn by including each element of $\overline{M}$ with probability p_{no}). We start with T:

1. First we use M to define an equivalence relation $\sim$ over the query set X, where $x_i \sim x_j$ if $(x_i)_{|M} = (x_j)_{|M}$. Let $X_1, \ldots, X_d$, $d \geq 1$, denote the equivalence classes of X, and let us write $\rho(\ell)$ for each $\ell \in [d]$ to denote the value $\Gamma(x) \in [N]$ that is shared by all strings $x \in X_\ell$.
2. Next we define a sequence of subsets of $\overline{M}$, $T = (T_1, \ldots, T_d)$, as the set queries of ALG, where

$$T_\ell = \Big\{ i \in \overline{M} \colon \exists x, y \in X_\ell \text{ such that } x_i \neq y_i \Big\}. \tag{5.7}$$

To upper bound $|T_\ell|$, fixing an arbitrary string $x \in X_\ell$ and recalling that M is good, we have that

$$|T_\ell| \leq \sum_{y \in X_\ell} \|x - y\|_1 \leq \sum_{y \in X_\ell} \tau = \tau \cdot |X_\ell|.$$

As a result, the query complexity of ALG (using T as its set queries) is at most

$$\sum_{\ell=1}^{d} |T_\ell| \leq \tau \cdot \sum_{\ell=1}^{d} |X_\ell| \leq \tau \cdot q.$$

It remains to define h and then prove that the advantage of $\text{ALG} = (g, T)$ for SSSQ is at least $2/3$. Indeed the g that we define is a randomized map and we describe it as a randomized procedure below (by Remark 56 one can extract from g a deterministic map that achieves the same advantage):

1. Given $v_1, \ldots, v_d$, $v_\ell \in \{0,1\}^{T_\ell}$, as the strings returned by the oracle upon being given T, let

$$R_\ell = \Big\{ j \in T_\ell : v_{\ell,j} = 1 \Big\}. \tag{5.8}$$

   For each $\ell \in [d]$, the procedure draws a random function $\boldsymbol{f}_\ell : \{0,1\}^{R_\ell} \to \{0,1\}$, by flipping $2^{|R_\ell|}$ many independent and unbiased random bits.

2. Next for each query $x \in X_\ell$, $\ell \in [d]$, we feed $\boldsymbol{f}_\ell(x_{|R_\ell})$ to h as the bit that the oracle returns upon the query x. Finally the procedure returns the result ("yes" or "no") that h returns.

In the rest of the proof we show that the advantage of $\textsc{Alg} = (g, T)$ is exactly the same as the LHS of (5.6) and thus, is at least $2/3$.

For convenience, we use $\mathcal{V}_{\text{yes}}$ to denote the distribution of responses $\boldsymbol{v} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_d)$ to T when $\mathbf{A} \sim \mathcal{A}_{\text{yes}}$, and $\mathcal{V}_{\text{no}}$ to denote the distribution when $\mathbf{A} \sim \mathcal{A}_{\text{no}}$. Then the advantage of $\textsc{Alg}$ is

$$\mathop{\mathbf{Pr}}_{\boldsymbol{v} \sim \mathcal{V}_{\text{yes}}} \left[g(\boldsymbol{v}) = \text{“yes”} \right] - \mathop{\mathbf{Pr}}_{\boldsymbol{v} \sim \mathcal{V}_{\text{no}}} \left[g(\boldsymbol{v}) = \text{“yes”} \right].$$

It suffices to show that

$$\mathop{\mathbf{Pr}}_{\boldsymbol{v} \sim \mathcal{V}_{\text{yes}}} \left[g(\boldsymbol{v}) = \text{“yes”} \right] = \mathop{\mathbf{Pr}}_{(\mathbf{A},\mathbf{H}) \sim \mathcal{E}_{\text{yes}}(M)} \left[\textsc{Alg}^*(\boldsymbol{f}_{M,\mathbf{A},\mathbf{H}}) = \text{“yes”} \right] \quad \text{and} \tag{5.9}$$

$$\mathop{\mathbf{Pr}}_{\boldsymbol{v} \sim \mathcal{V}_{\text{no}}} \left[g(\boldsymbol{v}) = \text{“yes”} \right] = \mathop{\mathbf{Pr}}_{(\mathbf{A},\mathbf{H}) \sim \mathcal{E}_{\text{no}}(M)} \left[\textsc{Alg}^*(\boldsymbol{f}_{M,\mathbf{A},\mathbf{H}}) = \text{“yes”} \right]. \tag{5.10}$$

We show (5.9); the proof of (5.10) is similar. From the definition of $\mathcal{V}_{\text{yes}}$ and $\mathcal{E}_{\text{yes}}(M)$ the distribution of $(\mathbf{R}_\ell : \ell \in [d])$ derived from $\boldsymbol{v} \sim \mathcal{V}_{\text{yes}}$ using (5.8) is the same as the distribution of $(\mathbf{S}_{\rho(\ell)} \cap T_\ell : \ell \in [d])$: both are sampled by first drawing a random subset $\mathbf{A}$ of $\overline{M}$ and then drawing a random subset of $\mathbf{A} \cap T_\ell$ independently by including each element of $\mathbf{A} \cap T_\ell$ with the same probability $\varepsilon/\sqrt{n}$ (recall in particular equation (5.4) and step 3 of the randomized procedure specifying $\mathcal{D}_{\text{yes}}$ in Section 5.2). Since $\boldsymbol{f}_{M,\mathbf{A},\mathbf{H}}(x)$ for $x \in X_\ell$ is determined by a random Boolean function $\boldsymbol{h}_{\rho(\ell)}$ from $\{0,1\}^{\mathbf{S}_{\rho(\ell)}}$ to $\{0,1\}$, and since all the queries in X_ℓ only differ by coordinates in T_ℓ, the distribution of the q bits that g feeds to h when $\boldsymbol{v} \sim \mathcal{V}_{\text{yes}}$ is the same as the distribution of $(\boldsymbol{f}(x) : x \in X)$ when $\boldsymbol{f} \sim \mathcal{E}_{\text{yes}}(M)$. This finishes the proof of (5.9), and concludes our reduction argument.

5.5 A lower bound on the non-adaptive query complexity of SSSQ

We will prove Lemma 5.3.1 by first giving a reduction from an even simpler algorithmic task, which we describe next in Section 5.5.1. We will then prove a lower bound for the simpler task in Section 5.5.2.

5.5.1 Set-Size-Element-Queries (SSEQ)

Recall the parameters m, p, p_{no} and ε and the two distributions $\mathcal{A}_{\text{yes}}$ and $\mathcal{A}_{\text{no}}$ used in the definition of problem SSSQ. We now introduce a simpler algorithmic task called the Set-Size-Element-Queries (SSEQ) problem using the same parameters and distributions. As in the SSSQ problem, the goal is to distinguish between the case in which a hidden subset A is drawn from $\mathcal{A}_{\text{yes}}$ or from $\mathcal{A}_{\text{no}}$.

Let A be a subset of $[m]$ hidden in an oracle. An algorithm accesses the oracle to tell whether it is drawn from $\mathcal{A}_{\text{yes}}$ or $\mathcal{A}_{\text{no}}$. The difference between SSSQ and SSEQ is the way A is accessed. In SSEQ, an algorithm $\text{ALG}' = (h, \ell)$ submits a vector $\ell = (\ell_1, \ldots, \ell_m)$ of nonnegative integers.

- On receiving ℓ, the oracle returns a random response vector $\boldsymbol{b} \in \{0,1\}^m$, where each entry $\boldsymbol{b}_i$ is distributed independently as follows: if $i \notin A$ then $\boldsymbol{b}_i = 0$, and if $i \in A$ then

$$\boldsymbol{b}_i = \begin{cases} 1 & \text{with probability } \lambda(\ell_i) \\ 0 & \text{with probability } 1 - \lambda(\ell_i) \end{cases}, \qquad \text{where } \lambda(\ell_i) = 1 - \left(1 - \frac{\varepsilon}{\sqrt{n}}\right)^{\ell_i}.$$

  Equivalently, for each $i \in A$, the oracle independently flips ℓ_i coins, each of which is 1 with probability $\varepsilon/\sqrt{n}$, and at the end returns $\boldsymbol{b}_i = 1$ to the algorithm if and only if at least one of the coins is 1. Thus, we refer to each ℓ_i as ℓ_i *element-queries* for the ith element.

- After receiving the vector $\boldsymbol{b}$ from the oracle, ALG' returns the value $h(\boldsymbol{b}) \in \{\text{“yes”}, \text{“no”}\}$. Here h is a deterministic map from $\{0,1\}^m$ to $\{\text{“yes”}, \text{“no”}\}$.

Similar to before, the performance of ALG' is measured by its query complexity and its advantage:

- The query complexity of $\text{ALG}' = (h, \ell)$ is defined as $\|\ell\|_1 = \sum_{i=1}^{m} \ell_i$. For its advantage, we let $\mathcal{B}_{\text{yes}}$ denote the distribution of response vectors $\boldsymbol{b}$ to query ℓ when $\mathbf{A} \sim \mathcal{A}_{\text{yes}}$, and $\mathcal{B}_{\text{no}}$ denote the distribution when $\mathbf{A} \sim \mathcal{D}_{\text{no}}$. The advantage of $\text{ALG}' = (h, \ell)$ is then defined as

$$\Pr_{\boldsymbol{b} \sim \mathcal{B}_{\text{yes}}} \left[h(\boldsymbol{b}) = \text{“yes”} \right] - \Pr_{\boldsymbol{b} \sim \mathcal{B}_{\text{no}}} \left[h(\boldsymbol{b}) = \text{“yes”} \right].$$

Remark 57. *It is worth pointing out (we will use it later) that the highest possible advantage over all deterministic maps h is a monotonically non-decreasing function of the coordinates of ℓ. To see this, let A be the underlying set and let ℓ and ℓ' be two vectors with $\ell_i \leq \ell'_i$ for every $i \in [m]$. Let $\boldsymbol{b}$ and $\boldsymbol{b}'$ be the random vectors returned by the oracle upon ℓ and ℓ'. Then we can define $\boldsymbol{b}^*$ using $\boldsymbol{b}'$ as follows: $\boldsymbol{b}^*_i = 0$ if $\boldsymbol{b}'_i = 0$; otherwise when $\boldsymbol{b}'_i = 1$, we set*

$$\boldsymbol{b}^*_i = \begin{cases} 1 & \textit{with probability } \lambda(\ell_i)/\lambda(\ell'_i) \\ 0 & \textit{with probability } 1 - \lambda(\ell_i)/\lambda(\ell'_i) \end{cases}.$$

One can easily verify that the distribution of $\boldsymbol{b}$ is exactly the same as the distribution of $\boldsymbol{b}^*$. Hence there is a randomized map h' such that the advantage of (h', ℓ') is at least as large as the highest possible advantage achievable using ℓ. The remark now follows by our earlier observation in Remark 56 that the highest possible advantage using ℓ' is always achieved by a deterministic h'.

The following lemma reduces the proof of Lemma 5.3.1 to proving a lower bound for SSEQ.

Lemma 5.5.1. *Given any deterministic and non-adaptive algorithm* $\text{ALG} = (g, T)$ *for* SSSQ*, there is a deterministic and non-adaptive algorithm* $\text{ALG}' = (h, \ell)$ *for* SSEQ *with the same query complexity as* ALG *and advantage at least as large as that of* ALG*.*

Proof. We show how to construct $\text{ALG}' = (h, \ell)$ from $\text{ALG} = (g, T)$, where h is a randomized map, such that ALG' has exactly the same query complexity and advantage as those of ALG. The lemma then follows from the observation we made earlier in Remark 56.

We define ℓ first. Given $T = (T_1, \ldots, T_d)$ for some $d \geq 1$, $\ell = (\ell_1, \ldots, \ell_m)$ is defined as

$$\ell_j = \Big|\{i \in [d] : j \in T_i\}\Big|.$$

So $\|\ell\|_1 = \sum_{i=1}^{d} |T_i|$. Recall from Section 5.4 that $\mathcal{V}_{\text{yes}}$ and $\mathcal{V}_{\text{no}}$ denote the distributions supported on responses $\boldsymbol{v} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_d)$ to T in SSSQ when $\mathbf{A} \sim \mathcal{A}_{\text{yes}}$ and $\mathbf{A} \sim \mathcal{A}_{\text{no}}$, respectively. To define h, we describe a randomized procedure P that, given any $b \in \{0,1\}^m$, outputs a sequence of random vectors $\boldsymbol{v} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_d)$ which simulates $\mathcal{V}_{\text{yes}}$ if $\boldsymbol{b} \sim \mathcal{B}_{\text{yes}}$ and $\mathcal{V}_{\text{no}}$ if $\boldsymbol{b} \sim \mathcal{B}_{\text{no}}$.In other words, we define P below and prove the following claim:

Claim 5.5.2. *If* $\boldsymbol{b} \sim \mathcal{B}_{yes}$ *(or* $\mathcal{B}_{no}$*), then* $P(\boldsymbol{b})$ *is distributed the same as* $\mathcal{V}_{yes}$ *(or* $\mathcal{V}_{no}$*, respectively*)*.*

Assuming Claim 5.5.2, we can set $h = g \circ P$ and the advantage of ALG' would be the same as that of ALG. In the rest of the proof, we describe the randomized procedure P and prove Claim 5.5.2.

Given $b \in \{0,1\}^m$, P outputs a sequence of random vectors $\boldsymbol{v} = (\boldsymbol{v}_1, \ldots, \boldsymbol{v}_d)$ as follows:

- If $b_j = 0$, then for each $i \in [d]$ with $j \in T_i$, P sets $\boldsymbol{v}_{i,j} = 0$.
- If $b_j = 1$ (this implies that $\ell_j > 0$ and $j \in T_i$ for some $i \in [d]$), P sets $(\boldsymbol{v}_{i,j} : i \in [d], j \in T_i)$ to be a length-r, where $r = |\{i \in [d] : j \in T_i\}|$, binary string in which each bit is independently 1 with probability $\varepsilon/\sqrt{n}$ and 0 with probability $1 - \varepsilon/\sqrt{n}$, conditioned on its not being 0^r.

Proof of Claim 5.5.2. It suffices to prove that, fixing any $A \subseteq [m]$ as the underlying set hidden in the oracle, the distribution of $\boldsymbol{v}$ is the same as that of $P(\boldsymbol{b})$. The claim then follows

since in the definitions of both $\mathcal{B}_{\text{yes}}$ and $\mathcal{V}_{\text{yes}}$ (or $\mathcal{B}_{\text{no}}$ and $\mathcal{V}_{\text{no}}$), A is drawn from $\mathcal{A}_{\text{yes}}$ (or $\mathcal{A}_{\text{no}}$, respectively).

Consider a sequence v of d vectors $v_1, \ldots, v_d$ with $v_i \in \{0,1\}^{T_i}$ for each $i \in [d]$, and let

$$n_{j,1} = |\{i \in [d] : j \in T_i \text{ and } v_{i,j} = 1\}| \quad \text{and} \quad n_{j,0} = |\{i \in [d] : j \in T_i \text{ and } v_{i,j} = 0\}|,$$

for each $j \in [m]$. Then the $\boldsymbol{v}$ returned by the oracle (in SSSQ) is equal to v with probability:

$$\mathbf{1}\Big[\forall j \notin A,\, n_{j,1} = 0\Big] \cdot \prod_{j \in A} \left(\frac{\varepsilon}{\sqrt{n}}\right)^{n_{j,1}} \left(1 - \frac{\varepsilon}{\sqrt{n}}\right)^{n_{j,0}}, \tag{5.11}$$

since all coordinates $\boldsymbol{v}_{i,j}$ are independent. (Here $\mathbf{1}$ denotes the indicator function, so $\mathbf{1}[E]$ is 1 if event E holds and is 0 otherwise.) On the other hand, the probability of $P(\boldsymbol{b}) = v$ is

$$\mathbf{1}\Big[\forall j \notin A,\, n_{j,1} = 0\Big] \cdot \prod_{j \in A} \left(\mathbf{1}\Big[n_{j,0} = \ell_j\Big] \cdot \left(1 - \frac{\varepsilon}{\sqrt{n}}\right)^{\ell_j} + \mathbf{1}\Big[n_{j,1} \geq 1\Big] \cdot \left(\frac{\varepsilon}{\sqrt{n}}\right)^{n_{j,1}} \left(1 - \frac{\varepsilon}{\sqrt{n}}\right)^{n_{j,0}}\right),$$

which is exactly the same as the probability of $\boldsymbol{v} = v$ in (5.11). □

This finishes the proof of Lemma 5.5.1. □

5.5.2 A lower bound for SSEQ

We prove the following lower bound for SSEQ, from which Lemma 5.3.1 follows:

Lemma 5.5.3. *Any deterministic, non-adaptive* ALG′ *for* SSEQ *with advantage at least* $2/3$ *satisfies*

$$\|\ell\|_1 > s \stackrel{\text{def}}{=} \frac{n^{3/2}}{\varepsilon \cdot \log^3 n \cdot \log^2(n/\varepsilon)}.$$

Proof. Assume for contradiction that there is an algorithm $\text{ALG}' = (h, \ell)$ with $\|\ell\|_1 \leq s$ and advantage at least $2/3$. Let ℓ^* be the vector obtained from ℓ by rounding each positive ℓ_i to the smallest power of 2 that is at least as large as ℓ_i (and taking $\ell_i^* = 0$ if $\ell_i = 0$). From Remark 57, there must be a map h^* such that (h^*, ℓ^*) also has advantage at least $2/3$ but now we have 1) $\|\ell^*\|_1 \leq 2s$ and 2) every positive entry of ℓ^* is a power of 2. Below we abuse notation and still use $\text{ALG}' = (h, \ell)$ to denote (h^*, ℓ^*): $\text{ALG}' = (h, \ell)$ satisfies

$\|\ell\|_1 \leq 2s$, every positive entry of ℓ is a power of 2, and has advantage at least $2/3$. We obtain a contradiction below by showing that any such ℓ can only have an advantage of $o(1)$.

Let $L = \lceil \log(2s) \rceil = O(\log(n/\varepsilon))$. Given that $\|\ell\|_1 \leq 2s$ we can partition $\{i \in [m] : \ell_i > 0\}$ into $L+1$ bins $C_0, \ldots, C_L$, where bin C_j contains those coordinates $i \in [m]$ with $\ell_i = 2^j$. We may make two further assumptions on $\text{ALG}' = (h, \ell)$ that will simplify the lower bound proof:

- We may reorder the entries in decreasing order and assume without loss of generality that
$$\ell = \left(\underbrace{2^L, \ldots, 2^L}_{c_L}, \underbrace{2^{L-1}, \ldots, 2^{L-1}}_{c_{L-1}}, \ldots, \underbrace{1, \ldots, 1}_{c_0}, 0, \ldots, 0 \right), \tag{5.12}$$
where $c_j = |C_j|$ satisfies $\sum_j c_j \cdot 2^j \leq 2s$. This is without loss of generality since $\mathcal{A}_{\text{yes}}$ and $\mathcal{A}_{\text{no}}$ are symmetric in the coordinates (and so are $\mathcal{B}_{\text{yes}}$ and $\mathcal{B}_{\text{no}}$).
- For the same reason we may assume that the map $h(b)$ depends only on the number of 1's of b in each set C_j, which we refer to as the *summary* $S(b)$ of b:
$$S(b) \stackrel{\text{def}}{=} \left(\|b_{|C_L}\|_1, \|b_{|C_{L-1}}\|_1, \ldots, \|b_{|C_0}\|_1 \right) \in \mathbb{Z}_{\geq 0}^{L+1}.$$
To see that this is without loss of generality, consider a randomized procedure P that, given $b \in \{0,1\}^m$, applies an independent random permutation over the entries of C_j for each bin $j \in [0 : L]$. One can verify that the random map $h' = h \circ P$ only depends on the summary $S(b)$ of b but achieves the same advantage as h.

Given a query ℓ as in (5.12), we define $\mathcal{S}_{\text{yes}}$ to be the distribution of $S(\boldsymbol{b})$ for $\boldsymbol{b} \sim \mathcal{B}_{\text{yes}}$ (recall that $\mathcal{B}_{\text{yes}}$ is the distribution of the vector $\boldsymbol{b}$ returned by the oracle upon the query ℓ when $\mathbf{A} \sim \mathcal{A}_{\text{yes}}$). Similarly we define $\mathcal{S}_{\text{no}}$ as the distribution of $S(\boldsymbol{b})$ for $\boldsymbol{b} \sim \mathcal{B}_{\text{no}}$. As h only depends on the summary the advantage is at most $d_{\text{TV}}(\mathcal{S}_{\text{yes}}, \mathcal{S}_{\text{no}})$, which we upper bound below by $o(1)$.

From the definition of $\mathcal{B}_{\text{yes}}$ (or $\mathcal{B}_{\text{no}}$, respectively) and the fact that $\mathcal{A}_{\text{yes}}$ (or $\mathcal{A}_{\text{no}}$, respectively) is symmetric over the m coordinates, we have that the $L+1$ entries of $\mathcal{S}_{\text{yes}}$ (of $\mathcal{S}_{\text{no}}$, respectively) are mutually independent, and that their entries for each C_j,

$j \in [0:L]$, are distributed as $\mathrm{Bin}(c_j, p\lambda_j)$ (as $\mathrm{Bin}(c_j, p_{\mathrm{no}}\lambda_j)$, respectively), where we have $\lambda_j = 1 - (1 - (\varepsilon/\sqrt{n}))^{2^j}$.

In order to prove that $d_{\mathrm{TV}}(\mathcal{S}_{\mathrm{yes}}, \mathcal{S}_{\mathrm{no}}) = o(1)$ and achieve the desired contradiction, we will give upper bounds on the total variation distance between their C_j-entries for each $j \in \{0, \ldots, L\}$.

Claim 5.5.4. *For every $j \in [0:L]$, we have $d_{\mathrm{TV}}(\mathbf{X}, \mathbf{Y}) \le o(1/L)$, where $\mathbf{X} \sim \mathrm{Bin}(c_j, p\lambda_j)$ and $\mathbf{Y} \sim \mathrm{Bin}(c_j, p_{\mathrm{no}}\lambda_j)$.*

We delay the proof of Claim 5.5.4, but assuming it we may simply apply the following well-known proposition to conclude that $d_{\mathrm{TV}}(\mathcal{S}_{\mathrm{yes}}, \mathcal{S}_{\mathrm{no}}) = o(1)$.

Proposition 5.5.5 (Subadditivity of total variation distance)**.** *Let $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_k)$ and $\mathbf{Y} = (\mathbf{Y}_1, \ldots, \mathbf{Y}_k)$ be two tuples of independent random variables. Then $d_{\mathrm{TV}}(\mathbf{X}, \mathbf{Y}) \le \sum_{i=1}^{k} d_{\mathrm{TV}}(\mathbf{X}_i, \mathbf{Y}_i)$.*

This gives us a contradiction and finishes the proof of Lemma 5.5.3. □

Below we prove Claim 5.5.4.

Proof of Claim 5.5.4. Consider any fixed $j \in [0:L]$. The claim is trivial when $c_j = 0$ so we assume below that $c_j > 0$.

Let $r_j = p\lambda_j$ and $x_j = \log n \cdot \lambda_j/\sqrt{n}$. Then $\mathbf{X} \sim \mathrm{Bin}(c_j, r_j)$ and $\mathbf{Y} \sim \mathrm{Bin}(c_j, r_j + x_j)$. As indicated in Equation (2.15) of [1], Equation (15) of [124] gives

$$d_{\mathrm{TV}}(\mathbf{X}, \mathbf{Y}) \le O\left(\frac{\tau_j(x_j)}{(1 - \tau_j(x_j))^2}\right), \quad \text{where} \quad \tau_j(x_j) \stackrel{\text{def}}{=} x_j\sqrt{\frac{c_j + 2}{2r_j(1 - r_j)}}, \tag{5.13}$$

whenever $\tau_j(x_j) < 1$. Substituting for x_j and r_j, we have (using $c_j \ge 1$, $r_j \le 1/2$ and $p = 1/2$)

$$\tau_j(x_j) = O\left(\frac{\log n \cdot \lambda_j}{\sqrt{n}} \cdot \sqrt{\frac{c_j}{r_j}}\right) = O\left(\log n \cdot \sqrt{\frac{\lambda_j \cdot c_j}{n}}\right) = O\left(\frac{1}{L} \cdot \sqrt{\frac{n^{1/2} \cdot \lambda_j}{2^j \cdot \varepsilon \cdot \log n}}\right),$$

where the last inequality follows from

$$c_j \cdot 2^j \leq 2s \leq O\left(\frac{n^{3/2}}{\varepsilon \cdot \log^3 n \cdot L^2}\right).$$

Finally, note that (using $1 - x > e^{-2x}$ for small positive x and $1 - x \leq e^{-x}$ for all x):

$$1 - \lambda_j = \left(1 - \frac{\varepsilon}{\sqrt{n}}\right)^{2^j} \geq \left(e^{-2\varepsilon/\sqrt{n}}\right)^{2^j} = e^{-2^{j+1}\varepsilon/\sqrt{n}} \geq 1 - O(2^j\varepsilon/\sqrt{n})$$

so that $\frac{\sqrt{n} \cdot \lambda_j}{2^j \cdot \varepsilon} = O(1)$. This implies $\tau_j(x_j) = o(1/L)$. The claim then follows from (5.13). □

5.6 Proof of Theorem 54 assuming Theorem 65

We prove the following claim in Appendix 5.6.1.

Claim 5.6.1. *Let $\varepsilon(n)$ be a function that satisfies $2^{-n} \leq \varepsilon(n) \leq 1/5$ for sufficiently large n. Then any non-adaptive algorithm that accepts the all-0 function with probability at least $5/6$ and rejects every function that is ε-far from $(n-1)$-juntas with probability at least $5/6$ must make $\Omega(1/\varepsilon)$ queries.*

Next let $k(n)$ and $\varepsilon(n)$ be the pair of functions from the statement of Theorem 54. We consider a sufficiently large n (letting $k = k(n)$ and $\varepsilon = \varepsilon(n)$ below) and separate the proof into two cases:

$$2^{-(2\alpha-1)k/(2\alpha)} \leq \varepsilon \leq 1/6 \quad \text{and} \quad 2^{-n} \leq \varepsilon < 2^{-(2\alpha-1)k/(2\alpha)}.$$

For the first case, if $k = O(1)$ then the bound we aim for is simply $\widetilde{\Omega}(1/\varepsilon)$, which follows trivially from Claim 7.3.4 (since $k \leq \alpha n < n - 1$ and the all-0 function is a k-junta). Otherwise we combine the following reduction with Theorem 65: any ε-tester for k-juntas over n-variable functions can be used to obtain an ε-tester for k-juntas over (k/α)-variable functions. This can be done by adding $n - k/\alpha$ dummy variables to any (k/α)-variable function to make the number of variables n (as $k \leq \alpha n$). The lower bound then follows

from Theorem 65 since α is a constant. For the second case, the lower bound claimed in Theorem 54 is $\widetilde{\Omega}(1/\varepsilon)$, which follows again from Claim 7.3.4. This concludes the proof of Theorem 54 given Theorem 65 and Claim 7.3.4. □

5.6.1 Proof of Claim 7.3.4

Let C be a sufficiently large constant. We prove Claim 7.3.4 by considering two cases:

$$\varepsilon \geq \frac{C \log n}{2^n} \quad \text{and} \quad \varepsilon < \frac{C \log n}{2^n}.$$

For the first case of $2^n\varepsilon \geq C \log n$, we use $\mathcal{D}_1$ to denote the following distribution over n-variable Boolean functions: to draw $\boldsymbol{g} \sim \mathcal{D}_1$, independently for each $x \in \{0,1\}^n$ the value of $\boldsymbol{g}(x)$ is set to 0 with probability $1 - 3\varepsilon$ (recall that $\varepsilon \leq 1/5$) and 1 with probability 3ε.

We prove the following lemma for the distribution $\mathcal{D}_1$:

Lemma 5.6.2. *With probability at least $1 - o(1)$, $\boldsymbol{g} \sim \mathcal{D}_1$ is ε-far from every $(n-1)$-junta.*

Proof. Note that every $(n-1)$-junta is such that for some $i \in [n]$, the function does not depend on the i-th variable; we refer to such a function as a type-i junta. An easy lower bound for the distance from a function g to all type-i juntas is the number of g-bichromatic edges $(x, x^{(i)})$ divided by 2^n. When $\boldsymbol{g} \sim \mathcal{D}_1$ each edge $(x, x^{(i)})$ is independently $\boldsymbol{g}$-bichromatic with probability $6\varepsilon(1-3\varepsilon) \geq 12\varepsilon/5$ (as $\varepsilon \leq 1/5$). Thus when $2^n\varepsilon \geq C \log n$, the expected number of such edges is at least

$$2^{n-1} \cdot (12\varepsilon/5) \geq (6/5) \cdot 2^n\varepsilon \geq (6/5) \cdot C \log n.$$

Using a Chernoff bound, the probability of having fewer than $2^n\varepsilon$ bichromatic edges along direction i is at most $1/n^2$ when C is sufficiently large. The lemma follows from a union bound over i. □

As a result, when $2^n\varepsilon \geq C \log n$, if $\mathcal{A}$ is a non-adaptive algorithm with the property

described in Claim 7.3.4, then $\mathcal{A}$ must satisfy

$$\mathbf{Pr}\left[\mathcal{A} \text{ accepts the all-0 function}\right] - \mathop{\mathbf{Pr}}_{\boldsymbol{g}\sim\mathcal{D}_1}\left[\mathcal{A} \text{ accepts } \boldsymbol{g}\right] \geq 2/3 - o(1).$$

But any such non-adaptive algorithm must make $\Omega(1/\varepsilon)$ queries as otherwise with high probability all of its queries to $\boldsymbol{g} \sim \mathcal{D}_1$ would be answered 0, and hence its behavior would be the same as if it were running on the all-0 function.

Finally we work on the case when $1 \leq 2^n\varepsilon = O(\log n)$. The proof is the same except that we let $\boldsymbol{g}$ be drawn from $\mathcal{D}_2$, which we define to be the distribution where all entries of $\boldsymbol{g} \sim \mathcal{D}_2$ are 0 except for exactly $2^n\varepsilon$ of them picked uniformly at random. The claim follows from the following lemma:

Lemma 5.6.3. *With probability at least $1 - o(1)$, $\boldsymbol{g} \sim \mathcal{D}_2$ is ε-far from every $(n-1)$-junta.*

Proof. This follows from the observation that, with probability $1 - o(1)$, no two points picked form an edge. When this occurs, we have $2^n\varepsilon$ bichromatic edges along the ith direction for all i. □

Chapter 6

Lower Bounds for Testing Monotonicity and Unateness

The main results of this chapter are lower bounds for testing monotonicity and unateness, in both adaptive, and non-adaptive settings.

Theorem 58 (Adaptive Monotonicity Testing Lower Bound)**.** *There exists a constant $\varepsilon_0 > 0$ such that any two-sided and adaptive algorithm for testing whether an unknown Boolean function $f : \{0,1\}^n \to \{0,1\}$ is monotone or ε_0-far from monotone must make $\Omega(n^{1/3}/\log^2 n)$ queries.*

In [34], Belovs and Blais obtained their $\tilde{\Omega}(n^{1/4})$ lower bound using a family of random functions known as *Talagrand's random DNFs* (or simply as the Talagrand function) [130]. A function drawn from this family is the disjunction of $N \equiv 2^{\sqrt{n}}$ many monotone terms T_i with each T_i being the conjunction of $\sqrt{n}$ variables sampled uniformly from $[n]$. So such a function looks like

$$f(x) = \bigvee_{i \in [N]} T_i(x) = \bigvee_{i \in [N]} \left(\bigwedge_{k \in S_i} x_k \right).$$

However, it turns out that there is a matching $\tilde{O}(n^{1/4})$-query, one-sided algorithm for functions of [34]. (See Section 6.5 for a sketch of the algorithm.) So the analysis of [34] is tight.

Our main contribution behind the lower bound of Theorem 58 is a new and harder family of random partitions for monotonicity testing, which we call *two-level Talagrand functions*. This starts by reexamining the construction of [34] from a slightly different angle, which leads to both natural generalizations and simpler analysis of such functions. The techniques developed in the proof of Theorem 58 can be easily adapted to prove a

	Best Upper Bound	Best Lower Bound	This Work
Non-adaptive			
Monotonicity	$\tilde{O}(\sqrt{n}/\varepsilon^2)$ [87]	$\tilde{\Omega}(n^{1/2-c})$ [54]	$\tilde{\Omega}(\sqrt{n})$
Unateness	$\tilde{O}(n/\varepsilon)$ [43]	$\Omega(\sqrt{n})$ (one-sided) [24]	$\tilde{\Omega}(n)$ (one-sided)
Adaptive			
Monotonicity	$\tilde{O}(\sqrt{n}/\varepsilon^2)$ [87]	$\tilde{\Omega}(n^{1/4})$ [34]	$\tilde{\Omega}(n^{1/3})$
Unateness	$\tilde{O}(n^{2/3}/\varepsilon^2)$ [50]		$\tilde{\Omega}(n^{2/3})$

Figure 6.1: Previous work and our results on monotonicity testing and unateness testing.

tight $\tilde{\Omega}(n^{1/2})$ lower bound for non-adaptive monotonicity testing, removing the $-c$ in the exponent of [54] (see Section 6.4).

Theorem 59 (Non-adaptive Monotonicity Testing Lower Bound)**.** *There exists a constant $\varepsilon_0 > 0$ such that any two-sided and non-adaptive algorithm for testing whether an unknown Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ is monotone or ε_0-far from monotone must make $\Omega(\sqrt{n}/\log^2 n)$ queries.*

Next for testing unateness, we present an $\tilde{\Omega}(n^{2/3})$ lower bound against adaptive algorithms.

Theorem 60 (Adaptive Unateness Testing Lower Bound)**.** *There exists a constant $\varepsilon_0 > 0$ such that any two-sided and adaptive algorithm for testing whether an unknown Boolean function $f : \{0,1\}^n \to \{0,1\}$ is unate versus ε_0-far from unate must make $\Omega(n^{2/3}/\log^3 n)$ queries.*

The lower bound construction behind Theorem 60 follows a similar framework. Some of the new ideas and techniques developed for the monotonicity lower bound are adapted to prove Theorem 60 though with a few twists that are unique to unateness.

Moreover, we obtain a linear lower bound for one-sided and non-adaptive unateness algorithms. This improves the $\Omega(\sqrt{n})$ lower bound of Baleshzar et al. [24] and matches the upper bound of Chakrabarty and Seshadhri [43] for such algorithms.

Theorem 61 (One-sided and Non-adaptive Unateness Testing Lower Bound)**.** *There exists a constant $\varepsilon_0 > 0$ such that any one-sided and non-adaptive algorithm for testing whether an unknown Boolean function is unate versus ε_0-far from unate must make $\Omega(n/\log^2 n)$ queries.*

We summarize previous work and our new results in Figure 6.1.

6.0.1 Distance to monotonicity and unateness

We review some characterizations of distance to monotonicity and unateness.

Lemma 6.0.1 (Lemma 4 in [63])**.** *Let $f\colon \{0,1\}^n \to \{0,1\}$ be a Boolean function. Then*

$$\operatorname{dist}\Big(f, \textsc{Mono}\Big) = |M|\Big/2^n,$$

where M is the maximal set of disjoint violating pairs of f.

Lemma 6.0.2. *Given $f\colon \{0,1\}^n \to \{0,1\}$, let $(E_i^+, E_i^- : i \in [n])$ be a tuple of sets such that (1) each set E_i^+ consists of monotone bi-chromatic edges $(x, x^{(i)})$ along direction i with $x_i = 0$, $f(x) = 0$ and $f(x^{(i)}) = 1$; (2) each set E_i^- consists of anti-monotone bi-chromatic edges $(x, x^{(i)})$ along direction i with $x_i = 0$, $f(x) = 1$ and $f(x^{(i)}) = 0$; (3) all edges in these $2n$ sets are disjoint. Then*

$$\operatorname{dist}\Big(f, \textsc{Unate}\Big) \geq \frac{1}{2^n}\sum_{i=1}^{n} \min\Big\{|E_i^+|, |E_i^-|\Big\}.$$

Proof. By definition, the distance of f to unateness is given by

$$\operatorname{dist}\Big(f, \textsc{Unate}\Big) = \min_{r\in\{0,1\}^n} \operatorname{dist}\Big(f_r, \textsc{Mono}\Big),$$

where $f_r(x) = f(x \oplus r)$. On the other hand, since all edges in the $2n$ sets E_i^+ and E_i^- are disjoint, it follows from Lemma 6.0.1 that

$$\operatorname{dist}\Big(f_r, \textsc{Mono}\Big) \geq \frac{1}{2^n}\left(\sum_{i:r_i=0}\Big|E_i^-\Big| + \sum_{i:r_i=1}\Big|E_i^+\Big|\right) \geq \frac{1}{2^n}\sum_{i=1}^{n}\min\Big\{|E_i^+|, |E_i^-|\Big\}.$$

This finishes the proof of the lemma. □

6.0.2 Tree pruning lemmas

We consider a rather general setup where a q-query deterministic algorithm A has oracle access to an object $\boldsymbol{O}$ drawn from a distribution $\mathcal{D}$: Upon each query w, the oracle with an object O returns $\eta(w, O)$, an element from a finite set $\mathfrak{P}$. Such an algorithm can be equivalently viewed as a tree of depth q, where each internal node u is labelled a query w to make and has $|\mathfrak{P}|$ edges (u, v) leaving u, each labelled a distinct element from $\mathfrak{P}$. (In general the degree of u can be much larger than two; this is the case for all our applications later since we will introduce new oracles that upon a query string $x \in \{0,1\}^n$ returns more information than just $f(x)$.) For this section we do not care about labels of leaves of A. Given A, we present two basic pruning techniques that will help our analysis of algorithms in our lower bound proofs later.

Both lemmas share the following setup. Given A and a set E of edges of A we use L_E to denote the set of leaves ℓ that has at least one edge in E along the path from the root to ℓ. Each lemma below states that if E satisfies certain properties with respect to $\mathcal{D}$ that we are interested in, then

$$\Pr_{\boldsymbol{O}\sim\mathcal{D}}\left[\boldsymbol{O} \text{ reaches a leaf in } L_E\right] = o(1). \tag{6.1}$$

This will later allow us to focus on root-to-leaf paths that do not take any edge in E.

For each node u of tree A, we use $\mathbf{Pr}[u]$ to denote the probability of $\boldsymbol{O} \sim \mathcal{D}$ reaching u. When u is an internal node with $\mathbf{Pr}[u] > 0$ we use $q(u)$ to denote the following conditional probability:

$$q(u) = \Pr_{\boldsymbol{O}\sim\mathcal{D}}\left[\boldsymbol{O} \text{ follows an edge in } E \text{ at } u \,\middle|\, \boldsymbol{O} \text{ reaches } u\right] = \frac{\sum_{(u,v)\in E}\mathbf{Pr}[v]}{\mathbf{Pr}[u]}.$$

We start with the first pruning lemma; it is trivially implied by the second pruning lemma, but we keep it because of its conceptual simplicity.

Lemma 6.0.3. *Given E, if $q(u) = o(1/q)$ for every internal node u with $\mathbf{Pr}[u] > 0$, then (6.1) holds.*

Proof. We can partition the set L_E of leaves into $L_E = \bigcup_{i\in[q]} L_i$, where L_i contains leaves with its first edge from E being the ith edge along its root-to-leaf path. We also write E_i as the set of edges in E at the ith level (i.e., they appear as the ith edge along root-to-leaf paths). Then for each i,

$$\Pr_{\boldsymbol{O}\sim\mathcal{D}}\left[\boldsymbol{O} \text{ reaches } L_i\right] \le \sum_{(u,v)\in E_i} \mathbf{Pr}[v] = \sum_u \sum_{(u,v)\in E_i} \mathbf{Pr}[v] = \sum_u \mathbf{Pr}[u]\cdot o(1/q).$$

Note that the sum is over certain nodes u at the same depth $(i-1)$. Therefore, $\sum_u \mathbf{Pr}[u] \le 1$ and the proof is completed by taking a union bound over L_i, $i \in [q]$. □

Next, for each leaf ℓ with $\mathbf{Pr}[\ell] > 0$ and the root-to-ℓ path being $u_1u_2\cdots u_{k+1} = \ell$, we let $q^*(\ell)$ denote $\sum_{i\in[k]} q(u_i)$. The second pruning lemma states that (6.1) holds if $q^*(\ell) = o(1)$ for all such ℓ.

Lemma 6.0.4. *If every leaf ℓ of A with $\mathbf{Pr}[\ell] > 0$ satisfies $q^*(\ell) = o(1)$, then (6.1) holds.*

Proof. The first part of the proof goes exactly the same as in the proof of the first lemma.

Let A' be the set of internal nodes u with $\mathbf{Pr}[u] > 0$. After a union bound over L_i, $i \in [q]$,

$$\Pr_{\boldsymbol{O}\sim\mathcal{D}}\left[\boldsymbol{O} \text{ reaches } L_E\right] \le \sum_{u\in A'} \mathbf{Pr}[u]\cdot q(u).$$

Let L_u be the leaves in the subtree rooted at $u \in A'$. We can rewrite $\mathbf{Pr}[u]$ as $\sum_{\ell\in L_u} \mathbf{Pr}[\ell]$. Thus,

$$\Pr_{\boldsymbol{O}\sim\mathcal{D}}\left[\boldsymbol{O} \text{ reaches } L_E\right] \le \sum_{u\in A'} \sum_{\ell\in L_u} \mathbf{Pr}[\ell]\cdot q(u) = \sum_\ell \mathbf{Pr}[\ell]\cdot q^*(\ell),$$

where the last sum is over leaves ℓ with $\mathbf{Pr}[\ell] > 0$; the last equation follows by switching the order of the two sums. The lemma follows from $q^*(\ell) = o(1)$ and $\sum_\ell \mathbf{Pr}[\ell] = 1$. □

6.1 Monotonicity Lower Bound

6.1.1 Distributions

For a fixed $n > 0$, we describe a pair of distributions $\mathcal{D}_{\text{yes}}$ and $\mathcal{D}_{\text{no}}$ supported on Boolean functions $f : \{0,1\}^n \to \{0,1\}$. We then show that every $\boldsymbol{f} \sim \mathcal{D}_{\text{yes}}$ is monotone, and $\boldsymbol{f} \sim \mathcal{D}_{\text{no}}$ is $\Omega(1)$-far from monotone with probability $\Omega(1)$. Recall that $N = 2^{\sqrt{n}}$.

A function $\boldsymbol{f} \sim \mathcal{D}_{\text{yes}}$ is drawn using the following procedure:

1. Sample a pair $(\boldsymbol{T}, \boldsymbol{C}) \sim \mathcal{E}$ (which we describe next). The pair $(\boldsymbol{T}, \boldsymbol{C})$ is then used to define
   a *multiplexer* map $\boldsymbol{\Gamma} = \boldsymbol{\Gamma}_{\boldsymbol{T},\boldsymbol{C}} : \{0,1\}^n \to (N \times N) \cup \{0^*, 1^*\}$.[1]
2. Sample $\boldsymbol{H} = (\boldsymbol{h}_{i,j} : i, j \in [N])$ from a distribution $\mathcal{E}_{\text{yes}}$, where each
   $\boldsymbol{h}_{i,j} : \{0,1\}^n \to \{0,1\}$
   is a random dictatorship Boolean function, i.e., $\boldsymbol{h}_{i,j}(x) = x_k$ with k sampled independently for each $\boldsymbol{h}_{i,j}$ and uniformly at random from $[n]$.
3. Finally, $\boldsymbol{f} = \boldsymbol{f}_{\boldsymbol{T},\boldsymbol{C},\boldsymbol{H}} : \{0,1\}^n \to \{0,1\}$ is defined as follows: $\boldsymbol{f}(x) = 1$ if $|x| > (n/2) + \sqrt{n}$; $\boldsymbol{f}(x) = 0$ if $|x| < (n/2) - \sqrt{n}$; if $(n/2) - \sqrt{n} \le |x| \le (n/2) + \sqrt{n}$, we have

$$\boldsymbol{f}(x) = \begin{cases} 0 & \text{if } \boldsymbol{\Gamma}(x) = 0^* \\ 1 & \text{if } \boldsymbol{\Gamma}(x) = 1^* \\ \boldsymbol{h}_{\boldsymbol{\Gamma}(x)}(x) & \text{otherwise (i.e., } \boldsymbol{\Gamma}(x) \in N \times N) \end{cases}$$

On the other hand a function $\boldsymbol{f} = \boldsymbol{f}_{\boldsymbol{T},\boldsymbol{C},\boldsymbol{H}} \sim \mathcal{D}_{\text{no}}$ is drawn using the same procedure, with the only difference being that $\boldsymbol{H} = (\boldsymbol{h}_{i,j} : i, j \in [N])$ is drawn from $\mathcal{E}_{\text{no}}$ (instead of $\mathcal{E}_{\text{yes}}$): each $\boldsymbol{h}_{i,j}(x) = \overline{x_k}$ is a random anti-dictatorship function with k drawn independently and uniformly from $[n]$.

[1] We use 0^* and 1^* to denote two special symbols (instead of the Kleene closure of 0 and 1).

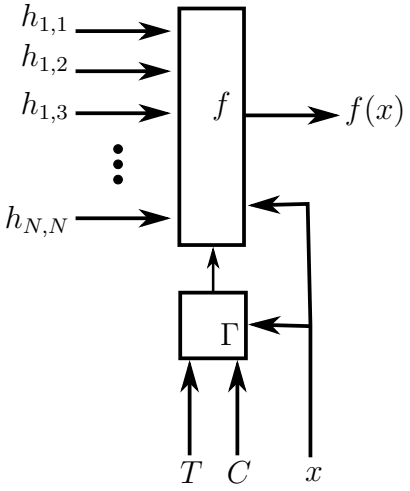

Figure 6.2: An illustration of the function $f = f_{T,C,H}$ and its dependency on T, C and H.

Remark 62. *Given the same truncation done in both $\mathcal{D}_{yes}$ and $\mathcal{D}_{no}$, it suffices to show a lower bound against algorithms that query strings in the* middle layers *only:* $(n/2) - \sqrt{n} \leq |x| \leq (n/2) + \sqrt{n}$.

Next we describe the distribution $\mathcal{E}$ in details. $\mathcal{E}$ is uniform over all pairs (T, C) of the following form: $T = (T_i : i \in [N])$ with $T_i : [\sqrt{n}] \to [n]$ and $C = (C_{i,j} : i, j \in [N])$ with $C_{i,j} : [\sqrt{n}] \to [n]$. We call T_i's the *terms* and $C_{i,j}$'s the *clauses*. Equivalently, to draw a pair $(\boldsymbol{T}, \boldsymbol{C}) \sim \mathcal{E}$:

- For each $i \in [N]$, we sample a random term $\boldsymbol{T}_i$ by sampling $\boldsymbol{T}_i(k)$ independently and uniformly from $[n]$ for each $k \in [\sqrt{n}]$, with $\boldsymbol{T}_i(k)$ viewed as the kth variable of $\boldsymbol{T}_i$.
- For each $i, j \in [N]$, we sample a random clause $\boldsymbol{C}_{i,j}$ by sampling $\boldsymbol{C}_{i,j}(k)$ independently and uniformly from $[n]$ for each $k \in [\sqrt{n}]$, with $\boldsymbol{C}_{i,j}(k)$ viewed as the kth variable of $\boldsymbol{C}_{i,j}$.

Given a pair (T, C), we interpret T_i as a (DNF) term and abuse the notation to write

$$T_i(x) = \bigwedge_{k \in [\sqrt{n}]} x_{T_i(k)}$$

as a Boolean function over n variables. We say x *satisfies* T_i when $T_i(x) = 1$. We

interpret each $C_{i,j}$ as a (CNF) clause and abuse the notation to write

$$C_{i,j}(x) = \bigvee_{k \in [\sqrt{n}]} x_{C_{i,j}(k)}$$

as a Boolean function over n variables. Similarly we say x *falsifies* $C_{i,j}$ when $C_{i,j}(x) = 0$.

Each pair (T, C) in the support of $\mathcal{E}$ defines a multiplexer map $\Gamma = \Gamma_{T,C} : \{0,1\}^n \to (N \times N) \cup \{0^*, 1^*\}$. Informally speaking, Γ consists of two levels: the first level uses the terms T_i in T to pick the first index $i' \in [N]$; the second level uses the clauses $C_{i',j}$ in C to pick the second index $j' \in [N]$. Sometimes Γ may choose to directly determine the value of the function by setting $\Gamma(x) \in \{0^*, 1^*\}$.

Formally, (T, C) defines Γ as follows. Given an $x \in \{0,1\}^n$ we have $\Gamma(x) = 0^*$ if $T_i(x) = 0$ for all $i \in [N]$ and $\Gamma(x) = 1^*$ if $T_i(x) = 1$ for at least two different i's in $[N]$. Otherwise there is a unique i' with $T_{i'}(x) = 1$, and the multiplexer enters the second level. Next, we have $\Gamma(x) = 1^*$ if $C_{i',j}(x) = 1$ for all $j \in [N]$ and $\Gamma(x) = 0^*$ if $C_{i',j}(x) = 0$ for at least two different j's in $[N]$. Otherwise there is a unique $j' \in [N]$ with $C_{i',j'}(x) = 0$ and in this case the multiplexer outputs $\Gamma(x) = (i', j')$.

This finishes the definition of $\mathcal{D}_{\text{yes}}$ and $\mathcal{D}_{\text{no}}$. Figure 6.3 above gives a graphical representation of such functions. We now prove the properties of $\mathcal{D}_{\text{yes}}$ and $\mathcal{D}_{\text{no}}$ promised at the beginning.

Lemma 6.1.1. *Every function f in the support of $\mathcal{D}_{yes}$ is monotone.*

Proof. Consider $f = f_{T,C,H}$ with (T, C) from the support of $\mathcal{E}$ and H from the support of $\mathcal{E}_{\text{yes}}$. Let $x \in \{0,1\}^n$ be a string with $f(x) = 1$ and $x_i = 0$ for some i. Let $x' = x^{(i)}$. We show that $f(x') = 1$.

First note that every term in T satisfied by x remains satisfied by x'; every clause satisfied by x remains satisfied by x'. As a result if $\Gamma(x) = 1^*$ then $\Gamma(x') = 1^*$ as well. Assume that $\Gamma(x) = (i, j)$. Then $h_{i,j}(x) = f(x) = 1$. For this case we have either $\Gamma(x') = 1^*$ and $f(x') = 1$, or $f(x') = h_{i,j}(x')$ and $h_{i,j}(x') = h_{i,j}(x) = 1$ because $h_{i,j}$ here is a dictatorship function. □

Lemma 6.1.2. *A function $\boldsymbol{f} \sim \mathcal{D}_{no}$ is $\Omega(1)$-far-from monotone with probability $\Omega(1)$.*

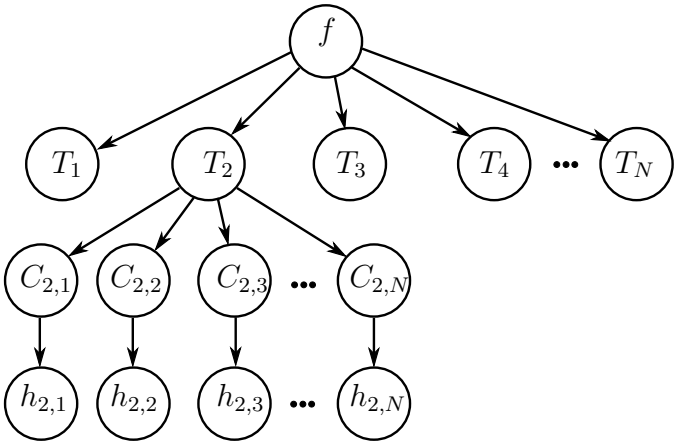

Figure 6.3: Picture of a function f in the support of $\mathcal{D}_{\text{yes}}$ and $\mathcal{D}_{\text{no}}$. We think of evaluating $f(x)$ as following the arrows down the tree. The first level represents multiplexing $x \in \{0,1\}^n$ with respect to the terms in T. If x satisfies no terms, or multiple terms, then f outputs 0, or 1, respectively. If x satisfies T_i for a *unique* term T_i (T_2 in the picture), then we follow the arrow to T_i and proceed to the second level. If x falsifies no clause, or multiple clauses, then f outputs 1, or 0, respectively. If x falsifies a unique clause $C_{i,j}$, then we follow the arrow to $C_{i,j}$ and output $h_{i,j}(x)$.

Proof. Fix a pair (T, C) from the support of $\mathcal{E}$ and an H from the support of $\mathcal{E}_{\text{no}}$. Let $f = f_{T,C,H}$.

Consider the set $X \subset \{0,1\}^n$ consisting of strings x in the middle layers (i.e., $|x| \in (n/2) \pm \sqrt{n}$) with $f(x) = 1$, $\Gamma(x) = (i,j)$ for some $i, j \in [N]$ (instead of 0^* or 1^*), and $h_{i,j}$ being an anti-dictator function on the kth variable for some $k \in [n]$ (so $x_k = 0$). For each $x \in X$, we write $\eta(x)$ to denote the anti-dictator variable k in $h_{i,j}$ and use x^* to denote $x^{(\eta(x))}$. (Ideally, we would like to conclude that (x, x^*) is a violating edge of f as $h_{i,j}(x^*) = 0$. However, flipping one bit potentially may also change the value of the multiplexer map Γ. So we need to further refine the set X.)

Next we define the following two events with respect to a string $x \in X$ (with $\Gamma(x) = (i,j)$):

- $E_1(x)$: This event occurs when $\eta(x) \neq C_{i,j}(\ell)$ for any $\ell \in [\sqrt{n}]$ (and thus, $C_{i,j}(x^*) = 0$);
- $E_2(x)$: This event occurs when $T_{i'}(x^*) = 0$ for all $i' \neq i \in [N]$.

We use X' to denote the set of strings $x \in X$ such that both $E_1(x)$ and $E_2(x)$ hold. The

following claim shows that (x, x^*) for every $x \in X'$ is a violating edge of f.

Claim 6.1.3. *For each $x \in X'$, (x, x^*) is a violating edge of f.*

Proof. It suffices to show that $f(x^*) = 0$. As x satisfies a unique term T_i (T_i cannot have $\eta(x)$ as a variable because $x_{\eta(x)} = 0$), it follows from $E_2(x)$ that x^* uniquely satisfies the same T_i. It follows from $E_1(x)$ that x^* uniquely falsifies the same clause $C_{i,j}$. As a result, $f(x^*) = h_{i,j}(x^*) = 0$. □

Furthermore, the violating edges (x, x^*) induced by strings $x \in X'$ are indeed disjoint. (This is because, given x^*, one can uniquely reconstruct x by locating $h_{i,j}$ using $\Gamma(x^*)$ and flipping the kth bit of x^* if $h_{i,j}$ is an anti-dictator function over the kth variable.) Therefore, it suffices to show that $\boldsymbol{X}'$ (as a random set) has size $\Omega(2^n)$ with probability $\Omega(1)$, over choices $(\boldsymbol{T}, \boldsymbol{C}) \sim \mathcal{E}$ and $\boldsymbol{H} \sim \mathcal{E}_{\text{no}}$. The lemma then follows from the characterization of [63] as stated in Lemma 6.0.1.

Finally we work on the size of $\boldsymbol{X}'$. Fix a string $x \in \{0,1\}^n$ in the middle layers. The next claim shows that, when $(\boldsymbol{T}, \boldsymbol{C}) \sim \mathcal{E}$ and $\boldsymbol{H} \sim \mathcal{E}_{\text{no}}$, $\boldsymbol{X}'$ contain x with $\Omega(1)$ probability.

Claim 6.1.4. *For each $x \in \{0,1\}^n$ with $(n/2) - \sqrt{n} \le |x| \le (n/2) + \sqrt{n}$, we have*

$$\Pr_{(\boldsymbol{T},\boldsymbol{C})\sim\mathcal{E},\, \boldsymbol{H}\sim\mathcal{E}_{no}} \left[x \in \boldsymbol{X}' \right] = \Omega(1).$$

Proof. Fix an $x \in \{0,1\}^n$ in the middle layers. We calculate the probability of $x \in \boldsymbol{X}'$.

We partition the event of $x \in \boldsymbol{X}'$ into $\Theta(nN^2)$ subevents indexed by $i, j \in [N]$ and $k \in [n]$ with $x_k = 0$. Each subevent corresponds to 1) Condition on $\boldsymbol{T}$: both x and $x^{(k)}$ satisfy uniquely the ith term; 2) Condition on $\boldsymbol{C}$: both x and $x^{(k)}$ falsify uniquely the jth term; 3) Condition on $\boldsymbol{H}$: $h_{i,j}$ is the anti-dictatorship function over the kth variable. The probability of 3) is clearly $1/n$.

The probability of 1) is at least

$$\left(1 - \left(\frac{n/2 + \sqrt{n} + 1}{n}\right)^{\sqrt{n}}\right)^{N-1} \times \left(\frac{n/2 - \sqrt{n}}{n}\right)^{\sqrt{n}} = \Omega\left(\frac{1}{N}\right).$$

The probability of 2) is at least

$$\left(1-\left(\frac{n/2+\sqrt{n}}{n}\right)^{\sqrt{n}}\right)^{N-1}\times\left(\frac{n/2-\sqrt{n}+1}{n}\right)^{\sqrt{n}}=\Omega\left(\frac{1}{N}\right).$$

As a result, the probability of $x \in \boldsymbol{X}'$ is $\Omega(nN^2)\times\Omega(1/N)\times\Omega(1/N)\times\Omega(1/n)=\Omega(1)$. □

From Claim 6.1.4 and the fact that there are $\Omega(2^n)$ strings in the middle layer, the expected size of $\boldsymbol{X}'$ is $\Omega(2^n)$. Via Markov, $|\boldsymbol{X}'| = \Omega(2^n)$ with probability $\Omega(1)$. This finishes the proof. □

Given Lemma 6.1.1 and 6.1.2, Theorem 58 follows directly from the following lemma which we show in the rest of the section. For the rest of the proof we fix the number of queries $q = n^{1/3}/\log^2 n$.

Lemma 6.1.5. *Let B be any q-query, deterministic algorithm with oracle access to f. Then*

$$\Pr_{\boldsymbol{f}\sim\mathcal{D}_{yes}}\left[B \text{ accepts } \boldsymbol{f}\right] \leq \Pr_{\boldsymbol{f}\sim\mathcal{D}_{no}}\left[B \text{ accepts } \boldsymbol{f}\right] + o(1).$$

Since f is truncated in both distributions, we may assume WLOG that B queries strings in the middle layers only (i.e., strings x with $|x|$ between $(n/2)-\sqrt{n}$ and $(n/2)+\sqrt{n}$).

6.1.2 Signatures and the new oracle

Let (T, C) be a pair from the support of $\mathcal{E}$ and H be a tuple from the support of $\mathcal{E}_{\text{yes}}$ or $\mathcal{E}_{\text{no}}$. Towards Lemma 6.1.5, we are interested in deterministic algorithms that have oracle access to $f = f_{T,C,H}$ and attempt to distinguish $\mathcal{D}_{\text{yes}}$ from $\mathcal{D}_{\text{no}}$ (i.e., accept if H is from $\mathcal{E}_{\text{yes}}$ and reject if it is from $\mathcal{E}_{\text{no}}$).

For convenience of our lower bound proof, we assume below that the oracle returns more than just $f(x)$ for each query $x \in \{0,1\}^n$; instead of simply returning $f(x)$, the oracle returns a 4-tuple (σ, τ, a, b) called the *full signature* of $x \in \{0,1\}^n$ with respect to (T, C, H) (see Definition 43 below). It will become clear later that $f(x)$ can always be derived correctly from the full signature of x and thus, query lower bounds against the new oracle carry over to the standard oracle. Once the new oracle is introduced, we may actually

ignore the function f and view any algorithm as one that has oracle access to the hidden triple (T, C, H) and attempts to tell whether H is from $\mathcal{E}_{\text{yes}}$ or $\mathcal{E}_{\text{no}}$.

We first give the syntactic definition of *full signatures*.

Definition 42. *We use $\mathfrak{P}$ to denote the set of all 4-tuples (σ, τ, a, b) with $\sigma \in \{0,1,*\}^N$ and $\tau \in \{0,1,*\}^N \cup \{\perp\}$ and $a, b \in \{0,1,\perp\}$ satisfying the following properties:*

1. *σ is either 1) the all-0 string 0^N; 2) e_i for some $i \in [N]$; or 3) $e_{i,i'}$ for some $i < i' \in [N]$.*
2. *$\tau = \perp$ if σ is of case 1) or 3). Otherwise (when $\sigma = e_i$ for some i), $\tau \in \{0,1,*\}^N$ is either 1) the all-1 string 1^N; 2) $\overline{e}_j$ for some $j \in [N]$; or 3) $\overline{e}_{j,j'}$ for some $j < j' \in [N]$.*
3. *$a = b = \perp$ unless: 1) If $\sigma = e_i$ and $\tau = \overline{e}_j$ for some $i, j \in [N]$, then $a \in \{0,1\}$ and $b = \perp$; or 2) If $\sigma = e_i$ and $\tau = \overline{e}_{j,j'}$ for some $i \in [N]$ and $j < j' \in [N]$, then $a, b \in \{0,1\}$.*

We next define semantically the full signature of $x \in \{0,1\}^n$ with respect to (T, C, H).

Definition 43 (Full signature)**.** *We say (σ, τ, a, b) is the* full signature *of a string $x \in \{0,1\}^n$ with respect to (T, C, H) if it satisfies the following properties:*

1. *First, $\sigma \in \{0,1,*\}^N$ is determined by T according to one of the following three cases: 1) σ is the all-0 string 0^N if $T_i(x) = 0$ for all $i \in [N]$; 2) If there is a unique $i \in [N]$ with $T_i(x) = 1$, then $\sigma = e_i$; or 3) If there are more than one index $i \in [N]$ with $T_i(x) = 1$, then $\sigma = e_{i,i'}$ with $i < i' \in [N]$ being the smallest two such indices. We call σ the* term signature *of x.*
2. *Second, $\tau = \perp$ if σ is of case 1) or 3) above. Otherwise, assuming that $\sigma = e_i$, $\tau \in \{0,1,*\}^N$ is determined by $(C_{i,j} : j \in [N])$, according to one of the following cases: 1) τ is the all-1 string 1^N if $C_{i,j}(x) = 1$ for all $j \in [N]$; 2) If there is a unique $j \in [N]$ with $C_{i,j}(x) = 0$, then $\tau = \overline{e}_j$; or 3) If there are more than one index $j \in [N]$ with $C_{i,j}(x) = 0$, then $\tau = \overline{e}_{j,j'}$ with $j < j' \in [N]$ being the smallest two such indices. We call τ the* clause signature *of x.*

3. *Finally, $a = b = \perp$ unless: 1) If $\sigma = e_i$ and $\tau = \overline{e}_j$ for some $i, j \in [N]$, then $a = h_{i,j}(x)$ and $b = \perp$; or 2) If $\sigma = e_i$ and $\tau = \overline{e}_{j,j'}$ for some $i, j < j' \in [N]$, then $a = h_{i,j}(x)$ and $b = h_{i,j'}(x)$.*

It follows from the definitions that the full signature of x with respect to (T, C, H) is in $\mathfrak{P}$. We also define the *full signature* of a set of strings Q with respect to (T, C, H).

Definition 44. *The* full signature (*map*) *of a set $Q \subseteq \{0,1\}^n$ with respect to a triple (T, C, H) is a map $\phi\colon Q \to \mathfrak{P}$ such that $\phi(x)$ is the full signature of x with respect to (T, C, H) for each $x \in Q$.*

For simplicity, we will write $\phi(x) = (\sigma_x, \tau_x, a_x, b_x)$ to specify the term and clause signatures of x as well as the values of a and b in the full signature $\phi(x)$ of x. Intuitively we may view ϕ as two levels of tables with entries in $\{0, 1, *\}$. The (unique) top-level table "stacks" the term signatures σ_x, where each row corresponds to a string $x \in Q$ and each column corresponds to a term T_i in T. In the second level a table appears for a term T_i if the term signature of some string $x \in Q$ is e_i. In this case the second-level table at T_i "stacks" the clause signatures τ_x for each $x \in Q$ with $\sigma_x = e_i$ where each row corresponds to such an x and each column corresponds to a clause $C_{i,j}$ in C. (The number of columns is still N since we only care about clauses $C_{i,j}$, $j \in [N]$, in the table at T_i.)

The lemma below shows that the new oracle is at least as powerful as the standard oracle.

Lemma 6.1.6. *Let (T, C) be from the support of $\mathcal{E}$ and H from the support of $\mathcal{E}_{yes}$ or $\mathcal{E}_{no}$. Given any string $x \in \{0,1\}^n$, $f_{T,C,H}(x)$ is determined by its full signature with respect to (T, C, H).*

Proof. First if x does not lie in the middle layers, then $f(x)$ is determined by $|x|$. Below we assume that x lies in the middle layers. Let (σ, τ, a, b) be the full signature of x. There are five cases:

1. (No term satisfied) If $\sigma = 0^N$, then $f(x) = 0$.
2. (Multiple terms satisfied) If $\sigma = e_{i,i'}$ for some $i, i' \in [N]$, then $f(x) = 1$.
3. (Unique term satisfied, no clause falsified) If $\sigma = e_i$ but $\tau = 1^N$, then $f(x) = 1$.

4. (Unique term satisfied, multiple clauses falsified) If $\sigma = e_i$ but $\tau = \overline{e}_{j,j'}$, then $f(x) = 0$.
5. (Unique term satisfied, unique clause satisfied) If $\sigma = e_i$ and $\tau = \overline{e}_j$, then $f(x) = a$.

This finishes the proof of the lemma. □

Given Lemma 6.1.6, it suffices to consider deterministic algorithms with the new oracle access to a hidden triple (T, C, H), and Lemma 6.1.5 follows directly from the following lemma:

Lemma 6.1.7. *Let B be any q-query algorithm with the new oracle access to (T, C, H). Then*

$$\Pr_{(\boldsymbol{T},\boldsymbol{C})\sim\mathcal{E},\boldsymbol{H}\sim\mathcal{E}_{yes}}\Big[B \text{ accepts } (\boldsymbol{T},\boldsymbol{C},\boldsymbol{H})\Big] \leq \Pr_{(\boldsymbol{T},\boldsymbol{C})\sim\mathcal{E},\boldsymbol{H}\sim\mathcal{E}_{no}}\Big[B \text{ accepts } (\boldsymbol{T},\boldsymbol{C},\boldsymbol{H})\Big] + o(1).$$

Such a deterministic algorithm B can be equivalently viewed as a decision tree of depth q (and we will abuse the notation to also denote this tree by B). Each leaf of the tree B is labeled either "accept" or "reject." Each internal node u of B is labeled with a query string $x \in \{0,1\}^n$, and each of its outgoing edges (u, v) is labeled a tuple from $\mathfrak{P}$. We refer to such a tree as a *signature tree*.

As the algorithm executes, it traverses a root-to-leaf path down the tree making queries to the oracle corresponding to queries in the nodes on the path. For instance at node u, after the algorithm queries x and the oracle returns the full signature of x with respect to the unknown (T, C, H), the algorithm follows the outgoing edge (u, v) with that label. Once a leaf ℓ is reached, B accepts if ℓ is labelled "accept" and rejects otherwise.

Note that the number of children of each internal node is $|\mathfrak{P}|$, which is huge. Algorithms with the new oracle may adapt its queries to the full signatures returned by the oracle, while under the standard oracle, the queries may only adapt to the value of the function at previous queries. Thus, while algorithms making q queries in the standard oracle model can be described by a tree of size 2^q, q-query algorithms with this new oracle are given by signature trees of size $(2^{\Theta(\sqrt{n})})^q$.

We associate each node u in the tree B with a map $\phi_u : Q_u \to \mathfrak{P}$ where Q_u is the set of queries made along the path from the root to u so far, and $\phi_u(x)$ is the label of the edge that the root-to-u path takes after querying x. We will be interested in analyzing the following two quantities:

$$\mathop{\mathbf{Pr}}_{(\boldsymbol{T},\boldsymbol{C})\sim\mathcal{E},\boldsymbol{H}\sim\mathcal{E}_{\text{yes}}}\Big[(\boldsymbol{T},\boldsymbol{C},\boldsymbol{H}) \text{ reaches } u\Big] \quad \text{and} \quad \mathop{\mathbf{Pr}}_{(\boldsymbol{T},\boldsymbol{C})\sim\mathcal{E},\boldsymbol{H}\sim\mathcal{E}_{\text{no}}}\Big[(\boldsymbol{T},\boldsymbol{C},\boldsymbol{H}) \text{ reaches } u\Big].$$

In particular, Lemma 6.1.7 would follow trivially if for every leaf ℓ of B:

$$\mathop{\mathbf{Pr}}_{(\boldsymbol{T},\boldsymbol{C})\sim\mathcal{E},\boldsymbol{H}\sim\mathcal{E}_{\text{yes}}}\Big[(\boldsymbol{T},\boldsymbol{C},\boldsymbol{H}) \text{ reaches } \ell\Big] \leq (1+o(1)) \cdot \mathop{\mathbf{Pr}}_{(\boldsymbol{T},\boldsymbol{C})\sim\mathcal{E},\boldsymbol{H}\sim\mathcal{E}_{\text{no}}}\Big[(\boldsymbol{T},\boldsymbol{C},\boldsymbol{H}) \text{ reaches } \ell\Big]. \tag{6.2}$$

However, (6.2) above does not hold in general. Our plan for the rest of the proof is to prune an $o(1)$-fraction of leaves (measured in terms of their total probability under the yes-case) and show (6.2) for the rest. To better understand these probabilities, we need to first introduce some useful notation.

6.1.3 Notation for full signature maps

Given a map $\phi : Q \to \mathfrak{P}$ for some $Q \subseteq \{0,1\}^n$, we write $\phi(x) = (\sigma_x, \tau_x, a_x, b_x)$ for each $x \in Q$ and use $\sigma_{x,i}, \tau_{x,j}$ to denote the ith entry and jth entry of σ_x and τ_x, respectively. Note that $\tau_{x,j}$ is not defined if $\tau_x = \perp$. (Below we will only be interested in $\tau_{x,j}$ if $\sigma_x = e_i$ for some $i \in [N]$.)

We introduce the following notation for ϕ. We say ϕ *induces a tuple* $(I; J; P; R; A; \rho)$, where

- The set $I \subseteq [N]$ is given by $I = \{i \in [N] : \exists x \in Q \text{ with } \sigma_{x,i} = 1\}$. (So in terms of the first-level table, I consists of columns that contain at least one 1-entry.)
- $J = (J_i \subseteq [N] : i \in I)$ is a tuple of sets indexed by $i \in I$. For each $i \in I$, we have

$$J_i = \Big\{j \in [N] : \exists x \in Q \text{ with } \sigma_x = e_i \text{ and } \tau_{x,j} = 0\Big\}.$$

(In terms of the second-level table at T_i, J_i consists of columns that contain at least

one 0-entry.) By the definition of $\mathfrak{P}$, each x with $\sigma_x = e_i$ can contribute at most two j's to J_i. Also x does not contribute any j to J_i if $\sigma_x = e_{i,i'}$ or $e_{i',i}$, in which case $\tau_x = \perp$, or if $\sigma_x = e_i$ but $\tau_x = 1^N$. So in general J_i can be empty for some $i \in I$.

- $P = (P_i, P_{i,j} : i \in I, j \in J_i)$ is a tuple of two types of subsets of Q. For $i \in I$ and $j \in J_i$,

$$P_i = \Big\{x \in Q : \sigma_{x,i} = 1\Big\} \quad \text{and} \quad P_{i,j} = \Big\{x \in Q : \sigma_x = e_i \text{ and } \tau_{x,j} = 0\Big\}.$$

  (In terms of the first-level table, P_i consists of rows that are 1 on the ith column; in terms of the second-level table at T_i, $P_{i,j}$ consists of rows that are 0 on the jth column.) Note that both P_i and $P_{i,j}$ are not empty by the definition of I and J_i.

- $R = (R_i, R_{i,j} : i \in I, j \in J_i)$ is a tuple of two types of subsets of Q. For $i \in I$ and $j \in J_i$,

$$R_i = \Big\{x \in Q : \sigma_{x,i} = 0\Big\} \quad \text{and} \quad R_{i,j} = \Big\{x \in Q : \sigma_x = e_i \text{ and } \tau_{x,j} = 1\Big\}.$$

  (In terms of the first-level table, R_i consists of rows that are 0 on the ith column; in terms of the second-level table at T_i, $R_{i,j}$ consists of rows that are 1 on the jth column.)

- $A = (A_{i,0}, A_{i,1}, A_{i,j,0}, A_{i,j,1} : i \in I, j \in J_i)$ is a tuple of subsets of $[n]$. For $i \in I$ and $j \in J_i$,

$$A_{i,1} = \Big\{k \in [n] : \forall x \in P_i,\, x_k = 1\Big\} \quad \text{and} \quad A_{i,0} = \Big\{k \in [n] : \forall x \in P_i,\, x_k = 0\Big\}$$
$$A_{i,j,1} = \Big\{k \in [n] : \forall x \in P_{i,j},\, x_k = 1\Big\} \quad \text{and} \quad A_{i,j,0} = \Big\{k \in [n] : \forall x \in P_{i,j},\, x_k = 0\Big\}.$$

  Note that all the sets are well-defined since P_i and $P_{i,j}$ are not empty.

- $\rho = (\rho_{i,j} : i \in I, j \in J_i)$ is a tuple of functions $\rho_{i,j} : P_{i,j} \to \{0,1\}$. For each $x \in P_{i,j}$, we have $\rho_{i,j}(x) = a_x$ if $\tau_x = \overline{e}_j$ or $\tau_x = \overline{e}_{j,j'}$ for some $j' > j$; $\rho_{i,j}(x) = b_x$ if $\tau_x = \overline{e}_{j',j}$ for some $j' < j$.

Intuitively I is the set of indices of terms with some string $x \in Q$ satisfying the term T_i

as reported in σ_x, and P_i is the set of such strings while R_i is the set of strings which do not satisfy T_i. For each $i \in I$, J_i is the set of indices of clauses with some string $x \in P_i$ satisfying T_i *uniquely* and falsifying the clause $C_{i,j}$. $P_{i,j}$ is the set of such strings, and $R_{i,j}$ is the set of strings which satisfy T_i uniquely but also satisfy $C_{i,j}$. We collect the following facts which are immediate from the definition.

Fact 6.1.8. *Let $(I; J; P; R; A; \rho)$ be the tuple induced by a map $\phi\colon Q \to \Sigma$. Then we have*

- $|I| \leq \sum_{i \in I} |P_i| \leq 2|Q|$.
- *For each $i \in I$, $|J_i| \leq \sum_{j \in J_i} |P_{i,j}| \leq 2|P_i|$.*
- *For each $i \in I$ and $j \in J_i$, $|R_i|$ and $|R_{i,j}|$ are at most $|Q|$ (as they are subsets of Q).*
- *For each $i \in I$ and $j \in J_i$, $P_{i,j} \subseteq P_i$, $A_{i,0} \subseteq A_{i,j,0}$, and $A_{i,1} \subseteq A_{i,j,1}$.*

Note that $|I|$ and $\sum_{i \in I} |J_i|$ can be strictly larger than $|Q|$, as some x may satisfy more than one (but at most two) term with $\sigma_x = e_{i,i'}$ and some x may falsify more than one clause with $\tau_x = \overline{e}_{j,j'}$.

The sets in A are important for the following reasons that we summarize below.

Fact 6.1.9. *Let $\phi : Q \to \mathfrak{P}$ be the full signature map of Q with respect to (T, C, H). Then*

- *For each $i \in I$, $T_i(k) \in A_{i,1}$ for all $k \in [\sqrt{n}]$ and $T_i(x) = 0$ for each $x \in R_i$.*
- *For each $i \in I$ and $j \in J_i$, $C_{i,j}(k) \in A_{i,j,0}$ for all $k \in [\sqrt{n}]$ and $C_{i,j}(x) = 1$ for each $x \in R_{i,j}$.*

Before moving back to the proof, we introduce the following consistency condition on P.

Definition 45. *Let $(I; J; P; R; A; \rho)$ be the tuple induced by a map $\phi : Q \to \mathfrak{P}$. We say that $P_{i,j}$ for some $i \in I$ and $j \in J_i$ is* 1-consistent *if $\rho_{i,j}(x) = 1$ for all $x \in P_{i,j}$, and* 0-consistent *if $\rho_{i,j}(x) = 0$ for all $x \in P_{i,j}$; otherwise we say $P_{i,j}$ is* inconsistent.

Let ϕ be the full signature map of Q with respect to (T, C, H). If $P_{i,j}$ is 1-consistent, the index k of the variable x_k in the dictatorship or anti-dictatorship function $h_{i,j}$ must lie in $A_{i,j,0}$ (when $h_{i,j}$ is an anti-dictator) or $A_{i,j,1}$ (when $h_{i,j}$ is a dictator); the situation is similar if $P_{i,j}$ is 0-consistent but would be more complicated if $P_{i,j}$ is inconsistent. Below we prune

an edge whenever some $P_{i,j}$ in P becomes inconsistent. This way we make sure that $P_{i,j}$'s in every leaf left are consistent.

6.1.4 Tree pruning

Consider an edge (u, v) in B. Let $\phi_u \colon Q \to \mathfrak{P}$ and $\phi_v \colon Q \cup \{x\} \to \mathfrak{P}$ be the maps associated with u and v, with x being the query made at u and $\phi_v(x)$ being the label of (u, v). Let $(I; J; P; R; A; \rho)$ and $(I'; J'; P'; R'; A'; \rho')$ be the two tuples induced by ϕ_u and ϕ_v, respectively.

We list some easy facts about how $(I; J; P; R; A; \rho)$ is updated to obtain $(I'; J'; P'; R'; A'; \rho')$.

Fact 6.1.10. *Let $\phi_v(x) = (\sigma_x, \tau_x, a_x, b_x)$ for the string x queried at u. Then we have*

- *The new string x is placed in P'_i if $\sigma_{x,i} = 1$, and is placed in $P'_{i,j}$ if $\sigma_x = e_i$ and $\tau_{x,j} = 0$.*
- *Each new set in P' (i.e., P'_i with $i \notin I$ or $P'_{i,j}$ with either $i \notin I$ or $i \in I$ but $j \notin J_i$), if any, is $\{x\}$ and the corresponding $A'_{i,1}$ or $A'_{i,j,1}$ is $\{k : x_k = 1\}$ and $A'_{i,0}$ or $A'_{i,j,0}$ is $\{k : x_k = 0\}$.*
- *Each old set in P' (i.e., P'_i with $i \in I$ or $P'_{i,j}$ with $i \in I$ and $j \in J_i$) either stays the same or has x being added to the set. For the latter case, $\{k : x_k = 0\}$ is removed from $A_{i,1}$ or $A_{i,j,1}$ and $\{k : x_k = 1\}$ is removed from $A_{i,0}$ or $A_{i,j,0}$ to obtain the new sets in A'.*

Now we are ready to define a set of so-called *bad* edges of B, which will be used to prune B. In the rest of the proof we use α to denote a large enough positive constant.

Definition 46. *An edge (u, v) is called a* bad *edge if at least one of the following events occur at (u, v) and none of these events occur along the path from the root to u (letting ϕ_u and ϕ_v be the maps associated with u and v, x be the new query string at u, $(I; J; P; R; A; \rho)$ and $(I'; J'; P'; R'; A'; \rho')$ be the tuples that ϕ_u and ϕ_v induce, respectively):*

- *For some $i \in I$, $\left|A_{i,1} \setminus A'_{i,1}\right| \geq \alpha\sqrt{n}\log n$.*
- *For some $i \in I$ and $j \in J_i$, $\left|A_{i,j,0} \setminus A'_{i,j,0}\right| \geq \alpha\sqrt{n}\log n$.*

- *For some $i \in I$ and $j \in J_i$, $P_{i,j}$ is* 0*-consistent but $P'_{i,j}$ is inconsistent* (*meaning that x is added to $P_{i,j}$ with $\rho_{i,j}(y) = 0$ for all $y \in P_{i,j}$ but $\rho'_{i,j}(x) = 1$, instead of* 0*).*
- *For some $i \in I$ and $j \in J_i$, $P_{i,j}$ is* 1*-consistent but $P'_{i,j}$ is inconsistent* (*meaning that x is added to $P_{i,j}$ with $\rho_{i,j}(y) = 1$ for all $y \in P_{i,j}$ but $\rho'_{i,j}(x) = 0$, instead of* 1*).*

Moreover, a leaf ℓ is bad *if one of the edges along the root-to-ℓ path is bad; ℓ is* good *otherwise.*

The following pruning lemma states that the probability of $(\boldsymbol{T}, \boldsymbol{C}, \boldsymbol{H})$ reaching a bad leaf of B is $o(1)$, when $(\boldsymbol{T}, \boldsymbol{C}) \sim \mathcal{E}$ and $\boldsymbol{H} \sim \mathcal{E}_{\text{yes}}$. We delay the proof to Section 6.1.6.

Lemma 6.1.11 (Pruning Lemma)**.** $\mathbf{Pr}_{(\boldsymbol{T},\boldsymbol{C})\sim\mathcal{E},\boldsymbol{H}\sim\mathcal{E}_{yes}}\Big[(\boldsymbol{T}, \boldsymbol{C}, \boldsymbol{H})$ *reaches a bad leaf of* $B\Big] = o(1)$.

The pruning lemma allow us to focus on the good leaves ℓ of B only. In particular we know that along the root-to-ℓ path the sets $A_{i,1}$ and $A_{i,j,0}$ each cannot shrink by more than $\alpha\sqrt{n}\log n$ with a single query (otherwise the path contains a bad edge and ℓ is a bad leaf which we ignore). Moreover every set $P_{i,j}$ in P at the end must remain consistent (either 0-consistent or 1-consistent).

We use these properties to prove the following lemma in Section 6.1.5 for good leaves of B.

Lemma 6.1.12 (Good Leaves are Nice)**.** *For each good leaf ℓ of B, we have*

$$\mathop{\mathbf{Pr}}_{(\boldsymbol{T},\boldsymbol{C})\sim\mathcal{E},\boldsymbol{H}\sim\mathcal{E}_{yes}}\Big[(\boldsymbol{T}, \boldsymbol{C}, \boldsymbol{H}) \text{ reaches } \ell\Big] \leq (1 + o(1)) \cdot \mathop{\mathbf{Pr}}_{(\boldsymbol{T},\boldsymbol{C})\sim\mathcal{E},\boldsymbol{H}\sim\mathcal{E}_{no}}\Big[(\boldsymbol{T}, \boldsymbol{C}, \boldsymbol{H}) \text{ reaches } \ell\Big].$$

We can now combine Lemma 6.1.11 and Lemma 6.1.12 to prove Lemma 6.1.7.

Proof of Lemma 6.1.7. Let L be the leaves labeled "accept," and $L^* \subset L$ be the good leaves labeled "accept." Below we ignore $(\boldsymbol{T}, \boldsymbol{C}) \sim \mathcal{E}$ in the subscript since it appears in every

probability.

$$\begin{aligned}
\Pr_{\boldsymbol{H}\sim\mathcal{E}_{\text{yes}}}\Big[B \text{ accepts } (\boldsymbol{T},\boldsymbol{C},\boldsymbol{H})\Big] &= \sum_{\ell\in L}\Pr_{\boldsymbol{H}\sim\mathcal{E}_{\text{yes}}}\Big[(\boldsymbol{T},\boldsymbol{C},\boldsymbol{H}) \text{ reaches } \ell\Big] \\
&\le \sum_{\ell\in L^*}\Pr_{\boldsymbol{H}\sim\mathcal{E}_{\text{yes}}}\Big[(\boldsymbol{T},\boldsymbol{C},\boldsymbol{H}) \text{ reaches } \ell\Big] + o(1) \\
&\le (1+o(1))\cdot\sum_{\ell\in L^*}\Pr_{\boldsymbol{H}\sim\mathcal{E}_{\text{no}}}\Big[(\boldsymbol{T},\boldsymbol{C},\boldsymbol{H}) \text{ reaches } \ell\Big] + o(1) \\
&\le \Pr_{\boldsymbol{H}\sim\mathcal{E}_{\text{no}}}\Big[B \text{ accepts } (\boldsymbol{T},\boldsymbol{C},\boldsymbol{H})\Big] + o(1),
\end{aligned}$$

where the second line used Lemma 6.1.11 and the third line used Lemma 6.1.12. □

6.1.5 Proof of Lemma 6.1.12 for good leaves

We prove Lemma 6.1.12 in this section. Let ℓ be a good leaf associated with ϕ_ℓ and $(I; J; P; R; A; \rho)$ be the tuple that ϕ_ℓ induces. Note that along the root-to-ℓ path, when a set $A_{i,0}, A_{i,1}, A_{i,j,0}, A_{i,j,1}$ is created for the first time in A, its size is between $(n/2) \pm \sqrt{n}$ (since all queries made by B lie in the middle layers). As a result, it follows from Definition 46 that for $i \in I$ and $j \in J_i$:

i) $|A_{i,1}| \ge (n/2) - O(|P_i| \cdot \sqrt{n}\log n)$ and $|A_{i,j,0}| \ge (n/2) - O(|P_{i,j}| \cdot \sqrt{n}\log n)$;

ii) $|A_{i,0}|, |A_{i,1}|, |A_{i,j,0}|, |A_{i,j,1}| \le (n/2) + \sqrt{n}$;

iii) $P_{i,j}$ is consistent (either 1-consistent or 0-consistent).

We start with the following claim:

Claim 6.1.13. *For each $i \in I$ and $j \in J_i$, $|A_{i,j,1}| \ge (n/2) - O\Big(|P_{i,j}|^2 \cdot \sqrt{n}\log n\Big)$.*

Proof. For any two strings $x, y \in P_{i,j}$, we have

$$\Big|\{k \in [n] : x_k = y_k = 0\}\Big| \ge |A_{i,j,0}| \ge (n/2) - O\Big(|P_{i,j}| \cdot \sqrt{n}\log n\Big).$$

As a result, it follows from $|\{k : y_k = 0\}| \le (n/2) + \sqrt{n}$ and $P_{i,j}$ being nonempty that

$$\Big|\{k \in [n] : x_k = 1, y_k = 0\}\Big| \le O\Big(|P_{i,j}| \cdot \sqrt{n}\log n\Big).$$

Finally we have

$$|A_{i,j,1}| \geq \Big|\{k : x_k = 1\}\Big| - \sum_{y \in P_{i,j} \setminus \{x\}} \Big|\{k : x_k = 1, y_k = 0\}\Big| \geq (n/2) - O\Big(|P_{i,j}|^2 \cdot \sqrt{n} \log n\Big). \tag{6.3}$$

This finishes the proof of the lemma. □

Additionally, notice that $A_{i,1} \subseteq A_{i,j,1}$; thus from i) we have

$$|A_{i,j,1}| \geq |A_{i,1}| \geq (n/2) - O\Big(|P_i| \cdot \sqrt{n} \log n\Big). \tag{6.4}$$

The following claim is an immediate consequence of this fact and Claim 6.1.13.

Claim 6.1.14. *For each $i \in I$ and $j \in J_i$, we have*

$$\Big||A_{i,j,1}| - |A_{i,j,0}|\Big| \leq O\left(\sqrt{n} \log n \cdot \min\Big\{|P_{i,j}|^2, |P_i|\Big\}\right)$$

Proof. We have from i) and ii) that

$$|A_{i,j,1}| - |A_{i,j,0}| \leq (n/2) + \sqrt{n} - \Big((n/2) - O\Big(|P_{i,j}| \cdot \sqrt{n} \log n\Big)\Big) = O\Big(|P_{i,j}| \cdot \sqrt{n} \log n\Big).$$

On the other hand, from ii), (6.3) and (6.4), we have

$$|A_{i,j,0}| - |A_{i,j,1}| \leq O\Big(\sqrt{n} \log n \cdot \min\Big\{|P_{i,j}|^2, |P_i|\Big\}\Big).$$

Note that $|P_{i,j}| \leq |P_i|$. The lemma then follows. □

We are now ready to prove Lemma 6.1.12.

Proof of Lemma 6.1.12. Let ℓ be a good leaf and let $\phi : Q \to \mathfrak{P}$ be the map associated with ℓ.

Let $|\mathcal{E}|$ denote the support size of $\mathcal{E}$. We may rewrite the two probabilities as follows:

$$\Pr_{(\boldsymbol{T},\boldsymbol{C})\sim\mathcal{E},\boldsymbol{H}\sim\mathcal{E}_{\text{yes}}}\Big[(\boldsymbol{T},\boldsymbol{C},\boldsymbol{H}) \text{ reaches } \ell\Big] = \frac{1}{|\mathcal{E}|}\sum_{(T,C)}\Pr_{\boldsymbol{H}\sim\mathcal{E}_{\text{yes}}}\Big[(T,C,\boldsymbol{H}) \text{ reaches } \ell\Big]$$
$$\Pr_{(\boldsymbol{T},\boldsymbol{C})\sim\mathcal{E},\boldsymbol{H}\sim\mathcal{E}_{\text{no}}}\Big[(\boldsymbol{T},\boldsymbol{C},\boldsymbol{H}) \text{ reaches } \ell\Big] = \frac{1}{|\mathcal{E}|}\sum_{(T,C)}\Pr_{\boldsymbol{H}\sim\mathcal{E}_{\text{no}}}\Big[(T,C,\boldsymbol{H}) \text{ reaches } \ell\Big],$$

where the sum is over the support of $\mathcal{E}$. Hence, it suffices to show that for each (T, C) such that

$$\Pr_{\boldsymbol{H}\sim\mathcal{E}_{\text{yes}}}\Big[(T,C,\boldsymbol{H}) \text{ reaches } \ell\Big] > 0, \tag{6.5}$$

we have the following inequality:

$$\frac{\mathbf{Pr}_{\boldsymbol{H}\sim\mathcal{E}_{\text{no}}}\left[(T,C,\boldsymbol{H}) \text{ reaches } \ell\right]}{\mathbf{Pr}_{\boldsymbol{H}\sim\mathcal{E}_{\text{yes}}}\left[(T,C,\boldsymbol{H}) \text{ reaches } \ell\right]} \geq 1 - o(1). \tag{6.6}$$

Fix a pair (T, C) such that (6.5) holds. Recall that (T, C, H) reaches ℓ if and only if the signature of each $x \in Q$ with respect to (T, C, H) matches exactly $\phi(x) = (\sigma_x, \tau_x, a_x, b_x)$. Given (6.5), the term and clause signatures of x are already known to match σ_x and τ_x (otherwise the LHS of (6.5) is 0). The rest, i.e., a_x and b_x for each $x \in Q$, depends on $H = (h_{i,j})$ only.

Since ℓ is consistent, there is a $\rho_{i,j} \in \{0, 1\}$ for each $P_{i,j}$ such that every $x \in P_{i,j}$ should satisfy $h_{i,j}(x) = \rho_{i,j}$. These are indeed the only conditions for H to match a_x and b_x for each $x \in Q$, and as a result, below we give the conditions on $H = (h_{i,j})$ for the triple (T, C, H) to reach ℓ:

- For $\mathcal{E}_{\text{yes}}$, (T, C, H) reaches ℓ, where $H = (h_{i,j})$ and $h_{i,j}(x) = x_{k_{i,j}}$, if and only if $k_{i,j} \in A_{i,j,\rho_{i,j}}$ for each $i \in I$ and $j \in J_i$ (so that each $x \in P_{i,j}$ has $h_{i,j}(x) = \rho_{i,j}$).
- For $\mathcal{E}_{\text{no}}$, (T, C, H) reaches ℓ, where $H = (h_{i,j})$ and $h_{i,j}(x) = \overline{x_{k_{i,j}}}$, if and only if $k_{i,j} \in A_{i,j,1-\rho_{i,j}}$ for each $i \in I$ and $j \in J_i$ (so that each $x \in P_{i,j}$ has $h_{i,j}(x) = \rho_{i,j}$).

With this characterization, we can rewrite the LHS of (6.6) as follows:

$$\frac{\mathbf{Pr}_{\boldsymbol{H}\sim\mathcal{E}_{\text{no}}}\left[(T,C,\boldsymbol{H}) \text{ reaches } \ell\right]}{\mathbf{Pr}_{\boldsymbol{H}\sim\mathcal{E}_{\text{yes}}}\left[(T,C,\boldsymbol{H}) \text{ reaches } \ell\right]} = \prod_{i\in I, j\in J_i}\left(\frac{|A_{i,j,1-\rho_{i,j}}|}{|A_{i,j,\rho_{i,j}}|}\right) = \prod_{i\in I, j\in J_i}\left(1 + \frac{|A_{i,j,1-\rho_{i,j}}| - |A_{i,j,\rho_{i,j}}|}{|A_{i,j,\rho_{i,j}}|}\right).$$

Thus, applying Claim 6.1.14 and noting that $|A_{i,j,\rho_{i,j}}| \leq n$ (whether $\rho_{i,j} = 0$ or 1),

$$\frac{\mathbf{Pr}_{\boldsymbol{H}\sim\mathcal{E}_{\text{no}}}\left[(T, C, \boldsymbol{H}) \text{ reaches } \ell\right]}{\mathbf{Pr}_{\boldsymbol{H}\sim\mathcal{E}_{\text{yes}}}\left[(T, C, \boldsymbol{H}) \text{ reaches } \ell\right]} \geq \prod_{i\in I, j\in J_i}\left(1 - O\left(\frac{\log n \cdot \min\{|P_{i,j}|^2, |P_i|\}}{\sqrt{n}}\right)\right)$$
$$\geq 1 - O\left(\frac{\log n}{\sqrt{n}}\right)\sum_{i\in I, j\in J_i}\min\left\{|P_{i,j}|^2, |P_i|\right\}.$$

As $\sum_j |P_{i,j}| \leq 2|P_i|$, $\sum_{j\in J_i}\min\left\{|P_{i,j}|^2, |P_i|\right\}$ is maximized if $|J_i| = 2\sqrt{|P_i|}$ and $|P_{i,j}| = \sqrt{|P_i|}$. So

$$\sum_{i\in I, j\in J_i}\min\left\{|P_{i,j}|^2, |P_i|\right\} \leq \sum_{i\in I} 2|P_i|^{3/2} \leq O(q^{3/2}),$$

since $\sum_i |P_i| \leq 2q$. This finishes the proof of the lemma since q is chosen to be $n^{1/3}/\log^2 n$.

□

6.1.6 Proof of the pruning lemma

Let E be the set of bad edges as defined in Definition 46 (recall that if (u, v) is a bad edge, then the root-to-u path cannot have any bad edge). We split the proof of Lemma 6.1.11 into four lemmas, one lemma for each type of bad edges. To this end, we define four sets E_1, E_2, E_3 and E_4 (we follow the same notation of Definition 46): An edge $(u, v) \in E$ belongs to

1. E_1 if $|A_{i,1} \setminus A'_{i,1}| \geq \alpha\sqrt{n}\log n$ for some $i \in I$;
2. E_2 if $|A_{i,j,0} \setminus A'_{i,j,0}| \geq \alpha\sqrt{n}\log n$ for some $i \in I$ and $j \in J_i$;
3. E_3 if it is not in E_2 and for some $i \in I$ and $j \in J_i$, $P_{i,j}$ is 0-consistent but $P'_{i,j}$ is inconsistent (when $(u, v) \in E_3$ and the above occurs, we say (u, v) is E_3-bad at (i, j));
4. E_4 if it is not in E_1 or E_2 and for some $i \in I$ and $j \in J_i$, $P_{i,j}$ is 1-consistent but $P'_{i,j}$ is inconsistent (when $(u, v) \in E_4$ and the above occurs, we say (u, v) is E_4-bad at (i, j)).

It is clear that $E = E_1 \cup E_2 \cup E_3 \cup E_4$. (These four sets are not necessarily pairwise disjoint though we did exclude edges of E_2 from E_3 and edges of E_1 and E_2 from E_4 explicitly.)

Each lemma below states that the probability of $(\boldsymbol{T}, \boldsymbol{C}) \sim \mathcal{E}$ and $\boldsymbol{H} \sim \mathcal{E}_{\text{yes}}$ taking an edge in E_i is $o(1)$. Lemma 6.1.11 then follows directly from a union bound over the four sets.

Lemma 6.1.15. *The probability of $(\boldsymbol{T}, \boldsymbol{C}) \sim \mathcal{E}$ and $\boldsymbol{H} \sim \mathcal{E}_{yes}$ taking an edge in E_1 is $o(1)$.*

Proof. Let u be an internal node. We prove that, when $(\boldsymbol{T}, \boldsymbol{C}) \sim \mathcal{E}$ and $\boldsymbol{H} \sim \mathcal{E}_{\text{yes}}$, either $(\boldsymbol{T}, \boldsymbol{C}, \boldsymbol{H})$ reaches node u with probability 0 or

$$\Pr_{(\boldsymbol{T},\boldsymbol{C})\sim\mathcal{E},\boldsymbol{H}\sim\mathcal{E}_{\text{yes}}}\left[(\boldsymbol{T}, \boldsymbol{C}, \boldsymbol{H}) \text{ takes an } E_1\text{-edge at } u \,\middle|\, (\boldsymbol{T}, \boldsymbol{C}, \boldsymbol{H}) \text{ reaches } u\right] = o(1/q). \tag{6.7}$$

Lemma 6.1.15 follows from Lemma 6.0.3. Below we assume that the probability of $(\boldsymbol{T}, \boldsymbol{C}, \boldsymbol{H})$ reaching node u is positive. Let $\phi : Q \to \mathfrak{P}$ be the map associated with u, and let $x \in \{0,1\}^n$ be the string queried at u. Whenever we discuss a child node v of u below, we use $\phi' : Q \cup \{x\} \to \mathfrak{P}$ to denote the map associated with v and $(I; J; P; R; A; \rho)$ and $(I'; J'; P'; R'; A'; \rho')$ to denote the tuples ϕ and ϕ' induce. (Note that v is not a specific node but can be any child of u.)

Fix an $i \in I$. We upperbound by $o(1/q^2)$ the conditional probability of $(\boldsymbol{T}, \boldsymbol{C}, \boldsymbol{H})$ following an edge (u, v) with $|A_{i,1} \setminus A'_{i,1}| \geq \alpha\sqrt{n}\log n$. (6.7) follows directly from a union bound over $i \in I$.

With i fixed, observe that any edge (u, v) has either $A'_{i,1} = A_{i,1}$ or $A'_{i,1} = A_{i,1} \setminus \Delta_i$ with

$$\Delta_i = \left\{\ell \in A_{i,1} : x_\ell = 0\right\} \subseteq A_{i,1}.$$

The latter occurs if and only if $P'_i = P_i \cup \{x\}$. Therefore, we assume WLOG that $|\Delta_i| \geq \alpha\sqrt{n}\log n$ (otherwise the conditional probability is 0 for i), and now it suffices to upperbound by $o(1/q^2)$ the conditional probability of $(\boldsymbol{T}, \boldsymbol{C}, \boldsymbol{H})$ taking an edge (u, v) with $P'_i = P_i \cup \{x\}$.

To analyze this conditional probability for $i \in I$, we fix a triple (T_{-i}, C, H), where we use T_{-i} to denote a sequence of $N - 1$ terms with only the ith term missing, such that

$$\Pr_{\boldsymbol{T}_i}\left[((T_{-i}, \boldsymbol{T}_i), C, H) \text{ reaches } u\right] > 0,$$

where $\boldsymbol{T}_i$ is a term drawn uniformly at random. It suffices to prove for any such (T_{-i}, C, H):

$$\begin{aligned}\Pr_{\boldsymbol{T}_i}\Big[((T_{-i}, \boldsymbol{T}_i), C, H) \text{ reaches } u \text{ and } P_i' = P_i \cup \{x\}\Big] \\ \leq o(1/q^2) \cdot \Pr_{\boldsymbol{T}_i}\Big[((T_{-i}, \boldsymbol{T}_i), C, H) \text{ reaches } u\Big].\end{aligned} \tag{6.8}$$

Recalling Fact 6.1.9, the latter event, $((T_{-i}, \boldsymbol{T}_i), C, H)$ reaching u, imposes two conditions on $\boldsymbol{T}_i$:

1. For each $y \in P_i$, $\boldsymbol{T}_i(y) = 1$, and
2. For each $z \in R_i$, $\boldsymbol{T}_i(z) = 0$.

Let U denote the set of all such terms $T : \sqrt{n} \to [n]$. Then equivalently $T \in U$ if and only if

$$U\text{: } T(k) \in A_{i,1} \text{ for all } k \in [\sqrt{n}] \text{ and each } z \in R_i \text{ has } z_{T(k)} = 0 \text{ for some } k \in [\sqrt{n}].$$

Regarding the former event in (6.8), i.e. $((T_{-i}, \boldsymbol{T}_i), C, H)$ reaching u and $P_i' = P_i \cup \{x\}$, a necessary condition over $\boldsymbol{T}_i$ is the same as above but in addition we require $\boldsymbol{T}_i(x) = 1$. (Note that this is not a sufficient condition since for that we also need $\boldsymbol{T}_i$ to be one of the first two terms that x satisfies, which depends on T_{-i}.) Let V denote the set of all such terms. Then $T \in V$ if

$$V\text{: } T(k) \in A_{i,1} \setminus \Delta_i \text{ for all } k \in [\sqrt{n}] \text{ and each } z \in R_i \text{ has } z_{T(k)} = 0 \text{ for some } k \in [\sqrt{n}].$$

In the rest of the proof we prove that $|V|/|U| = o(1/q^2)$, from which (6.8) follows. Let $\ell = \log n$. First we write U' to denote the following subset of U: $T' \in U$ is in U' if

$$\Big|\{k \in [\sqrt{n}] : T'(k) \in \Delta_i\}\Big| = \ell,$$

and it suffices to show $|V|/|U'| = o(1/q^2)$. Next we define the following bipartite graph G between U' and V (inspired by similar arguments of [34]): $T' \in U'$ and $T \in V$ have an edge if and only if $T'(k) = T(k)$ for all $k \in [\sqrt{n}]$ with $T'(k) \notin \Delta_i$. Each $T' \in U'$ has degree at most $|A_{i,1} \setminus \Delta_i|^\ell$, as one can only move each $T'(k) \in \Delta_i$ to $A_{i,1} \setminus \Delta_i$.

To lowerbound the degree of a $T \in V$, note that one only needs at most q many variables of T to kill all strings in R_i. Let $H \subset [\sqrt{n}]$ be any set of size at most q such that for each string $z \in R_i$, there exists a $k \in H$ with $z_{T(k)} = 0$.[2] Then one can choose any ℓ distinct indices $k_1, \ldots, k_\ell$ from $\overline{H}$, as well as any ℓ (not necessarily distinct) variables $t_1, \ldots, t_\ell$ from Δ_i, and let T' be a term where

$$T'(k) = \begin{cases} t_i & k = k_i \text{ for some } i \in [\ell] \\ T(k) & \text{otherwise.} \end{cases}$$

The resulting T' is in U' and (T, T') is an edge in G. As a result, the degree of $T \in V$ is at least

$$\binom{\sqrt{n} - q}{\ell} \cdot |\Delta_i|^\ell.$$

By counting the number of edges in G in two different ways and using $|A_{i,1}| \leq (n/2) + \sqrt{n}$,

$$\frac{|U'|}{|V|} \geq \binom{\sqrt{n} - q}{\ell} \cdot \left(\frac{|\Delta_i|}{|A_{i,1} \setminus \Delta_i|} \right)^\ell \geq \left(\frac{\sqrt{n}}{2\ell} \cdot \frac{\alpha \sqrt{n} \ell}{(n/2) + \sqrt{n}} \right)^\ell > \omega(q^2),$$

by choosing a large enough constant $\alpha > 0$. This finishes the proof of the lemma. □

Lemma 6.1.16. *The probability of $(\boldsymbol{T}, \boldsymbol{C}) \sim \mathcal{E}$ and $\boldsymbol{H} \sim \mathcal{E}_{yes}$ taking an edge in E_2 is $o(1)$.*

Proof. The proof of this lemma is similar to that of Lemma 6.1.15. Let u be any internal node of the tree. We prove that, when $(\boldsymbol{T}, \boldsymbol{C}) \sim \mathcal{E}$, $\boldsymbol{H} \sim \mathcal{E}_{\text{yes}}$, either $(\boldsymbol{T}, \boldsymbol{C}, \boldsymbol{H})$ reaches u with probability 0 or

$$\Pr_{(\boldsymbol{T}, \boldsymbol{C}) \sim \mathcal{E}, \boldsymbol{H} \sim \mathcal{E}_{\text{yes}}} \left[(\boldsymbol{T}, \boldsymbol{C}, \boldsymbol{H}) \text{ takes an } E_2\text{-edge at } u \,\middle|\, (\boldsymbol{T}, \boldsymbol{C}, \boldsymbol{H}) \text{ reaches } u \right] = o(1/q). \tag{6.9}$$

Assume below WLOG that the probability of $(\boldsymbol{T}, \boldsymbol{C}, \boldsymbol{H})$ reaching u is positive.

Fix $i \in I$ and $j \in J_i$. We upperbound the conditional probability of $(\boldsymbol{T}, \boldsymbol{C}, \boldsymbol{H})$ taking an edge (u, v) with $|A_{i,j,0} \setminus A'_{i,j,0}| \geq \alpha \sqrt{n} \log n$ by $o(1/q^3)$. (6.9) follows by a union bound.

[2]For example, since $|R_i| \leq q$, one can set H to contain the smallest $k \in [\sqrt{n}]$ such that $z_{T(k)} = 0$, for each $z \in R_i$.

Similarly let

$$\Delta_{i,j} = \Big\{\ell \in A_{i,j,0} : x_\ell = 1\Big\} \subseteq A_{i,j,0}, \tag{6.10}$$

and assume WLOG that $|\Delta_{i,j}| \geq \alpha\sqrt{n}\log n$ (as otherwise the conditional probability is 0 for i, j). Then it suffices to upperbound the conditional probability of $(\boldsymbol{T}, \boldsymbol{C}, \boldsymbol{H})$ going along an edge (u, v) with $P'_{i,j} = P_{i,j} \cup \{x\}$ by $o(1/q^3)$. The rest of the proof is symmetric to that of Lemma 6.1.15. □

Lemma 6.1.17. *The probability of $(\boldsymbol{T}, \boldsymbol{C}) \sim \mathcal{E}$ and $\boldsymbol{H} \sim \mathcal{E}_{yes}$ taking an edge in E_3 is $o(1)$.*

Proof. We fix any pair (T, C) from the support of $\mathcal{E}$ and prove that

$$\Pr_{\boldsymbol{H}\sim\mathcal{E}_{\text{yes}}}\Big[(T, C, \boldsymbol{H}) \text{ takes an } E_3\text{-edge}\Big] = o(1). \tag{6.11}$$

The lemma follows by averaging (6.11) over all pairs (T, C) in the support of $\mathcal{E}$. To prove (6.11) we fix any internal node u such that the probability of $(T, C, \boldsymbol{H})$ reaching u is positive, and prove that

$$\Pr_{\boldsymbol{H}\sim\mathcal{E}_{\text{yes}}}\left[(T, C, \boldsymbol{H}) \text{ takes an } E_3\text{-edge leaving } u \,\middle|\, (T, C, \boldsymbol{H}) \text{ reaches } u\right] = o(1/q). \tag{6.12}$$

(6.11) follows by Lemma 6.0.3. Below we assume the probability of $(T, C, \boldsymbol{H})$ reaching u is positive.

We assume WLOG that there is no edge in E along the root-to-u path; otherwise, (6.12) is 0. We follow the same notation used in the proof of Lemma 6.1.15, i.e., $\phi_u : Q \to \mathfrak{P}$ as the map associated with u, x as the query made at u, and $(I; J; P; R; A; \rho)$ as the tuple induced by ϕ_u. We also write F to denote the set of pairs (i, j) such that $i \in I$ and $j \in J$.

Observe that since (T, C) is fixed, the term and clause signatures of every string are fixed, and in particular the term and clause signatures (denoted σ_x and τ_x) of x are fixed. We assume WLOG that $\sigma_x = e_k$ for some $k \in [N]$ (otherwise x will never be added to any $P_{i,j}$ when $(T, C, \boldsymbol{H})$ leaves u and (6.12) is 0 by the definition of E_3). In this case we write D to denote the set of $\{(k, j) : \tau_{x,j} = 0\}$ with $|D| \leq 2$. As a result, whenever $(T, C, \boldsymbol{H})$ takes an E_3-edge leaving from u, this edge must be E_3-bad at one of the pairs $(k, j) \in D$.

Thus, the LHS of (6.12) is the same as

$$\sum_{(k,j)\in D} \mathop{\mathbf{Pr}}_{\boldsymbol{H}\sim\mathcal{E}_{\text{yes}}} \Big[(T, C, \boldsymbol{H}) \text{ takes a } (u, v) \text{ that is } E_3\text{-bad at } (k, j) \;\Big|\; (T, C, \boldsymbol{H}) \text{ reaches } u\Big]. \tag{6.13}$$

To bound the conditional probability for (k, j) above by $o(1/q)$, we assume WLOG that $(k, j) \in F$ (otherwise x would create a new $P_{k,j}$ whenever $(T, C, \boldsymbol{H})$ takes an edge (u, v) leaving u, and the latter cannot be E_3-bad at (k, j)). Next we define ($A_{k,j,0}$ below is well defined since $(k, j) \in F$)

$$\Delta_{k,j} = \Big\{\ell \in A_{k,j,0} : x_\ell = 1\Big\}.$$

We may assume WLOG that $|\Delta_{k,j}| < \alpha\sqrt{n}\log n$; otherwise $(T, C, \boldsymbol{H})$ can never take an edge (u, v) in E_3 because E_2-edges are explicitly excluded from E_3. Finally, we assume WLOG $\rho_{k,j}(y) = 0$ for all $y \in P_{k,j}$; otherwise the edge (u, v) that $(T, C, \boldsymbol{H})$ takes can never be E_3-bad at (k, j).

With all these assumptions on (k, j) in place, we prove the following inequality:

$$\begin{aligned}\mathop{\mathbf{Pr}}_{\boldsymbol{H}\sim\mathcal{E}_{\text{yes}}} \Big[(T, C, \boldsymbol{H}) \text{ takes a } (u, v) \text{ that is } E_3\text{-bad at } (k, j)\Big] \\ \le \frac{|\Delta_{k,j}|}{|A_{k,j,0}|} \cdot \mathop{\mathbf{Pr}}_{\boldsymbol{H}\sim\mathcal{E}_{\text{yes}}} \Big[(T, C, \boldsymbol{H}) \text{ reaches } u\Big].\end{aligned} \tag{6.14}$$

Given $|\Delta_{k,j}| = O(\sqrt{n}\log n)$ and $|A_{i,j,0}| \ge (n/2) - O(q\sqrt{n}\log n) = \Omega(n)$ (since there is no bad edge particularly no E_2-edge, from the root to u), (6.12) follows by summing over D, with $|D| \le 2$.

We work on (6.14) in the rest of the proof. Fix any tuple $H_{-(k,j)}$ (with its (k, j)th entry missing) such that the probability of $(T, C, (H_{-(k,j)}, \boldsymbol{h}))$ reaching u is positive, where $\boldsymbol{h}$ is a random dictator function with its dictator variable drawn from $[n]$ uniformly. Then (6.14) follows from

$$\begin{aligned}\mathop{\mathbf{Pr}}_{\boldsymbol{h}} \Big[(T, C, (H_{-(k,j)}, \boldsymbol{h})) \text{ takes } (u, v) \text{ that is } E_3\text{-bad at } (k, j)\Big] \\ \le \frac{|\Delta_{k,j}|}{|A_{k,j,0}|} \cdot \mathop{\mathbf{Pr}}_{\boldsymbol{h}} \Big[(T, C, (H_{-(k,j)}, \boldsymbol{h})) \text{ reaches } u\Big].\end{aligned} \tag{6.15}$$

The event on the RHS, i.e., that $(T, C, (H_{-(k,j)}, \boldsymbol{h}))$ reaches u, imposes the following

condition on r the dictator variable of $\boldsymbol{h}$: $r \in A_{k,j,0}$, since $\rho_{k,j}(y) = 0$ for all $y \in P_{k,j}$. Hence the probability on the RHS of (6.15) is $|A_{i,j,0}|/n$. On the other hand, the event on the LHS of (6.15), that $(T, C, (H_{-(i,j)}, \boldsymbol{h}))$ follows a (u, v) that is E_3-bad at (k, j), imposes the following necessary condition for r: $r \in \Delta_{k,j}$.[3] As a result, the probability on the LHS of (6.15) is at most $|\Delta_{k,j}|/n$. (6.15) then follows. □

Lemma 6.1.18. *The probability of $(\boldsymbol{T}, \boldsymbol{C}) \sim \mathcal{E}$ and $\boldsymbol{H} \sim \mathcal{E}_{yes}$ taking an edge in E_4 is $o(1)$.*

Proof. We fix a pair (T, C) from the support of $\mathcal{E}$ and prove that

$$\Pr_{\boldsymbol{H} \sim \mathcal{E}_{\text{yes}}} \Big[(T, C, \boldsymbol{H}) \text{ takes an } E_4\text{-edge} \Big] = o(1). \tag{6.16}$$

The lemma follows by averaging (6.16) over all (T, C) in the support of $\mathcal{E}$. To prove (6.16), fix a leaf ℓ such that the probability of $(T, C, \boldsymbol{H})$ reaching ℓ is positive. Let $u_1 \cdots u_{t'} u_{t'+1} = \ell$ be the root-to-ℓ path and let $q(u_s)$ denote the following conditional probability:

$$\Pr_{\boldsymbol{H} \sim \mathcal{E}_{\text{yes}}} \Big[(T, C, \boldsymbol{H}) \text{ takes an } E_4\text{-edge leaving } u_s \,\Big|\, (T, C, \boldsymbol{H}) \text{ reaches } u_s \Big].$$

It then suffices to show for every such leaf ℓ,

$$\sum_{s \in [t']} q(u_s) = o(1), \tag{6.17}$$

since (6.16) would then follow by Lemma 6.0.4. To prove (6.17), we use t to denote the smallest integer such that (u_{t+1}, u_{t+2}) is an edge in E_1 or E_2 with $t = t'$ by default if there is no such edge along the path. By the choice of t, there is no edge in E_1 or E_2 along $u_1 \cdots u_{t+1}$. For (6.17) it suffices to show

$$\sum_{s \in [t]} q(u_s) = o(1). \tag{6.18}$$

To see this we consider two cases. If there is no E_1, E_2 edge along the root-to-ℓ path, then the two sums in (6.17) and (6.18) are the same. If (u_{t+1}, u_{t+2}) is an edge in E_1 or E_2,

[3] Note that this is *not* a sufficient condition, because the other pair $(k, j') \in D$ may have $|\Delta_{k,j'}| \geq \alpha\sqrt{n} \log n$.

then $q(u_s) = 0$ if $s \geq t+2$ (since $(u,v) \notin E$ if there is already an edge in E along the path to u). We claim that $q(u_{t+1})$ must be 0 as well. This is because, given that (T, C) is fixed and that $(T, C, \boldsymbol{H})$ takes (u_{t+1}, u_{t+2}) with a positive probability, whenever $(T, C, \boldsymbol{H})$ follows an edge (u_{t+1}, v) from u_{t+1}, v has the same term and clause signatures (σ_x, τ_x) as u_{t+2} and thus, also has the same P and A (as part of the tuple its map induces). As a result (u_{t+1}, v) is also in E_1 or E_2 and cannot be an edge in E_4 (recall that we explicitly excluded E_1 and E_2 from E_4). Below we focus on u_s with $s \in [t]$ and upperbound $q(u_s)$.

For each $s \in [t]$ we write x^s to denote the string queried at u_s and let $(I^s; J^s; P^s; Q^s; R^s; \rho^s)$ be the tuple induced by the map associated with u_s. We also write F_s to denote the set of pairs (i,j) with $i \in I^s, j \in J^s_i$. Following the same arguments used to derive (6.13) in the proof of Lemma 6.1.17, let $D_s \subseteq F_s$ denote the set of at most two pairs (i,j) such that x^s is added to $P^s_{i,j}$ when $(T, C, \boldsymbol{H})$ reaches u_s. Note that if x^s just creates a new $P_{i,j}$ (so $(i,j) \notin F_s$), we do not include it in D_s. As a result, whenever $(T, C, \boldsymbol{H})$ takes an E_4-edge (u,v), the latter must be E_4-bad at one of $(i,j) \in D_s$.

Next for each pair $(i,j) \in D_s$, we can follow the analysis of (6.14) to show that

$$\Pr_{\boldsymbol{H} \sim \mathcal{E}_{\text{yes}}}\Big[(T, C, \boldsymbol{H}) \text{ takes a } (u,v) \text{ that is } E_4\text{-bad at } (i,j)\Big] \leq \frac{\left|\Delta^s_{i,j}\right|}{\left|A^s_{i,j,1}\right|} \cdot \Pr_{\boldsymbol{H} \sim \mathcal{E}_{\text{yes}}}\Big[(T, C, \boldsymbol{H}) \text{ reaches } u\Big],$$

where the set $\Delta^s_{i,j}$ is defined as

$$\Delta^s_{i,j} = \Big\{k \in A^s_{i,j,1} : x^s_k = 0\Big\}.$$

As there is no E_1 or E_2 edge along the path to u_s, we have by (6.4) that $A^s_{i,j,1}$ has size $\Omega(n)$. Thus,

$$q(u_s) \leq O(1/n) \cdot \sum_{(i,j) \in D_s} \left|\Delta^s_{i,j}\right| \quad \text{and} \quad \sum_{s \in [t]} q(u_s) \leq O(1/n) \cdot \sum_{s \in [t]} \sum_{(i,j) \in D_s} \left|\Delta^s_{i,j}\right|. \tag{6.19}$$

Let $(I^*; J^*; P^*; R^*; A^*; \rho^*)$ be the tuple induced by the map associated with u_{t+1} and let F^* be the set of (i,j) with $i \in I^*$ and $j \in J^*_i$. We upperbound the second sum in (6.19)

above by focusing on any fixed pair $(i,j) \in F^*$ and observing that

$$\sum_{s:(i,j)\in D_s} \left|\Delta^s_{i,j}\right| + \left|A^*_{i,j,1}\right| \leq (n/2) + \sqrt{n}.$$

This is because $\Delta^s_{i,j}$ and $A^*_{i,j,1}$ are pairwise disjoint and their union is indeed exactly the number of 1-entries of the query string along the path that first creates $P_{i,j}$. The latter is at most $(n/2) + \sqrt{n}$ because we assumed that strings queried in the tree lie in the middle layers. On the other hand,

$$\left|A^*_{i,j,1}\right| \geq (n/2) - O\Big(\sqrt{n}\log n \cdot \min\Big\{|P^*_{i,j}|^2, |P^*_i|\Big\}\Big).$$

This follows directly from (6.3) and (6.4) and our choice of t at the beginning of the proof so that there is no E_1 or E_2 edge from u_1 to u_{t+1}. We finish the proof by plugging the two inequalities into (6.19) and follow the same arguments used at the end of the proof of the lemma for good leaves. □

6.2 Unateness Lower Bound

We start with some notation for strings. Given $A \subseteq [n]$ and $x \in \{0,1\}^n$, we use x_A to denote the string in $\{0,1\}^A$ that agrees with x over A. Given $y \in \{0,1\}^A$ and $z \in \{0,1\}^{\overline{A}}$, we use $x = y \circ z$ (as their concatenation) to denote the string $x \in \{0,1\}^n$ that agrees with y over A and z over $\overline{A}$. Given $x \in \{0,1\}^n$ and $y \in \{0,1\}^A$ with $A \subseteq [n]$, we use $x \oplus y$ to denote the n-bit string x' with $x'_i = x_i$ for all $i \notin A$ and $x'_i = x_i \oplus y_i$ for all $i \in A$, i.e., x' is obtained from x by an XOR with y over A.

6.2.1 Distributions

For a fixed $n > 0$ we describe a pair of distributions, $\mathcal{D}_{\text{yes}}$ and $\mathcal{D}_{\text{no}}$, supported on Boolean functions $f\colon \{0,1\}^n \to \{0,1\}$ that will be used to obtain a two-sided and adaptive lower bound for unateness testing. After defining the distributions, we show in this subsection that any $\boldsymbol{f} \sim \mathcal{D}_{\text{yes}}$ is unate, and $\boldsymbol{f} \sim \mathcal{D}_{\text{no}}$ is $\Omega(1)$-far from being unate with probability $\Omega(1)$.

Let N be the following parameter:

$$N = \left(1 + \frac{1}{\sqrt{n}}\right)^{n/4} \approx e^{\sqrt{n}/4}.$$

A function $\boldsymbol{f} \sim \mathcal{D}_{\text{yes}}$ is drawn using the following procedure:

1. Sample a subset $\mathbf{M} \subset [n]$ uniformly at random from all subsets of size $n/2$.
2. Sample $\mathbf{T} \sim \mathcal{E}(\mathbf{M})$ (which we describe next). $\mathbf{T}$ is a sequence of terms $(\mathbf{T}_i : i \in [N])$. $\mathbf{T}$ is then used to define a multiplexer map $\boldsymbol{\Gamma} = \boldsymbol{\Gamma}_{\mathbf{T}} \colon \{0,1\}^n \to [N] \cup \{0^*, 1^*\}$.
3. Sample $\mathbf{H} \sim \mathcal{E}_{\text{yes}}(\mathbf{M})$ where $\mathbf{H} = (\boldsymbol{h}_i : i \in [N])$. For each $i \in [N]$, $\boldsymbol{h}_i \colon \{0,1\}^n \to \{0,1\}$ is a dictatorship function $\boldsymbol{h}_i(x) = x_k$ with k sampled independently and uniformly from $\overline{\mathbf{M}}$. We will refer to $\boldsymbol{h}_i$ as the dictatorship function and x_k (or simply its index k) as the *special* variable associated with the ith term $\mathbf{T}_i$.
4. Sample two strings $\boldsymbol{r} \in \{0,1\}^{\mathbf{M}}$ and $\boldsymbol{s} \in \{0,1\}^{\mathbf{M}}$ uniformly at random. Finally, the function $\boldsymbol{f} = \boldsymbol{f}_{\mathbf{M},\mathbf{T},\mathbf{H},\boldsymbol{r},\boldsymbol{s}} \colon \{0,1\}^n \to \{0,1\}$ is defined as follows:

$$\boldsymbol{f}_{\mathbf{M},\mathbf{T},\mathbf{H},\boldsymbol{r},\boldsymbol{s}}(x) = f_{\mathbf{M},\mathbf{T},\mathbf{H}}\Big(x \oplus (\boldsymbol{r} \circ \boldsymbol{s})\Big),$$

where $\boldsymbol{f}_{\mathbf{M},\mathbf{T},\mathbf{H}}$ is defined as follows (with the truncation done first):

$$\boldsymbol{f}_{\mathbf{M},\mathbf{T},\mathbf{H}}(x) = \begin{cases} 0 & \text{if } |x_{\mathbf{M}}| < (n/4) - \sqrt{n} \\ 1 & \text{if } |x_{\mathbf{M}}| > (n/4) + \sqrt{n} \\ 0 & \text{if } \boldsymbol{\Gamma}(x) = 0^* \\ 1 & \text{if } \boldsymbol{\Gamma}(x) = 1^* \\ h_{\boldsymbol{\Gamma}(x)}(x) & \text{otherwise (i.e., when } \boldsymbol{\Gamma}(x) \in [N]) \end{cases}$$

This finishes the definition of our yes-distribution $\mathcal{D}_{\text{yes}}$.

A function $\boldsymbol{f} = \boldsymbol{f}_{\mathbf{M},\mathbf{T},\mathbf{H},\boldsymbol{r},\boldsymbol{s}} \sim \mathcal{D}_{\text{no}}$ is drawn using a similar procedure, with the only difference being that $\mathbf{H} = (\boldsymbol{h}_i : i \in [N])$ is sampled from $\mathcal{E}_{\text{no}}(\mathbf{M})$ instead of $\mathcal{E}_{\text{yes}}(\mathbf{M})$:

each $\boldsymbol{h}_i$ is a dictatorship function $\boldsymbol{h}_i(x) = x_k$ with probability $1/2$ and an anti-dictatorship $\boldsymbol{h}_i(x) = \overline{x_k}$ with probability $1/2$, where k is chosen independently and uniformly at random from $\overline{\mathbf{M}}$. We will also refer to $\boldsymbol{h}_i$ as the dictatorship or anti-dictatorship function and x_k as the special variable associated with $\mathbf{T}_i$.

Remark 63. *Note that the truncation in $\boldsymbol{f}_{\mathbf{M},\mathbf{T},\mathbf{H},\boldsymbol{r},\boldsymbol{s}}$ is done after sampling $\boldsymbol{r}$. As a result, we may* not *assume all queries are made in the middle layers, like we did in Section 6.1.*

Fixing an $M \subset [n]$ of size $n/2$, we now describe $\mathbf{T} \sim \mathcal{E}(M)$ to finish the description of the two distributions. Each term $\mathbf{T}_i$ in $\mathbf{T}$, $i \in [N]$, is drawn independently and is a random *subset* of M with each $j \in M$ included with probability $1/\sqrt{n}$ independently. We also abuse the notation and interpret each term $\mathbf{T}_i$ as a Boolean function that is the conjunction of its variables:

$$\mathbf{T}_i(x) = \bigwedge_{j \in \mathbf{T}_i} x_j.$$

Note that, for some technical reason that will become clear later in the proof of Lemma 6.2.13, the definition of terms here is slightly different from that used in the monotonicity lower bound, though both are the conjunction of roughly $\sqrt{n}/2$ ($\sqrt{n}$ in monotonicity) variables. Given $\mathbf{T}$, the multiplexer map $\mathbf{\Gamma}_{\mathbf{T}} \colon \{0,1\}^n \to [N] \cup \{0^*, 1^*\}$ indicates the index of the term $\mathbf{T}_i$ that is satisfied by x, if there is a unique one; it returns 0^* if no term is satisfied, or 1^* if more than one term are satisfied:

$$\mathbf{\Gamma}_{\mathbf{T}}(x) = \begin{cases} 0^* & \forall\, i \in [N],\ \mathbf{T}_i(x) = 0 \\ 1^* & \exists\, i \neq j \in [N],\ \mathbf{T}_i(x) = \mathbf{T}_j(x) = 1 \\ i & \mathbf{T}_i(x) = 1 \text{ for a unique } i \in [N] \end{cases}$$

We give some intuition for the reason why the two distributions are hard to distinguish and can be used to obtain a much better lower bound for unateness testing, despite of being much simpler than the two-level construction used in the previous section. Note that $\mathcal{D}_{\text{yes}}$ and $\mathcal{D}_{\text{no}}$ are exactly the same except that (1) in $\mathcal{D}_{\text{yes}}$, $\boldsymbol{h}_i$'s are random dictatorship or anti-dictatorship functions (if one takes $\boldsymbol{s}$ into consideration) but are *consistent* in the sense that all $\boldsymbol{h}_i$'s with the same special variable x_k are either all dictatorship or anti-dictatorship

functions; (2) in contrast, whether $\boldsymbol{h}_i$ is a dictatorship or anti-dictatorship is independent for each $i \in [N]$ in $\mathcal{D}_{\text{no}}$. Informally, the only way for an algorithm to be sure that f is from $\mathcal{D}_{\text{no}}$ (instead of $\mathcal{D}_{\text{yes}}$) is to find two terms with the same special variable x_k but one with a dictatorship and the other with an anti-dictatorship function over x_k. As a result, one can interpret our $\tilde{\Omega}(n^{2/3})$ lower bound (at a high level) as the product of two quantities: the number of queries one needs to *breach* a term $\mathbf{T}_i$ (see Section 6.2.3 for details) and find its special variable, and the number of terms one needs to breach in order to find two with the same special variable. This is different from monotonicity testing since we are done once a term is breached there, and enables us to obtain a much better lower bound for unateness testing.

Next we prove that $\boldsymbol{f} \sim \mathcal{D}_{\text{yes}}$ is unate and $\boldsymbol{f} \sim \mathcal{D}_{\text{no}}$ is far from unate with high probability.

Lemma 6.2.1. *Every f in the support of $\mathcal{D}_{yes}$ is unate.*

Proof. Given the definition of $f = f_{M,T,H,r,s}$ using $f_{M,T,H}$, it suffices to show that $f_{M,T,H}$ is monotone. The rest of the proof is similar to that of Lemma 6.1.1. □

Lemma 6.2.2. *A function $\boldsymbol{f} \sim \mathcal{D}_{no}$ is $\Omega(1)$-far from unate with probability $\Omega(1)$.*

Proof. Consider a fixed subset $M \subset [n]$ of size $n/2$. It suffices to prove that, when $\mathbf{T} \sim \mathcal{E}(M)$ and $\mathbf{H} \sim \mathcal{E}_{\text{no}}(M)$, the function $\boldsymbol{f} = \boldsymbol{f}_{M,\mathbf{T},\mathbf{H}}$ is $\Omega(1)$-far from unate. This is due to the fact that flipping variables of a function as we do using $\boldsymbol{r}$ and $\boldsymbol{s}$ does not change its distance to unateness.

Fix T in the support of $\mathcal{E}(M)$ and H in the support of $\mathcal{E}_{\text{no}}(M)$. We let $X \subset \{0,1\}^n$ denote the set of $x \in \{0,1\}^n$ in the middle layers (i.e. $|x_M|$ is within $n/4 \pm \sqrt{n}$) such that $\Gamma_T(x) = i$ for some $i \in [N]$ (rather than 0^* or 1^*). For each $x \in X$ with $\Gamma_T(x) = i$, we also let $\rho(x) = k$ be the special variable associated with T_i (i.e., $h_i(x) = x_k$ or $h_i(x) = \overline{x_k}$). As $\rho(x) \in \overline{M}$ and $\Gamma_T(x)$ depends only on variables in M, we have that

$$\Gamma_T\Big(x^{(\rho(x))}\Big) = \Gamma_T(x),$$

i.e., after flipping the $\rho(x)$th bit of x, the new string still satisfies uniquely the same term as x.

Let $x^* = x^{(\rho(x))}$ for each string $x \in X$ (then $(x^*)^* = x$). The claim below shows that (x, x^*) is a bi-chromatic edge along the $\rho(x)$th direction. As a result, one can decompose $|X|$ into $|X|/2$ many *disjoint* bi-chromatic edges (x, x^*).

Claim 6.2.3. *For all $x \in X$, (x, x^*) is a bi-chromatic edge of $f_{M,T,H}$.*

Proof. Let $k = \rho(x) \in \overline{M}$. Then $f_{M,T,H}(x)$ and $f_{M,T,H}(x^*)$ are either x_k and x_k^* or $\overline{x_k}$ and $\overline{x_k^*}$. The claim follows directly from $x^* = x^{(k)}$ and thus, $x_k^* = \overline{x_k}$. □

For each $k \in \overline{M}$, we partition strings $x \in X$ with $\rho(x) = k$ and $f(x) = 0$ into

$$X_k^+ = \Big\{x \in X : \rho(x) = k,\, x_k = 0,\, f(x) = 0\Big\} \quad \text{and} \quad X_k^- = \Big\{x \in X : \rho(x) = k,\, x_k = 1,\, f(x) = 0\Big\}.$$

Note that for each $x \in X_k^+$, (x, x^*) is a monotone bi-chromatic edge; for each $x \in X_k^-$, (x, x^*) is an anti-monotone bi-chromatic edge. Since all these $|X|/2$ edges are disjoint, by Lemma 6.0.2 we have:

$$\operatorname{dist}\Big(f_{M,T,H}, \textsc{Unate}\Big) \geq \frac{1}{2^n} \cdot \sum_{k \in \overline{M}} \min\Big\{|X_k^+|, |X_k^-|\Big\}.$$

Therefore, it suffices to show that with probability $\Omega(1)$ over $\mathbf{T} \sim \mathcal{E}(M)$ and $\mathbf{H} \sim \mathcal{E}_{\text{no}}(M)$, both $\mathbf{X}_k^+$ and $\mathbf{X}_k^-$ (as random variables derived from $\mathbf{T}$ and $\mathbf{H}$) have size $\Omega(2^n/n)$ for every $k \in \overline{M}$.

To simplify the proof we introduce a new distribution $\mathcal{E}'(M)$ that is the same as $\mathcal{E}(M)$ but conditioned on that every T_i in T contains at least $n^{1/3}$ elements. Our goal is to show that

$$\mathop{\mathbf{Pr}}_{\mathbf{T} \sim \mathcal{E}'(M),\, \mathbf{H} \sim \mathcal{E}_{\text{no}}(M)} \Big[\forall\, k \in \overline{M}, \text{ both } \mathbf{X}_k^+ \text{ and } \mathbf{X}_k^- \text{ have size } \Omega(2^n/n)\Big] = \Omega(1). \tag{6.20}$$

This implies the desired claim over $\mathbf{T} \sim \mathcal{E}(M)$ as the probability of $\mathbf{T} \sim \mathcal{E}(M)$ lying in the support of $\mathcal{E}'(M)$ is at least $1 - \exp\left(-\Omega(\sqrt{n})\right)$. To see this is the case, the probability of

$\mathbf{T}_i$ having less than $n^{1/3}$ many elements can be bounded from above by

$$\begin{aligned}\mathbf{Pr}\left[|\mathbf{T}_i| \le n^{1/3}\right] &= \sum_{j \le n^{1/3}} \binom{n/2}{j} \cdot \left(1 - \frac{1}{\sqrt{n}}\right)^{n/2-j} \cdot \left(\frac{1}{\sqrt{n}}\right)^{j} \\ &\le (n^{1/3}+1) \cdot \binom{n/2}{n^{1/3}} \cdot \left(1 - \frac{1}{\sqrt{n}}\right)^{n/2-n^{1/3}} < e^{-0.49\sqrt{n}}.\end{aligned}$$

Taking a union bound over all $N \approx e^{\sqrt{n}/4}$ terms, we conclude that $\mathbf{T} \sim \mathcal{E}(M)$ lies in the support of $\mathcal{E}'(M)$ with probability at least $1 - \exp(-0.24\sqrt{n})$.

In Claim 6.2.4, we prove a lower bound for the expectation of $|\mathbf{X}|$:

Claim 6.2.4. *We have (below we use* $\mathbf{H}$ *as an abbreviation for* $\mathbf{H} \sim \mathcal{E}_{no}(M)$*)*

$$\mathop{\mathbf{E}}_{\mathbf{T}\sim\mathcal{E}(M),\,\mathbf{H}}\left[|\mathbf{X}|\right] = \Omega(2^n) \quad \textit{and} \quad \mathop{\mathbf{E}}_{\mathbf{T}\sim\mathcal{E}'(M),\,\mathbf{H}}\left[|\mathbf{X}|\right] = \Omega(2^n). \tag{6.21}$$

Proof. By linearity of expectation, we have

$$\mathop{\mathbf{E}}_{\mathbf{T}\sim\mathcal{E}(M),\,\mathbf{H}}\left[|\mathbf{X}|\right] = \sum_{\text{middle } x} \mathop{\mathbf{Pr}}_{\mathbf{T}\sim\mathcal{E}(M),\,\mathbf{H}}\left[x \in \mathbf{X}\right].$$

Fix a string $x \in \{0,1\}^n$ in the middle layers (i.e., $|x_M|$ lies in $n/4 \pm \sqrt{n}$). We decompose the probability on the RHS for x into N disjoint subevents. The ith subevent corresponds to $\mathbf{T}_i$ being the unique term which x satisfies. The probability of the ith subevent is at least

$$\left(1 - \frac{1}{\sqrt{n}}\right)^{\frac{n}{4}+\sqrt{n}} \times \left(1 - \left(1 - \frac{1}{\sqrt{n}}\right)^{\frac{n}{4}-\sqrt{n}}\right)^{N-1} = \Omega\left(\frac{1}{N}\right).$$

As a result, the probability of $x \in \mathbf{X}$ is $N \cdot \Omega(1/N) = \Omega(1)$. The first part of (6.21) follows from the fact that there are $\Omega(2^n)$ many strings x in the middle layers.

The second part of (6.21) follows from the first part and the fact that $|\mathbf{X}| \le 2^n$ and $\mathbf{T} \sim \mathcal{E}(M)$ does not lie in the support of $\mathcal{E}'(M)$ with probability $o(1)$ as shown above. □

Let $\mu^* = \Omega(2^n)$ be the expectation of $|\mathbf{X}|$ over $\mathbf{T} \sim \mathcal{E}'(M)$ and $\mathbf{H} \sim \mathcal{E}_{\text{no}}(M)$, and let p

be the probability of $|\mathbf{X}| \geq \mu^*/2$. Then we have

$$\mu^* \leq p \cdot 2^n + (1-p) \cdot (\mu^*/2) \leq p \cdot 2^n + \mu^*/2$$

and thus, $p = \Omega(1)$. As a result, it suffices to consider a T in the support of $\mathcal{E}'(M)$ that satisfies $|X| \geq \mu^*/2$ and show that, over $\mathbf{H} \sim \mathcal{E}_{\text{no}}(M)$, all $|\mathbf{X}_k^+|$ and $|\mathbf{X}_k^-|$ are $\Omega(2^n/n)$ with probability $\Omega(1)$. To this end, we focus on $\mathbf{X}_k^+$ and then use symmetry and a union bound on all the n sets.

Given T and its X (with $|X| \geq \mu^*/2$), we note that half of $x \in X$ have $x_k = 0$ (since whether $x \in X$ only depends on x_M) and for each $x \in X$ with $x_k = 0$, the probability of $x \in \mathbf{X}_k^+$ (over $\mathbf{H}$) is $1/(2n)$. Hence, the expectation of $|\mathbf{X}_k^+|$ is $|X|/4n \geq \mu^*/8n = \Omega(2^n/n)$. Let $\mu = |X|/4n$. To obtain a concentration bound on $|\mathbf{X}_k^+|$, we apply Hoeffding's inequality over $\mathbf{H} \sim \mathcal{E}_{\text{no}}(M)$ in the next claim.

Claim 6.2.5. *For each $k \in \overline{M}$, we have*

$$\Pr_{\mathbf{H} \sim \mathcal{E}_{no}(M)} \left[\mu - |\mathbf{X}_k^+| \geq \mu/2 \right] \leq \exp\left(-\Omega\left(2^{n^{1/3}}/n^2 \right) \right).$$

Proof. Consider the size of X_k^+ as a function over $h_1, \ldots, h_N$ for a particular fixed T in the support of $\mathcal{E}'(M)$ with $|X| \geq \Omega(2^n)$. We have that X_k^+ is a sum of independent random variables taking values between 0 and $2^{n-n^{1/3}}$, and the expectation of $|\mathbf{X}_k^+|$ is μ because the choices in $\mathbf{H}$ partitions half of X into $2n$ disjoint parts. Therefore, we can now apply Hoeffding's inequality:

$$\Pr_{\mathbf{H} \sim \mathcal{E}_{\text{no}}(M)} \left[\mu - |\mathbf{X}_k^+| \geq \frac{\mu}{2} \right] \leq \exp\left(-\frac{\Omega(2^{2n}/n^2)}{2^{2n-n^{1/3}}} \right)$$

As each term has length at least $n^{1/3}$, each T_i can add at most $b_i < (1/2) \cdot 2^{n-n^{1/3}}$ to $|\mathbf{X}_k^+|$, then

$$\sum_{i \in [N]} b_i^2 \leq 2^{n-n^{1/3}} \sum_{i \in [N]} b_i \leq 2^{2n-n^{1/3}}.$$

This finishes the proof of the claim. □

The same argument works for $|\mathbf{X}_k^-|$. (6.20) then follows from a union bound on $k \in \overline{M}$

and both sets $\mathbf{X}_k^+$ and $\mathbf{X}_k^-$. This finishes the proof of Lemma 6.2.2. □

Given Lemmas 6.2.1 and 6.2.2, our lower bound for testing unateness (Theorem 60) follows directly from the lemma below. We fix $q = n^{2/3}/\log^3 n$ as the number of queries in the rest of the proof. The remainder of this section will prove the following lemma.

Lemma 6.2.6. *Let B be any q-query deterministic algorithm with oracle access to f. Then*

$$\mathop{\mathbf{Pr}}_{\boldsymbol{f}\sim\mathcal{D}_{no}}\Big[B \text{ rejects } \boldsymbol{f}\Big] \leq \mathop{\mathbf{Pr}}_{\boldsymbol{f}\sim\mathcal{D}_{yes}}\Big[B \text{ rejects } \boldsymbol{f}\Big] + o(1).$$

6.2.2 Balanced decision trees

Let B be a q-query deterministic algorithm, i.e., a binary decision tree of depth at most q in which each internal node is labeled a query string $x \in \{0,1\}^n$ and each leaf is labelled "accept" or "reject." Each internal node u has one 0-child and one 1-child. For each internal node u, we use Q_u to denote the set of strings queried so far (not including the query x to be made at u).

Next we give the definition of a q-query tree B being *balanced* with respect to a subset $M \subset [n]$ of size $n/2$ and a string $r \in \{0,1\}^M$ (as the $\mathbf{M}$ and $\boldsymbol{r}$ in the procedure that generates $\mathcal{D}_{\text{yes}}$ and $\mathcal{D}_{\text{no}}$). After the definition we show that, when both $\mathbf{M}$ and $\boldsymbol{r}$ are drawn uniformly at random (as in the procedure), B is balanced with respect to $\mathbf{M}$ and $\boldsymbol{r}$ with probability at least $1 - o(1)$.

Definition 47 (Balance)**.** *We say B is* balanced *with respect to a subset $M \subset [n]$ of size $n/2$ and $r \in \{0,1\}^M$ if for every internal node u of B (letting x be the query at u) and every $Q \subseteq Q_u$, with*

$$A = \Big\{k \in [n] : \forall\, y, y' \in Q,\, y_k = y'_k\Big\} \quad \textit{and} \quad A' = \Big\{k \in [n] : \forall\, y, y' \in Q \cup \{x\},\, y_k = y'_k\Big\}, \tag{6.22}$$

the set $\Delta = A \setminus A'$ having size at least $n^{2/3}\log n$ implies that

$$\Delta_1 = \Big\{k \in \Delta \cap M : x_k \oplus r_k = 0 \textit{ and } \forall\, y \in Q,\, y_k \oplus r_k = 1\Big\} \tag{6.23}$$

has size at least $n^{2/3}\log n/8$.

Lemma 6.2.7. *Let* B *be a* q*-query decision tree. Then* B *is balanced with respect to a subset* $\mathbf{M} \subset [n]$ *of size* $n/2$ *and an* $\boldsymbol{r} \in \{0,1\}^{\mathbf{M}}$*, both drawn uniformly at random, with probability at least* $1 - o(1)$

Proof. Fix an internal node u and a $Q \subseteq Q_u$ such that $|\Delta| \geq n^{2/3}\log n$. Then the probability over the draw of $\mathbf{M}$ and $\boldsymbol{r}$ of $\boldsymbol{\Delta}_1$ being smaller than $n^{2/3}\log n/8$ is at most $\exp(-\Omega(n^{2/3}\log n))$ using the Chernoff bound. The lemma follows by a union bound as there are at most $O(2^q)$ choices for u and 2^q choices for Q. □

Lemma 6.2.6 follows from the following lemma.

Lemma 6.2.8. *Let* B *be a* q*-query tree that is balanced with respect to* M *and* r*. Then we have*

$$\Pr_{\mathbf{T},\mathbf{H}\sim\mathcal{E}_{no}(M),\boldsymbol{s}}\left[B \textit{ rejects } f_{M,\mathbf{T},\mathbf{H},r,\boldsymbol{s}}\right] \leq \Pr_{\mathbf{T},\mathbf{H}\sim\mathcal{E}_{yes}(M),\boldsymbol{s}}\left[B \textit{ rejects } f_{M,\mathbf{T},\mathbf{H},r,\boldsymbol{s}}\right] + o(1). \quad (6.24)$$

where $\mathbf{T} \sim \mathcal{E}(M)$ *and* $\boldsymbol{s} \sim \{0,1\}^{\overline{M}}$.

Proof of Lemma 6.2.6 assuming Lemma 6.2.8. To simplify the notation, in the sequence of equations below we ignore in the subscripts names of distributions from which certain random variables are drawn when it is clear from the context. Using Lemma 6.2.7 and Lemma 6.2.8, we have

$$\begin{aligned}
&\Pr_{\mathbf{M},\mathbf{T},\mathbf{H}\sim\mathcal{E}_{\text{no}}(M),\boldsymbol{r},\boldsymbol{s}}\left[B \text{ rejects } f_{\mathbf{M},\mathbf{T},\mathbf{H},\boldsymbol{r},\boldsymbol{s}}\right] \\
&\quad\leq \frac{1}{2^{n/2}\cdot\binom{n}{n/2}}\cdot\sum_{M,r}\Pr_{\mathbf{T},\mathbf{H}\sim\mathcal{E}_{\text{no}}(M),\boldsymbol{s}}\left[B \text{ rejects } f_{M,\mathbf{T},\mathbf{H},r,\boldsymbol{s}}\right] \\
&\quad\leq \frac{1}{2^{n/2}\cdot\binom{n}{n/2}}\cdot\sum_{M,r:\text{ balanced } B}\Pr_{\mathbf{T},\mathbf{H}\sim\mathcal{E}_{\text{no}}(M),\boldsymbol{s}}\left[B \text{ rejects } f_{M,\mathbf{T},\mathbf{H},r,\boldsymbol{s}}\right] + o(1) \\
&\quad\leq \frac{1}{2^{n/2}\cdot\binom{n}{n/2}}\cdot\sum_{M,r:\text{ balanced } B}\Pr_{\mathbf{T},\mathbf{H}\sim\mathcal{E}_{\text{yes}}(M),\boldsymbol{s}}\left[B \text{ rejects } f_{M,\mathbf{T},\mathbf{H},r,\boldsymbol{s}}\right] + o(1) \\
&\quad\leq \Pr_{\mathbf{M},\mathbf{T},\mathbf{H}\sim\mathcal{E}_{\text{yes}}(M),\boldsymbol{r},\boldsymbol{s}}\left[B \text{ rejects } f_{\mathbf{M},\mathbf{T},\mathbf{H},\boldsymbol{r},\boldsymbol{s}}\right] + o(1).
\end{aligned}$$

This finishes the proof of Lemma 6.2.6. □

To prove Lemma 6.2.8, we may consider an adversary that has M of size $n/2$ and $r \in \{0,1\}^M$ in hand and can come up with any q-query decision tree B as long as B is balanced with respect to M and r. Our goal is to show that any such tree B satisfies (6.24). This inspires us to introduce the definition of *balanced* decision trees.

Definition 48 (Balanced Decision Trees)**.** *A q-query tree B is said to be* balanced *if it is balanced with respect to $M^* = [n/2]$ and $r^* = 0^{[n/2]} \in \{0,1\}^M$. Equivalently, for every internal node u of B and every $Q \subseteq Q_u$ (letting A and A' denote the sets as defined in* (6.22))*, if $\Delta = A \setminus A'$ has size at least $n^{2/3}\log n$, then the set Δ_1 as defined in* (6.23) *using M^* and r^* has size at least $n^{2/3}\log n/8$.*

With Definition 48 in hand, we use the following lemma to prove Lemma 6.2.8.

Lemma 6.2.9. *Let B be a balanced q-query decision tree. Then we have*

$$\Pr_{\mathbf{T},\mathbf{H}\sim\mathcal{E}_{no}(M^*),\boldsymbol{s}}\left[B \text{ rejects } f_{M^*,\mathbf{T},\mathbf{H},r^*,\boldsymbol{s}}\right] \leq \Pr_{\mathbf{T},\mathbf{H}\sim\mathcal{E}_{yes}(M^*),\boldsymbol{s}}\left[B \text{ rejects } f_{M^*,\mathbf{T},\mathbf{H},r^*,\boldsymbol{s}}\right] + o(1), \tag{6.25}$$

where $\mathbf{T} \sim \mathcal{E}(M^*)$ and $\boldsymbol{s} \sim \{0,1\}^{\overline{M^*}}$.

Proof of Lemma 6.2.8 assuming Lemma 6.2.9. Let B be a q-query tree that is balanced with respect to M and $r \in \{0,1\}^M$, which are not necessarily the same as M^* and r^*. Then we use B, M and r to define a new q-query tree B' that is balanced (i.e., with respect to M^* and r^*): B' is obtained by replacing every query x made in B by x', where x' is obtained by first doing an XOR of x with r over coordinates in M and then reordering the coordinates of the new x using a bijection between M and M^*. Note that B' is balanced and satisfies that the LHS of (6.24) for B' is the same as the LHS of (6.25). The same holds the RHS as well. Lemma 6.2.8 then follows from Lemma 6.2.9. □

For simplicity in notation, we fix M and r to be $[n/2]$ and $0^{[n/2]}$ in the rest of the section. We also write $\mathcal{E}$ for $\mathcal{E}(M)$, $\mathcal{E}_{\text{yes}}$ for $\mathcal{E}_{\text{yes}}(M)$, and $\mathcal{E}_{\text{no}}$ for $\mathcal{E}_{\text{no}}(M)$. Given T in the support of

$\mathcal{E}$, H from the support of $\mathcal{E}_{\text{yes}}$ or $\mathcal{E}_{\text{no}}$, and $s \in \{0,1\}^{\overline{M}}$, we write

$$f_{T,H,s} \stackrel{\text{def}}{=} f_{M,T,H,r,s}$$

for convenience. Then the goal (6.25) of Lemma 6.2.9 becomes

$$\Pr_{\mathbf{T},\mathbf{H}\sim\mathcal{E}_{\text{no}},\boldsymbol{s}}\left[B \text{ rejects } f_{\mathbf{T},\mathbf{H},\boldsymbol{s}}\right] \leq \Pr_{\mathbf{T},\mathbf{H}\sim\mathcal{E}_{\text{yes}},\boldsymbol{s}}\left[B \text{ rejects } f_{\mathbf{T},\mathbf{H},\boldsymbol{s}}\right] + o(1),$$

where $\mathbf{T} \sim \mathcal{E}$ and $\boldsymbol{s} \sim \{0,1\}^{\overline{M}}$ in both probabilities.

Remark 64. *Since B works on $f_{\mathbf{T},\mathbf{H},\boldsymbol{s}}$ and r is all-0, the multiplexer $\mathbf{\Gamma_T}$ is first truncated according to $|x_M|$, the number of 1's in the first $n/2$ coordinates. As a consequence, we may assume without loss generality from now on that B only queries strings x that have $|x_M|$ lying between $n/4 \pm \sqrt{n}$. We will refer to them as strings in the middle layers in the rest of the section.*

6.2.3 Balanced signature trees

At a high level we proceed in a similar fashion as in the monotonicity lower bound. We first define a new and stronger oracle model that returns more than just $f(x) \in \{0,1\}$ for each query $x \in \{0,1\}^n$. Upon each query $x \in \{0,1\}^n$, the oracle returns the so-called *signature* of $x \in \{0,1\}^n$ with respect to (T, H, s) when hidden function is $f_{T,H,s}$ (and it will become clear that $f_{T,H,s}(x)$ is determined by the signature of x); in addition, the oracle also reveals the special variable k of a term T_i when the latter is *breached* (see Definition 52). Note that the revelation of special variables is unique in the unateness lower bound. On the other hand, the definition of signatures in this section is much simpler due to the single-level construction of the multiplexer map.

After the introduction of the stronger oracle model, ideally we would like to prove that every q-query deterministic algorithm C with access to the new oracle can only have at most $o(1)$ advantage in rejecting the function $f_{\mathbf{T},\mathbf{H},\boldsymbol{s}}$ when $\mathbf{T} \sim \mathcal{E}$, $\mathbf{H} \sim \mathcal{E}_{\text{no}}$ and $\boldsymbol{s} \sim \{0,1\}^{\overline{M}}$ as compared to $\mathbf{T}$, $\mathbf{H} \sim \mathcal{E}_{\text{yes}}$ and $\boldsymbol{s}$. It turns out that we are only able to prove this when C is represented by a so-called *balanced signature tree*, a definition closely inspired by that of

balanced decision trees in Definition 48. This suffices for us to prove Lemma 6.2.9 since only balanced decision trees are considered there.

Recall the definition of e_i and $e_{i,i'}$ from Section 6.1. We first define signatures syntactically and then semantically. The two definitions below are simpler than their counterparts in Section 6.1 (as we only have one level of multiplexing in Γ_T). By Remark 64, we can assume without loss of generality that every string queried lies in the middle layers.

Definition 49. *We use $\mathfrak{P}$ to denote the set of all triples (σ, a, b), where $\sigma \in \{0,1,*\}^N$ and $a, b \in \{0,1,\perp\}$ satisfy the following properties:*

1. *σ is either 1) the all 0-string 0^N, 2) e_i for some $i \in [N]$, or 3) $e_{i,i'}$ for some $i < i' \in [N]$.*
2. *If σ is of case 1), then $a = b = \perp$. If σ is of case 2), then $a \in \{0,1\}$ and $b = \perp$. Lastly, if σ is of case 3), then we have $a, b \in \{0,1\}$.*

Definition 50. *We say $(\sigma, a, b) \in \mathfrak{P}$ is the* signature *of a string $x \in \{0,1\}^n$ in the middle layers with respect to (T, H, s) if it satisfies the following properties:*

1. *$\sigma \in \{0,1,*\}^N$ is set according to the following three cases: 1) $\sigma = 0^N$ if $T_i(x) = 0$ for all $i \in [N]$; 2) $\sigma = e_i$ if $T_i(x) = 1$ is the unique term that is satisfied by x; 3) $\sigma = e_{i,i'}$ if $i < i'$ and $T_i(x) = T_{i'}(x) = 1$ are the first two terms that are satisfied by x.*
2. *If σ is in case 1), then $a = b = \perp$. If σ is in case 2) with $\sigma = e_i$, then $a = h_i(x \oplus s)$ [4] and $b = \perp$. If σ is in case 3) with $\sigma = e_{i,i'}$, then $a = h_i(x \oplus s)$ and $b = h_{i'}(x \oplus s)$.*

The signature of a set $Q \subset \{0,1\}^n$ of strings in the middle layers with respect to (T, H, s) is the map $\phi\colon Q \to \mathfrak{P}$ such that $\phi(x)$ is the signature of x with respect to (T, H, s).

Next we show that $f_{T,H,s}(x)$ is uniquely determined by the signature of x. Thus, the new oracle is at least as powerful as the standard one. The proof is similar to that of Lemma 6.1.6.

[4] Recall that $x \oplus s$ is the n-bit string obtained from x after an XOR with s over coordinates in $\overline{M}$.

Lemma 6.2.10. *Let T be from the support of $\mathcal{E}$, H be from the support of $\mathcal{E}_{yes}$ or $\mathcal{E}_{no}$ and $s \in \{0,1\}^{\overline{M}}$. Given an $x \in \{0,1\}^n$ in the middle layers, $f_{T,H,s}(x)$ is uniquely determined by the signature (σ, a, b) of x with respect to (T, H, s).*

Proof. Let $f = f_{T,H,s}$. We consider the following three cases:

1. (No term is satisifed) If $\sigma = 0^N$, then $f(x) = 0$.
2. (Unique term satisfied) If If $\sigma = e_i$ for some $i \in [N]$, then $f(x) = h_i(x \oplus s) = a$.
3. (Multiple terms satisfied) If $\sigma = e_{i,i'}$ for some $i < i' \in [N]$, then $f(x) = 1$.

This finishes the proof of the lemma. □

We have defined the signature of x with respect to (T, H, s), which is the first thing that the new oracle returns upon a query x. Let $Q \subset \{0,1\}^n$ be a set of strings in the middle layers (and consider Q as the set of queries made so far by an algorithm). Next we define terms *breached* by Q with respect to a triple (T, H, s). Upon a query x, the new oracle checks if there is any term(s) newly breached after x is queried; if so, the oracle also reveals its special variable in $\overline{M}$.

For this purpose, let $\phi : Q \to \mathfrak{P}$ be the signature of Q with respect to (T, H, s), where $\phi(x) = (\sigma_x, a_x, b_x)$. We say ϕ induces a 5-tuple $(I; P; R; A; \rho)$ if it satisfies the following properties:

1. The set $I \subseteq [N]$ is given by

$$I = \Big\{ i \in [N] : \exists x \in Q \text{ with } \sigma_{x,i} = 1 \Big\}.$$

2. $P = (P_i : i \in I)$ and $R = (R_i : i \in I)$ are two tuples of subsets of Q. For each $i \in I$,

$$P_i = \Big\{ x \in Q : \sigma_{x,i} = 1 \Big\} \quad \text{and} \quad R_i = \Big\{ x \in Q : \sigma_{x,i} = 0 \Big\}.$$

3. $A = (A_i, A_{i,0}, A_{i,1} : i \in I)$ is a tuple of subsets of $[n]$. For each $i \in I$, $A_i = A_{i,0} \cup A_{i,1}$ and

$$A_{i,1} = \Big\{ k \in [n] : \forall x \in P_i, x_k = 1 \Big\} \quad \text{and} \quad A_{i,0} = \Big\{ k \in [n] : \forall x \in P_i, x_k = 0 \Big\}.$$

4. $\rho = (\rho_i : i \in I)$ is a tuple of functions $\rho_i : P_i \to \{0,1\}$ with $\rho_i(x) = a_x$ if either
   $\sigma_x = e_i$
   or $\sigma_x = e_{i,i'}$ for some $i' > i$, and $\rho_i(x) = b_x$ if $\sigma_x = e_{i',i}$ for some $i' < i$, for each $x \in P_i$,
   i.e., $\rho_i(x)$ gives us the value of $h_i(x \oplus s)$ for each $x \in P_i$.

The following fact is reminiscent of Fact 6.1.9.

Fact 6.2.11. *Let $\phi : Q \to \mathfrak{P}$ be the signature of Q with respect to (T, H, s). Then for each $i \in I$, we have $T_i \subseteq A_{i,1} \cap M$, $T_i(x) = 0$ for all $x \in R_i$, and $h_i(x \oplus s) = \rho_i(x)$ for each $x \in P_i$.*

We introduce the similar concept of consistency as in Definition 45.

Definition 51. *Let $(I; P; R; A; \rho)$ be the tuple induced by $\phi : Q \to \mathfrak{P}$. For each $i \in I$, we say P_i is* 1-consistent *if $\rho_i(x) = 1$ for all $x \in P_i$, and* 0-consistent *if $\rho_i(x) = 0$ for all $x \in P_i$. We say P_i is* consistent *if it is either 1-consistent or 0-consistent; we say P_i is* inconsistent *otherwise.*

We are now ready to define terms breached by Q with respect to (T, H, s).

Definition 52 (Breached Terms)**.** *Let $Q \subset \{0,1\}^n$ be a set of strings in the middle layers. Let T be from the support of $\mathcal{E}$, H be from the support of $\mathcal{E}_{yes}$ or $\mathcal{E}_{no}$, and $s \in \{0,1\}^{\overline{M}}$. Let $(I; P; R; A; \rho)$ be the tuple induced by the signature of Q with respect to (T, H, s). We say the ith term is breached by Q with respect to (T, H, s), for some $i \in I$, if at least one of the following two events occurs: (1) P_i is inconsistent or (2) $|A_i \cap \overline{M}| \leq n/10$. We say the ith term is* safe *if it is not breached.*

We can now finish the formal definition of our new oracle model. Upon each query x, the oracle first returns the signature of x with respect to the hidden triple (T, H, s). It then examines if there is any newly breached term(s) (by Definition 52 there can be at most two such terms since x can be added to at most two P_i's) and return the special variable $k \in \overline{M}$ of the newly breached term(s). As a result, if Q is the set of queries made so far, the information returned by the new oracle can be summarized as a 6-tuple $(I; P; R; A; \rho; \delta)$, where

1. $(I; P; R; A; \rho)$ is the tuple induced by the signature of Q with respect to (T, H, s);
2. Let $I_B \subseteq I$ be the set of indices of terms breached by Q, and let $I_S = I \setminus I_B$ denote the safe terms. Then $\delta : I_B \to \overline{M}$ satisfies that $k = \delta(i)$ is the special variable of the ith term in h_i.

We view a q-query deterministic algorithm C with access to the new oracle as a *signature tree*, in which each leaf is labeled "accept" or "reject" and each internal node u is labeled a query string $x \in \{0,1\}^n$ in the middle layers. Each internal node u has $|\mathfrak{P}| \cdot O(n^2)$ children with each of its edges (u, v) labeled by (1) a triple $(\sigma, a, b) \in \mathfrak{P}$ as the signature of x with respect to the hidden (T, H, s), and (2) the special variable of any newly breached (at most two) term(s). Each node u is associated with a set Q_u as the set of queries made so far (not including x), its signature $\phi : Q_u \to \mathfrak{P}$, and a tuple $(I; P; R; A; \rho; \delta)$ as the summary of all information received from the oracle so far. (Note that one can fully reconstruct the signature ϕ from $(I; P; R; A; \rho)$ so it is redundant to keep ϕ. We keep it because sometimes it is (notation-wise) easier to work with ϕ directly.)

Finally we define *balanced* signature trees.

Definition 53 (Balanced Signature Trees)**.** *We say that a signature tree C is* balanced *if for any internal node u of C (letting x be the query to make and $(I; P; R; A; \rho; \delta)$ be the summary so far) and any $i \in I$, $\Delta = \{j \in A_i : x_j$ disagrees with y_j of $y \in P_i\}$ having size at least $n^{2/3} \log n$ implies that $\Delta_1 = \{k \in \Delta \cap M : x_k = 0$ and $\forall\, y \in P_i,\ y_k = 1\}$ has size at least $n^{2/3} \log n/8$.*

Note that the definition above is weaker compared to Definition 48 of balanced decision trees, in the sense that the condition on Δ_1 in the latter applies to any subset of queries $Q \subseteq Q_u$ (instead of only P_i's). Lemma 6.2.9 follows from the lemma below on balanced signature trees.

Lemma 6.2.12. *Let C be a q-query balanced signature tree. Then we have*

$$\Pr_{\mathbf{T},\mathbf{H} \sim \mathcal{E}_{no}, \boldsymbol{s}} \Big[C \text{ rejects } (\mathbf{T}, \mathbf{H}, \boldsymbol{s}) \Big] \leq \Pr_{\mathbf{T},\mathbf{H} \sim \mathcal{E}_{yes}, \boldsymbol{s}} \Big[C \text{ rejects } (\mathbf{T}, \mathbf{H}, \boldsymbol{s}) \Big] + o(1). \tag{6.26}$$

Proof of Lemma 6.2.9 assuming Lemma 6.2.12. Let B be a q-query balanced decision tree. We use B to obtain a q-query algorithm C with access to the new oracle by simulating B as follows: Each time a string x is queried, C uses the signature of x returned by the oracle to extract $f(x)$ (using Lemma 6.2.10) and then continue the simulation of B. One can verify that the corresponding signature tree of C is balanced and the probabilities of C rejecting $(\mathbf{T}, \mathbf{H}, \boldsymbol{s})$ in both cases are the same as B. □

Before moving on to the proof of Lemma 6.2.12, let us remark on how the new oracle may help an algorithm distinguish between functions in $\mathcal{D}_{\text{yes}}$ and $\mathcal{D}_{\text{no}}$. Suppose that a deterministic algorithm C is at some internal node u with a tuple $(I; P; R; A; \rho; \delta)$. For each breached $i \in I_B$, the algorithm knows that h_i is either a dictator or anti-dictator with special variable x_k with $k = \delta(i)$. By inspecting the y_k of a $y \in P_i$ and $\rho_i(y)$, the algorithm can also deduce whether $h_i(x \oplus s)$ is x_k or $\overline{x_k}$. The former suggests that x_k is monotone and the latter suggests that x_k is anti-monotone.

However, unlike monotonicity testing, observing $h_i(x \oplus s) = \overline{x_k}$ has no indication on whether f is drawn from $\mathcal{D}_{\text{yes}}$ or $\mathcal{D}_{\text{no}}$: indeed $h_i(x \oplus s)$ is equally possible to be x_k or $\overline{x_k}$ in both distributions because of the random bit s_k. But if the algorithm observes a so-called *collision*, i.e. $i, i' \in I_B$ such that $h_i(x \oplus s) = x_k$ and $h_i(x \oplus s) = \overline{x_k}$, then one can safely assert that the hidden function belongs to $\mathcal{D}_{\text{no}}$. This gives us the crucial insight (as sketched earlier in Section 6.2.1) that leads to a higher unateness testing lower bound than monotonicity testing: for testing monotonicity, deducing that a variable goes in an anti-monotone direction suffices for a violation; for testing unateness, however, one needs to find a collision in order to observe a violation. While the proof of Lemma 6.2.12 is quite technical, it follows the intuition that with q queries, it is hard for a balanced signature tree to find a collision in breached terms I_B, and when no collision is found, it is hard to tell where the hidden function is drawn from.

6.2.4 Tree pruning

To prove Lemma 6.2.12 on a given balanced q-query signature tree C, we start by identifying a set of *bad edges* of C and using them to prune the tree.

Definition 54. *An edge (u, v) in C is a* bad *edge if at least one of the following events occurs at (u, v) and none of these events occurs along the root-to-u path (letting x be the string queried at u, and $(I_B \cup I_S; P; R; A; \rho; \delta)$ and $(I'_B \cup I'_S; P'; R'; A'; \rho'; \delta')$ be the summaries at u and v, respectively):*

1. *For some $i \in I_S$, $|A_i \setminus A'_i| \geq n^{2/3} \log n$;*
2. *$|I'_B| > n^{1/3} \big/ \log n$; or*
3. *There exist two distinct indices $i, j \in I'_B$ with $\delta'(i) = \delta'(j)$.*

We say a leaf ℓ of C is a *good* leaf if there is no bad edge along the root-to-ℓ path; otherwise, ℓ is *bad*. The following lemma allows us to focus on good leaves. We defer the proof to Section 6.2.6.

Lemma 6.2.13 (Pruning Lemma)**.** *Let C be a balanced q-query signature tree. Then*

$$\Pr_{\mathbf{T},\mathbf{H}\sim\mathcal{E}_{no},\boldsymbol{s}}\Big[(\mathbf{T}, \mathbf{H}, \boldsymbol{s}) \textit{ reaches a bad leaf}\Big] = o(1).$$

We prove the following lemma for good leaves in Section 6.2.14:

Lemma 6.2.14 (Good Leaves are Nice)**.** *For any good leaf ℓ of C, we have*

$$\Pr_{\mathbf{T},\mathbf{H}\sim\mathcal{E}_{no},\boldsymbol{s}}\Big[(\mathbf{T}, \mathbf{H}, \boldsymbol{s}) \textit{ reaches } \ell\Big] \leq (1 + o(1)) \cdot \Pr_{\mathbf{T},\mathbf{H}\sim\mathcal{E}_{yes},\boldsymbol{s}}\Big[(\mathbf{T}, \mathbf{H}, \boldsymbol{s}) \textit{ reaches } \ell\Big].$$

Assuming Lemma 6.2.13 and Lemma 6.2.14, we can prove Lemma 6.2.12:

Proof of Lemma 6.2.12 assuming Lemma 6.2.13 and Lemma 6.2.14. Let L be the set of leaves

of C that are labeled "reject" and let $L^* \subseteq L$ be the good ones in L. Then we have

$$\begin{aligned}
\Pr_{\mathbf{T},\mathbf{H}\sim\mathcal{E}_{\text{no}},\boldsymbol{s}}\Big[C \text{ reject } (\mathbf{T},\mathbf{H},\boldsymbol{s})\Big] &= \sum_{\ell\in L} \Pr_{\mathbf{T},\mathbf{H}\sim\mathcal{E}_{\text{no}},\boldsymbol{s}}\Big[(\mathbf{T},\mathbf{H},\boldsymbol{s}) \text{ reaches } \ell\Big] \\
&\le \sum_{\ell\in L^*} \Pr_{\mathbf{T},\mathbf{H}\sim\mathcal{E}_{\text{no}},\boldsymbol{s}}\Big[(\mathbf{T},\mathbf{H},\boldsymbol{s}) \text{ reaches } \ell\Big] + o(1) \\
&\le (1+o(1))\cdot \sum_{\ell\in L^*} \Pr_{\mathbf{T},\mathbf{H}\sim\mathcal{E}_{\text{yes}},\boldsymbol{s}}\Big[(\mathbf{T},\mathbf{H},\boldsymbol{s}) \text{ reaches } \ell\Big] + o(1) \\
&\le (1+o(1))\cdot \Pr_{\mathbf{T},\mathbf{H}\sim\mathcal{E}_{\text{yes}},\boldsymbol{s}}\Big[C \text{ rejects } (\mathbf{T},\mathbf{H},\boldsymbol{s})\Big] + o(1) \\
&\le \Pr_{\mathbf{T},\mathbf{H}\sim\mathcal{E}_{\text{yes}},\boldsymbol{s}}\Big[C \text{ rejects } (\mathbf{T},\mathbf{H},\boldsymbol{s})\Big] + o(1),
\end{aligned}$$

where we used Lemma 6.2.13 in the second line and Lemma 6.2.14 in the third line. □

6.2.5 Proof of Lemma 6.2.14 for good leaves

The proof of Lemma 6.2.14 is similar in spirit to Lemma 6.1.12 for monotonicity.

Fix a good leaf ℓ in C. We let Q be the set of queries made along the root-to-ℓ path, $\phi : Q \to \mathfrak{P}$ be the signature of Q with $\phi(x) = (\sigma_x, a_x, b_x)$ for each $x \in Q$, and let $(I_B \cup I_S; P; R; A; \rho; \delta)$ be the summary associated with ℓ. Since ℓ is a good leaf, there are no bad edges along the root-to-ℓ path. Combining this with the definition of breached/safe terms, we have the following list of properties:

1. For each $i \in I_S$, $|A_i \cap \overline{M}| \ge n/10$;
2. Every $i \in I_S$ is either 1-consistent or 0-consistent;
3. $|I_B| \le n^{1/3}\big/\log n$; and
4. For any two distinct indices $i, j \in I_B$, we have $\delta(i) \ne \delta(j)$.

Let $D = \{\delta(i) : i \in I_B\} \subset \overline{M}$ be the special variables of breach terms. We have $|D| = |I_B|$.

Next we fix a tuple T from the support of $\mathcal{E}$ such that the probability of $(T, \mathbf{H}, \boldsymbol{s})$ reaching ℓ is positive, when $\mathbf{H} \sim \mathcal{E}_{\text{no}}$ and $\boldsymbol{s} \sim \{0,1\}^{\overline{M}}$. It then suffices to show that

$$\Pr_{\mathbf{H}\sim\mathcal{E}_{\text{yes}},\boldsymbol{s}}\Big[(T,\mathbf{H},\boldsymbol{s}) \text{ reaches } \ell\Big] \ge (1-o(1)) \Pr_{\mathbf{H}\sim\mathcal{E}_{\text{no}},\boldsymbol{s}}\Big[(T,\mathbf{H},\boldsymbol{s}) \text{ reaches } \ell\Big]. \tag{6.27}$$

The properties below follow directly from the assumption that the probability of $(T, \mathbf{H}, \boldsymbol{s})$ reaching ℓ is positive when $\mathbf{H} \sim \mathcal{E}_{\text{no}}$ and $\boldsymbol{s} \sim \{0,1\}^{\overline{M}}$:

1. For every $x \in Q$ and $i \in [N]$ such that $\sigma_{x,i} \in \{0,1\}$, we have $T_i(x) = \sigma_{x,i}$; and
2. For each $i \in I_B$, letting $k = \delta(i)$, there exists a bit b such that $\rho_i(x) = x_k \oplus b$ for all $x \in P_i$.

For each $i \in I_B \cup I_R$ we pick a string y_i from P_i arbitrarily as a representative and let $\alpha_i = \rho_i(y_i)$.

We first derive an explicit expression for the probability over $\mathcal{E}_{\text{no}}$ in (6.27). To this end, we note that, given properties listed above, (T, H, s) (with H from the support of $\mathcal{E}_{\text{no}}$) reaches ℓ iff

1. For each $i \in I_S$, let k be the special variable of h_i. Then we have $k \in A_i \cap \overline{M}$, and h_i is a dictatorship function if $y_{i,k} \oplus s_k = \alpha_i$ or an anti-dictatorship if $y_{i,k} \oplus s_k \neq \alpha_i$;
2. For each $i \in I_B$, the special variable of h_i is the same as $k = \delta(i)$ and similarly, h_i is a dictatorship function if $y_{i,k} \oplus s_k = \alpha_i$ or an anti-dictatorship if $y_{i,k} \oplus s_k \neq \alpha_i$.

Thus, once s is fixed, there is exactly one choice of h_i for each $i \in I_B$ and $|A_i \cap \overline{M}|$ choices of h_i for each $i \in I_S$. Since there are $(n/2) \cdot 2$ choices overall for each h_i, the probability over $\mathcal{E}_{\text{no}}$ in (6.27) is

$$\left(\frac{1}{n}\right)^{|I_B|} \cdot \prod_{i \in I_S} \left(\frac{|A_i \cap \overline{M}|}{n}\right).$$

Next we work on the more involved probability over $\mathcal{E}_{\text{yes}}$ in (6.27). Given properties listed above (T, H, s) (with H from the support of $\mathcal{E}_{\text{yes}}$ so every h_i is a dictatorship function) reaches ℓ iff

1. For each $i \in I_S$, let k be the special variable of the dictatorship function h_i. Then we have $k \in A_i \cap \overline{M}$ and s_k satisfies that $y_{i,k} \oplus s_k = \alpha_i$;
2. For each $i \in I_B$, the special variable of h_i is the same as $k = \delta(i)$ and $y_{i,k} \oplus s_k = \alpha_i$.

Note that once s is fixed, these are independent conditions over h_i's (among the overall

$n/2$ choices for each h_i). As a result, we can rewrite the probability for $\mathcal{E}_{\text{yes}}$ as

$$\mathop{\mathbf{E}}_{\boldsymbol{s}\sim\{0,1\}^{\overline{M}}}\left[\prod_{i\in I}\mathbf{Z}_i\right], \tag{6.28}$$

where $\mathbf{Z}_i$'s are (correlated) random variables that depend on $\boldsymbol{s}$. For each $i \in I_B$, $\mathbf{Z}_i = 2/n$ if

$$\alpha_i = y_{i,\delta(i)} \oplus \boldsymbol{s}_{\delta(i)}$$

and $\mathbf{Z}_i = 0$ otherwise. For each $i \in I_S$, we have

$$\mathbf{Z}_i = \frac{|\{k \in A_i \cap \overline{M} : y_{i,k} \oplus \boldsymbol{s}_k = \alpha_i\}|}{n/2}$$

For some technical reason, for each $i \in I_S$, let $\mathbf{B}_i$ be the following random set that depends on $\boldsymbol{s}$:

$$\mathbf{B}_i = \Big\{k \in (A_i \cap \overline{M}) \setminus D : y_{i,k} \oplus \boldsymbol{s}_k = \alpha_i\Big\}.$$

Using $|D| = |I_B|$, we may now simplify (6.28) by:

$$\mathop{\mathbf{E}}_{\boldsymbol{s}\sim\{0,1\}^{\overline{M}}}\left[\prod_{i\in I}\mathbf{Z}_i\right] = \frac{1}{2^{|I_B|}}\cdot\left(\frac{2}{n}\right)^{|I_B|}\mathop{\mathbf{E}}_{\boldsymbol{s}\sim\{0,1\}^{\overline{M}\setminus D}}\left[\prod_{i\in I_S}\mathbf{Z}_i\right] \;\geq \left(\frac{1}{n}\right)^{|I_B|}\mathop{\mathbf{E}}_{\boldsymbol{s}\sim\{0,1\}^{\overline{M}\setminus D}}\left[\prod_{i\in I_S}\left(\frac{|\mathbf{B}_i|}{n/2}\right)\right].$$

Therefore, it remains to show that

$$\mathop{\mathbf{E}}_{\boldsymbol{s}\sim\{0,1\}^{\overline{M}\setminus D}}\left[\prod_{i\in I_S}\left(\frac{2|\mathbf{B}_i|}{|A_i\cap\overline{M}|}\right)\right] \geq 1 - o(1). \tag{6.29}$$

Next we further simplify (6.29) by introducing new, simpler random variables. We may re-write

$$|\mathbf{B}_i| = \sum_{k\in(A_i\cap\overline{M})\setminus D}\mathbf{X}_{i,k}, \quad \text{where} \quad \mathbf{X}_{i,k} = \begin{cases} 1 & \text{if } y_{i,k}\oplus\boldsymbol{s}_k = \alpha_i \\ 0 & \text{otherwise} \end{cases}$$

For each $i \in I_S$ and $k \in (A_i \cap \overline{M}) \setminus D$, let $\mathbf{Y}_{i,k}$ and $\mathbf{Y}_i$ be the following random variables:

$$\mathbf{Y}_{i,k} = \frac{1 - 2\mathbf{X}_{i,k} + 2\tau_i}{|A_i \cap \overline{M}|} \quad \text{and} \quad \mathbf{Y}_i = \sum_{k \in A_i \cap M \setminus D} \mathbf{Y}_{i,k}, \quad \text{where} \quad \tau_i = \frac{|A_i \cap \overline{M} \cap D|}{2|(A_i \cap \overline{M}) \setminus D|}.$$

(Note that $|(A_i \cap \overline{M}) \setminus D|$ is $\Omega(n)$ so τ_i's are well-defined.) A simple derivation shows that

$$\prod_{i \in I_S} \left(\frac{2|\mathbf{B}_i|}{|A_i \cap \overline{M}|} \right) = \prod_{i \in I_S} \left(1 - \sum_{k \in (A_i \cap M) \setminus D} \mathbf{Y}_{i,k} \right) = \prod_{i \in I_S} \big(1 - \mathbf{Y}_i\big). \tag{6.30}$$

Using the fact that each fraction on the LHS is between 0 and 2, we have that $\mathbf{Y}_i$ always satisfies $|\mathbf{Y}_i| \leq 1$. The difficulty in lowerbounding (6.30) is that $\mathbf{Y}_i$'s are not independent. But with a fixed i, $\mathbf{Y}_{i,k}$'s are indeed independent with respect to the randomness in $\boldsymbol{s}$ and each $\mathbf{Y}_{i,k}$ is either

$$\frac{1}{|A_i \cap \overline{M}|} + O\left(\frac{1}{n^{5/3} \log n}\right) \quad \text{or} \quad -\frac{1}{|A_i \cap \overline{M}|} + O\left(\frac{1}{n^{5/3} \log n}\right)$$

with equal probabilities, where we used the fact that $|A_i \cap \overline{M}| = \Omega(n)$ and $|D| \leq n^{1/3}/\log n$.

For each $i \in I_S$, let $\mathbf{W}_i$ be the random variable defined as

$$\mathbf{W}_i = \begin{cases} \mathbf{Y}_i & \text{if } |\mathbf{Y}_i| \leq \log^2 n/\sqrt{n} \\ 2|I_S| & \text{otherwise} \end{cases}$$

We prove the following claim that helps us avoid the correlation between $\mathbf{Y}_i$'s.

Claim 6.2.15. *The following inequality always holds:*

$$\prod_{i \in I_S} \big(1 - \mathbf{Y}_i\big) \geq \big(1 - o(1)\big) \cdot \left(1 - \sum_{i \in I_S} \mathbf{W}_i\right).$$

Proof. The inequality holds trivially if $|\mathbf{Y}_j| \geq \log^2 n/\sqrt{n}$ for some $j \in I_S$. This is because $|\mathbf{Y}_i| \leq 1$ and thus, the LHS is nonnegative. On the other hand $\mathbf{W}_j = 2|I_S|$ implies that the RHS is negative even when every other $\mathbf{W}_i$ is -1. So we may assume that $|\mathbf{Y}_i| \leq \log^2 n/\sqrt{n}$ for every i. The proof in this case follows directly from Claim **??** in the appendix. □

Given Claim 6.2.15, it suffices to upperbound the expectation of each $\mathbf{W}_i$ over $\boldsymbol{s} \sim \{0,1\}^{\overline{M}\setminus D}$:

$$\mathop{\mathbf{E}}_{\boldsymbol{s}\sim\{0,1\}^{\overline{M}\setminus D}}\left[\mathbf{W}_i\right] \le \mathop{\mathbf{E}}_{\boldsymbol{s}\sim\{0,1\}^{\overline{M}\setminus D}}\left[\mathbf{Y}_i\right]+\left(2|I_S|+1\right)\cdot\mathbf{Pr}_{\boldsymbol{s}}\left[\mathbf{Y}_i \ge \log^2 n/\sqrt{n}\right] = O\left(\frac{1}{n^{2/3}\log n}\right) \tag{6.31}$$

where we used $|I_S| \le n^{2/3}$ and that the probability of $\mathbf{Y}_i \ge \log^2 n/\sqrt{n}$ is superpolynomially small, by a Chernoff bound. Our goal, (6.29), then follows directly from (6.31) and Claim 6.2.15.

6.2.6 Proof of the pruning lemma

Let E be the set of bad edges in C. We start by partitioning E into three (disjoint) subsets E_1, E_2 and E_3 according the the event that occurs at $(u,v) \in E$. Let $(u,v) \in E$ and let $(I_B \cup I_S; P; R; A; \rho; \delta)$ and $(I'_B \cup I'_S; P'; R'; A'; \rho'; \delta')$ be the summaries associated with u and v, respectively. Then

1. $(u,v) \in E_1$ if for some $i \in I_S$, we have $|A_i \setminus A'_i| \ge n^{2/3}\log n$;
2. $(u,v) \in E_2$ if $(u,v) \notin E_1$ and $|I'_B| \ge n^{1/3}/\log n$; or
3. $(u,v) \in E_3$ if $(u,v) \notin E_1 \cup E_2$ and for two distance indices $i, j \in I'_B$, we have $\delta(i) = \delta(j)$.

Note that E_1, E_2 and E_3 are disjoint. Moreover, by the definition of bad edges none of these events occurs at any edge along the root-to-u path.

Our plan below is to show that the probability of $(\mathbf{T}, \mathbf{H}, \boldsymbol{s})$, as $\mathbf{T} \sim \mathcal{E}, \mathbf{H} \sim \mathcal{E}_{\text{no}}$ and $\boldsymbol{s} \sim \{0,1\}^{\overline{M}}$, passing through an edge in E_i is $o(1)$ for each i. The pruning lemma follows from a union bound.

For edge sets E_1 and E_3, we show that for any internal node u of C, the probability of $(\mathbf{T}, \mathbf{H}, \boldsymbol{s})$ taking an edge (u,v) that belongs to E_1 or E_3 is at most $o(1/q)$, conditioning on $(\mathbf{T}, \mathbf{H}, \boldsymbol{s})$ reaching u when $\mathbf{T} \sim \mathcal{E}, \mathbf{H} \sim \mathcal{E}_{\text{no}}$ and $\boldsymbol{s} \sim \{0,1\}^{\overline{M}}$. This allows us to apply Lemma 6.0.3. We handle E_2 using a different argument by showing that, roughly speaking, I_B goes up with very low probability after each round of query and thus, the probability of $|I_B|$ reaching $n^{1/3}/\log n$ is $o(1)$.

Edge Set E_1. Fix an internal node u of C. We show that the probability of $(\mathbf{T}, \mathbf{H}, \boldsymbol{s})$ leaving u with an E_1-edge, conditioning on it reaching u, is $o(1/q)$. It then follows from Lemma 6.0.3 that the probability of $(\mathbf{T}, \mathbf{H}, \boldsymbol{s})$ passing through an E_1-edge is $o(1)$.

Let x be the query made at u, and let $(I_B \cup I_S; P; R; A; \rho; \delta)$ be the summary associated with u. Fix an index $i \in I_S$. We upperbound by $o(1/q^2)$ the conditional probability of $(\mathbf{T}, \mathbf{H}, \boldsymbol{s})$ taking an E_1-edge with $|A_i \setminus A'_i| \geq n^{2/3} \log n$. The claim follows by a union bound on $i \in I_S$ (as $|I| = O(q)$).

Note that either $A'_i = A_i$ or $A'_i = A_i \setminus \Delta$, where

$$\Delta = \Big\{k \in A_i : x_k \text{ disagrees with } y_k \text{ of } y \in P_i\Big\}.$$

Thus, a necessary condition for $|A_i \setminus A'_i| \geq n^{2/3} \log n$ to happen is $|\Delta| \geq n^{2/3} \log n$ and $\mathbf{T}_i(x) = 1$.

Since C is balanced, $|\Delta| \geq n^{2/3} \log n$ implies that

$$\Delta_1 = \Big\{k \in A_i \cap M : x_k = 0 \text{ and } y_k = 1, y \in P_i\Big\}$$

has size at least $n^{2/3} \log n/8$. On the other hand, fix any triple (T_{-i}, H, s), where T_{-i} is a tuple of $N-1$ terms with T_i missing, H is from the support of $\mathcal{E}_{\text{no}}$ and $s \in \{0,1\}^{\overline{M}}$ such that

$$\Pr_{\mathbf{T}_i}\Big[((T_{-i}, \mathbf{T}_i), H, s) \text{ reaches } u\Big] > 0, \tag{6.32}$$

where $\mathbf{T}_i$ is drawn by including each index in M with probability $1/\sqrt{n}$. It suffices to show that

$$\Pr_{\mathbf{T}_i}\Big[((T_{-i}, \mathbf{T}_i), H, s) \text{ reaches } u \text{ and } \mathbf{T}_i(x) = 1\Big] \leq o(1/q^2) \cdot \Pr_{\mathbf{T}_i}\Big[((T_{-i}, \mathbf{T}_i), H, s) \text{ reaches } u\Big]. \tag{6.33}$$

For this purpose, note that given (6.32), the event on the RHS of (6.33) occurs at T_i if and only if T_i is a subset of $A^*_{i,1} = A_{i,1} \cap M$ and $T_i(y) = 0$ for every $y \in R_i$; we use U to denote the set of all such terms T_i (U cannot be empty by (6.32)). On the other hand, the event on the LHS of (6.33) occurs if and only if T_i further avoids picking variables from Δ_1,

i.e. $T_i \subseteq A^*_{i,1} \setminus \Delta_1$. We use V to denote the set of all such T_i's. To prove (6.33), note that we can take any T_i in V, add an arbitrary subset of Δ_1, and the result must be a set in U. As a result we have (note that the bound is very loose here)

$$\frac{\mathbf{Pr}[\mathbf{T}_i \in V]}{\mathbf{Pr}[\mathbf{T}_i \in U]} \leq \left(1 - \frac{1}{\sqrt{n}}\right)^{|\Delta_1|} = o(1/q^2).$$

This finishes the proof for E_1. Next we work on the edge set E_3.

Edge set E_3. Fix an internal node u of C. We show that the probability of $(\mathbf{T}, \mathbf{H}, \boldsymbol{s})$ leaving u with an E_3-edge, conditioning on it reaching u, is $o(1/q)$. By definition, we can assume that there is no bad edge along the root-to-u path and thus, $|I_B| \leq n^{1/3}/\log n$ and I_B has no collision, i.e. there are no distinct $i, j \in I_B$ such that $\delta(i) = \delta(j)$. For $(\mathbf{T}, \mathbf{H}, \boldsymbol{s})$ to leave u with an E_3-edge, it must be the case that some (at most two) terms are breached after the query x and a collision occurs (either between a newly breached term and a term in I_B, or between the two newly breached terms).

Fix a pair (T, s), where T is from the support of $\mathcal{E}$ and $s \in \{0,1\}^{\overline{M}}$, such that $(T, \mathbf{H}, s)$ reaches u with a non-zero probability when $\mathbf{H} \sim \mathcal{E}_{\text{no}}$. It suffices to show that

$$\Pr_{\mathbf{H}}\Big[(T, \mathbf{H}, s) \text{ reaches } u \text{ and a collision occurs}\Big] \leq o(1/q) \cdot \Pr_{\mathbf{H}}\Big[(T, \mathbf{H}, s) \text{ reaches } u\Big]. \tag{6.34}$$

Note that set of (at most two) $i \in I_S$ such that x is added to P_i after it is queried is determined by T (if x starts a new P_i, then this i is safe for sure). If there exists no such i, then the probability on the LHS of (6.34) is 0 since no term is newly breached and we are done. Below we prove (6.34) for the case when $i \in I_S$ is the only index such that x is added to P_i. The case when there are two such i's can be handled similarly.

The proof of (6.34) easily follows from the following simple but useful claim:

Claim 6.2.16. *Let T and s be such that $(T, \mathbf{H}, s)$ reaches u with non-zero probability when $\mathbf{H} \sim \mathcal{E}_{no}$. Then conditioning on reaching u, $\boldsymbol{h}_i$ has its special variable uniformly distributed in $A_i \cap \overline{M}$.*

Proof. As $i \in I_S$, P_i is consistent. For (T, H, s) to reach u, the only condition on h_i and its

special variable k is that (1) if $y_k \oplus s_k = \rho_i(y)$ for some $y \in P_i$, then h_i is a dictatorship function x_k; (2) if $y_k \oplus s_k \neq \rho_i(y)$ for some $y \in P_i$, then h_i is an anti-dictatorship function $\overline{x_k}$. Given T and s, there are $|A_i \cap \overline{M}|$ choices for h_i among the $2 \cdot (n/2)$ choices and they are all equally likely. □

Our goal, (6.34), follows easily from $|A_i \cap \overline{M}| = \Omega(n)$ since $i \in I_S$, Claim 6.2.16, $|I_B| \leq n^{1/3}/\log n$, our choice of $q = n^{2/3}/\log^3 n$, and the fact that, for the event on the LHS to happen, the special variable of $\boldsymbol{h}_i$ must fall inside I_B.

Edge set E_2. Let (u, v) be a bad edge in E_2 with $|I'_B| \geq n^{1/3}/\log n$. We decompose I'_B into K and L: $i \in I'_B$ is in K if *at the edge (u', v^*) along the root-to-v path where i becomes newly breached*, we have $|A^*_i \cap \overline{M}| \leq n/10$, where A^*_i is the set at v^*, and $i \in I'_B$ is in L otherwise (i.e. $|A^*_i \cap \overline{M}| > n/10$ but P^*_i at v^* becomes inconsistent after the query at u'). The claim below shows that K is small:

Claim 6.2.17. *For every E_2-bad edge (u, v), we have $|K| \leq O(n^{1/3}/\log^2 n)$.*

Proof. Fix an $i \in K$ and let (u', v^*) be the edge along the root-to-v path where i becomes breached. Note that when A_i is first created along the path, $A_i = \overline{M}$ and $|A_i \cap \overline{M}| = n/2$ (since at that time P_i consists of a single string). As we walk down the root-to-u^* path, every time a string is added to P_i, the size of A_i can only drop by $n^{2/3} \log n$ (otherwise, this edge is an E_1-edge, contradicting with the assumption that $(u, v) \in E_2$ since E_1 edges have a higher priority) and thus, $|A_i \cap \overline{M}|$ can drop by at most $n^{2/3} \log n$. As a result, we have that $|P^*_i|$ at v^* is at least

$$1 + \frac{n/2 - n/10}{n^{2/3} \log n} = \Omega\left(\frac{n^{1/3}}{\log n}\right).$$

Using the fact that each of the q queries can be added to at most two P_i's, we have

$$|K| \leq \frac{2q}{\Omega(n^{1/3}/\log n)} = O\left(\frac{n^{1/3}}{\log^2 n}\right).$$

This finishes the proof of the claim. □

It follows directly from Claim 6.2.17 that every bad $(u, v) \in E_2$ has $|L| \geq n^{1/3}/(2 \log n)$. This inspires us to consider the following random process of walking down the tree C from

its root, with respect to $(\mathbf{T}, \mathbf{H}, \boldsymbol{s})$ over $\mathbf{T} \sim \mathcal{E}$, $\mathbf{H} \sim \mathcal{E}_{\text{no}}$, and $\boldsymbol{s} \sim \{0,1\}^{\overline{M}}$. As we walk down an edge (u, v) of C, letting $(I_B \cup I_S; P; R; A; \rho; \delta)$ and $(I'_B \cup I'_S; P'; R'; A'; \rho'; \delta')$ be the summaries associated with u and v, if $|A_i \setminus A'_i| \geq n^{2/3} \log n$ for some $i \in I_S$, then we fail and terminate the random process; if not we add the newly breached term(s) i with and $|A'_i \cap \overline{M}| > n/10$ (so P'_i becomes inconsistent), if any, to $\mathbf{L}$. We succeed if $|\mathbf{L}| \geq n^{1/3}/(2 \log n)$, and it suffices for us to show that we succeed with probability $o(1)$ over $\mathbf{T}, \mathbf{H}$ and $\boldsymbol{s}$.

For the analysis, let u be an internal node of C, and fix any pair (T, s) such that $(T, \mathbf{H}, s)$ can reach u with a non-zero probability. As discussed earlier, the set of (at most two) P_i, $i \in I_S$, that the query string x joins is determined only by T. If one of them has $|A_i \setminus A'_i| \geq n^{2/3} \log n$ then the process would always fail; otherwise, we have that $\mathbf{L}$ can grow by at most two and this occurs with probability (over the randomness of $\mathbf{H}$ but conditioning on $(T, \mathbf{H}, s)$ reaching u) at most

$$p = O\left(\frac{n^{2/3} \log n}{n}\right) = O\left(\frac{\log n}{n^{1/3}}\right)$$

because $|A_i \cap \overline{M}| = \Omega(n)$ ($i \in I_S$), the special variable of $\boldsymbol{h}_i$ is uniform over $A_i \cap \overline{M}$ by Claim 6.2.16, and for i to be added to $\mathbf{L}$, the special variable of $\boldsymbol{h}_i$ must lie in $A_i \setminus A'_i$ (of size at most $n^{2/3} \log n$).

In summary, after each query the random process either fails, or if it does not fail, $\mathbf{L}$ can grow by at most two with probability at most p. Therefore, the probability that we succeed is at most

$$\mathop{\mathbf{Pr}}_{\mathbf{m} \sim \text{Bin}(q,p)} \left[2\mathbf{m} \geq \frac{n^{1/3}}{2 \log n}\right] = o(1),$$

since $q = n^{2/3}/\log^3 n$ and $p = O(\log n / n^{1/3})$.

This finishes the proof that $(\mathbf{T}, \mathbf{H}, \boldsymbol{s})$ passes through an edge in E_2 with probability $o(1)$.

6.3 Non-Adaptive One-Sided Unateness Lower Bound

In this section we prove Theorem 61: an $\Omega(n/\log^2 n)$ lower bound on the query complexity of testing unateness for *one-sided* and *non-adaptive* algorithms. This lower bound matches the upper bound of [43] up to a poly-logarithmic factor. Our arguments are an adaptation of Theorem 19 of [63] to the setting of unateness, with one additional observation that allows us to obtain a higher lower bound. Previously [24] proved a lower bound of $\Omega(\sqrt{n})$ for one-sided, non-adaptive algorithms. For the rest of the section, we fix $q = n/\log^2 n$.

For a fixed $n > 0$, we describe a distribution $\mathcal{D}_{\text{no}}$ supported on Boolean functions f over $n+2$ variables. We then show that every $\boldsymbol{f} \sim \mathcal{D}_{\text{no}}$ is $\Omega(1)$-far from unate. An $\boldsymbol{f} \sim \mathcal{D}_{\text{no}}$ is drawn by first drawing an index $\boldsymbol{i} \sim [n]$ uniformly at random, and then letting $\boldsymbol{f} = f_{\boldsymbol{i}}$, where for each $x \in \{0,1\}^n$:

$$f_{\boldsymbol{i}}(0,0,x) = 0,$$
$$f_{\boldsymbol{i}}(0,1,x) = \overline{x_{\boldsymbol{i}}},$$
$$f_{\boldsymbol{i}}(1,0,x) = x_{\boldsymbol{i}},$$
$$f_{\boldsymbol{i}}(1,1,x) = 1.$$

In order to simplify the notation, given $a, b \in \{0,1\}$ and $i \in [n]$, we write $f_{i,ab} \colon \{0,1\}^n \to \{0,1\}$ to denote the function $f_{i,ab}(x) = f_i(a,b,x)$ that agrees with f_i when a and b are the first two inputs.

Figure 6.4 gives a simple visual representation of f_i. We show that f_i is the $\Omega(1)$-far from unate.

Lemma 6.3.1. *For all $i \in [n]$, f_i is $\Omega(1)$-far from unate.*

Proof. This is immediate from Lemma 6.0.2, because there are $\Omega(2^n)$ monotone bi-chromatic edges in direction i, as well as $\Omega(2^n)$ anti-monotone bi-chromatic edges in direction i. □

We consider *non-adaptive, one-sided*, deterministic q-query algorithm B with oracle access to a Boolean function. Note that a non-adaptive, deterministic algorithm B is simply a set of q query strings $x_1, \ldots, x_q$, as well as a decision procedure which outputs "accept" or

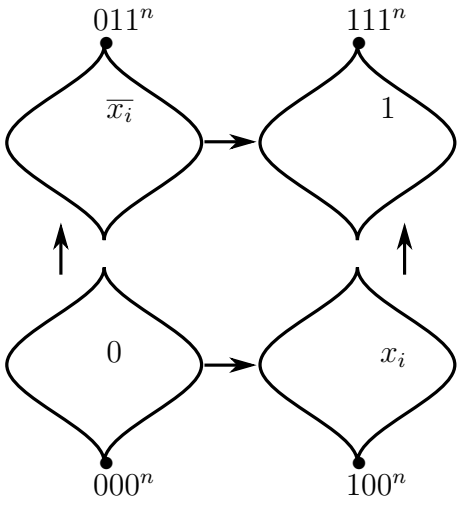

Figure 6.4: An illustration of $f_i \colon \{0,1\}^{n+2} \to \{0,1\}$. The first two coordinates index the sub-cubes.

"reject" given $f(x_k)$ for each $k \in [q]$. Furthermore, since B is one-sided, B outputs "reject" only if it observes a *violation* to unateness (which we formally define next).

Definition 55. *A* violation *to unateness for a function $f \colon \{0,1\}^n \to \{0,1\}$ is a function $v \colon \{0,1\}^n \to (\{0,1\}^n)^2$, such that for each $r \in \{0,1\}^n$: $v(r) = (x, y)$ where $x, y \in \{0,1\}^n$ and*

$$x \oplus r \prec y \oplus r \quad \textit{and} \quad f(x) = 1, f(y) = 0.$$

Intuitively, a violation to unateness consists of a violation to monotonicity, for every possibly orientation $r \in \{0,1\}^n$. We refer to $f^r \colon \{0,1\}^n \to \{0,1\}$ as the function $f^r(x) = f(x \oplus r)$, for any $r \in \{0,1\}^n$. So a violation to unateness for f consists of a violation to monotonicity for each f^r.

Thus, the algorithm B with oracle access to $f \colon \{0,1\}^n \to \{0,1\}$ works in the following way:

1. Query the oracle with queries $Q = \{x_1, \ldots, x_q\} \subset \{0,1\}^n$.
2. If there exists a violation to unateness of f, $v \colon \{0,1\}^n \to (\{0,1\}^n)^2$ where the image of v, $\{v(r) \colon r \in \{0,1\}^n\}$, is a subset of $Q \times Q$, then output "reject"; otherwise, output "accept".

Note that if B does not find a violation, then there exists some unate function $f' \colon \{0,1\}^n \to$

$\{0,1\}$ which is consistent with Q (i.e., $f'(x_k) = f(x_k)$ for all $k \in [q]$). In order to say that B does not find a violation, it suffices to exhibit some $r \in \{0,1\}^n$ such that B does not find a violation to monotonicity of f^r. Given Lemma 6.3.1, Theorem 61 follows from the following lemma:

Lemma 6.3.2. *For any q-query non-adaptive algorithm B, there exists some $r \in \{0,1\}^{n+2}$ such that with probability $1 - o(1)$ over $\boldsymbol{i} \sim [n]$, B does not observe any violations to monotonicity of $f^r_{\boldsymbol{i}}$.*

Proof of Theorem 61 assuming Lemma 6.3.2. Lemma 6.3.2 implies that with probability $1 - o(1)$ over the draw of $\boldsymbol{f} \sim \mathcal{D}_{\text{no}}$, B does not observe any violation to unateness, since there is some $r \in \{0,1\}^{n+2}$ where B does not observe any violation for monotonicity of $\boldsymbol{f}^r$. Thus, any q-query algorithm B does not output "reject" on inputs drawn from $\mathcal{D}_{\text{no}}$ with probability at least $\frac{2}{3}$. □

We now proceed to prove Lemma 6.3.2. For two strings $y, z \in \{0,1\}^n$, we denote the Hamming distance between y and z as $d(y,z) = |\{k \in [n] \colon y_k \neq z_k\}|$.

Lemma 6.3.3. *For any q strings $x_1, \ldots, x_q \in \{0,1\}^n$, there exists an $r \in \{0,1\}^n$ such that for any $j, k \in [q]$, if $x_j \oplus r \prec x_k \oplus r$, then $d(x_j, x_k) \leq 2 \log n$.*

Proof. Consider a random n-bit $\boldsymbol{r} \sim \{0,1\}^n$. Suppose x_j and x_k have $d(x_j, x_k) > 2 \log n$. Then:

$$\mathop{\mathbf{Pr}}_{\boldsymbol{r} \sim \{0,1\}^n} \left[x_j \oplus \boldsymbol{r} \prec x_k \oplus \boldsymbol{r} \right] < 2^{-2 \log n} = n^{-2},$$

since if x_j and x_k differ at i, $\boldsymbol{r}_i$ can only take one of two possible values to make them comparable. Thus we can union bound over all possible pairs of queries with distance at least $2 \log n$ to obtain

$$\mathop{\mathbf{Pr}}_{\boldsymbol{r} \sim \{0,1\}^n} \left[\exists\, j, k \in [q], d(x_j, x_k) > 2 \log n \text{ and } x_j \oplus \boldsymbol{r} \prec x_k \oplus \boldsymbol{r} \right] < n^2 / n^2 = 1.$$

Therefore, there exists an r such that for all $j, k \in [q]$, $x_j \oplus r \prec x_k \oplus r$ implies $d(x_j, x_k) > 2 \log n$. □

Proof of Lemma 6.3.2. Consider a non-adaptive, deterministic algorithm B making q queries $x'_1, \ldots, x'_q \in \{0,1\}^{n+2}$, and let $x_1, \ldots, x_q$ be the last n bits of these strings. We will focus on $x_1, \ldots, x_q$ and refer to the sub-functions the strings query. For example x_k will query the sub-function f_{ab} corresponding to $a = x'_{k,1}$ and $b = x'_{k,2}$. We may partition the set of queries $Q = \{x_1, \ldots, x_q\}$, according to the sub-function queried:

$$\begin{aligned}
Q_{00} &= \{x_k \in Q \colon x'_{k,1} = x'_{k,2} = 0\} \\
Q_{01} &= \{x_k \in Q \colon x'_{k,1} = 0, x'_{k,2} = 1\} \\
Q_{10} &= \{x_k \in Q \colon x'_{k,1} = 1, x'_{k,2} = 0\} \\
Q_{11} &= \{x_k \in Q \colon x'_{k,1} = x'_{k,2} = 1\}.
\end{aligned}$$

Let $r \in \{0,1\}^n$ be the string such that all comparable pairs among $x_1 \oplus r, \ldots, x_q \oplus r$ have distance at most $2 \log n$, which is guaranteed to exist by Lemma 6.3.3. We will show that when $r' = (0, 0, r) \in \{0,1\}^{n+2}$, with probability $1 - o(1)$ over the draw of $\boldsymbol{i} \sim [n]$, B does not observe any violation to monotonicity of $f_{\boldsymbol{i}}^{r'}$.

Consider any $i \in [n]$ and one possible violation to monotonicity, given by the pair (x_k, x_j) where

$$x'_k \oplus r' \prec x'_j \oplus r' \quad \text{and} \quad f_i^{r'}(x'_k) = 1, f_i^{r'}(x'_j) = 0$$

Then $x_k \notin Q_{00}$ and $x_j \notin Q_{11}$ since $f^r_{i,00}$ and $f^r_{i,11}$ are the constant 0 and 1 functions, respectively. Additionally, if $x_j \in Q_{00}$, then $x_k \in Q_{00}$ since $r'_1 = r'_2 = 0$, but this contradicts the fact that $f_i^{r'}(x'_k) = 1$, so $x_j \notin Q_{00}$. Similarly, $x_k \notin Q_{11}$.

Additionally, if $x_k \in Q_{01}$ (or Q_{10}) and $x_j \in Q_{10}$ (or Q_{01}), x'_k and x'_j are incomparable, so $x'_k \oplus r'$ and $x'_j \oplus r'$ are incomparable. Also, for any $i \in [n]$, either $f^r_{i,01}$ or $f^r_{i,10}$ is monotone, so it suffices to consider pairs (x_k, x_j) where either both $x_k, x_j \in Q_{01}$, or both $x_k, x_j \in Q_{10}$. Consider the case $f^r_{i,10}$ is monotone, since the other case is symmetric. Therefore, it suffices to show that with probability $1 - o(1)$ over the choice of $\boldsymbol{i} \sim [n]$, B does not observe any violations to monotonicity for $f^r_{\boldsymbol{i},01}$ from queries in Q_{01}.

Similarly to [63], consider the graph of the queries where x_j and x_k are connected if $x_j \oplus r$ and $x_k \oplus r$ are comparable. Additionally, consider a spanning forest T over this

graph. For any $i \in [n]$, if $f^r_{i,01}(x_j) \neq f^r_{i,01}(x_k)$ when x_j and x_k are connected in T, then there exists an edge in T, (y, z), where $f^r_{i,01}(y) \neq f^r_{i,01}(z)$. Thus, it suffices to upper-bound the probability that some edge (y, z) in T has $f^r_{i,01}(y) \neq f^r_{i,01}(z)$, and this only happens when $y \oplus r$ and $z \oplus r$ differ at index i.

We have:

$$\mathop{\mathbf{Pr}}_{\boldsymbol{i}\sim[n]}\left[\exists (y,z) \in T : f^r_{\boldsymbol{i},01}(y) \neq f^r_{\boldsymbol{i},01}(z)\right] \leq \frac{q \cdot 2\log n}{n}$$

since the two end points of each edge have hamming distance at most $2\log n$ (recall our choice for r). We union bound over at most q edges in T to conclude that with probability at least $1 - 2q\log n/n$ over the draw $\boldsymbol{i} \sim [n]$, B does not observes a violation to monotonicity for $f^r_{\boldsymbol{i},01}$ in Q_{01}. When $q = n/\log^2 n$, this probability is at least $1 - o(1)$. □

6.4 Non-Adaptive Monotonicity Lower Bound

In this section, we present the proof that *non-adaptive* monotonicity testing requires $\tilde{\Omega}(\sqrt{n})$ queries. The previous best non-adaptive lower bound for testing monotonicity is from [54], where they show that for any $c > 0$, testing monotonicity requires $\Omega(n^{1/2-c})$ many queries. Since this lower bound matches the known upper bound from [87], our result is tight up to poly-logarithmic factors. The following distribution and proof is very similar to the work in [34].

We use distributions over Boolean functions very similar to the distributions used in [34]. A function $\boldsymbol{f} \sim \mathcal{D}_{\text{yes}}$ is drawn using the following procedure:

1. Sample $\mathbf{T} \sim \mathcal{E}$ ($\mathcal{E}$ is the same distribution over terms used in Section 6.2). Then $\mathbf{T}$ is used to define the multiplexer map $\mathbf{\Gamma} = \mathbf{\Gamma}_{\mathbf{T}} \colon \{0,1\}^n \to [N] \cup \{0^*, 1^*\}$.
2. Sample $\mathbf{H} = (\boldsymbol{h}_i \colon i \in [N])$ from a distribution $\mathcal{E}_{\text{yes}}$, where each $\boldsymbol{h}_i \colon \{0,1\}^n \to \{0,1\}$ is a random dictatorship Boolean function, i.e., $\boldsymbol{h}_i(x) = x_k$ with k sampled independently and uniformly at random from $[n]$.
3. Finally, $\boldsymbol{f} \colon \{0,1\}^n \to \{0,1\}$ is defined as follows: $\boldsymbol{f}(x) = 1$ if $|x| > (n/2) + \sqrt{n}$;

$\boldsymbol{f}(x) = 0$ if $|x| < (n/2) - \sqrt{n}$; if $(n/2) - \sqrt{n} \leq |x| \leq (n/2) + \sqrt{n}$, we have

$$\boldsymbol{f}(x) = \begin{cases} 0 & \boldsymbol{\Gamma}(x) = 0^* \\ 1 & \boldsymbol{\Gamma}(x) = 1^* \\ \boldsymbol{h}_{\boldsymbol{\Gamma}(x)}(x) & \text{otherwise (i.e., } \boldsymbol{\Gamma}(x) \in [N]) \end{cases}$$

A function $\boldsymbol{f} \sim \mathcal{D}_{\text{no}}$ is drawn using the same procedure, with the only difference being that $\mathbf{H} = (\boldsymbol{h}_i \colon i \in [N])$ is drawn from $\mathcal{E}_{\text{no}}$ (instead of $\mathcal{E}_{\text{yes}}$): each $\boldsymbol{h}_i(x) = \overline{x_k}$ is a random anti-dictatorship Boolean function with k drawn independently and uniformly from $[n]$.

Similarly to Section 6.1, the truncation allows us to show lower bounds against algorithms that query strings in the middle layers. The following two lemmas are easy extensions of Lemma 6.1.1 and Lemma 6.1.2 in Section 6.1.

Lemma 6.4.1. *Every function in the support of $\mathcal{D}_{yes}$ is monotone.*

Lemma 6.4.2. *A function $\boldsymbol{f} \sim \mathcal{D}_{no}$ is $\Omega(1)$-far from monotone with probability $\Omega(1)$.*

Below, we fix $q = \sqrt{n}/\log^2 n$. Recall from Section 6.3 that a non-adaptive, deterministic algorithm B is a set of q query strings $x_1, \ldots, x_q$, as well as a decision procedure which outputs "accept" or "reject" given $f(x_k)$ for each $k \in [q]$. Thus, in order to prove the lower bound, it suffices to prove the following lemma:

Lemma 6.4.3. *Let B be any non-adaptive deterministic algorithm with oracle access to f making $q = \sqrt{n}/\log^2 n$ queries. Then*

$$\Pr_{\boldsymbol{f} \sim \mathcal{D}_{yes}}[B \text{ accepts } \boldsymbol{f}] \leq \Pr_{\boldsymbol{f} \sim \mathcal{D}_{no}}[B \text{ accepts } \boldsymbol{f}] + o(1)$$

We follow in a similar fashion to Subsection 6.2.3 by considering a stronger oracle model that results more than just $f(x) \in \{0, 1\}$. In particular, we will use the oracle model from Subsection 6.2.3, where on query $x \in \{0,1\}^n$, the oracle reveals the signature of x with respect to (T, H) as described in Definition 50. From Lemma 6.2.10, this new oracle is at least as powerful as the standard oracle. Recall the definitions of the 5-tuple $(I; P; R; A; \rho)$ from Subsection 6.2.3. To summarize, the algorithm B with oracle access to the signatures with respect to (T, H) works in the following way:

1. Query the oracle with queries $Q = \{x_1, \ldots, x_q\} \subset \{0,1\}^n$.
2. Receive the full signature map of Q with respect to (T, H), and build the 5-tuple $(I; P; R; A; \rho)$.
3. Output "accept" or "reject".

We think of an algorithm B as a list of possible outcome, $L = \{\ell_1, \ell_2, \ldots\}$, where each outcome corresponds to an execution of the algorithm. Thus, each ℓ_i is labelled with a full-signature map of Q (and therefore, a 5-tuple) as well as "accept" or "reject". These possible outcomes are similar in nature to the leaves in Section 6.1 and Section 6.2.

We proceed in a similar fashion to Section 6.1 and Section 6.2, by first identifying some *bad outcomes*, and then proving that for the remaining good outcomes, B cannot distinguish between $\mathcal{D}_{\text{yes}}$ and $\mathcal{D}_{\text{no}}$. Note that since our algorithm is non-adaptive, B is not a tree; thus, there are no edges like in Section 6.1 and Section 6.2. For the remainder of the section, we let $\alpha > 0$ be a large constant.

Definition 56. *For a fixed 5-tuple, $(I; P; R; A; \rho)$, we say the tuple is* bad *if:*

- *For some $i \in I$, there exists $x, y \in P_i$ where $|\{k \in [n] \mid x_k = y_k = 1\}| \leq (n/2) - \alpha\sqrt{n}\log n$.*
- *For some $i \in I$, P_i is inconsistent (recall definition of inconsistent from Definition 51).*

We will say an outcome ℓ is bad if the 5-tuple at ℓ, given by $(I; P; R; A; \rho)$ from the full signature map at ℓ is bad. Thus, we may divide the outcomes into L_B, consisting of the bad outcomes, and L_G, consisting of the good outcomes. Similarly to Section 6.1 and Section 6.2, Lemma 6.4.3 follows from the following two lemmas.

Lemma 6.4.4. *Let B be a non-adaptive q-query algorithm. Then*

$$\Pr_{\mathbf{T}\sim\mathcal{E},\mathbf{H}\sim\mathcal{E}_{yes}}[(\mathbf{T},\mathbf{H}) \textit{ results an outcome in } L_B] = o(1).$$

We prove the following lemma for good outcomes.

Lemma 6.4.5. *For any non-adaptive, q-query algorithm B, if $\ell \in L_G$ is a good outcome,*

$$\Pr_{\mathbf{T}\sim\mathcal{E},\mathbf{H}\sim\mathcal{E}_{yes}}[(\mathbf{T},\mathbf{H}) \text{ results in outcome } \ell] \leq (1+o(1)) \Pr_{\mathbf{T}\sim\mathcal{E},\mathbf{H}\sim\mathcal{E}_{no}}[(\mathbf{T},\mathbf{H}) \text{ results in outcome } \ell].$$

Proof. Fix a good outcome $\ell \in L_G$, and let $\phi\colon Q \to \mathfrak{P}$ be the associated full signature map and $(I; P; R; A; \rho)$ be the associated 5-tuple. Since $(I; P; R; A; \rho)$ is not bad:

- For all $i \in I$, and $x, y \in P_i$, $|\{k \in [n] \mid x_k = y_k = 1\}| \geq (n/2) - \alpha\sqrt{n}\log n$; hence, by Lemma 19 in [34],

$$\Big||A_{i,1}| - |A_{i,0}|\Big| \leq O(|P_i|\sqrt{n}\log n)$$

- For all $i \in I$, P_i is either 1-consistent, or 0-consistent. We use the ρ_i to denote the value $\rho_i(x)$ shared by all $x \in P_i$.

Consider a fixed T in the support of $\mathcal{E}$ such that the probability of $(T, \mathbf{H})$ resulting in outcome ℓ is positive when $\mathbf{H} \sim \mathcal{E}_{\text{yes}}$. Then it suffices to show that

$$\frac{\mathbf{Pr}_{\mathbf{H}\sim\mathcal{E}_{\text{no}}}[(T,\mathbf{H}) \text{ results in outcome } \ell]}{\mathbf{Pr}_{\mathbf{H}\sim\mathcal{E}_{\text{yes}}}[(T,\mathbf{H}) \text{ results in outcome } \ell]} \geq 1 - o(1).$$

We know that T matches the full signature ϕ at ℓ. Now, to match the a_x and b_x for each $x \in Q$ given in ϕ, H (from either $\mathcal{E}_{\text{yes}}$ and $\mathcal{E}_{\text{no}}$) needs to satisfy the following condition:

- If $H = (h_i\colon i \in [N])$ is from the support of $\mathcal{E}_{\text{yes}}$, then the dictator variable of each h_i, $i \in I$, is in A_{i,ρ_i}.
- If $H = (h_i\colon i \in [N])$ is from the support of $\mathcal{E}_{\text{no}}$, then the dictator variable of each h_i, $i \in I$, is in $A_{i,1-\rho_i}$.
- If $i \notin I$, there is no condition posed on h_i.

As a result, we have:

$$\begin{aligned}\frac{\mathbf{Pr}_{\mathbf{H}\sim\mathcal{E}_{\text{no}}}[(T,\mathbf{H}) \text{ results in outcome } \ell]}{\mathbf{Pr}_{\mathbf{H}\sim\mathcal{E}_{\text{yes}}}[(T,\mathbf{H}) \text{ results in outcome } \ell]} &= \prod_{i\in I}\left(\frac{|A_{i,1-\rho_i}|}{|A_{i,\rho_i}|}\right)\\ &\geq \prod_{i\in I}\left(1-\frac{\left||A_{i,\rho_i}|-|A_{i,1-\rho_i}|\right|}{|A_{i,\rho_i}|}\right)\\ &\geq \prod_{i\in I}\left(1-O\left(\frac{|P_i|\log n}{\sqrt{n}}\right)\right)=1-o(1),\end{aligned}$$

when $q=\sqrt{n}/\log^2 n$. □

We now prove Lemma 6.4.4, which allows us to only consider good outcomes.

Proof of Lemma 6.4.4. We first handle the first case of bad outcomes: some $i\in I$ has $x,y\in P_i$ where $|\{k\in[n]\mid x_k=y_k=1\}\leq(n/2)-\alpha\sqrt{n}\log n$. This case is almost exactly the same as Lemma 16 of [34]. Since the probability some $\mathbf{T}\sim\mathcal{E}$ is sampled with the above event happening is at most:

$$2^{\sqrt{n}}q^2\left(\frac{(n/2)-\alpha\sqrt{n}\log n}{n}\right)^{\sqrt{n}}=q^2\left(1-\alpha n^{-1/2}\log n\right)^{\sqrt{n}}\leq q^2n^{-\alpha}=o(1)$$

since $\alpha>0$ is a large constant and $q^2\leq n$. Thus, by Lemma 19 in [34], all $i\in I$ satisfy

$$\left|[n]\setminus A_{i,0}\setminus A_{i,1}\right|\leq O(|P_i|\sqrt{n}\log n).$$

For the second case, in order for some P_i to be inconsistent, $\boldsymbol{h}_i(x)=x_k$ sampled according to $\mathcal{E}_{\text{yes}}$ must have $k\in[n]\setminus A_{i,0}\setminus A_{i,1}$. Thus, taking a union bound over all possible $i\in I$, the probability over $\mathbf{H}\sim\mathcal{E}_{\text{yes}}$ of resulting in an outcome where some $i\in I$ is inconsistent is at most

$$\sum_{i\in I}\left(\frac{\left|[n]\setminus A_{i,0}\setminus A_{i,1}\right|}{n}\right)\leq\sum_{i\in I}\left(\frac{O(|P_i|\sqrt{n}\log n)}{n}\right)=o(1)$$

since $\sum_{i\in I}|P_i|\leq 2q=2\sqrt{n}/\log^2 n$. □

6.5 Tightness of Distributions for Monotonicity

In this section, we provide the reader with some intuition of why the analyses of [34] and this thesis are tight. In particular, we sketch one-sided algorithms to find violating pairs in the far-from-monotone functions from the distributions considered. We maintain this discussion at a high level.

6.5.1 An $O(n^{1/4})$-query algorithm for distributions of [34]

Belovs and Blais define a pair of distributions $\mathcal{D}^*_{\text{yes}}$ and $\mathcal{D}^*_{\text{no}}$ over functions of n variables. To describe $\mathcal{D}^*_{\text{yes}}$ and $\mathcal{D}^*_{\text{no}}$, recall Talagrand's random DNF [130] (letting $N = 2^{\sqrt{n}}$): A function f drawn from Tal is the disjunction of N terms T_i, $i \in [N]$, where each T_i is the conjunction of $\sqrt{n}$ variables sampled independently and uniformly from $[n]$.

Next we use Tal to define $\mathsf{Tal}_{\pm}$. To draw a function $\boldsymbol{g}$ from $\mathsf{Tal}_{\pm}$, one samples an $\boldsymbol{f}$ from Tal and a random $\sqrt{n}$-subset $\boldsymbol{S}$ of $[n]$.[5] Then $\boldsymbol{g}(x) = \boldsymbol{f}(x^{(\boldsymbol{S})})$, where $x^{(\boldsymbol{S})}$ is the string obtained from x by flipping each coordinate in $\boldsymbol{S}$. Equivalently variables in $T_i \cap \boldsymbol{S}$ appear negated in the conjunction of T_i. The $\mathcal{D}^*_{\text{yes}}$ distribution is then the truncation of Tal, and the $\mathcal{D}^*_{\text{no}}$ distribution is the truncation of $\mathsf{Tal}_{\pm}$. Every $\boldsymbol{f} \sim \mathcal{D}^*_{\text{yes}}$ is monotone by definition; [34] shows that $\boldsymbol{g} \sim \mathcal{D}^*_{\text{no}}$ is far from monotone using the extremal noise sensitivity property of Talagrand functions [107].

We now sketch a $O(n^{1/4})$-query one-sided algorithm that rejects $\boldsymbol{g} \sim \mathcal{D}^*_{\text{no}}$ with high probability. Note that the description below is not a formal analysis; the goal is to discuss the main idea behind the algorithm. Let g be a function in the support of $\mathcal{D}^*_{\text{no}}$ defined by T_i and S with $T'_i = T_i \setminus S$. Then the algorithm starts by sampling a random $x \in \{0,1\}^n$ in the middle layers with $g(x) = 1$. It is likely ($\Omega(1)$ probability by a simple calculation) that:

1. x satisfies a unique term T'_k among all T'_i's.
2. $T_k \cap S$ contains a unique $\ell \in [n]$ (by 1).
3. $T_k = T'_k \cup \{\ell\}$ and x has $x_\ell = 0$ (since $g(x) = 1$).

[5] Formally, $\boldsymbol{S}$ is sampled by including each element of $[n]$ independently with probability $1/\sqrt{n}$.

Assume this is the case, and let A_0 and A_1 denote the set of 0-indices and 1-indices of x, respectively. Then $T'_k \subseteq A_1$ and $\ell \in A_0$.

The first stage of the algorithm goes as follows:

> **Stage 1.** Repeat the following for $n^{1/4}$ times: Pick a random subset $R \subset A_1$ of size $\sqrt{n}$ and query $g(x^{(R)})$. By 1) and 2) above, $g(x^{(R)})) = 1$ if and only if $R \cap T'_k = \emptyset$, which happens with $\Omega(1)$ probability. Let A'_1 denote A_1 after removing those indices of R with $g(x^{(R)})) = 1$ encountered. Then we have $T'_k \subset A'_1$ and most likely, $C = A_1 \setminus A'_1$ has size $\Theta(n^{3/4})$.

After the first stage, the algorithm has shrunk A_1 by $\Theta(n^{3/4})$ while still making sure that variables of T'_k lie in A'_1. In the second stage, the algorithm takes advantage of the smaller A_1 to search for ℓ in A_0, with each query essentially covering $\Theta(n^{3/4})$ indices of A_0:

> **Stage 2.** Randomly partition A_0 into $O(n^{1/4})$ many disjoint parts $A_{0,1}, A_{0,2}, \ldots$, each of size $|C| = \Theta(n^{3/4})$. For each $A_{0,j}$, query $g(x^{(A_{0,j} \cup C)})$. For each $A_{0,j}$ with $\ell \notin A_{0,j}$, g must return 1; for the $A_{0,h}$ with $\ell \in A_{0,h}$, g returns 0 with $\Omega(1)$ probability[6] and when this happens, the algorithm has found a $O(n^{3/4})$-size subset $A_{0,h}$ of A_0 containing ℓ. Let $y = x^{(A_{0,j} \cup C)}$.

Note that the algorithm cannot directly query $g(x^{(A_{0,j})})$ since the new string will be outside of the middle layers (unless $|A_{0,j}| = O(\sqrt{n})$, in which case one needs $\Omega(\sqrt{n})$ queries to cover A_0). This is only achieved by flipping $A_{0,j}$ and C at the same time (in different directions) and this is the reason why we need the first stage to shrink A_1. In the last stage, the algorithm will find a violation for y, by providing $z \prec y$ with $g(z) = 1$.

> **Stage 3.** Randomly partition $A_{0,h}$ into $O(n^{1/4})$ many disjoint parts $\Delta_1, \Delta_2, \ldots$, each of size $O(\sqrt{n})$. For each Δ_i, query $g(y^{(\Delta_i)})$. When $\ell \in \Delta_i$, $g(y^{(\Delta_i)}) = 1$ with probability $\Omega(1)$, and $y^{(\Delta_i)} \prec y$.

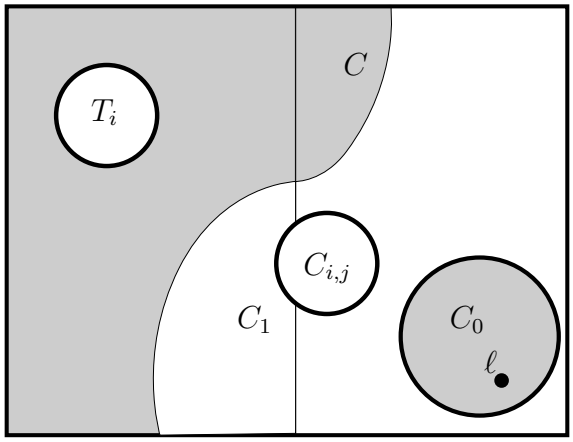

Figure 6.5: A visual representation of the algorithm for finding violations in the two-level Talagrand construction. The whole rectangle represents the set $[n]$, which is shaded for coordinates which are set to 1, and clear for coordinates which are set to 0. T_i is the unique term satisfied and $C_{i,j}$ is the unique clause falsified. The functions $h_{i,j}$ is an anti-dictator of coordinate ℓ. The sets illustrated represent the current knowledge at the end of Stage 3 of the algorithm. Note that $|C_1| = \Theta(n^{5/6})$, $|C| = \Theta(n^{2/3})$, $|C_0| = n^{5/6}$, $|T_i| = |C_{i,j}| = \Theta(\sqrt{n})$.

6.5.2 An $O(n^{1/3})$-query algorithm for our distributions

The idea sketched above can be applied to our far from monotone distribution $\mathcal{D}_{\text{no}}$ from Section 6.1. It is slightly more complicated, since now the algorithm must attack two levels of Talagrand, which will incur the query cost of $\tilde{O}(n^{1/3})$ rather than $O(n^{1/4})$. Similarly to Subsection 6.5.1 above, we will give a high level description, and not a formal analysis. The goal is to show the main obstacle one faces in improving the lower bound.

Assume g is in the support of $\mathcal{D}_{\text{no}}$. The algorithm works in stages and follows a similar pattern to the one described in Subsection 6.5.1 above. We may assume the algorithm has a string $x \in \{0,1\}^n$ where x satisfies a unique term T_i, and falsifies no clauses, so $g(x) = 1$ (this happens with $\Omega(1)$ probability for a random x).

Stage 1. Repeat the following for $n^{1/3}$ times: Pick a random subset $R \subset A_1$ of size $\sqrt{n}$

[6]Informally speaking, this is because the values of $g(x)$ and $g(y)$ essentially become independent when x and y are far from each other.

and query $g(x^{(R)})$. Let A_1' denote A_1 after removing those indices of R with $g(x^{(R)})) = 1$ encountered. Then we have $T_i \subset A_1'$ and most likely, $C_1 = A_1 \setminus A_1'$ has size $\Theta(n^{5/6})$.

The following stages will occur $n^{1/6}$ many times, and each makes $n^{1/6}$ many queries.

Stage 2. Pick a random subset $C_0 \subset A_0$ of size $n^{5/6}$. Let $y = x^{(C_1 \cup C_0)}$ and query $g(y)$. With probability $\Omega(1)$, $g(y)$ satisfies the unique term T_i (as did x), falsifies a unique clause $C_{i,j}$, and $h_{i,j}(y) = 0$. Additionally, with probability $\Omega(n^{-1/6})$, $h_{i,j}(y) = \overline{y_\ell}$, where $\ell \in C_0$.

Assume that $\ell \in C_0$, which happens with $\Omega(n^{-1/6})$ probability. In the event this happens, we will likely find a violation.

Stage 3. Repeat the following for $n^{1/6}$ times: Pick a random subset $R \subset A_0 \setminus C_0$ of size $\sqrt{n}$ and query $g(y^{(R)})$. Let A_0' denote $A_0 \setminus C_0$ after removing those indices of R with $g(y^{(R)}) = 0$. Let $C = (A_0 \setminus C_0) \setminus A_0'$, where very likely $|C| = \Theta(n^{2/3})$. Our sets satisfy the following three conditions: 1) $T_i \subset A_1'$, 2) $C_{i,j} \subset A_0' \cup C_1 \setminus C_0$, and 3) $\ell \in C_0$. See Figure 6.5 for a visual representation of these sets.

Stage 4. Partition C_0 into $O(n^{1/6})$ many disjoint parts $C_{0,1}, C_{0,2}, \ldots$, each of size $\Theta(n^{2/3})$ and query $g(y^{(C_{0,j} \cup C)})$. For each $C_{0,j}$ with $\ell \notin C_{0,j}$ and no new terms are satisfied, g must return 0. If for some sets $C_{0,j}$, g returns 1, then either $\ell \in C_{0,j}$ and no new terms are satisfied, or new terms are satisfied; however, we can easily distinguish these cases with a statistical test.

The final stage is very similar to the final stage of Subsection 6.5.1. After Stage 4, we assume we have found a set $C_{0,j}$ containing ℓ. We further partition $C_{0,j}$ (when $g(y^{(C_{0,j} \cup C)}) = 1$) into $O(n^{1/6})$ parts of size $\sqrt{n}$ to find a violation. One can easily generalize the above algorithm sketch to $O(1)$-many levels of Talagrand. This suggests that the simple extension of our construction to $O(1)$ many levels (which still gives a far-from-monotone function) cannot achieve lower bounds better than $n^{1/3}$.

Chapter 7

An Algorithm for Testing Unateness

Our main contribution is an $\tilde{O}(n^{2/3}/\varepsilon^2)$-query, adaptive algorithm for testing unateness. This essentially settles the problem since it matches the $\tilde{\Omega}(n^{2/3})$ adaptive lower bound of [51] up to a poly-logarithmic factor (when ε is a constant).

Theorem 65 (Main)**.** *There is an $\tilde{O}(n^{2/3}/\varepsilon^2)$-query, adaptive algorithm with the following property: Given $\varepsilon > 0$ and query access to an unknown Boolean function $f\colon \{0,1\}^n \to \{0,1\}$, it always accepts when f is unate and rejects with probability at least $2/3$ when f is ε-far from unate.*

Progress made on the upper bound side is due, in part, to new *directed* isoperimetric inequalities on the hypercube. In particular, [43] and [87] showed that various isoperimetric inequalities on the hypercube have directed analogues, where the edge boundary is now measured by considering anti-monotone bichromatic edges[1]. In addition to the bipartite graph structure implied by the isoperimetric inequality of [87], the algorithm relies on novel applications of the standard binary search procedure on long random paths.

Given a path between two points x and y in the hypercube with $f(x) \neq f(y)$, the binary search (see Figure 7.1) returns a bichromatic edge along the path with $\log \ell$ queries where ℓ is the length of the path. The idea of using binary search in Boolean function property testing is not new. In every application we are aware of in this area (e.g., in testing conjunctions [62, 53], testing juntas [33, 99], unateness [89] and monotonicity [45]), one runs binary search to find bichromatic edges (or pairs, as in testing juntas) that can be directly used to form a violation (or at least part of it) to the property being tested. This is indeed how we use

[1] An edge $(x, x^{(i)})$ (where $x^{(i)}$ denotes the point obtained from x by flipping the ith bit) in $\{0,1\}^n$ is bichromatic if $f(x) \neq f(x^{(i)})$, is monotone (bichromatic) if $x_i = f(x)$, and is anti-monotone (bichromatic) if $x_i \neq f(x)$.

binary search in one of the cases of the algorithm (Case 2) to search for an *edge violation* (i.e., a pair of bichromatic edges along the same variable, one is monotone and the other is anti-monotone). However, in the most challenging case (Case 1) of the algorithm, binary search plays a completely different role. Instead of searching for an edge violation, binary search is used to *preprocess* a large set $S_0 \subseteq [n]$ of variables to obtain a subset $S \subseteq S_0$. This set S is used to search for bichromatic edges more efficiently using a procedure called AE-SEARCH from [52]. Analyzing the performance of S for AE-SEARCH is technically the most demanding part of the proof, where new ideas are needed for understanding the behavior of binary search running along long random paths in the hypercube.

7.0.1 Technical overview

In this section we present a high-level overview of the algorithm, focusing on why and how we use binary search in Case 1 of the algorithm. For simplicity we assume ε is a constant.

First our algorithm rejects a function only when an edge violation to unateness is found. Since an edge violation is a certificate of non-unateness, the algorithm always accepts a function when it is unate and thus, it makes one-sided error. As a result, it suffices to show that the algorithm finds an edge violation with high probability when the unknown function f is far from unate.

For simplicity, we explain Case 1 of the algorithm using the following setting:[2]

> All edge violations of f are along a hidden set $\mathcal{I} \subset [n]$ of $\Omega(n)$ variables. For each variable $i \in \mathcal{I}$, there are $\Theta(2^n/n)$ monotone edges and $\Theta(2^n/n)$ anti-monotone edges. Let P_i^+ denote the set of points incident on monotone edges along i and P_i^- denote the set of points incident on anti-monotone edges along i. The sets P_i^+'s for $i \in \mathcal{I}$ are disjoint, so monotone edges along variables in $\mathcal{I}$ form a matching of size $\Theta(2^n)$; similarly, the sets P_i^-'s are disjoint and anti-monotone edges along $\mathcal{I}$ also form a matching of size $\Theta(2^n)$. Along each $i \notin \mathcal{I}$, there are $\Theta(2^n/\sqrt{n})$ bichromatic edges along i which are all either monotone or anti-monotone, but not both.

[2]The following conditions on the function f are satisfied by the hard functions in [51] used for proving the $\tilde{\Omega}(n^{2/3})$ lower bound.

This particular case will highlight some of the novel ideas in the algorithm and the analysis, so we focus on this case for the technical overview.

An appealing approach for finding an edge violation is to keep running binary search on points $\boldsymbol{x}, \boldsymbol{y}$ that are drawn independently and unifomly at random. Since a function that is far from unate must be ε-far from constant as well, $f(\boldsymbol{x}) \neq f(\boldsymbol{y})$ with a constant probability and when this happens, binary search returns a bichromatic edge. Now in order to analyze the chance of observing an edge violation by repeating this process, two challenges arise. First, the output distribution given by the variable of the bichromatic edge found by binary search can depend on f in subtle ways, and becomes difficult to analyze formally (partly because of its adaptivity). Second, since the influence of variables outside $\mathcal{I}$ is $\Omega(1/\sqrt{n})$, a random path between $\boldsymbol{x}$ and $\boldsymbol{y}$ of $\Omega(n)$ edges may often cross $\Omega(\sqrt{n})$ bichromatic edges along variables outside of $\mathcal{I}$ and $O(1)$ bichromatic edges along variables in $\mathcal{I}$. In this case, binary search will likely return a bichromatic edge along a variable outside $\mathcal{I}$, which is useless for finding an edge violation.

A less adaptive (and thus much simpler to analyze) variant of binary search called AE-SEARCH was introduced [52] to overcome these two difficulties. The subroutine AE-SEARCH(f, x, S)[3] queries f and takes two additional inputs: $x \in \{0,1\}^n$ and a set $S \subseteq [n]$ of variables, uses $O(\log n)$ queries and satisfies the following property:

> **Property of AE-SEARCH:** If $(x, x^{(i)})$ is a bichromatic edge with $i \in S$ and both x and $x^{(i)}$ are $(S \setminus \{i\})$-*persistent* (which for x informally means that $f(x) = f(x^{(\mathbf{T})})$ with high probability when $\mathbf{T}$ is a uniformly random subset of $S \setminus \{i\}$ of half of its size), then AE-SEARCH(f, x, S) finds the edge $(x, x^{(i)})$ with probability at least $2/3$. (1)

[3] See Figure 7.14 in Appendix 7.9 for a formal description of the AE-SEARCH subroutine.

In some sense, AE-SEARCH(f, x, S) efficiently checks whether there exists an $i \in S$ such that $(x, x^{(i)})$ is bichromatic, whereas the trivial algorithm for this task takes $O(|S|)$ queries.[4]

In this simplified setting, the algorithm of [52] starts by drawing a size-$\sqrt{n}$ set $\mathbf{S} \subseteq [n]$ uniformly at random and runs AE-SEARCH$(f, \boldsymbol{x}, \mathbf{S})$ on independent samples $\boldsymbol{x}$ for $n^{3/4}$ times, hoping to find an edge violation. To see why this works we first note that $|\mathbf{S} \cap \mathcal{I}| = \Omega(\sqrt{n})$ with high probability. Moreover, the following property holds for $\mathbf{S}$:

> **Property of the Random Set S:** With $\Omega(1)$ probability over the randomness of $\mathbf{S}$, most $i \in \mathbf{S} \cap \mathcal{I}$ satisfy that most points in P_i^+ and P_i^- are $(\mathbf{S} \setminus \{i\})$-persistent. (2)

We sketch its proof since it highlights the technical challenge we will face later.

First we view the sampling of $\mathbf{S}$ as $\mathbf{S}' \cup \{\boldsymbol{i}\}$, where $\mathbf{S}'$ is a random set of size $\sqrt{n} - 1$ and $\boldsymbol{i}$ is a random variable in $[n]$. Since the influence of each variable in $\mathbf{S}'$ is at most $O(1/\sqrt{n})$, for many points $x \in \{0,1\}^n$ most random paths of length $O(\sqrt{n})$ along variables in $\mathbf{S}'$ starting at x will not cross any bichromatic edges. In other words, most random sets $\mathbf{S}'$ of size $\sqrt{n} - 1$ satisfy that most of points in $\{0,1\}^n$ are $\mathbf{S}'$-persistent with high constant probability. Given that $\cup_i P_i^+$ and $\cup_i P_i^-$ are both $\Omega(1)$-fraction of $\{0,1\}^n$, most points in $\cup_i P_i^+$ and $\cup_i P_i^-$ must be $\mathbf{S}'$-persistent as well. On the other hand, given that $\boldsymbol{i}$ is *independent* from $\mathbf{S}'$ and that $\mathcal{I}$ is $\Omega(n)$, with probability $\Omega(1)$ many points in $P_{\boldsymbol{i}}^+$ and $P_{\boldsymbol{i}}^-$ are $\mathbf{S}'$-persistent. The property of $\mathbf{S}$ follows by an argument of expectation.

With properties of both $\mathbf{S}$ and AE-SEARCH in hand in (1) and (2), as well as the fact that $|\mathbf{S} \cap \mathcal{I}| = \Omega(\sqrt{n})$ with high probability, we expect to find a bichromatic edge along a variable in $\mathbf{S} \cap \mathcal{I}$ after $\sqrt{n}$ executions of AE-SEARCH (since the union of P_i^+ and P_i^- for $i \in \mathbf{S} \cap \mathcal{I}$ consists of $\Omega(1/\sqrt{n})$-fraction of $\{0,1\}^n$). Moreover, the variable is (roughly) uniformly over $\mathbf{S} \cap \mathcal{I}$ and (roughly) equally likely to be monotone or anti-monotone. It follows from the birthday paradox that repeating AE-SEARCH for $O(n^{1/4}) \cdot \sqrt{n}$ rounds is enough to find an edge violation.

[4] See Definition 57 and its relation to the performance of AE-SEARCH in Lemma 7.1.2 for a formal description

The natural question is whether we can make $\mathbf{S}$ larger (e.g., of size $n^{2/3}$) without breaking property (2). This would lead to an $\tilde{O}(n^{2/3})$-query algorithm (for the simplified setting). However, it is no longer true that many random paths of length $\Omega(n^{2/3})$ do not cross bichromatic edges because the influence of variables along variables in $\mathbf{S} \setminus \mathcal{I}$ is $\Omega(1/\sqrt{n})$. Therefore, large $\mathbf{S}$ may not satisfy property (2) and as a result, AE-SEARCH may never output bichromatic edges along variables in $\mathbf{S} \cap \mathcal{I}$. This limit to sets of size at most $O(\sqrt{n})$ was a similar bottleneck in [87], and the connection between $|\mathbf{S}|$ and the total influence of f was later explored in [45]. Indeed, if (2) held for $\mathbf{S}$ of size larger than $\sqrt{n}$, then one could improve on the $O(\sqrt{n})$-query algorithm of [87] for testing monotonicity. Consequently, if one believes that monotonicity testing requires $\Omega(\sqrt{n})$ *adaptive* queries, it is natural to conjecture that the algorithm in [52] is optimal for testing unateness.

The key insight in this work is to *preprocess* the set $\mathbf{S}$ before using AE-SEARCH. For our simplified setting, we first sample $\mathbf{S}_0 \subset [n]$ of size $n^{2/3}$ (much larger than what the analysis in [87, 52, 45] would allow) uniformly at random. Then, we set $\mathbf{S} = \mathbf{S}_0$, and repeat the following steps for $n^{2/3} \cdot \mathrm{polylog}(n)$ many iterations:

> `Preprocess`: Sample $\boldsymbol{x} \in \{0,1\}^n$ uniformly at random. Check if $\boldsymbol{x}$ is $\mathbf{S}$-persistent by drawing $\mathrm{polylog}(n)$ many subsets $\mathbf{T} \subseteq \mathbf{S}$ of half of its size uniformly at random. If a $\mathbf{T}$ with $f(\boldsymbol{x}) \neq f(\boldsymbol{x}^{(\mathbf{T})})$ is found, run binary search on a random path from $f(\boldsymbol{x})$ to $f(\boldsymbol{x}^{(\mathbf{T})})$ to find a bichromatic edge along variable $\boldsymbol{i}$ and remove $\boldsymbol{i}$ from $\mathbf{S}$.

At a high level, the analysis of the algorithm would proceed as follows. At the end of `Preprocess`, for every $i \in \mathbf{S}$, most points in $\{0,1\}^n$ are $(\mathbf{S} \setminus \{i\})$-persistent. Otherwise, `Preprocess` would remove more variables from $\mathbf{S}$ since points which are not $(\mathbf{S} \setminus \{i\})$-persistent cannot be very $\mathbf{S}$-persistent. At the same time, most variables in $\mathbf{S}_0 \cap \mathcal{I}$ at the beginning survive in $\mathbf{S}$ at the end (given that variables in $\mathcal{I}$ have very low influence). It may seem that we can now conclude property (2) holds for $\mathbf{S}$, and that a violation is found after $O(n^{2/3})$ rounds of AE-SEARCH$(f, \boldsymbol{x}, \mathbf{S})$ when $\boldsymbol{x}$ is uniform.

However, the tricky (and somewhat subtle) problem is that, even though most points in $\{0,1\}^n$ are $(\mathbf{S} \cap \{i\})$-persistent for every $i \in \mathbf{S} \cap \mathcal{I}$, it is not necessarily the case that

points *inside* P_i^+ and P_i^- are $(\mathbf{S} \cap \{i\})$-persistent, since $P_i^+ \cup P_i^-$ is only a $O(1/n)$-fraction of the hypercube. Compared with the argument from [52] above for $\sqrt{n}$-sized uniformly random sets, after preprocessing $\mathbf{S}_0$ (which was a uniform random set) with multiple rounds of binary search, the set $\mathbf{S}$ left can be very far from random. More specifically, the set $\mathbf{S}$ obtained from $\mathbf{S}_0$ will heavily depend on the function f and, in principle, a clever adversary could design a function so that `Preprocess` running on $\mathbf{S}_0$ deliberately outputs a set $\mathbf{S}$ that where points in P_i^+ and P_i^- are not $(\mathbf{S} \setminus \{i\})$-persistent.

The main technical challenge is to show that this is not possible when variables in $\mathcal{I}$ have low influence,[5] and the desired property for $\mathbf{S}$ remains valid. To this end, we show that for any variable i with low influence, the following two distributions supported on preprocessed sets $\mathbf{S}$ have small total variation distance. The first distribution samples $\mathbf{S}'_0 \subset [n]$ of size $(n^{2/3} - 1)$ and outputs the set $\mathbf{S}' \cup \{i\}$ obtained from preprocessing $\mathbf{S}'_0$. The second distribution $\mathbf{S}'_0 \subset [n]$ of size $(n^{2/3} - 1)$ and outputs the set $\mathbf{S}$ obtained from preprocessing $\mathbf{S}'_0 \cup \{i\}$. Intuitively this means that a low-influence variable i has little impact on the result $\mathbf{S}$ of `Preprocess` and thus, `Preprocess` is oblivious to i and cannot deliberately exclude P_i^+ and P_i^- from the set of $\mathbf{S}$-persistent points.

To analyze the total variation distance between the results of running `Preprocess` on $\mathbf{S}'_0$ and $\mathbf{S}'_0 \cup \{i\}$, we need to understand how a low-influence variable i can affect the result of a binary search on a *long random path* (given that `Preprocess` is just a sequence of calls to binary search). The random paths have length $|\mathbf{S}_0| = n^{2/3}$ at the beginning of `Preprocess`, and are repeated for $\tilde{O}(n^{2/3})$ rounds. Giving more details, we show that a variable i with influence $\mathbf{Inf}_f[i]$ can affect the result of a binary search on a random path of length ℓ with probability at most $\log \ell \cdot \mathbf{Inf}_f[i]$, instead of the trivial upper bound of $\ell \cdot \mathbf{Inf}_f[i]$, which is the probability that a variable i affects the evaluation of f on vertices of a random path of length ℓ. This is proved in Claim 7.2.3 (although the formal statement is slightly different since we need to introduce a placeholder when running binary search on the set without i so that the two paths have the same length; see Section 7.1.1).

In order to go beyond the assumptions on the function given in this overview, the

[5] In the simplified setting, each variable $i \in \mathcal{I}$ has influence only $O(1/n)$; In the real situation, we need to handle the case even when each variable has influence as high as $1/n^{2/3}$.

algorithm needs to deal with more general cases: (1) Monotone (or anti-monotone) edges of $\mathcal{I}$ may not form a matching, but rather, a large and almost-regular bipartite graph whose existence follows from the directed isoperimetric inequality of [87]. (2) Although [87] implies the existence of such graphs with bichromatic edges from $\mathcal{I}$, there may be more bichromatic edges along $\mathcal{I}$ outside of these two graphs, which would raise the influence of these variables to the point where `Preprocess` is no longer oblivious of these variables. Intuitively, this implies that bichromatic edges which give rise to edge violations are abundant, so finding them becomes easier. This is handled in Case 2, where we give an algorithm (also based on binary search) which finds many bichromatic edges along these high influence variables, and combine it with the techniques from [52] to find an edge violation. (3) The set $\mathcal{I}$ can be much smaller than n, in which case, the techniques from [52] actually achieves better query complexity. We formalize this in Case 3 of the algorithm.

Organization We review preliminaries, recall the binary search procedure and review the definition of persistency and the AE-SEARCH procedure in Section 7.1. We present the preprocessing procedure in Section 7.2 and prove that a low-influence variable has small impact on its output. We use the directed isoperimetric lemma of [87] to establish a so-called Scores Lemma in Section 7.3, which roughly speaking helps us understand how good the set $\mathbf{S}$ is after preprocessing (in terms of using it to run AE-SEARCH to find a bichromatic edge along a certain variable). We separate our main algorithm into three cases in Section 7.4, depending on different combinations of parameters. Case 1, 2 and 3 of the algorithm are presented and analyzed in Sections 7.5, 7.7 and 7.8, respectively. Section 7.6 presents a procedure used in Case 2 to find bichromatic edges of variables with relatively high influence.

7.1 Preliminaries

We will use bold-faced letters such as $\mathbf{T}$ and $\boldsymbol{x}$ to denote random variables. For $n \geq 1$, we write $[n] = \{1, \ldots, n\}$. In addition, we write $g = \tilde{O}(f)$ to mean $g = O(f \cdot \mathrm{polylog}(f))$ and $g = \tilde{\Omega}(f)$ to mean $g = \Omega(f/\mathrm{polylog}(f))$.

For $x \in \{0,1\}^n$, and a set $S \subset [n]$, we write $x^{(S)} \in \{0,1\}^n$ as the point given by letting $x^{(S)}_k = x_k$ for all $k \notin S$, and $x^{(S)}_k = 1 - x_k$ for all $k \in S$ (i.e., $x^{(S)}$ is obtained from x by *flipping* variables in S). When $S = \{i\}$ is a singleton set, we abbreviate $x^{(i)} = x^{(\{i\})}$ and say that $x^{(i)}$ is obtained from x by flipping the ith variable. Throughout the chapter, we use $n+1$ as the name of a *placeholder* variable (i.e., a dummy variable). If $x \in \{0,1\}^n$ and $S \subseteq [n+1]$, then $x^{(S)} := x^{(S \setminus \{n+1\})}$, and in particular, $x^{(n+1)} = x$. We will refer to this as *flipping* variable $n+1$ (see Section 7.1.1) although no change is made on x. For a subset $S \subseteq [n+1]$ and a variable $i \in [n]$, we let $\texttt{Sub}(S,i) \subseteq [n+1]$ be the subset obtained by *substituting* $n+1$ with i and i with $n+1$. In other words,

$$\texttt{Sub}(S,i) = \begin{cases} S & \text{if } i, n+1 \in S \text{ or } i, n+1 \notin S \\ (S \cup \{n+1\}) \setminus \{i\} & \text{if } i \in S \text{ and } n+1 \notin S \\ (S \cup \{i\}) \setminus \{n+1\} & \text{if } n+1 \in S \text{ and } i \notin S. \end{cases}$$

We will at times endow $S \subseteq [n+1]$ with an *ordering* $\pi \colon [|S|] \to S$ which is a bijection indicating that $\pi(i)$ is the ith element of S under π. When $T \subset S$, the ordering $\tau \colon [|T|] \to T$ obtained from π is the unique bijection such that for all $i, j \in T$, $\tau^{-1}(i) < \tau^{-1}(j)$ if and only if $\pi^{-1}(i) < \pi^{-1}(j)$. Moreover, when $S \subseteq [n+1]$ and π is an ordering of S, the ordering π' of $\texttt{Sub}(S,i)$ obtained from π is obtained by substituting $n+1$ with i and i with $n+1$ in the ordering, i.e., $\pi'(k) = \pi(k)$ when $\pi(k) \notin \{i, n+1\}$, $\pi'(k) = n+1$ if $\pi(k) = i$ and $\pi'(k) = i$ if $\pi(k) = n+1$.

Given a Boolean function $f \colon \{0,1\}^n \to \{0,1\}$, and a variable $i \in [n]$, we say that $(x, x^{(i)})$ is a *bichromatic edge* of f along variable i if $f(x) \neq f(x^{(i)})$; it is monotone (bichromatic) if $x_i = f(x)$ and anti-monotone (bichromatic) if $x_i \neq f(x)$. The *influence* of variable i in f is defined as

$$\mathbf{Inf}_f[i] = \mathop{\mathbf{Pr}}_{\boldsymbol{x} \sim \{0,1\}^n} \left[f(\boldsymbol{x}) \neq f(\boldsymbol{x}^{(i)}) \right],$$

which is twice the number of bichromatic edges of f along i divided by 2^n. The *total influence* of f, $\mathbf{I}_f = \sum_{i \in [n]} \mathbf{Inf}_f[i]$, is twice the number of bichromatic edges of f divided

by 2^n. Given distributions μ_1 and μ_2 on some sample space Ω, the *total variation distance* between μ_1 and μ_2 is given by

$$d_{\mathrm{TV}}(\mu_1, \mu_2) = \max_{S \subseteq \Omega} \left| \mu_1(S) - \mu_2(S) \right|.$$

7.1.1 Binary search with a placeholder

We use the subroutine $\texttt{BinarySearch}(f, x, S, \pi)$ described in Figure 7.1, where $f\colon \{0,1\}^n \to \{0,1\}$ is a Boolean function, $x \in \{0,1\}^n$, S is a nonempty subset of $[n+1]$, and π is an ordering of S.

When $S \subseteq [n]$, $\texttt{BinarySearch}(f, x, S, \pi)$ performs as the standard binary search algorithm: $x = x_0, x_1, \ldots, x_{|S|} = x^{(S)}$ is a path from x to $x^{(S)}$ in which x_t is obtained from x_{t-1} by flipping variable $\pi(t) \in S \subseteq [n]$, and when $f(x) \neq f(x^{(S)})$, the binary search is done along this path to find an edge that is bichromatic. Now in general S may also contain $n+1$, which we use as the name of a placeholder variable. Similarly, when $f(x) \neq f(x^{(S)})$, the binary search is done along the path $x = x_0, x_1, \ldots, x_{|S|} = x^{(S)}$ (recall that $x^{(S)}$ is defined as $x^{(S \setminus \{n+1\})}$ when S contains $n+1$) where x_t is obtained from x_{t-1} by flipping variable $\pi(t)$ (in particular, when $\pi(t) = n+1$, $x_t = x_{t-1}$).

Note that even though $n+1$ is a placeholder variable, given $S \subseteq [n+1]$ with $n+1 \in S$ and an ordering π of S, queries made by $\texttt{BinarySearch}(f, x, S, \pi)$ and $\texttt{BinarySearch}(f, x, S \setminus \{n+1\}, \pi')$ (where π' is the ordering of $S \setminus \{n+1\}$ obtained from π) are different, so their results may also be different. We summarize properties of `BinarySearch` in the following lemma.

Lemma 7.1.1. $\texttt{BinarySearch}(f, x, S, \pi)$ *uses* $O(\log n)$ *queries and satisfies the following property. If* $f(x) = f(x^{(S)})$*, it returns* `nil`*; if* $f(x) \neq f(x^{(S)})$*, it returns a variable* $i \in S \setminus \{n+1\}$ *and a point* $y \in \{0,1\}^n$ *along the path from* x *to* $x^{(S)}$ *with ordering* π *such that* $(y, y^{(i)})$ *is bichromatic.*

Subroutine `BinarySearch` (f, x, S, π)

Input: Query access to $f\colon \{0,1\}^n \to \{0,1\}$, a point $x \in \{0,1\}^n$, a nonempty set $S \subseteq [n+1]$ and an ordering π of S.
Output: Either $i \in S$ and a point $y \in \{0,1\}^n$ where $(y, y^{(i)})$ is a bichromatic edge, or `nil`.

1. Query $f(x)$ and $f(x^{(S)})$ and return `nil` if $f(x) = f(x^{(S)})$.
2. Let $m = |S|$ and $x = x_0, x_1, \ldots, x_m = x^{(S)}$ be the sequence of points obtained from x by flipping variables in the order of $\pi(1), \ldots, \pi(m)$: $x_i = x_{i-1}^{(\pi(i))}$. Let $\ell = 0$ and $r = m$.
3. While $r - \ell > 1$ do
4. Let $t = \lceil (\ell + r)/2 \rceil$ and query $f(x_t)$. If $f(x_\ell) \neq f(x_t)$ set $r = t$; otherwise set $\ell = t$.
5. Return $\pi(r)$ and $y = x_\ell$.

Figure 7.1: Description of the binary search subroutine for finding a bichromatic edge.

7.1.2 Persistency with respect to a set of variables

We need the following notion of persistency for points and edges with respect to a set of variables.

Definition 57. *Given a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$, a set $S \subseteq [n+1]$ of variables and a point $x \in \{0,1\}^n$, we say that x is* S-persistent *if the following two conditions hold:*

$$\Pr_{\substack{\mathbf{T}\subseteq S \\ |\mathbf{T}|=\lfloor |S|/2 \rfloor}} \left[f(x) = f(x^{(\mathbf{T})}) \right] \geq 1 - \frac{1}{\log^2 n} \quad \textit{and} \quad \Pr_{\substack{\mathbf{T}\subseteq S \\ |\mathbf{T}|=\lfloor |S|/2 \rfloor + 1}} \left[f(x) = f(x^{(\mathbf{T})}) \right] \geq 1 - \frac{1}{\log^2 n}.$$

where $\mathbf{T}$ is a subset of S of certain size drawn uniformly at random. Note that when $S = \emptyset$, every point in $\{0,1\}^n$ is trivially S-persistent.

Let e be an edge in $\{0,1\}^n$. We say that e is S-persistent *if both points of e are S-persistent.*

The notion of persistency above is useful because it can be used to formulate a clean sufficient condition for AE-SEARCH (f, x, S) to find a bichromatic edge $(x, x^{(i)})$ for some $i \in S$ with high probability. This is captured in Lemma 7.1.2 (see Lemma 6.5 in [52]) below.

For completeness we include the description of AE-SEARCH [52] and the proof of Lemma 7.1.2 in Appendix 7.9.

Lemma 7.1.2. *Given a point $x \in \{0,1\}^n$ and a set $S \subseteq [n+1]$,* AE-SEARCH(f, x, S) *makes $O(\log n)$ queries to f, and returns either an $i \in S$ such that $(x, x^{(i)})$ is a bichromatic edge, or "fail."*

Let $(x, x^{(i)})$ be a bichromatic edge of f along $i \in [n]$. If $i \in S$ and $(x, x^{(i)})$ is $(S \setminus \{i\})$-persistent, then both AE-SEARCH(f, x, S) *and* AE-SEARCH$(f, x^{(i)}, S)$ *output i with probability at least $2/3$.*

Lemma 7.1.2 has the following immediate corollary.

Corollary 7.1.3. *Given a set $S \subseteq [n+1]$ and a point $x \in \{0,1\}^n$, there exists at most one variable $i \in S$ such that $(x, x^{(i)})$ is both bichromatic and $(S \setminus \{i\})$-persistent.*

Proof. If the condition holds for both $i \neq j \in S$, then from Lemma 7.1.2 AE-SEARCH(f, x, S) would return both i and j with probability at least $2/3$, a contradiction. □

7.2 Preprocessing Variables

Our goal in this section is to present a preprocessing procedure called `Preprocess`. Given query access to a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$, a nonempty set $S_0 \subseteq [n+1]$ (again, $n+1$ serves here as a placeholder variable), an ordering π of S_0 and a parameter $\xi \in (0,1)$, `Preprocess`(f, S_0, π, ξ) makes $(|S_0|/\xi)\cdot\text{polylog}(n)$ queries and returns a subset $\mathbf{S}$ of S_0. At a high level, `Preprocess` keeps running `BinarySearch` to remove variables from S_0 until the set $\mathbf{S} \subseteq S_0$ left satisfies that at least $(1-\xi)$-fraction of points in $\{0,1\}^n$ are $\mathbf{S}$-persistent (recall Definition 57).

In addition to proving the above property for `Preprocess` in Lemma 7.2.1, we show in Lemma 7.2.2 the following: When $i \in S_0 \subseteq [n]$ has low influence, then the result of running `Preprocess` on S_0 is *close* (see Lemma 7.2.1 for the formal statement) to that of running it on `Sub`(S_0, i) (in which we substitute i with the placeholder variable $n+1$).

Subroutine `CheckPersistence` (f, S, π, ξ)

Input: Query access to $f\colon \{0,1\}^n \to \{0,1\}$, a nonempty set $S \subseteq [n+1]$, an ordering π of S and a parameter $\xi \in (0,1)$.
Output: Either `nil` or a variable $i \in S$.

1. Repeat the following steps $\log^4 n/\xi$ many times:
   a) Sample a point $\boldsymbol{x}$ from $\{0,1\}^n$ uniformly at random.
   b) Flip a fair coin and perform one of the following tasks:
      - Sample $\mathbf{T} \subseteq S$ with size $\lfloor |S|/2 \rfloor$ uniformly. Run $\texttt{BinarySearch}(f, \boldsymbol{x}, \mathbf{T}, \boldsymbol{\pi}')$ where $\boldsymbol{\pi}'$ is the ordering of $\mathbf{T}$ defined by π restricted on $\mathbf{T}$. If $\texttt{BinarySearch}(f, \boldsymbol{x}, \mathbf{T}, \boldsymbol{\pi}')$ returns a variable i and a point y, output i.
      - Sample $\mathbf{T} \subseteq S$ with size $\lfloor |S|/2 \rfloor + 1$ uniformly. Run $\texttt{BinarySearch}(f, \boldsymbol{x}, \mathbf{T}, \boldsymbol{\pi}')$ where $\boldsymbol{\pi}'$ is the ordering of $\mathbf{T}$ defined by π restricted on $\mathbf{T}$. If $\texttt{BinarySearch}(f, \boldsymbol{x}, \mathbf{T}, \boldsymbol{\pi}')$ returns a variable i and a point y, output i.
2. If `BinarySearch` always returned `nil`, output `nil`.

Figure 7.2: Description of the subroutine `CheckPersistence`.

7.2.1 The preprocessing procedure

The procedure `Preprocess` (f, S_0, π, ξ) is described in Figure 7.3. It uses a subroutine `CheckPersistence` (f, S, π, ξ) described in Figure 7.2. Roughly speaking, `CheckPersistence` checks if at least $(1-\xi)$-fraction of points in $\{0,1\}^n$ are S-persistent for the current set S. This is done by sampling points $\boldsymbol{x}$ and subsets $\mathbf{T}$ of S of the right sizes uniformly at random, and checking if $f(\boldsymbol{x}) = f(\boldsymbol{x}^{(\mathbf{T})})$, for $\log^4 n/\xi$ many rounds. If `CheckPersistence` finds $\boldsymbol{x}$ and $\mathbf{T}$ such that $f(\boldsymbol{x}) \neq f(\boldsymbol{x}^{(\mathbf{T})})$, it runs binary search on them to find a bichromatic edge along some variable $i \in S$ and outputs i; otherwise it returns `nil`.

The main property we prove for `CheckPersistence` (see Lemma 7.10.1 in Appendix 7.10) is that when the fraction of points that are not S-persistent is at least ξ, it returns a variable $i \in S$ with high probability.

The procedure `Preprocess` (f, S_0, π, ξ) sets $S = S_0$ and $\tau = \pi$ at the beginning and keeps calling `CheckPersistence` (f, S, τ, ξ) and removing the variable `CheckPersistence`(f, S, τ, ξ)

Procedure `Preprocess` (f, S_0, π, ξ)

Input: Query access to $f\colon \{0,1\}^n \to \{0,1\}$, a nonempty set $S_0 \subseteq [n+1]$, an ordering π of S_0 and a parameter $\xi \in (0,1)$.
Output: A subset $S \subseteq S_0$.

1. Initially, let $S = S_0$ and $\tau = \pi$.
2. While S is nonempty do
3. Run `CheckPersistence` (f, S, τ, ξ).
4. If it returns `nil`, return S; otherwise (it returns an $i \in S$), remove i from S and τ.
5. Return S (which must be the empty set to reach this line).

Figure 7.3: Description of the procedure `Preprocess` for preprocessing a set of variables.

returns from both S and the ordering τ, until `CheckPersistence` returns `nil` or S becomes empty in which case `Preprocess` terminates and returns S. As a result, `Preprocess` makes at most $|S_0|$ calls to `CheckPersistence`. Using the property of `CheckPersistence` from Lemma 7.10.1, it is unlikely that ξ-fraction of points are not S-persistent but somehow `CheckPersistence` (f, S, τ, ξ) returns `nil`. This implies that at least $(1-\xi)$-fraction of $\{0,1\}^n$ are $\mathbf{S}$-persistent for $\mathbf{S} =$ `Preprocess` (f, S_0, π, ξ) at the end with high probability.

We summarize our discussion above in the following lemma but delay its proof to Appendix 7.10 since it follows from standard applications of Chernoff bounds and union bounds.

Lemma 7.2.1. *Given a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$, a nonempty $S_0 \subseteq [n+1]$, an ordering π of S_0 and a parameter $\xi \in (0,1)$,* `Preprocess` (f, S_0, π, ξ) *makes at most $O(|S_0| \log^5 n/\xi)$ queries to f and with probability at least $1 - \exp\left(-\Omega(\log^2 n)\right)$, it outputs a subset $\mathbf{S} \subseteq S_0$ such that at least $(1-\xi)$-fraction of points in $\{0,1\}^n$ are $\mathbf{S}$-persistent.*

7.2.2 Low influence variables have low impact on `Preprocess`

In the rest of the section, we show that when $S_0 \subseteq [n]$, a variable $i \in S_0$ with low influence $\mathbf{Inf}_f[i]$ has low impact on the result of $\mathbf{S} =$ `Preprocess` (f, S_0, π, ξ). More formally,

we show that one can substitute i by the placeholder $n+1$ and the result of running `Preprocess` on $\texttt{Sub}(S_0, i)$ is almost the same (after substituting $n+1$ back to i in the result of `Preprocess`).

This is made more precise in the following lemma:

Lemma 7.2.2. *Let $f\colon \{0,1\}^n \to \{0,1\}$ be a Boolean function. Let $i \in S_0 \subseteq [n]$, π be an ordering of S and $\xi \in (0,1)$. Let $S_0' = \texttt{Sub}(S_0, i)$ be the subset of $[n+1]$ and let π' be the ordering of S_0' obtained from π by substituting i with $n+1$. Then we have*

$$d_{\mathrm{TV}}\Big(\texttt{Preprocess}(f, S_0, \pi, \xi), \texttt{Sub}\Big(\texttt{Preprocess}(f, S_0', \pi', \xi), i\Big)\Big) \le O\left(\frac{|S_0| \log^5 n}{\xi}\right) \cdot \mathbf{Inf}_f[i].$$

Because `Preprocess` keeps calling `CheckPersistence` which keeps calling `BinarySearch`, we start the proof of Lemma 7.2.2 with the following claim concerning the binary search procedure.

Claim 7.2.3. *Let $i \in S \subseteq [n]$ and π be an ordering of S. Let $S' = \texttt{Sub}(S, i)$, and π' be the ordering of S' obtained from π by substituting i with $n+1$. We let $\boldsymbol{u}$ and $\boldsymbol{v}$ be the random variables where*

- *$\boldsymbol{u}$ is the output of $\texttt{BinarySearch}(f, \boldsymbol{x}, S, \pi)$ when $\boldsymbol{x}$ is drawn from $\{0,1\}^n$ uniformly, and*
- *$\boldsymbol{v}$ is the output of $\texttt{BinarySearch}(f, \boldsymbol{z}, S', \pi')$ when $\boldsymbol{z}$ is drawn from $\{0,1\}^n$ uniformly.*

Then, we have $d_{\mathrm{TV}}(\boldsymbol{u}, \boldsymbol{v}) \le O(\log n) \cdot \mathbf{Inf}_f[i]$.

Proof. Our plan is to show that for every point $x \in \{0,1\}^n$ with a certain property, we have

$$\Big\{\texttt{BinarySearch}(f, x, S, \pi), \texttt{BinarySearch}(f, x^{(i)}, S, \pi)\Big\} \tag{7.1}$$

as a multiset is the same as

$$\Big\{\texttt{BinarySearch}(f, x, S', \pi'), \texttt{BinarySearch}(f, x^{(i)}, S', \pi')\Big\}. \tag{7.2}$$

It turns out that the property holds for most points in $\{0,1\}^n$. The lemma then follows.

To describe the property we let $m = |S| = |S'|$ and let $k = \pi^{-1}(i)$ (with $\pi'(k) = n+1$). We let $J \subseteq [0:m]$ denote the set of indices taken by variables ℓ and r (see Figure 7.1 for settings of ℓ and r) in an execution of `BinarySearch` along a path of length m that outputs the kth edge at the end. For example, ignoring the rounding issue, J always contains $0, m$ and $m/2$: these are indices of the first three points that binary search examines. It contains $3m/4$ if $k > m/2$, or $m/4$ if $k \leq m/2$, so on and so forth. The set J also always contains $k-1$ and k: these are indices of the last two points that binary search examines before returning the kth edge.

Now we describe the property. Given $x \in \{0,1\}^n$ we let $x = x_0, \ldots, x_m = x^{(S)}$ with $x_t = x_{t-1}^{(\pi(t))}$ for all $t \in [m]$. We let $\mathcal{C}(x)$ be the indicator of the condition that:

$$f(x_j) = f(x_j^{(i)}), \quad \text{for all } j \in J. \tag{7.3}$$

We show that $\boldsymbol{x} \sim \{0,1\}^n$ satisfies $\mathcal{C}(\boldsymbol{x})$ with high probability. Because $\boldsymbol{x}$ is drawn uniformly from $\{0,1\}^n$, $\boldsymbol{x}_j$ defined above is also distributed uniformly for each $j \in J$ and thus, the probability that a specific $j \in J$ violates the condition above is at most $\mathbf{Inf}_f[i]$. It then follows from a union bound over $j \in J$ that the fraction of points that violate the condition $\mathcal{C}(x)$ is at most $\mathbf{Inf}_f[i] \cdot O(\log n)$.

It suffices to prove that when $x \in \{0,1\}^n$ satisfies $\mathcal{C}(x)$, the two multisets in (7.1) and (7.2) are the same. To this end we write down the two paths in the multiset (7.1) that start with x and $x^{(i)}$ as

$$x_0, x_1, \ldots, x_m \quad \text{and} \quad y_0, y_1, \ldots, y_m$$

in which $x_t = x_{t-1}^{(\pi(t)}$ and $y_t = x_t^{(i)}$. Similarly we write down the two paths for (7.2) as

$$z_0, z_1, \ldots, z_m \quad \text{and} \quad w_0, w_1, \ldots, w_m,$$

in which we have $z_t = x_t$ for all $t < k$ and $z_t = y_t$ for all $t \geq k$; $w_t = y_t$ for all $t < k$ and

$w_t = x_t$ for all $t \geq k$. It follows from the property (7.3) of x that

$$f(x_j) = f(y_j) = f(z_j) = f(w_j), \quad \text{for all } j \in J. \tag{7.4}$$

Since $0, m \in J$ we have that $f(x_0) = f(x_m)$ implies the same holds for y, z and w in which case (7.1) and (7.2) are trivially the same since they all return `nil`. So we assume below that $f(x_0) \neq f(x_m)$ and thus, all four binary searches return a variable and a bichromatic edge.

Next, since $k - 1, k \in J$ we have

$$f(x_{k-1}) = f(x_k) = f(y_{k-1}) = f(y_k) = f(z_{k-1}) = f(z_k) = f(w_{k-1}) = f(w_k).$$

As a result, the kth edge is not bichromatic in all four paths and thus, during each run of binary search, k is removed from the interval $[\ell : r]$ (see Figure 7.1) after a certain number of rounds. Moreover, it follows from the definition of J and (7.4) that in all four runs of binary search, the values of ℓ and r are the same at the moment when k is removed from consideration (i.e., at the first time when either ℓ or r is updated so that $k \notin [\ell : r]$. We consider two cases for the values of ℓ and r.

1. $\ell > k$: In this case, $\texttt{BinarySearch}\,(f, x, S, \pi)$ continues to search on the path $x_\ell, \ldots, x_r$ and $\texttt{BinarySearch}\,(f, x^{(i)}, S', \pi')$ continues to search on the path $w_\ell, \ldots, w_r$ which is the same as $x_\ell, \ldots, x_r$ given that $\ell > k$. As a result, their outputs are the same. Similarly we have that $\texttt{BinarySearch}\,(f, x^{(i)}, S, \pi)$ is the same as $\texttt{BinarySearch}\,(f, x, S', \pi')$ in this case. See Figure 7.4 for example executions.
2. $r < k$: In this case, $\texttt{BinarySearch}\,(f, x, S, \pi)$ continues to search on the path $x_\ell, \ldots, x_r$ and $\texttt{BinarySearch}\,(f, x, S', \pi')$ continues to search on the path $z_\ell, \ldots, z_r$ which is the same as $x_\ell, \ldots, x_r$ given that $r < k$. As a result, their outputs are the same. Similarly we have that $\texttt{BinarySearch}\,(f, x^{(i)}, S, \pi)$ is the same as $\texttt{BinarySearch}\,(f, x^{(i)}, S', \pi')$ in this case. See Figure 7.5 for example executions.

As a result, the two multisets are the same when x satisfies the condition $\mathcal{C}(x)$. □

Claim 7.2.3 gives the following corollary using a union bound:

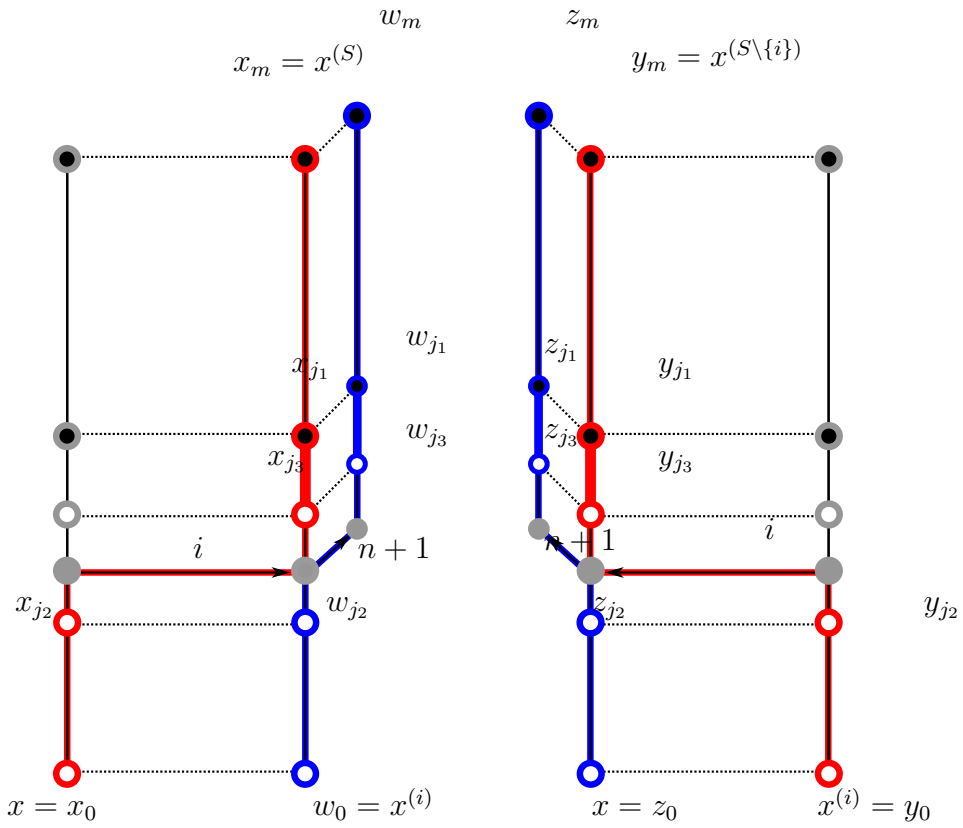

Figure 7.4: Example executions of $\texttt{BinarySearch}(f, x, S, \pi)$ and $\texttt{BinarySearch}(f, x^{(i)}, S', \pi')$ on the left-hand side, and executions of $\texttt{BinarySearch}(f, x, S', \pi')$ and $\texttt{BinarySearch}(f, x^{(i)}, S, \pi)$ on the right-hand side, assuming that $\mathcal{C}(x)$ is satisfied, and corresponding to the case when $k \leq \ell$. Queries made only during executions of $\texttt{BinarySearch}(f, x, S, \pi)$ and $\texttt{BinarySearch}(f, x^{(i)}, S, \pi)$ are displayed by red dots, and the corresponding paths considered are outlined in red; queries made only during executions of $\texttt{BinarySearch}(f, x, S', \pi')$ and $\texttt{BinarySearch}(f, x^{(i)}, S', \pi')$ are displayed by blue dots, and the corresponding paths considered are outlined in blue. Points are filled in with black if f evaluates to 1, and points which are not filled in if f evaluates to 0. Dotted lines indicates that condition $\mathcal{C}(x)$ or the fact that $n+1$ is a dummy variable implies points evaluate to the same value under f. From the above executions, it is clear to see that $\texttt{BinarySearch}(f, x, S, \pi)$ on the left-hand side considers the path (drawn in red) between x_{j_3} and x_{j_1}, and $\texttt{BinarySearch}(f, x^{(i)}, S', \pi')$ considers the path between w_{j_3} and w_{j_1} (drawn in blue); since variable $n+1$ represents a dummy variable, f has the same evaluation on both of these paths, so both output the same variable. Similarly, $\texttt{BinarySearch}(f, x^{(i)}, S, \pi)$ on the right-hand side considers the path (drawn in red) between y_{j_3} and y_{j_1}, and $\texttt{BinarySearch}(f, x^{(i)}, S', \pi')$ considers the same path between z_{j_3} and z_{j_1} (drawn in blue); as a result of $n+1$ being a dummy variable, both output the same variable.

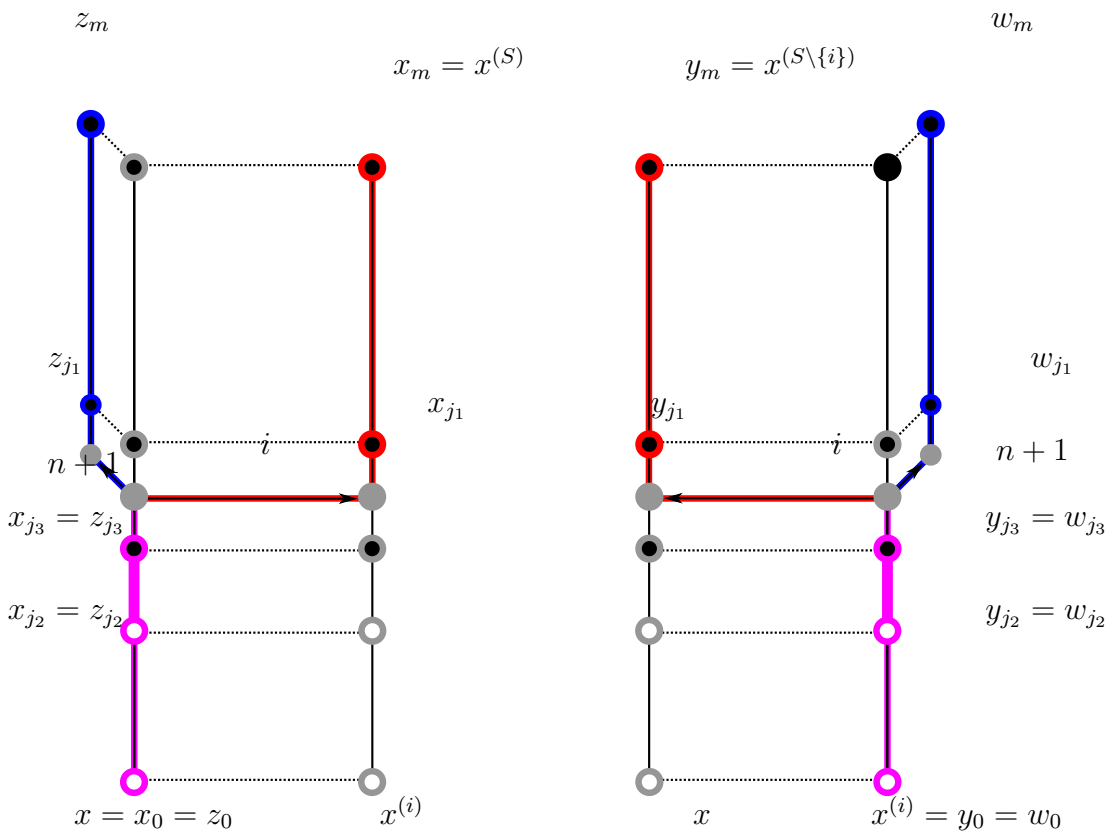

Figure 7.5: Example executions of $\texttt{BinarySearch}(f, x, S, \pi)$ and $\texttt{BinarySearch}(f, x, S', \pi')$ on the left-hand side, and executions of $\texttt{BinarySearch}(f, x^{(i)}, S, \pi)$ and $\texttt{BinarySearch}(f, x^{(i)}, S', \pi')$ on the right-hand side, assuming that $\mathcal{C}(x)$ is satisfied, and corresponding to the case when $k > r$. Queries made only during executions of $\texttt{BinarySearch}(f, x, S, \pi)$ and $\texttt{BinarySearch}(f, x^{(i)}, S, \pi)$ are displayed by red dots, and the corresponding paths considered are outlined in red; queries made only during executions of $\texttt{BinarySearch}(f, x, S', \pi')$ and $\texttt{BinarySearch}(f, x^{(i)}, S', \pi')$ are displayed by blue dots, and the corresponding paths considered are outlined in blue; queries which are made during both are displayed with purple dots, and the intersection of the paths considered in both are purple. Similarly to Figure 7.4, points filled in evaluate to 1 under f, and points which are not filled in evaluates to 0 under f. Dotted lines implies points evaluate to the same value under f. Note that $\texttt{BinarySearch}(f, x, S, \pi)$ considers the path (drawn in purple) between x_{j_2} and x_{j_3}, and $\texttt{BinarySearch}(f, x, S', \pi')$ considers the same path between z_{j_2} and z_{j_3}; thus, both output the same variable. Similarly, $\texttt{BinarySearch}(f, x^{(i)}, S, \pi)$ considers the path (drawn in purple) between y_{j_2} and y_{j_3}, and $\texttt{BinarySearch}(f, x^{(i)}, S', \pi')$ considers the same path between w_{j_2} and w_{j_3}; as a result, both output the same variable.

Corollary 7.2.4. *Let $i \in S \subseteq [n]$ and π be an ordering of S. Let $S' = \texttt{Sub}(S, i)$ and π' be the ordering of S' obtained from π by substituting i with $n+1$. Then we have*

$$d_{\mathrm{TV}}\Big(\texttt{CheckPersistence}(f, S, \pi, \xi), \texttt{CheckPersistence}(f, S', \pi', \xi)\Big) \leq O\left(\frac{\log^5 n}{\xi}\right) \cdot \mathbf{Inf}_f[i].$$

Proof. We use the following coupling to run $\texttt{CheckPersistence}(f, S, \pi, \xi)$ and $\texttt{CheckPersistence}(f, S', \pi'$ in parallel.

For each round of `CheckPersistence` we first flip a fair coin and draw a subset $\mathbf{T}$ of S of the size indicated by the coin uniformly. Then we couple the binary search on $\boldsymbol{x} \sim \{0,1\}^n$ and $\mathbf{T}$ and the binary search on $\boldsymbol{z} \sim \{0,1\}^n$ and $\texttt{Sub}(\mathbf{T}, i)$ using the best coupling between them.

It then follows from Claim 7.2.3 and a union bound over the $\log^4 n/\xi$ rounds that the probability of this coupling of $\texttt{CheckPersistence}(f, S, \pi, \xi)$ and $\texttt{CheckPersistence}(f, S', \pi', \xi)$ returning different results is at most

$$(\log^4 n/\xi) \cdot \mathbf{Inf}_f[i] \cdot O(\log n).$$

This finishes the proof of the corollary. □

Now we prove Lemma 7.2.2.

Proof of Lemma 7.2.2. Let $m = |S| = |S'|$. For each $j \in [m]$, let $\mathbf{X}_j$ denote the output of the jth call to `CheckPersistence` in $\texttt{Preprocess}(f, S_0, \pi, \xi)$ with $\mathbf{X}_j$ set to `nil` by default if the procedure terminates before the jth call. Similarly we use $\mathbf{Y}_j$ to denote the output of the jth call in $\texttt{Preprocess}(f, S_0', \pi', \xi)$. Let $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_m)$ and $\mathbf{Y} = (\mathbf{Y}_1, \ldots, \mathbf{Y}_m)$. Then $\mathbf{X} = \mathbf{Y}$ implies that $\mathbf{S} = \texttt{Preprocess}(f, S_0, \pi, \xi)$ is the same as $\mathbf{S}' = \texttt{Preprocess}(f, S_0', \pi', \xi)$. As a result, it suffices to show that

$$d_{\mathrm{TV}}(\mathbf{X}, \mathbf{Y}) \leq \frac{m \log^5 n}{\xi} \cdot \mathbf{Inf}_f[i].$$

To this end, we first note that by Corollary 7.2.4 the total variation distance between $\mathbf{X}_1$ and $\mathbf{Y}_1$ is at most $\beta := O(\log^5 n/\xi) \cdot \mathbf{Inf}_f[i]$. On the other hand, note that if the outputs

from the first $\ell - 1$ calls in $\texttt{Preprocess}(f, S_0, \pi, \xi)$ and $\texttt{Preprocess}(f, S_0', \pi', \xi)$ are the same, say $a_1, \ldots, a_{\ell-1}$, then before the ℓth call, the set $\mathbf{S}$ in the former still contains i and the $\mathbf{S}'$ in the latter can be obtained by substituting its i with $n+1$. It follows from Corollary 7.2.4 that, for any $\ell > 1$ and any $a_1, \ldots, a_{\ell-1}$, the total variation distance between the distribution of $\mathbf{X}_\ell$ conditioning on $\mathbf{X}_1 = a_1, \ldots, \mathbf{X}_{\ell-1} = a_{\ell-1}$ and the distribution of $\mathbf{Y}_\ell$ conditioning on $\mathbf{Y}_1 = a_1, \ldots, \mathbf{Y}_{\ell-1} = a_{\ell-1}$ is also at most β. We prove that these properties together imply that $d_{\mathrm{TV}}(\mathbf{X}, \mathbf{Y}) \leq m\beta$, from which the lemma follows.[6]

For this purpose we use the following coupling of $\mathbf{X}$ and $\mathbf{Y}$. First we use the best coupling for the distribution of $\mathbf{X}_1$ and the distribution of $\mathbf{Y}_1$ to draw $(\boldsymbol{a}_1, \boldsymbol{b}_1)$. Then we draw $(\boldsymbol{a}_2, \boldsymbol{b}_2)$ from the the best coupling for the distribution of $\mathbf{X}_2$ conditioning on $\mathbf{X}_1 = \boldsymbol{a}_1$ and the distribution of $\mathbf{Y}_2$ conditioning on $\mathbf{Y}_1 = \boldsymbol{b}_1$. We then repeat until $(\boldsymbol{a}_m, \boldsymbol{b}_m)$ is drawn. It follows from the description that the marginal distribution of $\boldsymbol{a} = (\boldsymbol{a}_1, \ldots, \boldsymbol{a}_m)$ is the same as $\mathbf{X}$ and the marginal distribution of $\boldsymbol{b} = (\boldsymbol{b}_1, \ldots, \boldsymbol{b}_m)$ is the same as $\mathbf{Y}$. Moreover, we have

$$\begin{aligned} d_{\mathrm{TV}}(\mathbf{X}, \mathbf{Y}) &\leq \mathbf{Pr}\left[\boldsymbol{a} \neq \boldsymbol{b}\right] \\ &= \mathbf{Pr}\left[\boldsymbol{a}_1 \neq \boldsymbol{b}_1\right] + \mathbf{Pr}\left[\boldsymbol{a}_1 = \boldsymbol{b}_1 \wedge \boldsymbol{a}_2 \neq \boldsymbol{b}_2\right] + \cdots + \mathbf{Pr}\left[\boldsymbol{a}_j = \boldsymbol{b}_j \text{ for } j < m \wedge \boldsymbol{a}_m \neq \boldsymbol{b}_m\right], \end{aligned}$$

which is at most $m\beta$ by the description of the coupling and properties of $\mathbf{X}$ and $\mathbf{Y}$. □

7.3 The Scores Lemma

By definition when f is ε-far from unate, $f(x \oplus a)$ is ε-far from monotone for every $a \in \{0,1\}^n$. This means that we can utilize the directed isoperimetric inequality of [87] to show the existence of relatively large and almost-regular bipartite graphs that consist of bichromatic edges (see Definition 60 and Lemma 7.3.3). The goal of this section is to show that, using these bipartite graphs, there exist certain probability distributions over subsets of variables such that a set $\mathbf{S}$ drawn from any of these distributions can be used to search for bichromatic edges via AE-SEARCH efficiently.

[6] We suspect that this is probably known in the literature but were not able to find a reference.

To this end, we start by introducing three distributions $\mathcal{H}_{\xi,m}, \mathcal{D}_{\xi,m}$ and $\mathcal{P}_{i,m}$ in Section 7.3.1. We then use them to define a *score* for each variable $i \in [n]$ which aims to quantify the chance of finding a bichromatic edge along i using AE-SEARCH and a set $\mathbf{S}$ drawn from some of those distributions. Finally we prove the Scores Lemma in Section 7.3.2, which shows that the sum of scores over $i \in [n]$ is large when f is ε-far from unate and has total influence $O(\sqrt{n})$.

7.3.1 Distributions $\mathcal{D}_{\xi,m}, \mathcal{H}_{\xi,m}$ and $\mathcal{P}_{i,m}$ and the definition of scores

We start by defining two distributions $\mathcal{D}_{\xi,m}$ and $\mathcal{H}_{\xi,m}$.

Definition 58. *Given $\xi \in (0,1)$ and $m : 1 \leq m \leq n$, we let $\mathcal{D}_{\xi,m}$ denote the following distribution supported on subsets of $[n]$: $\mathbf{S} \sim \mathcal{D}_{\xi,m}$ is drawn by first sampling a subset $\mathbf{S}_0$ of $[n]$ of size m and an ordering $\boldsymbol{\pi}$ of $\mathbf{S}_0$ uniformly at random. We then call $\texttt{Preprocess}(f, \mathbf{S}_0, \boldsymbol{\pi}, \xi)$ to obtain $\mathbf{S}$.*

Similarly, let $\mathcal{H}_{\xi,m}$ denote the following distribution supported on subsets of $[n+1]$: $\mathbf{S} \sim \mathcal{H}_{\xi,m}$ is drawn by first sampling a subset $\mathbf{S}_0$ of $[n+1]$ of size m with $n+1 \in \mathbf{S}_0$ and an ordering $\boldsymbol{\pi}$ of $\mathbf{S}_0$ uniformly at random. We then call $\texttt{Preprocess}(f, \mathbf{S}_0, \boldsymbol{\pi}, \xi)$ to obtain $\mathbf{S}$. Notice that as $n+1$ is just a placeholder, we always have $n+1 \in \mathbf{S} \sim \mathcal{H}_{\xi,m}$.

As it will become clear later, our unateness tester will sample subsets according to the distribution $\mathcal{D}_{\xi,m}$ and use them to find an edge violation to unateness when f is far from unate. While this section is mainly concerned about $\mathcal{H}_{\xi,m}$, it will only be used in the analysis to help us understand how good those samples from $\mathcal{D}_{\xi,m}$ are in terms of revealing an edge violation to unateness.

Let $\Lambda = \lceil 2\log(n/\varepsilon) \rceil$ in the rest of the chapter. Given $i \in [n]$ and $m : 1 \leq m \leq n-1$ we use $\mathcal{P}_{i,m}$ to denote the uniform distribution over all size-m subsets of $[n] \setminus \{i\}$.

Next we use $\mathcal{H}_{\xi,m}$ and $\mathcal{P}_{i,m}$ to define strong edges.

Definition 59 (Strong edges)**.** *Let e be a bichromatic edge of f along variable $i \in [n]$. We say e is* ℓ-strong*, for some integer $\ell \in [\Lambda]$, if the following two conditions hold:*

1. *For every* $m \leq n^{2/3}$ *as a power of* 2 *and every* $\xi = 1/2^k$ *with* $\ell \leq k \leq \Lambda$*, the edge* e *is* $\mathbf{S}$*-persistent* (*recall Definition 57*) *with probability at least* $1 - (1/\log n)$ *when* $\mathbf{S} \sim \mathcal{H}_{\xi,m}$.
2. *The edge* e *is* $\mathbf{S}$*-persistent with probability at least* $1 - (1/\log n)$ *when* $\mathbf{S} \sim \mathcal{P}_{i,\lceil\sqrt{n}/2^\ell\rceil}$.

For each $i \in [n]$ and $\ell \in [\Lambda]$, we define

$$\textsc{Score}^+_{i,\ell}(f) = \frac{1}{2^n} \cdot \text{number of } \ell\text{-strong monotone edges along variable } i.$$

We analogously define $\textsc{Score}^-_{i,\ell}(f)$ for anti-monotone edges along variable i. Finally we define

$$\textsc{Score}^+_i(f) = \max_{\ell \in [\Lambda]} \left\{ \textsc{Score}^+_{i,\ell}(f) \cdot \frac{1}{2^\ell} \right\}, \tag{7.5}$$

and we analogously define $\textsc{Score}^-_i(f)$.

7.3.2 The Scores Lemma

We state the Scores Lemma:

Lemma 7.3.1 (The Scores Lemma)**.** *Let* $f \colon \{0,1\}^n \to \{0,1\}$ *be a Boolean function that is* ε*-far from unate with total influence* $\mathbf{I}_f < 6\sqrt{n}$*. Then we have*

$$\sum_{i \in [n]} \min \left\{ \textsc{Score}^+_i(f), \textsc{Score}^-_i(f) \right\} \geq \Omega\left(\frac{\varepsilon^2}{\Lambda^8} \right).$$

We note that our Scores Lemma above looks very similar to Lemma 4.3 from [52]. Thus the proof follows a similar trajectory. The main difference is that we are varying the distributions from which the set $\mathbf{S}$ of variables is drawn. Compared to [52] we not only consider the quality of $\mathbf{S}$ drawn from the $\mathcal{P}$ distribution in the definition of strong edges but also those drawn from $\mathcal{H}_{\xi,m}$ with a number of possible combinations of ξ and m in the indicated range. This makes the proof of the lemma slightly more involved than that of Lemma 4.3 in [52].

We prove Lemma 7.3.1 by proving the following simpler version, which avoids the minimum.

Lemma 7.3.2. *Assume that f is ε-far from monotone and satisfies $\mathbf{I}_f < 6\sqrt{n}$. Then we have*

$$\sum_{i\in[n]} \textsc{Score}_i^-(f) \geq \Omega\left(\frac{\varepsilon^2}{\Lambda^8}\right).$$

Proof of Lemma 7.3.1 assuming Lemma 7.3.2. Let $f\colon \{0,1\}^n \to \{0,1\}$ be a Boolean function which is ε-far from unate. We let $a \in \{0,1\}^n$ be defined by setting, for each $i \in [n]$,

$$a_i = \begin{cases} 0 & \text{if } \textsc{Score}_i^+(f) \geq \textsc{Score}_i^-(f) \\ 1 & \text{otherwise} \end{cases}.$$

Consider the function $g\colon \{0,1\}^n \to \{0,1\}$ defined using f by $g(x) = f(x \oplus a)$. We note that $\mathbf{I}_f = \mathbf{I}_g$ and g is ε-far from unate and thus, ε-far from monotone. So Lemma 7.3.2 implies that

$$\sum_{i\in[n]} \textsc{Score}_i^-(g) \geq \Omega\left(\frac{\varepsilon^2}{\Lambda^8}\right). \tag{7.6}$$

Finally, we claim that out choice of s implies that

$$\min\left\{\textsc{Score}_i^+(f), \textsc{Score}_i^-(f)\right\} = \textsc{Score}_i^-(g). \tag{7.7}$$

This can be observed by checking that (1) the distributions $\mathcal{H}_{\xi,m}$ defined using f and g are exactly the same; and (2) a point $x \in \{0,1\}^n$ is S-persistent for some $S \subseteq [n+1]$ in f if and only if $x \oplus a$ is S-persistent in g. As a result, an edge $(x, x^{(i)})$ is ℓ-strong in f if and only if $(x \oplus a, x^{(i)} \oplus a)$ is ℓ-strong in g but of course whether they are monotone or anti-monotone may change depending on a_i. (7.7) follows from these observations and Lemma 7.3.1 follows from (7.6) and (7.7). □

Before proving Lemma 7.3.2, we need a definition and a key technical lemma from [87].

Definition 60. *Given a Boolean function $f : \{0,1\}^n \to \{0,1\}$, we write G_f^- to denote its bipartite graph of anti-monotone edges:*

1. *Vertices on the LHS of G_f^- correspond to points $x \in \{0,1\}^n$ with $f(x) = 1$ and vertices on the RHS correspond to points $y \in \{0,1\}^n$ with $f(y) = 0$;*
2. *(x, y) is an edge in G_f^- if and only if (x, y) is an anti-monotone edge in f.*

Let $G = (U, V, E)$ be a subgraph of G_f, where U is a set of points x with $f(x) = 1$, V is a set of points y with $f(y) = 0$, and E consists of all anti-monotone edges between U and V (i.e., G is the induced subgraph of G_f^- on (U, V)). We say that G is right-d-good *for some positive integer d if the degree of every $y \in V$ lies in $[d : 2d]$ and the degree of every $x \in U$ is at most $2d$; We say that G is* left-d-good *if the degree of every $x \in U$ lies in $[d : 2d]$ and the degree of every $y \in V$ is at most $2d$.*

Lemma 7.3.3 (Lemma 7.1 in [87])**.** *If $f : \{0,1\}^n \to \{0,1\}$ is ε-far from monotone, G_f^- contains a bipartite subgraph $G = (U, V, E)$ induced on (U, V) that satisfies one of the following conditions:*

1. *G is left-d-good for some positive integer d and $\sigma = |U|/2^n$ satisfies*

$$\sigma^2 d = \Theta\left(\frac{\varepsilon^2}{\log^4 n}\right). \tag{7.8}$$

2. *G is right-d-good for some positive integer d and $\sigma = |V|/2^n$ satisfies (7.8).*

We note that each vertex in G_f^- has degree at most n. As a result, the two parameters d and σ in Lemma 7.3.3 always satisfy that $1 \le d \le n$ and

$$1 \ge \sigma \ge \Omega\left(\frac{\varepsilon}{\sqrt{n}\log^2 n}\right). \tag{7.9}$$

Proof of Lemma 7.3.2. Let $f : \{0,1\}^n \to \{0,1\}$ be a function that is ε-far from monotone with total variance $\mathbf{I}_f \le 6\sqrt{n}$. It follows from Lemma 7.3.3 that there is a subgraph $G = (U, V, E)$ of G_f^- that satisfies one of the two conditions in Lemma 7.3.3. Below we assume without loss of generality that G is left-d-good and $\sigma = |U|/2^n$ satisfies (7.8); the proof for G being right-d-good is symmetric.

In the rest of the proof we set ℓ to be the positive integer such that

$$\frac{1}{2^\ell} < \frac{\sigma}{\Lambda^4} \leq \frac{1}{2^{\ell-1}}.$$

So $\ell \in [\Lambda]$ using (7.9). Our goal is to show that at least half of edges in G are ℓ-strong. As a result,

$$\sum_{i\in[n]} \textsc{Score}_i^-(f) \geq \sum_{i\in[n]} \textsc{Score}_{i,\ell}^-(f)\cdot\frac{1}{2^\ell} \geq \frac{\Omega(|E|)}{2^n}\cdot\frac{1}{2^\ell} = \Omega(\sigma d)\cdot\frac{1}{2^\ell} = \Omega\left(\sigma d\cdot\frac{\sigma}{\Lambda^4}\right) = \Omega\left(\frac{\varepsilon^2}{\Lambda^8}\right).$$

The fact that at least half of edges in G are ℓ-strong follows directly from the next two claims:

Claim 7.3.4. *At least $(1-o(1))$-fraction of edges in G satisfy the first condition of being ℓ-strong.*

Claim 7.3.5. *At least $(1-o(1))$-fraction of edges in G satisfy the second condition of being ℓ-strong.*

The proof of Claim 7.3.5 follows from the arguments in Section 6.2 in [52]. Specifically, given the definition of *robust sets* for a bichromatic edge e of a certain size in Definition 6.4 of [52], Claim 7.3.5 is equivalent to applying Lemma 6.11 and Lemma 6.12 twice.

We prove Claim 7.3.4 in the rest of the proof. To this end, let $m \leq n^{2/3}$ and $\xi \leq 1/2^\ell$ such that both m and $1/\xi$ are powers of 2. We consider the quantity α as the fraction of $e \in E$ such that e is not $\mathbf{S}$-persistent with probability at least $1/\log n$ when $\mathbf{S} \sim \mathcal{H}_{\xi,m}$. Using α we have

$$\Pr_{\boldsymbol{e},\mathbf{S}}\Big[\boldsymbol{e} \text{ is } \mathbf{S}\text{-persistent}\Big] \leq (1-\alpha) + \alpha\left(1-\frac{1}{\log n}\right) = 1-\alpha/\log n,$$

where $\boldsymbol{e}$ is drawn uniformly from E and $\mathbf{S} \sim \mathcal{H}_{\xi,m}$. On the other hand, we consider the probability

$$\Pr_{\boldsymbol{e},\mathbf{S}}\Big[\boldsymbol{e} \text{ is not } \mathbf{S}\text{-persistent}\Big].$$

By Lemma 7.2.1, as well as the fact that each vertex of G is incident to at most $2d$ edges in E, for at least $(1 - \exp(-\Omega(\log^2 n)))$-fraction of $\mathbf{S} \sim \mathcal{H}_{\xi,m}$, there are at most $2\xi d \cdot 2^n$ many edges which are not $\mathbf{S}$-persistent out of a total of at least $\sigma d \cdot 2^n$ edges in E. Therefore, we have

$$\Pr_{\boldsymbol{e},\mathbf{S}}\left[\boldsymbol{e} \text{ is not } \mathbf{S}\text{-persistent}\right] \leq \exp\left(-\Omega(\log^2 n)\right) + \left(1 - \exp\left(-\Omega(\log^2 n)\right)\right) \cdot O\left(\xi/\sigma\right),$$

which is $O(1/\Lambda^4)$. Combining these inequalities we have that $\alpha = O(\log n/\Lambda^4)$. Claim 7.3.4 follows from a union bound over $O(\log n) \cdot \Lambda \leq O(\Lambda^2)$ many choices of the two parameters m and ξ. □

7.3.3 Bucketing scores

We will now use standard grouping techniques to make Lemma 7.3.1 easier to use.

From (7.5), we say that $i \in [n]$ is of *type*-(s,t) for some $s, t \in [\Lambda]$ if

$$\textsc{Score}_i^+ = \textsc{Score}_{i,s}^+ \cdot \frac{1}{2^s} \quad \text{and} \quad \textsc{Score}_i^- = \textsc{Score}_{i,t}^- \cdot \frac{1}{2^t}.$$

From Lemma 7.3.1, there exist $s, t \in [\Lambda]$ such that

$$\sum_{\substack{i \in [n] \\ \text{type-}(s,t)}} \min\left\{\textsc{Score}_i^+, \textsc{Score}_i^-\right\} \geq \Omega\left(\frac{\varepsilon^2}{\Lambda^{10}}\right). \tag{7.10}$$

Furthermore, we say a variable $i \in [n]$ has *weight* k for some positive integer k if

$$\frac{1}{2^k} < \min\left\{\textsc{Score}_i^+, \textsc{Score}_i^-\right\} \leq \frac{1}{2^{k-1}}.$$

Therefore, we have

$$\sum_{\substack{i\in[n]\\ \text{type-}(s,t)}} \min\left\{\textsc{Score}_i^+, \textsc{Score}_i^-\right\} = \sum_{k\geq 1} \sum_{\substack{i\in[n]\\ \text{type-}(s,t)\\ \text{weight } k}} \min\left\{\textsc{Score}_i^+, \textsc{Score}_i^-\right\}$$
$$\leq \sum_{k\in[3\Lambda]} \sum_{\substack{i\in[n]\\ \text{type-}(s,t)\\ \text{weight } k}} \min\left\{\textsc{Score}_i^+, \textsc{Score}_i^-\right\} + n\cdot\left(\frac{\varepsilon}{n}\right)^3,$$

which implies by (7.10) that there exists some $h \in [3\Lambda]$ such that

$$\sum_{\substack{i\in[n]\\ \text{type-}(s,t)\\ \text{weight } h}} \min\left\{\textsc{Score}_i^+, \textsc{Score}_i^-\right\} \geq \Omega\left(\frac{\varepsilon^2}{\Lambda^{11}}\right). \tag{7.11}$$

We let

$$\mathcal{I}^* = \Big\{i \in [n] : i \text{ is of type-}(s,t) \text{ and weight } h\Big\},$$

and let $\mathcal{I}$ be a subset of $\mathcal{I}^*$ such that $|\mathcal{I}|$ is the largest power of 2 that is not larger than $|\mathcal{I}^*|$.

We summarize the above discussion in the following lemma.

Lemma 7.3.6. *Let $f\colon \{0,1\}^n \to \{0,1\}$ be ε-far from unate with $\mathbf{I}_f < 6\sqrt{n}$. Then there are $s,t \in [\Lambda]$, $h \in [3\Lambda]$ and a set $\mathcal{I} \subseteq [n]$ such that $|\mathcal{I}|$ is a power of 2, $|\mathcal{I}|/2^h = \Omega(\varepsilon^2/\Lambda^{11})$ and every $i \in \mathcal{I}$ has*

$$\min\left\{\textsc{Score}_{i,s}^+ \cdot \frac{1}{2^s}, \textsc{Score}_{i,t}^- \cdot \frac{1}{2^t}\right\} \geq \frac{1}{2^h}. \tag{7.12}$$

7.4 The Main Algorithm

We now describe the main algorithm for testing unateness. The algorithm rejects a function f only when an edge violation has been found. As a result, for its correctness it suffices to show that when the input function f is ε-far from unate, the algorithm finds an edge violation with probability at least $2/3$. For convenience we will suppress $\text{polylog}(n/\varepsilon)$ factors using $\tilde{O}(\cdot)$ in the rest of analysis.

The main algorithm has four cases. Case 0 is when the input function f satisfies $\mathbf{I}_f > 6\sqrt{n}$. In this case an $\tilde{O}(\sqrt{n})$-query algorithm is known [23] (also see Lemma 2.1 of [52]).

From now on, we assume that f is not only ε-far from unate but also satisfies $\mathbf{I}_f \leq 6\sqrt{n}$. Then there are parameters $s, t \in [\Lambda]$ and $h \in [3\Lambda]$ and a set $\mathcal{I} \subseteq [n]$ with which Lemma 7.3.6 holds for f. We may assume that the algorithm knows s, t, h and $|\mathcal{I}| = 2^\ell$ (by trying all possibilities, which just incurs an addition factor of $O(\Lambda^4)$ in the query complexity). We may further assume without loss of generality that $s \geq t$ since the case of $s < t$ is symmetric.

We consider the following three cases of f:

Case 1: $|\mathcal{I}|/2^t \geq n^{2/3}$ and and at least half of $i \in \mathcal{I}$ satisfy

$$\mathbf{Inf}_f[i] \leq \left(\frac{\varepsilon^2}{\Lambda^{13}}\right) \cdot \frac{n^{1/3}}{|\mathcal{I}|}\,; \tag{7.13}$$

Case 2: $|\mathcal{I}|/2^t \geq n^{2/3}$ and and at least half of $i \in \mathcal{I}$ violate (7.13); and

Case 3: $|\mathcal{I}|/2^t \leq n^{2/3}$.

We prove the following two lemmas in Section 7.5 and 7.7 which cover the first two cases.

Lemma 7.4.1. *Let $s \geq t \in [\Lambda]$, $h \in [3\Lambda]$, and $\ell \in [\lfloor \log n \rfloor]$ with $2^\ell/2^t \geq n^{2/3}$. There is a $\tilde{O}(n^{2/3}/\varepsilon^2)$-query algorithm with the following property. Given any Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ that satisfies (i) Lemma 7.3.6 holds for f with s, t, h and a set $\mathcal{I} \subseteq [n]$ with $|\mathcal{I}| = 2^\ell$; and (ii) at least half of $i \in \mathcal{I}$ satisfy (7.13), the algorithm finds an edge violation to unateness with probability at least $2/3$.*

Lemma 7.4.2. *Let $s \geq t \in [\Lambda]$, $h \in [3\Lambda]$, and $\ell \in [\lfloor \log n \rfloor]$ with $2^\ell/2^t \geq n^{2/3}$. There is an algorithm that makes $\tilde{O}(n^{2/3}/\varepsilon^2)$ queries and satisfies the following property. Given any Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ that satisfies (i) Lemma 7.3.6 holds for f with s, t, h and a set $\mathcal{I} \subseteq [n]$ with $|\mathcal{I}| = 2^\ell$ and (ii) at least half of $i \in \mathcal{I}$ violate (7.13), the algorithm finds an edge violation to unateness with probability at least $2/3$.*

Case 3 can be handled using an algorithm presented in [52]. We include its description and the proof of the following lemma in Section 7.8 for completeness.

Lemma 7.4.3. *Let $s \geq t \in [\Lambda]$, $h \in [3\Lambda]$, and $\ell \in [\lfloor \log n \rfloor]$ with $2^\ell/2^t \leq n^{2/3}$. There is an algorithm that makes $\tilde{O}(n^{2/3} + \sqrt{n}/\varepsilon^2)$ queries and satisfies the following property. Given any Boolean function $f\colon \{0,1\}^n \to \{0,1\}$ that satisfies Lemma 7.3.6 with s, t, h and a set $\mathcal{I} \subseteq [n]$ of size $|\mathcal{I}| = 2^\ell$, the algorithm finds an edge violation of f to unateness with probability at least $2/3$.*

Theorem 65 follows by combining all these lemmas.

7.5 The Algorithm for Case 1

Let $s \geq t \in [\Lambda]$, $h \in [3\Lambda]$ and $\ell \in [\lfloor \log n \rfloor]$. In Case 1 the input function $f\colon \{0,1\}^n \to \{0,1\}$ satisfies Lemma 7.3.6 with parameters s, t, h and $\mathcal{I} \subseteq [n]$ of size $|\mathcal{I}| = 2^\ell$, with $|\mathcal{I}|/2^t \geq n^{2/3}$. At least half of the variables $i \in \mathcal{I}$ have low influence as given in (7.13). Let i be such a variable. Then by (7.13),

$$\left(\frac{\varepsilon^2}{\Lambda^{13}}\right) \cdot \frac{n^{1/3}}{|\mathcal{I}|} > \mathbf{Inf}_f[i] \geq 2 \cdot \textsc{Score}^+_{i,s} \geq \frac{2^s}{2^h}.$$

Letting $\xi = 1/2^s$ throughout this section, it follows from Lemma 7.3.6 that

$$\xi = \Omega\left(\frac{\Lambda^{13}}{\varepsilon^2} \cdot \frac{|\mathcal{I}|}{2^h n^{1/3}}\right) = \Omega\left(\frac{\Lambda^2}{n^{1/3}}\right). \tag{7.14}$$

7.5.1 Informative sets

We start with the notion of *informative sets*. Note that we will have different notions of informative sets in different cases of the algorithm. We use the same name because they serve similar purposes.

Given $i \in [n]$ and a set $S \subseteq [n+1]$ we use $\text{PE}^+_i(S)$ to denote the set of s-strong monotone edges along variable i that are S-persistent. We define $\text{PE}^-_i(S)$ similarly for antimonotone edges.

Definition 61 (Informative Sets)**.** *A set $S \subseteq [n+1]$ is* i-informative *for monotone edges if*

$$\frac{|\text{PE}^+_i(S)|}{2^n} \geq \frac{\textsc{Score}^+_{i,s}}{4} \geq \frac{2^{s-h}}{4} \tag{7.15}$$

and that $S \subseteq [n+1]$ *is* i-informative *for anti-monotone edges if*

$$\frac{|\mathrm{PE}_i^-(S)|}{2^n} \geq \frac{\mathrm{SCORE}_{i,t}^-}{4} \geq \frac{2^{t-h}}{4}. \tag{7.16}$$

We simply say that $S \subseteq [n+1]$ *is* i-informative *if* S *satisfies both (7.15) and (7.16).*

Lemma 7.5.1. *For each* $i \in \mathcal{I}$ *and each positive integer* $m \leq n^{2/3}$ *that is a power of* 2, $\mathbf{S} \sim \mathcal{H}_{\xi,m}$ *is* i*-informative with probability at least* $1 - o(1)$.

Proof. We first show that $\mathbf{S} \sim \mathcal{H}_{\xi,m}$ satisfies (7.15) with probability at least $1 - o(1)$. The same argument works to show $\mathbf{S} \sim \mathcal{H}_{\xi,m}$ satisfies (7.16). The lemma then follows from a union bound. To this end, let α be the probability of $\mathbf{S} \sim \mathcal{H}_{\xi,m}$ being i-informative for monotone edges. We examine

$$\Pr_{\boldsymbol{e},\mathbf{S}}\Big[\boldsymbol{e} \text{ is } \mathbf{S}\text{-persistent}\Big],$$

where $\boldsymbol{e}$ is an s-strong monotone edge along variable i drawn uniformly at random and $\mathbf{S} \sim \mathcal{H}_{\xi,m}$. It follows from the definition of strong edges that the probability is at least $1 - 1/\log n$. On the other hand, we can also upperbound the probability using α (and the definition of i-informative sets) as $(1-\alpha)/4 + \alpha$. Solving the inequality we get $\alpha \geq 1 - o(1)$. □

Next we introduce two new families of distributions that will help us connect $\mathcal{H}_{\xi,m}$ with $\mathcal{D}_{\xi,m}$.

Definition 62. *Given* $\xi \in (0,1)$, $m\colon 1 \leq m \leq n$ *and* $i \in [n]$, *we let* $\mathcal{D}_{\xi,m,i}$ *denote the following distribution supported on subsets of* $[n]$*:* $\mathbf{S} \sim \mathcal{H}_{\xi,m,i}$ *is drawn by first sampling a subset* $\mathbf{S}_0$ *of* $[n]$ *of size* m *with* $i \in \mathbf{S}_0$ *and an ordering* $\boldsymbol{\pi}$ *of* $\mathbf{S}_0$ *uniformly at random. We then call* $\texttt{Preprocess}(f, \mathbf{S}_0, \boldsymbol{\pi}, \xi)$ *and set* $\mathbf{S}$ *to be its output.*

Similarly, $\mathcal{H}_{\xi,m,i}$ *denotes the following distribution supported on subsets of* $[n+1]$*:* $\mathbf{S} \sim \mathcal{H}_{\xi,m,i}$ *is drawn by first sampling a subset* $\mathbf{S}_0$ *of* $[n+1] \setminus \{i\}$ *of size* m *with* $n+1 \in \mathbf{S}_0$ *and an ordering* $\boldsymbol{\pi}$ *of* $\mathbf{S}_0$ *uniformly at random. We then call* $\texttt{Preprocess}(f, \mathbf{S}_0, \boldsymbol{\pi}, \xi)$ *and set* $\mathbf{S}$ *to be its output.*

Using the fact that the total variation distance between the $\mathbf{S}_0$ used in $\mathcal{H}_{\xi,m}$ (at the beginning of the process) and the $\mathbf{S}_0$ used in $\mathcal{H}_{\xi,m,i}$ is at most m/n, we have

$$d_{\mathrm{TV}}\Big(\mathcal{H}_{\xi,m}, \mathcal{H}_{\xi,m,i}\Big) \leq m/n$$

and the following corollary from Lemma 7.5.1.

Corollary 7.5.2. *For every $i \in \mathcal{I}$ and every positive integer $m \leq n^{2/3}$ as a power of 2, we have that $\mathbf{S} \sim \mathcal{H}_{\xi,m,i}$ is i-informative with probability at least $1 - o(1)$.*

The next two lemmas allow us to draw random subsets and still obtain i-informative sets. They enable us to use techniques from [52] for particular cases of our algorithm.

Lemma 7.5.3. *For every $i \in \mathcal{I}$ we have $\mathbf{T} \sim \mathcal{P}_{i,\lceil\sqrt{n}/2^t\rceil}$ is i-informative for anti-monotone edges with probability at least $1 - o(1)$.*

Proof. Similarly to the proof of Lemma 7.5.1, we write α to denote the probability of $\mathbf{T} \sim \mathcal{P}_{i,\lceil\sqrt{n}/2^t\rceil}$ being i-informative for anti-monotone edges. We examine

$$\mathop{\mathbf{Pr}}_{\boldsymbol{e},\mathbf{T}}\Big[\boldsymbol{e} \text{ is } \mathbf{T}\text{-persistent}\Big],$$

where $\boldsymbol{e}$ is a t-strong anti-monotone edge along variable i drawn uniformly and $\mathbf{T} \sim \mathcal{P}_{i,\lceil\sqrt{n}/2^t\rceil}$. It follows from the definition of strong edges that this probability is at least $1 - 1/\log n$. On the other hand, we can also upperbound the probability using α (and the definition of i-informative sets for anti-monotone edges) as $(1-\alpha)/4 + \alpha$. Solving the inequality we get $\alpha \geq 1 - o(1)$. □

Similarly, we may conclude the analogous lemma for monotone edges, whose proof follows similarly to Lemma 7.5.3.

Lemma 7.5.4. *For every $i \in \mathcal{I}$ we have that $\mathbf{S} \sim \mathcal{P}_{i,\lceil\sqrt{n}/2^s\rceil}$ is i-informative for monotone edges with probability at least $1 - o(1)$.*

7.5.2 Catching variables: Relating $\mathcal{D}_{\xi,m}$ and $\mathcal{H}_{\xi,m}$

Now we focus on the variables in $\mathcal{I}$ that satisfy (7.13). To this end, we let $\mathcal{I}^*$ be a subset of $\mathcal{I}$ of size $\lceil |\mathcal{I}|/2 \rceil$ such that all variables in $\mathcal{I}^*$ satisfy (7.13). Given that the algorithm knows the size of $\mathcal{I}$, it also knows the size of $\mathcal{I}^*$ (though not variables within). Next we use m to denote the largest power of 2 that is at most $\xi|\mathcal{I}|/n^{1/3}$. In other words, m is the unique power of 2 satisfying

$$\frac{\xi|\mathcal{I}|}{2n^{1/3}} < m \leq \frac{\xi|\mathcal{I}|}{n^{1/3}}. \tag{7.17}$$

Given that $|\mathcal{I}| \geq |\mathcal{I}|/2^t \geq n^{2/3}$ and (7.14), we have $m \boldsymbol{g} 1$ and $m = \Theta(\xi|\mathcal{I}|/n^{1/3})$.

We now turn to analyzing the distribution $\mathcal{D}_{\xi,m}$ with the m defined above.

Definition 63 (Catching Variables)**.** *Let $i \in \mathcal{I}^*$. We say that a set $S \subseteq [n]$* catches *the variable i if $i \in S$ and $\texttt{Sub}(S,i) = (S \cup \{n+1\}) \setminus \{i\}$ is i-informative (see Definition 61). We let*

$$\textsc{Caught}(S) = \Big\{ i \in \mathcal{I}^* : S \textit{ catches } i \Big\}.$$

Intuitively, if we sample $\mathbf{S} \sim \mathcal{D}_{\xi,m}$ and $i \in \textsc{Caught}(\mathbf{S})$, then we have an upper bound for how many samples $\boldsymbol{x}$ we need for AE-$\textsc{Search}(f, \boldsymbol{x}, \mathbf{S} \cup \{n+1\})$ to reveal a bichromatic edge along i.

Claim 7.5.5. *For every $i \in \mathcal{I}^*$, we have*

$$\Pr_{\mathbf{S} \sim \mathcal{D}_{\xi,m,i}} \Big[\mathbf{S} \text{ catches } i \Big] \geq \Pr_{\mathbf{T} \sim \mathcal{H}_{\xi,m,i}} \Big[\mathbf{T} \text{ is } i\text{-informative } \Big] - o(1).$$

Proof. We show the total variation distance between $\mathbf{S} \sim \mathcal{D}_{\xi,m,i}$ and $\texttt{Sub}(\mathbf{T}, i)$ over $\mathbf{T} \sim \mathcal{H}_{\xi,m,i}$ is

$$O\left(\frac{m \log^8 n}{\xi} \cdot \mathbf{Inf}_f[i] \right) = O\left(\frac{|\mathcal{I}| \log^8 n}{n^{1/3}} \cdot \mathbf{Inf}_f[i] \right) = o(1), \tag{7.18}$$

given (7.13) and $i \in \mathcal{I}^*$. The lemma follows from the observations that $\texttt{Sub}(\mathbf{T}, i)$ contains i and when $\mathbf{T}$ is i-informative, $\texttt{Sub}(\mathbf{T}, i)$ catches i.

To upperbound the total variation distance between $\mathbf{S} \sim \mathcal{D}_{\xi,m,i}$ and $\texttt{Sub}(\mathbf{T}, i)$ over $\mathbf{T} \sim \mathcal{H}_{\xi,m,i}$, we use the following coupling. First we draw a subset $\mathbf{S}_0$ of $[n]$ with $i \in \mathbf{S}_0$

Procedure `AlgorithmCase1.1`(f)

Input: Query access to a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$
Output: Either "unate," or two edges constituting an edge violation of f to unateness.

1. Repeat the following $O(1)$ times:
2. Draw $\mathbf{S} \sim \mathcal{D}_{\xi,m}$: First draw a size-$m$ subset $\mathbf{S}_0$ of $[n]$ and an ordering $\boldsymbol{\pi}$ of $\mathbf{S}_0$ uniformly at random and then call `Preprocess`$(f, \mathbf{S}_0, \boldsymbol{\pi}, \xi)$.
3. Repeat q times, where $q = O\left(n^{2/3}\Lambda^{13}/\varepsilon^2\right)$:
4. Draw an $\boldsymbol{x} \in \{0,1\}^n$ uniformly and run AE-SEARCH$(f, \boldsymbol{x}, \mathbf{S} \cup \{n+1\})$
5. Let $\mathbf{A}$ be the set of $i \in [n]$ such that an anti-monotone edge along i is found
6. Repeat q times:
7. Draw an $\boldsymbol{y} \in \{0,1\}^n$ uniformly and run AE-SEARCH$(f, \boldsymbol{y}, \mathbf{S} \cup \{n+1\})$
8. Let $\mathbf{B}$ be the set of $i \in [n]$ such that a monotone edge along variable i is found
9. If $\mathbf{A} \cap \mathbf{B} \neq \emptyset$, output an edge violation of f to unateness.
10. Output "unate."

Figure 7.6: Algorithm for Case 1.1

and an ordering $\boldsymbol{\pi}$ of $\mathbf{S}_0$ uniformly at random. Then we set $\mathbf{S}_0' = \texttt{Sub}(\mathbf{S}_0, i)$ and $\boldsymbol{\pi}'$ to be the ordering of $\mathbf{S}_0'$ obtained from $\boldsymbol{\pi}$ by replacing i with $n+1$. Finally we draw the output from the best coupling for $\texttt{Preprocess}(f, \mathbf{S}_0, \boldsymbol{\pi}, \xi)$ and $\texttt{Sub}(\texttt{Preprocess}(f, \mathbf{S}_0', \boldsymbol{\pi}', \xi), i)$. The upper bound in (7.18) follows directly from Lemma 7.2.2. □

7.5.3 Algorithm for Case 1.1

There are two sub-cases in Case 1. Specifically, for the remainder of Section 7.5.3, we assume that

$$m \geq \frac{n^{1/3}\log^2 n}{2^t}, \tag{7.19}$$

and handle the other case in Case 1.2. We let

$$r := \frac{m|\mathcal{I}^*|}{n} = \Omega(\log^2 n), \tag{7.20}$$

the expected size of the intersection of a random size-m subset of $[n]$ with $\mathcal{I}^*$, where we used (7.19) and $|\mathcal{I}^*|/2^t = \Omega(n^{2/3})$. Note that both m and r are known to the algorithm. We prove Lemma 7.4.1 assuming (7.19) using `AlgorithmCase1.1` in Figure 7.6, with the following query complexity.

Claim 7.5.6. *The query complexity of* `AlgorithmCase1.1` *is* $\tilde{O}(n^{2/3}/\varepsilon^2)$.

Proof. By Lemma 7.2.1, line 2 of `AlgorithmCase1.1` requires $\tilde{O}(m/\xi)$ queries. By (7.17),

$$\frac{m}{\xi} = O\left(\frac{\xi|\mathcal{I}|}{\xi \cdot n^{1/3}}\right) = O\left(\frac{|\mathcal{I}|}{n^{1/3}}\right) = O(n^{2/3})$$

using the trivial bound of $|\mathcal{I}| \le n$. The claim then follows from our choice of q in the algorithm. □

The algorithm for Case 1.1 starts by sampling a set $\mathbf{S} \sim \mathcal{D}_{\xi,m}$. It then keeps drawing points $\boldsymbol{x}$ uniformly at random to run AE-SEARCH$(f, \boldsymbol{x}, \mathbf{S} \cup \{n+1\})$ to find bichromatic edges, with the hope to find an edge violation along one of the variables in $\mathcal{I}^*$. We break lines 3–8 into the search of anti-monotone edges and the search of monotone edges separately only for the analysis later; algorithm wise there is really no need to do so. (The reason why we use $\mathbf{S} \cup \{n+1\}$ instead of $\mathbf{S}$ in the algorithm will become clear in the proof of Lemma 7.5.7; roughly speaking, we need it to establish a connection between $\mathcal{D}_{\xi,m}$ and $\mathcal{H}_{\xi,m}$ so that we can carry the analysis on $\mathcal{H}_{\xi,m}$ that has been done so far over to $\mathcal{D}_{\xi,m}$.) On the one hand, recall from Lemma 7.1.2 that if $i \in S \subseteq [n]$ and a bichromatic edge e along variable i is $\texttt{Sub}(S,i) = (S \cup \{n+1\}) \setminus \{i\}$-persistent, then running AE-SEARCH on $S \cup \{n+1\}$ and any of the two points of e would reveal e with high probability. On the other hand, if a set (e.g., $\texttt{Sub}(S,i)$) is i-informative then it is persistent on a large fraction of edges along variable i.

Our first goal is to prove Lemma 7.5.7, which states that $\mathbf{S} \sim \mathcal{D}_{\xi,m}$ catches many variables.

Lemma 7.5.7. *We have* $|\textsc{Caught}(\mathbf{S})| \geq r/6$ *with probability* $\Omega(1)$.

Proof. Let α be the probability we are interested in:

$$\alpha = \mathop{\mathbf{Pr}}_{\mathbf{S}\sim\mathcal{D}_{\xi,m}}\left[\left|\textsc{Caught}(\mathbf{S})\right| \geq \frac{r}{6}\right].$$

For each $i \in \mathcal{I}^*$, as the $\mathbf{S}_0$ drawn in $\mathcal{D}_{\xi,m}$ at the beginning contains i with probability m/n, we have

$$\begin{aligned}\mathop{\mathbf{Pr}}_{\mathbf{S}\sim\mathcal{D}_{\xi,m}}\left[\mathbf{S} \text{ catches } i\right] &\geq \frac{m}{n}\cdot\mathop{\mathbf{Pr}}_{\mathbf{S}\sim\mathcal{D}_{\xi,m,i}}\left[\mathbf{S} \text{ catches } i\right]\\ &\geq \frac{m}{n}\cdot\left(\mathop{\mathbf{Pr}}_{\mathbf{T}\sim\mathcal{H}_{\xi,m,i}}\left[\mathbf{T} \text{ is } i\text{-informative}\right] - o(1)\right) \geq (1-o(1))\cdot\frac{m}{n}.\end{aligned}$$

Furthermore, since $\mathbf{S}\sim\mathcal{D}_{\xi,m}$ is a subset of $\mathbf{S}_0$ drawn at the beginning which is a random subset of $[n]$ of size m, we have by Lemma 7.11.1 that with probability at least $1-\exp(-\Omega(r))$ over the draw of $\mathbf{S}\sim\mathcal{D}_{\xi,m}$ that $|\mathbf{S}\cap\mathcal{I}^*| \leq 4r$. Therefore, we have

$$\begin{aligned}(1-o(1))r = (1-o(1))\cdot\frac{m|\mathcal{I}^*|}{n} &\leq \sum_{i\in\mathcal{I}^*}\mathop{\mathbf{Pr}}_{\mathbf{S}\sim\mathcal{D}_{\xi,m}}\left[\mathbf{S} \text{ catches } i\right] = \mathop{\mathbf{E}}_{\mathbf{S}\sim\mathcal{D}_{\xi,m}}\left[\left|\textsc{Caught}(\mathbf{S})\right|\right]\\ &\leq m\cdot\exp\left(-\Omega(\log^2 n)\right) + (1-\alpha)\cdot\frac{r}{6} + \alpha\cdot 4r.\end{aligned}$$

Solving for α gives the desired bound of $\alpha = \Omega(1)$. □

Given Lemma 7.5.7, a constant fraction of the intersection of $\mathbf{S}$ and $\mathcal{I}^*$ will be caught, and therefore, AE-$\textsc{Search}(f, \boldsymbol{x}, \mathbf{S}\cup\{n+1\})$ will output a bichromatic edge along a variable from these caught coordinates for sufficiently many points $\boldsymbol{x}$. In the rest of the proof, we fix S to be a set that catches at least $r/6$ many variables in $\mathcal{I}^*$, and prove in the rest of the proof that during this loop, an edge violation is found with probability $1-o(1)$.

Given S, we write $\mathcal{J}\subseteq\mathcal{I}^*$ to denote the set of variables caught by S with $|\mathcal{J}| \geq r/6$. Then by definition we have that $\mathcal{J}\subseteq S$ and $\texttt{Sub}(S, j)$ is j-informative for every $j\in\mathcal{J}$.

We start by showing that $\mathbf{A}\cap\mathcal{J}$ is large with high probability.

Lemma 7.5.8. *We have* $|\mathbf{A}\cap\mathcal{J}| \geq \Omega\left(m2^t/n^{1/3}\right)$ *with probability at least* $1-o(1)$.

Proof. For each $j \in \mathcal{J}$, we let X_j be the set of $x \in \{0,1\}^n$ such that AE-SEARCH$(f, x, S \cup \{n+1\})$ returns an anti-monotone edge along h with probability at least $2/3$. Then by definition we have

$$|X_j| = \Omega\left(\frac{2^t}{2^h}\right) \cdot 2^n,$$

and the X_j's are disjoint by Corollary 7.1.3 so we have $r2^t/2^h = O(1)$. To analyze $\mathbf{A} \cap \mathcal{J}$, we break the rounds on line 3 into $\lceil m2^t/n^{1/3} \rceil$ many phases, each consisting of

$$\left\lceil \frac{2^h}{r2^t} \right\rceil \cdot \log^2 n = O\left(\frac{2^h}{r2^t} \cdot \log^2 n\right)$$

many iterations of line 4 (using $r2^t/2^h = O(1)$). The q rounds we have are enough since

$$\left\lceil \frac{m2^t}{n^{1/3}} \right\rceil \cdot O\left(\frac{2^h}{r2^t} \cdot \log^2 n\right) = O\left(\frac{m2^h \log^2 n}{n^{1/3} r}\right) = O\left(n^{2/3}\Lambda^{13}/\varepsilon^2\right)$$

using $r = \Omega(m|\mathcal{I}|/n)$ and $|\mathcal{I}|/2^h = \Omega(\varepsilon^2/\Lambda^{11})$. Also note $r = \Omega(m2^t/n^{1/3})$ using $|\mathcal{I}| \geq n^{2/3}2^t$.

At the beginning of each phase, either $\Omega(r)$ anti-monotone edges along different variables in $\mathcal{J}$ have already been found, and we are done, or the number of variables in $\mathcal{J} \setminus \mathbf{A}$ at the moment is at least $\Omega(r)$. As a result, their union of X_j for such j is $\Omega(r2^t/2^h) \cdot 2^n$. Using the number of rounds in each phase, the probability of not finding any new anti-monotone edge along $\mathcal{J}$ during this phase is at most $1/\mathrm{poly}(n)$, and it remains negligible even after a union bound over the number of phases. The lemma follows from the fact that the number of phases is at least $\Omega(m2^t/n^{1/3})$. □

Fix an A such that $C = A \cap \mathcal{J}$ satisfies the lower bound of Lemma 7.5.8. We finally show that $C \cap \mathbf{B}$ with high probability. This finishes the proof of correctness in Case 1.

Lemma 7.5.9. *We have that $C \cap \mathbf{B}$ is not empty with probability at least $1 - o(1)$.*

Proof. For each $j \in C$, let Y_j denote the set of $y \in \{0,1\}^n$ such that AE-SEARCH$(f, y, S \cup \{n+1\})$ returns a monotone edge along variable j with probability at least $2/3$. Then we have

$$|Y_j| = \Omega\left(\frac{2^s}{2^h}\right) \cdot 2^n.$$

Procedure `AlgorithmCase1.2`(f)

Input: Query access to a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$
Output: Either "unate," or two edges constituting an edge violation of f to unateness.

1. Repeat the following $O(1)$ times:
2. Draw a set $\mathbf{T} \subset [n]$ of size p uniformly at random.
3. Repeat $O\left(n^{2/3}\Lambda^{11}\log^2 n \big/ \varepsilon^2\right)$ times:
4. Draw an $\boldsymbol{x} \in \{0,1\}^n$ uniformly and run AE-SEARCH$(f, \boldsymbol{x}, \mathbf{T})$
5. Let $\mathbf{A}$ be the set of $i \in [n]$ such that an anti-monotone edge along i is found.
6. Repeat $O\left(n^{1/3}\log^3 n \big/ (m2^t)\right)$ times:
7. Draw $\mathbf{S}$ by first drawing a subset $\mathbf{S}_0$ of $\mathbf{T}$ of size m and an ordering $\boldsymbol{\pi}$ of $\mathbf{S}_0$ both uniformly at random and then call `Preprocess`$(f, \mathbf{S}_0, \boldsymbol{\pi}, \xi)$.
8. Repeat $O\left((2^h/2^s)\cdot \log n\right)$ times:
9. Draw $\boldsymbol{y} \in \{0,1\}^n$ uniformly and run AE-SEARCH$(f, \boldsymbol{y}, \mathbf{S} \cup \{n+1\})$.
10. Let $\mathbf{B}$ be the set of $i \in [n]$ such that a monotone edge along variable i is found.
11. If $\mathbf{A} \cap \mathbf{B} \neq \emptyset$, output an edge violation of f to unateness.
12. Output "unate."

Figure 7.7: Algorithm for Case 1.2

Since they are disjoint, the fraction of points that Y_j, $j \in C$, cover is at least

$$\Omega\left(\frac{2^s}{2^h}\cdot\frac{m2^t}{n^{1/3}}\right) \geq \Omega\left(\frac{2^s}{2^h}\cdot\frac{\xi|\mathcal{I}|}{n^{1/3}}\cdot\frac{2^t}{n^{1/3}}\right) = \Omega\left(\frac{\varepsilon^2}{n^{2/3}\Lambda^{11}}\right)$$

using $s \geq t$ and $m2^t \geq n^{1/3}\log^2 n$. The lemma follows from our choice of q in the algorithm. □

7.5.4 Algorithm for Case 1.2

The second subcase of Case 1 occurs when

$$m < \frac{n^{1/3}\log^2 n}{2^t}, \tag{7.21}$$

and the crucial difference is that unlike Case 1.1, we cannot conclude with (7.20). More specificially, two potential issues which were not present in Section 7.5.3 may arise: 1) sampling a set $\mathbf{S} \sim \mathcal{D}_{\xi,m}$ may result in $\textsc{Caught}(\mathbf{S}) = \emptyset$, and 2) even if $|\textsc{Caught}(\mathbf{S})|$ is large, too few points $\boldsymbol{x}$ may result in AE-$\textsc{Search}\,(f, \boldsymbol{x}, \mathbf{S})$ returning an anti-monotone edge (for instance, when $2^t = 1$ and $2^h = n$). Thus, we address these two problems with `AlgorithmCase1.2`, which is described in Figure 7.7.

At a high level, `AlgorithmCase1.2` proceeds by sampling a set $\mathbf{T} \subseteq [n]$ of size

$$p := \lceil \sqrt{n}/2^t \rceil + 1 = \Theta(\sqrt{n}/2^t)$$

uniformly at random, and directly uses the set $\mathbf{T}$ to search for anti-monotone edges by repeating AE-$\textsc{Search}\,(f, \boldsymbol{x}, \mathbf{T})$ for $\tilde{O}(n^{2/3}/\varepsilon^2)$ many iterations. We show at the end the algorithm will obtain anti-monotone edges along at least $\Omega(n^{1/6})$ variables in $\mathbf{T} \cap \mathcal{I}$ (as an anti-monotone edge is found every $\tilde{O}(\sqrt{n}/\varepsilon^2)$ iterations of AE-$\textsc{Search}\,(f, \boldsymbol{x}, \mathbf{S})$). The algorithm will then sample $\mathbf{S}_0 \subset \mathbf{T}$ of size m (notice that $m \ll p$ by (7.21)), pass it through `Preprocess` to obtain $\mathbf{S} \subseteq \mathbf{S}_0$, and then repeat AE-$\textsc{Search}\,(f, \boldsymbol{y}, \mathbf{S}_0)$ in hopes of observing an edge violation. In order to do so, $\mathbf{S}$ must contain a variable where the algorithm has already observed an anti-monotone edge. Since the number of variables with anti-monotone edges observed may be $O(n^{1/6})$ and $m \cdot n^{1/6}$ could be much smaller than p, the algorithm needs to sample the subset $\mathbf{S}_0$ from $\mathbf{T}$ multiple times.

We state the query complexity of the algorithm for Case 1.2, where we note that the upper bound will follow from (7.21), (7.14), (7.17), the fact that $2^t \geq 1$ and $2^h/\mathcal{I} \geq \tilde{\Omega}(\varepsilon^2)$.

Lemma 7.5.10. *The query complexity of* `AlgorithmCase1.2` *is (using (7.14))*

$$\tilde{O}\left(\frac{n^{2/3}}{\varepsilon^2}\right) + \tilde{O}\left(\frac{n^{1/3}}{m2^t}\right)\left(\tilde{O}\left(\frac{m}{\xi}\right) + \tilde{O}\left(\frac{2^h}{2^s}\right)\right) = \tilde{O}(n^{2/3}/\varepsilon^2).$$

In order to analyze `AlgorithmCase1.2` we consider one of the main iterations and let T denote the set drawn in line 2. We define the following subset $C \subseteq T \cap \mathcal{I}^*$ to capture how good T is: A variable $i \in T \cap \mathcal{I}^*$ belongs to C if it satisfies both of the following conditions:

(i) The set $T \setminus \{i\}$ is i-informative for anti-monotone edges; and

(ii) If we draw $\mathbf{S}$ by first drawing a subset $\mathbf{S}_0$ of T of size m conditioning on $i \in \mathbf{S}_0$ and an ordering of $\boldsymbol{\pi}$ of $\mathbf{S}_0$ uniformly at random and then calling $\texttt{Preprocess}(f, \mathbf{S}_0, \boldsymbol{\pi}, \xi)$ to get $\mathbf{S}$, then the probability that $\mathbf{S}$ catches i is at least $1/2$.

We prove that when $\mathbf{T}$ is a random size-p set, the set $\mathbf{C}$ is large with constant probability.

Lemma 7.5.11. *With probability at least $\Omega(1)$ over the draw of $\mathbf{T} \subset [n]$ in line 2, we have*

$$|\mathbf{C}| = \Omega\left(\frac{p|\mathcal{I}^*|}{n}\right).$$

Proof. We first note that $p|\mathcal{I}^*|/n = \Omega(n^{1/6})$ since $|\mathcal{I}^*|/2^t = \Omega(n^{2/3})$. By Lemma 7.11.1,

$$|\mathbf{C}| \leq |\mathbf{T} \cap \mathcal{I}^*| \leq 4p|\mathcal{I}^*|/n$$

with probability at least $1 - \exp(-\Omega(n^{1/6}))$. We consider the quantity

$$\alpha = \Pr_{\mathbf{T} \subset [n]}\left[|\mathbf{C}| \geq \frac{1}{10} \cdot \frac{p|\mathcal{I}^*|}{n}\right],$$

which we will lower bound by $\Omega(1)$ by proving both upper and lower bounds for $\mathbf{E}_{\mathbf{T}}[|\mathbf{C}|]$.

For the lower bound, we use the following claim which we prove next.

Claim 7.5.12. *For every $i \in \mathcal{I}^*$, we have*

$$\Pr_{\mathbf{T} \subset [n]}\left[\mathbf{T} \textit{ and } i \textit{ satisfy conditions (i) and (ii)} \,\middle|\, i \in \mathbf{T}\right] \geq 1 - o(1).$$

It follows from Claim 7.5.12 that

$$\mathop{\mathbf{E}}_{\mathbf{T} \subset [n]}\left[|\mathbf{C}|\right] = \sum_{i \in \mathcal{I}^*} \Pr_{\mathbf{T} \subset [n]}\left[i \in \mathbf{T}\right] \cdot \Pr_{\mathbf{T} \subset [n]}\left[\text{(i) and (ii) satisfied} \,\middle|\, i \in \mathbf{T}\right] \geq (1 - o(1)) \cdot \frac{p|\mathcal{I}^*|}{n}.$$

On the other hand, we may upper bound $\mathbf{E}_{\mathbf{T}}[|\mathbf{C}|]$ using the definition of α,

$$\mathop{\mathbf{E}}_{\mathbf{T} \subset [n]}\left[|\mathbf{C}|\right] \leq n \cdot \exp\left(-\Omega(n^{1/6})\right) + \left(1 - \exp\left(-\Omega(n^{1/6})\right)\right)\left(\alpha \cdot \frac{4p|\mathcal{I}^*|}{n} + (1 - \alpha) \cdot \frac{p|\mathcal{I}^*|}{10n}\right),$$

which in turn, implies that $\alpha = \Omega(1)$. $\square$

Proof of Claim 7.5.12. Consider a fixed $i \in \mathcal{I}^*$, we will show that sampling $\mathbf{T}$ and conditioning on $i \in \mathbf{T}$, the probability of i satisfying condition (i) is at least $1 - o(1)$, and likewise, the probability of i satisfying condition (ii) is also at least $1 - o(1)$. The claim then follows from a union bound.

First, $\mathbf{T} \subset [n]$ conditioned on $i \in \mathbf{T}$ is distributed exactly as $\mathbf{T}' \cup \{i\}$ where $\mathbf{T}' \sim \mathcal{P}_{i,\lceil\sqrt{n}/2^t\rceil}$, by our choice of p. The part on condition (i) follows directly from Lemma 7.5.3.

Second, let β be the probability over $\mathbf{T} \subset [n]$ conditioning on $i \in \mathbf{T}$ that (ii) is not satisfied. We consider the following quantity

$$\Pr_{\mathbf{S} \sim \mathcal{D}_{\xi,m,i}} \Big[\mathbf{S} \text{ catches } i\Big].$$

On the one hand, we can we bound it from above by $1 - o(1)$ by combining Claim 7.5.5 and Corollary 7.5.2. On the other hand, $\mathbf{S} \sim \mathcal{D}_{\xi,m,i}$ is exactly distributed as first sampling $\mathbf{T} \subset [n]$, then sampling $\mathbf{S}_0 \subset \mathbf{T}$ conditioned on $i \in \mathbf{S}_0$ and finally running `Preprocess` using $\mathbf{S}_0$ to get $\mathbf{S}$. Thus, we may use the definition of β to upperbound the probability by $\beta/2 + 1 - \beta$, which implies $\beta = o(1)$. $\square$

Having established Lemma 7.5.11, we consider a fixed iteration of line 2 where the set C obtained from T satisfies the size lower bound from Lemma 7.5.11. Note that since line 2 is executed $O(1)$ times, this will happen with large constant probability. After fixing T and C, we will show that in this iteration, `AlgorithmCase1.2` finds an edge violation to unateness with high probability.

Lemma 7.5.13. *With probability* $1 - o(1)$ *over the randomness in lines 3–5,* $|\mathbf{A} \cap C| \geq \Omega(n^{1/6})$.

Proof. We consider breaking up the execution of lines 3–4 into $O(n^{1/6})$ phases, each consists of

$$O\left(\frac{\sqrt{n}\,\Lambda^{11}\log^2 n}{\varepsilon^2}\right)$$

execution of line 4 each. We note that

$$|C| = \Omega\left(\frac{p|\mathcal{I}^*|}{n}\right) = \Omega(n^{1/6})$$

since $|\mathcal{I}|/2^t \geq n^{2/3}$. Consider a particular phase of the algorithm, and assume that the number of anti-monotone edges along variables in C observed is less than $|C|/2$ (otherwise, we are done since $|C| = \Omega(n^{1/6})$). In this case, for each $j \in C$ along which the algorithm has not observed an anti-monotone edge, we let X_j be the set of points $x \in \{0,1\}^n$ such that AE-SEARCH(f, x, T) will output j with probability at least $2/3$. By condition (i), we have that

$$|X_j| = \Omega\left(\frac{2^t}{2^h}\right) \cdot 2^n,$$

and that the X_j's are disjoint by Corollary 7.1.3. As a result, there are at least

$$\Omega\left(\frac{2^t}{2^h}\right) \cdot 2^n \cdot \frac{|\mathcal{I}^*|}{2^t\sqrt{n}} = \Omega\left(\frac{|\mathcal{I}^*|}{2^h\sqrt{n}}\right) \cdot 2^n = \Omega\left(\frac{\varepsilon^2}{\sqrt{n} \cdot \Lambda^{11}}\right) \cdot 2^n$$

many such points $x \in \{0,1\}^n$. By the number of queries we have in each phase, we will observe an anti-monotone edge along some variable in C not yet seen with high probability. □

By Lemma 7.5.13 we consider the case when $|\mathbf{A} \cap C| = \Omega(n^{1/6})$. Fix a particular A and a subset of $\mathcal{J} = A \cap C$ such that $|\mathcal{J}| = \Theta(n^{1/6})$.

Lemma 7.5.14. *With probability at least $1 - o(1)$, at least one of the $\mathbf{S}$ sampled in line 7 catches at least one variable in $\mathcal{J}$.*

Proof. Let β be the probability that $\mathbf{S}$ catches at least one variable in $\mathcal{J}$. Consider

$$\mathop{\mathbf{E}}_{\mathbf{S}}\left[\left\{j \in \mathcal{J} : \mathbf{S} \text{ catches } j\right\}\right].$$

Using condition (ii) on variables in $\mathcal{J}$, we have

$$\mathop{\mathbf{E}}_{\mathbf{S}}\left[\left\{j \in \mathcal{J} : \mathbf{S} \text{ catches } j\right\}\right] = \sum_{j \in \mathcal{J}} \mathbf{Pr}\left[\mathbf{S}_0 \text{ contains } j\right] \cdot \mathbf{Pr}\left[\mathbf{S} \text{ catches } j \,\middle|\, j \in \mathbf{S}_0\right] = \Omega\left(\frac{m2^t}{n^{1/3}}\right).$$

Since $\mathbf{S}_0$ is a uniform subset of $\mathbf{T}$ of size m, and $|\mathcal{J}| = \Theta(n^{1/6})$, the expectation of $|\mathbf{S}_0 \cap \mathcal{J}|$ is

$$\Theta\left(\frac{mn^{1/6}}{\sqrt{n}/2^t}\right) = \Omega\left(\frac{m2^t}{n^{1/3}}\right) = O(\log^2 n).$$

Therefore, the probability of $|\mathbf{S} \cap \mathcal{J}| \leq |\mathbf{S}_0 \cap \mathcal{J}|$ being at most $O(\log^2 n)$ is at least $1 - \exp(-\Omega(\log^2 n))$ by Lemma 7.11.1. Thus, we may upperbound the above expectation using the definition of β by

$$\beta \cdot O(\log^2 n) + |\mathcal{J}| \cdot \exp\Big(-\Omega(\log^2 n)\Big),$$

which gives the following lower bound on β:

$$\beta \geq \Omega\left(\frac{m2^t}{n^{1/3}\log^2 n}\right).$$

The lemma follows from the number of times we repeat in line 7. □

Finally we show that if S catches a $j \in \mathcal{J}$, then line 9 finds a monotone edge along j.

Lemma 7.5.15. *If S catches $j \in \mathcal{J}$ in line 7, then line 9 finds a monotone edge along variable j with probability at least $1 - o(1)$.*

Proof. Since S catches j, the number of points $x \in \{0,1\}^n$ such that AE-SEARCH(f, x, S) returns a monotone edge along j with probability at least $2/3$ is at least

$$\Omega\left(\frac{2^s}{2^h}\right) \cdot 2^n$$

The lemma then follows from the number of times the algorithm repeats in line 8. □

7.6 Finding Bichromatic Edges of High Influence Variables

The goal of this section is to present a procedure `Find-Hi-Inf` which will be used in our algorithm for Case 2. The procedure `Find-Hi-Inf` takes four inputs: query access

to a Boolean function f, a set $S \subseteq [n]$ of variables, a positive integer m and a parameter $\alpha \in (0, 1]$. When f satisfies:

$$\text{There is a hidden subset } H \subseteq S \text{ such that } |H| \geq m \text{ and every } h \in H \text{ satisfies } \mathbf{Inf}_f[h] \geq \alpha, \tag{7.22}$$

the procedure `Find-Hi-Inf` efficiently finds a bichromatic edge for *almost all* variables in H.

At a high level, the procedure `Find-Hi-Inf` (described in Figure 7.11) will exploit the fact that if there is a hidden set H of variables with high influence, there will be sufficiently many points in the hypercube which are sensitive along many variables from H. As a result, every time one such high sensitivity point is identified, we may use it to observe bichromatic edges along multiple variables.

7.6.1 Revealing points

We start with the definition of *revealing points*. These are points that, once identified, can be used to observe bichromatic edges along multiple variables. We then present two subroutines that will be used in `Find-Hi-Inf`. Given a revealing point, the first subroutine `Get-Revealing-Edges` (see Figure 7.8) uses it to find bichromatic edges along a large number of variables. While it is expensive to run, we present a second subroutine `Check-Revealing` (see Figure 7.9) that can check whether a given point is revealing so we only run the first one when we are sure that the point is revealing.

Definition 64. *Let $\delta \in (0, 1]$ and $T = \{T_j\}_{j \in [r]}$ be a collection of disjoint (but possibly empty) subsets of $[n]$ for some $r \geq 1$. Given a point $x \in \{0, 1\}^n$, we consider the set:*

$$\textsc{Reveal}(x, T, \delta) = \left\{ j \in [r] : \mathop{\mathbf{Pr}}_{\mathbf{R} \subseteq T_j} \left[f(x) \neq f(x^{(\mathbf{R})}) \right] \geq \delta \right\},$$

where $\mathbf{R} \subseteq T_j$ is subset of T_j drawn uniformly at random (i.e., each element of T_j is included in $\mathbf{R}$ with probability $1/2$ independently). Moreover, for $\gamma \in (0, 1]$, we say x is (γ, δ)-revealing *for T if*

$$\left| \textsc{Reveal}(x, T, \delta) \right| \geq \gamma \cdot r.$$

Subroutine `Get-Revealing-Edges`(f, x, T, δ)

Input: Query access to $f\colon \{0,1\}^n \to \{0,1\}$, a point $x \in \{0,1\}^n$, a collection of disjoint subsets $T = \{T_j\}_{j\in[r]}$ of $[n]$ for some $r \geq 1$, and $\delta \in (0,1]$.
Output: Two random sets $\mathbf{B}$ and $\mathbf{Q}$, where $\mathbf{B}$ is a set of bichromatic edges of f and $\mathbf{Q}$ is a subset of the union of T_j's.

- Initialize $\mathbf{B} = \mathbf{Q} = \emptyset$. Repeat the following for each $j \in [r]$ for $\log^2 n/\delta$ iterations each.
  1. Sample $\mathbf{R} \subseteq T_j$ uniformly at random, and let $\boldsymbol{\pi}$ be an arbitrary ordering on $\mathbf{R}$.
  2. Run `BinarySearch`$(f, x, \mathbf{T}, \boldsymbol{\pi})$. If it outputs `nil`, do nothing. If it outputs a bichromatic edge $\boldsymbol{e}$ along variable $\boldsymbol{i} \in \mathbf{T}$. Set $\mathbf{B} \leftarrow \mathbf{B} \cup \{\boldsymbol{e}\}$ and $\mathbf{Q} \leftarrow \mathbf{Q} \cup \{\boldsymbol{i}\}$.
- Output $\mathbf{B}$ and $\mathbf{Q}$.

Figure 7.8: The subroutine `Get-Revealing-Edges`.

Next we present the first subroutine `Get-Revealing-Edges`. Given $x \in \{0,1\}^n$, $T = \{T_j\}_{j\in[r]}$, and $\delta \in (0,1]$, `Get-Revealing-Edges`(f, x, T, δ) (see Figure 7.8) makes $\tilde{O}(r/\delta)$ queries and outputs a set of bichromatic edges which contains one bichromatic edge for each variable in $\textsc{Reveal}(x, T, \delta)$.

Lemma 7.6.1. *Let $x \in \{0,1\}^n$, $\delta \in (0,1]$ and $T = \{T_j\}_{j\in[r]}$ be a collection of r disjoint subsets of $[n]$ for some $r \geq 1$. Then,* `Get-Revealing-Edges`(f, x, T, δ) *uses $\tilde{O}(r/\delta)$ queries and always returns a set $\mathbf{Q}$ of variables and a set $\mathbf{B}$ of bichromatic edges such that $\mathbf{Q}$ is a subset of the union of T_j's and $\mathbf{B}$ contains a bichromatic edge along each variable in $\mathbf{Q}$. Furthermore, $\mathbf{Q}$ contains $\textsc{Reveal}(x, T, \delta)$ with probability at least $1 - 1/\mathrm{poly}(n)$.*

Proof. For each index $j \in \textsc{Reveal}(x, T, \delta)$, the probability that none of the $\log^2 n/\delta$ many random subsets $\mathbf{R}$ of T_j satisfies $f(x) \neq f(x^{(\mathbf{R})})$ is at most

$$(1-\delta)^{\frac{\log^2 n}{\delta}} \ll 1/\mathrm{poly}(n),$$

When this happens `BinarySearch`$(f, x, \mathbf{R}, \boldsymbol{\pi})$ finds a bichromatic edge along a variable in $\mathbf{R}$ which is also in T_j. The lemma then follows from a union bound over all $j \in$

Subroutine `Check-Revealing`$(f, x, T, \gamma, \delta)$

Input: Query access to $f\colon \{0,1\}^n \to \{0,1\}$, a point $x \in \{0,1\}^n$, a collection of disjoint subsets $T = \{T_j\}_{j\in[r]}$ of $[n]$ for some $r \geq 1$, and $\gamma, \delta \in (0,1]$.
Output: Either "accept" or "reject."

1. Initialize $\boldsymbol{b} = 0$.
2. Repeat the following steps for $\log^2 n/\gamma$ iterations:
   - Sample an index $\boldsymbol{j} \sim [r]$ uniformly at random and initialize $\boldsymbol{c} = 0$.
   - Repeat the following $\log^2 n/\delta$ iterations:
     - Sample a subset $\mathbf{R} \subseteq T_{\boldsymbol{j}}$ uniformly and increment $\boldsymbol{c}$ if $f(x^{(\mathbf{R})}) \neq f(x)$.
   - If $\boldsymbol{c} \geq 3\log^2 n/4$, increment $\boldsymbol{b}$.
3. If $\boldsymbol{b} \geq 3\log^2 n/4$, output "accept;" otherwise, output "reject."

Figure 7.9: The subroutine `Check-Revealing`.

$\textsc{Reveal}(x, T, \delta)$. □

Next we describe the second subroutine `Check-Revealing` in Figure 7.9, which we use to check whether a given point x is (γ, δ)-revealing with respect to T.

Lemma 7.6.2. *Let $\gamma, \delta \in (0,1]$, $x \in \{0,1\}^n$ and $T = \{T_j\}_{j\in[r]}$ be a collection of disjoint subsets of $[n]$ for some $r \geq 1$. Then,* `Check-Revealing`$(f, x, T, \gamma, \delta)$ *makes $\tilde{O}(1/(\gamma\delta))$ many queries and satisfies the following two conditions: (1) If x is (γ, δ)-revealing with respect to T, then it outputs "accept" with probability at least $1 - 1/\mathrm{poly}(n)$; (2) If x is not $(\gamma/2, \delta/2)$-revealing with respect to T, then it outputs "reject" with probability at least $1 - 1/\mathrm{poly}(n)$.*

Proof. Suppose that x is (γ, δ)-revealing. It follows from Chernoff bound that with probability at least $1 - 1/\mathrm{poly}(n)$, the number of iterations in which the index $j \in [r]$ lies in $\textsc{Reveal}(x, T, \delta)$ is at least $3\log^2 n/4$. Moreover, for every such index j, the counter b is incremented with probability at least $1 - 1/\mathrm{poly}(n)$. By a union bound, the subroutine accepts with probability $1 - \mathrm{poly}(n)$.

Suppose that x is not $(\gamma/2, \delta/2)$-revealing. Then we have that $|\textsc{Reveal}(x, T, \delta/2)| \leq \gamma r/2$. The claim follows similarly by applying Chernoff bounds and then a union bound. $\square$

We can combine Lemma 7.6.1 and 7.6.2 to conclude that when `Check-Revealing` (x, T, γ, δ) returns "accept," it is safe to run `Get-Revealing-Edges` $(x, T, \delta/2)$ and we should expect the latter to find at least $\gamma r/2$ many bichromatic edges along different variables from the union of T_j's.

7.6.2 The `Find-Revealing` procedure

Recall that we are given an $S \subseteq [n]$ and two parameters m and α, and we are interested in the case when S contains a hidden set $H \subseteq S$ that satisfies (7.22).

Assuming this is the case, we prove in Corollary 7.6.6 that there exist many points z such that, with probability $\Omega(1)$ over the draw of a random partition $\mathbf{T}$ of S, z is $(\Omega(1), \Omega(1))$-revealing for $\mathbf{T}$. We then present a procedure called `Find-Revealing` that takes advantage of this property to find bichromatic edges along many different variables in S. Note that in establishing Corollary 7.6.6, we assume that there is a subset H of S that satisfies (7.22), and H is used in the analysis (and known to us in the proofs of these lemmas), even though H is hidden from the procedure `Find-Revealing`.

We start by using the set H to define the following bipartite graph $G_0 = (U_0, V_0, E_0)$ consisting of bichromatic edges of f along variables in H:

- $U_0 \subseteq \{0,1\}^n$ is the set of left vertices that contain all $x \in \{0,1\}^n$ with $f(x) = 0$.
- $V_0 \subseteq \{0,1\}^n$ is the set of right vertices that contain all $x \in \{0,1\}^n$ with $f(x) = 1$.
- E_0 is the set of bichromatic edges of f connecting vertices between U_0 and V_0 along variables in H. Given that $\mathbf{Inf}_f[i] \geq \alpha$, we have $|E_0| \geq (\alpha|H|/2) \cdot 2^n \geq (\alpha m/2) \cdot 2^n$.

Recall the definition of left-d-good and right-d-good bipartite graphs. The next lemma, similar to Lemma 6.5 of [87], shows the existence of an induced subgraph of G_0 that has roughly the same number of edges as G_0 but is either left-d-good or right-d-good.

Lemma 7.6.3. *There exist a positive integer $d \leq |H|$ that is a power of 2 as well as subsets $U \subseteq U_0$ and $V \subseteq V_0$ such that the subgraph $G = (U, V, E)$ of G_0 induced by vertices*

(U, V) *satisfies:*

- G *is either left-*d*-good or right-*d*-good (letting* $\sigma = |U|/2^n$ *or* $|V|/2^n$ *accordingly), and*
- σ *and* d *satisfy* $\sigma d \geq \alpha m/(6 \log n)$.

Proof. Note that G_0 has degree at most $|H|$. Consider the following procedure:

- Iterate through $d = 2^k, 2^{k-1}, \ldots, 1$, where 2^k is the largest power of 2 that is at most $|H|$, while maintaining a set of vertices D (as vertices deleted so far) which is initially empty:
  1. Consider the subgraph G' of G_0 induced by $(U_0 \setminus D, V_0 \setminus D)$ and let

$$U^* = \Big\{x \in U_0 \backslash D : \deg_{G'}(x) \geq d\Big\} \quad \text{and} \quad V^* = \Big\{y \in V_0 \backslash D : \deg_{G'}(y) \geq d\Big\}.$$

  2. Terminate if either U^* or V^* has size at least

$$\frac{\alpha m}{6d \log n} \cdot 2^n.$$

  3. Otherwise, we update $D \leftarrow D \cup U^* \cup V^*$ (i.e., delete all vertices in U^* and V^*).

By induction, we have that at the beginning of each round, every vertex in G' has degree at most $2d$. As a result, if the procedure terminates during one of the iterations, we have that the subgraph of G_0 induced by either $(U^*, V_0 \setminus D)$ or $(U_0 \setminus D, V^*)$ satisfies both properties and we are done.

So it suffices to show that the procedure terminates. Assume for contradiction that it does not terminate. Then the number of edges deleted (i.e., those adjacent to vertices in $U^* \cup V^*$) during each iteration is at most $(\alpha m/(3 \log n)) \cdot 2^n$. Since there are no more than $\log n$ iterations, the total number of edges deleted is at most $(\alpha m/3) \cdot 2^n$. This contradicts with the fact that all edges of H_0 will be deleted at the end if the procedure does not terminate, and that $|E_0| \geq (\alpha m/2) \cdot 2^n$. □

Consider such a subgraph $G = (U, V, E)$ of G_0 with parameters d and σ, given by Lemma 7.6.3. We assume without loss of generality G is left-d-good since the proof of Corollary 7.6.6 is symmetric when G is right-d-good by considering f' given by $f'(x) = f(x) \oplus 1$. For each $z \in U \cup V$, we define

$$N(z) = \Big\{ i \in H : (x, x^{(i)}) \in E \Big\}.$$

Thus, $|N(z)| = \deg_G(z)$. Corollary 7.6.6 follows from two technical lemmas (Lemma 7.6.4 and Lemma 7.6.5). Before stating them we need the following definition.

Definition 65. *Let $\gamma_0 = \delta_0 = \eta_0 = 0.1$ be three constants and $r = 100d$. We say that a point $y \in V$ is* good for r-partitions *of S if with probability at least η_0 over a uniformly random r-partition $\mathbf{T} = \{\mathbf{T}_j\}_{j \in [r]}$ of $S \setminus N(y)$, y is (γ_0, δ_0)-revealing with respect to $\mathbf{T}$. (To be more formal, such a partition $\mathbf{T}$ is drawn by first sampling a map $\boldsymbol{g}$ from $S \setminus N(y)$ to $[r]$ uniformly at random, i.e., each element is mapped to an index in $[r]$ uniformly and independently, and then setting $\mathbf{T}_j = \boldsymbol{g}^{-1}(j)$.)*

We use $\textsc{Good}(S)$ to denote the set of points $y \in V$ that are good for r-partitions of S.

We delay the proof of the following two technical lemmas.

Lemma 7.6.4. *For each point $y \in \textsc{Good}(S)$, with probability at least $\eta_0/2$ over the draw of a random r-partition $\mathbf{T}$ of S, y is $(\gamma_0/2, \delta_0/2)$-revealing for $\mathbf{T}$.*

Lemma 7.6.5. *If $|\textsc{Good}(S)| \leq (\sigma/100) \cdot 2^n$, there exist $\sigma 2^n/2$ points $x \in U$ such that with probability at least $1/3$ over the draw of a random r-partition $\mathbf{T}$ of S, x is $(1/4000, (1 - \delta_0)/2)$-revealing for $\mathbf{T}$.*

Using constants η_1, γ_1 and δ_1 defined in Figure 7.10, we get the following corollary:

Corollary 7.6.6. *Assume that there is an $H \subseteq S$ that satisfies (7.22). Then there are at least $\Omega(\sigma 2^n)$ many points $z \in \{0,1\}^n$ such that, with probability at least η_1 over the draw of a random r-partition $\mathbf{T}$ of S, z is (γ_1, δ_1)-revealing for $\mathbf{T}$.*

Procedure `Find-Revealing`(f, S, m, α)

Input: Query access to $f\colon \{0,1\}^n \to \{0,1\}$, $S \subseteq [n]$, $m \geq 1$ and $\alpha \in (0,1]$.
Output: Either a set of bichromatic edges $\mathbf{B}$ and a subset $\mathbf{Q}$ of S, or "fail."

- Let $\eta_1 = \min\left(\eta_0/2, 1/3\right)$, $\gamma_1 = \min\left(\gamma_0/2, 1/4000\right)$ and $\delta_1 = \min\left(\delta_0/2, (1-\delta_0)/2\right)$.
- Repeat the following for all d^* being a power of 2 less than $|S|$ from small to large:
  1. Let $r^* = 100d^*$ and $\sigma^* = \alpha m/(6d^* \log n)$. Skip this iteration if $\sigma^* > 1$.
  2. Sample $t = \log^2 n/\sigma^*$ points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t$ from $\{0,1\}^n$ uniformly at random.
  3. For each point $\boldsymbol{x}_j$, for $\log^2 n/\eta_1$ times, let $\mathbf{T}$ be an r^*-partition of S drawn uniformly at random, and run `Check-Revealing`$(f, \boldsymbol{x}_j, \mathbf{T}, \gamma_1, \delta_1)$.
     - If it outputs "reject," do nothing.
     - If it outputs "accept," run `Get-Revealing-Edges` $(f, \boldsymbol{x}_j, \mathbf{T}, \delta_1/2)$ to get $\mathbf{B}$ and $\mathbf{Q}$. If $|\mathbf{Q}| < \gamma_1 r^*/2$, output "fail" and terminate; otherwise, output $(\mathbf{Q}, \mathbf{B})$.
- When this line is reached (i.e. all calls to `Check-Revealing` return "fail") output "fail."

Figure 7.10: The procedure `Find-Revealing`.

Proof. We consider two cases: $|\textsc{Good}(S)| \geq (\sigma/100) \cdot 2^n$ and $|\textsc{Good}(S)| \leq (\sigma/100) \cdot 2^n$. Then the statement follows from Lemma 7.6.4 and Lemma 7.6.5 for these two cases respectively. □

The property stated in Corollary 7.6.6 inspires our next procedure `Find-Revealing` described in Figure 7.10. We use Corollary 7.6.6 to prove the following lemma:

Lemma 7.6.7. *Let $f : \{0,1\}^n \to \{0,1\}$, $S \subseteq [n]$, $m \geq 1$, and $\alpha \in (0,1]$.* `Find-Revealing`(f, S, m, α) *outputs either "fail" or a pair $(\mathbf{Q}, \mathbf{B})$ such that $\mathbf{Q} \subseteq S$ is nonempty and $\mathbf{B}$ contains one bichromatic edge for each variable in $\mathbf{Q}$. The number of queries it uses can be bounded*

from above by

$$\begin{cases} \tilde{O}(|\mathbf{Q}|) + \tilde{O}\big(|\mathbf{Q}|/(\alpha m)\big) & \textit{when it outputs a pair } (\mathbf{B}, \mathbf{Q})\textit{; and} \\ \tilde{O}(|S|) + \tilde{O}\big(|S|/\alpha m\big) & \textit{when it outputs ``fail.''} \end{cases}$$

Furthermore, when S contains a subset H that satisfies (7.22), `Find-Revealing`(f, S, m, α) *outputs a pair* $(\mathbf{B}, \mathbf{Q})$ *with probability at least* $1 - 1/\mathrm{poly}(n)$.

Proof. It follows from Lemma 7.6.1 that when `Find-Revealing` outputs a pair $(\mathbf{Q}, \mathbf{B})$, $\mathbf{Q}$ is a subset of S and $\mathbf{B}$ contains one bichromatic edge for each variable in $\mathbf{Q}$. We also have that $\mathbf{Q}$ is nonempty since we compare $|\mathbf{Q}|$ with $\gamma_1 r^*/2 > 0$ before returning $(\mathbf{B}, \mathbf{Q})$.

The number of queries used when `Find-Revealing` outputs "fail" follows from its description. Next we bound the number of queries used when it outputs a pair $(\mathbf{Q}, \mathbf{B})$.

Let d' be the value of d^* when `Find-Revealing` terminates. Then we have $|\mathbf{Q}| \geq \gamma_1 r'/2$, where $r' = 100d'$. The number of queries used by `Check-Revealing` in each iteration is at most

$$\log^2 n \cdot \frac{6d^* \log n}{\alpha m} \cdot O(\log^2 n) \cdot \tilde{O}(1) = \tilde{O}\left(\frac{d^*}{\alpha m}\right).$$

As a result, the total number of queries used by `Check-Revealing` is at most

$$\tilde{O}\left(\frac{1 + 2 + \cdots + d'}{\alpha m}\right) = \tilde{O}\left(\frac{d'}{\alpha m}\right) = \tilde{O}\left(\frac{|\mathbf{Q}|}{\alpha m}\right).$$

On the other hand, we only make one call to `Get-Revealing-Edges` during the last iteration of d'. So the number of queries it uses is at most $\tilde{O}(r') = \tilde{O}(|\mathbf{Q}|)$. It then follows that the total number of queries used can be bounded using the expression given in the lemma.

It is left to show that `Find-Revealing` returns a pair $(\mathbf{B}, \mathbf{Q})$ with high probability. Note that it returns "fail" for two cases. Either all calls to `Check-Revealing` return "fail" or one of these calls returns "accept" but the next call to `Get-Revealing-Edges` returns "fail." The first event happens with small probability because when $d^* = d$, we have $\sigma \geq \sigma^*$

from Lemma 7.6.3 and therefore, with probability at least $1 - \exp(-\log^2 n)$ we get a point $\boldsymbol{x}$ that satisfies Corollary 7.6.6 on line 2. For this $\boldsymbol{x}$ with probability $1 - \exp(-\log^2 n)$ we get a $\mathbf{T}$ such that $\boldsymbol{x}$ is (γ_1, δ_1)-revealing for $\mathbf{T}$ on line 3. When this happens, it follows from Lemma 7.6.2 that the subroutine `Check-Revealing` outputs "accept" with probability at least $1 - 1/\mathrm{poly}(n)$.

For the second event to happen, either one of the calls to `Check-Revealing` $(f, \boldsymbol{x}, \mathbf{T}, \gamma_1, \delta_1)$ returns "accept" while $\boldsymbol{x}$ is not $(\gamma_1/2, \delta_1/2)$-revealing for $\mathbf{T}$, or the only call to `Get-Revealing-Edges` $(f, \boldsymbol{x}, \mathbf{T}, \delta_1/2)$ fails to find enough bichromatic edges while $\boldsymbol{x}$ is indeed $(\gamma_1/2, \delta_1/2)$-revealing for $\mathbf{T}$. It follows from Lemma 7.6.1 and 7.6.2 and a union bound that this happens with low probability. □

Next we prove Lemma 7.6.4 and Lemma 7.6.5. We start with Lemma 7.6.4.

Proof of Lemma 7.6.4. Consider the following procedure to draw an r-partition $\mathbf{T} = \{\mathbf{T}_j\}_{j\in[r]}$ of S:

1. Sample $\boldsymbol{g}_0 \colon S \setminus N(y) \to [r]$ uniformly at random; let $\mathbf{T}^{(0)} = \left\{\mathbf{T}^{(0)}_j\right\}_{j\in[r]}$ with $\mathbf{T}^{(0)}_j = \boldsymbol{g}_0^{-1}(j)$.
2. Sample $\boldsymbol{g}_1 \colon N(y) \to [r]$ uniformly at random; let $\mathbf{T}^{(1)} = \left\{\mathbf{T}^{(1)}_j\right\}_{j\in[r]}$ with $\mathbf{T}^{(1)}_j = \boldsymbol{g}_1^{-1}(j)$.
3. Let $\mathbf{T} = \left\{\mathbf{T}_j\right\}_{j\in[r]}$ be given by $\mathbf{T}_j = \mathbf{T}^{(0)}_j \cup \mathbf{T}^{(1)}_j$.

We note that the above procedure samples a uniformly random r-partition $\mathbf{T}$ of S. Consider the following set $\mathbf{P} \subseteq [r]$ defined using $\mathbf{T}^{(0)}$ and $\mathbf{T}^{(1)}$:

$$\mathbf{P} = \left\{i \in [r] : i \in \textsc{Reveal}(y, \mathbf{T}^{(0)}, \delta_0) \text{ and } \left|\mathbf{T}^{(1)}_i\right| \le 1\right\}.$$

We note that every $i \in \mathbf{P}$ satisfies

$$\begin{aligned}\Pr_{\mathbf{R}\subseteq\mathbf{T}_i}\left[f(y) \neq f(y^{(\mathbf{R})})\right] &\ge \Pr_{\mathbf{R}\subseteq\mathbf{T}_i}\left[\mathbf{R}\cap\mathbf{T}^{(1)}_i = \emptyset\right] \cdot \Pr_{\mathbf{R}\subseteq\mathbf{T}_i}\left[f(y) \neq f(y^{(\mathbf{R})}) \,\middle|\, \mathbf{R}\cap\mathbf{T}^{(1)}_i = \emptyset\right] \\ &\ge 0.5 \cdot \Pr_{\mathbf{R}\subseteq\mathbf{T}^{(0)}_i}\left[f(y) \neq f(y^{(\mathbf{R})})\right] \ge \delta_0/2,\end{aligned}$$

where we used the fact that $i \in \mathbf{P}$ implies $|\mathbf{T}_i^{(1)}| \leq 1$ so that $\mathbf{R} \cap \mathbf{T}_i^{(1)} = \emptyset$ with probability at least $1/2$, and the fact that $i \in \textsc{Reveal}(y, \mathbf{T}^{(0)}, \delta_0)$.

Therefore, it suffices to show that with probability at least $\eta_0/2$ over the draw of $\mathbf{T}$, $|\mathbf{P}| \geq \gamma_0 r/2$. Towards this goal, we consider the event E which occurs when y is (γ_0, δ_0)-revealing with respect to $\mathbf{T}^{(0)}$. Since $y \in \textsc{Good}(S)$, the event E occurs with probability at least η_0.

Fix an r-partition $T^{(0)}$ of $S \backslash N(y)$ such that E occurs, and let $\textsc{Reveal} = \textsc{Reveal}(y, T^{(0)}, \delta_0)$. Then for each $j \in \textsc{Reveal}$ the probability of $|\mathbf{T}_j^{(1)}| \geq 2$ is at most (using $r = 100d$)

$$\binom{2d}{2} \cdot \frac{1}{r^2} \leq \frac{1}{5000}.$$

So the number of $j \in \textsc{Reveal}$ with $|\mathbf{T}_j^{(1)}| \geq 2$ being more than $|\textsc{Reveal}|/2$ can only happen with probability at most $1/2500$, and when this does not happen, the size of $\mathbf{P}$ is at least $|\textsc{Reveal}|/2 \geq \gamma_0 r/2$. Overall, this happens with probability at least $\eta(1 - 1/2500) \geq \eta_0/2$. □

Finally we prove Lemma 7.6.5. We start with some notation. Given a point $x \in U$, we let

$$Y(x) = \left\{x^{(i)} \in V : i \in N(x)\right\}$$

denote the set of neighbors of x in G, where $|Y(x)| = \deg_G(x) \in [d : 2d]$. Given a $y \in Y(x)$ and an r-partition $T = \{T_i\}$ of S, let $T^{(0,y)} = \{T_i^{(0,y)}\}$ denote the r-partition of $S \setminus N(y)$ with

$$T_i^{(0,y)} = T_i \setminus N(y).$$

Proof of Lemma 7.6.5. Let $U' = \left\{x \in U : |Y(x) \cap \textsc{Good}(S)| \geq (3/4) \cdot |Y(x)|\right\}$. We note that:

$$|U'| \cdot d \leq \sum_{x \in U'} \deg_G(x) \leq |\textsc{Good}(S)| \cdot 2d.$$

Using $|\textsc{Good}(S)| \leq (\sigma/100) \cdot 2^n$ and $|U| \geq \sigma 2^n$, at least $\sigma 2^n/2$ many points $x \in U$ satisfy $|Y(x) \cap \textsc{Good}(S)| \leq (3/4) \cdot |Y(x)|$. We prove in the rest of the proof that every such

$x \in U$ is $(1/4000, (1-\delta_0)/2)$-revealing for $\mathbf{T}$ with probability at least $1/3$ over the draw of a random r-partition $\mathbf{T}$ of S.

Given a partition $T = \{T_j\}_{j=1}^r$ of S we consider the following subset $A(T)$ of $Y(x) \setminus \textsc{Good}(S)$:

$$\left\{ x^{(i)} = y \in Y(x) \setminus \textsc{Good}(S) : \text{for the } k \in [r] \text{ with } i \in T_k, \begin{array}{l} \text{(i) } \ T_k \cap (N(x) \cup N(y)) = \{i\} \\ \text{(ii) } \ k \notin \textsc{Reveal}(y, T^{(0,y)}, \delta_0) \end{array} \right\}.$$

For each $x^{(i)} = y \in A(T)$, let $k \in [r]$ be the index with $i \in T_k$. Then we have

$$\begin{aligned} \Pr_{\mathbf{R} \subseteq T_k}\left[f(x) \neq f(x^{(\mathbf{R})})\right] &\geq \Pr_{\mathbf{R} \subseteq T_k}\left[i \in \mathbf{R}\right] \cdot \Pr_{\mathbf{R} \subseteq T_k}\left[f(x) \neq f(x^{(\mathbf{R})}) \,\middle|\, i \in \mathbf{R}\right] \\ &= 0.5 \cdot \Pr_{\mathbf{R}' \subseteq T_k^{(0,y)}}\left[f(x) \neq f(x^{(\mathbf{R}' \cup \{i\})})\right] \qquad (7.23) \\ &= 0.5 \cdot \Pr_{\mathbf{R}' \subseteq T_k^{(0,y)}}\left[f(y) = f(y^{(\mathbf{R}')})\right] \geq (1-\delta_0)/2. \qquad (7.24) \end{aligned}$$

Here (7.23) follows from the fact $i \in \mathbf{R}$ with probability $1/2$; In that case, letting $\mathbf{R}' = \mathbf{R} \setminus \{i\}$ gives $x^{(\mathbf{R})} = y^{(\mathbf{R}')}$, and because $T_k \cap N(y) = \{i\}$ we have that $\mathbf{R} \subseteq T_k$ conditioning on $i \in \mathbf{R}$ is distributed as $\mathbf{R}' \cup \{i\}$ with $\mathbf{R}' \subseteq T_k^{(0,y)}$. Additionally, (7.24) follows from the fact that $f(x) \neq f(y)$ since (x, y) is a bichromatic edge of f, and the fact that $k \notin \textsc{Reveal}(y, T^{(0,y)}, \delta_0)$.

Since $T_k \cap N(x) = \{i\}$ the number of $k \in [r]$ for which (7.24) holds is at least $|A(T)|$. As a result, x is $(|A(T)|/r, (1-\delta_0)/2)$-revealing for T. Therefore, in order to show the lemma it suffices to show that $|A(\mathbf{T})| \geq d/40$ with probability at least $1/3$ over the draw of a random r-partition $\mathbf{T}$ of $\mathbf{S}$.

For this purpose, we consider a fixed $x^{(i)} = y \in Y(x) \setminus \textsc{Good}(S)$ and show that $y \in A(\mathbf{T})$ with probability at least $1/2$. Similarly to the proof of Lemma 7.6.4, we can equivalently draw a uniformly random r-partition $\mathbf{T} = \{\mathbf{T}_j\}_{j \in [r]}$ of S by first drawing two uniformly random r-partitions

$$\mathbf{T}^{(0,y)} = \left\{\mathbf{T}_j^{(0,y)}\right\}_{j \in [r]} \quad \text{and} \quad \mathbf{T}^{(1,y)} = \left\{\mathbf{T}_j^{(1,y)}\right\}_{j \in [r]}$$

of $S \setminus N(y)$ and $N(y)$, respectively, and then setting $\mathbf{T}_j = \mathbf{T}_j^{(0,y)} \cup \mathbf{T}_j^{(1,y)}$ for each $j \in [r]$.

Then we have

$$\begin{aligned}
&\Pr_{\mathbf{T}}\left[y \notin A(\mathbf{T})\right] \\
&\quad \le \Pr_{\mathbf{T}}\left[\text{the } k \in [r] \text{ with } i \in \mathbf{T}_k \text{ satisfies } \mathbf{T}_k \cap (N(x) \cup N(y)) \neq \{i\}\right] \\
&\qquad + \Pr_{\mathbf{T}}\left[y \text{ is } (\gamma_0, \delta_0)\text{-revealing for } \mathbf{T}^{(0,y)}\right] \\
&\qquad + \Pr_{\mathbf{T}}\left[y \text{ is not } (\gamma_0, \delta_0)\text{-revealing for } \mathbf{T}^{(0,y)} \text{ and } \mathbf{T}_k \ni i \text{ has } k \notin \textsc{Reveal}(y, \mathbf{T}^{(0,y)}, \delta_0)\right] \\
&\quad \le \left(\frac{|N(x) \cup N(y)| - 1}{r}\right) + \eta_0 + (1-\eta_0)\gamma_0 \le \frac{4d}{r} + \eta_0 + \gamma_0 < 1/2,
\end{aligned}$$

using $r = 100d$. Therefore, the expected size of $A(\mathbf{T})$ is at least $|Y(x) \setminus \textsc{Good}(S)|/2$. Writing

$$\beta = \Pr_{\mathbf{T}}\left[|A(\mathbf{T})| \ge \frac{|Y(x) \setminus \textsc{Good}(S)|}{10}\right],$$

we have $1/2 \le \beta + (1-\beta)/10$ and thus, $\beta \ge (1/2) - (1/10) > 1/3$. So with probability at least $1/3$,

$$A(\mathbf{T}) \ge \frac{|Y(x) \setminus \textsc{Good}(S)|}{10} \ge \frac{1}{10} \cdot \frac{|Y(x)|}{4} \ge \frac{d}{40}$$

since $|Y(x) \cap \textsc{Good}(S)| \le (3/4) \cdot |Y(x)|$. This finishes the proof of the lemma. □

7.6.3 The `Find-Hi-Inf` procedure

Finally we describe the procedure `Find-Hi-Inf` in Figure 7.11 and prove the following lemma.

Lemma 7.6.8. *Let $S \subseteq [n]$, $m \ge 1$ and $\alpha \in (0,1]$. Then* `Find-Hi-Inf`(f, S, m, α) *makes at most*

$$\tilde{O}\left(|S| + \frac{|S|}{\alpha m}\right)$$

queries to f. When S contains a subset H that satisfies (7.22), with probability at least $1 - 1/\mathrm{poly}(n)$ `Find-Hi-Inf`(f, S, m, α) *outputs $(\mathbf{Q}, \mathbf{B})$ such that*

$$|\mathbf{Q} \cap H| \ge |H| - m$$

Subroutine `Find-Hi-Inf` (f, S, m, α)

Input: Query access to $f\colon \{0,1\}^n \to \{0,1\}$, $S \subseteq [n]$, $m \geq 1$ and $\alpha \in (0,1]$.
Output: A set of bichromatic edges $\mathbf{B}$ of f whose variables form a subset $\mathbf{Q} \subseteq S$.

1. Initialize $\mathbf{Q} = \mathbf{B} = \emptyset$, and $\mathbf{S}^* = S$.
2. Repeatedly run `Find-Revealing` $(f, \mathbf{S}^*, m, \alpha)$:
   - If it outputs "fail," terminate and output $\mathbf{Q}$ and $\mathbf{B}$.
   - Otherwise, if `Find-Revealing` outputs $(\mathbf{B}', \mathbf{Q}')$, update the sets as

$$\mathbf{Q} \leftarrow \mathbf{Q} \cup \mathbf{Q}', \quad \mathbf{B} \leftarrow \mathbf{B} \cup \mathbf{B}' \quad \text{and} \quad \mathbf{S}^* \leftarrow \mathbf{S}^* \setminus \mathbf{Q}'.$$

Figure 7.11: The procedure `Find-Hi-Inf`.

and $\mathbf{B}$ *contains one bichromatic edge for each variable in* $\mathbf{Q}$.

Proof. First note that the procedure terminates once `Find-Revealing` outputs "fail." On the other hand, whenever `Find-Revealing` outputs a pair $(\mathbf{B}', \mathbf{Q}')$, $|\mathbf{S}^*|$ decreases by $|\mathbf{Q}'|$. It then follows from Lemma 7.6.7 that the total number of queries used by calls to `Find-Revealing` except the last one that outputs "fail" is $\tilde{O}(|S|) + \tilde{O}(|S|/(\alpha m))$ since the total size of $\mathbf{Q}'$'s they output is at most $|S|$. By Lemma 7.6.7, the number of queries used by the last call can be bounded by the same expression.

Suppose that at some moment of the execution, we have $|\mathbf{Q} \cap H| < |H| - m$. Then $\mathbf{S}^* = S \setminus \mathbf{Q}$ satisfies $|\mathbf{S}^* \cap H| > m$. This implies that, for $|\mathbf{Q} \cap H| < |H| - m$ to happen at the end, a necessary condition is that one of the calls to `Find-Revealing` has $|\mathbf{S}^* \cap H| > m$ in input but outputs "fail." It follows from Lemma 7.6.7 and a union bound on at most $|S| \leq n$ many calls to `Find-Revealing` that this happens with probability at most $1/\mathrm{poly}(n)$. □

7.7 The Algorithm for Case 2

Below, we study Case 2 of the algorithm as described in Section 7.4. Let $s \geq t \in [\Lambda]$, $h \in [3\Lambda]$, and $\ell \in [\lfloor \log n \rfloor]$. We assume in Case 2 that the input function $f\colon \{0,1\}^n \to \{0,1\}$

Procedure `AlgorithmCase2`(f)

Input: Query access to a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$
Output: Either "unate," or two edges constituting an edge violation of f to unateness.

1. Repeat the following $O(1)$ times:
2. Draw $\mathbf{S} \subset [n]$ of size $n^{2/3}$ uniformly at random, and let $k = \lceil \frac{|\mathcal{I}^*|}{n^{1/3}\log n} \rceil$.
3. Let $(\mathbf{Q}, \mathbf{B}) \leftarrow$ `Find-Hi-Inf` $(f, \mathbf{S}, k, \alpha)$.
4. Repeat $O\left(\sqrt{n}\Lambda^{14}/\varepsilon^2\right)$ times:
5. Sample $\mathbf{T} \subset \mathbf{S}$ uniformly at random of size $\left\lceil \frac{\sqrt{n}}{2^s} \right\rceil$.
6. Sample $\boldsymbol{x} \in \{0,1\}^n$ uniformly at random and run AE-SEARCH$(f, \boldsymbol{x}, \mathbf{T})$
7. Let $\mathbf{A}_+$ be the set of $i \in [n]$ such that a monotone edge along variable i is found
8. Repeat $O\left(\sqrt{n}\Lambda^{14}/\varepsilon^2\right)$ times:
9. Sample $\mathbf{T} \subset \mathbf{S}$ uniformly at random of size $\left\lceil \frac{\sqrt{n}}{2^t} \right\rceil$.
10. Sample an $\boldsymbol{x} \in \{0,1\}^n$ uniformly at random and run AE-SEARCH$(f, \boldsymbol{x}, \mathbf{T})$
11. Let $\mathbf{A}_-$ be the set of $i \in [n]$ such that an anti-monotone edge along variable i is found
12. Output an edge violation of f to unateness if one is found in $\mathbf{B}$, $\mathbf{A}_+$ and $\mathbf{A}_-$.
13. Output "unate."

Figure 7.12: Algorithm for Case 2

satisfies Lemma 7.3.6 with parameters s, t, h, ℓ on a set $\mathcal{I} \subset [n]$ of size $|\mathcal{I}| = 2^\ell$. Every variable $i \in \mathcal{I}$ satisfies

$$\textsc{Score}^+_{i,s} \geq 2^{s-h} \qquad \textsc{Score}^-_{i,t} \geq 2^{t-h}, \qquad \text{and} \qquad \frac{|\mathcal{I}|}{2^h} = \Omega\left(\frac{\varepsilon^2}{\Lambda^{11}}\right). \tag{7.25}$$

We assume

$$|\mathcal{I}| \cdot \frac{1}{2^s} \geq |\mathcal{I}| \cdot \frac{1}{2^t} \geq n^{2/3}, \tag{7.26}$$

and that for al least half of the variables $i \in \mathcal{I}$,

$$\mathbf{Inf}_f(i) \geq \alpha := \frac{\varepsilon^2 \cdot n^{1/3}}{|\mathcal{I}| \cdot \Lambda^{13}}. \tag{7.27}$$

Similarly to case 1 in Section 7.5, we consider the subset $\mathcal{I}^* \subseteq \mathcal{I}$ of size at least $\lceil \frac{|\mathcal{I}|}{2} \rceil$ satisfying (7.27) for all $i \in \mathcal{I}^*$.

This algorithm `AlgorithmCase2`, show in Figure 7.12, finds an edge violation with high probability. At a high level, the algorithm first samples a uniformly random set $\mathbf{S}$ of siz e $n^{2/3}$ and uses `Find-Hi-Inf` to find bichromatic edges along almost all variables in $\mathbf{S} \cap \mathcal{I}^*$. Suppose first, that most of the bichromatic edges from $\mathbf{S} \cap \mathcal{I}^*$ found during `Find-Hi-Inf` are anti-monotone edges. Then via a similar analysis to [52], after running AE-SEARCH$(f, \boldsymbol{x}, \mathbf{S})$ on a uniform random sets $\mathbf{T} \subset \mathbf{S}$ of size $\lceil \sqrt{n}/2^s \rceil$ and a uniform point $\boldsymbol{x} \sim \{0,1\}^n$ for $\widetilde{O}(\sqrt{n}/\varepsilon^2)$ many iterations, we expect to find a monotone edge along some direction in $\mathbf{S} \cap \mathcal{I}^*$. Since the algorithm had already found an anti-monotone edge along many variables in $\mathbf{S} \cap \mathcal{I}^*$, the algorithm will very likely find a violation to unateness. Similarly, if most of the bichromatic edges along variables in $\mathbf{S} \cap \mathcal{I}^*$ during `Find-Hi-Inf` are monotone, then AE-SEARCH$(f, \boldsymbol{x}, \mathbf{T})$ will likely find an edge violation when $\mathbf{T}$ is a random set of $\mathbf{S}$ of size $\lceil \sqrt{n}/2^t \rceil$ after $\tilde{O}(\sqrt{n}/\varepsilon^2)$ iterations.

Lemma 7.7.1 (Query complexity of `AlgorithmCase2`)**.** `AlgorithmCase2`(f) *makes*

$$\tilde{O}\left(n^{2/3}/\varepsilon^2\right)$$

queries to f.

Proof. The query complexity of `AlgorithmCase2`(f) follows from the description in Figure 7.12. In particular, we lines 4–12 make a total of $\tilde{O}(\sqrt{n}/\varepsilon^2)$, so it remains to upper-bound the query complexity of line 3 invoking `Find-Hi-Inf`. By Lemma 7.6.8, `Find-Hi-Inf`$(f, \mathbf{S}, k, \alpha)$, where

$$|\mathbf{S}| = n^{2/3} \qquad \text{and} \qquad k = \lceil \frac{|\mathcal{I}^*|}{n^{1/3} \log n} \rceil,$$

makes

$$\tilde{O}\left(|\mathbf{S}| + \frac{|\mathbf{S}|}{\alpha k}\right) = \tilde{O}\left(n^{2/3}/\varepsilon^2\right),$$

using the assumptions of Case 2 in (7.25), (7.26), and (7.27). □

We start the analysis of `AlgorithmCase2` by defining the notion of *informative sets* for case 2. Recall that, in Section 7.5, Definition 61 gave a different definition of informative sets for case 1. Since these definitions serve very similar purposes in the analysis of the algorithm, we use the same name. Furthermore, for $T \subset [n]$ of size $\lceil \frac{\sqrt{n}}{2^s} \rceil$, $\text{PE}_i^+(T)$ is the set of s-strong monotone edges along variable i which are T-persistent. Similarly, for $T \subset [n] \setminus \{i\}$ of size $\lceil \frac{\sqrt{n}}{2^t} \rceil$, $\text{PE}_i^-(T)$ is the set of t-strong anti-monotone edges along variable i that are T-persistent. Note that the definitions of these sets are slightly different than in Subsection 7.5.1.

Definition 66. *We say that a set $S \subset [n] \setminus \{i\}$ of size $(n^{2/3} - 1)$ is i-informative if the following two conditions hold:*

i. with probability at least $1/10$ over the draw of $\mathbf{T} \subset S$ of size $\lceil \sqrt{n}/2^s \rceil$, $|PE_i^+(\mathbf{T})| \geq \frac{2^{s-h}}{10} \cdot 2^n$.

ii. with probability at least $1/10$ over the draw of $\mathbf{T} \subset S$ of size $\lceil \sqrt{n}/2^t \rceil$, $|PE_i^-(\mathbf{T}) \geq \frac{2^{t-h}}{10} \cdot 2^n$.

Lemma 7.7.2. *For every $i \in \mathcal{I}^*$, when sampling $\mathbf{S} \subset [n]$ uniformly of size $n^{2/3}$, we have*

$$\Pr_{\mathbf{S} \subset [n]}[i \in \mathbf{S} \text{ and } \mathbf{S} \setminus \{i\} \text{ is } i\text{-informative}] \geq \frac{1}{2n^{1/3}}.$$

Proof. Recall that for $m \in \mathbb{N}$, $\mathcal{P}_{i,m}$ is the uniform distribution over subsets of $[n] \setminus \{i\}$ of size $m - 1$. For $m = n^{2/3}$, we define the quantity:

$$\gamma = \Pr_{\mathbf{S} \sim \mathcal{P}_{i,m}}[\mathbf{S} \text{ does not satisfy (i) in Definition 66}].$$

For $m_1 = \left\lceil \frac{\sqrt{n}}{2^s} \right\rceil \leq m$, consider the quantity

$$\Pr_{\boldsymbol{e}, \mathbf{T}}\left[\boldsymbol{e} \in \text{PE}_i^+(\mathbf{T})\right],$$

where $\boldsymbol{e}$ is an s-strong monotone edge sampled uniformly at random, and $\mathbf{T} \sim \mathcal{P}_{i,m_1}$. By the definition of s-strong monotone edges, we have the above probability is at least $1 - o(1)$. On the other hand, we may use the definition of γ to upper bound the above probability by $\gamma(\frac{1}{10} + \frac{9}{10} \cdot \frac{1}{10}) + (1 - \gamma)$, which implies $\gamma = o(1)$. Analogously, letting $m_2 = \lceil \frac{\sqrt{n}}{2^t} \rceil$, we define the quantity:

$$\beta = \mathop{\mathbf{Pr}}_{\mathbf{S} \sim \mathcal{P}_{i,m_2}} [\mathbf{S} \text{ does not satisfy (ii) in Definition 66}] .$$

Similarly as above, we considering $\mathbf{Pr}[\boldsymbol{e} \in \mathrm{PE}_i^-(\mathbf{T})]$, for $\mathbf{T} \sim \mathcal{P}_{i,m_2}$ and $\boldsymbol{e}$ a uniformly random t-strong anti-monotone edge, to conclude $\beta = o(1)$. Therefore, the probability over $\mathbf{S} \sim \mathcal{P}_{i,m}$ that $\mathbf{S}$ is i-informative is at least $1 - \gamma - \beta \geq 1 - o(1)$. Finally,

$$\mathop{\mathbf{Pr}}_{\mathbf{S} \subset [n]} [i \in \mathbf{S} \text{ and } \mathbf{S} \setminus \{i\} \text{ is } i\text{-informative}] \geq \mathop{\mathbf{Pr}}_{\mathbf{S} \subset [n]} [i \in \mathbf{S}] \cdot \mathop{\mathbf{Pr}}_{\mathbf{S}' \sim \mathcal{P}_{i,m}} [\mathbf{S}' \text{ is } i\text{-informative}] \geq \frac{1}{2n^{1/3}}.$$

□

When sampling a set $\mathbf{S} \subset [n]$ of size $n^{2/3}$ in line 2 of $\texttt{AlgorithmCase2}(f)$, we may define the set

$$\boldsymbol{\mathcal{J}} = \{i \in \mathcal{I}^* : i \in \mathbf{S} \text{ and } \mathbf{S} \setminus \{i\} \text{ is } i\text{-informative}\} . \tag{7.28}$$

By Lemma 7.7.2, we immediately obtain

$$\mathop{\mathbf{E}}_{\mathbf{S} \subset [n]} [|\boldsymbol{\mathcal{J}}|] \geq \frac{|\mathcal{I}^*|}{2n^{1/3}} = \Omega\left(\frac{2^h \cdot \varepsilon^2}{n^{1/3} \cdot \Lambda^{11}}\right), \tag{7.29}$$

where the second inequality follows from (7.25). We consider the event $\boldsymbol{\mathcal{E}}$, defined over the randomness of sampling $\mathbf{S}$ in line 2, which occurs when $|\mathbf{S} \cap \mathcal{I}^*| \leq \frac{4|\mathcal{I}^*|}{n^{1/3}}$ and $|\boldsymbol{\mathcal{J}}| \geq \frac{|\mathcal{I}^*|}{10n^{1/3}}$.

Lemma 7.7.3. *In every iteration of line 2 of* $\texttt{AlgorithmCase2}(f)$*, event* $\boldsymbol{\mathcal{E}}$ *occurs with probability at least* $\Omega(1)$.

Proof. By (7.26) and the fact that $\frac{1}{2^s} \leq 1$, $\mathbf{E}_{\mathbf{S}}[|\mathbf{S} \cap \mathcal{I}^*|] \geq |\mathcal{I}^*|/n^{1/3} \geq n^{1/3}$. Thus, by Lemma 7.11.1, $|\mathbf{S} \cap \mathcal{I}^*| \leq 4|\mathcal{I}^*|/n^{1/3}$ with probability at least $1 - \exp\left(-\Omega(n^{1/3})\right)$. Let

β be the probability over $\mathbf{S} \subset [n]$ that $|\boldsymbol{\mathcal{J}}| \geq |\mathcal{I}^*|/(10n^{1/3})$. We may then upper bound $\mathbf{E}[|\boldsymbol{\mathcal{J}}|]$ using the definition of β by

$$\exp(-\Omega(n^{1/3})) \cdot n^{2/3} + \beta \cdot \frac{4|\mathcal{I}^*|}{n^{1/3}} + (1-\beta) \cdot \frac{|\mathcal{I}^*|}{10n^{1/3}},$$

which, combined with the lower bound in (7.29), implies $\beta \geq \Omega(1)$. Thus, the probability $\boldsymbol{\mathcal{E}}$ occurs is at least $\beta - \exp(-\Omega(n^{1/3})) = \Omega(1)$. □

Thus, consider some iteration of `AlgorithmCase2`(f) where event $\boldsymbol{\mathcal{E}}$ occurs, and by Lemma 7.7.3 there exists such an iteration with high constant probability. The rest of this section is devoted to showing that the iteration of `AlgorithmCase2`(f) where $\boldsymbol{\mathcal{E}}$ occurs will find a violation to unateness with high probability.

Lemma 7.7.4. *Suppose that a particular iteration of* `AlgorithmCase2`(f)*, event* $\boldsymbol{\mathcal{E}}$ *occurs. At that iteration,* `AlgorithmCase2`(f) *finds a violation with high probability.*

Proof. Fix the particular iteration where event $\boldsymbol{\mathcal{E}}$ occurs. We apply Lemma 7.6.8 with the hidden set $H = \boldsymbol{\mathcal{J}}$ to conclude that at line 3 of `AlgorithmCase2`, the set of variables $\mathbf{Q}$ for which bichromatic edges are observed during `Find-Hi-Inf` satisfies

$$|\mathbf{Q} \cap \boldsymbol{\mathcal{J}}| \geq |\boldsymbol{\mathcal{J}}| - k \geq \frac{|\mathcal{I}^*|}{20n^{1/3}}, \tag{7.30}$$

with probability at least $1 - 1/\mathrm{poly}(n)$, where in the last inequality, we used the fact that $k \leq |\boldsymbol{\mathcal{J}}|/2$, as well as the fact that $|\boldsymbol{\mathcal{J}}| \geq |\mathcal{I}^*|/(10n^{1/3})$ when $\boldsymbol{\mathcal{E}}$ occurs. Assume that for most of the variables in $\mathbf{Q} \cap \boldsymbol{\mathcal{J}}$, $\mathbf{B}$ contains *anti-monotone* edges in these variables, and let $\mathbf{C} \subset \mathbf{Q} \cap \boldsymbol{\mathcal{J}}$ be the set of these variables, which by assumption,

$$|\mathbf{C}| \geq \frac{|\mathcal{I}|}{40n^{1/3}}. \tag{7.31}$$

The case when most variables in $\mathbf{Q} \cap \boldsymbol{\mathcal{J}}$ contain monotone edges will follow by a symmetric argument. We will now show that during the execution of lines 4–7 of `AlgorithmCase2`(f), $\mathbf{A}_+$ will contain a monotone edge along some variable in $\mathbf{C}$ with high probability.

Towards this goal, consider a particular execution of line 5 which samples a set $\mathbf{T} \subset \mathbf{S}$ of size $m_1 = \left\lceil \frac{\sqrt{n}}{2^s} \right\rceil$ uniformly at random, and let

$$\mathbf{D} = \left\{ i \in \mathbf{C} \cap \mathbf{T} : \frac{|\mathrm{PE}_i^+(\mathbf{T} \setminus \{i\})|}{2^n} \geq \frac{2^{s-h}}{10} \right\}.$$

We note that since $\mathbf{C} \subset \boldsymbol{\mathcal{J}}$, every $i \in \mathbf{C}$ satisfies

$$\begin{aligned} \mathop{\mathbf{Pr}}_{\mathbf{T} \subset \mathbf{S}}[i \in \mathbf{D}] &= \mathop{\mathbf{Pr}}_{\mathbf{T} \subset \mathbf{S}}[i \in \mathbf{T}] \cdot \mathop{\mathbf{Pr}}_{\mathbf{T}' \sim \mathcal{P}_{i,m_1}} \left[\frac{|\mathrm{PE}_i^+(\mathbf{T}')|}{2^n} \geq \frac{2^{s-h}}{10} \right] \\ &\geq \frac{m_1}{n^{2/3}} \cdot \frac{1}{10} \geq \frac{1}{10 \cdot 2^s \cdot n^{1/6}}. \end{aligned}$$

which implies that the parameter

$$\beta = \mathop{\mathbf{E}}_{\mathbf{T} \subset \mathbf{S}}[|\mathbf{D}|] \geq |\mathbf{C}| \cdot \frac{1}{10 \cdot 2^s \cdot n^{1/6}} \geq \frac{|\mathcal{I}^*|}{400 \cdot 2^s \cdot \sqrt{n}} \tag{7.32}$$

by (7.31), and similarly, note that $\mathbf{E}[|\mathbf{C} \cap \mathbf{T}|] = \Theta(\beta)$.

Suppose first that $\beta \leq \log^2 n$, so that $\mathbf{E}[|\mathbf{C} \cap \mathbf{T}|] = O(\log^2 n)$. Then, by Lemma 7.11.1, $|\mathbf{D}| \leq O(\log^2 n)$ with probability at least $1 - \exp\left(-\Omega(\log^2 n)\right)$. This implies that during the execution of line 5 and 6,

$$\mathop{\mathbf{Pr}}_{\substack{\mathbf{T} \subset \mathbf{S} \\ \boldsymbol{x} \sim \{0,1\}^n}} \left[\exists i \in \mathbf{D} \text{ and } \boldsymbol{x} \in \mathrm{PE}_i^+(\mathbf{T} \setminus \{i\}) \right] \geq \mathop{\mathbf{Pr}}_{\mathbf{T} \subset \mathbf{S}}[|\mathbf{D}| \geq 1] \cdot \frac{2^{s-h}}{10} = \Omega\left(\frac{\beta}{\log^2 n} \cdot \frac{2^{s-h}}{10} \right) \tag{7.33}$$

$$\geq \Omega\left(\frac{1}{\log^2 n} \cdot \frac{|\mathcal{I}^*|}{400 \cdot 2^s \cdot \sqrt{n}} \cdot \frac{2^{s-h}}{10} \right) \tag{7.34}$$

$$\geq \Omega\left(\frac{|\mathcal{I}^*|}{2^h \cdot \sqrt{n} \cdot \log^2 n} \right) \geq \Omega\left(\frac{\varepsilon^2}{\Lambda^{13}\sqrt{n}} \right). \tag{7.35}$$

We note (7.33) follows from the fact that $i \in \mathbf{D}$ implies $\mathrm{PE}_i^+(\mathbf{T} \setminus \{i\}) \geq 2^n \cdot \frac{2^{s-h}}{10}$ and the fact that $\mathbf{Pr}[|\mathbf{D}| \geq 1] = \Omega(\frac{\beta}{\log^2 n})$; (7.34) follows from (7.32); and, (7.35) follows from (7.25). Thus, at least one iteration of the $O(\frac{\sqrt{n} \cdot \Lambda^{14}}{\varepsilon^2})$ iterations of lines 5 and 6 will output a monotone edge in some variable in $\mathbf{C}$ with high probability.

Suppose that $\beta \geq \log^2 n$. In this case, with probability at least $1 - \exp\left(-\Omega(\log^2 n)\right)$,

$$|\mathbf{D}| \geq \frac{\beta}{4},$$

and when this occurs,

$$\Pr_{\boldsymbol{x}\sim\{0,1\}^n}\left[\exists i \in \mathbf{D} \text{ and } \boldsymbol{x} \in \mathrm{PE}_i^+\left(\mathbf{T} \setminus \{i\}\right)\right] \geq \frac{\beta}{4} \cdot \frac{2^{s-h}}{10} = \Omega\left(\frac{\varepsilon^2}{\Lambda^{11}\sqrt{n}}\right)$$

by (7.32) and (7.25). Thus, in this case again, we may conclude that at least one iteration of lines 5 and 6 will output a monotone edge from $\mathbf{C}$ with high probability. □

7.8 The Algorithm for Case 3

Below, we prove correctness of `AlgorithmCase3`(f), which covers Case 3 of the algorithm. We let $s \geq t \in [\Lambda]$, $h \in [3\Lambda]$, and $l \in [\lceil \log n \rceil]$ be parameters, so that $f\colon \{0,1\}^n \to \{0,1\}$ is ε-far from unate and satisfies Lemma 7.3.6 with s, t, h, ℓ, for a hidden set $\mathcal{I} \subset [n]$ of size $|\mathcal{I}| = 2^\ell$. Similarly to case 1 and case 2, we assume the parameters s, t, h, ℓ, and the set $\mathcal{I}$ satisfy that every $i \in \mathcal{I}$,

$$\textsc{Score}^+_{i,s} \geq 2^{s-h} \qquad \textsc{Score}^-_{i,t} \geq 2^{t-h} \qquad \text{and} \qquad \frac{|\mathcal{I}|}{2^h} = \Omega\left(\frac{\varepsilon^2}{\Lambda^{11}}\right). \tag{7.36}$$

Lastly, we assume that $\mathcal{I}$ is not too large, i.e.,

$$|\mathcal{I}| \cdot \frac{1}{2^s} \leq |\mathcal{I}| \cdot \frac{1}{2^t} \leq n^{2/3}. \tag{7.37}$$

Instantiating the algorithm of [52] with the additional assumptions corresponding to Case 3 from Section 7.4 would give the desired upper bound on the query complexity. Specifically, one may derive from (7.37), that $\mathbf{E}_{\mathbf{S}\subset[n]}\left[|\mathbf{S} \cap \mathcal{I}|\right] \leq n^{1/6}$, which corresponds to the parameter α in [52]. As a result of Fact 5.4 and Fact 5.12, the query complexity of the

Procedure `AlgorithmCase3`(f)

Input: Query access to a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$
Output: Either "unate," or two edges constituting an edge violation of f to unateness.

1. Repeat the following $O\left(\lceil \frac{2^t\sqrt{n}\log^2 n}{|\mathcal{I}|} \rceil\right)$ times:
2. Draw $\mathbf{T} \subset [n]$ of size $\lceil \frac{\sqrt{n}}{2^t} \rceil$ uniformly at random.
3. Repeat $O\left(\frac{2^h}{2^t}\log^2 n\right)$ times:
4. Sample $\boldsymbol{x} \in \{0,1\}^n$ uniformly at random and run AE-SEARCH$(f, \boldsymbol{x}, \mathbf{T})$
5. Let $\mathbf{A}$ be the set of $i \in [n]$ such that an anti-monotone edge along variable i is found.
6. Repeat $O\left(2^s/2^t\right)$ times:
7. Draw $\mathbf{S} \subset \mathbf{T}$ of size $\lceil \frac{\sqrt{n}}{2^s} \rceil$ uniformly at random.
8. Repeat $O\left(\frac{2^h}{2^s}\log^2 n\right)$ times:
9. Sample an $\boldsymbol{y} \in \{0,1\}^n$ uniformly at random and run AE-SEARCH$(f, \boldsymbol{y}, \mathbf{S})$
10. Let $\mathbf{B}$ be the set of $i \in [n]$ such that a monotone edge along variable i is found
11. Output an edge violation of f to unateness if one is found in $\mathbf{B}$.
12. Output "unate."

Figure 7.13: Algorithm for Case 3

algorithm in [52] is

$$\tilde{O}(\sqrt{\alpha n}/\varepsilon^2) \leq \tilde{O}(n^{7/12}/\varepsilon^2) \qquad \text{and} \qquad \tilde{O}(\sqrt{n}/\varepsilon^2),$$

which are $\tilde{O}(n^{2/3}/\varepsilon^2)$. However, there is a (minor) technical caveat in the different definitions for strong edges and SCORE in Definition 59 and the analogous definitions in [52]. For the sake of completeness, we include a simple algorithm achieving an $\tilde{O}(n^{2/3}/\varepsilon^2)$-query upper bound in Figure 7.13.

Lemma 7.8.1 (Query complexity of `AlgorithmCase3`)**.** *`AlgorithmCase3`(f) makes*

at most

$$O\left(\left\lceil \frac{2^t\sqrt{n}\log^2 n}{|\mathcal{I}|}\right\rceil\right)\left(\tilde{O}\left(\frac{2^h}{2^t}\right)+O\left(\frac{2^s}{2^t}\right)\cdot\tilde{O}\left(\frac{2^h}{2^s}\right)\right)=\tilde{O}(\sqrt{n}/\varepsilon^2)+\tilde{O}(n^{2/3}).$$

queries to f

Proof. The query complexity upper bound is divided into two cases. If $2^t\sqrt{n}\log^2 = \Omega(|\mathcal{I}|)$, then the query complexity is $\tilde{O}\left(\sqrt{n}/\varepsilon^2\right)$. Otherwise, the query complexity is $\tilde{O}(\frac{2^h}{2^t}) = \tilde{O}(|\mathcal{I}|/2^t)$ and the bound follows from (7.37). □

We will use the definition of i-informative for monotone and anti-monotone edges given in Definition 61. The following lemma simply follows from applying Lemma 7.5.3 and Lemma 7.5.4, and taking a union bound.

Claim 7.8.2. *With probability $1-o(1)$ over the draw of $\mathbf{T}$ and $\mathbf{S}$ in lines 2 and 3 conditioned on $i \in \mathbf{S} \subset \mathbf{T}$, $\mathbf{S}\setminus\{i\}$ is i-informative for monotone edges and $\mathbf{T}\setminus\{i\}$ is i-informative for anti-monotone edges.*

Furthermore, assuming that $i \in \mathbf{S} \subset \mathbf{T}$ is sampled in line 2 and 3, where $\mathbf{T}\setminus\{i\}$ is i-informative for anti-monotone edges, and $\mathbf{S}\setminus\{i\}$ is i-informative for monotone edges, it follows that there exists two sets of points X_i and Y_i of size $\Omega(\frac{2^t}{2^h})\cdot 2^n$ and $\Omega(\frac{2^s}{2^h})\cdot 2^n$, respectively, such that if $\boldsymbol{x} \sim X_i$ and $\boldsymbol{y} \sim Y_i$ are sampled in lines 5 and 8, an edge violation is found with probability at least $\Omega(1)$. Since lines 5 and 8 are repeated sufficiently many times, such points $\boldsymbol{x}$ and $\boldsymbol{y}$ will be sampled from X_i and Y_i, respectively. Therefore, since $\mathbf{S}$ is sampled at least $\tilde{O}(\max\{2^s/2^t, 2^s\sqrt{n}/|\mathcal{I}|\})$ times, it suffices to prove the following claim.

Claim 7.8.3. *With probability $\Omega(\max\{1, \frac{|\mathcal{I}|}{2^s\sqrt{n}\log^2 n}\})$ over the draw of $\mathbf{T}$ and $\mathbf{S}$ in lines 2 and 3, $\mathbf{S}\cap\mathcal{I}\neq\emptyset$.*

Proof. Let γ be the probability over $\mathbf{S}\subset[n]$ that $\mathbf{S}\cap\mathcal{I}\neq\emptyset$, which we will lower bound in the remainder of the proof. Consider the quantity

$$\mathop{\mathbf{E}}_{\mathbf{S}\subset[n]}\left[|\mathbf{S}\cap\mathcal{I}|\right],$$

Subroutine AE-SEARCH(f, x, S)

Input: Query access to $f\colon \{0,1\}^n \to \{0,1\}$, $x \in \{0,1\}^n$, and a nonempty set $S \subseteq [n]$.
Output: Either a variable $i \in S$ with $f(x^{(i)}) \neq f(x)$, or "fail."

1. Query $f(x)$ and set $b \leftarrow f(x)$.
2. Draw $L = \lceil 4 \log n \rceil$ subsets $\mathbf{T}_1, \ldots, \mathbf{T}_L \subseteq S$ of size $t = \lfloor (|S| - 1)/2 \rfloor + 1$ uniformly.
3. Query $f(x^{(\mathbf{T}_\ell)})$ and set the output to be $\boldsymbol{b}_\ell$ for each $\ell \in [L]$. Let $\mathbf{C} \subseteq S$ where
$$\mathbf{C} = \bigcap_{\ell \in [L]\colon \boldsymbol{b}_\ell \neq b} \mathbf{T}_\ell \qquad (\mathbf{C} = \emptyset \text{ by default if } \boldsymbol{b}_\ell = b \text{ for all } \ell).$$
4. If $\mathbf{C} = \{\boldsymbol{i}\}$ for some $\boldsymbol{i}$, query $f(x^{(\boldsymbol{i})})$ and return $\boldsymbol{i}$ if $f(x^{(\boldsymbol{i})}) \neq b$; otherwise return "fail."

Figure 7.14: Description of the adaptive edge search subroutine.

and note that since $\mathbf{S}$ is a uniform random subset of $[n]$ of size $\lceil \sqrt{n}/2^s \rceil$, the above expectation is at least $|\mathcal{I}|/(2^s\sqrt{n})$. By Lemma 7.11.1, $|\mathbf{S} \cap \mathcal{I}| \leq 4 \max\{\log^2 n, |\mathcal{I}|/(2^s\sqrt{n})\}$ with probability at least $1 - \exp\left(-\Omega(\log^2 n)\right)$. As a result, we may upper bound the above expectation using the definition of γ by

$$n \exp\left(-\Omega(\log^2 n)\right) + \gamma \cdot 4 \max\{\log^2 n, |\mathcal{I}|/(2^s\sqrt{n})\}$$

which gives the desired lower bound on γ. □

7.9 Adaptive Edge Search

For completeness we present the adaptive edge search algorithm, AE-SEARCH in Figure 7.14, which first appeared in [52]. We recall the lemma and include its proof below.

Lemma 7.9.1. *Given a point $x \in \{0,1\}^n$ and a set $S \subseteq [n+1]$,* AE-SEARCH(f, x, S) *makes $O(\log n)$ queries to f, and returns either an $i \in S$ such that $(x, x^{(i)})$ is a bichromatic edge, or "fail."*

Let $(x, x^{(i)})$ be a bichromatic edge of f along i. If $i \in S$ and $(x, x^{(i)})$ is $(S \setminus \{i\})$-persistent, then both AE-SEARCH(f, x, S) *and* AE-SEARCH$(f, x^{(i)}, S)$ *output i with probability at least $2/3$.*

The first part of the lemma follows directly from the description of AE-SEARCH. For the second part, we prove it for AE-SEARCH(f, x, S) since the proof for AE-SEARCH$(f, x^{(i)}, S)$ is symmetric. The proof proceeds by exactly the same as Claim 6.6 and 6.7 of [52], except for some minor notational differences. We present a proof of the claims but adapted to the notation of this chapter.

Claim 7.9.2. *Let $(x, x^{(i)})$ be a bichromatic edge and $i \in S \subseteq [n+1]$. If $(x, x^{(i)})$ is $(S \setminus \{i\})$-persistent, then, in line 3 of* AE-SEARCH(f, x, S)*, $i \in \mathbf{C}$ with probability at least $1 - o(1)$.*

Proof. Let $\mathbf{T}_1, \ldots, \mathbf{T}_L \subseteq S$ be subsets sampled in line 2 of AE-SEARCH. The parameter t satisfies

$$t = \left\lfloor \frac{|S| - 1}{2} \right\rfloor + 1 \geq \frac{|S|}{2}.$$

We note that there are two events where $i \notin \mathbf{C}$: (1) either all $\mathbf{T}_\ell$ satisfy $f(x^{(\mathbf{T}_\ell)}) = b$, or (2) there is an $\ell \in [L]$ with $i \notin \mathbf{T}_\ell$ and $f(x^{(\mathbf{T}_\ell)}) \neq b$. We show that the probability of either event occurring is at most $o(1)$, so the claim follows by a union bound.

For the first event, a single sample of a random set $\mathbf{T} \subseteq S$ of size t satisfies

$$\Pr_{\substack{\mathbf{T} \subseteq S \\ |\mathbf{T}| = t}} \left[f(x^{(\mathbf{T})}) \neq b \right] \geq \Pr_{\substack{\mathbf{T} \subseteq S \\ |\mathbf{T}| = t}} \left[i \in \mathbf{T} \right] \cdot \Pr_{\substack{\mathbf{T}' \subseteq S \setminus \{i\} \\ |\mathbf{T}'| = t-1}} \left[f(x^{(i \cup \mathbf{T}')}) \neq b \right] \geq \frac{t}{|S|} \cdot \left(1 - \frac{1}{\log^2 n} \right) \geq \frac{1}{2} - o(1),$$

where we used the fact that $f(x^{(i)}) \neq b$ since $(x, x^{(i)})$ is bichromatic, and the assumption that $x^{(i)}$ is $(S \setminus \{i\})$-persistent. Thus the probability that all $\mathbf{T}_\ell$ satisfy $f(x^{(\mathbf{T}_\ell)}) = b$ is $(0.5 + o(1))^L = o(1)$.

For the second event, using the assumption that x is $(S \setminus \{i\})$-persistent, we have that

$$\Pr_{\mathbf{T}_1, \ldots, \mathbf{T}_L} \left[\exists \ell \in [L] : f(x^{(\mathbf{T}_\ell)}) \neq b \text{ and } i \notin \mathbf{T}_\ell \right] \leq L \cdot \Pr_{\substack{\mathbf{T} \subseteq S \\ |\mathbf{T}| = t}} \left[f(x^{(\mathbf{T})}) \neq b \,\middle|\, i \notin \mathbf{T} \right] \leq \frac{L}{\log^2 n} = o(1).$$

This finishes the proof of the claim. □

Claim 7.9.3. *Let $(x, x^{(i)})$ be a bichromatic edge and $i \in S \subseteq [n+1]$. If $(x, x^{(i)})$ is $(S \setminus \{i\})$-persistent, then in line 3 of* AE-SEARCH(f, x, S), $\mathbf{C}$ *does not contain any $j \neq i$ with probability at least $1 - o(1)$.*

Proof. Fix a $j \in S$ but $j \neq i$. Note that in order for $j \in \mathbf{C}$, every $\ell \in [L]$ with $f(x^{(\mathbf{T}_\ell)}) \neq b$ satisfies $j \in \mathbf{T}_\ell$. However, since $x^{(i)}$ is $(S \setminus \{i\})$-persistent we have

$$\begin{aligned}\Pr_{\substack{\mathbf{T}\subseteq S\\|\mathbf{T}|=t}}\left[j \notin \mathbf{T} \text{ and } f(x^{(\mathbf{T})}) \neq b\right] &\geq \Pr_{\substack{\mathbf{T}\subseteq S\\|\mathbf{T}|=t}}\left[i \in \mathbf{T}\right] \cdot \Pr_{\substack{\mathbf{T}'\subseteq S\setminus\{i\}\\|\mathbf{T}'|=t-1}}\left[j \notin \mathbf{T}' \text{ and } f(x^{(\mathbf{T}'\cup\{i\})}) \neq b\right] \\ &\geq \frac{t}{|S|} \cdot \left(1 - \frac{t-1}{|S|-1} - \frac{1}{\log^2 n}\right) \geq \frac{1}{4} - o(1).\end{aligned}$$

Therefore, $j \in \mathbf{C}$ with probability $(3/4 + o(1))^L = o(1/n)$. The claim follows from a union bound. □

It follows by combining these two claims that $\mathbf{C} = \{i\}$ in line 4 with probability at least $1 - o(1)$. This finishes the proof of the lemma.

7.10 Analysis of the Preprocessing Procedure

We start with the following property of `CheckPersistence`:

Claim 7.10.1. *Given a nonempty set $S \subseteq [n+1]$, an ordering π of S and $\xi \in (0,1)$,* `CheckPersistence` (f, S, π, ξ) *makes $O(\log^5 n/\xi)$ many queries to f. Furthermore, if the fraction of points that are not S-persistent is at least ξ,* `CheckPersistence`(f, S, π, ξ) *returns a variable $i \in S$ with probability at least $1 - \exp(-\Omega(\log^2 n))$.*

Proof. The first part of the claim follows from the description of `CheckPersistence`. For the second part we note that if the fraction of points that are not S-persistent is at least ξ, then the probability of $\boldsymbol{x}$ and $\mathbf{T}$ with $f(\boldsymbol{x}) \neq f(\boldsymbol{x}^{(\mathbf{T})})$ is $\Omega(\xi/\log^2 n)$. It then follows from the number of times we repeat in `CheckPersistence`. □

We recall the lemma we need for `Preprocess`:

Lemma 7.10.2. *Given a Boolean function $f\colon \{0,1\}^n \to \{0,1\}$, a nonempty $S_0 \subseteq [n+1]$, an ordering π of S_0 and a parameter $\xi \in (0,1)$,* `Preprocess`(f, S_0, π, ξ) *makes at most $O(|S_0| \log^5 n/\xi)$ queries to f and with probability at least $1 - \exp\left(-\Omega(\log^2 n)\right)$, it returns a subset $\mathbf{S} \subseteq S_0$ such that at least $(1-\xi)$-fraction of points in $\{0,1\}^n$ are $\mathbf{S}$-persistent.*

Proof. The query complexity follows from the fact that `Preprocess` makes at most $|S_0|$ many calls to `CheckPersistence`. In addition, for each call, it follows from Claim 7.10.1 that the probability of `CheckPersistence` returning `nil` while S is actually persistent over less than $(1-\xi)$-fraction of points is at most $\exp(-\Omega(\log^2 n))$. The lemma follows from a union bound over $|S_0| \le n$ calls. □

7.11 Overlap of Two Random Sets of Certain Sizes

Let $k, \ell \in [n]$ be two positive integers with $\alpha = k\ell/n$. We are interested in the size of $|\mathbf{S} \cap \mathbf{T}|$ where $\mathbf{S}$ is a random k-sized subset of $[n]$ and $\mathbf{T}$ is a random ℓ-sized subset of $[n]$, both drawn uniformly.

Lemma 7.11.1. *For any $t \ge 4\alpha$, the probability of $|\mathbf{S} \cap \mathbf{T}| \ge t$ is at most $\exp(-\Omega(t))$.*

Proof. We assume without loss of generality that $k \ge \ell$. If $\ell > n/2$, the claim is trivial as $\alpha > n/4$ and $t \ge 4\alpha > n$. We assume $\ell \le n/2$ below.

We consider the following process. We draw $\mathbf{S}$ first. Then we add random (and distinct) indices of $[n]$ to $\mathbf{T}$ round by round for ℓ rounds. In each round we pick an index uniformly at random from those that have not been added to $\mathbf{T}$ yet. Clearly this process generates the same distribution of $\mathbf{S}$ and $\mathbf{T}$ that we are interested in.

For each $i \in [\ell]$, we let $\mathbf{X}_i$ be the random variable that is set to 1 if the index in the ith round belongs to $\mathbf{S}$ and is 0 otherwise. Although $\mathbf{X}_i$'s are not independent, the probability of $\mathbf{X}_i = 1$ is at most $k/(n-\ell) \le 2k/n$ using $\ell \le n/2$, for any fixed values of $\mathbf{X}_1, \ldots, \mathbf{X}_{i-1}$. Thus, the expectation of $\sum_{i\in[\ell]} \mathbf{X}_i$ is at most $2k\ell/n = 2\alpha$. The lemma follows directly from the Chernoff bound (together with a standard coupling argument). □

Bibliography

[1] José A Adell and Pedro Jodrá. "Exact Kolmogorov and total variation distances between some familiar discrete distributions." In: *Journal of Inequalities and Applications* 2006.1 (2006), pp. 1–8.

[2] Thomas D. Ahle, Rasmus Pagh, Ilya Razenshteyn, and Francesco Silvestri. "On the Complexity of Inner Product Similarity Join." In: *Proceedings of the 35th ACM Symposium on Principles of Database Systems (PODS '2016)*. Available as arXiv:1510.02824. 2016, pp. 151–164.

[3] Noga Alon, Yossi Matias, and Mario Szegedy. "The space complexity of approximating the frequency moments." In: *Journal of Computer and System Sciences* 58.1 (1999), pp. 137–147.

[4] Noga Alon and Vitaly D. Milman. "$\lambda 1$, isoperimetric inequalities for graphs, and superconcentrators." In: *Journal of Combinatorial Theory, Series B* 38.1 (1985), pp. 73–88.

[5] Alexandr Andoni. "Nearest Neighbor Search: the Old, the New, and the Impossible." PhD thesis. MIT, 2009.

[6] Alexandr Andoni and Piotr Indyk. "Near-Optimal Hashing Algorithms for Approximate Nearest Neighbor in High Dimensions." In: *Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS '2006)*. 2006, pp. 459–468.

[7] Alexandr Andoni, Piotr Indyk, and Robert Krauthgamer. "Earth mover distance over high-dimensional spaces." In: *Proceedings of the 19th ACM-SIAM Symposium on Discrete Algorithms (SODA '2008)*. 2008, pp. 343–352.

[8] Alexandr Andoni, Piotr Indyk, and Robert Krauthgamer. "Overcoming the ℓ_1 Non-Embeddability Barrier: Algorithms for Product Metrics." In: *Proceedings of the 20th ACM-SIAM Symposium on Discrete Algorithms (SODA '2009)*. 2009, pp. 865–874.

[9] Alexandr Andoni, Piotr Indyk, and Ilya Razenshteyn. “Approximate nearest neighbor search in high dimensions.” In: *Proceedings of the International Congress of Mathematicians (ICM ’2018)*. 2018.

[10] Alexandr Andoni and Ilya Razenshteyn. “Optimal Data-Dependent Hashing for Approximate Near Neighbors.” In: *Proceedings of the 47th ACM Symposium on the Theory of Computing (STOC ’2015)*. Available as arXiv:1501.01062. 2015, pp. 793–801.

[11] Alexandr Andoni and Ilya Razenshteyn. “Tight Lower Bounds for Data-Dependent Locality-Sensitive Hashing.” In: *Proceedings of the 32nd International Symposium on Computational Geometry (SoCG ’2016)*. Available as arXiv:1507.04299. 2016, 9:1–9:11.

[12] Alexandr Andoni, Huy L. Nguyen, Aleksandar Nikolov, Ilya Razenshteyn, and Erik Waingarten. “Approximate near neighbor for general symmetric norms.” In: *Proceedings of the 50th ACM Symposium on the Theory of Computing (STOC ’2017)*. 2017.

[13] Alexandr Andoni, Piotr Indyk, Huy L. Nguyen, and Ilya Razenshteyn. “Beyond Locality-Sensitive Hashing.” In: *Proceedings of the 25th ACM-SIAM Symposium on Discrete Algorithms (SODA ’2014)*. Available as arXiv:1306.1547. 2014, pp. 1018–1028.

[14] Alexandr Andoni, Assaf Naor, Aleksandar Nikolov, Ilya Razenshteyn, and Erik Waingarten. “Data-dependent Hashing via Non-linear Spectral Gaps.” In: *Proceedings of the 50th ACM Symposium on the Theory of Computing (STOC ’2018)*. 2018.

[15] Alexandr Andoni, Khanh Do Ba, Piotr Indyk, and David Woodruff. “Efficient sketches for earth-mover distance, with applications.” In: *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS ’2009)*. 2009.

[16] Alexandr Andoni, Assaf Naor, Aleksandar Nikolov, Ilya Razenshteyn, and Erik Waingarten. “Hölder Homeomorphism and Approximate Nearest Neighbors.” In: *Proceedings of the 59th Annual IEEE Symposium on Foundations of Computer Science (FOCS ’2018)*. 2018.

[17] Alexandr Andoni, Thijs Laarhoven, Ilya Razenshteyn, and Erik Waingarten. “Lower Bounds on Time–Space Trade-Offs for Approximate Near Neighbors.” Available as arXiv:1605.02701. 2016.

[18] Alexandr Andoni, Thijs Laarhoven, Ilya Razenshteyn, and Erik Waingarten. "Optimal Hashing-based Time–Space Trade-offs for Approximate Near Neighbors." In: *Proceedings of the 28th ACM-SIAM Symposium on Discrete Algorithms (SODA '2017)*. Available as arXiv:1608.03580. 2017.

[19] Mark Anthony Armstrong. *Basic Topology*. Springer-Verlag, 1983.

[20] Sanjeev Arora, Elad Hazan, and Satyen Kale. "The multiplicative weights update method: a meta-algorithm and applications." In: *Theory of Computing* 8.1 (2012), pp. 121–164.

[21] Brian Babcock, Shivnath Babu, Mayur Datar, Rajeev Motwani, and Jennifer Widom. "Models and issues in data stream systems." In: *Proceedings of the 21st ACM Symposium on Principles of Database Systems (PODS '2002)*. 2002.

[22] Roksana Baleshzar, Deeparnab Chakrabarty, Ramesh Krishnan S. Pallavoor, Sofya Raskhodnikova, and C. Seshadhri. "A lower bound for nonadaptive, one-sided error testing of unateness of Boolean functions over the hypercube." In: *arXiv preprint arXiv:1706.00053* (2017).

[23] Roksana Baleshzar, Deeparnab Chakrabarty, Ramesh Krishnan S. Pallavoor, Sofya Raskhodnikova, and C. Seshadhri. "Optimal Unateness Testers for Real-Values Functions: Adaptivity Helps." In: *Proceedings of the 44th International Colloquium on Automata, Languages and Programming (ICALP '2017)*. 2017.

[24] Roksana Baleshzar, Meiram Murzabulatov, Ramesh Krishnan S. Pallavoor, and Sofya Raskhodnikova. "Testing unateness of real-valued functions." In: *arXiv preprint arXiv:1608.07652* (2016).

[25] Keith Ball. *An Elementary Introduction to Modern Convex Geometry*. Vol. 31. MSRI Publications. Cambridge University Press, 1997.

[26] Keith Ball, Eric A. Carlen, and Elliott H. Lieb. "Sharp uniform convexity and smoothness inequalities for trace norms." In: *Inventiones Mathematicae* 115.1 (1994), 463—482.

[27] Yair Bartal and Lee-Ad Gottlieb. "Approximate nearest neighbor search for ℓ_p-spaces $(2 < p < \infty)$ via embeddings." In: *Theoretical Computer Science* 757 (2019), pp. 27–35.

[28] Aleksandrs Belovs and Eric Blais. “A polynomial lower bound for testing monotonicity.” In: *Proceedings of the 48th ACM Symposium on the Theory of Computing (STOC ’2016)*. 2016, pp. 1021–1032.

[29] Yoav Benyamini and Joram Lindenstrauss. *Geometric Nonlinear Functional Analysis*. Vol. 1. American Mathematical Society, 1998.

[30] Yoav Benyamini and Joram Lindenstrauss. *Geometric Nonlinear Functional Analysis*. Vol. 48. American Mathematical Society, 2000.

[31] Jöran Bergh and Jörgen Löfström. *Interpolation Spaces*. Springer-Verlag, 1976.

[32] Eric Blais. “Improved bounds for testing juntas.” In: *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*. Springer, 2008, pp. 317–330.

[33] Eric Blais. “Testing juntas nearly optimally.” In: *Proceedings of the 41st ACM Symposium on the Theory of Computing (STOC ’2009)*. 2009, pp. 151–158.

[34] Eric Blais and Aleksandrs Belovs. “Quantum algorithm for monotonicity testing on the hypercube.” In: *Theory of Computing* 11.16 (2015), pp. 403–412.

[35] Jaroslaw Blasiok, Vladimir Braverman, Stephen R. Chestnut, and Robert Krauthgamerand Lin F. Yang. “Streaming symmetric norms via measure concentration.” In: *Proceedings of the 50th ACM Symposium on the Theory of Computing (STOC ’2017)*. 2017.

[36] Avrim Blum. *Relevant Examples and Relevant Features–Thoughts from Computational Learning Theory*. Tech. rep. AAAI Fall Symposium on Relevance, 1994.

[37] Avrim Blum and Pat Langley. “Selection of Relevant Features and Examples in Machine Learning.” In: *Artificial Intelligence* 97.1-2 (1997), pp. 245–271.

[38] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. “Self-Testing/Correcting with Applications to Numerical Problems.” In: *Journal of Computer and System Sciences* 47.3 (1993), pp. 549–595.

[39] Jean Bourgain. “On Lipschitz embedding of finite metric spaces in Hilbert space.” In: *Israel Journal of Mathematics* 52.1-2 (1985), pp. 46–52.

[40] Andrei Z. Broder. “On the Resemblance and Containment of Documents.” In: *Proceedings of the Compression and Complexity of Sequences (SEQUENCES ’1997)*. 1997, pp. 21–29.

[41] Alberto Calderón. “Intermediate spaces and interpolation, the complex method.” In: *Studia Mathematica* 24.2 (1964), pp. 113–190.

[42] Clément L. Canonne and Tom Gur. “An adaptivity hierarchy theorem for property testing.” In: *arXiv preprint arXiv:1702.05678* (2017).

[43] Deeparnab Chakrabarty and Seshadhri Comandur. “An o(n) monotonicity tester for boolean functions over the hypercube.” In: *SIAM Journal on Computing* 45.2 (2016), pp. 461–472.

[44] Deeparnab Chakrabarty and C. Seshadhri. “A $\widetilde{O}(n)$ non-adaptive tester for unateness.” In: *arXiv preprint arXiv:1608.06980* (2016). URL: `http://arxiv.org/abs/1608.06980`.

[45] Deeparnab Chakrabarty and C. Seshadhri. “Adaptive Boolean Monotonicity Testing in Total Influence Time.” In: *Proceedings of the 2019 ACM Conference on Innovations in Theoretical Computer Science (ITCS ’2019)*. 2019.

[46] Girish Chandrashekar and Ferat Sahin. “A survey on feature selection methods.” In: *Computers & Electrical Engineering* 40.1 (2014), pp. 16–28.

[47] Moses Charikar. “Similarity estimation techniques from rounding algorithms.” In: *Proceedings of the 34th ACM Symposium on the Theory of Computing (STOC ’2002)*. 2002, pp. 380–388.

[48] Jeff Cheeger. “A lower bound for the smallest eigenvalue of the Laplacian.” In: *Proceedings of the Princeton conference in honor of Professor S. Bochner*. 1969, pp. 195–199.

[49] Xi Chen, Rocco A. Servedio, and Li-Yang Tan. “New algorithms and lower bounds for monotonicity testing.” In: *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science (FOCS ’2014)*. 2014, pp. 285–295.

[50] Xi Chen and Erik Waingarten. “Testing Unateness Nearly Optimally.” In: *Proceedings of the 51th ACM Symposium on the Theory of Computing (STOC ’2019)*. 2019.

[51] Xi Chen, Erik Waingarten, and Jinyu Xie. "Beyond Talagrand functions: new lower bounds for testing monotonicity and unateness." In: *Proceedings of the 50th ACM Symposium on the Theory of Computing (STOC '2017)*. 2017.

[52] Xi Chen, Erik Waingarten, and Jinyu Xie. "Boolean unateness testing with $\widetilde{\Omega}(n^{3/4})$ adaptive queries." In: *Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science (FOCS '2017)*. 2017.

[53] Xi Chen and Jinyu Xie. "Tight Bounds for the Distribution-Free Testing of Monotone Conjunctions." In: *Proceedings of the 27th ACM-SIAM Symposium on Discrete Algorithms (SODA '2016)*. 2016.

[54] Xi Chen, Anindya De, Rocco A. Servedio, and Li-Yang Tan. "Boolean function monotonicity testing requires (almost) $n^{1/2}$ non-adaptive queries." In: *Proceedings of the 47th ACM Symposium on the Theory of Computing (STOC '2015)*. 2015, pp. 519–528.

[55] Xi Chen, Rocco A. Servedio, Li-Yang Tan, Erik Waingarten, and Jinyu Xie. "Settling the query complexity of non-adaptive junta testing." In: *Proceedings of the 32nd Conference on Computational Complexity (CCC '2017)*. 2017.

[56] Xi Chen, Rocco A. Servedio, Li-Yang Tan, Erik Waingarten, and Jinyu Xie. "Settling the query complexity of non-adaptive junta testing." In: *Journal of the ACM* 65.6 (2018), pp. 1–18.

[57] Hana Chockler and Dan Gutfreund. "A lower bound for testing juntas." In: *Information Processing Letters* (2004), pp. 301–305.

[58] Tobias Christiani. "A framework for similarity search with space-time tradeoffs using locality-sensitive filtering." In: *Proceedings of the 28th ACM-SIAM Symposium on Discrete Algorithms (SODA '2017)*. 2017.

[59] Mohamed Daher. "Homéomorphismes uniformes entre les sphères unité des espaces d'interpolation." In: *Comptes Rendus Mathematique* 316.10 (1993), 1051–1054.

[60] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S. Mirrokni. "Locality-sensitive hashing scheme based on p-stable distributions." In: *Proceedings of the 20th ACM Symposium on Computational Geometry (SoCG '2004)*. 2004, pp. 253–262.

[61] Yevgeniy Dodis, Oded Goldreich, Eric Lehman, Sofya Raskhodnikova, Dana Ron, and Alex Samorodnitsky. "Improved testing algorithms for monotonocity." In: *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*. 1999.

[62] Elya Dolev and Dana Ron. "Distribution-Free Testing for Monomials with a Sublinear Number of Queries." In: *Theory of Computing* 7.1 (2011), pp. 155–176.

[63] Eldar Fischer, Eric Lehman, Ilan Newman, Sofya Raskhodnikova, Ronitt Rubinfeld, and Alex Samorodnitsky. "Monotonicity testing over general poset domains." In: *Proceedings of the 34th ACM Symposium on the Theory of Computing (STOC '2002)*. 2002, pp. 474–483.

[64] Eldar Fischer, Guy Kindler, Dana Ron, Shmuel Safra, and Alex Samordinsky. "Testing juntas." In: *Journal of Computer and System Sciences* 68.4 (2004), pp. 753–787.

[65] Yoav Freund and Robert E. Schapire. "Adaptive game playing using multiplicative weights." In: *Games and Economic Behavior* 29.1-2 (1999), pp. 79–103.

[66] Oded Goldreich. *Introduction to property testing*. Cambridge University Press, 2017.

[67] Oded Goldreich, ed. *Property Testing: Current Research and Surveys*. Vol. 6390. Springer-Verlag Berlin Heidelberg, 2010.

[68] Oded Goldreich, Shafi Goldwasser, and Dana Ron. "Property testing and its connection to learning and approximation." In: *Journal of the ACM* 45.4 (1998), pp. 653–750.

[69] Oded Goldreich and Dana Ron. "Algorithmic aspects of property testing in the dense graph model." In: *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*. 2009.

[70] Oded Goldreich, Shafi Goldwasser, Eric Lehman, Dana Ron, and Alex Samordinsky. "Testing Monotonicity." In: *Combinatorica* 20.3 (2000), pp. 301–337.

[71] Mira Gonen and Dana Ron. "On the benefits of adaptivity in property testing of dense graphs." In: *Algorithmica* 58.4 (2010), pp. 811–830.

[72] Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric algorithms and combinatorial optimization*. Vol. 2. Springer Science & Business Media, 2012.

[73] Isabelle Guyon and André Elisseeff. "An introduction to variable and feature selection." In: *Journal of Machine Learning Research* 3 (2003), pp. 1157–1182.

[74] Moritz Hardt and Guy N. Rothblum. "A multiplicative weights mechanism for privacy-preserving data analysis." In: *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS '2010)*. 2010, pp. 61–70.

[75] Kevin Hartnett. *Universal Method to Sort Complex Information Found*. Quanta Magazine. `https://www.quantamagazine.org/universal-method-to-sort-complex-information-found-20180813/`. Aug. 2018.

[76] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *Elements of statistical learning: data mining, inference, and prediction*. Springer, 2001.

[77] Piotr Indyk. "Approximate nearest neighbor algorithms for Fréchet distance via product metrics." In: *Proceedings of the 18th ACM Symposium on Computational Geometry (SoCG '2002)*. 2002, pp. 102–106.

[78] Piotr Indyk. "On Approximate Nearest Neighbors under ℓ_∞ Norm." In: *Journal of Computer and System Sciences* 63.4 (2001), pp. 627–638.

[79] Piotr Indyk and Rajeev Motwani. "Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality." In: *Proceedings of the 30th ACM Symposium on the Theory of Computing (STOC '1998)*. 1998, pp. 604–613.

[80] Piotr Indyk and Assaf Naor. "Nearest-neighbor-preserving embeddings." In: *ACM Transactions on Algorithms* 3.3 (2007).

[81] Piotr Indyk and Nitin Thaper. "Fast Color Image Retrieval via Embeddings." In: *Workshop on Statistical and Computational Theories of Vision (at ICCV)*. 2003.

[82] Fritz John. "Extremum problems with inequalities as subsidiary conditions." In: *Studies and Essays Presented to R. Courant on his 60th Birthday, January 8, 1948*. Interscience Publishers Inc. New York, N. Y., 1948, pp. 187–204.

[83] William B. Johnson and Gideon Schechtman. "Embedding ℓ_p^m into ℓ_1^n." In: *Acta Mathematica* 149 (1982), pp. 71–85.

[84] Ravi Kannan, László Lovász, and Miklós Simonovits. "Random walks and an o*(n5) volume algorithm for convex bodies." In: *Random Structures & Algorithms* 11.1 (1997), pp. 1–50.

[85] Michael Kapralov and Rina Panigrahy. "NNS Lower Bounds via Metric Expansion for ℓ_∞ and EMD." In: *Proceedings of the 39th International Colloquium on Automata, Languages and Programming (ICALP '2012)*. 2012, pp. 545–556.

[86] Yitzhak Katznelson. *An introduction to harmonic analysis*. Cambridge University Press, 2004.

[87] Subhash Khot, Dor Minzer, and Muli Safra. "On monotonicity testing and boolean isoperimetric type theorems." In: *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS '2015)*. IEEE Computer Society. 2015, pp. 52–58.

[88] Subhash Khot, Dor Minzer, and Muli Safra. "On monotonicity testing and Boolean isoperimetric-type theorems." In: *SIAM Journal on Computing* 47.6 (2018), pp. 2238–2276.

[89] Subhash Khot and Igor Shinkar. "An $\widetilde{O}(n)$ queries adaptive tester for unateness." In: *Approximation, Randomization and Combinatorial Optimization. Algorithms and Techniques*. 2016, 37:1–37:7.

[90] Erica Klarreich. "Good algorithms make good neighbors." In: *Communications of the ACM* 62.7 (2019), pp. 11–13.

[91] Robert Krauthgamer and James R. Lee. "The black-box complexity of nearest-neighbor search." In: *Theoretical Computer Science* 348.2-3 (2005), pp. 262–276.

[92] Ilan Kremer, Noam Nisan, and Dana Ron. "On randomized one-round communication complexity." In: *Computational Complexity* (1999), pp. 21–49.

[93] Eyal Kushilevitz, Rafail Ostrovsky, and Yuval Rabani. "Efficient Search for Approximate Nearest Neighbor in High Dimensional Spaces." In: *SIAM Journal on Computing* 30.2 (2000), pp. 457–474.

[94] Yin Tat Lee, Aaron Sidford, and Santosh Vempala. "Efficient convex optimization with membership oracles." In: *Proceedings of the 31st Annual Conference on Learning Theory (COLT '2018)*. 2018.

[95] Tom Leighton and Satish Rao. "Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms." In: *Journal of the ACM* 46.6 (1999), pp. 787–832.

[96] Amit Levi and Erik Waingarten. "Lower Bounds for Tolerant Junta and Unateness Testing via Rejection Sampling of Graphs." In: *Proceedings of the 2019 ACM Conference on Innovations in Theoretical Computer Science (ITCS '2019)*. 2019.

[97] Nathan Linial, Eran London, and Yuri Rabinovich. "The geometry of graphs and some of its algorithmic applications." In: *Combinatorica* 15.2 (1995), pp. 215–245.

[98] Huan Liu and Hiroshi Motoda. *Feature selection for knowledge discovery and data mining*. Vol. 454. Springer Science & Business Media, 2012.

[99] Zhengyang Liu, Xi Chen, Rocco A. Servedio, Ying Sheng, and Jinyu Xie. "Distribution-free junta testing." In: *ACM Transactions on Algorithms* 15.1 (2018), pp. 1–23.

[100] Michael W. Mahoney. "Randomized algorithms for matrices and data." In: *Foundations and Trends® in Machine Learning* 3.2 (2011), pp. 123–224.

[101] Jiří Matoušek. *Lecture notes on metric embeddings*. Tech. rep. ETH Zürich, 2013.

[102] Jiří Matoušek. *Lectures on discrete geometry*. Vol. 212. Graduate texts in mathematics. Springer, 2002.

[103] Jiří Matoušek. "On embedding expanders into ℓ_p spaces." In: *Israel Journal of Mathematics* 102 (1997), pp. 189–197.

[104] Stanisław Mazur. "Une remarque sur l'homéomorphie des champs fonctionnels." In: *Studia Mathematica* 1.1 (1929), pp. 83–85.

[105] Manor Mendel and Assaf Naor. "Nonlinear spectral calculus and super-expanders." In: *Publications mathématiques de l'IHÉS* 119.1 (2014), pp. 1–95.

[106] Marvin Minsky and Seymour Papert. *Perceptrons - an introduction to computational geometry*. MIT Press, 1987.

[107] Elchanan Mossel and Ryan O'Donnell. "On the noise sensitivity of monotone functions." In: *Random Structures and Algorithms* 23.3 (2003), pp. 33–50.

[108] Elchanan Mossel, Ryan O'Donnell, and Rocco A. Servedio. "Learning juntas." In: *Proceedings of the 35th ACM Symposium on the Theory of Computing (STOC '2003)*. 2003, pp. 206–212.

[109] S. Muthukrishnan. "Data Streams: Algorithms and Applications." In: *Foundations and Trends® in Theoretical Computer Science* 1.2 (2005), pp. 117–236.

[110] Assaf Naor. "A spectral gap precludes low-dimensional embeddings." In: *Proceedings of the 33rd International Symposium on Computational Geometry (SOCG '2017)* (2017).

[111] Assaf Naor. "An average John theorem." In: *arXiv preprint arXiv:1905.01280* (2019).

[112] Assaf Naor. "An introduction to the Ribe program." In: *Japanese Journal of Mathematics* 7.2 (2012), pp. 167–233.

[113] Assaf Naor. "Comparison of metric spectral gaps." In: *Analysis and Geometry in Metric Spaces* 2.1 (2014).

[114] Assaf Naor and Yuval Rabani. "On approximate nearest neighbor search in ℓ_p, $p > 2$." Manuscript. 2006.

[115] Edward Odell and Thomas Schlumprecht. "The distortion problem." In: *Acta Mathematica* 173 (1994), pp. 259–281.

[116] Rafail Ostrovsky and Yuval Rabani. "Low distortion embeddings for edit distance." In: *Journal of the ACM* 54.4 (2007), p. 23.

[117] Ramesh Krishnan S. Pallavoor, Sofya Raskhodnikova, and Erik Waingarten. "Approximating the Distance to Monotonicity of Boolean Functions." In: *Proceedings of the 31st ACM-SIAM Symposium on Discrete Algorithms (SODA '2020)*. 2020.

[118] Rina Panigrahy, Kunal Talwar, and Udi Wieder. “Lower Bounds on Near Neighbor Search via Metric Expansion.” In: *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science (FOCS ’2010)*. 2010, pp. 805–814.

[119] Franco P. Preparata and Michael Ian Shamos. *Computational Geometry - An Introduction*. Texts and Monographs in Computer Science. Springer, 1985.

[120] Ilya Razenshteyn. “High-Dimensional Similarity Search and Sketching: Algorithms and Hardness.” PhD thesis. Massachusetts Institute of Technology, 2017.

[121] Éric Ricard. “Hölder estimates for the noncommutative Mazur maps.” In: *Archiv der Mathematik* 104.1 (2015), pp. 37–45.

[122] Dana Ron. “Algorithmic and analysis techniques in property testing.” In: *Foundations and Trends® in Theoretical Computer Science* 5.2 (2010), pp. 73–205.

[123] Dana Ron. “Property testing: A learning theory perspective.” In: *Foundations and Trends® in Machine Learning* 1.3 (2008), pp. 307–402.

[124] Bero Roos. “Binomial approximation to the Poisson binomial distribution: the Krawtchouk expansion.” In: *Theory of Probability & Its Applications* 45.2 (2001), pp. 258–272.

[125] Ronitt Rubinfeld and Madhu Sudan. “Robust characterization of polynomials with applications to program testing.” In: *SIAM Journal on Computing* 25.2 (1996), 252—271.

[126] Mert Saglam. “Near log-convexity of measured heat in (discrete) time and consequences.” In: *Proceedings of the 59th Annual IEEE Symposium on Foundations of Computer Science (FOCS ’2018)*. 2018.

[127] Steven L. Salzberg, David B. Searls, and Simon Kasif. *Computational methods in molecular biology*. Elsevier, 1998.

[128] Gregory Shakhnarovich, Trevor Darrell, and Piotr Indyk. *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice*. MIT Press, 2006.

[129] Elias M. Stein and Rami Shakarchi. *Complex Analysis*. Princeton University Press, 2003.

[130] Michel Talagrand. “How much are increasing sets positively correlated?” In: *Combinatorica* 16.2 (1996), pp. 243–258.

[131] Michel Talagrand. “Isoperimetry, logarithmic Sobolev inequalities on the discrete cube, and Margulis’ graph connectivity theorem.” In: *Geometric & Functional Analysis* 3.3 (1993), pp. 295–314.

[132] Gregory Valiant. “Finding Correlations in Subquadratic Time, with Applications to Learning Parities and the Closest Pair Problem.” In: *Journal of the ACM* 62.2 (2015), p. 13.

[133] David V. Widder. “Functions harmonic in a strip.” In: *Proceedings of the American Mathematical Society* 12.1 (1961), pp. 67–72.

[134] Ryan Williams. “A new algorithm for optimal 2-constraint satisfaction and its implications.” In: *Theoretical Computer Science* 348.2-3 (2005), pp. 357–365.

[135] Virginia Vassilevska Williams. “On some fine-grained questions in algorithms and complexity.” In: *Proceedings of the International Congress of Mathematicians (ICM ’2018)*. 2018.

[136] David P. Woodruff. “Sketching as a tool for numerical linear algebra.” In: *Foundations and Trends® in Theoretical Computer Science* 10.1–2 (2014), pp. 1–157.