On the Power of Learning from k-Wise Queries

Vitaly Feldman[1] **and Badih Ghazi**[*2]

1 **IBM Research - Almaden, USA**
vitaly@post.harvard.edu
2 **Computer Science and Artificial Intelligence Laboratory, MIT, Cambridge, USA**
badih@mit.edu

Abstract

Several well-studied models of access to data samples, including statistical queries, local differential privacy and low-communication algorithms rely on queries that provide information about a function of a single sample. (For example, a statistical query (SQ) gives an estimate of $\mathbb{E}_{x\sim D}[q(x)]$ for any choice of the query function $q : X \to \mathbb{R}$, where D is an unknown data distribution.) Yet some data analysis algorithms rely on properties of functions that depend on multiple samples. Such algorithms would be naturally implemented using k-wise queries each of which is specified by a function $q : X^k \to \mathbb{R}$. Hence it is natural to ask whether algorithms using k-wise queries can solve learning problems more efficiently and by how much.

Blum, Kalai, Wasserman [9] showed that for any weak PAC learning problem over a fixed distribution, the complexity of learning with k-wise SQs is smaller than the (unary) SQ complexity by a factor of at most 2^k. We show that for more general problems over distributions the picture is substantially richer. For every k, the complexity of distribution-independent PAC learning with k-wise queries can be exponentially larger than learning with $(k+1)$-wise queries. We then give two approaches for simulating a k-wise query using unary queries. The first approach exploits the structure of the problem that needs to be solved. It generalizes and strengthens (exponentially) the results of Blum et al. [9]. It allows us to derive strong lower bounds for learning DNF formulas and stochastic constraint satisfaction problems that hold against algorithms using k-wise queries. The second approach exploits the k-party communication complexity of the k-wise query function.

1998 ACM Subject Classification I.2.6 Learning

Keywords and phrases Statistical Queries, PAC Learning, Differential Privacy, Lower bounds, Communication Complexity

Digital Object Identifier 10.4230/LIPIcs.ITCS.2017.41

1 Introduction

In this paper, we consider several well-studied models of learning from i.i.d. samples that restrict the algorithm's access to samples to evaluation of functions of an individual sample. The primary model of interest is the statistical query model introduced by Kearns [31] as a restriction of Valiant's PAC learning model [39]. The SQ model allows the learning algorithm to access the data only via *statistical queries*, which are estimates of the expectation of any function of labeled examples with respect to the input distribution D. More precisely, if the domain of the functions is Z, then a statistical query is specified by a function

* Part of this work was done while the author was at IBM Research - Almaden. The author is supported in part by NSF STC Award CCF 0939370 and NSF Award CCF-1217423.

$\phi : Z \times \{\pm 1\} \to [-1, 1]$ and by a tolerance parameter τ. Given ϕ and τ, the statistical query oracle returns a value v which satisfies $|v - \mathbb{E}_{(z,b)\sim D}[\phi(z, b)]| \leq \tau$.

The SQ model is known to be closely-related to several other models and concepts: linear statistical functionals [41], learning with a distance oracle [5], approximate counting (or linear) queries extensively studied in differential privacy (e.g., [16, 7, 20, 34]), local differential privacy [30], evolvability [40, 23], and algorithms that extract a small amount of information from each sample [4, 26, 28, 36]. This allows to easily extend the discussion in the context of the SQ model to these related models and we will formally state several such corollaries.

Most standard algorithmic approaches used in learning theory are known to be implementable using SQs (e.g., [8, 17, 7, 11, 28, 3, 27]) leading to numerous theoretical (e.g., [2, 14, 19]) and practical (e.g., [11, 35, 38, 18]) applications. SQ algorithms have also been recently studied outside the context of learning theory [26, 28, 27]. In this case we denote the domain of data samples by X.

Another reason for the study of SQ algorithms is that it is possible to prove information-theoretic lower bounds on the complexity of any SQ algorithm that solves a given problem. Given that a large number of algorithmic approaches to problems defined over data sampled i.i.d. from some distribution can be implemented using statistical queries, this provides a strong and unconditional evidence of the problem's hardness. For a number of central problems in learning theory and complexity theory, unconditional lower bounds for SQ algorithms are known that closely match the known *computational* complexity upper bounds for those problems (*e.g.* [6, 26, 28, 12, 15]).

A natural strengthening of the SQ model (and other related models) is to allow function over k-tuples of samples instead of a single sample. That is, for a k-ary query function $\phi : X^k \to [-1, 1]$, the algorithm can obtain an estimate of $\mathbb{E}_{x_1,\ldots,x_k\sim D}[\phi(x_1, \ldots, x_k)]$. It can be seen as interpolating between the power of algorithms that can see all the samples at once and those that process a single sample at a time. While most algorithms can be implemented using standard unary queries, some algorithms are known to require such more powerful queries. The most well-known example is Gaussian elimination over $\mathbb{F}_2^n$ that is used for learning parity functions. Standard hardness amplification techniques rely on mapping examples of a function $f(z)$ to examples of a function $g(f(z_1), \ldots, f(z_k))$ (for example [10, 22]). Implementing such reduction requires k-wise queries and, consequently, to obtain a lower bound for solving an amplified problem with unary queries one needs a lower bound against solving the original problem with k-wise queries. A simple example of 2-wise statistical query is collision probability $\Pr_{x_1,x_2\sim D}[x_1 = x_2]$ that is used in several distribution property testing algorithms.

1.1 Previous work

Blum, Kalai and Wasserman [9] introduced and studied the power of k-wise SQs in the context of weak *distribution-specific* PAC learning: that is the learning algorithm observes pairs (z, b), where z is chosen randomly from some fixed and known distribution P over Z and $b = f(z)$ for some unknown function f from a class of functions $\mathcal{C}$. They showed that if a class of functions $\mathcal{C}$ can be learned with error $1/2 - \lambda$ relative to distribution P using q k-wise SQs of tolerance τ then it can be learned with error $\max\{1/2 - \lambda, 1/2 - \tau/2^k\}$ using $O(q \cdot 2^k)$ unary SQs of tolerance $\tau/2^k$.

More recently, Steinhardt et al. [36] considered k-wise queries in the b-bit sampling model in which for any query function $\phi : X^k \to \{0, 1\}^b$ an algorithm get the value $\phi(x_1, \ldots, x_k)$ for $x_1, \ldots, x_k$ drawn randomly and independently from D (it is referred to as one-way communication model in their work). They give a general technique for proving lower bounds on the number of such queries that are required to solve a given problem.

1.2 Our results

In this work, we study the relationship between the power of k-wise queries and unary queries for arbitrary problems in which the input is determined by some unknown input distribution D that belongs a (known) family of distributions $\mathcal{D}$ over domain X.

1.2.1 Separation for distribution-independent learning

We first demonstrate that for distribution-independent PAC learning $(k+1)$-wise queries are exponentially stronger than k-wise queries. We say that the k-wise SQ complexity of a certain problem is m if m is the smallest such that there exists an algorithm that solves the problem using m k-wise SQs of tolerance $1/m$.

▶ **Theorem 1.** *(Informal) For every positive integer k and any prime number p, there is a concept class $\mathcal{C}$ of Boolean functions defined over a domain of size p^{k+1} such that the $(k+1)$-wise SQ complexity of distribution-independent PAC learning $\mathcal{C}$ with is $O_k(\log p)$ whereas the k-wise SQ complexity of distribution-independent PAC learning of $\mathcal{C}$ is $\Omega_k(p^{1/4})$.*

The class of functions we use consists of all indicator functions of k-dimensional affine subspaces of $\mathbb{F}_p^{k+1}$. Our lower bound is a generalization of the lower bound for unary SQs in [25] (that corresponds to $k=1$ case of the lower bound). A simple but important observation that allows us to easily adapt the techniques from earlier works on SQs to the k-wise case is that a k-wise SQ for an input distribution $D \in \mathcal{D}$ are equivalent to unary SQ for a product distribution D^k.

The upper bound relies on the ability to find the affine subspace given $k+1$ positively labeled and linearly independent points in $\mathbb{F}_p^{k+1}$. Unfortunately, for general distributions the probability of observing such a set of points can be arbitrarily small. Nevertheless, we argue that there will exist a unique lower-dimensional affine subspace that contains enough probability mass of all the positive points in this case. This upper bound essentially implies that given k-wise queries one can solve problems that require Gaussian elimination over a system of k equations.

1.2.2 Reduction for flat $\mathcal{D}$

The separation in Theorem 1 relies on using an unrestricted class of distributions $\mathcal{D}$. We now prove that if $\mathcal{D}$ is "flat" relative to some "central" distribution $\bar{D}$ then one can upper bound the power of k-wise queries in terms of unary queries.

▶ **Definition 1.1** (Flat class of distributions)**.** *Let $\mathcal{D}$ be a set of distributions over X, and $\bar{D}$ a distribution over X. For $\gamma \geq 1$ we say that $\mathcal{D}$ is γ-flat if there exists some distribution $\bar{D}$ over X such that for all $D \in \mathcal{D}$ and all measurable subsets $E \subseteq X$, we have that $\Pr_{x\sim D}[x \in E] \leq \gamma \cdot \Pr_{x\sim \bar{D}}[x \in E]$.*

We now state our upper bound for flat classes of distributions, where we use $\mathsf{STAT}_D^{(k)}(\tau)$ to refer to the oracle that answers k-wise SQs for D with tolerance τ.

▶ **Theorem 2.** *Let $\gamma \geq 1$, $\tau > 0$ and k be any positive integer. Let X be a domain and $\mathcal{D}$ a γ-flat class of distributions over X. There exists a randomized algorithm that given any $\delta > 0$ and a k-ary function $\phi : X^k \to [-1,1]$ estimates $D^k[\phi]$ within τ for every (unknown) $D \in \mathcal{D}$ with success probability at least $1-\delta$ using*

$$\tilde{O}\left(\frac{\gamma^{k-1}\cdot k^3}{\tau^3}\cdot \log(1/\delta)\right)$$

queries to $\mathsf{STAT}_D^{(1)}(\tau/(6 \cdot k))$.

To prove this result, we use a recent general characterization of SQ complexity [25]. This characterization reduces the problem of estimating $D^k[\phi]$ to the problem of distinguishing between D^k and D_1^k for every $D \in \mathcal{D}$ and some fixed D_1. We show that when solving this problem, any k-wise query can be replaced by a randomly chosen set of unary queries. Finding these queries requires drawing samples from D^{k-1}. As we do not know D, we use $\bar{D}$ instead incurring the γ^{k-1} overhead in sampling. In Section 4 we show that weaker notions of "flatness" based on different notions of divergence between distributions can also be used in this reduction.

It is easy to see that, when PAC learning $\mathcal{C}$ with respect to a fixed distribution P over Z, the set of input distributions is 2-flat (relative to the distribution that is equal to P on Z and gives equal weight $1/2$ to each label). Therefore, our result generalizes the results in [9]. More importantly, the tolerance in our upper bound scales linearly with k rather than exponentially (namely, $\tau/2^k$).

This result can be used to obtain lower bounds against k-wise SQs algorithms from lower bounds against unary SQ algorithms. In particular, it can be used to rule out reductions that require looking at k points of the original problem instance to obtain each point of the new problem instance. As an application, we obtain exponential lower bounds for solving constraint stochastic satisfaction problems and DNF learning by k-wise SQ algorithm with $k = n^{1-\alpha}$ for any constant $\alpha > 0$ from lower bounds for CSPs given in [28]. We state the result for learning DNF here. Definitions and the lower bound for CSPs can be found in Section 4.3.

▶ **Theorem 3.** *For any constant $\alpha > 0$ (independent of n), there exists a constant $\beta > 0$ such that any algorithm that learns DNF formulas of size n with error $< 1/2 - n^{-\beta \log n}$ and success probability at least $2/3$ requires at least $2^{n^{1-\alpha}}$ calls to $\mathsf{STAT}_D^{(n^{1-\alpha})}(n^{-\beta \log n})$.*

This lower bound is based on a simple and direct reduction from solving the stochastic CSP that arises in Goldreich's proposed PRG [29] to learning DNF that is of independent interest (see Lemma 15). For comparison, the standard SQ lower bound for learning polynomial size DNF [6] relies on hardness of learning parities of size $\log n$ over the uniform distribution. Yet, parities of size $\log n$ can be easily learned from $(\log^2 n)$-wise statistical queries (since solving a system of $\log^2 n$ linear equations will uniquely identify a $\log n$-sparse parity function). Hence our lower bound holds against qualitatively stronger algorithms. Our lower bound is also exponential in the number of queries whereas the known argument implies only a quasipolynomial lower bound[1].

1.2.3 Reduction for low-communication queries

Finally, we point out that k-wise queries that require little information about each of the inputs can also be simulated using unary queries. This result is a simple corollary of the recent work of Steinhardt et al. [36] who show that any computation that extracts at most b bits from each of the samples (not necessarily at once) can be simulated using unary SQs.

▶ **Theorem 4.** *Let $\phi : X^k \to \{\pm 1\}$ be a function, and assume that ϕ has k-party public-coin randomized communication complexity of b bits per party with success probability $2/3$. Then,*

[1] We remark that an exponential lower bound on the number of queries has not been previously stated even for unary SQs. The unary version can be derived from known results as explained in Section 4.3.

there exists a randomized algorithm that, with probability at least $1-\delta$*, estimates* $\mathbb{E}_{x\sim D^k}[\phi(x)]$ *within* τ *using* $O(b\cdot k\cdot\log(1/\delta)/\tau^2)$ *queries to* $\mathsf{STAT}_D^{(1)}(\tau')$ *for some* $\tau'=\tau^{O(b)}/k$.

As a simple application of Theorem 4, we show a unary SQ algorithm that estimates the collision probability of an unknown distribution D within τ using $1/\tau^2$ queries $\mathsf{STAT}_D^{(1)}(\tau^{O(1)})$. The details appear in Section 5.

1.2.4 Corollaries for related models

Our separation result and reductions imply similar results for k-wise versions of two well-studied learning models: local differential privacy and the b-bit sampling model.

Local differentially private algorithms [30] (also referred to as randomized response) are differentially private algorithms in which each sample goes through a differentially private transformation chosen by the analyst. This model is the focus of recent privacy preserving industrial applications by Google [21] and Apple. We define a k-wise version of this model in which analyst's differentially private transformations are applied to k-tuples of samples. This model interpolates naturally between the usual (or global) differential privacy and the local model.

Kasiviswanathan et al. [30] showed that a concept class is learnable by a local differentially private algorithm if and only if it is learnable in the SQ model. Hence up to polynomial factors the models are equivalent (naturally, such polynomial factors are important for applications but here we focus only on the high-level relationships between the models). This result also implies that k-local differentially private algorithms (formally defined in Section 6.1) are equivalent to k-wise SQ algorithms (up to a polynomial blow-up in the complexity). Theorem 1 then implies an exponential separation between k-wise and $(k+1)$-wise local differentially private algorithms (see Corollary 21 for details). It can be seen as a substantial strengthening of a separation between the local model and the global one also given in [30]. The reductions in Theorem 2 and Theorem 4 imply two approaches for simulating k-local differentially private algorithms using 1-local algorithms.

The SQ model is also known to be equivalent (up to a factor polynomial in 2^b) to the b-bit sampling model introduced by Ben-David and Dichterman [4] and studied more recently in [26, 28, 43, 37, 36]. Lower bounds for the k-wise version of this model are given in [43, 36]. Our results can be easily translated to this model as well. We provide additional details in Section 6.

2 Preliminaries

For any distribution D over a domain X and any positive integer k, we denote by D^k the distribution over X^k obtained by drawing k i.i.d. samples from D. For a distribution D over a domain X and a function $\phi: X\to\mathbb{R}$, we denote $D[\phi]\doteq\mathbb{E}_{x\sim D}[\phi(x)]$.

Next, we formally define the k-wise SQ oracle.

▶ **Definition 2.1.** *Let* D *be a distribution over a domain* X *and* $\tau>0$*. A* k*-wise statistical query oracle* $\mathsf{STAT}_D^{(k)}(\tau)$ *is an oracle that given as input any function* $\phi: X^k\to[-1,+1]$*, returns some value* v *such that* $|v-\mathbb{E}_{x\sim D^k}[\phi(x)]|\le\tau$.

We say that a k-wise SQ algorithm is given access to $\mathsf{STAT}^{(k)}(\tau)$, if for every when the algorithm is given access to $\mathsf{STAT}_D^{(k)}(\tau)$, where D is the input distribution. We note that for $k=1$, Definition 2.1 reduces to the usual definition of an SQ oracle that was first introduced by Kearns [31]. The k-wise SQ complexity of solving a problem with access to $\mathsf{STAT}^{(k)}(\tau)$

is the minimum number of queries q for which exists a k-wise SQ algorithm with access to $\mathsf{STAT}^{(k)}(\tau)$ that solves the problem using at most q queries. Our discussion and results can also be easily extended to the stronger VSTAT oracle defined in [26] and to more general real-valued queries using the reductions in [24].

The PAC learning [39] is defined as follows.

▶ **Definition 2.2.** *For a class $\mathcal{C}$ of Boolean-valued functions over a domain Z, a PAC learning algorithm for $\mathcal{C}$ is an algorithm that for every P distribution over Z and $f \in \mathcal{C}$, given an error parameter $\epsilon > 0$, failure probability $\delta > 0$ and access to i.i.d. labeled examples of the form $(x, f(x))$ where $x \sim P$, outputs a hypothesis function h that, with probability at least $1 - \delta$, satisfies $\Pr_{x\sim P}[h(x) \neq f(x)] \leq \epsilon$.*

We next define one-vs-many decision problems, which will be used in the proofs in our Section 3 and Section 4.

▶ **Definition 2.3** (Decision problem $\mathcal{B}(\mathcal{D}, D_0)$)**.** *Let $\mathcal{D}$ be a set of distributions and D_0 a reference distribution over a set X. We denote by $\mathcal{B}(\mathcal{D}, D_0)$ the decision problem where we are given access to a distribution $D \in \mathcal{D} \cup \{D_0\}$ and wish to distinguish whether $D \in \mathcal{D}$ or $D = D_0$.*

3 Separation of $(k+1)$-wise from k-wise queries

We start by describing the concept class $\mathcal{C}$ that we use to prove Theorem 1. Let ℓ and k be positive integers with $\ell \geq k + 1$. The domain will be $\mathbb{F}_p^\ell$. For every $a = (a_1, \ldots, a_\ell) \in \mathbb{F}_p^\ell$, we consider the hyperplane

$$\mathsf{Hyp}_a \doteq \{z = (z_1, \ldots, z_\ell) \in \mathbb{F}_p^\ell : z_\ell = a_1 z_1 + \cdots + a_{\ell-1} z_{\ell-1} + a_\ell\}.$$

We then define the Boolean-valued function $f_a : \mathbb{F}_p^\ell \to \{\pm 1\}$ to be the indicator function of the subset $\mathsf{Hyp}_a \subseteq \mathbb{F}_p^\ell$, i.e., for every $z \in \mathbb{F}_p^\ell$,

$$f_a(z) = \begin{cases} +1 & \text{if } z \in \mathsf{Hyp}_a, \\ -1 & \text{otherwise.} \end{cases}$$

Then, we will consider the concept classes $\mathcal{C}_\ell \doteq \{f_a : a \in \mathbb{F}_p^\ell\}$. We denote $\mathcal{C} \doteq \mathcal{C}_{k+1}$. We start by stating our upper bound on the $(k+1)$-wise SQ complexity of the distribution-independent PAC learning of $\mathcal{C}_{k+1}$.

▶ **Lemma 3.1** ($(k+1)$-wise upper bound)**.** *Let p be a prime number and k be a positive integer. There exists a distribution-independent PAC learning algorithm for $\mathcal{C}_{k+1}$ that makes at most $t \cdot \log(1/\epsilon)$ queries to $\mathsf{STAT}^{(k+1)}(\epsilon/t)$, for some $t = O_k(\log p)$.*

We next state our lower bound on the k-wise SQ complexity of the same tasks considered in Lemma 3.1.

▶ **Lemma 3.2** (k-wise lower bound)**.** *Let p be a prime number and ℓ, k be positive integers with $\ell \geq k+1$ and $k = O(p)$. There exists $t = \Omega\left(p^{(\ell-k)/4}\right)$ such that any distribution-independent PAC learning alogrithm for $\mathcal{C}_\ell$ with error at most $1/2 - 2/t$ that is given access to $\mathsf{STAT}^{(k)}(1/t)$ needs at least t queries.*

Note that Lemma 3.1 and Lemma 3.2 imply Theorem 1.

3.1 Upper bound

3.1.1 Notation

We first introduce some notation that will be useful in the description of our algorithm. For any matrix M with entries in the finite field $\mathbb{F}_p$, we denote by $\mathsf{rk}(M)$ the rank of M over $\mathbb{F}_p$. Let $(a_1, \ldots, a_{k+1}) \in \mathbb{F}_p^{k+1}$ be the unknown vector that defines f_a and P be the unknown distribution over tuples $(z_1, \ldots, z_{k+1}) \in \mathbb{F}_p^{k+1}$.

Note that Hyp_a is an affine subspace of $\mathbb{F}_p^{k+1}$. To simplify our treatment of affine subspaces, we embed the points of $\mathbb{F}_p^{k+1}$ into $\mathbb{F}_p^{k+2}$ by mapping each $z \in \mathbb{F}_p^{k+1}$ to $(z, 1)$. This embedding maps every affine subspace V of $\mathbb{F}_p^{k+1}$ to a linear subspace W of $\mathbb{F}_p^{k+2}$, namely the span of the image of V under our embedding. Note that this mapping is one-to-one and allows us to easily recover V from W as $V = \{z \in \mathbb{F}_p^{k+1} \mid (z, 1) \in W\}$. Hence given $k+1$ examples

$$\big((z_{1,1}, \ldots, z_{1,k+1}), b_1\big), \big((z_{2,1}, \ldots, z_{2,k+1}), b_2\big), \ldots, \big((z_{k+1,1}, \ldots, z_{k+1,k+1}), b_{k+1}\big)$$

we define the matrix:

$$Z \doteq \begin{bmatrix} z_{1,1} & z_{1,2} & \cdot & z_{1,k+1} & 1 \\ z_{2,1} & z_{2,2} & \cdot & z_{2,k+1} & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ z_{k+1,1} & z_{k+1,2} & \cdot & z_{k+1,k+1} & 1 \end{bmatrix}. \tag{1}$$

For $\ell \in [k+1]$ we also denote by Z_ℓ the matrix that consists of the top ℓ rows of Z. Further, for a $(k+1)$-wise query function $\phi\big((z_1, b_1), \ldots, (z_{k+1}, b_{k+1})\big)$, we use Z to refer to the matrix obtained from the inputs to the function.

Let Q be the distribution defined by sampling a random example $\big((z_1, \ldots, z_{k+1}), b\big)$, conditioning on the event that $b = 1$ and outputting $(z_1, \ldots, z_{k+1}, 1)$. Note that if the examples from which Z is built are positively labeled i.i.d. examples then each row of Z is sampled i.i.d. from Q and hence Z_ℓ is distributed according to Q^ℓ. We denote by 1^{k+1} the all $+1$'s vector of length $k+1$.

3.1.2 Learning algorithm

We start by explaining the main ideas behind the algorithm. On a high level, in order to be able to use $(k+1)$-wise SQs to learn the unknown subspace, we need to make sure that there exists an affine subspace that contains most of the probability mass of the positively-labeled points and that is spanned by $k+1$ random positively-labeled points with noticeable probability. Here, the probability is with respect to the unknown distribution over labeled examples. Thus, for positively labeled tuples $(z_{1,1}, \ldots, z_{1,k+1})$, $(z_{2,1}, \ldots, z_{2,k+1})$, $\ldots$, $(z_{k+1,1}, \ldots, z_{k+1,k+1})$, we consider the $(k+1) \times (k+2)$ matrix Z defined in Equation (1). If W is the row-span of Z, then the desired (unknown) affine subspace is the set V of all points $(z_1, \ldots, z_{k+1})$ such that $(z_1, \ldots, z_{k+1}, 1) \in W$.

If the (unknown) distribution over labeled examples is such that with noticeable probability, $k+1$ random positively-labeled points form a full-rank linear system (i.e., the matrix Z has full-rank with noticeable probability conditioned on $(b_1, \ldots, b_{k+1}) = 1^{k+1}$), we can use $(k+1)$-wise SQs to find, one bit at a time, the $(k+1)$-dimensional row-span W of Z, and we can then output the set V of all points $(z_1, \ldots, z_{k+1})$ such that $(z_1, \ldots, z_{k+1}, 1) \in W$ as the desired affine subspace (below, we refer to this step as the Recovery Procedure).

We now turn to the (more challenging) case where the system is not full-rank with noticeable probability (i.e., the matrix Z is rank-deficient with high probability conditioned

Algorithm 1 $(k+1)$-wise SQ Algorithm.

Inputs. $k \in \mathbb{N}$, error probability $\epsilon > 0$.

Output. Function $f : \mathbb{F}_p^{k+1} \to \{\pm 1\}$.

Set tolerance of each SQ to $\tau = (\epsilon/2^{c\cdot(k+2)})^{(k+1)^{k+3}}$, where $c > 0$ is a large enough absolute constant.
Define the threshold $\tau_i = 2^{c\cdot(k+2-i)} \cdot k \cdot \tau^{1/(k+1)^{k+2-i}}$ for every $i \in [k+1]$.
Ask the SQ $\phi(z, b) \doteq \mathbb{1}(b = 1)$ and let w be the response.
if $w \leq \epsilon - \tau$ **then**
    Output the all -1's function.
end if
Let $\widetilde{\phi}\big((z_1, b_1), \ldots, (z_{k+1}, b_{k+1})\big) \doteq \mathbb{1}((b_1, \ldots, b_{k+1}) = 1^{k+1})$.
Ask the SQ $\widetilde{\phi}$ and let v be the response.
for $i = k+1$ down to 1 **do**
    Let $\phi_i\big((z_1, b_1), \ldots, (z_{k+1}, b_{k+1})\big) \doteq \mathbb{1}((b_1, \ldots, b_{k+1}) = 1^{k+1}$ and $\mathsf{rk}(Z) = i)$.
    Ask the SQ ϕ_i and let v_i be the response.
    if $v_i/v \geq \tau_i$ **then**
        Run Recovery Algorithm on input (i, v_i) and let $\widehat{V}$ be the subspace of $\mathbb{F}_p^{k+1}$ it outputs.
        Define function $f : \mathbb{F}_p^{k+1} \to \{-1, 1\}$ by:
        $f(z_1, \ldots, z_{k+1}) = +1$ if $(z_1, \ldots, z_{k+1}) \in \widehat{V}$.
        $f(z_1, \ldots, z_{k+1}) = -1$ otherwise.
        Return f.
    end if
end for

on $(b_1, \ldots, b_{k+1}) = 1^{k+1})$. Then, the system has rank at most i with high probability, for some $i < k+1$. There is a large number of possible i-dimensional subspaces and therefore it is no longer clear that there exists a single i-dimensional subspace that contains most of the mass of the positively-labeled points. However, we demonstrate that for every i, if the rank of Z is at most i with sufficiently high probability, then there exists a *fixed* subspace W of dimension at most i that contains a large fraction of the probability under the row-distribution of Z (it turns out that if this subspace has rank equal to i, then it should be *unique*). We can then use $(k+1)$-wise SQs to output the affine subspace V consisting of all points $(z_1, \ldots, z_{k+1})$ such that $(z_1, \ldots, z_{k+1}, 1) \in W$ (via the Recovery Procedure).

The general description of the algorithm is given in Algorithm 1, and the Recovery Procedure (allowing the reconstruction of the affine subspace V) is separately described in Algorithm 2. We denote the indicator function of event E by $\mathbb{1}(E)$. Note that the statistical query corresponding to the event $\mathbb{1}(E)$ gives an estimate of the probability of E.

3.1.3 Analysis

We now turn to the analysis of Algorithm 1 and the proof of Lemma 3.1. We will need the following lemma, which shows that if the rank of Z is at most i with high probability, then there is a *fixed* subspace of dimension at most i containing most of the probability mass under the row-distribution of Z.

▶ **Lemma 3.3.** *Let $i \in [k+1]$. If $\Pr_{Q^{k+1}}[\mathsf{rk}(Z) \leq i] \geq 1 - \xi$, then there exists a subspace W of $\mathbb{F}_p^{k+2}$ of dimension at most i such that $\Pr_{z\sim Q}[z \notin W] \leq \xi^{1/k}$.*

Algorithm 2 Recovery Procedure

Input. Integer $i \in [k+1]$.

Output. Subspace $\widehat{V}$ of $\mathbb{F}_p^{k+1}$ of dimension i.

Let $m_i = (k+2) \cdot i \cdot \lceil \log p \rceil$
for each bit $j \leq m_i$ **do**
  Define event $E_j(Z) = \mathbb{1}(\text{bit } j \text{ of row span of } Z \text{ is } 1)$.
  Let $\phi_{i,j}\big((z_1, b_1), \ldots, (z_{k+1}, b_{k+1})\big) \doteq \mathbb{1}(E_j(Z) \text{ and } (b_1, \ldots, b_{k+1}) = 1^{k+1} \text{ and } \mathsf{rk}(Z) = i)$.
  Ask the SQ $\phi_{i,j}$ and let $u_{i,j}$ be the response.
  if $u_{i,j}/v_i \geq (9/10)$ **then**
    Set bit j in binary representation of $\widehat{W}$ to 1.
  else
    Set bit j in binary representation of $\widehat{W}$ to 0.
  end if
end for
Let $\widehat{V}$ be the set all points $(z_1, \ldots, z_{k+1})$ such that $(z_1, \ldots, z_{k+1}, 1) \in \widehat{W}$.

▶ Remark. We point out that the exponential dependence on $1/k$ in the probability upper bound in Lemma 3.3 is tight. To see this, let $p = 2$, and $\{e_1, \ldots, e_k\}$ be the standard basis in $\mathbb{F}_2^k$. Consider the base distribution P on $\mathbb{F}_2^k$ that puts probability mass $1 - \alpha$ on e_1, and probability mass $\alpha/(k-1)$ on each of e_2, e_3, …, e_k. Then, a Chernoff bound implies that if we draw k i.i.d. samples from P, then the dimension of their span is at most $2 \cdot \alpha \cdot k$ with probability at least $1 - \exp(-k)$. On the other hand, for any subspace W of $\mathbb{F}_2^k$ of dimension $2 \cdot \alpha \cdot k$, the probability that a random sample from P lies inside W is only $1 - \Theta(\alpha)$.

To prove Lemma 3.3, we will use the following proposition.

▶ **Proposition 3.4.** *Let $\ell \in [k+1]$, $i \in [\ell - 1]$ and $\eta > 0$. If $\Pr_{Q^\ell}[\mathsf{rk}(Z_\ell) \leq i] \geq 1 - \eta$, then for every $\nu \in (0, 1]$, either there exists a subspace W of $\mathbb{F}_p^{k+2}$ of dimension i such that $\Pr_{z \sim Q}[z \notin W] \leq \nu$ or $\Pr_{Q^i}[\mathsf{rk}(Z_i) \leq i - 1] \geq 1 - \eta/\nu$.*

Proof. Let $p \doteq \Pr_{Q^i}[\mathsf{rk}(Z_i) \leq i - 1]$. For every (fixed) matrix $A_i \in \mathbb{F}_p^{i \times (k+2)}$, define

$$\mu(A_i) \doteq \Pr_{Q^\ell}[\mathsf{rk}(Z_\ell) \leq i \mid Z_i = A_i].$$

Then,

$$\begin{aligned}\Pr_{Q^\ell}[\mathsf{rk}(Z_\ell) \leq i] &= p + (1-p) \cdot \Pr_{Q^\ell}[\mathsf{rk}(Z_\ell) \leq i \mid \mathsf{rk}(Z_i) = i] \\ &= p + (1-p) \cdot \mathbb{E}_{Q^i}\Big[\mu(Z_i) \Big| \mathsf{rk}(Z_i) = i\Big].\end{aligned}$$

Since $\Pr_{Q^\ell}[\mathsf{rk}(Z_\ell) \leq i] \geq 1 - \eta$, we have that

$$\mathbb{E}_{Q^i}\Big[\mu(Z_i) \Big| \mathsf{rk}(Z_i) = i\Big] \geq 1 - \eta/(1-p).$$

Hence, there exists a setting $A_i \in \mathbb{F}_p^{i \times (k+2)}$ of Z_i such that $\mathsf{rk}(A_i) = i$ and

$$\Pr[\mathsf{rk}(Z_\ell) \leq i \mid Z_i = A_i] \geq 1 - \eta/(1-p).$$

We let W be the $\mathbb{F}_p$-span of the rows of A_i. Note that the dimension of W is equal to i and that $\Pr_{z \sim Q}[z \notin W] \leq \eta/(1-p)$. Thus, we conclude that for every $\nu \in (0, 1]$, either $p \geq 1 - \eta/\nu$ or $\Pr_{z \sim Q}[z \notin W] \leq \nu$, as desired. ◀

We now complete the proof of Lemma 3.3.

Proof of Lemma 3.3. Starting with $\ell = k+1$ and $\eta = \xi$, we inductively apply Proposition 3.4 with $\nu = \xi^{1/k}$ until we either get the desired subspace W or we get to the case where $i = 1$. In this case, we have that $\Pr_{Q^\ell}[\mathsf{rk}(Z_\ell) \leq 1] \geq 1 - \xi^{1/k}$ for $\ell \geq 2$. Since the last column of Z_ℓ is the all 1's vector, we conclude that there exists $z^* \in \mathbb{F}_p^{k+1}$ such that $\Pr_{z\sim Q}[z \neq (z^*, 1)] \leq \xi^{1/k}$. We can then set our subspace W to be the $\mathbb{F}_p$-span of the vector $(z^*, 1)$. ◀

For the proof of Lemma 3.1 we will also need the following lemma, which states sufficient conditions under which the Recovery Procedure (Algorithm 2) succeeds.

▶ **Lemma 3.5.** *Let $i \in [k+1]$. Assume that in Algorithm 1, $v > \epsilon^{k+1}/2$ and $v_i/v \geq \tau_i$. If there exists a subspace W of $\mathbb{F}_p^{k+2}$ of dimension equal to i such that*

$$\Pr_{z\sim Q}[z \notin W] < \frac{\tau_i}{4 \cdot (k+1)}, \tag{2}$$

then the affine subspace $\widehat{V}$ output by Algorithm 2 (i.e., the Recovery Procedure) consists of all points $(z_1, \ldots, z_{k+1})$ such that $(z_1, \ldots, z_{k+1}, 1) \in W$.

We note that Lemma 3.5 would still hold under quantitatively weaker assumptions on v, v_i/v and $\Pr_{z\sim Q}[z \notin W]$ in Equation (2). In order to keep the expressions simple, we however choose to state the above version which will be sufficient to prove Lemma 3.1. The proof of Lemma 3.5 appears in Section A.1. We are now ready to complete the proof of Lemma 3.1.

Proof of Lemma 3.1. If Algorithm 1 terminates at Step 5, then the error of the output hypothesis is at most ϵ, as desired. Henceforth, we assume that Algorithm 1 does not terminate at Step 5. Then, we have that $\Pr[b = 1] > \epsilon$, and hence $\Pr[(b_1, \ldots, b_{k+1}) = 1^{k+1}] > \epsilon^{k+1}$. Thus, the value v obtained in Step 8 of Algorithm 1 satisfies $v > \epsilon^{k+1} - \tau \geq \epsilon^{k+1}/2$, where the last inequality follows from the setting of τ. Let i^* be the first (i.e., largest) value of $i \in [k+1]$ for which $v_i/v \geq \tau_i$. To prove that such an i^* exists, we proceed by contradiction, and assume that for all $i \in [k+1]$, it is the case that $v_i/v < \tau_i$. Note that Z has an all 1's column, so it has rank at least 1. Moreover, it has rank at most $k+1$. Therefore, we have that

$$\begin{aligned}
1 &= \Pr[1 \leq \mathsf{rk}(Z) \leq k+1 \mid (b_1, \ldots, b_{k+1}) = 1^{k+1}] \\
&= \sum_{i=1}^{k+1} \Pr[\mathsf{rk}(Z) = i \mid (b_1, \ldots, b_{k+1}) = 1^{k+1}] \\
&\leq \sum_{i=1}^{k+1} \frac{v_i + \tau}{v - \tau} \\
&\leq 2 \cdot \sum_{i=1}^{k+1} \frac{v_i + \tau}{v} \\
&\leq 2 \cdot \sum_{i=1}^{k+1} \left(\frac{v_i}{v} + \frac{2\tau}{\epsilon^{k+1}}\right) \\
&< 2 \cdot \sum_{i=1}^{k+1} \tau_i + 4 \cdot (k+1) \cdot \frac{\tau}{\epsilon^{k+1}}.
\end{aligned}$$

Using the fact that τ_i is monotonically non-increasing in i and the settings of τ_1 and τ, the last inequality gives

$$1 \leq 2 \cdot (k+1) \cdot \tau_1 + 4 \cdot (k+1) \cdot \frac{\tau}{\epsilon^{k+1}} < 1,$$

a contradiction.

We now fix i^* as above. We have that

$$\begin{aligned}
\Pr[\mathsf{rk}(Z) \le i^* \mid (b_1,\dots,b_{k+1}) = 1^{k+1}] &= 1 - \sum_{i=i^*+1}^{k+1} \Pr[\mathsf{rk}(Z) = i \mid (b_1,\dots,b_{k+1}) = 1^{k+1}] \\
&\ge 1 - \sum_{i=i^*+1}^{k+1} \frac{v_i + \tau}{v - \tau} \\
&\ge 1 - 2 \cdot \sum_{i=i^*+1}^{k+1} (\frac{v_i}{v} + \frac{2\tau}{\epsilon^{k+1}}) \\
&> 1 - 2 \cdot \sum_{i=i^*+1}^{k+1} (\tau_i + 2 \cdot \frac{\tau}{\epsilon^{k+1}}) \\
&\ge 1 - 4 \cdot \sum_{i=i^*+1}^{k+1} \tau_i \\
&\ge 1 - 4 \cdot k \cdot \tau_{i^*+1}.
\end{aligned}$$

By Lemma 3.3, there exists a subspace W of $\mathbb{F}_p^{k+2}$ of dimension at most i^* such that

$$\Pr_{z \sim Q}[z \notin W] \le (4 \cdot k)^{1/k} \cdot \tau_{i^*+1}^{1/k}. \tag{3}$$

▶ **Proposition 3.6.** *For every $i \in [k]$, we have that $(k+1) \cdot (4 \cdot k)^{1/k} \cdot \tau_{i+1}^{1/k} \le \tau_i/4$.*

We note that Proposition 3.6 follows immediately from the definitions of τ_i and τ (and by letting c by a sufficiently large positive absolute constant). Moreover, Proposition 3.6 (applied with $i = i^*$) along with Equation (3) imply that $\Pr_{z \sim Q}[z \notin W]$ is at most $\tau_{i*}/(4(k+1))$.

By a union bound, we get that with probability at least

$$1 - (k+1) \cdot \Pr_{z \sim Q}[z \notin W] \ge 1 - \frac{\tau_{i^*}}{4}, \tag{4}$$

all the rows of Z belong to W.

Since $v_{i*}/v \ge \tau_{i*}$, we also have that:

$$\begin{aligned}
\Pr[\mathsf{rk}(Z) = i^* \mid (b_1,\dots,b_{k+1}) = 1^{k+1}] &\ge \frac{v_{i*} - \tau}{v + \tau} \\
&\ge \frac{1}{2} \cdot \frac{(v_{i*} - \tau)}{v} \\
&\ge \frac{1}{2} \cdot (\tau_{i^*} - \frac{2 \cdot \tau}{\epsilon^{k+1}}) \\
&\ge \frac{\tau_{i^*}}{3}
\end{aligned} \tag{5}$$

Combining Equation (4) and Equation (5), we get that the rank of W is *equal to* i^*.

Let V be the affine subspace consisting of all points $(z_1,\dots,z_{k+1})$ such that $(z_1,\dots,z_{k+1},1) \in W$. By Lemma 3.5, we get that Algorithm 2 (and hence Algorithm 1) correctly recovers the affine subspace V.

We note that the function f output by Algorithm 1 is the ± 1 indicator of a subspace of the true hyperplane Hyp_a. To see this, note that f is the ± 1 indicator function of the subspace V, and by Equations (3) and (5), we have that with probability at least $\tau_{i*}/12$ over $Z \sim Q^{k+1}$, all the columns of Z belong to W and $\mathsf{rk}(Z) = i^*$. Since the dimension of W

is equal to i^* and since we are conditioning on $(b_1, \ldots, b_{k+1}) = 1^{k+1}$, this implies that the correct label of all the points in V is $+1$. Hence, f only possibly errs on positively-labeled points (by wrongly giving them the label -1). Moreover, Algorithm 1 ensures that the output function f gives the label $+1$ to every $(z_1, \ldots, z_{k+1}) \in \mathbb{F}_p^{k+1}$ for which $(z_1, \ldots, z_{k+1}, 1) \in W$. Therefore, the function f that is output by Algorithm 1 (when it does not terminate at Step 5) has error at most the right hand side of (3). So to upper-bound the error probability, it suffices for us to verify that the right-hand side of (3) is at most ϵ. This is obtained by applying the next proposition with $i = i^* + 1$.

▶ **Proposition 3.7.** *For every $i \in [k+1]$, we have that $(4 \cdot k)^{1/k} \cdot \tau_i^{1/k} \leq \epsilon^k$.*

The proof of Proposition 3.7 follows immediately from the definitions of τ_i and τ and by letting c be a sufficiently large positive absolute constant.

The number of queries performed by the $(k+1)$-wise algorithm is at most $O(k^2 \cdot \log p)$, and their tolerance is $\tau \geq (\epsilon / 2^{c \cdot (k+2)})^{(k+1)^{k+3}}$, where c is a positive absolute constant. Finally, we remark that the dependence of the SQ complexity of the above algorithm on the error parameter ϵ is $\epsilon^{-k^{O(k)}}$. It can be improved to a linear dependence on $1/\epsilon$ by learning with error $1/3$ and then using boosting in the standard way (boosting in the SQ model works essentially as in the regular PAC model [1]). ◀

3.2 Lower bound

Our proof of lower bound is a generalization of the lower bound in [25] (for $\ell = 2$ and $k = 1$). It relies on a notion of *combined randomized statistical dimension* ("combined" refers to the fact that it examines a single parameter that lower bounds both the number of queries and the inverse of the tolerance). In order to apply this approach we need to extend it to k-wise queries. This extension follows immediately from a simple observation. If we define the domain to be $X' \doteq X^k$ and the input distribution to be $D' \doteq D^k$ then asking a k-wise query $\phi : X^k \to [-1, 1]$ to $\mathsf{STAT}_D^{(k)}(\tau)$ is equivalent to asking a unary query $\phi : X' \to [-1, 1]$ to $\mathsf{STAT}_{D'}^{(k)}(\tau)$. Using this observation we define the k-wise versions of the notions from [25] and give their properties that are needed for the proof of Lemma 3.2.

3.2.1 Preliminaries

Combined randomized statistical dimension is based on the following notion of average discrimination.

▶ **Definition 3.8** (k-wise average κ_1-discrimination)**.** *Let k be any positive integer. Let μ be a probability measure over distributions over X and D_0 be a reference distribution over X. Then,*

$$\bar{\kappa}_1^{(k)}(\mu, D_0) \doteq \sup_{\phi : X^k \to [-1,+1]} \left\{ \mathbb{E}_{D \sim \mu}[|D^k[\phi] - D_0^k[\phi]|] \right\}.$$

We denote the problem of PAC learning a concept class $\mathcal{C}$ of Boolean functions up to error ϵ by $\mathcal{L}_{PAC}(\mathcal{C}, \epsilon)$. Let Z be the domain of the Boolean functions in $\mathcal{C}$. For any distribution D_0 over labeled examples (i.e., over $Z \times \{\pm 1\}$), we define the Bayes error rate of D_0 to be

$$\mathsf{err}(D_0) = \sum_{z \in Z} \min\{D_0(z, 1), D_0(z, -1)\} = \min_{h : Z \to \{\pm 1\}} \Pr_{(z,b) \sim D_0}[h(z) \neq b].$$

▶ **Definition 3.9** (k-wise combined randomized statistical dimension)**.** *Let k be any positive integer. Let $\mathcal{D}$ be a set of distributions and D_0 a reference distribution over X. The k-wise combined randomized statistical dimension of the decision problem $\mathcal{B}(\mathcal{D}, D_0)$ is then defined as*

$$\mathsf{cRSD}^{(k)}_{\bar{\kappa}_1}(\mathcal{B}(\mathcal{D}, D_0)) \doteq \sup_{\mu \in S^{\mathcal{D}}} (\bar{\kappa}^{(k)}_1(\mu, D_0))^{-1},$$

where $S^{\mathcal{D}}$ denotes the set of all probability distributions over $\mathcal{D}$.

Further, for any concept class $\mathcal{C}$ of Boolean functions over a domain Z, and for any $\epsilon > 0$, the k-wise combined randomized statistical dimension of $\mathcal{L}_{PAC}(\mathcal{C}, \epsilon)$ is defined as

$$\mathsf{cRSD}^{(k)}_{\bar{\kappa}_1}(\mathcal{L}_{PAC}(\mathcal{C}, \epsilon)) \doteq \sup_{D_0 \in S^{Z \times \{\pm 1\}} : \mathsf{err}(D_0) > \epsilon} \mathsf{cRSD}^{(k)}_{\bar{\kappa}_1}(\mathcal{B}(\mathcal{D}_{\mathcal{C}}, D_0)),$$

where $\mathcal{D}_{\mathcal{C}} \doteq \{P^f : P \in S^Z, f \in \mathcal{C}\}$ with P^f denoting the distribution on labeled examples $(x, f(x))$ with $x \sim P$.

The next theorem lower bounds the randomized k-wise SQ complexity of PAC learning a concept class in terms of its k-wise combined randomized statistical dimension.

▶ **Theorem 5** ([25])**.** *Let $\mathcal{C}$ be a concept class of Boolean functions over a domain Z, k be a positive integer and $\epsilon, \delta > 0$. Let $d \doteq \mathsf{cRSD}^{(k)}_{\bar{\kappa}_1}(\mathcal{L}_{PAC}(\mathcal{C}, \epsilon))$. Then, the randomized k-wise SQ complexity of solving $\mathcal{L}_{PAC}(\mathcal{C}, \epsilon - 1/\sqrt{d})$ with access to $\mathsf{STAT}^{(k)}(1/\sqrt{d})$ and success probability $1 - \delta$ is at least $(1 - \delta) \cdot \sqrt{d} - 1$.*

To lower bound the statistical dimension we will use the following "average correlation" parameter introduced in [26].

▶ **Definition 3.10** (k-wise average correlation)**.** *Let k be any positive integer. Let $\mathcal{D}$ be a set of distributions and D_0 a reference distribution over X. Assume that the support of every distribution $D \in \mathcal{D}$ is a subset of the support of D_0. Then, for every $x \in X^k$, define $\hat{D}(x) \doteq \frac{D^k(x)}{D_0^k(x)} - 1$. Then, the k-wise average correlation is defined as*

$$\rho^{(k)}(\mathcal{D}, D_0) \doteq \frac{1}{|\mathcal{D}|^2} \cdot \sum_{D, D' \in \mathcal{D}} |D_0^k[\hat{D} \cdot \hat{D}']|.$$

Lemma 3.11 relates the average correlation to the average discrimination (from Definition 3.8).

▶ **Lemma 3.11** ([25])**.** *Let k be any positive integer. Let $\mathcal{D}$ be a set of distributions and D_0 a reference distribution over X. Let μ be the uniform distribution over $\mathcal{D}$. Then,*

$$\bar{\kappa}^{(k)}_1(\mu, D_0) \leq 4 \cdot \sqrt{\rho^{(k)}(\mathcal{D}, D_0)}.$$

3.2.2 Proof of Lemma 3.2

Denote $X \doteq \mathbb{F}_p^\ell \times \{\pm 1\}$. Let $\mathcal{D}$ be the set of all distributions over X^k that are obtained by sampling from any given distribution over $(\mathbb{F}_p^\ell)^k$ and labeling the k samples according to any given hyperplane indicator function f_a. Let D_0 be the uniform distribution over X^k. We now show that $\mathsf{cRSD}_{\bar{\kappa}_1}(\mathcal{B}(\mathcal{D}, D_0)) = \Omega\big(p^{(\ell-k)/2}\big)$. By definition,

$$\mathsf{cRSD}_{\bar{\kappa}_1}(\mathcal{B}(\mathcal{D}, D_0)) \doteq \sup_{\mu \in S^{\mathcal{D}}} (\bar{\kappa}_1(\mu, D_0))^{-1}.$$

We now choose the distribution μ. For $a \in \mathbb{F}_p^\ell$, we define P_a to be the distribution over $\mathbb{F}_p^\ell$ that has density $\alpha = 1/(2(p^\ell - p^{\ell-1}))$ on each of the $p^\ell - p^{\ell-1}$ points outside Hyp_a, and density $\beta = 1/p^{\ell-1} - \alpha p + \alpha = 1/(2p^{\ell-1})$ on each of the $p^{\ell-1}$ points inside Hyp_a. We then define D_a to be the distribution obtained by sampling k i.i.d. random examples of Hyp_a, the marginal of each over $\mathbb{F}_p^\ell$ being P_a. Let $\mathcal{D}' \doteq \{D_a \mid a \in \mathbb{F}_p^\ell\}$, and let μ be the uniform distribution over $\mathcal{D}'$. By Lemma 3.11, we have that $\bar{\kappa}_1(\mu, D_0) \leq 4 \cdot \sqrt{\rho(\mathcal{D}, D_0)}$, so it is enough to upper bound $\rho(\mathcal{D}, D_0)$.

We first note that for $a, a' \in \mathbb{F}_p^\ell$, we have

$$\begin{aligned}
D_0[\hat{D}_a \cdot \hat{D}_{a'}] &= \mathbb{E}_{(z,b)\sim D_0}[\hat{D}_a(z,b) \cdot \hat{D}_{a'}(z,b)] \\
&= \mathbb{E}_{(z,b)\sim D_0}\left[\left(\frac{D_a(z,b)}{D_0(z,b)} - 1\right) \cdot \left(\frac{D_{a'}(z,b)}{D_0(z,b)} - 1\right)\right] \\
&= \mathbb{E}_{(z,b)\sim D_0}\left[\frac{D_a(z,b) \cdot D_{a'}(z,b)}{D_0^2(z,b)} - \frac{D_a(z,b)}{D_0(z,b)} - \frac{D_{a'}(z,b)}{D_0(z,b)} + 1\right] \\
&= \mathbb{E}_{(z,b)\sim D_0}\left[\frac{D_a(z,b) \cdot D_{a'}(z,b)}{D_0^2(z,b)}\right] - 2 \cdot \mathbb{E}_{(z,b)\sim D_0}\left[\frac{D_a(z,b)}{D_0(z,b)}\right] + 1 \\
&= 2^{2k} \cdot p^{2k\ell} \cdot \mathbb{E}_{(z,b)\sim D_0}[D_a(z,b) \cdot D_{a'}(z,b)] \\
&\quad - 2^{k+1} \cdot p^{k\ell} \cdot \mathbb{E}_{(z,b)\sim D_0}[D_a(z,b)] + 1
\end{aligned}$$

We now compute each of the two expectations that appear in the last equation above.

▶ **Proposition 3.12.** *For every $a \in \mathbb{F}_p^\ell$,*

$$\mathbb{E}_{(z,b)\sim D_0}[D_a(z,b)] = \frac{1}{2^k} \cdot \left(\frac{1}{p} \cdot \beta + \left(1 - \frac{1}{p}\right) \cdot \alpha\right)^k = \frac{1}{2^k \cdot p^{k\cdot\ell}}.$$

The proof of Proposition 3.12 appears in the appendix.

▶ **Proposition 3.13.** *For every $a, a' \in \mathbb{F}_p^\ell$,*

$$\mathbb{E}_{(z,b)\sim D_0}[D_a(z,b) \cdot D_{a'}(z,b)] = \begin{cases} \frac{1}{2^k} \cdot (\frac{1}{p} \cdot \beta^2 + (1 - \frac{1}{p}) \cdot \alpha^2)^k & \text{if } \mathsf{Hyp}_a = \mathsf{Hyp}_{a'}, \\ \frac{1}{2^k} \cdot (\alpha^2 \cdot (1 - \frac{2}{p}))^k & \text{if } \mathsf{Hyp}_a \cap \mathsf{Hyp}_{a'} = \emptyset, \\ \frac{1}{2^k} \cdot (\frac{\beta^2}{p^2} + \alpha^2 \cdot (1 - \frac{2}{p} + \frac{1}{p^2}))^k & \text{otherwise.} \end{cases}$$

The proof of Proposition 3.13 appears in the appendix. Using Proposition 3.12 and Proposition 3.13, we now compute $D_0[\hat{D}_a \cdot \hat{D}_{a'}]$.

▶ **Proposition 3.14.** *For every $a, a' \in \mathbb{F}_p^\ell$,*

$$D_0[\hat{D}_a \cdot \hat{D}_{a'}] = \begin{cases} (p + 1 - \frac{1}{p-1})^k - 1 & \text{if } \mathsf{Hyp}_a = \mathsf{Hyp}_{a'}, \\ \frac{1}{2^k} \cdot \frac{(1-\frac{2}{p})^k}{(1-\frac{1}{p})^{2k}} - 1 & \text{if } \mathsf{Hyp}_a \cap \mathsf{Hyp}_{a'} = \emptyset, \\ 0 & \text{otherwise.} \end{cases}$$

The proof of Proposition 3.14 appears in the appendix. When computing $\rho(\mathcal{D}, D_0)$, we will also use the following simple proposition.

▶ **Proposition 3.15.**

1. *The number of pairs $(a, a') \in (\mathbb{F}_p^\ell)^2$ such that $\mathsf{Hyp}_a = \mathsf{Hyp}_{a'}$ is equal to p^ℓ.*
2. *The number of pairs $(a, a') \in (\mathbb{F}_p^\ell)^2$ such that Hyp_a and $\mathsf{Hyp}_{a'}$ are distinct and parallel is equal to $p^\ell \cdot (p-1)$.*

3. *The number of pairs* $(a, a') \in (\mathbb{F}_p^\ell)^2$ *such that* Hyp_a *and* $\mathsf{Hyp}_{a'}$ *are distinct and intersecting is equal to* $p^{2\cdot\ell} - p^{\ell+1}$.

Using Proposition 3.14 and Proposition 3.15, we are now ready to compute $\rho(\mathcal{D}, D_0)$ as follows

$$\begin{aligned}\rho(\mathcal{D}, D_0) &\leq \frac{1}{p^{2\cdot\ell}} \cdot \left[p^\ell \cdot (p + 1 - \frac{1}{p-1})^k + p^\ell \cdot (p-1) + p^{2\cdot\ell} \cdot 0\right] \\ &\leq O\left(\frac{1}{p^{\ell-k}}\right) + \frac{1}{p^{\ell-1}} \\ &= O\left(\frac{1}{p^{\ell-k}}\right),\end{aligned}$$

where we used above the assumption that $k = O(p)$. We deduce that $\bar{\kappa}_1(\mu, D_0) = O\left(1/p^{(\ell-k)/2}\right)$, and hence $\mathsf{cRSD}_{\bar{\kappa}_1}(\mathcal{B}(\mathcal{D}, D_0)) = \Omega\left(p^{(\ell-k)/2}\right)$. This lower bound on $\mathsf{cRSD}_{\bar{\kappa}_1}(\mathcal{B}(\mathcal{D}, D_0))$, along with Definition 3.9, Theorem 5 and the fact that D_0 has Bayes error rate equal to $1/2$, imply Lemma 3.2.

4 Reduction for flat distributions

To prove Theorem 2 we use the characterization of the SQ complexity of the problem of estimating $D^k[\phi]$ for $D \in \mathcal{D}$ using a notion of statistical dimension from [25]. Specifically, we use the characterization of the complexity of solving this problem using unary SQs and also the generalization of this characterization that characterizes the complexity of solving a problem using k-wise SQs. The latter is equal to 1 (since a single k-wise SQ suffices to estimate $D^k[\phi]$). Hence the k-wise statistical dimension is also equal to 1. We then upper bound the unary statistical dimension by the k-wise statistical dimension. The characterization then implies that an upper bound on the unary statistical dimension gives an upper bound on the SQ complexity of estimating $D^k[\phi]$.

We also give a slightly different way to define flatness that makes it easier to extend our results to other notions of divergence.

▶ **Definition 4.1.** *Let* $\mathcal{D}$ *be a set of distributions over* X. *Define*

$$R_\infty(\mathcal{D}) \doteq \inf_{\bar{D} \in S^X} \sup_{D \in \mathcal{D}} \mathrm{D}_\infty(D \| \bar{D}),$$

where S^X *denotes the set of all probability distributions over* X *and*

$$\mathrm{D}_\infty(D \| \bar{D}) \doteq \sup_{y \in X} \ln \frac{\Pr_{x \sim D}[x = y]}{\Pr_{x \sim \bar{D}}[x = y]}$$

denotes the max-divergence. We say that $\mathcal{D}$ *is* γ*-flat if* $R_\infty(\mathcal{D}) \leq \ln \gamma$.

For simplicity, we will start by relating the k-wise SQ complexity to unary SQ complexity for decision problems. The statistical dimension for this type of problems is substantially simpler than for the general problems but is sufficient to demonstrate the reduction. We then build on the results for decision problems to obtain the proof of Theorem 2.

4.1 Decision problems

The k-wise generalization of the statistical dimension for decision problems from [25] is defined as follows.

▶ **Definition 4.2.** *Let k be any positive integer. Consider a set of distributions $\mathcal{D}$ and a reference distribution D_0 over X. Let μ be a probability measure over $\mathcal{D}$ and let $\tau > 0$. The k-wise maximum covered μ-fraction is defined as*

$$\kappa_1\text{-}\mathsf{frac}^{(k)}(\mu, D_0, \tau) \doteq \sup_{\phi: X^k \to [-1,+1]} \left\{ \Pr_{D\sim\mu}[|D^k[\phi] - D_0^k[\phi]| > \tau] \right\}.$$

▶ **Definition 4.3** (k-wise randomized statistical dimension of decision problems)**.** *Let k be any positive integer. For any set of distributions $\mathcal{D}$, a reference distribution D_0 over X and $\tau > 0$, we define*

$$\mathsf{RSD}^{(k)}_{\kappa_1}(\mathcal{B}(\mathcal{D}, D_0), \tau) \doteq \sup_{\mu \in S^{\mathcal{D}}} (\kappa_1\text{-}\mathsf{frac}^{(k)}(\mu, D_0, \tau))^{-1},$$

where $S^{\mathcal{D}}$ denotes the set of all probability distributions over $\mathcal{D}$.

As shown in [25], RSD tightly characterizes the randomized statistical query complexity of solving the problem using k-wise queries. As observed before, the k-wise versions below are implied by the unary version in [25] simply by defining the domain to be $X' \doteq X^k$ and the set of input distributions to be $\mathcal{D}' \doteq \{D^k \mid D \in \mathcal{D}\}$.

▶ **Theorem 6** ([25])**.** *Let $\mathcal{B}(\mathcal{D}, D_0)$ be a decision problem, $\tau > 0, \delta \in (0, 1/2)$, $k \in \mathbb{N}$ and $d = \mathsf{RSD}^{(k)}_{\kappa_1}(\mathcal{B}(\mathcal{D}, D_0), \tau)$. Then there exists a randomized algorithm that solves $\mathcal{B}(\mathcal{D}, D_0)$ with success probability $\geq 1 - \delta$ using $d \cdot \ln(1/\delta)$ queries to $\mathsf{STAT}^{(k)}_D(\tau/2)$. Conversely, any algorithm that solves $\mathcal{B}(\mathcal{D}, D_0)$ with success probability $\geq 1 - \delta$ requires at least $d \cdot (1 - 2\delta)$ queries to $\mathsf{STAT}^{(k)}_D(\tau)$.*

We will also need the following dual formulation of the statistical dimension given in Theorem 4.3.

▶ **Lemma 4.4** ([25])**.** *Let k be any positive integer. For any set of distributions $\mathcal{D}$, a reference distribution D_0 over X and $\tau > 0$, the statistical dimension $\mathsf{RSD}^{(k)}_{\kappa_1}(\mathcal{B}(\mathcal{D}, D_0), \tau)$ is equal to the smallest d for which there exists a distribution $\mathcal{P}$ over functions from X^k to $[-1, +1]$ such that for every $D \in \mathcal{D}$,*

$$\Pr_{\phi\sim\mathcal{P}}[|D^k[\phi] - D_0^k[\phi]| > \tau] \geq \frac{1}{d}.$$

We can now state the relationship between $\mathsf{RSD}^{(k)}_{\kappa_1}$ and $\mathsf{RSD}^{(1)}_{\kappa_1}$ for any γ-flat $\mathcal{D}$.

▶ **Lemma 4.5.** *Let $\gamma \geq 1$, $\tau > 0$ and $k \in \mathbb{N}$. Let X be a domain, $\mathcal{D}$ be a γ-flat class of distributions over X and D_0 be any distribution over X. Then*

$$\mathsf{RSD}^{(1)}_{\kappa_1}(\mathcal{B}(\mathcal{D}, D_0), \tau/(2k)) \leq \frac{4k \cdot \gamma^{k-1}}{\tau} \cdot \mathsf{RSD}^{(k)}_{\kappa_1}(\mathcal{B}(\mathcal{D}, D_0), \tau).$$

Proof. Let $d \doteq \mathsf{RSD}^{(k)}_{\kappa_1}(\mathcal{B}(\mathcal{D}, D_0), \tau)$. Fact 4.4 implies the existence of a distribution $\mathcal{P}$ over k-wise functions such that for every $D \in \mathcal{D}$,

$$\Pr_{\phi\sim\mathcal{P}}[|D^k[\phi] - D_0^k[\phi]| > \tau] \geq \frac{1}{d}.$$

We now fix D and let ϕ be such that $|D^k[\phi] - D_0^k[\phi]| > \tau$.

By the standard hybrid argument,

$$\mathbb{E}_{j\sim[k]}\left[\left|D^j D_0^{k-j}[\phi] - D^{j-1} D_0^{k-j+1}[\phi]\right|\right] > \frac{\tau}{k}, \tag{6}$$

where $j \sim [k]$ denotes a random and uniform choice of j from $[k]$. This implies that

$$\mathbb{E}_{j\sim[k]}\mathbb{E}_{x_{<j}\sim D^{j-1}}\mathbb{E}_{x_{>j}\sim D_0^{k-j}}\left[\left|D[\phi(x_{<j},\cdot,x_{>j})] - D_0[\phi(x_{<j},\cdot,x_{>j})]\right|\right] > \frac{\tau}{k}.$$

By an averaging argument (and using the fact that ϕ takes values between -1 and $+1$), we get that with probability at least $\tau/(4\cdot k)$ over the choice of $j \sim [k]$, $x_{<j} \sim D^{j-1}$ and $x_{>j} \sim D_0^{k-j}$, we have that

$$\left|D[\phi(x_{<j},\cdot,x_{>j})] - D_0[\phi(x_{<j},\cdot,x_{>j})]\right| > \frac{\tau}{2\cdot k}.$$

Since $\mathcal{D}$ is a γ-flat class of distributions, there exists a (fixed) distribution $\bar{D}$ over X such that for every measurable event $E \subset X$, $\Pr_{x\sim D}[x \in E] \leq \gamma \cdot \Pr_{x\sim\bar{D}}[x \in E]$. Thus, we can replace the unknown input distribution D by the distribution $\bar{D}$ and get that, with probability at least $\tau/(4\cdot k\cdot\gamma^{k-1})$ over the choice of $j \sim [k]$, $x_{<j} \sim \bar{D}^{j-1}$ and $x_{>j} \sim D_0^{k-j}$, we have

$$\left|D[\phi(x_{<j},\cdot,x_{>j})] - D_0[\phi(x_{<j},\cdot,x_{>j})]\right| > \frac{\tau}{2\cdot k}. \tag{7}$$

We now consider the following distribution $\mathcal{P}'$ over unary SQ functions (i.e., over $[-1,+1]^X$): Independently sample ϕ from $\mathcal{P}$, j uniformly from $[k]$, $x_{<j} \sim \bar{D}^{j-1}$ and $x_{>j} \sim D_0^{k-j}$, and output the (unary) function $\phi'(x) = \phi(x_{<j}, x, x_{>j})$. Then, for every $D \in \mathcal{D}$, we have that with probability at least $\frac{1}{d}\cdot\frac{\tau}{4k}\cdot\frac{1}{\gamma^{k-1}}$ over the choice of ϕ' from $\mathcal{P}'$, we have that $|D[\phi'] - D_0[\phi']| > \tau/(2\cdot k)$. Thus, by Fact 4.4

$$\mathsf{RSD}^{(1)}_{\kappa_1}\left(\mathcal{B}(\mathcal{D},D_0),\frac{\tau}{2\cdot k}\right) \leq \frac{4d\cdot\gamma^{k-1}\cdot k}{\tau}.$$ ◀

Lemma 4.5 together with the characterization in Theorem 6 imply the following upper bound on the SQ complexity of a decision problem in terms of its k-wise SQ complexity.

▶ **Theorem 7.** *Let $\gamma \geq 1$, $\tau > 0$ and $k \in \mathbb{N}$. Let X be a domain, $\mathcal{D}$ be a γ-flat class of distributions over X and D_0 be any distribution over X. If there exists an algorithm that, with probability at least $2/3$ solves $\mathcal{B}(\mathcal{D},D_0)$ using t queries to $\mathsf{STAT}^{(k)}_D(\tau)$, then for every $\delta > 0$, there exists an algorithm that, with probability at least $1-\delta$ solves $\mathcal{B}(\mathcal{D},D_0)$ using $t\cdot 12k\cdot\gamma^{k-1}\cdot\ln(1/\delta)/\tau$ queries to $\mathsf{STAT}^{(1)}_D(\tau/(4k))$.*

4.2 General problems

We now define the general class of problems over sets of distributions and a notion of statistical dimension for these types of problems.

▶ **Definition 4.6** (Search problems). *A search problem $\mathcal{Z}$ over a class $\mathcal{D}$ of distributions and a set $\mathcal{F}$ of solutions is a mapping $\mathcal{Z}: \mathcal{D} \to 2^{\mathcal{F}} \setminus \{\emptyset\}$, where $2^{\mathcal{F}}$ denotes the set of all subsets of $\mathcal{F}$. Specifically, for every distribution $D \in \mathcal{D}$, $\mathcal{Z}(D) \subseteq \mathcal{F}$ is the (non-empty) set of valid solutions for D. For a solution $f \in \mathcal{F}$, we denote by $\mathcal{Z}_f$ the set of all distributions for which f is a valid solution.*

▶ **Definition 4.7** (Statistical dimension for search problems [25]). *For $\tau > 0$, $k \in \mathbb{N}$, a domain X and a search problem $\mathcal{Z}$ over a class of distributions $\mathcal{D}$ over X and a set of solutions $\mathcal{F}$, we define the k-*wise statistical dimension *with κ_1-discrimination τ of $\mathcal{Z}$ as*

$$\mathsf{SD}^{(k)}_{\kappa_1}(\mathcal{Z},\tau) \doteq \sup_{D_0\in S^X}\inf_{f\in\mathcal{F}}\mathsf{RSD}^{(k)}_{\kappa_1}(\mathcal{B}(\mathcal{D}\setminus\mathcal{Z}_f,D_0),\tau),$$

where S^X denotes the set of all probability distributions over X.

Lemma 8 lower-bounds the deterministic k-wise SQ complexity of a search problem in terms of its (k-wise) statistical dimension.

▶ **Theorem 8** ([25]). *Let $\mathcal{Z}$ be a search problem, $\tau > 0$ and $k \in \mathbb{N}$. The deterministic k-wise SQ complexity of solving $\mathcal{Z}$ with access to $\mathsf{STAT}^{(k)}(\tau)$ is at least $\mathsf{SD}^{(k)}_{\kappa_1}(\mathcal{Z}, \tau)$.*

The following theorem from [25] gives an upper bound on the SQ complexity of a search problem in terms of its statistical dimension. It relies on the multiplicative weights update method to reconstruct the unknown distribution sufficiently well for solving the problem. The use of this algorithm introduces dependence on KL-radius of $\mathcal{D}$. Namely, we define

$$R_{\mathrm{KL}}(\mathcal{D}) \doteq \inf_{\bar{D} \in S^X} \sup_{D \in \mathcal{D}} \mathrm{KL}(D \| \bar{D}),$$

where $\mathrm{KL}(\cdot\|\cdot)$ denotes the KL-divergence.

▶ **Theorem 9** ([25]). *Let $\mathcal{Z}$ be a search problem, $\tau, \delta > 0$ and $k \in \mathbb{N}$. There is a randomized k-wise SQ algorithm that solves $\mathcal{Z}$ with success probability $1 - \delta$ using*

$$O\left(\mathsf{SD}^{(k)}_{\kappa_1}(\mathcal{Z}, \tau) \cdot \frac{R_{\mathrm{KL}}(\mathcal{D})}{\tau^2} \cdot \log\left(\frac{R_{\mathrm{KL}}(\mathcal{D})}{\tau \cdot \delta}\right)\right)$$

queries to $\mathsf{STAT}^{(k)}(\tau/3)$.

Note that KL-divergence between two distributions is upper-bounded (and is usually much smaller) than the max-divergence we used in the definition of γ-flatness. Specifically, if $\mathcal{D}$ is γ-flat then $R_{\mathrm{KL}}(\mathcal{D}) \leq \ln \gamma$. We are now ready to prove Theorem 2 which we restate here for convenience.

▶ **Theorem 2** (restated). *Let $\gamma \geq 1$, $\tau > 0$ and k be any positive integer. Let X be a domain and $\mathcal{D}$ be a γ-flat class of distributions over X. There exists a randomized algorithm that given any $\delta > 0$ and a k-ary function $\phi : X^k \to [-1, 1]$, estimates $D^k[\phi]$ within τ for every (unknown) $D \in \mathcal{D}$ with success probability at least $1 - \delta$ using*

$$\tilde{O}\left(\frac{\gamma^{k-1} \cdot k^3}{\tau^3} \cdot \log(1/\delta)\right)$$

queries to $\mathsf{STAT}^{(1)}_D(\tau/(6 \cdot k))$.

Proof. We first observe that the task of estimating $D^k[\phi]$ up to additive τ can be viewed as a search problem $\mathcal{Z}$ over the set $\mathcal{D}$ of distributions and over the class $\mathcal{F}$ of solutions that corresponds to the interval $[-1, +1]$. Next, observe that one can easily estimate $D^k[\phi]$ up to additive τ using a single query to $\mathsf{STAT}^{(k)}_D(\tau)$. Lemma 8 implies that $\mathsf{SD}^{(k)}_{\kappa_1}(\mathcal{Z}, \tau) = 1$. By Definition 4.7, for every $D_1 \in S^X$, there exists $f \in \mathcal{F}$, such that $\mathsf{RSD}^{(k)}_{\kappa_1}(\mathcal{B}(\mathcal{D} \setminus \mathcal{Z}_f, D_1), \tau) = 1$. By Lemma 4.5,

$$\mathsf{RSD}^{(1)}_{\kappa_1}\left(\mathcal{B}(\mathcal{D} \setminus \mathcal{Z}_f, D_1), \frac{\tau}{2 \cdot k}\right) \leq \frac{4 \cdot \gamma^{k-1} \cdot k}{\tau}.$$

Thus, Fact 4.4 and Definition 4.7 imply that

$$\mathsf{SD}^{(1)}_{\kappa_1}(\mathcal{Z}, \frac{\tau}{2 \cdot k}) \leq \frac{4 \cdot \gamma^{k-1} \cdot k}{\tau}.$$

Applying Lemma 9, we conclude that there exists a randomized unary SQ algorithm that solves $\mathcal{Z}$ with probability at least $1 - \delta$ using at most

$$O\left(\gamma^{k-1} \cdot k^3 \cdot \frac{R_{\mathrm{KL}}(\mathcal{D})}{\tau^3} \cdot \log\left(\frac{k \cdot R_{\mathrm{KL}}(\mathcal{D})}{\tau \cdot \delta}\right)\right)$$

queries to $\mathsf{STAT}^{(1)}(\tau/(6\cdot k))$. This – along with the fact that $R_{\mathrm{KL}}(\mathcal{D}) \leq \ln(\gamma)$ whenever $\mathcal{D}$ is a γ-flat set of distributions – concludes the proof of Theorem 2. ◀

4.2.1 Other divergences

While the max-divergence that we used for measuring flatness suffices for the applications we give in this paper (and is relatively simple), it might be too conservative in other problems. For example, such divergence is infinite even for two Gaussian distributions with the same standard deviation but different means. A simple way to obtain a more robust version of our reduction is to use approximate max-divergence. For $\delta \in [0,1)$ it is defined as:

$$\mathrm{D}^{\delta}_{\infty}(D\|\bar{D}) \doteq \ln \sup_{E\subseteq X} \frac{\Pr_{x\sim D}[x\in E]-\delta}{\Pr_{x\sim \bar{D}}[x\in E]}.$$

Note that $\mathrm{D}^{0}_{\infty}(D\|\bar{D}) = \mathrm{D}_{\infty}(D\|\bar{D})$. Similarly, we can define a radius of $\mathcal{D}$ in this divergence

$$R^{\delta}_{\infty}(\mathcal{D}) \doteq \inf_{\bar{D}\in S^X} \sup_{D\in\mathcal{D}} \mathrm{D}^{\delta}_{\infty}(D\|\bar{D}).$$

Now, it is easy to see that, if $\mathrm{D}^{\delta}_{\infty}(D\|\bar{D}) \leq r$ then $\mathrm{D}^{k\delta}_{\infty}(D^k\|\bar{D}^k) \leq kr$. This means that if in the proof of Lemma 4.5 we use the condition $R^{\tau/(8k^2)}_{\infty}(\mathcal{D}) \leq \ln\gamma$ instead of γ-flatness then we will obtain that the event in Equation (7) holds with probability at least

$$\left(\frac{\tau}{4k} - (k-1)\cdot\frac{\tau}{8k^2}\right)/\gamma^{k-1} \geq \frac{\tau}{\gamma^{k-1}\cdot 8k}$$

over the same random choices.

This implies the following generalization of Theorem 2.

▶ **Theorem 10.** *Let $\tau > 0$ and k be any positive integer. Let $\mathcal{D}$ be a class of distributions over a domain X and $\gamma = \exp(R^{\tau/(8k^2)}_{\infty}(\mathcal{D}))$. There exists a randomized algorithm that given any $\delta > 0$ and a k-ary function $\phi : X^k \to [-1,1]$, estimates $D^k[\phi]$ within τ for every (unknown) $D \in \mathcal{D}$ with success probability at least $1-\delta$ using*

$$\tilde{O}\left(\frac{\gamma^{k-1}\cdot k^3\cdot R_{\mathrm{KL}}(\mathcal{D})}{\tau^3}\cdot\log(1/\delta)\right)$$

queries to $\mathsf{STAT}^{(1)}_{D}(\tau/(6\cdot k))$.

An alternative approach is to use Renyi divergence of order $\alpha > 1$ defined as follows:

$$\mathrm{D}_{\alpha}(D\|\bar{D}) \doteq \frac{1}{1-\alpha}\cdot\ln\left(\mathbb{E}_{y\sim D}\left[\left(\frac{\Pr_{x\sim D}[x=y]}{\Pr_{x\sim\bar{D}}[x=y]}\right)^{\alpha-1}\right]\right).$$

The corresponding radius is defined as

$$R_{\alpha}(\mathcal{D}) \doteq \inf_{\bar{D}\in S^X} \sup_{D\in\mathcal{D}} \mathrm{D}_{\alpha}(D\|\bar{D}).$$

To use it in our application we need the standard property of the Renyi divergence for product distributions $\mathrm{D}_{\alpha}(D^k\|\bar{D}^k) = k\cdot\mathrm{D}_{\alpha}(D\|\bar{D})$ and also the following simple lemma from [33, Lemma 1]:

▶ **Lemma 4.8.** *For $\alpha > 1$, any two distributions $D, \bar{D}$ over X and an event $E \subseteq X$:*

$$\Pr_{x\sim D}[x\in E] \leq \left(\exp(\mathrm{D}_{\alpha}(D\|\bar{D}))\cdot\Pr_{x\sim\bar{D}}[x\in E]\right)^{\frac{\alpha-1}{\alpha}}.$$

We will need the inverted version of this lemma:

$$\Pr_{x\sim\bar{D}}[x \in E] \geq \frac{(\Pr_{x\sim D}[x \in E])^{\frac{\alpha}{\alpha-1}}}{\exp(\mathrm{D}_\alpha(D\|\bar{D}))}.$$

Applying this in the proof of Lemma 4.5 for $\gamma = \exp(R_\alpha(\mathcal{D}))$, we obtain that the event in Equation (7) holds with probability at least

$$\left(\frac{\tau}{4k}\right)^{\frac{\alpha}{\alpha-1}} / \gamma^{k-1}.$$

This gives the following generalization of Theorem 2.

▶ **Theorem 11.** *Let $\tau > 0, \alpha > 1$ and k be any positive integer. Let $\mathcal{D}$ be a class of distributions over a domain X and $\gamma = \exp(R_\alpha(\mathcal{D}))$. There exists a randomized algorithm that given any $\delta > 0$ and a k-ary function $\phi : X^k \to [-1, 1]$, estimates $D^k[\phi]$ within τ for every (unknown) $D \in \mathcal{D}$ with success probability at least $1 - \delta$ using*

$$\tilde{O}\left(\gamma^{k-1} \cdot \left(\frac{k}{\tau}\right)^{2+\frac{\alpha}{\alpha-1}} \cdot \log(1/\delta)\right)$$

queries to $\mathsf{STAT}_D^{(1)}(\tau/(6 \cdot k))$.

4.3 Applications to solving CSPs and learning DNF

We now give some examples of the application of our reduction to obtain lower bounds against k-wise SQ algorithms. Our applications for stochastic constraint satisfaction problems (CSPs) and DNF learning. We start with the definition of a stochastic CSP with a *planted solution* which is a pseudo-random generator based on Goldreich's proposed one-way function [29].

▶ **Definition 12.** Let $t \in \mathbb{N}$ and $P : \{\pm 1\}^t \to \{\pm 1\}$ be a fixed predicate. We are given access to samples from a distribution P_σ, corresponding to a ("planted") assignment $\sigma \in \{\pm 1\}^n$. A sample from this distribution is a uniform-random t-tuple $(i_1, \ldots, i_t)$ of distinct variable indices along with the value $P(\sigma_{i_1}, \ldots, \sigma_{i_t})$. The goal is to recover the assignment σ when given m independent samples from P_σ. A (potentially) easier problem is to distinguish any such planted distribution from the distribution U_t in which the value is an independent uniform-random coin flip (instead of $P(\sigma_{i_1}, \ldots, \sigma_{i_t})$).

We say that a predicate $P : \{\pm 1\}^t \to \{\pm 1\}$ has complexity r if r is the degree of the lowest-degree non-zero Fourier coefficient of P. It can be as large as t (for the parity function). A lower bound on the (unary) SQ complexity of solving such CSPs was shown by [28] (their result is for the stronger VSTAT oracle but here we state the version for the STAT oracle).

▶ **Theorem 13** ([28]). *Let $t, q \in \mathbb{N}$ and $P : \{\pm 1\}^t \to \{\pm 1\}$ be a fixed predicate of complexity r. Then for any $q > 0$, any algorithm that, given access to a distribution $D \in \{P_\sigma \mid \sigma \in \{\pm 1\}^n\} \cup \{U_t\}$ decides correctly whether $D = P_\sigma$ or $D = U_t$ with probability at least $2/3$ needs $q/2^{O(t)}$ queries to $\mathsf{STAT}_D^{(1)}\left(\left(\frac{\log q}{n}\right)^{r/2}\right)$.*

The set of input distributions in this problem is 2-flat relative to U_t and it is one-to-many decision problem. Hence Theorem 7 implies[2] the following lower bound for k-wise SQ algorithms.

[2] We can also get essentially the same result by applying the simulation of a k-wise SQ using unary SQs from Theorem 2.

▶ **Theorem 14.** *Let $t \in \mathbb{N}$ and $P : \{\pm 1\}^t \to \{\pm 1\}$ be a fixed predicate of complexity r. Then for any $\alpha > 0$, any algorithm that, given access to a distribution $D \in \{P_\sigma \mid \sigma \in \{\pm 1\}^n\} \cup \{U_t\}$ decides correctly whether $D = P_\sigma$ or $D = U_t$ with probability at least $2/3$ needs $2^{n^{1-\alpha}-O(t)}$ queries to $\mathsf{STAT}_D^{(n^{1-\alpha})}\left((2/n^\alpha)^{r/2} \cdot n^{1-\alpha}/4\right)$.*

Proof. Let $\mathcal{A}$ be a k-wise SQ algorithm using q' queries to $\mathsf{STAT}_D^{(n^{1-\alpha})}\left((2/n^\alpha)^{r/2} \cdot n^{1-\alpha}/6\right)$ which solves the problem with success probability $2/3$. We let $k = n^{1-\alpha}$ and apply Theorem 7 to obtain an algorithm that uses unary SQs and solves the problem with success probability $2/3$. This algorithm uses $q_0 = q' \cdot 2^{n^{1-\alpha}} \cdot n^{O(r)}$ queries to $\mathsf{STAT}_D^{(1)}\left((2/n^\alpha)^{r/2}\right)$. Now choosing $q = 2^{2n^{1-\alpha}}$ we get that $\left(\frac{\log q}{n}\right)^{r/2} \leq (2/n^\alpha)^{r/2}$. This means that $q_0 \geq q/2^{O(t)} = 2^{2n^{1-\alpha}-O(t)}$. Hence $q' = 2^{2n^{1-\alpha}-O(t)-n^{1-\alpha}-O(r)} = 2^{n^{1-\alpha}-O(t)}$. ◀

Similar lower bounds can be obtained for other problems considered in [28], namely, planted satisfiability and t-SAT refutation.

To obtain a lower bound for learning DNF formulas we can use a simple reduction from the Goldreich's PRG defined above to learning DNF formulas of polynomial size. It is based on ideas implicit in the reduction from t-SAT refutation to DNF learning from [13].

▶ **Lemma 15.** *$P : \{\pm 1\}^t \to \{\pm 1\}$ be a fixed predicate. There exists a mapping M from t-tuples of indices in $[n]$ to $\{0,1\}^{tn}$ such that for every $\sigma \in \{\pm 1\}^n$ there exists a DNF formula f_σ of size 2^t satisfying $P(\sigma_{i_1}, \ldots, \sigma_{i_t}) = f_\sigma(M(i_1, \ldots, i_t))$.*

Proof. The mapping M maps $(i_1, \ldots, i_t)$ to the concatenation of the indicator vectors of each of the indices. Namely, for $j \in [t]$ and $\ell \in [n]$, $M(i_1, \ldots, i_t)_{j,\ell} = 1$ if and only if $i_j = \ell$, where we use the double index j, ℓ to refer to element $n(j-1)+\ell$ of the vector. Let $v_{j,\ell}$ denote the variable with the index j, ℓ. Let σ be any assignment and we denote by z_j^σ the j-th variable of our predicate P when the assignment is equal to σ. We first observe that $z_j^\sigma \equiv \bigwedge_{\ell \in [n], \sigma_\ell = 0} \bar{v}_{j,\ell}$. This is true since, by definition, the value of the j-th variable of our predicate is σ_{i_j}. This value is 1 if and only if $i_j \notin \{\ell \in [n] \mid \sigma_\ell = 0\}$. This is equivalent to $v_{j,\ell}$ being equal to 0 for all $\ell \in [n]$ such that $\sigma_\ell = 0$. Analogously, $\bar{z}_j^\sigma \equiv \bigwedge_{\ell \in [n], \sigma_\ell = 1} \bar{v}_{j,\ell}$. This implies that any conjunction of variables $z_1^\sigma, \bar{z}_1^\sigma, \ldots, z_t^\sigma, \bar{z}_t^\sigma$ can be expressed as a conjunction over variables $\bar{v}_{j,\ell}$. Any predicate P can be expressed as a disjunction of at most 2^t conjunctions and hence there exists a DNF formula f_σ of size at most 2^t whose value on $M(i_1, \ldots, i_t)$ is equal to $P(\sigma_{i_1}, \ldots, \sigma_{i_t})$. ◀

This reduction implies that by converting a sample $((i_1, \ldots, i_t), b)$ to a sample $(M(i_1, \ldots, i_t), b)$ we can transform the Goldreich's PRG problem into a problem in which our goal is to distinguish examples of some DNF formula f_σ from randomly labeled examples. Naturally, an algorithm that can learn DNF formulas can output a hypothesis which predicts the label (with some non-trivial accuracy), whereas such hypothesis cannot exist for predicting random labels. Hence known SQ lower bounds on planted CSPs [28] immediately imply lower bounds for learning DNF. Further, by applying Lemma 15 together with Thm. 14 for $t = r = \log n$ we obtain the first lower bounds for learning DNF against $n^{1-\alpha}$-wise SQ algorithms.

▶ **Theorem 16.** *For any constant (independent of n) $\alpha > 0$, there exists a constant $\beta > 0$ such that any algorithm that PAC learns DNF formulas of size n with error $< 1/2 - n^{-\beta \log n}$ and success probability at least $2/3$ needs at least $2^{n^{1-\alpha}}$ queries to $\mathsf{STAT}_D^{(n^{1-\alpha})}(n^{-\beta \log n})$.*

We remark that this is a lower bound for PAC learning polynomial size DNF formulas with respect to some fixed (albeit non-uniform) distribution over $\{0,1\}^n$. The approach for relating k-wise SQ complexity to unary SQ complexity given in [9] applies to this setting. Yet, in their proof the tolerance needed for the unary SQ algorithm is $\tau/2^k$ and therefore it would not give a non-trivial lower bounds beyond $k = O(\log n)$.

5 Reduction for low-communication queries

In this section, we prove Theorem 4 using a recent result of Steinhardt, Valiant and Wager [36]. Their result can be seen giving a SQ algorithm that simulates a communication protocol between n parties. Each party is holding a sample drawn i.i.d. from distribution D and broadcasts at most b bits about its sample (to all the other parties). The bits can be sent over multiple rounds. This is essentially the standard model of multi-party communication complexity (*e.g.* [32]) but with the goal of solving some problem about the unknown distribution D rather than computing a specific function of the inputs. Alternatively, one can also see this model as a single algorithm that extracts at most b-bits of information about each random sample from D and is allowed to extract the bits in an arbitrary order (generalizing the b-bit sampling model that we discuss in Section 6.2 and in which b-bits are extracted from each sample at once). We refer to this model simply as algorithms that extract at most b bits per sample.

▶ **Theorem 17** ([36]). *Let $\mathcal{A}$ be an algorithm that uses n samples drawn i.i.d. from a distribution D and extracts at most b bits per sample. Then, for every $\beta > 0$, there is an algorithm $\mathcal{B}$ that makes at most $2 \cdot b \cdot n$ queries to $\mathsf{STAT}_D^{(1)}(\beta/(2^{b+1} \cdot k))$ and the output distributions of $\mathcal{A}$ and $\mathcal{B}$ are within total variation distance β.*

We will use this simulation to estimate the expectation of k-wise functions that have low communication complexity. Specifically, we recall the following standard model of public-coin randomized k-party communication complexity.

▶ **Definition 5.1.** *For a function $\phi : X^k \to \{\pm 1\}$ we say that ϕ has a k-party public-coin randomized communication complexity of at most b bits per party with success probability $1 - \delta$ if there exist a protocol satisfying the following conditions. Each of the parties is given $x_i \in X$ and access to shared random bits. In each round one of the parties can compute one or more bits using its input, random bits and all the previous communication and then broadcast it to all the other parties. In the last round one of the parties computes a bit that is the output of the protocol. Each of the parties communicates at most b bits in total. For every $x_1, \ldots, x_k \in X$, with probability at least $1 - \delta$ over the choice of the random bits the output of the protocol is equal to $\phi(x_1, \ldots, x_k)$.*

We are now ready to prove Theorem 4 which we restate here for convenience.

▶ **Theorem 4** (restated). *Let $\phi : X^k \to \{\pm 1\}$ be a function, and assume that ϕ has k-party public-coin randomized communication complexity of b bits per party with success probability $2/3$. Then, there exists a randomized algorithm that, with probability at least $1 - \delta$, estimates $\mathbb{E}_{x \sim D^k}[\phi(x)]$ within τ using $O(b \cdot k \cdot \log(1/\delta)/\tau^2)$ queries to $\mathsf{STAT}_D^{(1)}(\tau')$ for some $\tau' = \tau^{O(b)}/k$.*

Proof. We first amplify the success probability of the protocol for computing ϕ to $\delta' \doteq \tau/8$ using the majority vote of $O(\log(1/\delta'))$ repetitions. By Yao's minimax theorem [42] there exists a deterministic protocol Π' that succeeds with probability at least $1 - \delta'$ for $(x_1, \ldots, x_k) \sim D^k$. Applying Theorem 17, we obtain a unary SQ algorithm $\mathcal{A}$ whose output

is within total variation distance at most $\beta \doteq \tau/8$ from $\Pi'(x_1, \ldots, x_k)$ (and we can assume that the output of $\mathcal{A}$ is in $\{\pm 1\}$). Therefore:

$$|\mathbb{E}[\mathcal{A}] - D^k[\phi]| \leq |\mathbb{E}[\mathcal{A}] - \mathbb{E}_{D^k}[\Pi'(x_1, \ldots, x_k)]| + |\mathbb{E}_{D^k}[\Pi'(x_1, \ldots, x_k)] - D^k[\phi]| \leq \frac{2\tau}{8} + \frac{2\tau}{8} = \frac{\tau}{2}.$$

Repeating $\mathcal{A}$ $O(\log(1/\delta)/\tau^2)$ times and taking the mean, we get an estimate of $D^k[\phi]$ within τ with probability at least $1-\delta$. This algorithm uses $O(b \cdot k \cdot \log(1/\delta)/\tau^2)$ queries to $\mathsf{STAT}_D^{(1)}(\tau')$ for $\tau' = \frac{\tau}{8}/(2^{O(\log(8/\tau)\cdot b)} \cdot k) = \tau^{O(b)}/k$. ◀

The collision probability for a distribution D is defined as $\Pr_{(x_1,x_2)\sim D^2}[x_1 = x_2]$. This corresponds to $\phi(x_1, x_2)$ being the Equality function which, as is well-known, has randomized 2-party communication complexity of $O(1)$ bits per party with success probability $2/3$ (see, e.g., [32]). Applying Theorem 4 with $k = 2$ we get the following corollary.

▶ **Corollary 18.** *For any $\tau, \delta > 0$, there is a SQ algorithm that estimates the collision probability of an unknown distribution D within τ with success probability $1 - \delta$ using $O(\log(1/\delta)/\tau^2)$ queries to $\mathsf{STAT}_D^{(1)}(\tau^{O(1)})$.*

6 Corollaries for other models

6.1 k-local differential privacy

We start by formally defining the k-wise version of the *local differentially privacy* model from [30].

▶ **Definition 6.1** (k-local randomizer)**.** *A k-local ϵ-differentially private (DP) randomizer is a randomized map $R : X^k \to W$ such that for all $u, u' \in X^k$ and all $w \in W$, we have that $\Pr[R(u) = w] \leq e^\epsilon \cdot \Pr[R(u') = w]$ where the probabilities are taken over the coins of R.*

The following definition gives a k-wise generalization of the local randomizer (LR) oracle which was used in [30].

▶ **Definition 6.2** (k-local Randomizer Oracle)**.** *Let $z = (z_1, \ldots, z_n) \in X^n$ be a database. A k-LR oracle $\mathsf{LR}_z(\cdot, \cdot)$ gets a k-tuple of indices $\bar{i} \in [n]^k$ and a k-local ϵ-DP randomizer as inputs, and outputs an element $w \in W$ which is sampled from the distribution $R(z_{i_1}, \ldots, z_{i_k})$.*

We are now ready to give the definition of k-local differential privacy.

▶ **Definition 6.3** (k-local differentially private algorithm)**.** *A k-local ϵ-differentially private algorithm is an algorithm that accesses a database $z \in X^n$ via a k-LR oracle LR_z with the restriction that for all $i \in [n]$, if $\mathsf{LR}_z(\bar{i}_1, R_1), \ldots, \mathsf{LR}_z(\bar{i}_t, R_t)$ are the algorithm's invocations of LR_z on k-tuples of indices that include index i, where for each $j \in [t]$ R_j is a k-local ϵ_j-DP randomizer, then $\epsilon_1 + \cdots + \epsilon_t \leq \epsilon$.*

The following two theorems – which follow from Theorem 5.7 and Lemma 5.8 of [30] – show that k-local differentially private algorithms are equivalent (up to polynomial factors) to k-wise statistical query algorithms.

▶ **Theorem 19.** *Let $\mathcal{A}_{SQ}$ be a k-wise SQ algorithm that makes at most t queries to $\mathsf{STAT}_D^{(k)}(\tau)$. Then, for every $\beta > 0$, there exists a k-local ϵ-DP algorithm $\mathcal{A}_{DP}$ such that if the database z has $n \geq n_0 = O(k \cdot t \cdot \log(t/\beta)/(\epsilon^2 \cdot \tau^2))$ entries sampled i.i.d. from the distribution D, then $\mathcal{A}_{DP}$ makes n_0/k queries and the total variation between $\mathcal{A}_{DP}$'s and $\mathcal{A}_{SQ}$'s output distributions is at most β.*

▶ **Theorem 20.** *Let $z \in X^n$ be a database with entries drawn i.i.d. from a distribution D. For every k-local ϵ-DP algorithm $\mathcal{A}_{DP}$ making t queries to LR_z and $\beta > 0$, there exists a k-wise statistical query algorithm $\mathcal{A}_{SQ}$ that in expectation makes $O(t \cdot e^{\epsilon})$ queries to $\mathsf{STAT}_D^{(k)}(\tau)$ for $\tau = \Theta(\beta/(e^{2\epsilon} \cdot t))$ such that the total variation between $\mathcal{A}_{SQ}$'s and $\mathcal{A}_{DP}$'s output distributions is at most β.*

By combining Theorem 1, Theorem 19 and Theorem 20 we then obtain the following corollary.

▶ **Corollary 21.** *For every positive integer k and any prime number p, there is a concept class $\mathcal{C}$ of Boolean functions defined over a domain of size p^{k+1} for which there exists a $(k+1)$-local 1-DP distribution-independent PAC learning algorithm using a database consisting of $\widetilde{O}_k(\log p)$ i.i.d. samples, whereas any k-local 1-DP distribution-independent PAC learning algorithm requires $\Omega_k(p^{1/4})$ samples.*

The reduction in Theorem 2 then implies that for γ-flat classes of distributions a k-local DP algorithm can be simulated by a 1-local DP algorithm with an overhead that is linear in γ^{k-1} and polynomial in other parameters.

▶ **Theorem 22.** *Let $\gamma \geq 1$, k be any positive integer. Let X be a domain and $\mathcal{D}$ a γ-flat class of distributions over X. Let $z \in X^n$ be a database with entries drawn i.i.d. from a distribution $D \in \mathcal{D}$. For every k-local ϵ-DP algorithm $\mathcal{A}$ making t queries to a k-LR oracle LR_z and $\beta > 0$, there exists a 1-local ϵ-DP algorithm $\mathcal{B}$ such that if $n \geq n_0 = \tilde{O}\left(\frac{\gamma^{k-1} \cdot t^6 \cdot k^6 \cdot e^{11\epsilon}}{\beta^3 \epsilon^2}\right)$ then for every $D \in \mathcal{D}$, $\mathcal{B}$ makes n_0/k queries to 1-LR oracle LR'_z and the total variation distance between $\mathcal{B}$'s and $\mathcal{A}$'s output distributions is at most β.*

The reduction from Theorem 4 can be translated to this model analogously.

6.2 k-wise b-bit sampling model

For an integer $b > 0$, a b-bit sampling oracle $\mathrm{BS}_D(b)$ is defined as follows: Given any function $\phi : X \to \{0,1\}^b$, $\mathrm{BS}_D(b)$ returns $\phi(x)$ for x drawn randomly and independently from D, where D is the unknown input distribution. This oracle was first studied by Ben-David and Dichterman [4] as a *weak Restricted Focus of Attention* model. They showed that algorithms in this model can be simulated efficiently using statistical queries and vice versa. Lower bounds against algorithms that use such an oracle have been studied in [26, 28]. More recently, motivated by communication constraints in distributed systems, the sample complexity of several basic problems in statistical estimation has been studied in this and related models [43, 37, 36]. These works also study the natural k-wise generalization of this model. Specifically, $\mathrm{BS}_D^{(k)}(b)$ is the oracle that given any function $\phi : X^k \to \{0,1\}^b$, returns $\phi(x)$ for x drawn randomly and independently from D^k.

The following two theorems – which follow from Theorem 5.2 in [4] and Proposition 3 in [36] (that strengthens a similar result in [4]) – show that k-wise algorithms in the b-bit sampling model are equivalent (up to polynomial and 2^b factors) to k-wise statistical query algorithms.

▶ **Theorem 23.** *Let $\mathcal{A}_{SQ}$ be a k-wise SQ algorithm that makes at most t Boolean queries to $\mathsf{STAT}_D^{(k)}(\tau)$. Then, for every $\beta > 0$, there exists a k-wise 1-bit sampling algorithm $\mathcal{A}_{1\text{-}bit}$ that uses $O(\frac{t}{\tau^2} \cdot \log(t/\beta))$ queries to $BS_D^{(k)}(b)$ and the total variation distance between $\mathcal{A}_{SQ}$'s and $\mathcal{A}_{1\text{-}bit}$'s output distributions is at most β.*

▶ **Theorem 24.** *Let $\mathcal{A}_{b\text{-}bit}$ be a k-wise b-bit sampling algorithm that makes at most t queries to $BS_D^{(k)}(b)$. Then, for every $\beta > 0$, there exists a k-wise SQ algorithm $\mathcal{A}_{SQ}$ that makes $2bt$ queries to $\mathsf{STAT}_D^{(k)}(\beta/(2^{b+1}t))$ and the total variation distance between $\mathcal{A}_{SQ}$'s and $\mathcal{A}_{b\text{-}bit}$'s output distributions is at most β.*

Feldman et al. [26] give a tighter correspondence between the BS oracle and the slightly stronger VSTAT oracle. Their simulations can be extended to the k-wise case in a similar way.

The following corollary now follows by combining Theorem 1, Theorem 23 and Theorem 24.

▶ **Corollary 25.** *Let $b = O(1)$. For every positive integer k and any prime number p, there is a concept class $\mathcal{C}$ of Boolean functions defined over a domain of size p^{k+1} for which there exists a $(k+1)$-wise b-bit sampling distribution-independent PAC learning algorithm making $\widetilde{O}_k(\log p)$ queries, whereas any k-wise b-bit sampling distribution-independent PAC learning algorithm requires $\widetilde{\Omega}_k(p^{1/12})$ queries.*

The reduction in Theorem 2 then implies that for γ-flat classes of distributions a k-wise 1-bit sampling algorithm can be simulated by a 1-wise 1-bit sampling algorithm.

▶ **Theorem 26.** *Let $\gamma \geq 1$, k be any positive integer. Let X be a domain and $\mathcal{D}$ a γ-flat class of distributions over X. For every algorithm $\mathcal{A}$ making t queries to $BS_D^{(k)}(1)$ and every $\beta > 0$, there exists a 1-bit sampling algorithm $\mathcal{B}$ that for every $D \in \mathcal{D}$, uses $\tilde{O}\left(\frac{\gamma^{k-1} \cdot t^6 \cdot k^5}{\beta^3}\right)$ queries to $BS_D(1)$ and the total variation distance between $\mathcal{B}$'s and $\mathcal{A}$'s output distributions is at most β.*

References

1. Javed A Aslam and Scott E Decatur. General bounds on statistical query learning and pac learning with noise via hypothesis boosting. In *Foundations of Computer Science, 1993. Proceedings., 34th Annual Symposium on*, pages 282–291. IEEE, 1993.
2. Maria-Florina Balcan, Avrim Blum, Shai Fine, and Yishay Mansour. Distributed learning, communication complexity and privacy. In Shie Mannor, Nathan Srebro, and Robert C. Williamson, editors, *COLT 2012 - The 25th Annual Conference on Learning Theory, June 25-27, 2012, Edinburgh, Scotland*, volume 23 of *JMLR Proceedings*, pages 26.1–26.22. JMLR.org, 2012. URL: `http://www.jmlr.org/proceedings/papers/v23/balcan12a/balcan12a.pdf`.
3. Maria-Florina Balcan and Vitaly Feldman. Statistical active learning algorithms for noise tolerance and differential privacy. *Algorithmica*, 72(1):282–315, 2015. `doi:10.1007/s00453-014-9954-9`.
4. Shai Ben-David and Eli Dichterman. Learning with restricted focus of attention. *J. Comput. Syst. Sci.*, 56(3):277–298, 1998. `doi:10.1006/jcss.1998.1569`.
5. Shai Ben-David, Alon Itai, and Eyal Kushilevitz. Learning by distances. In Mark A. Fulk and John Case, editors, *Proceedings of the Third Annual Workshop on Computational Learning Theory, COLT 1990, University of Rochester, Rochester, NY, USA, August 6-8, 1990.*, pages 232–245. Morgan Kaufmann, 1990. URL: `http://dl.acm.org/citation.cfm?id=92644`.
6. A. Blum, M. Furst, J. Jackson, M. Kearns, Y. Mansour, and S. Rudich. Weakly learning DNF and characterizing statistical query learning using Fourier analysis. In *Proceedings of STOC*, pages 253–262, 1994.

7 Avrim Blum, Cynthia Dwork, Frank McSherry, and Kobbi Nissim. Practical privacy: the sulq framework. In Chen Li, editor, *Proceedings of the Twenty-fourth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 13-15, 2005, Baltimore, Maryland, USA*, pages 128–138. ACM, 2005. doi:10.1145/1065167.1065184.

8 Avrim Blum, Alan Frieze, Ravi Kannan, and Santosh Vempala. A polynomial-time algorithm for learning noisy linear threshold functions. *Algorithmica*, 22(1-2):35–52, 1998.

9 Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM (JACM)*, 50(4):506–519, 2003.

10 D. Boneh and R. Lipton. Amplification of weak learning over the uniform distribution. In *Proceedings of the Sixth Annual Workshop on Computational Learning Theory*, pages 347–351, 1993.

11 Cheng Chu, Sang Kyun Kim, Yi-An Lin, YuanYuan Yu, Gary Bradski, Andrew Y Ng, and Kunle Olukotun. Map-reduce for machine learning on multicore. *Advances in neural information processing systems*, 19:281, 2007.

12 Dana Dachman-Soled, Vitaly Feldman, Li-Yang Tan, Andrew Wan, and Karl Wimmer. Approximate resilience, monotonicity, and the complexity of agnostic learning. In *Proceedings of SODA*, 2015.

13 Amit Daniely and Shai Shalev-Shwartz. Complexity theoretic limitations on learning dnf's. In *COLT*, pages 815–830, 2016. URL: http://jmlr.org/proceedings/papers/v49/daniely16.html.

14 Anindya De, Ilias Diakonikolas, and Rocco A Servedio. Learning from satisfying assignments. In *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 478–497. SIAM, 2015.

15 Ilias Diakonikolas, Daniel M. Kane, and Alistair Stewart. Statistical query lower bounds for robust estimation of high-dimensional gaussians and gaussian mixtures. *CoRR*, abs/1611.03473, 2016. URL: http://arxiv.org/abs/1611.03473.

16 Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In Frank Neven, Catriel Beeri, and Tova Milo, editors, *Proceedings of the Twenty-Second ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, June 9-12, 2003, San Diego, CA, USA*, pages 202–210. ACM, 2003. doi:10.1145/773153.773173.

17 John Dunagan and Santosh Vempala. A simple polynomial-time rescaling algorithm for solving linear programs. In László Babai, editor, *Proceedings of the 36th Annual ACM Symposium on Theory of Computing, Chicago, IL, USA, June 13-16, 2004*, pages 315–320. ACM, 2004. doi:10.1145/1007352.1007404.

18 Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toni Pitassi, Omer Reingold, and Aaron Roth. Generalization in adaptive data analysis and holdout reuse. In *Advances in Neural Information Processing Systems*, pages 2341–2349, 2015.

19 Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Leon Roth. Preserving statistical validity in adaptive data analysis. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pages 117–126. ACM, 2015.

20 Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam D. Smith. Calibrating noise to sensitivity in private data analysis. In Shai Halevi and Tal Rabin, editors, *Theory of Cryptography, Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006, Proceedings*, volume 3876 of *Lecture Notes in Computer Science*, pages 265–284. Springer, 2006. doi:10.1007/11681878_14.

21 Úlfar Erlingsson, Vasyl Pihur, and Aleksandra Korolova. RAPPOR: randomized aggregatable privacy-preserving ordinal response. In *ACM SIGSAC Conference on Computer and Communications Security*, pages 1054–1067, 2014.

22 V. Feldman, H. Lee, and R. Servedio. Lower bounds and hardness amplification for learning shallow monotone formulas. In *COLT*, volume 19, pages 273–292, 2011.

23 Vitaly Feldman. Evolvability from learning algorithms. In *Proceedings of the fortieth annual ACM symposium on Theory of computing*, pages 619–628. ACM, 2008.

24 Vitaly Feldman. Dealing with range anxiety in mean estimation via statistical queries. *arXiv*, abs/1611.06475, 2016. URL: `http://arxiv.org/abs/1611.06475`.

25 Vitaly Feldman. A general characterization of the statistical query complexity. *CoRR*, abs/1608.02198, 2016. URL: `http://arxiv.org/abs/1608.02198`.

26 Vitaly Feldman, Elena Grigorescu, Lev Reyzin, Santosh Vempala, and Ying Xiao. Statistical algorithms and a lower bound for detecting planted cliques. *arXiv, CoRR*, abs/1201.1214, 2012. Extended abstract in STOC 2013.

27 Vitaly Feldman, Cristobal Guzman, and Santosh Vempala. Statistical query algorithms for mean vector estimation and stochastic convex optimization. *CoRR*, abs/1512.09170, 2015. Extended abstract in SODA 2017. URL: `http://arxiv.org/abs/1512.09170`.

28 Vitaly Feldman, Will Perkins, and Santosh Vempala. On the complexity of random satisfiability problems with planted solutions. *CoRR*, abs/1311.4821, 2013. Extended abstract in STOC 2015.

29 Oded Goldreich. Candidate one-way functions based on expander graphs. *IACR Cryptology ePrint Archive*, 2000:63, 2000.

30 Shiva Prasad Kasiviswanathan, Homin K Lee, Kobbi Nissim, Sofya Raskhodnikova, and Adam Smith. What can we learn privately? *SIAM Journal on Computing*, 40(3):793–826, 2011.

31 Michael Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM (JACM)*, 45(6):983–1006, 1998.

32 Eyal Kushilevitz and Noam Nisan. *Communication complexity*. Cambridge University Press, 1997.

33 Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Multiple source adaptation and the rényi divergence. In *UAI*, pages 367–374, 2009. URL: `https://dslpitt.org/uai/displayArticleDetails.jsp?mmnu=1&smnu=2&article_id=1600&proceeding_id=25`.

34 Aaron Roth and Tim Roughgarden. Interactive privacy via the median mechanism. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 765–774. ACM, 2010.

35 Indrajit Roy, Srinath TV Setty, Ann Kilzer, Vitaly Shmatikov, and Emmett Witchel. Airavat: Security and privacy for mapreduce. In *NSDI*, volume 10, pages 297–312, 2010.

36 J. Steinhardt, G. Valiant, and S. Wager. Memory, communication, and statistical queries. In *COLT*, pages 1490–1516, 2016.

37 Jacob Steinhardt and John C. Duchi. Minimax rates for memory-bounded sparse linear regression. In *COLT*, pages 1564–1587, 2015. URL: `http://jmlr.org/proceedings/papers/v40/Steinhardt15.html`.

38 Arvind Sujeeth, HyoukJoong Lee, Kevin Brown, Tiark Rompf, Hassan Chafi, Michael Wu, Anand Atreya, Martin Odersky, and Kunle Olukotun. Optiml: an implicitly parallel domain-specific language for machine learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 609–616, 2011.

39 Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

40 Leslie G Valiant. Evolvability. *Journal of the ACM (JACM)*, 56(1):3, 2009.

41 Larry Wasserman. *All of statistics: a concise course in statistical inference*. Springer Science & Business Media, 2013.

42 Andrew Yao. Probabilistic computations: Toward a unified measure of complexity. In *FOCS*, pages 222–227, 1977.

43 Yuchen Zhang, John C. Duchi, Michael I. Jordan, and Martin J. Wainwright. Information-theoretic lower bounds for distributed statistical estimation with communication constraints. In *Proceedings of NIPS*, pages 2328–2336, 2013.

A Omitted proofs

A.1 Proof of Lemma 3.5

In the following, we denote by $o_c(\cdot)$ and $\omega_c(\cdot)$ asymptotic functions obtained by taking the limit as the parameter c goes to infinity. In particular, $o_c(1)$ can be made arbitrarily close to 0 by letting c be large enough.

Let W be as in the statement of Lemma 3.5. To prove the lemma, it suffices to show that each bit j in the binary representation of the subspace $\widehat{W}$ constructed by Algorithm 2 is equal to the corresponding bit of W. Henceforth, we fix j. We consider the two cases where bit j of W is equal to 1, and where it is equal to 0.

First, we assume that bit j of W is equal to 1, and prove that in the execution of Algorithm 2, it will be the case that $u_{i,j}/v_i \geq 1 - o_c(1)$. We can then set c to be sufficiently large to ensure that $u_{i,j}/v_i \geq (9/10)$. Note that for any positive real numbers N, D and τ such that $\tau = o(N)$ and $\tau = o(D)$, we have that

$$\frac{N-\tau}{D+\tau} \geq \frac{N}{D} \cdot (1 - o(1)).$$

Thus, it is enough to show that the next three statements hold:

(i) $\tau = o_c(\overline{v}_i)$,
(ii) if bit j of W is 1, then $(\overline{u}_{i,j}/\overline{v}_i) \geq 1 - o_c(1)$,
(iii) if bit j of W is 1, then $\tau = o_c(\overline{u}_{i,j})$,

where $\overline{u}_{i,j} \triangleq \mathbb{E}[\phi_{i,j}]$ and $\overline{v}_i \triangleq \mathbb{E}[\phi_i]$.

To show (i) above, note that

$$\begin{aligned}
\overline{v}_i &= \Pr\left[(b_1, \ldots, b_{k+1}) = 1^{k+1} \text{ and } \mathsf{rk}(Z) = i\right] \\
&\geq v_i - \tau \\
&\geq v \cdot \tau_i - \tau \\
&\geq \omega_c(\tau),
\end{aligned}$$

where the first inequality follows from the definition of v_i and the SQ guarantee, the second inequality follows from the given assumption (in the statement of Lemma 3.5) that $(v_i/v) \geq \tau_i$, and the last inequality follows from the fact that since $v > \epsilon^{k+1}/2$, for every $i \in [k+1]$, we have that

$$\tau = o_c\Big((v \cdot \tau_i - \tau) \cdot (1 - \tau_i/4)\Big).$$

.

Recall the definition of the event $E_j(Z)$ from the description of Algorithm 2. To show (ii) above, note that

$$\begin{aligned}
\frac{\overline{u}_{i,j}}{\overline{v}_i} &= \Pr\left[E_j(Z) \mid (b_1,\ldots,b_{k+1}) = 1^{k+1} \text{ and } \mathsf{rk}(Z) = i\right] \\
&\geq \Pr\left[\text{all rows of } Z \text{ belong to } W \mid (b_1,\ldots,b_{k+1}) = 1^{k+1} \text{ and } \mathsf{rk}(Z) = i\right] \\
&= 1 - \Pr\left[\exists \text{ a row of } Z \text{ that } \notin W \mid (b_1,\ldots,b_{k+1}) = 1^{k+1} \text{ and } \mathsf{rk}(Z) = i\right] \\
&\geq 1 - (k+1) \cdot \Pr_{z\sim Q}[z \notin W] \\
&\geq 1 - \frac{\tau_i}{4} \\
&\geq 1 - o_c(1),
\end{aligned}$$

where the first inequality uses the assumption that bit j in the binary representation of W is 1 and the facts that the dimension of W is equal to i and that we are conditioning on $\mathsf{rk}[Z] = i$. The second inequality follows from the union bound, the third inequality follows from the assumption given in Lemma 2, and the last inequality follows from the fact that for every $i \in [k+1]$, we have that $\tau_i = o_c(1)$.

To show (iii) above, note that

$$\begin{aligned}
\overline{u}_{i,j} &= \overline{v}_i \cdot \frac{\overline{u}_{i,j}}{\overline{v}_i} \\
&\geq \omega_c(\tau) \cdot (1 - o_c(1)) \\
&\geq \omega_c(\tau),
\end{aligned}$$

where the first inequality follows from (i) and (ii) above.

We now turn to the (slightly different) case where bit j of W is equal to 0, and prove that in the execution of Algorithm 2, we will have that $u_{i,j}/v_i = o_c(1)$. Note that for any positive real numbers N, D and τ such that $\tau = o(D)$, we have that

$$\frac{N+\tau}{D-\tau} \leq \frac{N}{D} \cdot (1 + o(1)) + o(1).$$

Thus, it is enough to use the fact that $\tau = o_c(\overline{v}_i)$ (proven in (i) above) and to show the next statement:

(iv) if bit j of W is 0, then $(\overline{u}_{i,j}/\overline{v}_i) = o_c(1)$.

To prove (iv), note that since bit j of W is 0, we have that

$$\begin{aligned}
\frac{\overline{u}_{i,j}}{\overline{v}_i} &\leq \Pr\left[\exists \text{ a row of } Z \text{ that } \notin W \mid (b_1,\ldots,b_{k+1}) = 1^{k+1} \text{ and } \mathsf{rk}(Z) = i\right] \\
&\leq \frac{\tau_i}{4} \\
&\leq o_c(1),
\end{aligned}$$

where the first inequality above follows from the assumption that bit j in the binary representation of W is 0 and the facts that the dimension of W is equal to i and that we are conditioning on $\mathsf{rk}[Z] = i$. The second inequality above follows from the union bound and the assumption given in Lemma 2, and the last inequality follows from the fact that for every $i \in [k+1]$, we have that $\tau_i = o_c(1)$. As before, we choose c to be sufficiently large to ensure that this last probability is smaller than $(1/10)$.

A.2 Proof of Proposition 3.12

Let $a \in \mathbb{F}_p^\ell$. We have that:

$$\begin{aligned}
\mathbb{E}_{(z,b)\sim D_0}[D_a(z,b)] &= \mathbb{E}_{(z,b)\sim D_0}\left[\prod_{i=1}^{k}\mathbb{E}_{(z_i,b_i)\sim D_0}[D_a(z_i,b_i)\right] \\
&= \prod_{i=1}^{k}\mathbb{E}_{(z_i,b_i)\sim D_0}\left[D_a(z_i,b_i)\right] \\
&= \prod_{i=1}^{k}\mathbb{E}_{(z_i,b_i)\sim D_0}\left[D_a(z_i)\cdot\mathbb{1}(b_i = f_a(z_i))\right] \\
&= \prod_{i=1}^{k}\mathbb{E}_{z_i\sim D_0}\left[D_a(z_i)\cdot\mathbb{E}_{b_i\in_R\{\pm 1\}}[\mathbb{1}(b_i = f_a(z_i))]\right] \\
&= \frac{1}{2^k}\cdot\prod_{i=1}^{k}\mathbb{E}_{z_i\sim D_0}\left[D_a(z_i)\right] \\
&= \frac{1}{2^k}\cdot\left(\frac{1}{p}\cdot\beta + \left(1-\frac{1}{p}\right)\cdot\alpha\right)^k.
\end{aligned}$$

A.3 Proof of Proposition 3.13

Let $a, a' \in \mathbb{F}_p^\ell$. First, assume that $\mathsf{Hyp}_a = \mathsf{Hyp}_{a'}$, i.e., that $a = a'$. Then,

$$\begin{aligned}
\mathbb{E}_{(z,b)\sim D_0}[D_a(z,b)\cdot D_{a'}(z,b)] &= \mathbb{E}_{(z,b)\sim D_0}[D_a(z,b)^2] \\
&= \mathbb{E}_{(z,b)\sim D_0}\left[\prod_{i=1}^{k}D_a(z_i,b_i)^2\right] \\
&= \prod_{i=1}^{k}\mathbb{E}_{(z_i,b_i)\sim D_0}[D_a(z_i,b_i)^2] \\
&= \prod_{i=1}^{k}\mathbb{E}_{(z_i,b_i)\sim D_0}[D_a(z_i)^2\cdot\mathbb{1}(b_i = f_a(z_i))] \\
&= \prod_{i=1}^{k}\mathbb{E}_{z_i}\left[D_a(z_i)^2\cdot\mathbb{E}_{b_i}[\mathbb{1}(b_i = f_a(z_i))]\right]
\end{aligned}$$

Thus,

$$\begin{aligned}
\mathbb{E}_{(z,b)\sim D_0}[D_a(z,b)\cdot D_{a'}(z,b)] &= \frac{1}{2^k}\cdot\prod_{i=1}^{k}\mathbb{E}_{z_i}[D_a(z_i)^2] \\
&= \frac{1}{2^k}\cdot\prod_{i=1}^{k}\left(\frac{1}{p}\cdot\beta^2 + \left(1-\frac{1}{p}\right)\cdot\alpha^2\right) \\
&= \frac{1}{2^k}\cdot\left(\frac{1}{p}\cdot\beta^2 + \left(1-\frac{1}{p}\right)\cdot\alpha^2\right)^k.
\end{aligned}$$

Now we assume that $\mathsf{Hyp}_a \cap \mathsf{Hyp}_{a'} = \emptyset$. Then,

$$\begin{aligned}
\mathbb{E}_{(z,b)\sim D_0}[D_a(z,b)\cdot D_{a'}(z,b)] &= \mathbb{E}_{(z,b)\sim D_0}\left[\prod_{i=1}^{k} D_a(z_i,b_i)\cdot D_{a'}(z_i,b_i)\right] \\
&= \prod_{i=1}^{k} \mathbb{E}_{(z_i,b_i)\sim D_0}[D_a(z_i,b_i)\cdot D_{a'}(z_i,b_i)] \\
&= \prod_{i=1}^{k} \mathbb{E}_{(z_i,b_i)\sim D_0}[D_a(z_i)\cdot \mathbb{1}(b_i = f_a(z_i)) \\
&\quad \cdot D_{a'}(z_i)\cdot \mathbb{1}(b_i = f_{a'}(z_i))] \\
&= \prod_{i=1}^{k} \mathbb{E}_{z_i}\Big[D_a(z_i)\cdot D_{a'}(z_i)\cdot \mathbb{1}(f_a(z_i) = f_{a'}(z_i)) \\
&\quad \cdot \mathbb{E}_{b_i}[\mathbb{1}(b_i = f_a(z_i))]\Big] \\
&= \frac{1}{2^k}\cdot \prod_{i=1}^{k} \mathbb{E}_{z_i}\Big[D_a(z_i)\cdot D_{a'}(z_i)\cdot \mathbb{1}(f_a(z_i) = f_{a'}(z_i))\Big] \\
&= \frac{1}{2^k}\cdot \prod_{i=1}^{k} \left(\alpha^2\cdot\left(1-\frac{2}{p}\right)\right) \\
&= \frac{1}{2^k}\cdot \left(\alpha^2\cdot\left(1-\frac{2}{p}\right)\right)^k.
\end{aligned}$$

Finally, we assume that $\mathsf{Hyp}_a \neq \mathsf{Hyp}_{a'}$ and $\mathsf{Hyp}_a \cap \mathsf{Hyp}_{a'} \neq \emptyset$. Then,

$$\begin{aligned}
\mathbb{E}_{(z,b)\sim D_0}[D_a(z,b)\cdot D_{a'}(z,b)] &= \frac{1}{2^k}\cdot \prod_{i=1}^{k} \mathbb{E}_{z_i}\Big[D_a(z_i)\cdot D_{a'}(z_i)\cdot \mathbb{1}(f_a(z_i) = f_{a'}(z_i))\Big] \\
&= \frac{1}{2^k}\cdot \prod_{i=1}^{k} \left(\frac{\beta^2}{p^2} + \alpha^2\cdot\left(1-\frac{2}{p}+\frac{1}{p^2}\right)\right) \\
&= \frac{1}{2^k}\cdot \left(\frac{\beta^2}{p^2} + \alpha^2\cdot\left(1-\frac{2}{p}+\frac{1}{p^2}\right)\right)^k.
\end{aligned}$$

A.4 Proof of Proposition 3.14

First, we assume that $a, a' \in \mathbb{F}_p^\ell$ are such that $\mathsf{Hyp}_a = \mathsf{Hyp}_{a'}$, i.e., $a = a'$. Then, by Proposition 3.13 and by our settings of α and β, we have that

$$\begin{aligned}
\mathbb{E}_{(z,b)\sim D_0}[D_a(z,b)\cdot D_{a'}(z,b)] &= \frac{1}{2^k}\cdot \left(\frac{1}{p}\cdot\beta^2 + \left(1-\frac{1}{p}\right)\cdot\alpha^2\right)^k \\
&= \frac{1}{2^{2k}\cdot p^{(2\ell-1)\cdot k}}\cdot\left(1+\frac{1}{p-1}\right)^k.
\end{aligned}$$

Hence, $D_0[\hat{D}_a\cdot\hat{D}_{a'}] = \left(p+1-\frac{1}{p-1}\right)^k - 1$, as desired.

Next, we assume that $a, a' \in \mathbb{F}_p^\ell$ are such that $\mathsf{Hyp}_a \cap \mathsf{Hyp}_{a'} = \emptyset$. Then, by Proposition 3.13

and by our setting of α, we have that

$$\begin{aligned}\mathbb{E}_{(z,b)\sim D_0}[D_a(z,b)\cdot D_{a'}(z,b)] &= \frac{1}{2^k}\cdot(\alpha^2\cdot(1-\frac{2}{p}))^k \\ &= \frac{1}{2^{3k}\cdot p^{2k\ell}}\cdot\frac{(1-\frac{2}{p})^k}{(1-\frac{1}{p})^{2k}}.\end{aligned}$$

Hence, $D_0[\hat{D}_a\cdot\hat{D}_{a'}] = \frac{1}{2^k}\cdot\frac{(1-\frac{2}{p})^k}{(1-\frac{1}{p})^{2k}} - 1$, as desired.

Finally, we assume that $a, a' \in \mathbb{F}_p^\ell$ are such that $\mathsf{Hyp}_a \neq \mathsf{Hyp}_{a'}$ and $\mathsf{Hyp}_a \cap \mathsf{Hyp}_{a'} \neq \emptyset$. Then, by Proposition 3.13 and by our settings of α and β, we have that

$$\begin{aligned}\mathbb{E}_{(z,b)\sim D_0}[D_a(z,b)\cdot D_{a'}(z,b)] &= \frac{1}{2^k}\cdot(\frac{\beta^2}{p^2}+\alpha^2\cdot(1-\frac{2}{p}+\frac{1}{p^2}))^k \\ &= \frac{1}{2^{2k}\cdot p^{2k\ell}}.\end{aligned}$$

Hence, $D_0[\hat{D}_a\cdot\hat{D}_{a'}] = 0$, as desired.