

# Memory-Sample Tradeoffs for Linear Regression with Small Error\*

**Vatsal Sharan**

Stanford University

vsharan@stanford.edu

**Aaron Sidford**

Stanford University

sidford@stanford.edu

**Gregory Valiant**

Stanford University

valiant@stanford.edu

## Abstract

We consider the problem of performing linear regression over a stream of  $d$ -dimensional examples, and show that any algorithm that uses a subquadratic amount of memory exhibits a slower rate of convergence than can be achieved without memory constraints. Specifically, consider a sequence of labeled examples  $(a_1, b_1), (a_2, b_2) \dots$ , with  $a_i$  drawn independently from a  $d$ -dimensional isotropic Gaussian, and where  $b_i = \langle a_i, x \rangle + \eta_i$ , for a fixed  $x \in \mathbb{R}^d$  with  $\|x\|_2 = 1$  and with independent noise  $\eta_i$  drawn uniformly from the interval  $[-2^{-d/5}, 2^{-d/5}]$ . We show that any algorithm with at most  $d^2/4$  bits of memory requires at least  $\Omega(d \log \log \frac{1}{\epsilon})$  samples to approximate  $x$  to  $\ell_2$  error  $\epsilon$  with probability of success at least  $2/3$ , for  $\epsilon$  sufficiently small as a function of  $d$ . In contrast, for such  $\epsilon$ ,  $x$  can be recovered to error  $\epsilon$  with probability  $1 - o(1)$  with memory  $O(d^2 \log(1/\epsilon))$  using  $d$  examples. This represents the first nontrivial lower bounds for regression with super-linear memory, and may open the door for strong memory/sample tradeoffs for continuous optimization.

---

\*The contributions of Vatsal and Gregory were supported by NSF awards AF-1813049 and CCF-1704417, and ONR Young Investigator Award N00014-18-1-2295. Aaron's contributions were supported by NSF CAREER Award CCF-1844855.

# 1 Introduction

What are the implications of memory constraints on the ability to efficiently learn or optimize? As has been revealed in a recent series of striking results [Raz, 2016, 2017, Beame et al., 2018, Kol et al., 2017, Moshkovitz and Moshkovitz, 2017, 2018, Garg et al., 2018], for a broad class of natural learning problems over the Boolean hypercube and other finite fields, there is a sharp threshold for the amount of memory required to learn with a polynomial amount of data.

This line of work was sparked by Raz’ breakthrough result [Raz, 2016], which considered the problem of learning a parity: given access to a stream of labeled examples,  $(a_1, b_1), \dots$ , where each  $a_i \in \{0, 1\}^d$  is drawn uniformly from the  $d$ -dimensional hypercube and  $b_i = \langle a_i, x \rangle \bmod 2$  for some fixed vector  $x$ , Raz showed that any algorithm with  $o(n^2)$  memory would require an exponential number of examples to learn  $x$  (with any significant probability of success). Of course, given a quadratic amount of memory,  $x$  can be efficiently computed by taking the first  $O(n)$  examples and solving the corresponding linear system over  $\mathbb{F}_2$ . Subsequent work extended this result to a broad class of discrete learning problems, including Raz [2017], Moshkovitz and Moshkovitz [2017, 2018] which generalized the results to a class of Boolean learning problems that satisfy a certain combinatorial condition, Kol et al. [2017] which extended the techniques to the problems of learning sparse parities (parities involving  $o(d)$  coordinates) which implies hardness of several other natural Boolean learning problems including learning small juntas, small decision trees, and small DNF formulae, and the works Beame et al. [2018] and Garg et al. [2018] which strengthened the approach of Raz [2017] to yield tight tradeoffs for a larger class of learning problems over finite fields, including homogeneous  $m$ -variate polynomials over  $\mathbb{F}_2$ .

For continuous, real-valued optimization and learning problems, much less is known about memory/sample tradeoffs. This is in spite of fact that the problem of learning a linear regression—the real-valued analog of learning parities—lies at the core of machine learning and is a prototypical convex optimization problem. Indeed, one of the original motivations for the conjecture that learning a parity required either a quadratic memory or exponential time, originally stated in Steinhardt et al. [2016], was the question of the memory/sample tradeoffs for linear regression.

This question of the memory/sample tradeoffs for linear regression is also extremely important from a practical perspective. Gradient-based ‘first-order’ methods are *the* workhorse of modern machine learning, in contrast to ‘second-order’ methods. This is explained by the efficiency benefit conferred by the linear memory footprint of first-order methods as opposed to the quadratic memory requirements of second-order methods. For large-scale learning problems, this reduction in memory usage of first-order methods more than compensates for the increase in the number of iterations or datapoints processed. If methods with comparable memory usage to first-order methods (or at least significantly subquadratic) were capable of achieving similar convergence rates to second-order methods, that could have far-reaching practical implications.

The question of the memory/sample tradeoffs for linear regression is also a natural and largely unexplored frontier of continuous optimization research. There is classic line of research which has proven information theoretic lower bounds on continuous optimization [Nemirovski and Yudin, 1983, Nesterov, 2014, Bubeck, 2015] in a restricted oracle model where the input real-valued function can only be queried via black box queries to an oracle that returns local information about the function, e.g. function values, gradients, etc. Given only mild regularity assumptions on a function, e.g. Lipschitz continuity, smoothness, convexity, etc., proving tight bounds on the number of queries to an oracle needed to approximately minimize a function is well-studied and has been a driving force behind the development of modern optimization theory. While there is some work studying the effect of parallelism on these lower bounds [Nemirovski, 1994, Balkanski and Singer, 2018, Diakonikolas and Guzmán, 2018], we are unaware of previous work proving gaps between the query complexity of optimization problems under differing memory constraints.

In this work, we provide the first nontrivial memory/sample tradeoffs for linear regression which apply in the regime where the available memory is significantly larger than what would be required to store each datapoint to high precision.

**Theorem 1.** *Consider a sequence of labeled examples  $(a_1, b_1), (a_2, b_2), \dots$ , with  $a_i$  drawn independently from a  $d$ -dimensional isotropic Gaussian with an identity covariance matrix, and where  $b_i = \langle a_i, x \rangle + \eta_i$ , for a fixed  $x \in \mathbb{R}^d$  with  $\|x\|_2 = 1$  and with independent noise  $\eta_i$  drawn uniformly from the interval  $[-2^{-d/5}, 2^{-d/5}]$ . Let  $\epsilon = 1/d^r$  for any  $r \leq O(d/\log d)$ . Then any algorithm with at most  $d^2/4$  bits of memory requires at least  $\Omega(d \log r)$  samples to approximate  $x$  to  $\ell_2$  error  $\epsilon$  with probability of success at least  $2/3$ . In particular, for  $\epsilon \leq 1/d^{\Omega(\log d)}$ , this implies*

that any algorithm with  $d^2/4$  bits of memory requires at least  $\Omega(d \log \log \frac{1}{\epsilon})$  samples to approximate  $x$  to error  $\epsilon$  with probability of success at least  $2/3$ .

For comparison, note that for  $\epsilon \in [2^{-\Theta(d)}, 1/d]$ , the trivial algorithm recovers  $x$  to error  $\epsilon$  with probability  $1 - o(1)$  using  $O(d^2 \log(1/\epsilon))$  bits of memory and  $d$  examples.<sup>1</sup>

## 1.1 Further Directions

Our result establishes the existence of a sharp gap in sample complexities for regression with bounded memory. Nevertheless, this still leaves a significant margin between our lower bounds on the convergence rate of bounded memory algorithms, and those achieved by the best known first-order methods which use memory  $O(d \log(1/\epsilon))$ . For example, randomized Kaczmarz [Strohmer and Vershynin, 2006] can be easily shown to compute a point  $\tilde{x}$  such that  $\|x - \tilde{x}\|_2 \leq \epsilon$  with constant probability using  $O(d \log(1/\epsilon))$  samples and memory  $O(d \log(1/\epsilon))$ . This is the best known sample complexity for achieving error  $\epsilon$  given this amount of memory, though our lower bound leaves open the question of whether it is optimal.

Beyond tightening our result, it is also worth considering the analogous regression question in the setting where the datapoints are drawn from an ill-conditioned distribution. Both in practice, and theory, first-order methods suffer a convergence rate that degrades with the condition number. Over the past decade there has been extensive research on providing iterative methods which use sub-quadratic space and seek better dependencies on the eigenvalues of the covariance of the distribution of  $a$ . Though there have been several improvements to randomized Kaczmarz and variants of SGD in recent years [Hu et al., 2009, Lan, 2012, Ghadimi and Lan, 2013, Lee and Sidford, 2013, Frostig et al., 2015, Liu and Wright, 2016, Needell et al., 2016, Dieuleveut et al., 2017, Jain et al., 2018] the number of samples required by these methods all depend *polynomially* on some measure of eigenvalue range or conditioning of the underlying covariance matrix. This is in sharp contrast to second-order methods which can simply store  $\Theta(d)$  samples and invert an associated matrix to compute an  $\epsilon$  accurate solution with  $\Omega(d^2 \log(d\kappa/\epsilon))$  memory where  $\kappa$  is a condition number measure of the matrix.

**Conjecture.** *For any  $\kappa$  bounded by a polynomial of  $d$ , there exists a distribution  $\mathcal{D}_\kappa$  over  $d$ -dimensional Gaussian distributions whose covariance has condition number  $\kappa$ , such that for  $G \leftarrow \mathcal{D}_\kappa$ , given a sequence of examples  $(a_i, b_i)$  with  $a_i$  drawn from  $G$  and  $b_i = \langle a_i, x \rangle$ , any algorithm that recovers the unit vector  $x$  to small constant  $\ell_2$  error with constant probability either requires  $\Omega(d^2)$  bits of memory, or  $d \cdot \text{poly}(\kappa)$  examples.*

The work of this paper may be a first step towards proving the above conjecture. We hope this paper will inspire efforts to establish strong memory/query complexity trade-offs for continuous optimization more broadly.

In a different direction, it may be worth considering the extent to which results of the form of Theorem 1 apply beyond the stochastic streaming setting. Rather than considering a stream of independent examples, one could consider the analogous questions in a cell-probe setting: suppose there is a set of  $O(d)$  examples stored in read-only memory, and one is charged according to the number of times each example is ‘downloaded’ into working memory. What are the tradeoffs between the amount of working memory, error of the recovered linear regression, and number of ‘downloads’? This setting closely corresponds to the data pipeline employed in many large scale learning settings, and any strong results in this setting would be extremely interesting.

It is worth noting that, even in the setting of learning parities, the stochasticity of the examples is essential to the exponential sample complexity of memory-constrained algorithms. Analogous results are not true in the above cell-probe model. For example, given  $O(n)$  examples stored in read-only memory, there exists a successful learning algorithm for the parity problem with  $O(n)$  working memory that uses  $\text{poly}(n)$  runtime and cell-probes [Kong, 2018] (though, to the best of our knowledge, it is not known if there is a successful learning algorithm for the real-valued regression problem which uses  $O(n)$  working memory and  $\text{poly}(n)$  cell-probes). Still, establishing any nontrivial gap between memory-constrained and unconstrained learning (for either the real-valued regression or parity problems) in the cell-probe setting would be exciting, though may be quite difficult.

<sup>1</sup>This follows from the fact that the condition number of the system with  $d$  examples is at most  $1/\text{poly}(d)$  with high probability, and hence we can solve the system to accuracy  $\epsilon$  by doing all computations with  $O(\log(d/\epsilon)) = O(\log(1/\epsilon))$  bits of precision, for  $\epsilon \leq 1/d$ .

## 1.2 Related Work

A number of recent works have examined learning problems such as sparse linear regression [Steinhardt and Duchi, 2015] and detecting correlations [Shamir, 2014, Dagan and Shamir, 2018] under information constraints such as limited memory or communication constraints. These results usually develop information-theoretic inequalities [Braverman et al., 2016, Steinhardt et al., 2016, Dagan and Shamir, 2018] to show that unless a set of parties exchange a minimum amount of information, they cannot solve the learning problem—with the memory bound following as a consequence of the communication lower bound. At a high level, the idea is to show that if the learning problem requires distinguishing between a set of  $k$  distributions, and if the distributions are sufficiently uncorrelated, then at least  $\Omega(k)$  bits of communication are needed to solve the learning problem. While initial results only obtained lower bounds for settings where the memory budget is less than the size of each data point, the recent work Dagan and Shamir [2018] circumvented this barrier and showed strong lower bounds for detecting correlations for natural distributions under information constraints.

Many of these information theoretic tools seem to break down for learning problems such as parity learning where communication lower bounds do not directly give meaningful memory bounds. Hence these settings require explicitly taking into account the memory constraint of the algorithm; the recent line of work discussed in the introduction, starting with Raz [2016] achieves sharp lower bounds for memory-bounded learning by directly analyzing the structure of width-bounded branching programs for these problems [Raz, 2017, Kol et al., 2017, Moshkovitz and Moshkovitz, 2017, 2018, Beame et al., 2018, Garg et al., 2018]. Our work directly builds on the analysis framework developed in Raz [2017], and extended in Beame et al. [2018] and Garg et al. [2018], with the crucial difference that the geometry of the continuous space corresponding to linear regression lacks many of the combinatorial properties that are leveraged in the analysis of these prior works.

There is also a large literature on memory lower bounds for streaming algorithms (for e.g. Alon et al. [1999], Bar-Yossef et al. [2004]), although these are mostly for non-learning problems and assume that the input stream is constructed in an adversarial fashion.

On the optimization side, there is a long history of proving information theoretic lower bounds on optimization methods. These results typically show that given a type of restricted local oracle to access the input, i.e. an oracle which only returns information about values, gradients, higher derivatives, separation oracles, etc., lower bounds can be formally proven on the number of queries needed to approximately minimize the function. Such results date back early work of Nemirovski and Yudin [1983] on the oracle complexity of optimization and there are too many results to do a complete review (see Nesterov [2014], Bubeck [2015] for more recent surveys). Key results in this broad area of research include, tight oracle bounds known for computing approximate minimizer of smooth convex functions given a gradient oracle [Nemirovski and Yudin, 1983, Nesterov, 2014, Bubeck, 2015] even when randomization is allowed [Woodworth and Srebro, 2016], tight oracle bounds known for computing approximate minimizer of Lipschitz convex function given by a subgradient oracle [Braun et al., 2017, Nemirovski and Yudin, 1983, Nesterov, 2014], and even tight oracle bounds for computing critical points, that is points of small gradient, for smooth non-convex functions given by a gradient oracle [Nesterov, 2012, Cartis et al., 2012, 2017, Carmon et al., 2017a,b]. There has also been extensive research on the oracle complexity of stochastic optimization [Nemirovski and Yudin, 1983, Devolder et al., 2013, 2014, Shamir, 2013, Duchi et al., 2015] and work on the tradeoff between oracle complexity and parallelism for nonsmooth optimization with a subgradient oracle [Nemirovski, 1994, Balkanski and Singer, 2018, Diakonikolas and Guzmán, 2018]. However, to the best of the authors knowledge the problem of memory / query complexity tradeoffs for real-valued continuous optimization has been largely unexplored.

## 2 Setup and Proof Overview

In this section, we provide an overview of our proof approach. We begin by describing the notation and formalism we will use in analyzing the branching program representing a memory-bounded learning algorithm.

## Branching Programs for Learning

We model the learner by a branching program  $B$ . A branching program is a general non-uniform model for space bounded computation. The branching program has  $m$  layers, corresponding to  $m$  time steps, with each layer having at most  $w$  vertices, where  $w$  denotes the *width* of  $B$ . Each vertex of  $B$  corresponds to a memory state, and a branching program with width  $w$  corresponds to an algorithm with memory usage  $\log_2 w$ . A vertex with no outgoing edges is called a *leaf*, and all vertices in the last layer are leaves (though there may be additional leaves). Each non-leaf vertex  $v$  has an associated transition function  $f_v : \mathbb{R}^d \times \mathbb{R} \rightarrow [w]$ , representing the mapping from an example  $(a, b)$  to a vertex in the next layer. Without loss of generality, we may assume that these transition functions are deterministic, as randomization cannot improve the probability of success.<sup>2</sup> To be consistent with the literature on branching programs, we will refer to this transition function as a series of ‘edges’ indexed by the (infinite number of) potential examples  $(a, b)$ . Finally, each leaf vertex,  $v$ , of the branching program is labeled by a *label*,  $\tilde{x}(v)$ , representing the output value that the corresponding algorithm would produce on the sequence of examples that led to vertex  $v$ .

The success probability of the branching program  $B$  for a specified accuracy parameter  $\epsilon$  is the probability that  $\|\tilde{x} - x\|_2 \leq \epsilon$ , where  $\tilde{x}$  is the vector returned by  $B$ , and the probability is with respect to the randomness in the sequence of examples and choice of the true  $x$ .

## Setup

We consider branching programs whose goal is to learn some true  $x \in \mathbb{R}^d$  with  $\|x\|_2 = 1$  to  $\ell_2$  error  $\epsilon$ , in the setting where  $x$  is drawn uniformly at random from the  $d$  dimensional unit sphere. At every time step, a  $d$ -dimensional vector  $a$  is sampled from  $N(0, I_d)$ , and the branching program is given  $a$  and the (noisy) inner product  $b = a^T x + \eta$ , where the noise  $\eta$  is sampled from  $U[-\delta, \delta]$  for  $\delta = 2^{-d/5}$ . The addition of this noise facilitates the analysis, and we could have equivalently assumed that the true inner product  $a^T x$  is discretized according to some exponentially small discretization error  $\delta$ . Note that as long as the goal is to estimate  $x$  up to accuracy  $\epsilon \geq 2^{-\gamma d}$  for a small constant  $\gamma$ , the small uniform noise or discretization does not create any information theoretic obstacles.

## Proof Overview

Our proof follows and builds on the recent analytic framework for showing time-space lower bounds developed in Raz [2017], and further extended in Beame et al. [2018] and Garg et al. [2018]. The analysis in our case is complicated by the fact that the gap in the sample complexity of first order and second order methods for regression on well-conditioned matrices is not very large, and depends on the desired error  $\epsilon$ . To capture this dependence of the sample complexity on  $\epsilon$ , we divide the branching program into multiple *stages*, where a stage is a group of consecutive layers of the branching program. Each stage will intuitively correspond to the branching program reducing the  $\ell_2$  error of the estimate of  $x$  by a factor of two. We will argue that each of these stages cannot be too short if the algorithm has small memory. We now sketch the proof, describing the high-level framework of Raz [2017] and how we adapt it to our setup.

As in Raz [2017], we define a truncated computation path  $\mathcal{T}$  which follows the computation path of the branching program  $B$ , but may stop before reaching a leaf vertex. The conditions under which  $\mathcal{T}$  stops before reaching a leaf vertex will depend on the stage of the branching program which  $\mathcal{T}$  is in. For any vertex  $v$  in the branching program, let  $\mathbb{P}_{x|v}$  be the posterior distribution of  $x$  conditioned on being at  $v$ . We will quantify the progress made by a vertex  $v$  towards learning  $x$  by the  $\ell_2$  norm  $\|\mathbb{P}_{x|v}\|_2$  of the posterior distribution  $\mathbb{P}_{x|v}$  of  $v$ , note that a large norm indicates a concentrated posterior with an accurate estimate of  $x$ . The truncated path  $\mathcal{T}$  stops at any *significant* vertex  $s$  where  $\|\mathbb{P}_{x|s}\|_2$  is larger than some threshold, where the threshold is chosen as a function of the stage of the branching program being analyzed. Intuitively, if  $\|\mathbb{P}_{x|s}\|_2$  is larger than a given threshold, then  $\mathcal{T}$  has more information about  $x$  than we expect it to have at that stage. Most of the effort in Raz [2017] and in our work goes into ensuring that the probability of any significant vertex is small enough that the probability of  $\mathcal{T}$  stopping due to reaching a significant vertex is small.

<sup>2</sup>This can be easily seen by noting that any branching program with randomized transitions can be converted to a deterministic one by iteratively derandomizing each vertex by replacing its randomized transition function with the deterministic one that select the transition that maximizes the probability of success (breaking ties arbitrarily), where the probability is taken over the randomization in the subsequent examples and whatever randomization remains in the transition functions corresponding to other vertices.

We now sketch the argument for showing that the probability of reaching a significant vertex,  $s$ , is small, for any stage  $B_t$  of the branching program  $B$ . Let  $L_i$  denote the set of all vertices in the  $i$ -th layer of the  $t$ -th stage  $B_t$ . The following potential function tracks the progress which the  $i$ -th layer of  $B_t$  has made towards a fixed significant vertex  $s$  in  $B_t$ ,

$$\mathcal{Z}_i = \sum_{v \in L_i} \Pr(v) \cdot \langle \mathbb{P}_{x|v}, \mathbb{P}_{x|s} \rangle^{d/2}.$$

We claim that if  $\mathcal{Z}_i$  is small and the significant vertex  $s$  lies in the  $i$ -th layer of  $B_t$ , then the probability of  $s$  must also be small. This follows because we define significant vertices as those for which  $\|\mathbb{P}_{x|s}\|_2$  is large, and if  $s$  is in the  $i$ -th layer then  $\mathcal{Z}_i \geq \Pr(s) \cdot \|\mathbb{P}_{x|s}\|_2^d$ . Hence our goal will be to show that  $\mathcal{Z}_i$  is small. Note that raising to the power of  $d/2$  in our expression for  $\mathcal{Z}_i$  allows us to show that the probability of significant vertices is small enough that we can do a union bound over all vertices in the branching program, and  $d/2$  is the largest power to which we can raise while keeping the contribution of the low probability events small.

We prove that  $\mathcal{Z}_i$  is small via an induction argument. We first show that  $\mathcal{Z}_0$  must be small, as the previous stage  $B_{t-1}$  of the branching program could not have made too much progress. We next show that  $\mathcal{Z}_{i+1}$  cannot be much larger than  $\mathcal{Z}_i$ . To show this, we introduce another potential which tracks how much progress any edge  $e$  of the branching program has made towards  $s$ . Let  $\mathbb{P}_{x|e}$  be the posterior distribution of  $x$  conditioned on the event of traversing edge  $e$  in the branching program. Let  $\Gamma_i$  denote the set of all edges from the  $(i-1)$ -th layer to the  $i$ -th layer of the  $t$ -th stage  $B_t$  of the branching program, and let  $p(e)$  be the p.d.f. of the distribution over edges evaluated at edge  $e$ . For any  $i$ , we define the potential,

$$\mathcal{Z}'_i = \int_{e \in \Gamma_i} p(e) \cdot \langle \mathbb{P}_{x|e}, \mathbb{P}_{x|s} \rangle^{d/2} de.$$

A straightforward convexity argument shows that  $\mathcal{Z}_{i+1} \leq \mathcal{Z}'_i$ . Hence the main challenge is showing that  $\mathcal{Z}'_i$  cannot be much larger than  $\mathcal{Z}_i$ . This is where our analysis differs significantly from Raz [2017] (this is also where Beame et al. [2018] and Garg et al. [2018] differ the most from Raz [2017]). In these previous works which concern learning over finite fields, the learning problem is viewed as a certain matrix, and properties of this matrix are used to show that  $\mathcal{Z}'_i$  cannot be much larger than  $\mathcal{Z}_i$ . It is worth noting that in these settings, it is possible to argue that the example in the next time step looks almost random to the branching program if it does not have significant knowledge about the answer, and then use this to show that the branching program cannot make too much progress when it gets an example. In our case though, first-order methods which require only linear memory *can* learn  $x$  up to non-negligible error with only linear sample complexity, hence the examples do not have as much randomness. Also, as we work over continuous spaces we lack the combinatorial properties that enables the analysis in the previous works to go through, and need to develop different tools.

We now sketch our argument for showing that  $\mathcal{Z}'_i$  cannot be much larger than  $\mathcal{Z}_i$ . For intuition, we first describe the argument as it would pertain to the branching program corresponding to the linear memory, first-order method for regression. At a high level, by the end of the  $t$ -th stage of this branching program, the algorithm has learned  $x$  up to error roughly  $\epsilon_t = 1/2^t$ , and the posterior  $\mathbb{P}_{x|v}$  of a vertex  $v$  in this stage roughly corresponds to a spherical Gaussian with standard deviation  $\epsilon_t$  in every direction. A target significant vertex,  $s$ , in the  $t$ -th stage will have posterior  $\mathbb{P}_{x|s}$  roughly corresponding to another spherical Gaussian, but with standard deviation  $\epsilon_t/2$  in every direction. This significant vertex represents a memory state that has learned significantly more than is expected of a vertex in this stage, and we will show that the probability of reaching such a vertex is small. As every example  $(a, b)$  has some small uniform noise  $\eta \sim U[-\delta, \delta]$  added to  $b$ , if the branching program is initially at vertex  $v$  and then gets the example  $(a, b)$ , the posterior  $\mathbb{P}_{x|v}$  is updated by restricting it to the thin slice of the spherical Gaussian where  $a^T x = [b - \delta, b + \delta]$ . We need to argue that this slicing does not significantly increase the inner product with the posterior  $\mathbb{P}_{x|s}$  corresponding to the smaller, target Gaussian. This holds, provided the target Gaussian does not have significantly higher probability mass in the slice to which we are restricting. This is easy to analyze in this special setting where the posteriors are spherical Gaussians, by simply analyzing the projections of the two Gaussians along a random direction  $a$ . In our actual proof, to bound the rate of progress via this argument, we cannot assume that the posteriors have such a nice form. Nevertheless, we show a concentration result that guarantees that, for any distribution with sufficiently small  $\ell_2$



norm, the projections can not behave too much worse than projections of spherical Gaussians with the corresponding  $\ell_2$  norms.

To sketch the argument more formally, we need to define some notation. We define  $\tilde{f}$  as the point-wise product of the distributions  $\mathbb{P}_{x|v}$  and  $\mathbb{P}_{x|s}$ , with suitable normalization. Hence for any  $x'$  on the  $d$  dimensional unit sphere,

$$\tilde{f}(x') = \frac{\mathbb{P}_{x|v}(x') \cdot \mathbb{P}_{x|s}(x')}{\int_z \mathbb{P}_{x|v}(z) \cdot \mathbb{P}_{x|s}(z) dz}.$$

Let  $I_\delta(b)$  be the interval  $[b-\delta, b+\delta]$ . For any distribution  $f$  and fixed  $a$ , define  $G_{f,a}(I_\delta(b)) = \mathbb{E}_{x' \sim f} [\mathbf{1}(a^T x' \in I_\delta(b))]$ . Note that for a vertex  $v$  with posterior distribution  $\mathbb{P}_{x|v}$ ,  $G_{\mathbb{P}_{x|v},a}(I_\delta(b))$  is the probability mass on vectors  $x'$  which are consistent with the example  $(a, b)$ , up to the noise level  $\delta$ . With some technical work, we can approximately relate  $\langle \mathbb{P}_{x|v}, \mathbb{P}_{x|s} \rangle$  and  $\langle \mathbb{P}_{x|e}, \mathbb{P}_{x|s} \rangle$  for an edge  $e$  labelled by  $(a, b)$  as follows,

$$\langle \mathbb{P}_{x|e}, \mathbb{P}_{x|s} \rangle \approx \langle \mathbb{P}_{x|v}, \mathbb{P}_{x|s} \rangle \cdot \frac{G_{\tilde{f},a}(I_\delta(b))}{G_{\mathbb{P}_{x|v},a}(I_\delta(b))}.$$

Intuitively, the above relation says that the progress that the truncated path  $\mathcal{T}$  makes towards some target distribution  $\tilde{f}$  after receiving example  $(a, b)$  depends on the ratio of the probability mass of  $\tilde{f}$  which is consistent with  $(a, b)$ , and that of  $\mathbb{P}_{x|v}$  which is consistent with  $(a, b)$ . Hence in order to bound  $\mathbb{E}_e[\langle \mathbb{P}_{x|e}, \mathbb{P}_{x|s} \rangle^{d/2}]$  in terms of  $\langle \mathbb{P}_{x|v}, \mathbb{P}_{x|s} \rangle^{d/2}$ , our goal will be to upper bound

$$\mathbb{E}_{a,b} \left[ \frac{G_{\tilde{f},a}(I_\delta(b))}{G_{\mathbb{P}_{x|v},a}(I_\delta(b))} \right]^{d/2}.$$

Note that as  $b = a^T x + \eta$  where  $x \sim \mathbb{P}_{x|v}$ , we can show that examples  $(a, b)$  where  $G_{\mathbb{P}_{x|v},a}(I_\delta(b))$  is too small have small probability. Hence we can lower bound the denominator by making the truncated path  $\mathcal{T}$  stop if  $G_{\mathbb{P}_{x|v},a}(I_\delta(b))$  is too small, while still ensuring that the probability of  $\mathcal{T}$  stopping due to this reason is small.

It is more complicated to upper bound  $\mathbb{E}_{a,b}[G_{\tilde{f},a}(I_\delta(b))^{d/2}]$ . Note that  $G_{\tilde{f},a}(I_\delta(b))$  is the probability mass of the distribution  $\tilde{f}$  which lies in the interval  $I_\delta(b)$  when  $\tilde{f}$  is projected onto a random direction  $a$ . The linear projection of a high-dimensional distribution onto a random direction is a well-studied topic, and it is known that under mild conditions on  $\tilde{f}$  such as bounded second moments, its projection onto a random direction is approximately Gaussian [Bobkov, 2003, Anttila et al., 2003, von Weizsäcker, 1997] or a mixture of Gaussians [Dasgupta et al., 2006] with high probability. However, these results typically only give an additive  $O(1/\sqrt{d})$  error guarantee for the difference between the probability mass of  $\tilde{f}$  on any interval  $I$  and that of an appropriate Gaussian on that interval (and this is tight given only second moment constraints). Note that in our case the intervals  $I_\delta(b)$  have exponentially small width  $\delta$ , and we care about the multiplicative approximation error, hence these  $O(1/\sqrt{d})$  additive error guarantees are not strong enough. We show that we can obtain stronger guarantees in our case by ensuring that  $\|\tilde{f}\|_2$  is small, which we guarantee by appropriate conditions on the truncated path  $\mathcal{T}$ . With a bound on  $\|\tilde{f}\|_2$ , we prove the following concentration result for projections of high-dimensional distributions onto a random direction—

**Lemma 1.** *Let  $\tilde{f}$  be a distribution over the  $d$  dimensional sphere, with  $\|\tilde{f}\|_2 \leq \frac{(100/\epsilon)^d}{C_d}$  for some  $\epsilon \leq 1$ . For an absolute constant  $C$  and fixed  $b$ ,*

$$\mathbb{E}_a \left[ G_{\tilde{f},a}(I_\delta(b))^{d/2} \right] \leq (C\epsilon^{-20}\delta)^{d/2}.$$

Finally, note that the above bound is for a fixed  $b$ , but if the branching program knows  $x$  to a small error then it also knows the inner product  $b$  for any  $a$  to a small error, hence the distribution of  $b$  is itself highly dependent on  $a$ . To get around this, we prove a version of the above lemma where  $b$  is obtained by first sampling  $x$  from  $\mathbb{P}_{x|v}$  and then adding noise  $\eta$  to  $a^T x$ . These concentration bounds may be useful beyond this work, and it may be interesting to further develop our understanding of properties of the projection of high-dimensional distributions with small  $\ell_2$  norm onto a random direction.

### 3 Notation

Let  $\mathcal{S}_d$  be the set of all vectors on the  $d$ -dimensional unit sphere, and  $\mathcal{U}_d$  be the uniform distribution over  $\mathcal{S}_d$ . Hence  $\mathcal{U}_d(x) = 1/C_d$  for all  $x \in \mathcal{S}_d$ , for some  $C_d$  which depends on  $d$ .

Let  $E_v$  denote the event that the truncated path reaches a vertex  $v$ . For any random variable  $Z$  we denote the distribution of  $Z$  by  $\mathbb{P}_Z$ . We denote the probability of any vertex  $v$  in the branching program by  $\Pr(v)$ . As the edges  $e$  of the branching program are indexed by real valued  $(a, b)$ , for any edge  $e$  we denote the p.d.f. of the distribution over all edges of the branching program evaluated at the edge  $e$  by  $p(e)$ . Let the sample at the  $i$ th time step be  $(a_i, b_i)$ . Recall that the distribution of  $a_i$  is  $N(0, I_d)$ , and we will denote its p.d.f. at a vector  $a$  by  $p(a)$ . Similarly, we denote the p.d.f. of  $b$  conditioned on being at vertex  $v$  and seeing example  $a$  as  $p(b|a, E_v)$ . For any function  $f$  from  $\mathcal{S}_d \rightarrow \mathbb{R}$ , we denote by  $\|f\|_2$  the  $\ell_2$  norm of  $f$  with respect to the uniform distribution  $\mathcal{U}_d$  over  $\mathcal{S}_d$ ,

$$\|f\|_2 = \left( \mathbb{E}_{x \sim \mathcal{U}_d} [f(x)^2] \right)^{1/2}.$$

Recall from the previous section that  $G_{f,a}(I_\delta(b)) = \mathbb{E}_{x' \sim f} \left[ \mathbf{1}(a^T x' \in I_\delta(b)) \right]$  for any distribution  $f$ , where  $I_\delta(b)$  is the interval  $[b - \delta, b + \delta]$ . For notational convenience, we will subsequently denote  $G_{\mathbb{P}_{x|v},a}(I_\delta(b))$  for a vertex  $v$  by  $G_{v,a}(I_\delta(b))$ .

### 4 Proof of Theorem 1

In this section, we formally define the stages of the branching program and the truncated computation path  $\mathcal{T}$ , and then provide a proof for Theorem 1.

#### Stages of the Branching Program

We partition the branching program  $B$  into  $T$  stages  $\{B_t : 0 \leq t \leq T\}$ , for some  $T$  which depends on the desired accuracy  $\epsilon$ . The  $t$ th stage  $B_t$  continues for  $m_t$  time steps, where  $m_t = \lceil \frac{c_0 d}{\log d + t} \rceil$  and  $c_0$  is an absolute constant to be determined later. We define the stages inductively. The first stage  $B_0$  consists of all vertices up until and including the  $m_0$ -th layer of the branching program  $B$ . The  $t$ -th stage  $B_t$  consists of  $(m_t + 1)$  layers beginning with and including the last layer of the previous stage  $B_{t-1}$ .

#### Truncated Path

We define the truncated path  $\mathcal{T}$  corresponding to the branching program  $B$ . The truncated path  $\mathcal{T}$  follows the same path as  $B$ , except that it sometimes stops before reaching a leaf vertex. The conditions under which the truncated path stops before reaching a leaf vertex will be different depending on the stage  $t$ . Define  $\epsilon_t = 2^{-t}$ . Intuitively,  $\epsilon_t$  determines the accuracy to which  $B$  could know  $x$  in the  $t$ -th stage. In the  $t$ -th stage  $B_t$  of  $B$ , the truncated path stops at a non-leaf vertex  $v$  for any of the following three reasons—

1. If  $v$  is a *significant* vertex, where  $\|\mathbb{P}_{x|v}\|_2 > \frac{(2/\epsilon_t)^d}{C_d}$ .
2. If  $x$  belongs to the set of vectors  $\text{Sig}(v)$  which have non-trivial probability mass under  $\mathbb{P}_{x|v}(x)$ , defined as  $\text{Sig}(v) = \left\{ x' : \mathbb{P}_{x|v}(x') > \frac{(4/\epsilon_t)^{2d}}{C_d} \right\}$ .
3. If the branching program is about to traverse a *bad* edge. The set  $\text{Bad}(v)$  of bad edges for the vertex  $v$  is defined as the set of edges  $(a, b)$  for which either i)  $\|a\|_2 \geq 2\sqrt{d}$ , or ii)  $G_{v,a}(I_\delta(b)) \leq 2\delta/d^3$ .

If the truncated path  $\mathcal{T}$  does not stop at a non-leaf vertex, then it follows the same path as the computation path of the branching program  $B$ . Lemma 2 proved in Section 5 shows that the probability of the truncated path stopping at a non-leaf vertex is small.



**Lemma 2.** *If the number of samples  $m \leq d^{1.25}$  and the width of the branching program  $w \leq 2^{d^2/4}$ , then the probability of  $\mathcal{T}$  stopping at a non-leaf vertex is at most  $1/(2d)$ .*

To prove Lemma 2, we show that the probability of the truncated path stopping at a non-leaf vertex due to each of the three above reasons is small. Most of the effort goes into proving that the probability of stopping due to the first reason, reaching a significant vertex, is small. This is proved in Section 6. Using Lemma 2, we are now ready to prove our main theorem.

**Theorem 1.** *Let  $B$  be a branching program to find  $\tilde{x} : \|x - \tilde{x}\|_2 \leq \epsilon$ , where  $\epsilon = 1/d^r$  for some  $r \leq O(d/\log d)$ . For a small absolute constant  $c$ , if  $B$  has length at most  $cd \log r$  and width at most  $2^{d^2/4}$ , then the success probability of  $B$  is at most  $1/d$ .*

**Proof.** We partition the branching program into  $T$  stages and consider the truncated path  $\mathcal{T}$ . We first bound the number of stages  $T$  required to do the partition if  $m \leq cd \log r$ . We claim that  $T \leq (r/40) \log d$ . As the  $t$ -th stage consists of  $\lceil \frac{c_0 d}{\log d + t} \rceil$  steps, the number of steps in  $T = (r/40) \log d$  stages can be lower bounded as follows,

$$\begin{aligned} \sum_{t=1}^{(r/40) \log d} \left\lceil \frac{c_0 d}{\log d + t} \right\rceil &\geq c_0 d \sum_{t=\log d}^{(r/40) \log d} \frac{1}{\log d + t} \geq \frac{c_0 d}{2} \sum_{t=\log d}^{(r/40) \log d} \frac{1}{t} \\ &\geq \frac{c_0 d}{2} \log \left( \frac{r \log d}{40 \log d} \right) \geq \frac{c_0 d \log r}{100}. \end{aligned}$$

Hence taking  $c = c_0/100$ , the number of stages  $T$  in  $cd \log r$  steps is at most  $(r/40) \log d$ . Note that if  $\mathcal{T}$  does not stop before reaching a leaf, then it follows the same path as the branching program  $B$ . By Lemma 2, the probability that  $\mathcal{T}$  stops before reaching a leaf is at most  $1/(2d)$ . Hence we now only need to bound the probability that a non-significant leaf  $v$  of  $\mathcal{T}$  outputs  $\tilde{x}$  such that  $\|x - \tilde{x}\|_2 \leq \epsilon$ . However, for a non-significant leaf  $v$  we know that  $\|\mathbb{P}_{x|v}\|_2 \leq \frac{(2/\epsilon_T)^d}{C_d}$ . Further, the following lemma (proved in Section 7) shows that this condition implies that the probability of  $v$  outputting an  $\epsilon$  accurate answer is small.

**Lemma 3.** *Let  $f$  be a distribution over the  $d$  dimensional sphere  $\mathcal{S}_d$ , with  $\|f\|_2 \leq \frac{(2/\epsilon)^d}{C_d}$  for some  $\epsilon \leq 0.01$ . Then for any  $x \in \mathcal{S}_d$ ,*

$$\Pr_{x \sim f} \left[ x : \|x - \tilde{x}\|_2 \leq \epsilon^{40} \right] \leq 2^{-d/2}.$$

Now for  $T = (r/40) \log d$ ,  $\epsilon_T = 1/d^{r/40}$ . Hence by Lemma 3, the probability that a non-significant leaf  $v$  outputting an  $1/d^r$  accurate answer is at most  $2^{-d/2}$ . By a union bound over the probability of the truncated path stopping before a non-leaf vertex and the probability of a non-significant leaf outputting a valid answer, the probability of  $B$  outputting an  $1/d^r$  accurate answer is at most  $1/d$ . ■

## 5 Bounding the Probability of the Truncated Path Stopping Early

In this section, we show that the probability the truncated path  $\mathcal{T}$  stop at a non-leaf vertex is small. Lemma 4 shows that probability of  $\mathcal{T}$  stopping because of the first reason (reaching a significant vertex) is small. Most of the remainder of the paper will be devoted to proving Lemma 4.

**Lemma 4.** *If the total number of stages  $T \leq d^{1.25}$  and the width of the branching program  $w \leq 2^{d^2/4}$ , then the probability that  $\mathcal{T}$  reaches a significant vertex in any stage is at most  $2^{-d}$ .*

Lemma 5 and Lemma 6 show that the probability of the truncated path  $\mathcal{T}$  stopping due to reasons (2) and (3) respectively is small.

**Lemma 5.** *If  $v$  is not a significant vertex of  $B$ , then*

$$\Pr[x \in \text{Sig}(v) | E_v] \leq 2^{-2d}.$$

**Proof.** Assume  $v$  is in the  $t$ -th stage in the branching program. Since  $v$  is not a significant vertex,

$$\mathbb{E}_{x' \sim \mathbb{P}_{x|v}} [\mathbb{P}_{x|v}(x')] = \int_{x' \in \mathcal{S}_d} \mathbb{P}_{x|v}(x')^2 dx' = C_d \mathbb{E}_{x' \sim \mathcal{U}_d} [\mathbb{P}_{x|v}(x')^2] \leq \frac{(2/\epsilon_t)^{2d}}{C_d}.$$

Hence by Markov's inequality,

$$\Pr_{x' \sim \mathbb{P}_{x|v}} \left[ \mathbb{P}_{x|v}(x') > \frac{(4/\epsilon_t)^{2d}}{C_d} \right] \leq 2^{-2d}.$$

Since conditioned on  $E_v$ , the distribution of  $x$  is  $\mathbb{P}_{x|v}$ , we get,

$$\Pr[x \in \text{Sig}(v) | E_v] = \Pr_{x' \sim \mathbb{P}_{x|v}} \left[ \mathbb{P}_{x|v}(x') > \frac{(4/\epsilon_t)^{2d}}{C_d} \right] \leq 2^{-2d}.$$

■

**Lemma 6.**  $\Pr_{(a_{i+1}, b_{i+1})} [(a_{i+1}, b_{i+1}) \in \text{Bad}(v)] \leq 5/d^{2.5}.$

**Proof.** As  $a \sim N(0, I)$ , by standard concentration bounds for  $\chi^2$  random variables,  $\Pr[\|a\|_2 \geq 2\sqrt{d}] \leq e^{-d/10}$ . Conditioned on  $\|a\|_2 \leq 2\sqrt{d}$ ,  $|a^T x| \leq 2\sqrt{d}$ . As  $b$  is generated by adding noise drawn uniformly at random from  $[-\delta, \delta]$  to the true inner product  $a^T x$ ,  $p(b|a, E_v) = (2\delta)^{-1} G_{v,a}(I_\delta(b))$ , where we use our notation  $G_{v,a}(I_\delta(b)) = \mathbb{E}_{x' \sim \mathbb{P}_{x|v}} [\mathbf{1}(a^T x' \in I_\delta(b))]$ . Let  $u(b)$  be the p.d.f. of the uniform distribution on  $b$  with support  $[-2\sqrt{d}, 2\sqrt{d}]$ . Note that,

$$\int_{b \in \mathbb{R}} p(b|a, E_v) \mathbf{1}(p(b|a, E_v) \leq 4u(b)/d^{2.5}) db \leq 4 \int_{b \in \mathbb{R}} \frac{u(b)}{d^{2.5}} db \leq \frac{4}{d^{2.5}}.$$

Therefore as  $u(b) = 1/(4\sqrt{d})$  and  $G_{v,a}(I_\delta(b)) = 2\delta p(b|a, E_v)$ ,

$$\int_{b \in \mathbb{R}} p(b|a, E_v) \mathbf{1}(G_{v,a}(I_\delta(b)) \leq 2\delta/d^3) db \leq \frac{4}{d^{2.5}}.$$

By a union bound, it follows that  $\Pr_{(a_{i+1}, b_{i+1})} [(a_{i+1}, b_{i+1}) \in \text{Bad}(v)] \leq 5/d^{2.5}.$

■

Using these results, we can show that the probability of  $\mathcal{T}$  stopping at a non-leaf vertex is small—

**Lemma 2.** *If the number of samples  $m \leq d^{1.25}$  and the width of the branching program  $w \leq 2^{d^2/4}$ , then the probability of  $\mathcal{T}$  stopping at a non-leaf vertex is at most  $1/(2d)$ .*

**Proof.** By Lemma 4, the probability that  $\mathcal{T}$  reaches a significant vertex and hence stops due to the first reason is at most  $2^{-d}$ . If  $\mathcal{T}$  does not reach a significant vertex, then by Lemma 5, the probability of stopping due to the second reason at any non-significant vertex is at most  $2^{-2d}$ . Taking a union bound over all the  $m \leq d^{1.25}$  steps, the probability of stopping due to the second reason is at most  $2^{-d}$ . By Lemma 6, the probability of getting a bad sample  $(a, b)$  at any time step and hence stopping due to the third reason is at most  $5/d^{2.5}$ . Taking a union bound over the  $m \leq d^{1.25}$  time steps, the probability of stopping due to the third reason at any time step is at most  $5/d^{1.25}$ . Hence the overall probability of the truncated path  $\mathcal{T}$  stopping at a non-leaf vertex is at most  $1/(2d)$ .

■

## 6 Bounding the Probability of Significant Vertices

In this section, we bound the probability of the truncated path  $\mathcal{T}$  reaching a significant vertex in the  $t$ -th stage, for any  $t$ . We begin by first finding an expression for the posterior distribution  $\mathbb{P}_{x|e}$  of  $x$  conditioned on traversing an edge  $e$ , and then upper bound the norm of a significant vertex  $s$  in the  $t$ -th stage  $B_t$  of  $B$ .

**Relating  $\mathbb{P}_{x|v}$  and  $\mathbb{P}_{x|e}$ , and bounding  $\|\mathbb{P}_{x|s}\|_2$**

We relate  $\mathbb{P}_{x|v}$  and  $\mathbb{P}_{x|e}$ . Recall that  $I_\delta(b)$  is the interval  $[b - \delta, b + \delta]$ . We claim that,

**Lemma 7.** *For any  $e$  labeled by  $(a, b)$ , such that  $p(e) > 0$ ,*

$$\mathbb{P}_{x|e}(x') = \begin{cases} \mathbb{P}_{x|v}(x')/c_e & \text{if } x' \notin \text{Sig}(v) \text{ and } a^T x' \in I_\delta(b) \\ 0 & \text{if } x' \in \text{Sig}(v) \text{ or } a^T x' \notin I_\delta(b) \end{cases}$$

where  $c_e \geq \delta/d^3$ .

**Proof.** Let  $e$  be an edge labeled by  $(a, b)$ , such that  $p(e) > 0$ . Since  $p(e) > 0$ , the vertex  $v$  is not significant, as otherwise  $\mathcal{T}$  stops on  $v$ . Also, as  $p(e) > 0$ ,  $e \notin \text{Bad}(v)$ , as otherwise  $\mathcal{T}$  never traverses edge  $e$ .

If  $\mathcal{T}$  reaches  $v$  it traverses the edge  $e$  if and only if:  $x \notin \text{Sig}(v)$  (as otherwise  $\mathcal{T}$  stops on  $v$ ) and the next sample received is  $(a, b)$ . Also, note that  $b = a^T x + \eta$ , where the noise  $\eta$  is uniform on  $[-\delta, \delta]$ . Hence the set of  $x'$  which are consistent with the example  $(a, b)$  are those where  $a^T x' \in I_\delta(b)$ . Therefore for any  $x' \in \mathcal{S}_d$ ,

$$\mathbb{P}_{x|e}(x') = \begin{cases} \mathbb{P}_{x|v}(x')/c_e & \text{if } x' \notin \text{Sig}(v) \text{ and } a^T x' \in I_\delta(b) \\ 0 & \text{if } x' \in \text{Sig}(v) \text{ or } a^T x' \notin I_\delta(b) \end{cases}$$

where  $c_e$  is a normalization factor, given by

$$c_e = \int_{x': x' \notin \text{Sig}(v) \wedge a^T x' \in I_\delta(b)} \mathbb{P}_{x|v}(x') dx' = \Pr_x[(x \notin \text{Sig}(v) \wedge a^T x \in I_\delta(b) | E_v)].$$

Since  $v$  is not significant, by Lemma 5,

$$\Pr_x[x \in \text{Sig}(v) | E_v] \leq 2^{-2d}.$$

Also, since  $(a, b) \notin \text{Bad}(v)$ ,

$$\Pr_x[a^T x \notin I_\delta(b) | E_v] \leq 1 - 2\delta/d^3.$$

Hence by a union bound and using the fact that  $\delta \geq 2^{-d/5}$ ,

$$c_e \geq 1 - (1 - 2\delta/d^3 + 2^{-2d}) \geq \delta/d^3.$$

■

We now show that  $\|\mathbb{P}_{x|s}\|_2$  cannot be too large. To show this, we will first show that  $\|\mathbb{P}_{x|e}\|_2$  cannot be too large for any edge  $e$  such that the  $p(e) > 0$ .

**Lemma 8.** *For any edge  $e$  in the  $t$ -th stage  $B_t$  of the branching program such that  $p(e) > 0$ ,  $\|\mathbb{P}_{x|e}\|_2 \leq \frac{(d^3/\delta)(2/\epsilon_t)^d}{C_d}$ .*

**Proof.** Let  $e$  be an edge of the branching program from vertex  $v$  to vertex  $u$  such that  $p(e) > 0$ . Since  $p(e) > 0$ , the vertex  $v$  is not significant (as otherwise  $\mathcal{T}$  stops on  $v$  and  $p(e) = 0$ ). As  $v$  is not significant,

$$\|\mathbb{P}_{x|v}\|_2 \leq \frac{(2/\epsilon_t)^d}{C_d}.$$

By Lemma 7,

$$\mathbb{P}_{x|e}(x') = \begin{cases} \mathbb{P}_{x|v}(x')/c_e & \text{if } x' \notin \text{Sig}(v) \text{ and } a^T x' \in I_\delta(b) \\ 0 & \text{if } x' \in \text{Sig}(v) \text{ or } a^T x' \notin I_\delta(b) \end{cases}$$

where  $c_e \geq \delta/d^3$ . Therefore,  $\|\mathbb{P}_{x|e}\|_2 \leq \frac{(d^3/\delta)(2/\epsilon_t)^d}{C_d}$ .

■

We now use Lemma 8 to bound  $\|\mathbb{P}_{x|s}\|_2$ .

**Lemma 9.** *For any significant vertex  $s$  in the  $t$ -th stage  $B_t$  of the branching program,  $\|\mathbb{P}_{x|s}\|_2 \leq \frac{(d^3/\delta)(2/\epsilon_t)^d}{C_d}$ .*

**Proof.** Let  $s$  be a significant vertex in  $B_t$ . Let  $\Gamma_{in}(s)$  be the set of edges  $e$  going into  $s$ . We can write,

$$\int_{e \in \Gamma_{in}(s)} p(e) de = \Pr(s).$$

By the law of total probability, for every  $x' \in \mathcal{S}_d$ ,

$$\mathbb{P}_{x|s}(x') = \int_{e \in \Gamma_{in}(s)} \frac{p(e)}{\Pr(v)} \cdot \mathbb{P}_{x|e}(x') de.$$

By using Jensen's inequality,

$$\mathbb{P}_{x|s}(x')^2 \leq \int_{e \in \Gamma_{in}(s)} \frac{p(e)}{\Pr(v)} \cdot \mathbb{P}_{x|e}(x')^2 de.$$

Summing over all  $x' \in \mathcal{S}_d$ ,

$$\|\mathbb{P}_{x|s}\|_2^2 \leq \int_{e \in \Gamma_{in}(s)} \frac{p(e)}{\Pr(v)} \cdot \|\mathbb{P}_{x|e}\|_2^2 de.$$

By Lemma 8, for any edge  $e \in \Gamma_{in}(s)$ ,  $\|\mathbb{P}_{x|e}\|_2^2 \leq \left( \frac{(d^3/\delta)(2/\epsilon_t)^d}{C_d} \right)^2$ . Hence  $\|\mathbb{P}_{x|s}\|_2 \leq \frac{(d^3/\delta)(2/\epsilon_t)^d}{C_d}$ . ■

### Similarity to a Target Distribution

To show that the probability of  $\mathcal{T}$  reaching a significant vertex is small, we will argue that the posterior of  $x$  on seeing a new example is not significantly similar to the target posterior distribution of a significant vertex. We use the inner product of two distributions to measure their similarity, and define it as follows. For two functions  $f, g : \mathcal{S}_d \rightarrow \mathbb{R}^+$ , define the inner product

$$\langle f, g \rangle = \mathbb{E}_{z \in \mathcal{S}_d} [f(z)g(z)].$$

Note that for a significant vertex  $s$  in the  $t$ -th stage,

$$\langle \mathbb{P}_{x|s}, \mathbb{P}_{x|s} \rangle = \|\mathbb{P}_{x|s}\|_2^2 > \frac{(2/\epsilon_t)^{2d}}{C_d^2}. \quad (1)$$

We now bound the inner product of  $\mathbb{P}_{x|s}$  with all states  $v_0$  in the first layer of the  $t$ -th stage  $B_t$  of  $B$ .

**Lemma 10.** *For all states  $v_0$  with  $\Pr(v_0) > 0$  in the first layer of the  $t$ -th stage  $B_t$  of  $B$ ,*

$$\langle \mathbb{P}_{x|v_0}, \mathbb{P}_{x|s} \rangle \leq \frac{(d^3/\delta)(\sqrt{2}/\epsilon_t)^{2d}}{C_d^2}.$$

**Proof.** We claim that  $\|\mathbb{P}_{x|v_0}\|_2 \leq \frac{(d^3/\delta)(1/\epsilon_t)^d}{C_d}$  for all states  $v_0$  in the first layer of  $B_t$ . Consider the  $(t-1)$ -th stage  $B_{t-1}$  of the branching program  $B$ . The truncated path  $\mathcal{T}$  stops at any significant vertex, and recall that a significant vertex for the  $(t-1)$ th stage is defined as a vertex  $s$  where

$$\|\mathbb{P}_{x|s}\|_2 > \frac{(2/\epsilon_{t-1})^d}{C_d}.$$

Hence for all non-significant vertices  $v$  in the  $(t-1)$ -th stage  $B_{t-1}$  of  $B$ ,

$$\|\mathbb{P}_{x|v}\|_2 \leq \frac{(2/\epsilon_{t-1})^d}{C_d}.$$

Also, by Lemma 9 for all significant vertices  $s$  in  $B_{t-1}$ ,

$$\|\mathbb{P}_{x|s}\|_2 \leq \frac{(d^3/\delta)(2/\epsilon_{t-1})^d}{C_d}.$$

Hence for all vertices  $v$  in  $B_{t-1}$  with  $\Pr(v) > 0$ ,

$$\|\mathbb{P}_{x|v}\|_2 \leq \frac{(d^3/\delta)(2/\epsilon_{t-1})^d}{C_d}.$$

Note that  $\epsilon_{t-1} = 2\epsilon_t$ , hence  $\|\mathbb{P}_{x|v_0}\|_2 \leq \frac{(d^3/\delta)(1/\epsilon_t)^d}{C_d}$  for all states  $v_0 \in L_0$  with  $\Pr(v_0) > 0$ , as  $L_0$  is also the last layer of  $B_{t-1}$ . Now by using Cauchy Schwartz,

$$\langle \mathbb{P}_{x|v_0}, \mathbb{P}_{x|s} \rangle \leq \frac{(d^3/\delta)(\sqrt{2}/\epsilon_t)^{2d}}{C_d^2}.$$

■

Note that the inner product of  $\mathbb{P}_{x|s}$  with itself is larger than the inner product of  $\mathbb{P}_{x|s}$  with  $\mathbb{P}_{x|v}$  for  $v$  in the first layer by a factor of about  $2^{\Theta(d)}$ , and in the next section we will argue that this inner product cannot increase too quickly in a small number of time steps, via a suitable potential function.

## Progress Towards Target Distribution

In this section, we bound how much progress  $\mathcal{T}$  can make towards a significant vertex  $s$  in the  $t$ -th stage  $B_t$  of  $B$ . For notational convenience, we will reindex all the layers in the  $t$ -th stage  $B_t$  so that the first layer in  $B_t$  is labelled as  $L_0$ .

Let  $L_i$  denote the set of all vertices in the  $i$ th layer of the  $t$ -th stage  $B_t$ , with  $\Pr(v) > 0$ . Let  $\Gamma_i$  denote the set of all edges from the  $(i-1)$ th layer to the  $i$ th layer of  $B_t$ . For  $i \in \{0, \dots, m_t\}$  and  $\beta = 1/2$ , let

$$\mathcal{Z}_i = \sum_{v \in L_i} \Pr(v) \cdot \langle \mathbb{P}_{x|v}, \mathbb{P}_{x|s} \rangle^{\beta d}. \quad (2)$$

For  $i \in \{0, \dots, m_t\}$ , let

$$\mathcal{Z}'_i = \int_{e \in \Gamma_i} p(e) \cdot \langle \mathbb{P}_{x|e}, \mathbb{P}_{x|s} \rangle^{\beta d} de. \quad (3)$$

Note that by Lemma 10,

$$\mathcal{Z}_0 \leq \left( \frac{(d^3/\delta)(\sqrt{2}/\epsilon_t)^{2d}}{C_d^2} \right)^{\beta d}. \quad (4)$$

The goal of the next three Lemmas is to bound how much  $\mathcal{Z}_i$  can increase at every step. Lemma 11 does most of the heavy-lifting, and shows that for a fixed vertex  $v$ , the contribution to the potential  $\mathcal{Z}'_i$  from  $v$ 's outgoing edges is not much larger than  $v$ 's contribution to  $\mathcal{Z}_i$ . Lemma 13 uses Lemma 11 to show that  $\mathcal{Z}'_i$  is not much larger than  $\mathcal{Z}$ . Finally, Lemma 14 shows that  $\mathcal{Z}_{i+1} \leq \mathcal{Z}_i$  by a convexity argument.

**Lemma 11.** Consider the  $t$ -th stage  $B_t$  of  $B$ , and let  $s$  be a significant vertex in  $B_t$ . For every vertex  $v$  of  $B_t$  such that  $\Pr(v) > 0$ ,

$$\int_{e \in \Gamma_{out}(v)} \frac{p(e)}{\Pr(v)} \cdot \langle \mathbb{P}_{x|e}, \mathbb{P}_{x|s} \rangle^{\beta d} de \leq \langle \mathbb{P}_{x|v}, \mathbb{P}_{x|s} \rangle^{\beta d} \cdot (C' \epsilon_t^{-20} d^3)^{\beta d} + \left( \frac{d^3 (1/\epsilon_t)^{2d}}{\delta C_d^2} \right)^{\beta d},$$

where  $C'$  is an absolute constant.

**Proof.** If  $v$  is a significant vertex or if  $v$  is a leaf of the branching program, then  $\Gamma_{out}(v)$  is the empty set and hence the claim is trivially true. Hence we will assume that  $v$  is not a significant vertex or a leaf.

Define  $P : \mathcal{S}_d \rightarrow \mathbb{R}^+$  as follows. For any  $x' \in \mathcal{S}_d$ ,

$$P(x') = \begin{cases} 0 & \text{if } x' \in \text{Sig}(v) \\ \mathbb{P}_{x|v}(x') & \text{if } x' \notin \text{Sig}(v). \end{cases}$$

Note that by the definition of  $\text{Sig}(v)$ , for any  $x' \in \mathcal{S}_d$ ,

$$P(x') \leq \frac{(4/\epsilon_t)^{2d}}{C_d}. \quad (5)$$

Define  $f : \mathcal{S}_d \rightarrow \mathbb{R}^+$  as follows. For any  $x' \in \mathcal{S}_d$ ,

$$f(x') = P(x') \cdot \mathbb{P}_{x|s}(x').$$

By Lemma 9 and Eq. (5),

$$\|f\|_2 \leq \|P\|_\infty \|\mathbb{P}_{x|s}\|_2 \leq \frac{d^3 (4/\epsilon_t)^{3d}}{\delta C_d^2}. \quad (6)$$

By Lemma 7, for any edge  $e \in \Gamma_{out}(v)$  labeled by  $(a, b)$  and for any  $x' \in \mathcal{S}_d$ ,

$$\mathbb{P}_{x|e}(x') = \begin{cases} 0 & \text{if } a^T x' \notin I_\delta(b) \\ P(x') \cdot c_e^{-1} & \text{if } a^T x' \in I_\delta(b) \end{cases}$$

where  $c_e \geq \delta/d^3$ . Hence for any edge  $e \in \Gamma_{out}(v)$  labeled by  $(a, b)$  and any  $x' \in \mathcal{S}_d$  we can write,

$$\mathbb{P}_{x|e}(x') \mathbb{P}_{x|s}(x') = \begin{cases} 0 & \text{if } a^T x' \notin I_\delta(b) \\ f(x') \cdot c_e^{-1} & \text{if } a^T x' \in I_\delta(b). \end{cases}$$

Let

$$F = \int_{x' \in \mathcal{S}_d} f(x') dx'.$$

Recall that for uniform distribution  $\mathcal{U}_d$  on  $\mathcal{S}_d$ ,  $\mathcal{U}_d(x) = 1/C_d$  for all  $x \in \mathcal{S}_d$ . Hence we can write,

$$\begin{aligned} \langle \mathbb{P}_{x|e}, \mathbb{P}_{x|s} \rangle &= \mathbb{E}_{x' \sim \mathcal{U}_d} [\mathbb{P}_{x|e}(x') \cdot \mathbb{P}_{x|s}(x')] \\ &= (c_e C_d)^{-1} \int_{\{x' : a^T x' \in I_\delta(b)\}} f(x') dx'. \end{aligned} \quad (7)$$

We will now bound  $\langle \mathbb{P}_{x|e}, \mathbb{P}_{x|s} \rangle$  by considering two separate cases:



Case 1:  $F \leq (1/\epsilon_t)^{2d}/C_d$ .

In this case, we bound  $\int_{\{x': a^T x' \in I_\delta(b)\}} f(x') dx' \leq \int_{\{x' \in \mathcal{S}_d\}} f(x') dx' = F$ . As  $c_e^{-1} \leq d^3/\delta$ , using Eq. (7),

$$\langle \mathbb{P}_{x|e}, \mathbb{P}_{x|s} \rangle \leq \frac{d^3(1/\epsilon_t)^{2d}}{\delta C_d^2}.$$

Note that for an edge  $e$  labeled by  $(a, b)$ ,  $\frac{p(e)}{\Pr(v)} \leq p(a)p(b|a, E_v)$ , where the equality may not be true as the branching program could stop at  $v$  if  $e \in \text{Bad}(v)$ . Hence  $\int_{e \in \Gamma_{out}(v)} \frac{p(e)}{\Pr(v)} de \leq 1$  and Lemma 11 follows in this case.

Case 2:  $F > (1/\epsilon_t)^{2d}/C_d$ .

We rewrite  $\langle \mathbb{P}_{x|e}, \mathbb{P}_{x|s} \rangle$  as follows,

$$\langle \mathbb{P}_{x|e}, \mathbb{P}_{x|s} \rangle = F/C_d \cdot c_e^{-1} \int_{\{x': a^T x' \in I_\delta(b)\}} f(x')/F dx'. \quad (8)$$

For every  $x' \in \mathcal{S}_d$ , we define  $\tilde{f}(x') = f(x')/F$ . Note that  $\int_{x'} \tilde{f} dx' = 1$  and hence  $\tilde{f}$  is a distribution over  $\mathcal{S}_d$ . Also, we can bound  $\|\tilde{f}\|_2$  as follows,

$$\|\tilde{f}\|_2 \leq \|f\|_2/F \leq \frac{d^3(64/\epsilon_t)^d}{\delta C_d} \leq \frac{(100/\epsilon_t)^d}{C_d},$$

where we use the fact that  $\delta \geq 2^{-d/5}$  and Eq. (6) to bound  $\|f\|_2$ . By the definitions of  $P$  and  $f$ ,

$$F/C_d = \mathbb{E}_{x' \in \mathcal{R}\mathcal{S}_d} [f(x')] = \langle P, \mathbb{P}_{x|s} \rangle \leq \langle \mathbb{P}_{x|v}, \mathbb{P}_{x|s} \rangle. \quad (9)$$

By using Eqs. (8) and (9), we can write,

$$\begin{aligned} \int_{e \in \Gamma_{out}(v)} \frac{p(e)}{\Pr(v)} \langle \mathbb{P}_{x|e}, \mathbb{P}_{x|s} \rangle^{\beta d} de &\leq \int_{e \in \Gamma_{out}(v)} \frac{p(e)}{\Pr(v)} \left( F/C_d \cdot c_e^{-1} \int_{\{x': a^T x' \in I_\delta(b)\}} \tilde{f}(x') dx' \right)^{\beta d} de \\ &= (F/C_d)^{\beta d} c_e^{-\beta d} \int_{e \in \Gamma_{out}(v)} \frac{p(e)}{\Pr(v)} \left( \int_{\{x': a^T x' \in I_\delta(b)\}} \tilde{f}(x') dx' \right)^{\beta d} de \\ &\leq \langle \mathbb{P}_{x|v}, \mathbb{P}_{x|s} \rangle^{\beta d} c_e^{-\beta d} \int_{e \in \Gamma_{out}(v)} \frac{p(e)}{\Pr(v)} \left( \int_{\{x': a^T x' \in I_\delta(b)\}} \tilde{f}(x') dx' \right)^{\beta d} de. \end{aligned}$$

Recall that for an edge  $e$  labeled by  $(a, b)$ ,  $\frac{p(e)}{\Pr(v)} \leq p(a)p(b|a, E_v)$ . Using our notation  $\mathbb{E}_{x' \sim \tilde{f}}[\mathbf{1}(a^T x' \in I_\delta(b))] = G_{\tilde{f}, a}(I_\delta(b))$ , we can write,

$$\int_{e \in \Gamma_{out}(v)} \frac{p(e)}{\Pr(v)} \langle \mathbb{P}_{x|e}, \mathbb{P}_{x|s} \rangle^{\beta d} de \leq \langle \mathbb{P}_{x|v}, \mathbb{P}_{x|s} \rangle^{\beta d} c_e^{-\beta d} \int_{a \in \mathbb{R}^d} \int_{b \in \mathbb{R}} p(a) p(b|a, E_v) G_{\tilde{f}, a}(I_\delta(b))^{\beta d} db da.$$

Let  $\eta$  be a random variable uniform on  $[-\delta, \delta]$ , and  $u(\eta)$  denote its distribution. Note that  $p(b|a, E_v)$  is the distribution

of  $a^T x' + \eta$ , where  $x'$  is sampled from  $\mathbb{P}_{x|v}$  and noise  $\eta$  is added to  $a^T x'$ . Therefore,

$$\begin{aligned} \int_{e \in \Gamma_{out}(v)} \frac{p(e)}{\Pr(v)} \langle \mathbb{P}_{x|e}, \mathbb{P}_{x|s} \rangle^{\beta d} de &\leq \langle \mathbb{P}_{x|v}, \mathbb{P}_{x|s} \rangle^{\beta d} c_e^{-\beta d}. \\ &\int_{a \in \mathbb{R}^d} \int_{x' \in \mathcal{S}_d} \int_{\eta \in \mathbb{R}} p(a) \mathbb{P}_{x|v}(x') u(\eta) G_{\tilde{f},a}(I_\delta(a^T x' + \eta))^{\beta d} d\eta dx' da \\ &= \langle \mathbb{P}_{x|v}, \mathbb{P}_{x|s} \rangle^{\beta d} c_e^{-\beta d} \mathbb{E}_{\eta} \mathbb{E}_{x' \sim \mathbb{P}_{x|v}} \mathbb{E}_a \left[ G_{\tilde{f},a}(I_\delta(a^T x' + \eta))^{\beta d} \right]. \end{aligned}$$

We now use Lemma 12 to bound the expectation.

**Lemma 12.** *Let  $f$  be a distribution over the  $d$  dimensional sphere, with  $\|f\|_2 \leq \frac{(100/\epsilon)^d}{C_d}$  for some  $\epsilon \leq 1$ . Let  $x_0$  be any vector on the  $d$  dimensional unit sphere, and  $\eta_0$  be some constant. Then for any  $\ell \leq d/2$  and a universal constant  $C$ ,*

$$\mathbb{E}_a \left[ G_{f,a}(I_\delta(a^T x_0 + \eta_0))^\ell \right] \leq (2(C/\epsilon)^{20}\delta)^\ell.$$

By Lemma 12, if  $\beta \leq 1/2$ , then  $\mathbb{E}_a G_{\tilde{f},a}(I_\delta(a^T x' + \eta))^{\beta d} \leq (2(C/\epsilon_t)^{-20}\delta)^{\beta d}$  for any  $x'$  and  $\eta$ . Let  $C' = 2C^{20}$ . Therefore,

$$\int_{e \in \Gamma_{out}(v)} \frac{p(e)}{\Pr(v)} \langle \mathbb{P}_{x|e}, \mathbb{P}_{x|s} \rangle^{\beta d} de \leq \langle \mathbb{P}_{x|v}, \mathbb{P}_{x|s} \rangle^{\beta d} c_e^{-\beta d} (C' \epsilon_t^{-20} \delta)^{\beta d}.$$

As  $c_e \geq \delta/d^3$ , therefore,

$$\int_{e \in \Gamma_{out}(v)} \frac{p(e)}{\Pr(v)} \langle \mathbb{P}_{x|e}, \mathbb{P}_{x|s} \rangle^{\beta d} de \leq \langle \mathbb{P}_{x|v}, \mathbb{P}_{x|s} \rangle^{\beta d} (C' \epsilon_t^{-20} d^3)^{\beta d}.$$

■

**Lemma 13.** *Consider the  $t$ -th stage  $B_t$  of  $B$ , and let  $s$  be a significant vertex in  $B_t$ . Consider the potential functions defined in Eqs. (2) and (3) which track progress towards  $s$ . Then for every  $i \in \{1, \dots, m_t\}$ ,*

$$\mathcal{Z}'_i \leq \mathcal{Z}_{i-1} \cdot (C' \epsilon_t^{-20} d^3)^{\beta d} + \left( \frac{d^3(1/\epsilon_t)^d}{\delta C_d^2} \right)^{\beta d}.$$

**Proof.** Using Lemma 11,

$$\begin{aligned} \mathcal{Z}'_i &= \int_{e \in \Gamma_i} p(e) \cdot \langle \mathbb{P}_{x|e}, \mathbb{P}_{x|s} \rangle^{\beta d} de = \sum_{v \in L_{i-1}} \Pr(v) \cdot \int_{e \in \Gamma_{out}(v)} \frac{p(e)}{\Pr(v)} \langle \mathbb{P}_{x|e}, \mathbb{P}_{x|s} \rangle^{\beta d} de \\ &\leq \sum_{v \in L_{i-1}} \Pr(v) \cdot \left( \langle \mathbb{P}_{x|v}, \mathbb{P}_{x|s} \rangle^{\beta d} \cdot (C' \epsilon_t^{-20} d^3)^{\beta d} + \left( \frac{d^3(1/\epsilon_t)^{2d}}{\delta C_d^2} \right)^{\beta d} \right) \\ &\leq \mathcal{Z}_{i-1} (C' \epsilon_t^{-20} d^3)^{\beta d} + \sum_{v \in L_{i-1}} \Pr(v) \left( \frac{d^3(1/\epsilon_t)^{2d}}{\delta C_d^2} \right)^{\beta d} \\ &\leq \mathcal{Z}_{i-1} (C' \epsilon_t^{-20} d^3)^{\beta d} + \left( \frac{d^3(1/\epsilon_t)^{2d}}{\delta C_d^2} \right)^{\beta d}. \end{aligned}$$

■

**Lemma 14.** For every  $i \in \{1, \dots, m_t\}$ ,

$$\mathcal{Z}_i \leq \mathcal{Z}'_i.$$

**Proof.** For any vertex  $v$ , let  $\Gamma_{in}(v)$  be the set of edges  $e = (a, b)$  going into  $v$ . We can write,

$$\int_{e \in \Gamma_{in}(v)} p(e) de = \Pr(v).$$

By the law of total probability, for every  $v \in L_i$  and every  $x' \in \mathcal{S}_d$ ,

$$\begin{aligned} \mathbb{P}_{x|v}(x') &= \int_{e \in \Gamma_{in}(v)} \frac{p(e)}{\Pr(v)} \cdot \mathbb{P}_{x|e}(x') de, \\ \implies \langle \mathbb{P}_{x|v}, \mathbb{P}_{x|s} \rangle &= \int_{e \in \Gamma_{in}(v)} \frac{p(e)}{\Pr(v)} \cdot \langle \mathbb{P}_{x|e}, \mathbb{P}_{x|s} \rangle de. \end{aligned}$$

By using Jensen's inequality,

$$\langle \mathbb{P}_{x|v}, \mathbb{P}_{x|s} \rangle^{\beta d} \leq \int_{e \in \Gamma_{in}(v)} \frac{p(e)}{\Pr(v)} \cdot \langle \mathbb{P}_{x|e}, \mathbb{P}_{x|s} \rangle^{\beta d} de.$$

Summing over all  $v \in L_i$ ,

$$\begin{aligned} \mathcal{Z}_i &= \sum_{v \in L_i} \Pr(v) \cdot \langle \mathbb{P}_{x|v}, \mathbb{P}_{x|s} \rangle^{\beta d} \leq \sum_{v \in L_i} \Pr(v) \cdot \int_{e \in \Gamma_{in}(v)} \frac{p(e)}{\Pr(v)} \cdot \langle \mathbb{P}_{x|e}, \mathbb{P}_{x|s} \rangle^{\beta d} de, \\ &= \int_{e \in \Gamma_{in}(v)} p(e) \cdot \langle \mathbb{P}_{x|e}, \mathbb{P}_{x|s} \rangle^{\beta d} de = \mathcal{Z}'_i. \end{aligned}$$

■

We now use the previous two results to bound the potential  $\mathcal{Z}_i$  for any layer  $i$  in the stage  $B_t$ .

**Lemma 15.** If the length of the  $t$ -th stage  $m_t = \lceil \frac{c_0 d}{\log d + t} \rceil$  for sufficiently small constant  $c_0$ , then for every  $i \in \{1, \dots, m_t\}$ ,

$$\mathcal{Z}_i \leq \frac{(d^3/\delta)^{\beta d} (1/\epsilon_t)^{2\beta d^2} 2^{1.25\beta d^2}}{C_d^2}.$$

**Proof.** By Lemma 13 and 14,

$$\mathcal{Z}_i \leq \mathcal{Z}_{i-1} \cdot (C' \epsilon_t^{-20} d^3)^{\beta d} + \left( \frac{d^3 (1/\epsilon_t)^{2d}}{\delta C_d^2} \right)^{\beta d}.$$

As  $(C' \epsilon_t^{-20} d^3)^{\beta d} > 1$ ,  $\mathcal{Z}_i$  is monotonically increasing and hence  $\mathcal{Z}_i \geq \left( \frac{d^3 (1/\epsilon_t)^{2d}}{\delta C_d^2} \right)^{\beta d}$  for all  $i > 0$ . Therefore we can write,

$$\mathcal{Z}_i \leq \mathcal{Z}_{i-1} (1 + (C' \epsilon_t^{-20} d^3)^{\beta d}) \leq \mathcal{Z}_{i-1} \cdot (2C' \epsilon_t^{-20} d^3)^{\beta d}.$$

By Eq. (4),  $\mathcal{Z}_0 \leq \left( \frac{(d^3/\delta)(\sqrt{2}/\epsilon_t)^{2d}}{C_d^2} \right)^{\beta d}$ . Hence for every  $i \in \{1, \dots, m_t\}$ ,

$$\mathcal{Z}_i \leq \left( \frac{(d^3/\delta)(\sqrt{2}/\epsilon_t)^{2d}}{C_d^2} \right)^{\beta d} \cdot (2C'_t \epsilon_t^{-20} d^3)^{\beta d m}.$$

Note that for sufficiently small  $c_0$  and using the fact that  $\epsilon_t = 2^{-t}$ ,  $m_t = \lceil \frac{c_0 d}{\log d + t} \rceil = \lceil \frac{c_0 d}{\log(d/\epsilon_t)} \rceil \leq \frac{d}{4 \log(2C'_t \epsilon_t^{-20} d^3)}$ . Therefore,

$$\mathcal{Z}_i \leq \left( \frac{(d^3/\delta)(\sqrt{2}/\epsilon_t)^{2d}}{C_d^2} \right)^{\beta d} \cdot 2^{\beta d^2/4} = \frac{(d^3/\delta)^{\beta d} (1/\epsilon_t)^{2\beta d^2} 2^{1.25\beta d^2}}{C_d^2}.$$

■

**Lemma 4.** *If the total number of stages  $T \leq d^{1.25}$  and the width of the branching program  $w \leq 2^{d^2/4}$ , then the probability that  $\mathcal{T}$  reaches a significant vertex in any stage is at most  $2^{-d}$ .*

**Proof.** Consider a significant vertex  $s$  in the  $t$ -th stage  $B_t$  of  $B$ . Assume that  $s$  is in the  $i$ th layer of  $B_t$ . Then by Eq. (1),

$$\mathcal{Z}_i \geq \Pr(s) \cdot \langle \mathbb{P}_{x|s}, \mathbb{P}_{x|s} \rangle^{\beta d} \geq \Pr(s) \cdot \frac{(2/\epsilon_t)^{2\beta d^2}}{C_d^2}.$$

But by Lemma 15,

$$\mathcal{Z}_i \leq \frac{(d^3/\delta)^{\beta d} (1/\epsilon_t)^{2\beta d^2} 2^{1.25\beta d^2}}{C_d^2}.$$

Therefore,

$$\Pr(s) \leq (d^3/\delta)^{\beta d} 2^{-0.75\beta d^2}.$$

By a union bound over the at most  $d^2 \cdot 2^{0.5\beta d^2}$  vertices in any stage of the branching program and using the fact that  $\delta \geq 2^{-d/5}$ , the probability that  $\mathcal{T}$  reaches a significant vertex in the  $t$ -th stage is at most  $(d^5/\delta)^{\beta d} 2^{-\beta d^2/4} \leq 2^{-\beta d^2/20} \leq 2^{-2d}$ . By taking a union bound over  $T \leq d^{1.25}$  stages, the probability that  $\mathcal{T}$  reaches a significant vertex in any stage is at most  $2^{-d}$ . ■

## 7 Concentration Theorem for Projections of Distributions

In this section, we prove our concentration theorem for projections of high-dimensional distributions onto a random direction.

**Lemma 12.** *Let  $f$  be a distribution over the  $d$  dimensional sphere, with  $\|f\|_2 \leq \frac{(100/\epsilon)^d}{C_d}$  for some  $\epsilon \leq 1$ . Let  $x_0$  be any vector on the  $d$  dimensional unit sphere, and  $\eta_0$  be some constant. Then for any  $\ell \leq d/2$  and a universal constant  $C$ ,*

$$\mathbb{E}_a \left[ G_{f,a}(I_\delta(a^T x_0 + \eta_0))^\ell \right] \leq (2(C/\epsilon)^{20}\delta)^\ell.$$

**Proof.** All expectations over  $\{x_i, i \in [\ell]\}$  in this proof will be with  $x_i$  sampled from the distribution  $f$ . Recall from our definition,

$$\mathbb{E}_a \left[ G_{f,a}(I_\delta(a^T x_0 + \eta_0))^\ell \right] = \mathbb{E}_a \left[ \prod_{i=1}^{\ell} \mathbb{E}_{x_i} [\mathbf{1}_{I_\delta(a^T x_0 + \eta_0)}(a^T x_i)] \right]$$

As the distribution of  $a$  is a Gaussian and the distribution of the projection of a Gaussian along a fixed direction is well-understood, we will first interchange the order of  $a$  and  $x_i$  in the expectation. We can write,

$$\mathbb{E}_a \left[ G_{f,a}(I_\delta(a^T x_0 + \eta_0))^\ell \right] = \mathbb{E}_{\{x_i, i \in [\ell]\}} \left[ \mathbb{E}_a \left[ \prod_{i=1}^\ell \mathbf{1}_{I_\delta(a^T x_0 + \eta_0)}(a^T x_i) \right] \right].$$

We will write the vectors  $\{x_i : 0 \leq i \leq \ell\}$  in terms of a suitable orthogonal basis which will facilitate the analysis of the projection of  $a$  onto  $x_i$ . For all  $0 \leq i \leq \ell$ , let  $x_i = u_i + v_i$ , where  $u_i$  lies in the span of  $\{x_j : j < i\}$  and  $v_i$  is orthogonal to the span of  $\{x_j : j < i\}$ . Note that because  $a \sim N(0, I)$ , the components of  $a$  along any orthogonal basis of  $\mathbb{R}^d$  are independent  $N(0, 1)$  random variables. Hence the components  $\{v_{a,0}, \dots, v_{a,\ell}\}$  of  $a$  along the orthogonal directions  $\{v_0, \dots, v_\ell\}$  are independent  $N(0, 1)$  random variables. Note that  $a^T x_i$  is independent of  $\{v_{a,j} : j > i\}$ , as  $x_i$  is orthogonal to  $\{v_j : j > i\}$ . Let  $a^T x_0 + \eta_0 = b_0$ . Using this independence, we can rewrite the expectation as follows,

$$\begin{aligned} & \mathbb{E}_{\{x_i, i \in [\ell]\}} \left[ \mathbb{E}_a \left[ \prod_{i=1}^\ell \mathbf{1}_{I_\delta(b_0)}(a^T x_i) \right] \right] \\ &= \mathbb{E}_{\{x_i, i \in [\ell]\}} \mathbb{E}_{v_{a,0}} \left[ \mathbb{E}_{v_{a,1}} \left[ \mathbf{1}_{I_\delta(b_0)}(a^T x_1) \right] \mathbb{E}_{v_{a,2}} \left[ \mathbf{1}_{I_\delta(b_0)}(a^T x_2) \right] \dots \mathbb{E}_{v_{a,\ell}} \left[ \mathbf{1}_{I_\delta(b_0)}(a^T x_\ell) \right] \right] \dots \right]. \end{aligned}$$

We will now upper bound  $\mathbb{E}_{v_{a,i}} \left[ \mathbf{1}_{I_\delta(b_0)}(a^T x_i) \right]$  for any value of  $\{v_{a,j} : j < i\}$ . Note that for a fixed value of the components  $v_i$  and  $u_i$  of  $x_i$ ,  $v_{a,i}^T v_i$  is a Gaussian with mean 0 and standard deviation  $\|v_i\|_2$  and is independent of  $a^T u_i$  as  $u_i$  and  $v_i$  are orthogonal. Also, note that for any value of  $a^T u_i$  the probability that  $a^T x_i$  lies within the interval  $I_\delta(b) = [b - \delta, b + \delta]$  equals the probability mass of the distribution of  $v_{a,i}^T v_i$  in the interval  $I_\delta(b - a^T u_i) = [b - a^T u_i - \delta, b - a^T u_i + \delta]$ . As the distribution of  $v_{a,i}^T v_i$  is Gaussian with mean 0, the probability mass in any interval  $I_\delta(b - a^T u_i)$  is upper bounded by the probability mass in the interval  $I_\delta(0) = [-\delta, \delta]$  centered at 0. Further, because the probability mass of a Gaussian with standard deviation  $\sigma$  in the interval  $I_\delta(0) = [-\delta, \delta]$  is at most  $\max\{\delta/\sigma, 1\}$  and the distribution of  $v_{a,i}^T v_i$  is a Gaussian with standard deviation  $\|v_i\|_2$ , the probability mass in the interval  $I_\delta(0)$  is at most  $\max\{\delta/\|v_i\|_2, 1\} \leq \delta \max\{1/\|v_i\|_2, 1/\delta\}$ . Hence we can simplify the expectation as,

$$\mathbb{E}_{\{x_i, i \in [\ell]\}} \left[ \mathbb{E}_a \left[ \prod_{i=1}^\ell \mathbf{1}_{I_\delta(b_0)}(a^T x_i) \right] \right] \leq \delta^\ell \mathbb{E}_{\{x_i, i \in [\ell]\}} \left[ \frac{1}{\prod_{i=1}^\ell \min\{\|v_i\|_2, \delta\}} \right].$$

Therefore our goal now is to lower bound the component  $v_i$  of  $x_i$  which is orthogonal to the vectors  $\{x_j, j < i\}$ . This is where we will use our upper bound on  $\|\tilde{f}\|_2$ . Intuitively, if  $\tilde{f}$  is not too large then  $\tilde{f}$  is not a highly concentrated distribution and hence random vectors drawn from  $\tilde{f}$  will not be too close to each other. We formalize this in Lemma 16 which shows that if  $\|\tilde{f}\|_2$  is not too large, then the probability of  $\|v_i\|_2$  being small is also small.

**Lemma 16.** *Let  $f$  be a distribution over the  $d$  dimensional sphere, with  $\|f\|_2 \leq \frac{(100/\epsilon)^d}{C_d}$  for some  $\epsilon \leq 1$ . Let  $\{x_0, \dots, x_{i-1}\}$  be an arbitrary set of  $i$  vectors for some  $i \leq d/2$ , and  $x_i$  be a vector sampled from  $f$ . Let  $v_i$  be the component of  $x_i$  orthogonal to  $\{x_0, \dots, x_{i-1}\}$ . Then for a sufficiently large universal constant  $C$ ,*

$$\Pr[\|v_i\|_2 \leq (\epsilon/C)^{20}] \leq (\epsilon/2)^{d/2}.$$

The result now follows with some computation. Let  $\{X_i, i \in [\ell]\}$  be independent random variables each of which take the value  $\delta$  with probability  $(\epsilon/2)^\ell$  and  $(\epsilon/C)^{20}$  otherwise. Note that  $\mathbb{E}[1/X_i] \leq (1/\delta)(\epsilon/2)^\ell + (C/\epsilon)^{20} \leq (2(C/\epsilon)^{20})$  for  $\delta \geq 2^{-d/5}$ . Hence by Lemma 16,

$$\mathbb{E}_{\{x_i, i \in [\ell]\}} \left[ \frac{1}{\prod_{i=1}^\ell \min\{\|v_i\|_2, \delta\}} \right] \leq \mathbb{E}_{\{X_i, i \in [\ell]\}} \left[ \frac{1}{\prod_{i=1}^\ell X_i} \right] = \prod_{i=1}^\ell \mathbb{E} \left[ \frac{1}{X_i} \right] \leq (2(C/\epsilon)^{20})^\ell.$$

Hence,

$$\mathbb{E}_{\{x_i, i \in [\ell]\}} \left[ \mathbb{E}_a \left[ \prod_{i=1}^\ell \mathbf{1}_{I_\delta(b_0)}(a^T x_i) \right] \right] \leq (2(C/\epsilon)^{20} \delta)^\ell.$$

■

We remark that the bound in Lemma 1 is tight up to the constant  $C$  and the constant 20 in the exponent of  $\epsilon$ . This follows from the case where  $\tilde{f}$  is the uniform distribution on all points  $x$  on the unit sphere which are distance at most  $\epsilon$  from a fixed point  $x_0$ .

We now prove Lemma 16. Lemma 3, which is used in the proof of Theorem 1, is a corollary of Lemma 16 and follows by appropriately rescaling the constants in the statement of Lemma 16.

**Lemma 16.** *Let  $f$  be a distribution over the  $d$  dimensional sphere, with  $\|f\|_2 \leq \frac{(100/\epsilon)^d}{C_d}$  for some  $\epsilon \leq 1$ . Let  $\{x_0, \dots, x_{i-1}\}$  be an arbitrary set of  $i$  vectors for some  $i \leq d/2$ , and  $x_i$  be a vector sampled from  $f$ . Let  $v_i$  be the component of  $x_i$  orthogonal to  $\{x_0, \dots, x_{i-1}\}$ . Then for a sufficiently large universal constant  $C$ ,*

$$\Pr[\|v_i\|_2 \leq (\epsilon/C)^{20}] \leq (\epsilon/2)^{d/2}.$$

**Proof.** We will first upper bound the probability that  $\|v_i\|_2 \leq (\epsilon/C)^{20}$  for any  $i \leq d/2$  if  $x_i$  is drawn uniformly at random from the unit sphere. As the Gaussian distribution is spherically symmetric, we can assume without loss of generality that  $\{x_j : j < i\}$  span the first  $i$  basis directions. Sampling  $x_i$  uniformly at random from the unit sphere is equivalent to first sampling a vector  $\tilde{x}_i$  whose each coordinate is sampled independently from a  $N(0, 1/d)$  distribution, and then setting  $x_i = \tilde{x}_i / \|\tilde{x}_i\|_2$ . Using this formulation, let the  $j$ th coordinate of  $\tilde{x}_i$  be  $z_j$ , where  $z_j \sim N(0, 1/d)$ . We will first show that with probability  $1 - \epsilon^d$ ,  $\|\tilde{x}_i\|_2 \leq 2\sqrt{1/\epsilon}$ . This follows from the following tail bound for  $\chi^2$  random variables. Note that  $r^2 = d \sum_{j=1}^d z_j^2$  is the sum of  $d$  standard Gaussian random variable and hence is a  $\chi^2$  random variables with  $d$  degrees of freedom. We use the following concentration inequality for a  $\chi^2$  random variable  $r$  with  $d$  degrees of freedom (Lemma 1 in Laurent and Massart [2000]),

$$\Pr[r^2 - d \geq 2\sqrt{dt} + 2t] \leq e^{-t}, \quad \forall t > 0.$$

Choosing  $t = d \log(1/\epsilon)$ ,

$$\Pr[r^2 - d \geq 2d\sqrt{\log(1/\epsilon)} + 2d \log(1/\epsilon)] \leq \epsilon^d.$$

Using the fact that  $4 \max(\log(1/\epsilon), \sqrt{\log(1/\epsilon)}) \leq 3/\epsilon$  for  $0 \leq \epsilon \leq 1$ ,  $\Pr[r^2 - d \geq 3d/\epsilon] \leq \epsilon^d$ . Hence  $\|\tilde{x}_i\|_2 \leq \sqrt{1 + 3/\epsilon} \leq \sqrt{4/\epsilon}$  with failure probability  $\epsilon^d$ .

We now claim that  $\Pr[\sum_{j=i+1}^d z_j^2 \leq \epsilon^2/64] \leq \epsilon^{d/4}$ . Note that if  $\sum_{j=i+1}^d z_j^2 \leq \epsilon^2/64$ , then by Markov's inequality  $z_j^2 \leq \epsilon^2/(32(d-i)) \leq \epsilon^2/(16d)$  for at least half of the random variables  $\{z_j : j \in [i+1, d]\}$ . As  $z_j$  is a  $N(0, 1/d)$  random variable, hence  $\Pr[|z_j| \leq \epsilon/(4\sqrt{d})] \leq \epsilon/4$ . Hence the probability that  $z_j^2 \leq \epsilon^2/(4d)$  for at least half of the random variables  $\{z_j : j \in [i+1, d]\}$  is at most the probability that at least half of a set of  $(d-i)$  independent coins land heads when all of them are flipped, given that each of them has probability  $\epsilon/4$  of landing head. Let  $X_j$  be the indicator random variable denoting the event  $|z_j| \leq \epsilon/(4\sqrt{d})$ , note that  $\Pr[X_j = 1] \leq \epsilon/4$ . Let  $X = \sum_{j=i+1}^d X_j$ . Then by standard Chernoff bounds,

$$\Pr[X \geq (d-i)/2] \leq \exp\{-(d-i)D(1/2 \parallel \epsilon/4)\}$$

where  $D(1/2 \parallel \epsilon/4)$  is the relative entropy of  $1/2$  with respect to  $\epsilon/4$ . Note that  $D(1/2 \parallel \epsilon/4) \geq \frac{-\ln(\epsilon)}{2}$  and  $(d-i) \geq d/2$ . Therefore,

$$\Pr[X \geq (d-i)/2] \leq \epsilon^{d/4}.$$

Hence for any  $i \leq d/2$ ,  $\sum_{j=i+1}^d z_j^2 \geq \epsilon^2/64$  with failure probability  $\epsilon^{d/4}$ . Note that  $\|v_i\|_2^2 = \sum_{j=i+1}^d z_j^2 / \|\tilde{x}_i\|_2^2$ , and we have shown that  $\|\tilde{x}_i\|_2^2 \leq 4/\epsilon$  with failure probability  $\epsilon^d$ . Therefore, by a union bound,  $\Pr[\|v_i\|_2 \leq \epsilon^{1.5}/16] \leq \epsilon^{d/4} + \epsilon^d \leq 2\epsilon^{d/4}$ . For any  $C \geq 200$  and  $\epsilon \in (0, 1]$ , this implies that  $\Pr[\|v_i\|_2 \leq (\epsilon/C)^{20}] \leq (\epsilon/100)^{3d}$  when  $x_i$  is drawn uniformly from the unit sphere. We next show that this implies that  $\Pr[\|v_i\|_2 \leq (\epsilon/C)^{20}]$  is also small when  $x_i$  is drawn uniformly from some distribution  $f$  such that  $\|f\|_2$  is small.

Let  $\mathcal{X}$  be the set of all  $x_i$  on the unit sphere such that  $\|v_i\|_2 \leq (\epsilon/C)^{20}$ . We have shown that for  $x_i$  drawn uniformly from the unit sphere,  $\Pr[x_i \in \mathcal{X}] \leq (\epsilon/100)^{3d}$ . Note that if  $\mathcal{X}$  has probability  $p$  under the distribution  $f$ , then  $\|f\|_2 \geq \frac{p(100/\epsilon)^{3d/2}}{C_d}$ . This is because the uniform distribution on  $\mathcal{X}$  has norm  $\frac{(100/\epsilon)^{3d/2}}{C_d}$ . Therefore if  $\|f\|_2 \leq \frac{(100/\epsilon)^d}{C_d}$ , then  $p \leq (\epsilon/2)^{d/2}$ . ■



## References

- Ran Raz. Fast learning requires good memory: A time-space lower bound for parity learning. In *Foundations of Computer Science (FOCS), 2016 IEEE 57th Annual Symposium on*, pages 266–275. IEEE, 2016.
- Ran Raz. A time-space lower bound for a large class of learning problems. In *Foundations of Computer Science (FOCS), 2017 IEEE 58th Annual Symposium on*, pages 732–742. IEEE, 2017.
- Paul Beame, Shayan Oveis Gharan, and Xin Yang. Time-space tradeoffs for learning finite functions from random evaluations, with applications to polynomials. In *Conference On Learning Theory*, pages 843–856, 2018.
- Gillat Kol, Ran Raz, and Avishay Tal. Time-space hardness of learning sparse parities. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1067–1080. ACM, 2017.
- Dana Moshkovitz and Michal Moshkovitz. Mixing implies lower bounds for space bounded learning. In *Conference on Learning Theory*, pages 1516–1566, 2017.
- Dana Moshkovitz and Michal Moshkovitz. Entropy samplers and strong generic lower bounds for space bounded learning. In *9th Innovations in Theoretical Computer Science Conference (ITCS)*, 2018.
- Sumegha Garg, Ran Raz, and Avishay Tal. Extractor-based time-space lower bounds for learning. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 990–1002. ACM, 2018.
- Jacob Steinhardt, Gregory Valiant, and Stefan Wager. Memory, communication, and statistical queries. In *Conference on Learning Theory*, pages 1490–1516, 2016.
- A.S. Nemirovski and D.B Yudin. Problem complexity and method efficiency in optimization. 1983.
- Yurii Nesterov. Introductory lectures on convex optimization: A basic course. 2014.
- Sébastien Bubeck. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning*, 8(3-4):231–357, 2015.
- Arkadi Nemirovski. On parallel complexity of nonsmooth convex optimization. *J. Complexity*, 10(4):451–463, 1994.
- Eric Balkanski and Yaron Singer. Parallelization does not accelerate convex optimization: Adaptivity lower bounds for non-smooth convex minimization. *CoRR*, abs/1808.03880, 2018.
- Jelena Diakonikolas and Cristóbal Guzmán. Lower bounds for parallel and randomized convex optimization. *CoRR*, abs/1811.01903, 2018.
- Thomas Strohmer and Roman Vershynin. A randomized solver for linear systems with exponential convergence. In *10th International Workshop on Randomization and Computation, RANDOM*, pages 499–507, 2006.
- Chonghai Hu, James T. Kwok, and Weike Pan. Accelerated gradient methods for stochastic optimization and online learning. In *Advances in Neural Information Processing Systems*, pages 781–789, 2009.
- Guanghui Lan. An optimal method for stochastic composite optimization. *Math. Program.*, 133(1-2):365–397, 2012. doi: 10.1007/s10107-010-0434-y.
- Saeed Ghadimi and Guanghui Lan. Optimal stochastic approximation algorithms for strongly convex stochastic composite optimization, II: shrinking procedures and optimal algorithms. *SIAM Journal on Optimization*, 23(4):2061–2089, 2013. doi: 10.1137/110848876.
- Yin Tat Lee and Aaron Sidford. Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 147–156, 2013.

- Roy Frostig, Rong Ge, Sham M. Kakade, and Aaron Sidford. Competing with the empirical risk minimizer in a single pass. In *Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3-6, 2015*, pages 728–763, 2015.
- Ji Liu and Stephen J. Wright. An accelerated randomized kaczmarz algorithm. *Math. Comput.*, 85(297):153–178, 2016.
- Deanna Needell, Nathan Srebro, and Rachel Ward. Stochastic gradient descent, weighted sampling, and the randomized Kaczmarz algorithm. *Math. Program.*, 155(1-2):549–573, 2016. doi: 10.1007/s10107-015-0864-7.
- Aymeric Dieuleveut, Nicolas Flammarion, and Francis Bach. Harder, better, faster, stronger convergence rates for least-squares regression. *J. Mach. Learn. Res.*, 18(1):3520–3570, 2017.
- Prateek Jain, Sham M. Kakade, Rahul Kidambi, Praneeth Netrapalli, and Aaron Sidford. Accelerating stochastic gradient descent for least squares regression. In *Conference On Learning Theory, COLT 2018, Stockholm, Sweden, 6-9 July 2018.*, pages 545–604, 2018.
- Weihao Kong. Personal communication. 2018.
- Jacob Steinhardt and John Duchi. Minimax rates for memory-bounded sparse linear regression. In *Conference on Learning Theory*, pages 1564–1587, 2015.
- Ohad Shamir. Fundamental limits of online and distributed algorithms for statistical learning and estimation. In *Advances in Neural Information Processing Systems*, pages 163–171, 2014.
- Yuval Dagan and Ohad Shamir. Detecting correlations with little memory and communication. In *Conference on Learning Theory (COLT)*, 2018.
- Mark Braverman, Ankit Garg, Tengyu Ma, Huy L Nguyen, and David P Woodruff. Communication lower bounds for statistical estimation problems via a distributed data processing inequality. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 1011–1020. ACM, 2016.
- Noga Alon, Yossi Matias, and Mario Szegedy. The space complexity of approximating the frequency moments. *Journal of Computer and system sciences*, 58(1):137–147, 1999.
- Ziv Bar-Yossef, Thathachar S Jayram, Ravi Kumar, and D Sivakumar. An information statistics approach to data stream and communication complexity. *Journal of Computer and System Sciences*, 68(4):702–732, 2004.
- Blake E. Woodworth and Nati Srebro. Tight complexity bounds for optimizing composite objectives. In *Advances in Neural Information Processing Systems*, pages 3639–3647, 2016.
- Gábor Braun, Cristobal Guzman, and Sebastian Pokutta. Lower bounds on the oracle complexity of nonsmooth convex optimization via information theory. *IEEE Trans. Information Theory*, 63(7):4709–4724, 2017.
- Yurii Nesterov. How to make the gradients small. *Optima*, 88:10–11, 2012.
- Coralie Cartis, Nicholas IM Gould, and Philippe L Toint. *How Much Patience to You Have?: A Worst-case Perspective on Smooth Nonconvex Optimization*. Science and Technology Facilities Council Swindon, 2012.
- Coralie Cartis, Nick IM Gould, and Philippe L Toint. Worst-case evaluation complexity and optimality of second-order methods for nonconvex smooth optimization. *arXiv preprint arXiv:1709.07180*, 2017.
- Yair Carmon, John C. Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points I. *CoRR*, abs/1710.11606, 2017a.
- Yair Carmon, John C Duchi, Oliver Hinder, and Aaron Sidford. Lower bounds for finding stationary points II: First-order methods. *arXiv preprint arXiv:1711.00841*, 2017b.

- Olivier Devolder, François Glineur, Yurii Nesterov, et al. First-order methods with inexact oracle: the strongly convex case. *CORE Discussion Papers*, 2013016, 2013.
- Olivier Devolder, François Glineur, and Yurii Nesterov. First-order methods of smooth convex optimization with inexact oracle. *Mathematical Programming*, 146(1-2):37–75, 2014.
- Ohad Shamir. On the complexity of bandit and derivative-free stochastic convex optimization. In *COLT 2013 - The 26th Annual Conference on Learning Theory, June 12-14, 2013, Princeton University, NJ, USA*, pages 3–24, 2013.
- John C. Duchi, Michael I. Jordan, Martin J. Wainwright, and Andre Wibisono. Optimal rates for zero-order convex optimization: The power of two function evaluations. *IEEE Trans. Information Theory*, 61(5):2788–2806, 2015.
- Sergey G Bobkov. On concentration of distributions of random weighted sums. *Annals of probability*, pages 195–215, 2003.
- Milla Anttila, Keith Ball, and Irini Perissinaki. The central limit problem for convex bodies. *Transactions of the American Mathematical Society*, 355(12):4723–4735, 2003.
- Heinrich von Weizsäcker. Sudakov’s typical marginals, random linear functionals and a conditional central limit theorem. *Probability theory and related fields*, 107(3):313–324, 1997.
- Sanjoy Dasgupta, Daniel Hsu, and Nakul Verma. A concentration theorem for projections. In *Proceedings of the Twenty-Second Conference on Uncertainty in Artificial Intelligence*, pages 114–121. AUAI Press, 2006.
- Beatrice Laurent and Pascal Massart. Adaptive estimation of a quadratic functional by model selection. *Annals of Statistics*, pages 1302–1338, 2000.