Agnostically Learning Multi-index Models with Queries

1st Ilias Diakonikolas
Computer Sciences
UW Madison
Madison, USA
ilias@cs.wisc.edu

2nd Daniel M. Kane
Computer Science
UCSD
San-Diego, USA
dakane@ucsd.edu

3rd Vasilis Kontonis
Computer Science
UT Austin
Austin, USA
vasilis@cs.utexas.edu

4th Christos Tzamos
Computer Science
UW Madison & University of Athens
Madison, USA
tzamos@wisc.edu

5th Nikos Zarifis
Computer Sciences
UW Madison
Madison, USA
zarifis@wisc.edu

2024 IEEE 65th Annual Symposium on Foundations of Computer Science (FOCS) | 979-8-3315-1674-1/24/$31.00 ©2024 IEEE | DOI: 10.1109/FOCS61266.2024.00116

***Abstract*—We study the power of query access for the fundamental task of agnostic learning under the Gaussian distribution. In the agnostic model, no assumptions are made on the labels of the examples and the goal is to compute a hypothesis that is competitive with the *best-fit* function in a known class, i.e., it achieves error $\mathrm{opt}+\epsilon$, where opt is the error of the best function in the class. We focus on a general family of Multi-Index Models (MIMs), which are d-variate functions that depend only on few relevant directions, i.e., have the form $g(\mathbf{W}\mathbf{x})$ for an unknown link function g and a $k\times d$ matrix $\mathbf{W}$. Multi-index models cover a wide range of commonly studied function classes, including real-valued function classes such as constant-depth neural networks with ReLU activations, and Boolean concept classes such as intersections of halfspaces.**

Our main result shows that query access gives significant runtime improvements over random examples for agnostically learning both real-valued and Boolean-valued MIMs. Under standard regularity assumptions for the link function (namely, bounded variation or surface area), we give an agnostic query learner for MIMs with running time $O(k)^{\mathrm{poly}(1/\epsilon)}\ \mathrm{poly}(d)$. In contrast, algorithms that rely only on random labeled examples inherently require $d^{\mathrm{poly}(1/\epsilon)}$ samples and runtime, even for the basic problem of agnostically learning a single ReLU or a halfspace. As special cases of our general approach, we obtain the following results:

- **For the class of depth-ℓ, width-S ReLU networks on $\mathbb{R}^d$, our agnostic query learner runs in time $\mathrm{poly}(d)2^{\mathrm{poly}(\ell S/\epsilon)}$. This bound qualitatively matches the runtime of an algorithm by [1] for the realizable PAC setting with random examples.**
- **For the class of arbitrary intersections of k halfspaces on $\mathbb{R}^d$, our agnostic query learner runs in time $\mathrm{poly}(d)\,2^{\mathrm{poly}(\log(k)/\epsilon)}$. Prior to our work, no improvement over the agnostic PAC model complexity (without queries) was known, even for the case of a single halfspace.**

In both these settings, we provide evidence that the $2^{\mathrm{poly}(1/\epsilon)}$ runtime dependence is required for proper query learners, even for agnostically learning a single ReLU or halfspace.

Our algorithmic result establishes a strong computational separation between the agnostic PAC and the agnostic PAC+Query models under the Gaussian distribution for a range of natural function classes. Prior to our work, no such separation was known for any natural concept class — even for the case of a single halfspace, for which it was an open problem posed by Feldman [2]. Our results are enabled by a general dimension-reduction technique that leverages query access to estimate gradients of (a smoothed version of) the underlying label function.

***Index Terms*—Agnostic Noise, Multi-index models, queries**

I. Introduction

a) PAC Learning with Queries: In Valiant's PAC learning model [3], [4], the learner is given access to random examples labeled according to an unknown function in a known concept class. The goal of the learner is to compute a hypothesis that is close to the target function with respect to a specified loss function[1]. The standard PAC learning model is "passive" in that the learning algorithm has no control over the selection of the training set. Interestingly, while this has become known as *the* PAC model, Valiant's landmark paper [4] allowed queries (in addition to random samples), i.e., black-box access to the target function. We will refer to this as PAC+Query model.

A *query oracle*[2] allows the learner to obtain the value of the target function on any desired point in the domain. PAC learning with access to a query oracle can be viewed as an "active" learning model, intuitively capturing the ability to perform experiments or the availability of expert advice. A long line of research in computational learning theory has explored the power of queries in the context of PAC learning. This line of investigation has spanned the distribution-free versus distribution-specific settings and the realizable (i.e., clean label) setting versus the agnostic (i.e., adversarial label noise) setting; see, e.g., [5]–[8] for some classical early works and [9], [10] for some more recent results in this broad area. A conceptual message of this line of work is that, in the realizable setting, access to queries can be stronger than random samples (from a computational standpoint) for a range of natural concept classes.

In addition to being a fundamental open question in learning theory, the general problem of understanding the effect of query access in the *computational complexity* of learning has received renewed attention over the past decade in the context of deep neural networks. A recent line of inquiry from the machine learning security community has studied *model extraction attacks* — see, e.g., [11]–[17] and references therein — where black-box query access to publicly deployed networks may allow efficient reconstruction of the hidden model

[1]For Boolean functions, one typically uses the 0-1 loss, while for real-valued functions a typical choice is the L_2 loss.

[2]In the special case of learning Boolean-valued functions, these are known as "membership" queries, as the answer to a query determines membership in the set of satisfying assignments of the target concept.

2575-8454/24/$31.00 ©2024 IEEE
DOI 10.1109/FOCS61266.2024.00116

– thus exposing potential vulnerability of the deployed models. These practical applications served as a motivation for the design of the first computationally efficient learners for simple neural networks using query access to the target function [18], [19]. Importantly, the latter algorithmic results apply in the realizable PAC model under the Gaussian distribution.

b) Multi-index Function Models (MIMs): A common (semi)-parametric modeling assumption in high-dimensional statistics is that the target function depends only on a few relevant directions. Specifically, multi-index models [20]–[25] prescribe that the target function is of the form $f(\mathbf{x}) = g(\mathbf{W}\mathbf{x})$ for a link function $g : \mathbb{R}^k \mapsto \mathbb{R}$ and a $k \times d$ weight matrix $\mathbf{W}$. In most settings, the link function g is assumed to be unknown and satisfies certain smoothness properties. Single-index models are the special case where the target function depends only on a single hidden-direction $\mathbf{w}$, i.e., $f(\mathbf{x}) = g(\mathbf{w} \cdot \mathbf{x})$ for some $g : \mathbb{R} \mapsto \mathbb{R}$ and $\mathbf{w} \in \mathbb{R}^d$ [26]–[29]. Standard examples of single-index models include one-bit compressed sensing [30]–[32] where $g(t) = \mathrm{sign}(t)$; and phase retrieval [33], [34], where $g(t) = |t|^2$.

Multi-index models capture a wide range of parametric models studied in the statistics and computer science literatures, including neural networks and classes of geometric Boolean functions. The fundamental class of halfspace intersections was studied in [35] over the Gaussian distribution. Subsequent work [36]–[40] gave improved algorithms and bounds for more general MIM classes.

More recently, an extensive line of work [41]–[52] has studied the efficient learnability of (natural classes of) MIMs from random examples under well-behaved marginal distributions — most notably under the Gaussian distribution on examples. The aforementioned works exclusively focus on the PAC model with random samples and the underlying algorithms succeed in the realizable setting (or in the presence of additive Gaussian label noise).

c) This Work: Agnostically Learning Multi-index Models with Queries: Here we study the power of queries in the *agnostic* PAC model [53], [54] for a wide class of multi-index models. In the agnostic model, no assumptions are made on the labels of the examples and the goal is to compute a hypothesis that is competitive with the *best-fit* function in a known class. This is a notoriously challenging model of learning with very few positive results in the distribution-free setting. For example, it is known that even weak distribution-free agnostic learning (i.e., outputting a hypothesis with non-trivial advantage over random) is computationally hard for very simple classes of single-index models with known link functions. These include linear threshold gates and single neurons with ReLU activations [55]–[58][3].

In this work, we focus on the general problem of agnostically learning multi-index models under the standard Gaussian distribution using queries. At a high-level, our results also encompass the challenging setting where the link function is *unknown* and only require an average smoothness condition on the target function. Classes covered by our framework include real-valued function classes such as constant-depth neural networks with ReLU activations and Boolean concept classes such as intersections of halfspaces. In summary, we are interested in the following question:

Question I.1. *Does* query access *affect the complexity of distribution-specific agnostic learning of multi-index models? In particular, does the availability of queries allow for* qualitatively *more efficient algorithms, compared to the vanilla random example setting?*

The main contribution of this paper is a simple and general methodology that answers this question in the affirmative for a broad family of multi-index function models (including all the aforementioned examples).

A special case of Question I.1 was explicitly asked — in the Boolean setting — for the class of Linear Threshold Functions by Feldman [2] and by Gopalan, Kalai, and Klivans [59] As a corollary of our approach, we answer this open question. Specifically, we provide a new query algorithm for agnostically learning halfspaces implying a super-polynomial separation between the two learning models (learning with random samples versus with queries), subject to standard cryptographic assumptions. In the following subsection, we describe our contributions in detail.

A. Our Results

a) Problem Definition: Before we formally state our main results, we define the agnostic learning model with queries. For concreteness, Definition I.2 concerns real-valued functions, where the accuracy is measured with respect to the L_2 loss. The definition for Boolean-valued concepts is essentially identical, where the L_2 loss is replaced by the 0-1 loss.

Definition I.2 (Agnostically Learning Real-valued Functions with Queries)**.** *Fix $\epsilon \in (0,1)$ and a class $\mathcal{C}$ of real-valued functions on $\mathbb{R}^d$. The adversary picks a label function $y(\mathbf{x}) \in \mathbb{R}$ for every $\mathbf{x} \in \mathbb{R}^d$. The learner is allowed to either draw $\mathbf{x} \sim \mathcal{N}$ (sample access) or select any desired point $\mathbf{x} \in \mathbb{R}^d$ (query access) and obtain the value $y(\mathbf{x})$. Let $N_s \in \mathbb{Z}_+$ be the number of samples and $N_q \in \mathbb{Z}_+$ the number of queries used by the learner. The goal of the learner is to output a hypothesis $h : \mathbb{R}^d \to \mathbb{R}$ that, with high probability, has excess L_2^2 error at most ϵ (with respect to $\mathcal{C}$), i.e., it satisfies $\mathcal{E}_2(h, \mathcal{C}; y) \coloneqq \mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[(h(\mathbf{x}) - y(\mathbf{x}))^2] - \inf_{c\in\mathcal{C}} \mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[(c(\mathbf{x}) - y(\mathbf{x}))^2] \leq \epsilon$.*

Remark I.3 (Boolean-valued Functions)**.** In the boolean-valued setting, we focus on learning with respect to the 0-1 loss. That is, the goal of the learner is to output a hypothesis $h : \mathbb{R}^d \mapsto \{\pm 1\}$ with excess 0-1 error at most ϵ, i.e., $\mathcal{E}_{0/1}(h, \mathcal{C}; y) \coloneqq \mathbf{Pr}_{\mathbf{x}\sim\mathcal{N}}[h(\mathbf{x}) \neq y(\mathbf{x})] - \inf_{c\in\mathcal{C}} \mathbf{Pr}_{\mathbf{x}\sim\mathcal{N}}[c(\mathbf{x}) \neq y(\mathbf{x})] \leq \epsilon$.

1) Agnostically Learning Real-valued Multi-index Models: We start by describing the family of multi-index models

[3]We note that these (distribution-free) hardness results hold even with query access, as follows from [2].

for which our results are applicable. Roughly speaking, our algorithmic approach can be used to agnostically learn any family of multi-index models $\mathcal{C}$ such that any function in $\mathcal{C}$ has "bounded variation", in the sense that the L_2-norm of its gradient is bounded with respect to the standard normal. We remark that similar "smoothness" assumptions, i.e., that f belongs in a Sobolev space, are standard (and necessary) in non-parametric and semi-parametric regression [60]. Under this assumption, we show that there exists an *efficient dimension-reduction* scheme that yields a "fixed parameter tractable" agnostic learner significantly improving over the best known algorithmic results in the agnostic PAC setting with random examples.

We are now ready to formally define the semi-parametric class of MIMs that we consider in this work. In the following definition, we require that the target function is bounded in L_4-norm (with respect to the standard normal distribution) and also that the norm of its gradient is bounded in L_2-norm.

Definition I.4 (Bounded Variation Multi-index Models)**.** *Fix $L, M > 0$ and $k \in \mathbb{Z}_+$. We define the class $\mathfrak{R}(M, L, k)$ of continuous, (almost everywhere) differentiable real-valued functions such that for every $f \in \mathfrak{R}(M, L, k)$:*

1) *It holds $(\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[f^4(\mathbf{x})])^{1/2} \leq M$ and $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[\|\nabla f(\mathbf{x})\|_2^2] \leq L$.*
2) *There exists a subspace U of $\mathbb{R}^d$ of dimension at most k such that f depends only on U, i.e., for every $\mathbf{x} \in \mathbb{R}^d$ it holds that $f(\mathbf{x}) = f(\mathrm{proj}_U \mathbf{x})$, where $\mathrm{proj}_U \mathbf{x}$ is the projection of $\mathbf{x}$ on U.*

We will subsequently see that this is a very broad class of functions subsuming commonly studied classes such as multi-layer neural networks with ReLUs and other activations.

Our main result is an efficient algorithm that exploits the power of queries to significantly reduce the runtime of agnostically learning the semi-parametric class of Definition I.4.

Theorem I.5 (Agnostic Query Learner for Real-valued Multi-index Models)**.** *Fix the function class $\mathfrak{R}(M, L, k)$ given in Definition I.4. There exists an algorithm that makes $N_q = \mathrm{poly}(dML/\epsilon)$ queries, draws $N_s = \mathrm{poly}(dML/\epsilon) + k^{\mathrm{poly}(L,M,1/\epsilon)}$ random labeled examples, runs in time $\mathrm{poly}(N_s, N_q, d)$, and outputs a polynomial $h : \mathbb{R}^d \mapsto \mathbb{R}$ such that with high probability h has L_2^2-excess error $\mathcal{E}_2(h, \mathfrak{R}(L, M, k); y) \leq \epsilon$.*

a) Comparison with Sample-Based Algorithms: As a corollary of Theorem I.5, we establish a strong separation between the agnostic PAC+Query model and the agnostic PAC model (with random samples only). We first compare with the best-known algorithm for agnostically PAC learning real-valued functions, which is the L_2-polynomial regression algorithm. To agnostically learn the class of Definition I.4 to excess error ϵ, one needs polynomials of degree $\mathrm{poly}(L, M, 1/\epsilon)$, and thus $d^{\mathrm{poly}(L,M,1/\epsilon)}$ samples and time are necessary. Theorem I.5 leverages the power of queries to efficiently reduce the dimensionality of the problem, and thus qualitatively improve the computational complexity of agnostic learning to $\mathrm{poly}(d)\, k^{\mathrm{poly}(L,M,1/\epsilon)}$.

Given the assumption of Definition I.4 that the target function depends only on an unknown k-dimensional subspace, it is natural to attempt some kind of dimension-reduction technique in order to reduce the sample and computational complexity of learning. Such reductions are indeed often possible *in the realizable setting* by using some form of PCA and then working in the obtained low-dimensional subspace; see, e.g., [61].

On the other hand, in the agnostic setting considered here, there is strong evidence that such dimension-reduction schemes, or any other runtime improvements whatsoever, are impossible using only sample access to the target function. Specifically, a recent line of work (see, e.g., [62], [63]) has shown that for agnostically learning real-valued MIMs (even very special cases thereof), the standard L_2-regression algorithm is qualitatively optimal computationally (e.g., under standard cryptographic assumptions) in the standard agnostic PAC model. This in particular implies that the best possible runtime without query access is $d^{\mathrm{poly}(1/\epsilon)}$. In fact, even for learning a single ReLU activation, which satisfies Definition I.4 with $L, M = O(1)$ and $k = 1$, $d^{\mathrm{poly}(1/\epsilon)}$ samples and time are required [62], [63]. In contrast, Theorem I.5 decouples the dimension dependence from the dependence on $1/\epsilon$ and yields an algorithm with runtime $\mathrm{poly}(d)\, 2^{\mathrm{poly}(1/\epsilon)}$.

b) Concrete Applications: Theorem I.5 applies to a fairly general non-parametric class of functions. Here we provide specific applications to well-studied classes of neural networks.

Single Non-Linear Gates. The simplest case is that of agnostically learning a ReLU, i.e., a function of the form $f(\mathbf{x}) = \mathrm{ReLU}(\mathbf{w} \cdot \mathbf{x})$, where $\mathbf{w} \in \mathbb{R}^d$ and $\mathrm{ReLU}(t) = \max\{0, t\}$. In the vanilla agnostic PAC setting, the complexity of this problem is $d^{\mathrm{poly}(1/\epsilon)}$(both upper and lower bounds). On the positive side, the L_2-polynomial regression algorithm has sample and computational complexity $d^{\Theta(\mathrm{poly}(1/\epsilon))}$. On the negative side, there is strong evidence that this complexity upper bound is qualitatively best possible, both for SQ algorithms [62], [64], [65] and under plausible cryptographic assumptions [63]. Our agnostic query learner has complexity $\mathrm{poly}(d)\, 2^{\mathrm{poly}(1/\epsilon)}$, implying a super-polynomial separation between the two learning models.

Corollary I.6 (Agnostic Query Learning for ReLUs)**.** *There exists an agnostic query learner for the class of ReLUs on $\mathbb{R}^d$ with running time $\mathrm{poly}(d)\, 2^{\mathrm{poly}(1/\epsilon)}$.*

Corollary I.6 follows from Theorem I.5 by observing that ReLUs satisfy Definition I.4 for $k = 1$ and $L, M = O(1)$ (assuming that the norm of the weight vector is bounded, i.e., $\|\mathbf{w}\|_2 = O(1)$).

Note that selecting the excess error to be $\epsilon = 1/\log^c(d)$, where $c > 0$ is a small constant, the query algorithm of Corollary I.6 has $\mathrm{poly}(d)$ runtime. On the other hand, the complexity of agnostic learning problem with random samples is super-polynomial in d for *any* $\epsilon = o_d(1)$.

Finally, we note that Corollary I.6 holds for other link functions satisfying smoothness assumptions, e.g., sigmoidal activations of the form $t \mapsto 1/(1+\exp(-t))$.

Single-index Models. Our first application above assumed that the link function is known a priori.We next consider learning Single-index models (SIMs) with an *unknown* Lipschitz link function $g : \mathbb{R} \mapsto \mathbb{R}$, i.e., $f(\mathbf{x}) = g(\mathbf{w} \cdot \mathbf{x})$. Classical results [66], [67] gave efficient algorithms for this setting in the realizable PAC setting (or with unbiased additive noise) under the additional assumption that g is non-decreasing. The agnostic setting was recently considered in [68] who gave an efficient algorithm achieving error $O(\sqrt{\mathrm{opt}}) + \epsilon$ for distributions with bounded second moments (similarly assuming weight vectors of bounded ℓ_2-norm). Using Theorem I.5, we can leverage query access to provide optimal agnostic guarantees with essentially the same complexity as for the case of known link function.

Corollary I.7 (Agnostic Query Learning for Lipschitz SIMs)**.** *There exists an agnostic query learner for the class of L-Lipschitz SIMs on $\mathbb{R}^d$, for $L = O(1)$, with running time* $\mathrm{poly}(d)\, 2^{\mathrm{poly}(1/\epsilon)}$.

One-Hidden Layer ReLU Networks. Our approach naturally extends to non-negative linear combinations (aka sums) of ReLUs, i.e., functions of the form $f(\mathbf{x}) = \sum_{i=1}^{k} \alpha^{(i)} \mathrm{ReLU}(\mathbf{w}^{(i)} \cdot \mathbf{x})$ for k non-negative weights $\alpha^{(i)} \geq 0$ and weight vectors $\mathbf{w}^{(i)} \in \mathbb{R}^d$. Prior work [41], [42], [46], [69] has studied this problem in the noiseless setting with random samples under the Gaussian distribution — with the best-known runtime being $\mathrm{poly}(d/\epsilon)\,(k/\epsilon)^{O(\log^2 k)}$ [69]. Using Theorem I.5, we obtain an agnostic query learner with complexity $\mathrm{poly}(d) O(k)^{\mathrm{poly}(1/\epsilon)}$. To see this, we note that as long as $\mathbf{E}[f^2(\mathbf{x})] = O(1)$ we also obtain that $\mathbf{E}[\|\nabla f(\mathbf{x})\|_2^2] = O(1)$ which implies only an $O(k)^{\mathrm{poly}(1/\epsilon)}$ runtime overhead.

Our approach can also be applied to the more general class of (unconstrained) linear combinations of k ReLUs, i.e., functions of the form $f(\mathbf{x}) = \sum_{i=1}^{k} \alpha^{(i)} \mathrm{ReLU}(\mathbf{w}^{(i)} \cdot \mathbf{x})$. This is known [46], [50]–[52] to be a more challenging class of functions to learn. In the noiseless setting, the best known runtime for general linear combinations is $(dk/\epsilon)^{O(k)}$ [52]. Using Theorem I.5, we obtain an agnostic query learner with complexity $\mathrm{poly}(d)\, 2^{\mathrm{poly}(k/\epsilon)}$.

Corollary I.8 (Agnostic Query Learning for 1-Hidden Layer ReLU Networks)**.** *There exists an agnostic query learner for sums of k ReLUs on $\mathbb{R}^d$ with running time* $\mathrm{poly}(d)\, O(k)^{\mathrm{poly}(1/\epsilon)}$. *For general linear combinations of ReLUs, the runtime is* $\mathrm{poly}(d)\, 2^{\mathrm{poly}(k/\epsilon)}$.

Bounded Depth Neural Networks. Our non-parametric function class of Definition I.4 includes deep ReLU networks with ℓ layers of width at most S. More precisely, we assume that $f(\mathbf{x}) = \mathbf{W}_L \mathrm{ReLU}(\mathbf{W}_{L-1} \cdots \mathrm{ReLU}(\mathbf{W}_1 \mathbf{x}))$, for matrices $\mathbf{W}_1 \in \mathbb{R}^{k_1 \times d}, \ldots, \mathbf{W}_\ell \in \mathbb{R}^{k_\ell \times 1}$, with $\|\mathbf{W}_i\|_{op} \leq O(1)$ and $k_i \leq S$; see Definition VI.23 for more details. The running time of our algorithm for this class is $\mathrm{poly}(d) 2^{\mathrm{poly}(\ell S/\epsilon)}$; see Theorem VI.24. We remark that a similar fixed-parameter tractability result for deep ReLU networks was recently shown in [1] for the realizable PAC setting (with access to random examples only). Our result exploits the power of queries to provide a learner with qualitatively similar running time in the much more challenging agnostic setting. We remark that the following result can be readily extended to other continuous activation functions, including sigmoids, LeakyReLUs, and combinations thereof.

TABLE I: Learning Real-Valued Functions using Queries: Running time comparisons of the best known PAC algorithms with our PAC+Query technique (Influence PCA).

Function Class	PAC (without queries) L_2 Regression	PAC+Query **Influence PCA (Ours)**
Single ReLU	$d^{\mathrm{poly}(1/\epsilon)}$	$\mathrm{poly}(d)\, 2^{\mathrm{poly}(1/\epsilon)}$
Sum of k ReLUs	$d^{\mathrm{poly}(1/\epsilon)}$	$\mathrm{poly}(d)\, O(k)^{\mathrm{poly}(1/\epsilon)}$
Linear Combinations of k ReLUs	$d^{\mathrm{poly}(k/\epsilon)}$	$\mathrm{poly}(d)\, 2^{\mathrm{poly}(k/\epsilon)}$
Deep Networks with ℓ-Layers, S-width	$d^{\mathrm{poly}(\ell S/\epsilon)}$	$\mathrm{poly}(d)\, 2^{\mathrm{poly}(\ell S/\epsilon)}$
Bounded Variation	$d^{\mathrm{poly}(k,L,M,1/\epsilon)}$	$\mathrm{poly}(d)\, 2^{\mathrm{poly}(k,L,M,1/\epsilon)}$

Corollary I.9 (Agnostic Query Learning for Bounded-Depth Networks)**.** *There exists an agnostic query learner for ℓ-depth, S-width, ReLU networks on $\mathbb{R}^d$ with running time* $\mathrm{poly}(d) 2^{\mathrm{poly}(\ell S/\epsilon)}$.

For a summary of our results for the above classes, we refer to Table I (where for the L_2-regression algorithm we only assume random sample access).

c) Proper versus Improper Learning: The hypothesis computed by the algorithm of Theorem I.5 is not necessarily in the target concept class. That is, the agnostic learner is *improper*. With some additional effort, our approach can be used to obtain *proper* learners. As a concrete example, for the class of ReLUs, we show the following:

Theorem I.10 (Proper Agnostic Query Learner of ReLUs)**.** *There exists an algorithm that makes* $\mathrm{poly}(d/\epsilon)$ *queries, runs in time* $\mathrm{poly}(d)\, 2^{\mathrm{poly}(1/\epsilon)}$*, and properly agnostically learns the class of ReLUs on $\mathbb{R}^d$, i.e., it outputs a ReLU hypothesis $h(\mathbf{x}) = \mathrm{ReLU}(\widehat{\mathbf{w}} \cdot \mathbf{x})$ with excess L_2^2 error at most ϵ with high probability.*

We note that in addition to computing a ReLU hypothesis, the learner of Theorem I.10 uses $\mathrm{poly}(d/\epsilon)$ labeled examples (queries plus random examples), removing the extraneous $2^{\mathrm{poly}(1/\epsilon)}$ term in our generic result.

It is natural to ask whether the $2^{\mathrm{poly}(1/\epsilon)}$ runtime dependence in Theorem I.10 is inherent. We provide evidence that such a dependence may be necessary for *proper* learners. Specifically, we show (Theorem VIII.4) that if there exists a $\mathrm{poly}(d/\epsilon)$ agnostic proper learning for our problem, there exists a polynomial-time algorithm for the small-set expansion (SSE) problem [70] (refuting the SSE hypothesis). This hardness result also extends to the Boolean class of halfspaces.

Obtaining a computational lower bound for improper learners is left as an interesting open problem.

2) Agnostically Learning Boolean Multi-index Models: We start by describing the family of Boolean functions for which our results are applicable. Roughly speaking, our algorithmic approach can be used to agnostically learn any Boolean concept class $\mathcal{C}$ satisfying the following conditions: (i) $\mathcal{C}$ has bounded Gaussian surface area, (ii) it depends on an unknown low-dimensional subspace, and (iii) it is closed under translations. Under these assumptions, we similarly obtain a "fixed parameter tractable" agnostic learner qualitatively improving over the agnostic PAC setting with random examples only.

The Gaussian surface area of a Boolean function is the surface area of its decision boundary weighted by the Gaussian density (Definition I.11). The Gaussian surface area of a concept class has played a significant role as a useful complexity measure in learning theory and related fields; see, e.g., [71]–[75]. A formal definition follows:

Definition I.11 (Gaussian Surface Area)**.** *For a Borel set $A \subseteq \mathbb{R}^d$, its Gaussian surface area is defined by $\Gamma(A) \coloneqq \liminf_{\delta \to 0} \frac{\mathcal{N}(A_\delta \setminus A)}{\delta}$, where $A_\delta = \{x : \mathrm{dist}(x, A) \leq \delta\}$. For a Boolean function $f : \mathbb{R}^d \mapsto \{\pm 1\}$, we overload notation and define its Gaussian surface area to be the surface area of its positive region $K = \{\mathbf{x} \in \mathbb{R}^d : f(\mathbf{x}) = +1\}$, i.e., $\Gamma(f) = \Gamma(K)$. For a class of Boolean concepts $\mathcal{C}$, we define $\Gamma(\mathcal{C}) \coloneqq \sup_{f \in \mathcal{C}} \Gamma(f)$.*

We are ready to define the class of Boolean multi-index models for which our approach applies.

Definition I.12 (Bounded Surface Area, Low-Dimensional Boolean Concepts)**.** *Fix $\Gamma > 0$ and $k \in \mathbb{Z}_+$. We define the class $\mathfrak{B}(\Gamma, k)$ of Boolean concepts with the following properties:*

1) *For every $f \in \mathfrak{B}(\Gamma, k)$, it holds $\Gamma(f_{\mathbf{r}}) \leq \Gamma$ for all $\mathbf{r} \in \mathbb{R}^d$, where $f_{\mathbf{r}}(\mathbf{x}) = f(\mathbf{x} + \mathbf{r})$.*
2) *For every $f \in \mathfrak{B}(\Gamma, k)$, there exists a subspace U of $\mathbb{R}^d$ of dimension at most k such that f depends only on U, i.e., for every $\mathbf{x} \in \mathbb{R}^d$ it holds $f(\mathbf{x}) = f(\mathrm{proj}_U \mathbf{x})$.*

We remark that $\mathfrak{B}(\Gamma, k)$ is a general *non-parametric class* that contains a range of natural and well-studied Boolean function classes. For example, $\mathfrak{B}(\Omega(k), k)$ contains arbitrary functions of k halfspaces.

Our main positive result in this context is a query algorithm that agnostically learns the class $\mathfrak{B}(\Gamma, k)$ with running time $\mathrm{poly}(d) k^{\mathrm{poly}(\Gamma/\epsilon)}$. In more detail, we establish the following theorem:

Theorem I.13 (Agnostic Learner for Boolean Multi-index Models)**.** *Fix the concept class $\mathfrak{B}(\Gamma, k)$ given in Definition I.12. There exists an algorithm that makes $N_q = \mathrm{poly}(d/\epsilon)$ queries, draws $N_s = \mathrm{poly}(d/\epsilon) + O(k)^{\mathrm{poly}(\Gamma/\epsilon)}$ random labeled examples, runs in sample-polynomial time, and outputs a hypothesis $h : \mathbb{R}^d \to \{\pm 1\}$ with excess 0-1 error $\mathcal{E}_{0/1}(h, \mathfrak{B}(\Gamma, k); y) \leq \epsilon$.*

a) Discussion: Some remarks are in order. We start by noting that, in the setting of Theorem I.13, an exponential dependence on the parameter Γ is *information-theoretically* necessary — even with access to queries. Specifically, as shown in [71], there exists a Boolean concept class with Gaussian surface area Γ (consisting of intersections of halfspaces) such that the total number of samples and queries required to obtain constant accuracy is $2^{\Omega(\Gamma)}$.

It is worth comparing Theorem I.13 with the best known algorithmic results in the standard agnostic PAC model (with random samples only). Klivans, O'Donnell and Servedio [71] showed that the L_1-polynomial regression algorithm of [76] agnostically learns any concept class on $\mathbb{R}^d$ whose Gaussian surface area is at most $\Gamma > 0$ with (sample and computational) complexity $d^{\mathrm{poly}(\Gamma/\epsilon)}$. Under the additional assumption that the concepts in the target class depend on an unknown k-dimensional subspace, for some parameter $k \ll d$, Theorem I.13 gives a significantly improved agnostic query algorithm with computational complexity $\mathrm{poly}(d)\, k^{\mathrm{poly}(\Gamma/\epsilon)}$.

For a concrete example, if the target class is the concept class consisting of any intersection of ℓ halfspaces, then we have that $k = \ell$ and $\Gamma = O(\sqrt{\log(\ell)})$ [71]. So, as long as $\ell = O(1)$ or even $\ell = \mathrm{polylog}(d)$, query access allows us to obtain a super-polynomial complexity improvement.

b) Concrete Applications: Theorem I.13 applies to a fairly general non-parametric class of functions. Here we provide specific applications to well-studied classes of Boolean functions.

Halfspaces. Arguably the simplest application is for the class of halfspaces. A halfspace (or Linear Threshold Function) is any Boolean-valued function $f : \mathbb{R}^d \to \{\pm 1\}$ of the form $f(\mathbf{x}) = \mathrm{sign}\,(\mathbf{w} \cdot \mathbf{x} - \theta)$, where $\mathbf{w} \in \mathbb{R}^d$ is the weight vector and $\theta \in \mathbb{R}$ is the threshold. (The function $\mathrm{sign} : \mathbb{R} \to \{\pm 1\}$ is defined as $\mathrm{sign}(t) = 1$ if $t \geq 0$ and $\mathrm{sign}(t) = -1$ otherwise.) The problem of PAC learning halfspaces is a textbook problem in machine learning, whose history goes back to Rosenblatt's Perceptron algorithm [77]. As a corollary of Theorem I.13, we obtain the following:

Corollary I.14 (Agnostic Query Learning of Halfspaces)**.** *There exists an agnostic query learner for the class of halfspaces on $\mathbb{R}^d$ with running time $\mathrm{poly}(d)\, 2^{\mathrm{poly}(1/\epsilon)}$.*

Corollary I.14 follows from Theorem I.13 by observing that halfspaces satisfy Definition I.12 for $k = 1$ and $\Gamma \leq 1/\sqrt{2\pi}$.

As mentioned in the introduction, Corollary I.14 answers an open question independently posed by Feldman [2] and by Gopalan, Kalai, and Klivans [59]. Specifically, as we explain below, it implies a *super-polynomial* computational separation between agnostic query learning and agnostic learning with random samples for the class of halfspaces.

In the vanilla agnostic PAC setting, the complexity of this problem is $d^{\mathrm{poly}(1/\epsilon)}$; the upper bound follows via the L_1-polynomial regression algorithm [76] which has complexity $d^{\Theta(1/\epsilon^2)}$ [78] in this setting. The matching lower bound follows from a recent line of work, both in the SQ model [62],

[64], [65] and under plausible cryptographic assumptions [58], [63].

Functions of Halfspaces. A more general concept class where our general approach is applicable is that consisting of all intersections (or arbitrary functions) of a bounded number of halfspaces. For the special case of intersections, we show:

Corollary I.15 (Agnostic Query Learning for Intersections of Halfspaces)**.** *There exists an agnostic query learner for intersections of ℓ halfspaces on $\mathbb{R}^d$ with running time* $\mathrm{poly}(d)\, O(\ell)^{\mathrm{poly}(\log(\ell)/\epsilon)}$.

Corollary I.15 follows from Theorem I.13 by observing that intersections of ℓ halfspaces satisfy Definition I.12 for $k = \ell$ and that their Gaussian surface area is bounded above by $\Gamma = O(\sqrt{\log(\ell)})$, as shown by Nazarov (see, e.g., [71], [79]).

Analogously to the case of a single halfspace, the complexity of the agnostic learning problem with random samples is significantly worse (as long as $\ell \ll d$), namely $d^{\mathrm{poly}(\log(\ell)/\epsilon)}$; the upper bound follows from [71] and a qualitatively matching SQ lower bound was given in [62], [80].

Finally, for arbitrary functions of ℓ halfspaces, the Gaussian surface area is bounded by $\Gamma = O(\ell)$, leading to the following corollary:

Corollary I.16 (Agnostic Query Learning for Functions of Halfspaces)**.** *There exists an agnostic query learner for arbitrary functions of ℓ halfspaces on $\mathbb{R}^d$ with running time* $\mathrm{poly}(d)\, O(\ell)^{\mathrm{poly}(\ell/\epsilon)}$.

Similarly, the best known complexity upper bound with random samples is $d^{\mathrm{poly}(\ell/\epsilon)}$.

Low-degree Polynomial Threshold Functions (PTFs). Another notable application is for the class of low-degree PTFs that depend on a low-dimensional subspace. A degree-ℓ PTF is any Boolean function $f : \mathbb{R}^d \to \{\pm 1\}$ of the form $h(\mathbf{x}) = \mathrm{sign}\,(p(\mathbf{x}))$, where $p : \mathbb{R}^d \to \mathbb{R}$ is a degree at most ℓ polynomial. Low-degree PTFs have been extensively studied in theoretical machine learning and specifically in the context of agnostic learning [72], [81]–[83].

Here we consider a natural subclass of low-degree PTFs where the underlying polynomial is a subspace junta. Specifically, we consider the class of Boolean functions of the form $f(\mathbf{x}) = \mathrm{sign}\,(p(\mathrm{proj}_U \mathbf{x}))$, where U is an unknown k-dimensional subspace and p is a degree-ℓ polynomial in k variables. Since the Gaussian surface area of this class of functions is bounded above by $\Gamma = O(\ell)$ [72], we obtain the following corollary:

Corollary I.17 (Agnostic Query Learning for Low-Dimensional PTFs)**.** *There exists an agnostic query learner for degree-ℓ PTFs on $\mathbb{R}^d$ that depend on an unknown k-dimensional subspace with running time* $\mathrm{poly}(d)\, O(k)^{\mathrm{poly}(\ell/\epsilon)}$.

The above running time bound should be compared with the best known complexity bound of $d^{\mathrm{poly}(\ell/\epsilon)}$ for agnostic learning with samples [72].

TABLE II: Learning Boolean Concepts using Queries: Running time comparisons of the best known agnostic learners (using random samples) with our Influence PCA technique (using queries).

Concept Class	PAC (without queries) L_1 Regression [71]	PAC+Query **Influence PCA (Ours)**
Single Halfspace	$d^{\mathrm{poly}(1/\epsilon)}$	$\mathrm{poly}(d)\, 2^{\mathrm{poly}(1/\epsilon)}$
Intersections of k Halfspaces	$d^{\mathrm{poly}(\log(k)/\epsilon)}$	$\mathrm{poly}(d)\, 2^{\mathrm{poly}(\log(k)/\epsilon)}$
Functions of k Halfspaces	$d^{\mathrm{poly}(k/\epsilon)}$	$\mathrm{poly}(d)\, 2^{\mathrm{poly}(k/\epsilon)}$
Degree-ℓ, k-Dim. PTFs[4]	$d^{\mathrm{poly}(\ell/\epsilon)}$	$\mathrm{poly}(d)\, O(k)^{\mathrm{poly}(\ell/\epsilon)}$
Low-Dim. Geometric Concepts	$d^{\mathrm{poly}(\Gamma/\epsilon)}$	$\mathrm{poly}(d)\, O(k)^{\mathrm{poly}(\Gamma/\epsilon)}$

Table II summarizes our contributions for Boolean concept classes in comparison to prior work on agnostic PAC learning (with random samples only).

II. Technical Overview

We leverage query access to develop a unified dimension-reduction framework for agnostically learning both real-valued and Boolean-valued multi-index models. As already explained after the statement of Theorem I.5, natural dimension-reduction approaches that work in the realizable (noiseless) setting inherently cannot be extended to the agnostic setting.

At a high-level, our framework reduces the problem of agnostically learning MIMs in d dimensions to agnostically learning the same class in $\mathrm{poly}(k/\epsilon)$ dimensions. It consists of three main steps:

- First we use queries to the label function to simulate gradient queries to a "smoothed" version $\widetilde{y}(\mathbf{x})$ of the adversarial label $y(\mathbf{x})$. We show that, as long as the concept class of interest has bounded variation (real-valued MIMs of Definition I.4) or bounded Gaussian surface area (Boolean MIMs of Definition I.12), a hypothesis that has low excess-error with respect to the smoothed label $\widetilde{y}$ will also have low excess error with respect to the original label $y(\mathbf{x})$; see Proposition II.1.
- The second step uses gradient queries to the function $\widetilde{y}$ in order to compute an accurate estimate of the influence matrix of the "smoothed" label, namely $\mathbf{M} = \mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[\nabla\widetilde{y}(\mathbf{x})(\nabla\widetilde{y}(\mathbf{x}))^\top]$. We perform PCA on $\mathbf{M}$ and find the top eigenvectors (i.e., the eigen-directions whose corresponding eigenvalues are larger than some threshold). This method is known as outer gradient product [24]; in the context of learning/testing Boolean concepts, it has been used in [75], [84]. (See Section III for a detailed summary of related work.) We show that those "high-influence" directions form a low-dimensional (i.e., of dimension $\mathrm{poly}(k/\epsilon)$) subspace such that there exists a hypothesis that (i) depends only on the low-dimensional subspace, (ii) has bounded surface area/variation, and (iii) is close to our target function. That is, we effectively reduce the dimension of our original learning task from d down to $\mathrm{poly}(k/\epsilon)$.

[4]The surface area bound was proved in [72].

- The third step is to solve an agnostic learning task of a bounded variation/surface area function in the low-dimensional subspace spanned by the top eigenvectors of $\mathbf{M}$. For this step, for learning real-valued MIMs, we rely on a generic L_2-regression algorithm; for learning Boolean concepts, we use the L_1-polynomial regression agnostic learner of [71], [76]. Those methods yield non-proper learning algorithms – to obtain proper-learners, we essentially perform a brute-force search over a net of the *low-dimensional* parameter space found in the previous step.

A. From Zero- to First-Order: Gradient Queries via Oracle Queries

Intuitively, having access to queries, for some example $\mathbf{x}$, we can ask for the values of $y(\mathbf{x})$ in a "small" neighborhood around $\mathbf{x}$ and therefore estimate the gradient $\nabla_{\mathbf{x}} y(\mathbf{x})$. The first issue that we have to overcome is that the observed label $y(\mathbf{x})$ is not guaranteed to be a differentiable function (even if the underlying target function is). To circumvent this issue, we employ a strategy similar to the Gaussian convolution technique used in zero-order (gradient-free) optimization [85]. In particular, to estimate the gradient of a function $y(\cdot)$ at $\mathbf{x}$ only having access to a value oracle, the method samples $\mathbf{z}$ from a mean-zero Gaussian with small covariance, i.e., $\mathbf{z} \sim \mathcal{N}(\mathbf{0}, \rho\mathbf{I})$ for some small ρ, and then asks for the value of the function at $\mathbf{x}+\rho\mathbf{z}$. Even if the function $y(\cdot)$ itself is non-smooth, then, by Stein's identity, we have $\mathbf{E}_{\mathbf{z}\sim\mathcal{N}}[\mathbf{z}\ y(\mathbf{x}+\rho\mathbf{z})] \propto \nabla\widetilde{y}(\mathbf{x})$ (see Lemma V.4), where $\widetilde{y}(\mathbf{x})$ is a smoothed version of $y(\mathbf{x})$, specifically $\widetilde{y}(\mathbf{x}) = \mathbf{E}_{\mathbf{z}\sim\mathcal{N}}[y(\mathbf{x}+\rho\mathbf{z})]$. By drawing $N = \mathrm{poly}(d/\epsilon)$ Gaussian samples $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(N)}$, we can empirically estimate the gradient of $\widetilde{y}(\cdot)$ at every desired point $\mathbf{x} \in \mathbb{R}^d$. Therefore, by performing N queries on the points $\mathbf{z}^{(i)}$, we obtain an approximation of the gradient $\nabla\widetilde{y}(\mathbf{x})$ for any $\mathbf{x}$. Even though the above technique yields gradient estimates, it comes with a cost: *to obtain the "smooth" label $\widetilde{y}(\mathbf{x})$, we add noise to the (already corrupted) label $y(\mathbf{x})$. Our plan is to argue that learning using the resulting smoothed labels $\widetilde{y}(\mathbf{x})$ yields a good classifier for the original instance — as long as the "smoothing" parameter ρ is sufficiently small.*

a) Ornstein–Uhlenbeck Smoothing: One could hope that if we add a small amount of noise to $y(\mathbf{x})$, the smooth label $\widetilde{y}(\mathbf{x})$ will be close to $y(\mathbf{x})$ (at least in the L_2-sense). Unfortunately, this is not true (even in one dimension), as $y(\mathbf{x})$ may be an arbitrarily complex function and after smoothing $\widetilde{y}(\mathbf{x})$ may be far from $y(\mathbf{x})$; see Figure 1. To be able to learn from the smoothed instance, we need two properties: (i) the resulting marginal distribution on the examples must be close to the initial $\mathbf{x}$-marginal, and (ii) the smoothing operation must not increase the excess error of the functions in the hypothesis class by a lot. In other words, a hypothesis that performs well with respect to the smoothed label $\widetilde{y}(\mathbf{x})$ should also perform well with respect to the original label $y(\mathbf{x})$. Applying the Gaussian convolution smoothing $\mathbf{x}+\rho\mathbf{z}$ yields a normal distribution that has covariance $(1+\rho)\mathbf{I}$. In order to make this distribution be close to a standard normal (say, in total variation distance), one would need to apply a tiny amount of noise, i.e., ρ should be at most $\mathrm{poly}(1/d)$. To avoid changing the $\mathbf{x}$-marginal of the instance, instead of simply convolving with a Gaussian kernel, we apply the Ornstein–Uhlenbeck noise operator T_ρ that rescales $\mathbf{x}$ and corresponds to the transformation $\widetilde{\mathbf{x}} = \sqrt{1-\rho^2}\mathbf{x}+\rho\mathbf{z}$. We observe that $\widetilde{\mathbf{x}}$ follows a standard normal distribution. The resulting "smoothed" label $\widetilde{y}$ is now defined as $T_\rho y(\mathbf{x}) = \mathbf{E}_{\mathbf{z}\sim\mathcal{N}}[y(\widetilde{\mathbf{x}})]$. Even though the marginal of $\widetilde{\mathbf{x}}$ matches exactly with the initial marginal, we have introduced noise to the instance and we still need to show that this does not significantly affect the performance of the hypotheses in the function class of interest.

We show that, regardless of how complex the label $y(\mathbf{x})$ is, if the function class of interest is "well-behaved" — in the sense that it only contains concepts with bounded variation/Gaussian surface area — the Ornstein–Uhlenbeck noise process will not significantly affect the excess error of a hypothesis h.

Proposition II.1 (Informal – Ornstein–Uhlenbeck Smoothing Preserves the Risk-Minimizer)**.** *Let $y : \mathbb{R}^d \mapsto \mathbb{R}$ and C be a class of functions over $\mathbb{R}^d$ such that for every $f \in C$ it holds $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[\|\nabla f(\mathbf{x})\|_2^2] \leq L$. Let $\widetilde{f} \in C$ be an L_2 risk minimizer with respect to the smoothed label $T_\rho y$ (see Definition V.1), i.e., $\widetilde{f} \in \mathrm{argmin}_{h\in C} \mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[(h(\mathbf{x}) - T_\rho y(\mathbf{x}))^2]$. Then we have that*

$$\mathop{\mathbf{Pr}}_{\mathbf{x}\sim\mathcal{N}}[(\widetilde{f}(\mathbf{x}) - y(\mathbf{x}))^2] \leq \inf_{f\in C} \mathop{\mathbf{Pr}}_{\mathbf{x}\sim\mathcal{N}}[(f(\mathbf{x}) - y(\mathbf{x}))^2] + O(\rho^2 L)\,.$$

At a high-level, the effect of the noise operator T_ρ on the risk minimizer is milder when the function does not change very rapidly. To prove Proposition II.1, we show that the correlation of any hypothesis f with bounded variation is approximately preserved when we replace $y(\mathbf{x})$ with $T_\rho y(\mathbf{x})$. The correlation of f with respect to $T_\rho y(\mathbf{x})$ is $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[f(\mathbf{x})T_\rho y(\mathbf{x})]$. However, since T_ρ is a symmetric linear operator, we can equivalently apply the smoothing T_ρ to f and consider $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[T_\rho f(\mathbf{x})y(\mathbf{x})]$. Since $f(\mathbf{x})$ has bounded variation, we can now show via a result on noise sensitivity for real-valued functions, that $T_\rho f(\mathbf{x})$ is indeed close to $f(\mathbf{x})$ in L_2^2. Therefore, the correlation $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[T_\rho f(\mathbf{x})y(\mathbf{x})]$ is close to $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[f(\mathbf{x})y(\mathbf{x})]$. The fact that $T_\rho f$ and f are close is intuitively clear: the smaller the variation of f, $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[\|\nabla f(\mathbf{x})\|_2^2]$, the smaller the effect of slightly perturbing a point $\mathbf{x}$ will have on the L_2^2, as the L_2^2 distance between $f(\mathbf{x})$ and $f(\sqrt{1-\rho}\mathbf{x}+\rho z)$ is roughly proportional to $\rho^2\|\nabla f(\mathbf{x})\|_2^2$. For more details, we refer to Section V and Proposition V.6.

For learning Boolean concepts, we identify their Gaussian Surface Area to be the crucial complexity measure that determines the effect the smoothing operator T_ρ has on the agnostic learning instance. Similarly to our result for real-valued functions, we reduce preserving the excess error to preserving the correlation of concepts, i.e., ensuring that $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[f(\mathbf{x})T_\rho y(\mathbf{x})] - \mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[f(\mathbf{x})y(\mathbf{x})]$ is small for all concepts of interest f — see Proposition V.10 — and then use a result of Ledoux [86] and Pisier [87] to show that correlations are indeed approximately preserved when the concepts have bounded Gaussian Surface Area; see Proposition V.10.

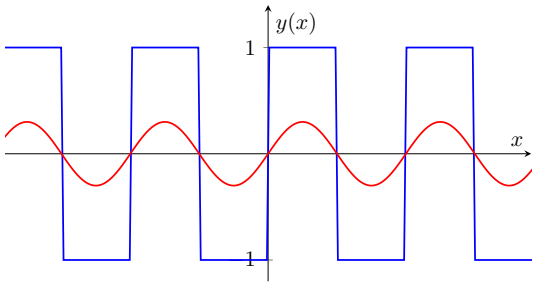

Fig. 1: Smoothing the label $y(\mathbf{x})$. The label $y(\mathbf{x})$ corresponds to the "square wave" (shown in blue). The smoothed version $\widetilde{y}(\mathbf{x})$ is the red curve. We observe that $y(\mathbf{x})$ and $\widetilde{y}(\mathbf{x})$ are far (in the L_2 sense).

B. Learning Bounded Variation Functions via Influence PCA

a) Real-Valued MIMs: Up to this point, we have established that (i) we can leverage query access in order to efficiently simulate gradient queries for the Ornstein–Uhlenbeck smoothed label $T_\rho y$, and (ii) learning from the smoothed label $T_\rho y$ is approximately equivalent to learning from the original label $y(\mathbf{x})$. We will now describe an efficient learner that uses the gradient queries to $T_\rho y$.

Our learner is based on estimating the influence matrix of $T_\rho y$, i.e., $\mathbf{M} = \mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[\nabla T_\rho y(\mathbf{x})(\nabla T_\rho y(\mathbf{x}))^\top]$, using gradient queries. Our main structural result is a general dimension-reduction tool establishing the following: given (an approximation of) the influence matrix of the smooth function $T_\rho y$, we can perform PCA and learn a low-dimensional subspace V so that a bounded variation function that depends only on V can achieve ϵ excess error with respect to $T_\rho y$ in L_2^2. This dimension-reduction step crucially relies on the target concept being low-dimensional (see Definition I.4).

In fact, our dimension-reduction proof for real-valued concepts shows directly that a low-degree polynomial that depends only on the low-dimensional space V exists.

Proposition II.2 (Informal Statement of Proposition VI.10–Dimension Reduction via Influence PCA: Real-Valued Functions)**.** *Let $\widetilde{y}(\mathbf{x}) = T_\rho y(\mathbf{x})$ and let $\mathbf{M} = \mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[\nabla\widetilde{y}(\mathbf{x})(\nabla\widetilde{y}(\mathbf{x}))^\top]$. Moreover, let V be the subspace spanned by all the eigenvectors of $\mathbf{M}$ whose corresponding eigenvalues are at least $\epsilon^2/(kM)$. The following holds:*

- *The dimension of V is at most* $\mathrm{poly}(M, k, 1/\rho, 1/\epsilon)$.
- *There exists a polynomial $q : V \mapsto \mathbb{R}$ of degree $m = O(L/\epsilon^2)$ such that*

$$\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[(q(\mathrm{proj}_V(\mathbf{x})) - \widetilde{y}(\mathbf{x}))^2] \leq \inf_{f\in\mathfrak{R}(M,L,k)} \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[(f(\mathbf{x}) - \widetilde{y}(\mathbf{x}))^2] + \epsilon\,.$$

To prove Proposition II.2, we explicitly construct a low-dimensional polynomial as follows: we first marginalize out the low-influence directions of $\widetilde{y}(\cdot)$, and then we keep its low-degree Hermite approximation.

b) Marginalizing Low-Influence Directions: We first construct a low-dimensional (not necessarily polynomial) version of the noisy label $\widetilde{y}$ that preserves the correlation with the target function $f(\cdot)$. By the assumption of Proposition II.2, all directions in the orthogonal complement $V^\perp$ are low-influence, i.e., for $\mathbf{h} \in V^\perp$ it holds $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[(\mathbf{h}\cdot\nabla\widetilde{y}(\mathbf{x}))^2] \leq O(\epsilon^2/k)$. In words, the function $\widetilde{y}$ is "approximately constant" along some low-influence direction $\mathbf{h}$. Let us first assume that $\widetilde{y}$ is exactly constant on all directions of $V^\perp$. Then, in order to preserve the correlation of $\widetilde{y}$ with f, *we only need to match the expected value of $\widetilde{y}$ over $V^\perp$*. This motivates the following "Gaussian Marginalization Operator" $(\Pi_V g)(\mathbf{x}) := \mathbf{E}_{\mathbf{z}\sim\mathcal{N}}[g(\mathrm{proj}_V\mathbf{x} + \mathrm{proj}_{V^\perp}\mathbf{z})]$ (see Definition VI.5 and Lemma VI.6). So a natural low-dimensional "approximation" of $\widetilde{y}$ is $\Pi_V\widetilde{y}$. Indeed, if $\widetilde{y}$ was constant on $V^\perp$, using the fact that $\mathrm{proj}_V\mathbf{x}$ and $\mathrm{proj}_{V^\perp}\mathbf{x}$ are independent standard Gaussians, we would obtain that

$$\mathop{\mathbf{E}}_{\mathbf{z}\sim\mathcal{N}}[\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}\widetilde{y}(\mathrm{proj}_V(\mathbf{x})+\mathrm{proj}_{V^\perp}(\mathbf{z}))f(\mathbf{x})]] - \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[\widetilde{y}(\mathbf{x})f(\mathbf{x})] = 0\,.$$

Our goal is to show that the Gaussian marginalization $\Pi_V\widetilde{y}$ achieves similar correlation with $\widetilde{y}$ as f, when $\widetilde{y}$ is not constant in $V^\perp$ but "approximately constant", i.e., it has low-influence in directions of $V^\perp$. In Lemma VI.12 we show that when $V^\perp$ contains only low-influence directions, the same is approximately true (up to some additive ϵ error): $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[(\widetilde{y}(\mathbf{x}) - \Pi_V\widetilde{y}(\mathbf{x}))f(\mathbf{x})] \leq O(\epsilon)$. To do this, we first observe that since f depends only on the subspace U, it holds that $\Pi_U f = f$; and since $\Pi_V\widetilde{y}$ depends only on V, we can restrict our attention inside the relevant subspace $W = U + V$. We can thus restrict our attention on W, i.e., $\mathbf{E}_{\mathbf{z}\sim\mathcal{N}_W}[(\widetilde{y}(\mathbf{z}) - \Pi_V\widetilde{y}(\mathbf{z}))f(\mathbf{z})]$, where $\mathcal{N}_W$ is a standard normal on the subspace W. We will show that this correlation difference can be bounded by the variance of $\widetilde{y}$ in the irrelevant directions. Indeed, by the Cauchy-Schwarz inequality, we have

$$\mathop{\mathbf{E}}_{\mathbf{z}\sim\mathcal{N}_W}[(\widetilde{y}(\mathbf{z}) - \Pi_V\widetilde{y}(\mathbf{z}))f(\mathbf{z})] \leq \left(\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}_W}[f^2(\mathbf{x})]\right)^{1/2}\left(\mathop{\mathbf{E}}_{\mathbf{z}\sim\mathcal{N}_W}[(\widetilde{y}(\mathbf{z}) - \Pi_V\widetilde{y}(\mathbf{z}))^2]\right)^{1/2}\,.$$

We next relate the L_2^2 error introduced by the marginalization operation Π_V on $\widetilde{y}$ with the influence matrix $\mathbf{M}$. We use the Gaussian Poincare inequality, which states that for some $g(t) : \mathbb{R} \mapsto \mathbb{R}$ it holds $\mathbf{Var}_{t\sim\mathcal{N}}[g(t)] \leq \mathbf{E}_{t\sim\mathcal{N}}[(g'(t))^2]$. We obtain that for any subspace $R = \mathbf{r}^\perp$ (the orthogonal complement to the direction $\mathbf{r}$) the variance $\mathbf{E}_{\mathbf{z}\sim\mathcal{N}_W}[(\widetilde{y}(\mathbf{z}) - \Pi_R\widetilde{y}(\mathbf{z}))^2]$ is bounded above by $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}_W}[(\nabla\widetilde{y}(\mathbf{x})\cdot\mathbf{r})^2] = \mathbf{r}^\top\mathbf{M}\mathbf{r}$. By repeatedly applying the Gaussian Poincare inequality on a basis of the (at most) k-dimensional subspace $V^\perp \cap W$, we show that

$$\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}_W}[(\widetilde{y}(\mathbf{z}) - \Pi_V\widetilde{y}(\mathbf{z}))^2] \leq Mk \max_{\mathbf{r}\in V^\perp, \|\mathbf{r}\|_2=1} \mathbf{r}^\top\mathbf{M}\mathbf{r} \leq k\; O(\epsilon^2/(kM)) = O(\epsilon^2)\,.$$

In the above bound, we observe that accepting eigenvectors with corresponding eigenvalues at least $\epsilon^2/(Mk)$ ensures that $\Pi_V\widetilde{y}$ achieves at most $O(\epsilon)$ worse correlation with f than $\widetilde{y}$.

c) The Low-Degree Polynomial Approximation: We have established that $\Pi_V \widetilde{y}$ is similar to $\widetilde{y}$ in the sense that it has similar (up to ϵ^2) correlation with the target function $f(\cdot)$. To obtain a polynomial with a similar behavior, we use the low-degree Hermite expansion of $\Pi_V \widetilde{y}$, which we denote by $P_m \Pi_V \widetilde{y}$, where $P_m g$ maps the function g to its m-degree Hermite expansion. We show that in order for $P_m \Pi_V \widetilde{y}$ to achieve low L_2^2 excess error, it suffices to pick the degree m so that $P_m f(\mathbf{x})$ is close to $f(\mathbf{x})$ (in L_2^2). We show that the following bound for the excess error defined as $\mathcal{E}_2(q, f; \widetilde{y}) = \mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[(\widetilde{y}(\mathbf{x}) - q(\mathbf{x}))^2] - \mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[(\widetilde{y}(\mathbf{x}) - f(\mathbf{x}))^2]$. We refer to Lemma VI.11 for the formal statement and proof.

Lemma II.3 (Informal – Excess L_2^2 Error Decomposition)**.** *It holds*

$$\mathcal{E}_2(P_m \Pi_V \widetilde{y}, f; \psi) \leq O(1)\Big(\underbrace{\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[(f(\mathbf{x}) - P_m f(\mathbf{x}))^2]}_{\textit{Polynomial Approximation Error}} + \underbrace{\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[(\widetilde{y}(\mathbf{x}) - \Pi_V \widetilde{y}(\mathbf{x})) f(\mathbf{x})]}_{\textit{Correlation Error}} \Big).$$

Since $f(\mathbf{x})$ has bounded variation (see Definition I.4), we can show using a result from [74] (see Lemma VI.4) that with degree $m = O(L/\epsilon^2)$, it holds that $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[(f(\mathbf{x}) - P_m f(\mathbf{x}))^2] = \epsilon$. Moreover, in the previous paragraph, we have already established that the correlation error is also $O(\epsilon)$.

d) Polynomial Regression in V: So far, we have identified the subspace V and we know that there exists a polynomial that depends on V and achieves low L_2^2 error with the smoothed label $\widetilde{y} = T_\rho y$. Since we have established that the smoothing operation T_ρ does not affect the excess error of a bounded-surface area concept by a lot (see Proposition II.1), we know that the same concept will achieve low excess-error with respect to the original label y. Having established this, for our final step we may directly perform polynomial regression in the low-dimensional subspace V to learn a polynomial with low-excess error. Since the dimension of V is roughly $\mathrm{poly}(Mk/\epsilon)$ and the degree of the polynomial is $\mathrm{poly}(L/\epsilon)$, the total sample and computational complexity of this task is roughly $k^{\mathrm{poly}(L/\epsilon)}$.

e) Boolean MIMs: At a high level, the proof and algorithm for Boolean MIMs is similar to that for real-valued MIMs. We show the following dimension reduction lemma that essentially reduces the initial problem to learning a bounded surface area concept in a $\mathrm{poly}(k/\epsilon)$-dimensional subspace V.

Proposition II.4 (Informal Statement – Dimension-Reduction via Influence PCA: Boolean Concepts)**.** *Let V be the subspace spanned by all the eigenvectors of $\mathbf{M} = \mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[\nabla T_\rho y(\mathbf{x})(\nabla T_\rho y(\mathbf{x}))^\top]$ whose corresponding eigenvalues are at least $\Omega(\epsilon^2/k)$. The following holds:*

- *The dimension of V is at most $\mathrm{poly}(k/(\epsilon\rho))$.*
- *There exists $g : \mathbb{R}^d \to \{\pm 1\}$ with $\Gamma(g) \leq \Gamma$ and $g(\mathbf{x}) = g(\mathrm{proj}_V \mathbf{x})$ for all $\mathbf{x} \in \mathbb{R}^d$ such that*

$$\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[|g(\mathbf{x}) - T_\rho y(\mathbf{x})|] \leq \inf_{f\in\mathfrak{B}(\Gamma,k)} \mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[|f(\mathbf{x}) - T_\rho y(\mathbf{x})|] + \epsilon\,.$$

So far, we have identified the subspace V and we know that there exists a bounded surface area Boolean concept that depends on V and achieves low L_1 error with the smoothed label $T_\rho y$. Since we have established that the smoothing operation T_ρ does not affect the excess error of a bounded-surface area concept by a lot (see Proposition II.1 and Lemma V.11), we know that the same concept will achieve low excess-error with respect to the original label y. Having established this, for our final step we may use the L_1-agnostic learner of [71] on the k-dimensional subspace V to learn a PTF of degree $\mathrm{poly}(\Gamma/\epsilon)$ with $(\dim(V))^{\mathrm{poly}(\Gamma/\epsilon)} = k^{\mathrm{poly}(\Gamma/\epsilon)}$ samples and time.

C. Hardness of Proper Agnostic Query Learning for ReLUs and Halfspaces

Here we sketch our hardness reduction, establishing that the exponential dependence in $1/\epsilon$ is inherent for proper agnostic learners, even with query access to the function (see Theorem VIII.3 and Theorem VIII.4). In particular, we show that assuming there are no polynomial-time algorithms for the Small-Set Expansion (SSE) problem [70], then there are no polynomial time *proper* agnostic learning algorithms for ReLUs and homogeneous halfspaces with respect to the Gaussian distribution.

The basic idea of our argument is to reduce to the problem of (approximately) optimizing a homogeneous degree-4 polynomial over the unit sphere (for the case of halfspaces we reduce to optimizing a degree-5 polynomial). As there are already known reductions from SSE to the problem of finding approximate maxima of degree-4 polynomials (and for halfspaces we can do a simple reduction from degree-4 to degree-5) this will suffice.

For this, we note that if $f(\mathbf{x})$ is a polynomial and $g(\mathbf{x}) = \mathrm{ReLU}(\mathbf{v}\cdot\mathbf{x})$ for $\mathbf{v}$ a unit vector, then $\mathbf{E}[f(\mathbf{x})g(\mathbf{x})]$ is a low-degree polynomial in $\mathbf{v}$. In fact, by specifying f, we can make this into any homogeneous degree-5 polynomial we desire. This gives us SSE hardness of approximating $\mathbf{E}[f(\mathbf{x})g(\mathbf{x})]$.

If f were a Boolean function we would be done. However, as this is not the case, we need two additional steps. Firstly, we scale down f and truncate it so that its values stay within $[-1, 1]$ (note that this introduces only a small error if the average size of f is small). Second, we replace f by a random Boolean function $\tilde{f}$ so that $\mathbf{E}[\tilde{f}(\mathbf{x})] = f(\mathbf{x})$. Doing this, it is not hard to see that with high probability over the randomness of defining $\tilde{f}$ that $\mathbf{E}[\tilde{f}(\mathbf{x})g(\mathbf{x})]$ is arbitrarily close to $\mathbf{E}[f(\mathbf{x})g(\mathbf{x})]$ for all functions g.

Now even if the algorithm was given an explicit description of our function $\tilde{f}$, finding a ReLU function g that approximately maximizes $\mathbf{E}[\tilde{f}(\mathbf{x})g(\mathbf{x})]$ is essentially equivalent to approximately optimizing a homogeneous degree-5 polynomial of the sphere, which is SSE-hard.

III. Related Work

Here we discuss prior and related work that was not already discussed in the introduction.

a) Comparison to Prior Work: We start by providing an explicit comparison with prior work.

Our algorithmic template involves two steps to agnostically learn multi-index models under the Gaussian distribution. First, we use queries to "smooth" the label function without adding a lot of noise to the instance. We then use PCA on the expected gradient outer-product of the "smoothed" concept $\mathbf{E}_{\mathbf{x}\sim D_{\mathbf{x}}}[\nabla f(\mathbf{x})\nabla f(\mathbf{x})^T]$ to find a low-dimensional space containing an (nearly) optimal hypothesis.

Using PCA on the expected gradient outer-product is a well-known dimension reduction technique that has been applied in many supervised learning settings, see, e.g., [24], [88]–[90]. We emphasize that prior results of this type focus on (i) the noiseless (realizable) setting, and (ii) the case of differentiable target functions. In comparison, we perform agnostic learning with non-differentiable functions by crucially exploiting query access. Using sample access only, estimating the gradient of $f(\mathbf{x})$ requires exponentially many examples in the dimension, see, e.g., [89].

[9] developed an efficient agnostic query learner for decision trees under the uniform distribution on the Boolean hypercube. The approach of [9] crucially relies on the fact that the target hypothesis can be represented as a sparse polynomial. The class of functions we consider (Definition I.12) — and in particular even a single halfspace or ReLU — does not have this property, and therefore methods relying on sparsity [7], [9] are not applicable.

In the context of property testing, [75] used a similar approach based on PCA on the expected outer gradient product to test whether the observed label is close to a smooth low-dimensional junta (similar to Definition I.12). An important difference with the current work is that in many interesting applications the link function may be assumed to be known, e.g., agnostically learning a ReLU or a halfspace, and the goal is to *learn* a good hypothesis — a task that information-theoretically requires $\Omega(d)$ samples. In contrast, [75] focuses on the semi-parametric task of only testing the unknown link function (and not identifying the underlying low-dimensional subspace) while avoiding a $\mathrm{poly}(d)$ dependence in the sample complexity.

Finally, related to our setting is the more recent work of [84], where a combination of polynomial regression and PCA on the average outer product of the gradient was employed for proper, agnostic learning of a single halfspace with runtime and sample complexity $d^{\mathrm{poly}(1/\epsilon)}$. In this work, we crucially exploit the query access to bypass the polynomial regression step and significantly improve the runtime to $\mathrm{poly}(d)2^{\mathrm{poly}(1/\epsilon)}$ (for the special case of a single halfspace).

b) Agnostically Learning Boolean Functions with Queries: In the context of learning Boolean functions, the study of distribution-specific agnostic learning with queries has a rich history. One of the earliest results in this vein is the classical algorithm of Goldreich and Levin [6] that uses queries to efficiently agnostically learn parity functions under the uniform distribution. (Recall that the problem of learning parities with noise is conjectured to be computationally hard with random samples only.) Kushilevitz and Mansour [7], building on the ideas of [6], developed an efficient (non-agnostic) query learner for decision trees under the uniform distribution. As already mentioned, [9] subsequently gave a polynomial-time agnostic query learner for decision trees under the uniform distribution.

It is known (see, e.g., [2]) that the availability of queries does *not* help computationally in the distribution-free agnostic setting. Specifically, Feldman [2] showed that every concept class that is agnostically learnable with queries is also agnostically learnable from random samples only (while preserving computational efficiency within a polynomial factor). This simple yet powerful fact has motivated the study of agnostic query learning *with respect to specific natural distributions*, such as the uniform distribution on the hypercube or the Gaussian distribution. [2] also showed that there exists a concept class that provides a computational separation (under cryptographic assumptions) between uniform distribution agnostic PAC learning and agnostic PAC+Query learning. Since this concept class is not natural, he asked whether queries are useful for *natural* concept classes such as halfspaces. As a special case of our main result, we answer this question in the affirmative.

IV. Roadmap, Notation, and Preliminaries

A. Roadmap

In Section V-A, we show that we can use queries to simulate gradient access to the Ornstein–Uhlenbeck smoothing $T_\rho y$. In Sections V-B and V-C, we show that the noise operator we use does not affect the agnostic learning task for real-valued functions and Boolean concepts. In Section VI, we show our result for learning real-valued functions and prove Theorem I.5. In Section VI-C, we show how Theorem I.5 implies agnostic learning for linear combinations of ReLU activations and deep networks. In Section VII, we give our agnostic learner for Boolean concepts with bounded surface area and establish Theorem I.13 and the associated applications. In Section VIII, we show that under the SSE hypothesis, no polynomial-time proper query learner for agnostically learning ReLUs or LTFs exists.

B. Notation and Preliminaries

a) Basic Notation: For $n \in \mathbb{Z}_+$, let $[n] \coloneqq \{1, \ldots, n\}$. We use small boldface characters for vectors and capital bold characters for matrices. For $\mathbf{x} \in \mathbb{R}^d$ and $i \in [d]$, $\mathbf{x}_i$ denotes the i-th coordinate of $\mathbf{x}$, and $\|\mathbf{x}\|_2 \coloneqq (\sum_{i=1}^d \mathbf{x}_i^2)^{1/2}$ denotes the ℓ_2-norm of $\mathbf{x}$. We will use $\mathbf{x} \cdot \mathbf{y}$ for the inner product of $\mathbf{x}, \mathbf{y} \in \mathbb{R}^d$ and $\theta(\mathbf{x}, \mathbf{y})$ for the angle between $\mathbf{x}, \mathbf{y}$. We slightly abuse notation and denote $\mathbf{e}_i$ the i-th standard basis vector in $\mathbb{R}^d$. We will use $\mathbb{1}_A$ to denote the characteristic function of the set A, i.e., $\mathbb{1}_A(\mathbf{x}) = 1$ if $\mathbf{x} \in A$ and $\mathbb{1}_A(\mathbf{x}) = 0$ if $\mathbf{x} \notin A$.

b) Asymptotic Notation: We use the standard $O(\cdot), \Theta(\cdot), \Omega(\cdot)$ asymptotic notation. We also use $\widetilde{O}(\cdot)$ to omit poly-logarithmic factors.

c) Probability Notation: We use $\mathbf{E}_{x\sim D}[x]$ for the expectation of the random variable x according to the distribution D and $\mathbf{Pr}[\mathcal{E}]$ for the probability of event $\mathcal{E}$. For simplicity of notation, we may omit the distribution when it is clear from the context. For $(\mathbf{x}, y)$ distributed according to D, we denote $D_{\mathbf{x}}$ to be the distribution of $\mathbf{x}$ and D_y to be the distribution of y. For unit vector $\mathbf{v} \in \mathbb{R}^d$, we denote $D_{\mathbf{v}}$ the distribution of $\mathbf{x}$ on the direction $\mathbf{v}$, i.e., the distribution of $\mathbf{x}_{\mathbf{v}}$.

d) Gaussian Space: Let $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ denote the d-dimensional Gaussian distribution with mean $\boldsymbol{\mu} \in \mathbb{R}^d$ and covariance $\boldsymbol{\Sigma} \in \mathbb{R}^{d\times d}$, we denote $\phi_d(\cdot)$ the pdf of the d-dimensional Gaussian and we use the $\phi(\cdot)$ for the pdf of the standard normal. In this work we usually consider the standard normal, i.e., $\mu = \mathbf{0}$ and $\boldsymbol{\Sigma} = \mathbf{I}$, and therefore, we denote it simply $\mathcal{N}$. We define the standard L^p norms with respect to the Gaussian measure, i.e., $\|g\|_{L^p} = (\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[|g(\mathbf{x})|^p)^{1/p}$. We denote by $L^2(\mathcal{N})$ the vector space of all functions $f : \mathbb{R}^d \to \mathbb{R}$ such that $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}_0}[f^2(x)] < \infty$. The usual inner product for this space is $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}_0}[f(\mathbf{x})g(\mathbf{x})]$. While, usually one considers the probabilists's or physicists' Hermite polynomials, in this work we define the *normalized* Hermite polynomial of degree i to be $H_0(x) = 1, H_1(x) = x, H_2(x) = \frac{x^2-1}{\sqrt{2}}, \ldots, H_i(x) = \frac{He_i(x)}{\sqrt{i!}}, \ldots$ where by $He_i(x)$ we denote the probabilists' Hermite polynomial of degree i. These normalized Hermite polynomials form a complete orthonormal basis for the single dimensional version of the inner product space defined above. To get an orthonormal basis for $L^2(\mathcal{N})$, we use a multi-index $V \in \mathbb{N}^d$ to define the d-variate normalized Hermite polynomial as $H_V(\mathbf{x}) = \prod_{i=1}^d H_{v_i}(x_i)$. The total degree of H_V is $|V| = \sum v_i \in V v_i$. Given a function $f \in L^2$ we compute its Hermite coefficients as $\hat{f}(V) = \mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[f(\mathbf{x})H_V(\mathbf{x})]$ and express it uniquely as $\sum_{V\in\mathbb{N}^d} \hat{f}(V)H_V(\mathbf{x})$. We denote by $\mathrm{P}_k f(\mathbf{x})$ the degree k partial sum of the Hermite expansion of f, $\mathrm{P}_k f(\mathbf{x}) = \sum_{|V|\le k} \hat{f}(V)H_V(\mathbf{x})$. Then, since the basis of Hermite polynomials is complete, we have $\lim_{k\to\infty} \mathbf{E}_{x\sim\mathcal{N}}[(f(\mathbf{x}) - \mathrm{P}_k f(\mathbf{x}))^2] = 0$. Parseval's identity states that $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[(f(\mathbf{x}) - \mathrm{P}_k f(\mathbf{x}))^2] = \sum_{|V|=k}^{\infty} \hat{f}(V)^2$.

V. From Zero- to First-Order: Derivative Queries via Oracle Queries

In this section, we show that we can efficiently simulate gradient access to a smoothed version of the label y using queries. In Section V-A we show how to use the Ornstein–Uhlenbeck operator to get acecss to gradient queries of y. In Section V-C and Section V-B we show that the noise that we introduce in order to simulate the gradient queries does not affect the agnostic learning task for Boolean and real valued concepts as long as the Gaussian surface area (for Boolean concepts) and the expected gradient norm (for real-valued functions) are bounded.

A. Gradient Queries via Oracle Queries

We first formally define the Ornstein–Uhlenbeck smoothing operator.

Definition V.1 (Ornstein–Uhlenbeck Operator)**.** *Let $\rho \in (0, 1)$. We denote as T_ρ the linear operator that maps a function $g \in L^2(\mathcal{N})$ to the function $T_\rho g$ defined as:*

$$(T_\rho g)(\mathbf{x}) \coloneqq \mathop{\mathbf{E}}_{\mathbf{z}\sim\mathcal{N}}\left[g(\sqrt{1-\rho^2}\mathbf{x} + \rho\mathbf{z})\right] .$$

To simplify notation, we often write $T_\rho g(\mathbf{x})$ instead of $(T_\rho g)(\mathbf{x})$.

The Ornstein–Uhlenbeck operator is well studied (see, e.g., [71], [91] and references therein) and has several structural properties that enable the analysis of our algorithm. Its crucial property is that regardless of how complex the initial function g is, $T_\rho g$ is always everywhere differentiable and also the norm of the gradient of $T_\rho g$ only depends on the maximum value of the function g. In the next fact we collect the properties that we use.

Fact V.2 (see, e.g., [91])**.** *Let $g : \mathbb{R}^d \mapsto \mathbb{R}$. For the function $T_\rho g(\mathbf{x})$ the following properties hold*

1. *$T_\rho g(\mathbf{x})$ is differentiable at every point $\mathbf{x}$.*
2. *$T_\rho g(\mathbf{x})$ is $1/\rho$-Lipschitz, i.e., $\|\nabla T_\rho g(\mathbf{x})\|_2 \le \|g\|_\infty/\rho$.*
3. *For any $p \ge 1$, T_ρ is a contraction with respect the $\|\cdot\|_p$, i.e., it holds $\|T_\rho g\|_{L^p} \le \|g\|_{L^p}$.*

Using it allows the gradient of the smoothed function $T_\rho g(\mathbf{x})$ to be computed directly given value access to the underlying function g. We now present the main result of this section showing that given query access to the label $y(\cdot)$ we can efficiently simulate gradient queries to the smoothed label $T_\rho y(\cdot)$ with roughly $\widetilde{O}(d/\epsilon)$ queries.

Lemma V.3 (Gradient Queries from Oracle Queries)**.** *Fix $\epsilon, \delta, \rho > 0$. Let $y(\mathbf{x}) : \mathbb{R}^d \mapsto \mathbb{R}$ be a function in $L^2_2(\mathcal{N})$ with $|y(\mathbf{x})| \le M$. There exists an algorithm (see Algorithm 1) that given a point $\mathbf{x} \in \mathbb{R}^d$ makes $N = \widetilde{\Omega}(dM/\epsilon)\log(1/\delta)$ queries to $y(\mathbf{x})$ and, in polynomial time, returns a vector $\widetilde{\xi}$ such that, with probability at least $1-\delta$, it holds $\|\widetilde{\xi} - \nabla T_\rho y(\mathbf{x})\|_2 \le \epsilon$.*

Proof. To show the lemma, we first need to show that for any point $\mathbf{x} \in \mathbb{R}^d$, we can use enough queries to estimate $D_\rho y(\mathbf{x})$ accurately, meaning that we need to estimate the random variable $\mathbf{Z} = \frac{\sqrt{1-\rho^2}}{\rho}\mathbf{E}_{\mathbf{z}\sim\mathcal{N}(\mathbf{0},\mathbf{I})}\left[y(\sqrt{1-\rho^2}\mathbf{x} + \rho\mathbf{z})\mathbf{z}\right]$ accurately. Note that by definition the random variable $\mathbf{Z}$ is $1/\rho^2$ sub-gaussian, therefore from a simple application of the Hoefding inequality, we get that with $O(dM/(\rho\epsilon)^2\log(1/\delta_1))$ queries, we can find a $\widetilde{\mathbf{Z}}$ such that $\|\widetilde{\mathbf{Z}} - \mathbf{E}[\mathbf{Z}]\|_2 \le \epsilon$ with probability at least $1-\delta_1$. □

Lemma V.4 (Gradient of Smoothed Label)**.** *Let $\rho \in (0, 1)$. We denote as D_ρ the linear operator that maps a function $g \in L^2(\mathcal{N})$ to the function $D_\rho g$ defined as: $(D_\rho g)(\mathbf{x}) \coloneqq \nabla(T_\rho g)(\mathbf{x})$. It holds that*

$$(D_\rho g)(\mathbf{x}) = \frac{\sqrt{1-\rho^2}}{\rho}\mathop{\mathbf{E}}_{\mathbf{z}\sim\mathcal{N}}\left[g(\sqrt{1-\rho^2}\mathbf{x} + \rho\mathbf{z})\mathbf{z}\right] .$$

To simplify notation, we often write $D_\rho g(\mathbf{x})$ instead of $(D_\rho g)(\mathbf{x})$.

Proof. We first observe that for any fixed $\mathbf{x}$ the random variable $\sqrt{1-\rho^2}\mathbf{x} + \rho\mathbf{z}$ is distributed according to $\mathcal{N}(\sqrt{1-\rho^2}\mathbf{x}, \rho^2\mathbf{I})$. Therefore, we have

$$T_\rho g(\mathbf{x}) = \mathop{\mathbf{E}}_{\mathbf{z}\sim\mathcal{N}}[g(\sqrt{1-\rho^2}\mathbf{x}+\rho\mathbf{z})] = \mathop{\mathbf{E}}_{\mathbf{u}\sim\mathcal{N}(\sqrt{1-\rho^2}\mathbf{x},\rho^2\mathbf{I})}[g(\mathbf{u})]$$

We can now directly compute the gradient of the smoothed function $T_\rho g$:

$$\begin{aligned}\nabla_{\mathbf{x}}(T_\rho g)(\mathbf{x}) &= \nabla_{\mathbf{x}} \mathop{\mathbf{E}}_{\mathbf{u}\sim\mathcal{N}(\sqrt{1-\rho^2}\mathbf{x},\rho^2\mathbf{I})}[g(\mathbf{u})] \\ &= \frac{\sqrt{1-\rho^2}}{\rho^2} \mathop{\mathbf{E}}_{\mathbf{u}\sim\mathcal{N}(\sqrt{1-\rho^2}\mathbf{x},\rho^2\mathbf{I})}\left[g(\mathbf{u})(\mathbf{u}-\sqrt{1-\rho^2}\mathbf{x})\right] \\ &= \frac{\sqrt{1-\rho^2}}{\rho} \mathop{\mathbf{E}}_{\mathbf{z}\sim\mathcal{N}}\left[g(\sqrt{1-\rho^2}\mathbf{x}+\rho\mathbf{z})\mathbf{z}\right] .\end{aligned}$$

□

Input: $\epsilon > 0$, $\delta > 0$, $\rho > 0$, location $\mathbf{x} \in \mathbb{R}^d$.
Requries: Sample and query access to distribution of labeled examples D
Output: An estimation $\widetilde{\xi}$ of $\nabla T_\rho y(\mathbf{x})$ such that $\|\widetilde{\xi} - \nabla T_\rho y(\mathbf{x})\|_2 \le \epsilon$.
1) Sample $N = \widetilde{O}(d/\epsilon)\log(1/\delta)$ points $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(N)} \sim \mathcal{N}$.
2) Perform N Queries at the locations $\mathbf{q}^{(j)} = \sqrt{1-\rho^2}\mathbf{x} + \rho\mathbf{z}^{(j)}$ and obtain $y^{(j)}$.
3) Return the empirical estimate $\widetilde{\xi} = \frac{\sqrt{1-\rho^2}}{N\rho}\sum_{j=1}^N y^{(j)}\mathbf{z}^{(j)}$.

Algorithm 1:*Simulating Gradient Queries with Queries*

B. Smoothing the Labels for Learning Real-valued Functions

In this section we show that adding noise to the label $y(\mathbf{x})$ in order to make it smooth and compute its gradients does not "change" the agnostic learning task significantly. Assume that there exists a learning algorithm that can learn a hypothesis $h(\cdot)$ that achieves ϵ-excess error compared to a class of concepts C, given access to the smooth labels $T_\rho y(\mathbf{x})$. In other words, assume that we are given a learner that finds a hypothesis $h(\cdot)$ that satisfies

$$\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[(h(\mathbf{x}) - T_\rho y(\mathbf{x}))^2] \le \inf_{f\in C} \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[(f(\mathbf{x}) - T_\rho y(\mathbf{x}))^2] + \epsilon .$$

Then, can we say that $h(\cdot)$ will perform well compared to the same class C under the original (non-smooth) label $y(\cdot)$? We show that this is true when (i) the hypothesis $h(\cdot)$ produced by the learner is not very complicated in the sense that it has bounded variation and (ii) the hypothesis class C that we are comparing $h(\cdot)$ against has also bounded variation.

In particular, we show that a hypothesis $h(\cdot)$ achieves ϵ-excess error compared to some concept class C in the *smoothed* instance, achieves $(\epsilon + O(\sqrt{\rho})$-excess error with respect to the original instance. In other words, as long as the variation and L_2^2 norms of the target concept class and the hypothesis produced by the learner are bounded, smoothing the noisy label $y(\mathbf{x})$ does not introduce significantly more noise to the instance. To simplify notation, we first define the excess error, i.e., the error of a classifier minus the error of the best-in-class classifier of some class C.

Definition V.5 (Excess Error)**.** *Given hypotheses $h, f : \mathbb{R}^d \mapsto \mathbb{R}$ we define the L_1-excess error of $h(\cdot)$ compared to $f(\cdot)$ with respect to the label $y(\cdot)$ to be $\mathcal{E}_1(h, f; y) = \mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[|h(\mathbf{x}) - y(\mathbf{x})|] - \mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[|f(\mathbf{x}) - y(\mathbf{x})|]$. Moreover, for a class of concepts C we define the excess error of $h(\cdot)$ compared to C with respect to $y(\cdot)$ as $\sup_{f\in C}\mathcal{E}_1(h, f; y)$. Similarly, we define the L_2^2-excess error as $\mathcal{E}_2(h, f; y) = \mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[(h(\mathbf{x}) - y(\mathbf{x}))^2] - \mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[(f(\mathbf{x}) - y(\mathbf{x}))^2]$ and $\mathcal{E}_2(h, C; y) = \sup_{f\in C}\mathcal{E}_2(h, f; y)$.*

We now show that that the Ornstein–Uhlenbeck noise operator also preserves the L_2^2-excess error of a classifier $h : \mathbb{R}^d \mapsto \mathbb{R}$ as long as the target class and the classifier h have bounded expected gradient.

Proposition V.6 (Smoothing the Noisy Labels)**.** *Fix $f \in \mathcal{R}(M, L, k)$. Let $y : \mathbb{R}^d \mapsto \mathbb{R}$ be a function in $L^2(\mathcal{N})$ with $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[y^2(\mathbf{x})] \le M$. Moreover, let $p(\mathbf{x}) : \mathbb{R}^d \mapsto \mathbb{R}$ be an almost everywhere differential function in $L_2(\mathcal{N})$ with $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[\|\nabla p(\mathbf{x})\|_2^2] \le L$. It holds that*

$$\mathcal{E}_2(p, C; y) \le \mathcal{E}_2(p, C; T_\rho y) + O(\sqrt{\rho ML}) .$$

Proof of Proposition V.6. We first prove the following lemma that connects the excess error of a real-valued function $h(\cdot)$ with respect to the smoothed label $T_\rho y(\cdot)$ to its excess error with respect to the original label $y(\cdot)$. If the operator T_ρ preserves the correlation of all concepts $f \in C$, i.e., $|\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[f(\mathbf{x})y(\mathbf{x})] - \mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[f(\mathbf{x})T_\rho y(\mathbf{x})]| \le \epsilon$ for all $f \in C$ and it also preserves the correlation of the hypothesis $h(\cdot)$, i.e., $|\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[h(\mathbf{x})y(\mathbf{x})] - \mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[h(\mathbf{x})T_\rho y(\mathbf{x})]| \le \epsilon$, then the excess error of $h(\cdot)$ with respect to $y(\cdot)$ is at most 2ϵ worse than its excess error with respect to the smoothed label $T_\rho y(\cdot)$. In the following lemma, we show that we can connect the L_2-excess error with the correlation of concepts.

Lemma V.7 (From Excess Error to Correlation Preservation)**.** *Let $h : \mathbb{R}^d \mapsto \mathbb{R}$ be a real-valued hypotheses and C be a class of real-valued hypotheses. It holds*

$$\begin{aligned}\mathcal{E}_2(h, C; T_\rho y) - \mathcal{E}_2(h, C; y) \le \\ 2\sup_{f\in C}\Big|\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[f(\mathbf{x})T_\rho y(\mathbf{x})] - \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[f(\mathbf{x})y(\mathbf{x})]\Big| + \\ 2\Big|\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[h(\mathbf{x})T_\rho y(\mathbf{x})] - \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[h(\mathbf{x})y(\mathbf{x})]\Big| .\end{aligned}$$

Proof. We first note that $\mathcal{E}_2(h, C; T_\rho y) - \mathcal{E}_2(h, C; y) = \sup_{f\in C}\mathcal{E}_2(h, f; T_\rho y) - \sup_{f\in C}\mathcal{E}_2(h, f; y) \le \sup_{f\in C}\big|\mathcal{E}_2(h, f; T_\rho y) - \mathcal{E}_2(h, f; y)\big|$. For some fixed concept $f \in C$, we have

$$\begin{aligned}\mathcal{E}_2(h, f; T_\rho y) = \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[h^2(\mathbf{x})] - \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[f^2(\mathbf{x})] \\ + 2\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[(f(\mathbf{x}) - h(\mathbf{x}))(T_\rho y)] .\end{aligned}$$

Therefore, we have

$$\mathcal{E}_2(h,f;T_\rho y)-\mathcal{E}_2(h,f;y)=2\Big(\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[f(\mathbf{x})(T_\rho y(\mathbf{x})-y(\mathbf{x}))] + \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[h(\mathbf{x})(T_\rho y(\mathbf{x})-y(\mathbf{x}))]\Big).$$

By taking the supremum over the f, we complete the proof. □

Note that $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[f(\mathbf{x})(T_\rho y(\mathbf{x})-y(\mathbf{x}))] = \mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[y(\mathbf{x})(T_\rho f(\mathbf{x})-f(\mathbf{x}))]$. Therefore, using Cauchy-Schwarz inequality we have that

$$\begin{aligned}&\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[y(\mathbf{x})(T_\rho f(\mathbf{x})-f(\mathbf{x}))]\\ &\le\Big(\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[y^2(\mathbf{x})]\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[(T_\rho f(\mathbf{x})-f(\mathbf{x}))^2]\Big)^{1/2}\\ &\le\sqrt{M}\Big(\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[(T_\rho f(\mathbf{x})-f(\mathbf{x}))^2]\Big)^{1/2},\end{aligned}$$

where we used that $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[y^2(\mathbf{x})]\le\sqrt{\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[y^4(\mathbf{x})]}\le M$. To bound the remaining term, we prove the following claim.

Claim V.8. *Let $f\in L^2(\mathcal{N})$ be a continuous and (almost everywhere) differentiable function. Then, $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[(T_\rho f(\mathbf{x})-f(\mathbf{x}))^2]\le 2\rho^2\,\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[\|\nabla f(\mathbf{x})\|_2^2]$.*

Proof. We will use the following result from [74].

Fact V.9 (Correlated Differences, (Lemma 7 in [74]))**.** *Let $f\in L^2(\mathcal{N})$ be an (almost everywhere) differentiable function. Denote by*

$$D_\tau=\mathcal{N}\Bigg(\mathbf{0},\begin{pmatrix}\mathbf{I} & (1-\tau)\mathbf{I}\\ (1-\tau)\mathbf{I} & \mathbf{I}\end{pmatrix}\Bigg).$$

It holds $\mathbf{E}_{(\mathbf{x},\mathbf{z})\sim D_\tau}[(f(\mathbf{x})-f(\mathbf{z}))^2]\le 2\tau\;\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[\|\nabla f(\mathbf{x})\|_2^2]$.

Therefore, using Jensen's inequality, we have that

$$\begin{aligned}&\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[(T_\rho f(\mathbf{x})-f(\mathbf{x}))^2]\\ &=\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[(\mathop{\mathbf{E}}_{\mathbf{z}\sim\mathcal{N}}[f(\sqrt{1-\rho^2}\mathbf{x}+\rho\mathbf{z})]-f(\mathbf{x}))^2]\\ &\le\mathop{\mathbf{E}}_{(\mathbf{x},\mathbf{z}')\sim D_\tau}[(f(\mathbf{z}')-f(\mathbf{x}))^2],\end{aligned}$$

for $\tau=1-\sqrt{1-\rho^2}$. Therefore, using Fact V.9, we obtain

$$\begin{aligned}\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[(T_\rho f(\mathbf{x})-f(\mathbf{x}))^2]&\le 2(1-\sqrt{1-\rho^2})\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[\|\nabla f(\mathbf{x})\|_2^2\\ &\le 2\rho^2\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[\|\nabla f(\mathbf{x})\|_2^2,\end{aligned}$$

where we used the fact that $\sqrt{1-\rho^2}\ge 1-\rho^2$ which holds for all $\rho\in[0,1]$ and implies that $1-\sqrt{1-\rho^2}\le\rho^2$. □

Therefore, from Claim V.8, we have that

$$\begin{aligned}&\mathcal{E}_2(p,C;y)\le\mathcal{E}_2(p,C;T_\rho y)\\ &+O(\sqrt{\rho M})\Big(\sqrt{\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[\|\nabla f(\mathbf{x})\|_2^2]}+\sqrt{\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[\|\nabla p(\mathbf{x})\|_2^2]}\Big).\end{aligned}$$

Using that $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[\|\nabla f(\mathbf{x})\|_2^2],\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[\|\nabla p(\mathbf{x})\|_2^2]\le L$, we complete the proof of Proposition V.6. □

C. Smoothing Labels for Learning Boolean Concepts

The following proposition shows that the L_1-excess error of a hypothesis h with respect to the original label y is close to its L_1-excess error with respect to the smoothed label $T_\rho y$ as long as (i) the class C contains concepts with bounded surface area and (ii) the classifier h also has bounded surface area.

Proposition V.10 (Smoothing the Noisy Labels Preservs L_1-Excess Error)**.** *Fix $y:\mathbb{R}^d\mapsto\{\pm1\}$ and let C be a class of Boolean concepts. It holds*

$$\mathcal{E}_1(h,C;y)\le\mathcal{E}_1(h,C;T_\rho y)+O(\rho)\;(\Gamma(C)+\Gamma(h)),$$

where $\mathcal{E}(\cdot,\cdot;\cdot)$ is the excess error defined in Definition V.5

Proof. We first prove the following lemma showing that connects the excess error of a classifier $h(\cdot)$ with respect to the smoothed label $T_\rho y(\cdot)$ to its excess error with respect to the original label $y(\cdot)$. This is analogous to the real-valued case (Lemma V.7). In the following lemma we show that we can connect the L_1-excess error with the correlation of concepts (which basically relies on the identity $|t-s|=1-ts$ when $t\in[-1,1]$ and $s\in\{\pm1\}$.

Lemma V.11 (From Excess Error to Correlation Preservation: Boolean Concepts)**.** *Let $h:\mathbb{R}^d\mapsto\{\pm1\}$ and C be a class of Boolean hypotheses. It holds*

$$\begin{aligned}&\mathcal{E}_1(h,C;T_\rho y)-\mathcal{E}_1(h,C;y)\\ &\le\sup_{f\in C}\Big|\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[f(\mathbf{x})T_\rho y(\mathbf{x})]-\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[f(\mathbf{x})y(\mathbf{x})]\Big|+\\ &\Big|\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[h(\mathbf{x})T_\rho y(\mathbf{x})]-\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[h(\mathbf{x})y(\mathbf{x})]\Big|.\end{aligned}$$

Proof. We first note that $\mathcal{E}_1(h,C;T_\rho y)-\mathcal{E}_1(h,C;y)=\sup_{f\in C}\mathcal{E}_1(h,f;T_\rho y)-\sup_{f\in C}\mathcal{E}_1(h,f;y)\le\sup_{f\in C}\big|\mathcal{E}_1(h,f;T_\rho y)-\mathcal{E}_1(h,f;y)\big|$. Using the fact that $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[|f_1(\mathbf{x})-f_2(\mathbf{x})|]=1-\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[f_1(\mathbf{x})f_2(\mathbf{x})]$, for any functions $f_1:\mathbb{R}^d\mapsto[-1,1]$ and $f_2:\mathbb{R}^d\mapsto\{\pm1\}$, we have that

$$\begin{aligned}\mathcal{E}_1(h,f;T_\rho y)&=\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[|T_\rho y(\mathbf{x})-h(\mathbf{x})|]-\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[|T_\rho y(\mathbf{x})-f(\mathbf{x})|]\\ &=\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[T_\rho y(\mathbf{x})f(\mathbf{x})]-\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[T_\rho y(\mathbf{x})h(\mathbf{x})].\end{aligned}$$

Therefore, for some concept $f\in C$, we have that

$$\begin{aligned}&\big|\mathcal{E}_1(h,f;T_\rho y)-\mathcal{E}_1(h,f;y)\big|=\\ &\Big|\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[(T_\rho y(\mathbf{x})-y(\mathbf{x}))f(\mathbf{x})]\Big|+\Big|\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[(T_\rho y(\mathbf{x})-y(\mathbf{x}))h(\mathbf{x})]\Big|.\end{aligned}$$

Taking the supremum over the C completes the proof. □

First, note that since $|y(\mathbf{x})|\le 1$, it also holds that $|T_\rho y(\mathbf{x})|\le 1$. Using Lemma V.11, we have that Proposition V.10 is equivalent to showing that for a Boolean function $f:\mathbb{R}^d\mapsto\{\pm1\}$ it holds $|\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[(T_\rho y(\mathbf{x})-y(\mathbf{x}))f(\mathbf{x})]|\le O(\rho)\;\Gamma(f)$. We do this in the following lemma.

Lemma V.12 (T_ρ Preserves Correlation)**.** *Let $y:\mathbb{R}^d\mapsto\{\pm1\}$ and let $f:\mathbb{R}^d\mapsto\{\pm1\}$ be a (Borel) Boolean function. It holds that*

$$\Big|\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[f(\mathbf{x})T_\rho y(\mathbf{x})]-\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[f(\mathbf{x})y(\mathbf{x})]\Big|\le O(\rho)\;\Gamma(f).$$

Proof. Using the fact that the Ornstein–Uhlenbeck noise operator T_ρ is a symmetric linear operator on $L^2(\mathcal{N})$, we have

$$\begin{aligned}&\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[f(\mathbf{x})T_\rho y(\mathbf{x})] = \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[y(\mathbf{x})T_\rho f(\mathbf{x})]\\ &= \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[y(\mathbf{x})f(\mathbf{x})] + \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[y(\mathbf{x})(T_\rho f(\mathbf{x}) - f(\mathbf{x}))]\,.\end{aligned}$$

Therefore,

$$\begin{aligned}&\Big|\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[f(\mathbf{x})T_\rho y(\mathbf{x})] - \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[f(\mathbf{x})y(\mathbf{x})]\Big|\\ &= \Big|\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[y(\mathbf{x})(T_\rho f(\mathbf{x}) - f(\mathbf{x}))]\Big|\\ &\le \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[|T_\rho f(\mathbf{x}) - f(\mathbf{x})|]\,,\end{aligned}$$

where, for the inequality we used the fact that the label $y(\mathbf{x}) \in \{\pm 1\}$. We next bound the term $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[|T_\rho f(\mathbf{x}) - f(\mathbf{x})|]$. We will use the following result from Ledoux and Pisier as stated in [71].

Fact V.13 (Ledoux-Pisier [92])**.** *Let $f : \mathbb{R}^d \mapsto \{\pm 1\}$ be a Boolean function. It holds $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[f(\mathbf{x})T_\rho f(\mathbf{x})] \ge 1 - 2\sqrt{\pi}\,\Gamma(f)\,\rho$.*

In what follows, we denote by K the set labeled as positive by the LTF $f(\mathbf{x})$. Using the fact that $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[|T_\rho f(\mathbf{x}) - f(\mathbf{x})|] = 1 - \mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[T_\rho f(\mathbf{x}) f(\mathbf{x})]$, which holds because $|T_\rho f(\mathbf{x})| \le 1$ and $f(\mathbf{x}) \in \{\pm 1\}$, we have

$$\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[|T_\rho f(\mathbf{x}) - f(\mathbf{x})|] = 1 - \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[f(\mathbf{x})T_\rho f(\mathbf{x})] \le O(\rho\Gamma(f))\,,$$

where the inequality follows from Fact V.13.

□

Applying Lemma V.12 on f and g gives the result. □

VI. Agnostically Learning Real-valued Multi-index Models

In this section we present our algorithmic result Theorem I.5 for learning real-valued function classes in the L_2^2 norm. For convenience, we first restate the class of bounded variation concepts that we consider.

Definition VI.1 (Bounded Variation, Low-Dimensional Concepts)**.** *Fix $L, M > 0$ and $k \in \mathbb{Z}_+$. We define the class $\mathfrak{R}(M, L, k)$ of continuous, (almost everywhere) differentiable real-valued functions with the following properties:*

1) *For every $f \in \mathfrak{R}(M, L, k)$, it holds $(\mathbf{E}_{\mathbf{x}\sim\mathcal{N}^d}[f^4(\mathbf{x})])^{1/2} \le M$ and $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}^d}[\|\nabla f(\mathbf{x})\|_2^2] \le L$.*
2) *There exists a subspace U of $\mathbb{R}^d$ of dimension at most k such that f depends only on U, i.e., for every $\mathbf{x} \in \mathbb{R}^d$, $f(\mathbf{x}) = f(\mathrm{proj}_U \mathbf{x})$.*

We now state the main result of this section (the formal version of Theorem I.5).

Theorem VI.2 (Improper Learner for Real-valued Functions)**.** *Fix $k \in \mathbb{N}$ and $M, L \in \mathbb{R}^+$. Let D be a distribution on $\mathbb{R}^d \times \mathbb{R}^+$ such that the $\mathbf{x}$-marginal of D is standard d-dimensional normal. There exists an algorithm that makes $N_q = \mathrm{poly}(d/\epsilon)$ queries, draws $N_s = \mathrm{poly}(d) + \mathrm{poly}((kM/\epsilon)^{L^2/\epsilon^4}, 1/\epsilon, \log(1/\delta))$ samples from D, runs in time $\mathrm{poly}(N_s, N_q, d)$ and outputs a polynomial $p : \mathbb{R}^d \mapsto \mathbb{R}$ so that with probability at least $1 - \delta$ it holds*

$$\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D}[(p(\mathbf{x}) - y)^2] \le \inf_{f\in\mathfrak{R}(M,L,k)} \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D}[(f(\mathbf{x}) - y)^2] + \epsilon\,.$$

Before we proceed to the proof we define the Hermite expansion operator that maps a function f to its degree m Hermite polynomial.

Definition VI.3 (Hermite Expansion Operator)**.** *Given a function $f \in L^2(\mathcal{N})$, we denote by $P_m(f)(\mathbf{x})$, the linear operator that maps f to the Hermite polynomial of degree m of f, i.e.,*

$$(\mathrm{P}_m f)(\mathbf{x}) = \sum_{|I|\le m} \widehat{f}(I) H_I(\mathbf{x}),$$

where H_I is the multivariate Hermite polynomial of degree $I \in \mathbb{N}^d$ and $\widehat{f}(I) = \mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[f(\mathbf{x})H_I(\mathbf{x})]$ is the corresponding Hermite coefficient of $f(\mathbf{x})$.

The following lemma bounds the error of the polynomial approximation of degree m for "smooth" functions. Its proof is implicit in [74]; we provide a short proof for completeness.

Lemma VI.4 (Polynomial Approximation of Smooth Functions)**.** *Let $f(\mathbf{x}) : \mathbb{R}^d \mapsto \mathbb{R}$ be an (almost everywhere) differentiable function and $m \in \mathbb{N}$. It holds*

$$\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[(f(\mathbf{x}) - \mathrm{P}_m f(\mathbf{x}))^2] \le O\Big(\frac{1}{m}\Big) \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[\|\nabla f(\mathbf{x})\|_2^2]\,.$$

Proof. We denote as $\mathrm{P}_{>m} f$ the Hermite expansion of f, which contains the terms with degrees higher than m. We have that

$$\begin{aligned}\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[(f(\mathbf{x}) - \mathrm{P}_m f(\mathbf{x}))^2] &= \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[(\mathrm{P}_{>m} f(\mathbf{x}))^2] = \sum_{I:|I|>m} (\widehat{f}(I))^2\\ &\le \frac{1}{m} \sum_{I:|I|>m} |I|(\widehat{f}(I))^2\,,\end{aligned}$$

where in the last inequality, we used that $1 \le |I|/m$. Furthermore, (see, e.g., the proof of Lemma 6 in [74]) we have that for a continuous and (almost everywhere) differentiable function f, it holds that $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[\|\nabla f(\mathbf{x})\|_2^2] = \sum_{I\in\mathbb{N}^d} |I|(\widehat{f}(I))^2$. Combining the above, the result follows. □

As we discussed in Section II to show that an approximately optimal, low-dimensional concept exists we will use the Gaussian Marginalization Operator defined below.

Definition VI.5 (Gaussian Marginalization Operator)**.** *Let U be a subspace of $\mathbb{R}^d$. Denote by $D_{U^\perp}$ the standard normal distribution on the subspace $U^\perp$ (we assume that a vector $\mathbf{z} \sim D_{U^\perp}$ is a d-dimensional vector that lies in $U^\perp$). Given a function $f \in L^2(\mathcal{N})$, we denote by $\Pi_U f$ the linear operator defined by*

$$(\Pi_U f)(\mathbf{x}) = \mathop{\mathbf{E}}_{\mathbf{z}\sim D_{U^\perp}}[f(\mathrm{proj}_U(\mathbf{x}) + \mathbf{z})]\,.$$

a) Motivation about the Gaussian Marginalization Operator, Π_V: By the assumption of Proposition II.4, all directions in the orthogonal complement $V^\perp$ are low-influence, i.e., for $\mathbf{h} \in V^\perp$ it holds $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[(\mathbf{h}\cdot\nabla\widetilde{y}(\mathbf{x}))^2] \leq O(\epsilon^2/k)$. In words, the function $\widetilde{y}$ is "approximately constant" along some low-influence direction $\mathbf{h}$. Let us first assume that $\widetilde{y}$ is exactly constant on all directions of $V^\perp$. Then, in order to preserve the correlation of $\widetilde{y}$ with f, *we only need to match the expected value of f over $V^\perp$*. This motivates the following "Gaussian Marginalization Operator" of Definition VI.5. Indeed, if $\widetilde{y}$ was constant on $V^\perp$, using the fact that $\mathrm{proj}_V\mathbf{x}$ and $\mathrm{proj}_{V^\perp}\mathbf{x}$ are independent standard Gaussians, we would obtain that $\mathbf{E}_{\mathbf{z}\sim\mathcal{N}}[\mathbf{E}_{\mathbf{x}\sim\mathcal{N}} f(\mathrm{proj}_V(\mathbf{x})+\mathrm{proj}_{V^\perp}(\mathbf{z}))\widetilde{y}(\mathbf{x})]] - \mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[f(\mathbf{x})\widetilde{y}(\mathbf{x})] = 0$. We observe that since $\Pi_V f$ is a convex combination of different translations of f and $\mathfrak{B}(\Gamma,k)$ is closed under translations, we obtain that the Gaussian surface area of f is also bounded above by Γ.

In the next lemmas, we collect some useful properties of the Gaussian Marginalization Operator. The proofs can be found in the full version of the paper.

Lemma VI.6. *Let $g \in L^2(\mathcal{N})$ and $V \subseteq \mathbb{R}^d$. We have the following properties for the operator Π_V.*

- *Π_V are contractions, i.e., $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[(\Pi_V g(\mathbf{x}))^2] \leq \mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[g^2(\mathbf{x})]$.*
- *Let $U, V \subseteq \mathbb{R}^d$, it holds that $\Pi_V\Pi_{U+V}g = \Pi_{V+U}\Pi_{V+U^\perp}g = \Pi_V g$.*

Lemma VI.7. *Let $g \in L^2(\mathcal{N})$, $m \in \mathbb{N}$ and $V \subseteq \mathbb{R}^d$. We have the following properties for the operators P_m and Π_V.*

- *P_m is a contraction, i.e., $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[(\mathrm{P}_m g(\mathbf{x}))^2] \leq \mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[g^2(\mathbf{x})]$.*
- *P_m and Π_V commute, i.e., $\mathrm{P}_m\Pi_V g = \Pi_V \mathrm{P}_m g$.*

Next, we show that Π_U and P_m commute. The proof can be found in the full version of the paper.

Claim VI.8 (P_m and Π_U commute)**.** *Let $g \in L^2(\mathcal{N})$, $m \in \mathbb{N}$, and V be a subspace of $\mathbb{R}^d$. It holds that $\mathrm{P}_m\Pi_V g = \Pi_V \mathrm{P}_m g$.*

Input: $\epsilon > 0$, $\delta > 0$ and sample and query access to distribution D
Output: An estimation of $\mathbf{M} = \mathbf{E}_{\mathbf{x}\sim D_\mathbf{x}}[D_\rho y(\mathbf{x}) D_\rho y(\mathbf{x})^\top]$.

1) $\rho \leftarrow C\epsilon^2$, $\eta \leftarrow C\epsilon^2$, for $C > 0$ sufficiently small constant.
2) Let S_N be the set that contains N samples $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(N)}$ from the distribution D.
3) For each $\mathbf{x} \in S_N$, use Algorithm 1 to get a gradient estimate $\widehat{(D_\rho y)}(\mathbf{x})$ of $(D_\rho y)(\mathbf{x})$.
4) **return** $\widehat{\mathbf{M}} = \frac{1}{N}\sum_{i=1}^N \widehat{(D_\rho y)}(\mathbf{x}^{(i)})\widehat{(D_\rho y)}(\mathbf{x}^{(i)})^\top$.

Algorithm 2:*Estimation of the influence matrix $\mathbf{M}$ with Queries*

Having access to the gradient, enables us to calculate the influence matrix of the function which captures the sensitivity of the function in different directions. We formally define the influence matrix of a function g.

Definition VI.9 (Influence Matrices)**.** *Given a differentiable $g \in L^2(\mathcal{N})$, we define the influence matrix as*

$$\mathbf{Inf}_g \coloneqq \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[\nabla g(\mathbf{x})\nabla g(\mathbf{x})^\top].$$

Fix $\rho \in (0,1)$. Given $g \in L^2(\mathcal{N})$ (not necessarily differentiable), we define its ρ-smoothed influence matrix as

$$\mathbf{Inf}_g^\rho \coloneqq \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[D_\rho g(\mathbf{x})(D_\rho g(\mathbf{x}))^\top].$$

A. Influence PCA for Learning in L_2^2

In this section we show that for learning real-valued concepts of bounded variation in L_2^2 we can effectively reduce the dimension of the problem via PCA in the influence of the smoothed label $T_\rho y$. We show that the low-degree polynomial approximation of the smoothed label $T_\rho y$ can be projected down to the subspace V via the Gaussian Marginalization Operator. In other words, we construct an explicit polynomial approximation of the label T_ρ that depends only on the low-dimensional subspace V. We now state our dimension-reduction result.

Proposition VI.10. *Fix $\epsilon, M, L, Q > 0$ and let $\psi : \mathbb{R}^d \mapsto \mathbb{R}$ with $|\psi(\mathbf{x})| \leq Q$ and $\|\nabla\psi(\mathbf{x})\|_2 \leq \Psi$. Let η be sufficiently small multiple of $\epsilon^2/(kM)$ and m be sufficiently large multiple of $(Q^2L)/\epsilon^2$. Let $\widehat{\mathbf{M}}$ be so that $\|\mathbf{Inf}_\psi - \widehat{\mathbf{M}}\|_2 \leq \eta/2$ and let V be the subspace spanned by all the eigenvectors of $\widehat{\mathbf{M}}$ whose corresponding eigenvalues are at least η. Then, it holds*

1)

$$\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[(\mathrm{P}_m\Pi_V\psi(\mathbf{x}) - \psi(\mathbf{x}))^2] \leq \inf_{f\in\mathfrak{R}(M,L,k)} \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[((\psi(\mathbf{x}) - f(\mathbf{x}))^2] + \epsilon\,.$$

2) The dimension of V is at most $O(\Psi^2/\eta)$.

Proof of Proposition VI.10. Fix $f \in \mathfrak{R}(M,L,k)$. By assumption, there exists a subspace U of dimension at most k, so that f depends only on U, i.e., $f(\mathbf{x}) = f(\mathrm{proj}_U\mathbf{x})$. Therefore, $\Pi_{U+V}f(\mathbf{x}) = f(\mathbf{x})$.

Lemma VI.11 (Excess L_2^2 Error Decomposition)**.** *We have*

$$\mathcal{E}_2(P_m\Pi_V\psi, f; \psi) \leq Q\underbrace{(\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[(f(\mathbf{x}) - P_m f(\mathbf{x}))^2])^{1/2}}_{\textit{Polynomial Approximation Error}} + 2\underbrace{\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[(\psi(\mathbf{x}) - \Pi_V\psi(\mathbf{x}))f(\mathbf{x})]}_{\textit{Correlation Error}}\,.$$

For the proof of Lemma VI.11 refer to the full version of the paper.

Lemma VI.12 (Correlation Error Bound)**.** *It holds*

$$\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[(\psi(\mathbf{x}) - \Pi_V\psi(\mathbf{x}))f(\mathbf{x})] \leq O(\epsilon)\,. \tag{1}$$

Proof. Note that $f(\mathbf{x})$ depends only on the subspace U, therefore, $\Pi_{U+V} f(\mathbf{x}) = f(\mathbf{x})$. Therefore, we have that

$$\begin{aligned}
&\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[(\psi(\mathbf{x}) - \Pi_V\psi(\mathbf{x}))f(\mathbf{x})] \\
&= \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[(\Pi_{V+U}\psi(\mathbf{x}) - \Pi_{V+U}\Pi_V\psi(\mathbf{x}))f(\mathbf{x})] \\
&= \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[(\Pi_{V+U}\psi(\mathbf{x}) - \Pi_V\Pi_{V+U}\psi(\mathbf{x}))f(\mathbf{x})] \\
&\leq \left(\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[(\Pi_{V+U}\psi(\mathbf{x}) - \Pi_V\Pi_{V+U}\psi(\mathbf{x}))^2] \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[f^2(\mathbf{x})]\right)^{1/2} ,
\end{aligned}$$

where in the last equality we used Lemma VI.7 and in the last inequality we used the Cauchy-Schwarz inequality. Note that $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[f^2(\mathbf{x})] \leq M$. To bound the other term we show that $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[(\Pi_{V+U}\psi(\mathbf{x}) - \Pi_V\Pi_{U+V}\psi(\mathbf{x}))^2]$ is small. For that, we prove the following:

Lemma VI.13 (Generalized Gaussian Marginalization Error)**.** *Let $g : \mathbb{R}^d \mapsto \mathbb{R}$ be a function in $L^2(\mathcal{N})$ such that $\nabla g \in L^2(\mathcal{N})$ and let V, U be subspaces of $\mathbb{R}^d$. It holds*

$$\begin{aligned}
&\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[(\Pi_{V+U}g(\mathbf{x}) - \Pi_V\Pi_{V+U}g(\mathbf{x}))^2] \\
&\leq \dim(V^\perp \cap U) \max_{\mathbf{v}\in V^\perp\cap U, \|\mathbf{v}\|_2=1} \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[(\nabla g(\mathbf{x})\cdot\mathbf{v})^2] .
\end{aligned}$$

The proof of Lemma VI.13 can be found in the version of the paper. From Lemma VI.13, we have that

$$\begin{aligned}
&\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[(\Pi_{V+U}\psi(\mathbf{x}) - \Pi_V\Pi_{U+V}\psi(\mathbf{x}))^2] \\
&\leq \dim(U \cap V^\perp) \max_{\mathbf{v}\in U\cap V^\perp, \|\mathbf{v}\|_2=1} \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[((\nabla\psi(\mathbf{x}))\cdot\mathbf{v})^2] .
\end{aligned}$$

Furthermore note that $\max_{\mathbf{v}\in U\cap V^\perp, \|\mathbf{v}\|_2=1} \mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[((\nabla\psi(\mathbf{x}))\cdot\mathbf{v})^2] \leq \eta/2 + \max_{\mathbf{v}\in U\cap V^\perp, \|\mathbf{v}\|_2=1} \mathbf{v}^\top\widehat{\mathbf{M}}\mathbf{v} \leq 2\eta$ because the subspace $U \cap V^\perp$ contains vectors with influence at most η. Note that $\dim(U \cap V^\perp) \leq \dim(U) \leq k$ and noting $\eta = O(\epsilon^2/(Mk))$ completes the proof of Lemma VI.12. □

Combining Lemmas VI.11 and VI.12 and using that $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[(f(\mathbf{x}) - \mathrm{P}_m f(\mathbf{x}))^2] \leq L/m$ from Lemma VI.4, we get that $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[(\mathrm{P}_m\Pi_V\psi(\mathbf{x}) - \psi(\mathbf{x}))^2] \leq \inf_{f\in\mathfrak{R}(M,L,k)} \mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[((\psi(\mathbf{x}) - f(\mathbf{x}))^2] + \epsilon$. To show that the subspace V has small dimension, we show the following lemma. The proof can be found in the full version of the paper.

Lemma VI.14. *Fix $\eta > 0, \rho \in (0,1)$. Let ψ be a function from $\mathbb{R}^d$ to $\mathbb{R}$ such that $\|\nabla\psi(\mathbf{x})\|_2 \leq \Psi$ and let V be the subspace spanned by all the eigenvectors of $\mathbf{Inf}_g$ with eigenvalue at least η. Then the dimension of the subspace V is $\dim(V) = O(\Psi^2/\eta)$.*

An application of the lemma above (Lemma VI.14) gives, which gives that the subspace it at most $O(\Psi^2/\eta)$. This completes the proof of Proposition VI.10 □

B. Proof of Theorem VI.2

We use the following fact about the L_2 polynomial regression.

Fact VI.15 (see, e.g., Theorem D.7 [84])**.** *Let D be a distribution on $\mathbb{R}^d \times \mathbb{R}$ such that the $\mathbf{x}$-marginal of D is standard d-dimensional normal and the labels y are bounded by M. The L_2-regression algorithm draws $N = \mathrm{poly}((dm)^{m^2}, 1/\epsilon, M, \log(1/\delta))$ samples from D, runs in time $\mathrm{poly}(N, d)$, and outputs a polynomial $p : \mathbb{R}^d \mapsto \mathbb{R}$ such that with probability at least $1-\delta$ it holds $\mathbf{E}_{(\mathbf{x},y)\sim D}[(p(\mathbf{x})-y)^2] \leq \min_{p\in\mathcal{P}_m} \mathbf{E}_{(\mathbf{x},y)\sim D}[(p(\mathbf{x}) - y)^2] + \epsilon$, where $\mathcal{P}_m$ is the class of polynomials with degree at most m.*

We first show that we can truncate the labels with $|y(\mathbf{x})| \geq M' = M^{1/2}/\epsilon^{1/2}$ without increasing the error by a lot. We show that for $\mathrm{trunc}(y(\mathbf{x})) = \mathrm{sign}(y(\mathbf{x}))\min(|y(\mathbf{x})|, M')$ it holds that (see the full version for the proof)

$$\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[(f(\mathbf{x}) - \mathrm{trunc}(y(\mathbf{x})))^2] \leq \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[(f(\mathbf{x}) - y(\mathbf{x}))^2] + \epsilon .$$

For the rest of the proof, we assume that $y(\mathbf{x})$ is truncated at M'. Let $\psi(\mathbf{x}) = T_\rho y$ for $\rho = \mathrm{poly}(\epsilon/(ML))$. Note that $\|\nabla\psi(\mathbf{x})\|_2 \leq M'$. From Lemma V.3, with $N = \mathrm{poly}(d/\epsilon)\log(1/\delta)$ queries, we get that with probability $1 - \delta/2$ a matrix $\mathbf{M}$, so that $\|\mathbf{M} - \mathbf{Inf}_\psi\|_F \leq \epsilon$. Applying Proposition VI.10 to the matrix $\mathbf{M}$, we get that in the subspace V spanned by the eigenvectors of the matrix $\mathbf{M}$ with eigenvalues larger than $\eta = \mathrm{poly}(\epsilon/Mk))$ with dimension at most $O(\mathrm{poly}(M', 1/\eta, 1/\epsilon))$, there exists a polynomial $p : V \mapsto \mathbb{R}$ of degree $m = \mathrm{poly}(M_2/\epsilon)$ with $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[p^2(\mathbf{x})] \leq \mathbf{E}[\psi^2(\mathbf{x})] \leq (M')^2$, so that

$$\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[(p(\mathbf{x}) - \psi(\mathbf{x}))^2] \leq \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[(f(\mathbf{x}) - \psi(\mathbf{x}))^2] + \epsilon/2 .$$

From Proposition V.6, we get that for the same polynomial and using that $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[\|\nabla p(\mathbf{x})\|_2 \leq m\,\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[p^2(\mathbf{x})]$, it also holds that

$$\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[(p(\mathbf{x}) - y(\mathbf{x}))^2] \leq \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[(f(\mathbf{x}) - y(\mathbf{x}))^2] + \epsilon/2 .$$

Let $\mathbf{P} : \mathbb{R}^d \mapsto V$ be the projection matrix to the subspace V. Let $(\mathbf{P}\mathbf{x}, y) \sim D'$, where $(\mathbf{x}, y) \sim D$. We use the L_2-regression algorithm on D' and from Fact VI.15, using $\mathrm{poly}((kM/\epsilon)^{L^2/\epsilon^4}, 1/\epsilon, \log(1/\delta))$ samples from D', we get a polynomial $p' : V \mapsto \mathbb{R}$ so that with probability at least $1-\delta$, it holds

$$\begin{aligned}
\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[(p'(\mathbf{P}\mathbf{x}) - y(\mathbf{x}))^2] &\leq \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[(p(\mathbf{x}) - y(\mathbf{x}))^2] + \epsilon/2 \\
&\leq \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[(f(\mathbf{x}) - y(\mathbf{x}))^2] + \epsilon .
\end{aligned}$$

This completes the proof of Theorem VI.2.

C. Applications of Theorem VI.2

In this section, we apply Theorem VI.2 for several real-valued activations. We start by applying our theorem for the class of ReLU activations.

Theorem VI.16 (Improper Learner for ReLUs Activations)**.** *Fix $M \in \mathbb{R}_+$. Let $\mathcal{C}$ be the concept class containing all the*

ReLU activations with normal vectors bounded in ℓ_2 norm by M. Let D be a distribution on $\mathbb{R}^d \times \mathbb{R}$ such that the $\mathbf{x}$-marginal of D is the standard d-dimensional normal. There exists an algorithm that makes $N_q = \mathrm{poly}(dM/\epsilon)$ queries, draws $N_s = \mathrm{poly}(d/\epsilon) + 2^{\mathrm{poly}(M/\epsilon)}\log(1/\delta)$ samples from D, runs in time $\mathrm{poly}(N_s, N_q, d)$ and outputs a polynomial $p : \mathbb{R}^d \mapsto \mathbb{R}$ so that with probability at least $1-\delta$ it holds

$$\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D}[(p(\mathbf{x})-y)^2] \leq \inf_{f\in\mathcal{C}} \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D}[(f(\mathbf{x})-y)^2] + \epsilon\,.$$

Proof. To prove the above theorem it suffices to show that $\mathcal{C} \subseteq \mathfrak{R}(\sqrt{3}M^2, M^2, 1)$. Note that $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[(\mathrm{ReLU}(\mathbf{w}\cdot\mathbf{x}))^4] \leq \mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[(\mathbf{w}\cdot\mathbf{x})^4] \leq 3M^4$. Furthermore, we bound the derivative of the activation. We have that

$$\mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[\|\nabla_{\mathbf{x}}\mathrm{ReLU}(\mathbf{w}\cdot\mathbf{x})\|_2^2] = \mathop{\mathbf{E}}_{\mathbf{x}\sim\mathcal{N}}[\|\mathbb{1}\{\mathbf{w}\cdot\mathbf{x}\geq 0\}\mathbf{w}\|_2^2] \leq M^2\,.$$

Therefore, it follows that $\mathcal{C} \subseteq \mathfrak{R}(\sqrt{3}M^2, M^2, 1)$. An application of Theorem VI.2 gives the result. □

We next consider learning Single-index models (SIMs) with an unknown Lipschitz link function $g : \mathbb{R} \mapsto \mathbb{R}$, i.e., $f(\mathbf{x}) = g(\mathbf{w}\cdot\mathbf{x})$.

Definition VI.17. *We define the class of L-Lipschitz SIMs on $\mathbb{R}^d$ denoted $\mathrm{SIM}(L, M)$ as follows. For each $f \in \mathrm{SIM}(L, M)$, $f(\mathbf{x}) = g(\mathbf{w}\cdot\mathbf{x})$, for L-Lipschitz $g : \mathbb{R} \mapsto \mathbb{R}$ and $\|\mathbf{w}\|_2 \leq M$.*

Theorem VI.18 (Improper Learner for SIMs)**.** *Fix $L, M \in \mathbb{R}_+$. Let D be a distribution on $\mathbb{R}^d \times \mathbb{R}$ such that the $\mathbf{x}$-marginal of D is the standard d-dimensional normal. There exists an algorithm that makes $N_q = \mathrm{poly}(dL/\epsilon)$ queries, draws $N_s = \mathrm{poly}(d/\epsilon) + 2^{\mathrm{poly}(LM/\epsilon)}\log(1/\delta)$ samples from D, runs in time $\mathrm{poly}(N_s, N_q, d)$ and outputs a polynomial $p : \mathbb{R}^d \mapsto \mathbb{R}$ so that with probability at least $1-\delta$ it holds*

$$\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D}[(p(\mathbf{x})-y)^2] \leq \inf_{f\in\mathrm{SIM}(L,M)} \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D}[(f(\mathbf{x})-y)^2] + \epsilon\,.$$

Proof. Note that for any $f \in \mathrm{SIM}(L)$ by definition if holds that $\|\nabla f(\mathbf{x})\|_2 \leq L$ and also that $\mathbf{E}[f^4(\mathbf{x})] \leq L^4\,\mathbf{E}[(\mathbf{w}\cdot\mathbf{x})^4] \lesssim M^4L^4$. Therefore, we have that $f \in \mathrm{SIM}(L, M) \subseteq \mathfrak{R}(M^2L^2, L, 1)$. An application of Theorem VI.2 gives the result. □

We define the class of linear combinations of ReLU networks.

Definition VI.19 (ReLU Networks)**.** *We define the class $\mathfrak{Re}(M, k)$ of ReLU networks as follows. For each $f \in \mathfrak{Re}(M, k)$, $f(\mathbf{x}) = \mathbf{W}_2\mathrm{ReLU}(\mathbf{W}_1\mathbf{x})$, for matrices $\mathbf{W}_1 \in \mathbb{R}^{k\times d}, \mathbf{W}_2 \in \{\pm 1\}^{k\times 1}$, with $\|\mathbf{W}_1\|_{op} \leq M$.*

We give our result for learning linear combinations of ReLUs, i.e., real-valued functions of the form $f(\mathbf{x}) = \sum_{i=1}^k a_i\mathrm{ReLU}(\mathbf{w}^{(i)}\cdot\mathbf{x})$, where $a_i \in \mathbb{R}$. The proof can be found in the full version of the paper.

Theorem VI.20 (Improper Learner for Linear Combinations of ReLUs)**.** *Fix $k \in \mathbb{N}$ and $M \in \mathbb{R}_+$. Let D be a distribution on $\mathbb{R}^d \times \mathbb{R}$ such that the $\mathbf{x}$-marginal of D is the standard d-dimensional normal. There exists an algorithm that makes $N_q = \mathrm{poly}(dM/\epsilon)$ queries, draws $N_s = \mathrm{poly}(d/\epsilon) + (kM/\epsilon)^{\mathrm{poly}(kM/\epsilon)}\log(1/\delta)$ samples from D, runs in time $\mathrm{poly}(N_s, N_q, d)$ and outputs a polynomial $p : \mathbb{R}^d \mapsto \mathbb{R}$ so that with probability at least $1-\delta$ it holds*

$$\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D}[(p(\mathbf{x})-y)^2] \leq \inf_{f\in\mathfrak{Re}(M,k)} \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D}[(f(\mathbf{x})-y)^2] + \epsilon\,.$$

We now give an improved result for learning sums of ReLUs, i.e., real-valued functions of the form $f(\mathbf{x}) = \sum_{i=1}^k \mathrm{ReLU}(\mathbf{w}^{(i)}\cdot\mathbf{x})$. We first define the class of sum of ReLUs.

Definition VI.21 (Sums of ReLU Networks)**.** *We define the class $\mathfrak{Re}_+(M, k)$ of ReLU networks as follows. For each $f \in \mathfrak{Re}_+(M, k)$, $f(\mathbf{x}) = \mathrm{ReLU}(\mathbf{W}\mathbf{x})$, for matrices $\mathbf{W} \in \mathbb{R}^{k\times d}$, with $\mathbf{E}[f^2(\mathbf{x})] \leq M$.*

Theorem VI.22 (Improper Learner for Sums of ReLUs)**.** *Fix $k \in \mathbb{N}$ and $M \in \mathbb{R}_+$. Let D be a distribution on $\mathbb{R}^d \times \mathbb{R}_+$ such that the $\mathbf{x}$-marginal of D is the standard d-dimensional normal. There exists an algorithm that makes $N_q = \mathrm{poly}(dM/\epsilon)$ queries, draws $N_s = \mathrm{poly}(d/\epsilon) + (kM/\epsilon)^{\mathrm{poly}(M/\epsilon)}\log(1/\delta)$ samples from D, runs in time $\mathrm{poly}(N_s, N_q, d)$ and outputs a polynomial $p : \mathbb{R}^d \mapsto \mathbb{R}$ so that with probability at least $1-\delta$ it holds*

$$\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D}[(p(\mathbf{x})-y)^2] \leq \inf_{f\in\mathfrak{Re}_+(M,k)} \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D}[(f(\mathbf{x})-y)^2] + \epsilon\,.$$

The proof Theorem VI.22 can be found in the full version of the paper.

Next we show our result for a general ReLU network. We first define the clas of Deep ReLU networks.

Definition VI.23 (Deep ReLU Networks)**.** *We define the class $\mathfrak{D}(M, L, k, S)$ of depth-$(L+1)$ ReLU networks as follows. For each $f \in \mathfrak{D}(M, L, k)$, $f(\mathbf{x}) = \mathbf{W}_L\mathrm{ReLU}(\mathbf{W}_{L-1}\cdots\mathrm{ReLU}(\mathbf{W}_1\mathbf{x}))$, for matrices $\mathbf{W}_1 \in \mathbb{R}^{k\times d}, \ldots, \mathbf{W}_L \in \mathbb{R}^{k_L\times 1}$, with $\|\mathbf{W}_i\|_{op} \leq M$ and $k_i \leq S$.*

We show the following theorem in the full version of the paper.

Theorem VI.24 (Agnostic Learner for Deep ReLU Networks)**.** *Fix $k, S, L \in \mathbb{N}$ and $M \in \mathbb{R}_+$. Let D be a distribution on $\mathbb{R}^d \times \mathbb{R}^+$ such that the $\mathbf{x}$-marginal of D is the standard d-dimensional normal. There exists an algorithm that makes $N_q = \mathrm{poly}(dM/\epsilon)$ queries, draws $N_s = \mathrm{poly}(d/\epsilon) + 2^{\mathrm{poly}(kSM/\epsilon)}\log(1/\delta)$ samples from D, runs in time $\mathrm{poly}(N_s, N_q, d)$ and outputs a polynomial $p : \mathbb{R}^d \mapsto \mathbb{R}$ so that with probability at least $1-\delta$ it holds*

$$\mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D}[(p(\mathbf{x})-y)^2] \leq \inf_{f\in\mathfrak{D}(M,L,k,S)} \mathop{\mathbf{E}}_{(\mathbf{x},y)\sim D}[(f(\mathbf{x})-y)^2] + \epsilon\,.$$

VII. Agnostically Learning Boolean Multi-index Models

In this section, we present our results for Boolean multi-index models of bounded surface area. For convenience, we restate the class of concepts that we consider.

Definition VII.1 (Bounded Surface Area, Low-Dimensional Boolean Concepts)**.** *We define the class $\mathfrak{B}(\Gamma, k)$ of Boolean concepts with the following properties:*

1) *For every $f \in \mathfrak{B}(\Gamma, k)$, it holds $\Gamma(f) \leq \Gamma$.*
2) *For every $f \in \mathfrak{B}(\Gamma, k)$, there exists a subspace U of $\mathbb{R}^d$ of dimension at most k such that f depends only on U, i.e., for every $\mathbf{x} \in \mathbb{R}^d$, $f(\mathbf{x}) = f(\mathrm{proj}_U \mathbf{x})$.*
3) *$\mathfrak{B}(\Gamma, k)$ is closed under translations, i.e., if $f(\mathbf{x}) \in \mathfrak{B}(\Gamma, k)$ then $f(\mathbf{x} + \mathbf{t}) \in \mathfrak{B}(\Gamma, k)$ for all $\mathbf{t} \in \mathbb{R}^d$.*

We remark that $\mathfrak{B}(\Gamma, k)$ is a general, *non-parametric class*. For example $\mathfrak{B}(\Omega(k), k)$ contains LTFs, intersections of k LTFs, and Polynomial Threhsold Functions (PTFs) of degree at most k (that depend on a k-dimensional subspace). Our learner is able to learn a hypothesis of low excess error when compared against all concepts of $\mathfrak{B}(\Gamma, k)$ with roughly $\mathrm{poly}(d/\epsilon) + k^{\mathrm{poly}(\Gamma/\epsilon)}$ runtime.

Theorem VII.2. *Fix $k \in \mathbb{N}$ and $M \in \mathbb{R}^+$. Let D be a distribution on $\mathbb{R}^d \times \{\pm 1\}$ such that the $\mathbf{x}$-marginal of D is standard d-dimensional normal. There exists an algorithm that makes $N_q = \mathrm{poly}(d/\epsilon)$ queries and draws $N_s = \mathrm{poly}(d/\epsilon) + \mathrm{poly}((k\Gamma/\epsilon)^{\Gamma^2/\epsilon^4}, 1/\epsilon, \log(1/\delta))$ samples from D and runs in time $\mathrm{poly}(N_s, N_q, d)$ and outputs a polynomial $p : \mathbb{R}^d \mapsto \mathbb{R}$ so that with probability at least $1 - \delta$ it holds*

$$\mathop{\mathbf{Pr}}_{(\mathbf{x},y)\sim D}[\mathrm{sign}(p(\mathbf{x})) \neq y] \leq \inf_{f \in \mathfrak{B}(\Gamma,k)} \mathop{\mathbf{Pr}}_{(\mathbf{x},y)\sim D}[f(\mathbf{x}) \neq y] + \epsilon \, .$$

We refer to the full version of the paper for the proof of Theorem VII.2.

A. Corollaries for Intersections of Halfspaces and PTFs

Using Theorem VII.2, we can show the following corollary for intersections of k halfspaces:

Corollary VII.3. *Let $\mathcal{C}$ be the class of intersections k halfspaces in $\mathbb{R}^d$. Let D be a distribution on $\mathbb{R}^d \times \{\pm 1\}$ such that the $\mathbf{x}$-marginal of D is the standard d-dimensional normal. There exists an algorithm that makes $N_q = \mathrm{poly}(d/\epsilon)$ queries and draws $N_s = \mathrm{poly}(d/\epsilon) + \mathrm{poly}((k/\epsilon)^{\log(k)/\epsilon^4}, 1/\epsilon, \log(1/\delta))$ samples from D and runs in time $\mathrm{poly}(N_s, N_q, d)$ and outputs a polynomial $p : \mathbb{R}^d \mapsto \mathbb{R}$ so that with probability at least $1 - \delta$ it holds*

$$\mathop{\mathbf{Pr}}_{(\mathbf{x},y)\sim D}[\mathrm{sign}(p(\mathbf{x})) \neq y] \leq \min_{f \in \mathcal{C}} \mathop{\mathbf{Pr}}_{(\mathbf{x},y)\sim D}[f(\mathbf{x}) \neq y] + \epsilon \, .$$

Proof of Corollary VII.3. For the proof, we use the fact that the Gaussian surface area of the intersection of k halfspaces is at most $O(\sqrt{\log k})$ (see Theorem 20 of [71]) and then the proof follows from Theorem VII.2. □

We show that we can use Theorem VII.2 to learn low-degree polynomial threshold functions (PTFs) that depend only on a small dimensional subspace.

Corollary VII.4. *Let $\mathcal{C}$ be the class of degree-ℓ PTFs in $\mathbb{R}^d$ that depend on an unknown k-dimensional subspace. Let D be a distribution on $\mathbb{R}^d \times \{\pm 1\}$ such that the $\mathbf{x}$-marginal of D is the standard d-dimensional normal. There exists an algorithm that makes $N_q = \mathrm{poly}(d/\epsilon)$ queries, draws $N_s = \mathrm{poly}(d/\epsilon) + \mathrm{poly}((k/\epsilon)^{\ell/\epsilon^4}, 1/\epsilon, \log(1/\delta))$ samples from D, runs in time $\mathrm{poly}(N_s, N_q, d)$ and outputs a polynomial $p : \mathbb{R}^d \mapsto \mathbb{R}$ so that with probability at least $1 - \delta$ it holds*

$$\mathop{\mathbf{Pr}}_{(\mathbf{x},y)\sim D}[\mathrm{sign}(p(\mathbf{x})) \neq y] \leq \min_{f \in \mathcal{C}} \mathop{\mathbf{Pr}}_{(\mathbf{x},y)\sim D}[f(\mathbf{x}) \neq y] + \epsilon \, .$$

Proof of Corollary VII.4. For the proof, we use the fact that the Gaussian surface area of degree-ℓ PTFs is at most $O(\ell)$ (see [72]) and the proof follows from Theorem VII.2. □

Finally, we show that we can use Theorem VII.2 to learn arbitrary functions of ℓ halfspaces.

Corollary VII.5. *Let $\mathcal{C}$ be the class of functions of ℓ halfspaces in $\mathbb{R}^d$. Let D be a distribution on $\mathbb{R}^d \times \{\pm 1\}$ such that the $\mathbf{x}$-marginal of D is the standard d-dimensional normal. There exists an algorithm that makes $N_q = \mathrm{poly}(d/\epsilon)$ queries, draws $N_s = \mathrm{poly}(d/\epsilon) + \mathrm{poly}((\ell/\epsilon)^{\ell/\epsilon^4}, 1/\epsilon, \log(1/\delta))$ samples from D, runs in time $\mathrm{poly}(N_s, N_q, d)$ and outputs a polynomial $p : \mathbb{R}^d \mapsto \mathbb{R}$ so that with probability at least $1 - \delta$ it holds*

$$\mathop{\mathbf{Pr}}_{(\mathbf{x},y)\sim D}[\mathrm{sign}(p(\mathbf{x})) \neq y] \leq \min_{f \in \mathcal{C}} \mathop{\mathbf{Pr}}_{(\mathbf{x},y)\sim D}[f(\mathbf{x}) \neq y] + \epsilon \, .$$

Proof of Corollary VII.5. We note that the Gaussian surface area of functions of ℓ halfspaces is bounded above by ℓ. From [71] (see, e.g., Fact 17), we have that the surface area of a Boolean function f that depends on ℓ halfspaces, is bounded above by the sum of the surface area of the individual halfspaces; therefore, we have that $\Gamma(f) \leq O(\ell)$. The proof follows from Theorem VII.2. □

VIII. Hardness of Agnostic Proper Learning of Halfspaces and ReLUs with Queries

One might ask if the exponential dependence on $1/\epsilon$ in our upper bound (Corollaries I.6 and I.14) is necessary or just an artifact of our algorithmic approach. In this section, we provide some evidence that it is inherent. Unfortunately, there are very few circumstances where one can prove computational lower bounds against improper learners with query access to the function. So our bounds will apply only to proper learners. The basic idea of our argument is that if $f(\mathbf{x}) = \mathrm{sign}(\mathbf{v} \cdot \mathbf{x})$ is a linear threshold function or $f(\mathbf{x}) = \mathrm{ReLU}(\mathbf{v} \cdot \mathbf{x})$ with $\mathbf{v}$ a unit vector and $p(\mathbf{x})$ a polynomial, then $\mathbf{E}[f(\mathbf{x})p(\mathbf{x})]$ will be a polynomial in $\mathbf{v}$. As approximately optimizing low-degree polynomials over the unit sphere is conjectured to be computationally hard, this will prove hardness for proper learning of linear threshold functions. In particular, our hardness reduction starts from the small-set expansion problem [70]. We then rely on results of [93] to reduce this problem to one about polynomial optimization. In particular we have:

Theorem VIII.1. *If there is a polynomial-time algorithm that given $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_n \in \mathbb{R}^d$ outputs a constant factor approximation to $\max_{\|\mathbf{x}\|_2 = 1} \frac{1}{n} \sum_{i=1}^n (\mathbf{a}_i \cdot \mathbf{x})^4$, then there is a polynomial time algorithm for the small-set expansion problem.*

We note here that $\max_{\|\mathbf{x}\|_2 = 1} \frac{1}{n} \sum_{i=1}^n (\mathbf{a}_i \cdot \mathbf{x})^4$ is a homogeneous degree-4 polynomial. It will be important for our

purposes that the polynomial in question have odd degree. Fortunately, we can reduce to this case.

Corollary VIII.2. *If there is a polynomial-time algorithm that given a homogeneous degree-5 polynomial p on $\mathbb{R}^d$ outputs a constant factor approximation to $\max_{\|\mathbf{x}\|_2=1} p(\mathbf{x})$, then there is a polynomial-time algorithm for the small-set expansion problem.*

Proof. We give a reduction to this problem from the problem in Theorem VIII.1. In particular, given $\mathbf{a}_1, \ldots, \mathbf{a}_n \in \mathbb{R}^d$, we let $q(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^n (\mathbf{a}_i \cdot \mathbf{x})^4$. We then define the homogeneous degree-5 polynomial p on $\mathbb{R}^{d+1}$ as $p(\mathbf{x}, y) = q(\mathbf{x})y$ (where x here represents the first d coordinates of the input and y represents the last one). We note that if $\|(\mathbf{x}, y)\|_2 = 1$, then $\|\mathbf{x}\|_2 = a$ and $y = b$ for some $a^2 + b^2 = 1$. Letting $\mathbf{x}' = \mathbf{x}/a$ and using the homogeneity of q, we have that $p(\mathbf{x}, y) = a^4 b q(\mathbf{x}')$. For fixed $\mathbf{x}'$, the maximum of this over a, b is obtained when $a = \sqrt{4/5}$ and $b = \sqrt{1/5}$. Thus, the maximum value of $p(\mathbf{x}, y)$ over the unit sphere equals the maximum value of $q(\mathbf{x}')$ over the unit sphere times $16/5^{2.5}$. Thus, finding a constant-factor approximation to the maximum value of one is equivalent to finding such an approximation of the other. This completes our proof. □

We are now ready to state our main theorem.

Theorem VIII.3 (Hardness of Proper Learning for LTFs)**.** *Suppose that there is an algorithm that given query access to a Boolean function f on $\mathbb{R}^d$ runs in $\mathrm{poly}(d)$ time and approximates the minimum misclassification error between f and a homogeneous LTF (with respect to the standard Gaussian distribution) to additive error ϵ for some $\epsilon < d^{-10}$. Then there is a polynomial-time algorithm for the small set expansion problem.*

Before we prove Theorem VIII.3, we note that any proper agnostic learner can be used to approximate this error merely by approximating the error between f and the learned function. Thus, this result will imply a lower bound for learning.

Proof. We assume throughout that d is sufficiently large, as otherwise there is nothing to prove. We proceed by a reduction from the problem in Corollary VIII.2. In particular, let p be a homogeneous degree-5 polynomial on $\mathbb{R}^d$. Let $\mathbf{T}$ be the unique symmetric tensor so that $p(\mathbf{x}) = \mathbf{T}(\mathbf{x}, \mathbf{x}, \mathbf{x}, \mathbf{x}, \mathbf{x})$. By scaling $\mathbf{T}$, we may assume that $\|\mathbf{T}\|_2 = 1$. Let $q(\mathbf{x}) = (\mathbf{T} \cdot H(\mathbf{x}))$, where $H(\mathbf{x})$ is the tensor whose entries are the degree-5 Hermite polynomials in $\mathbf{x}$.

Morally, we would like to take $f(\mathbf{x}) = q(\mathbf{x})$. Unfortunately, this does not work for two reasons.

First, $f(\mathbf{x})$ needs to be Boolean, while $q(\mathbf{x})$ distinctly is not. We can fix this by taking f to be a random function, where the expected value of $f(\mathbf{x})$ equals $q(\mathbf{x})$.

Unfortunately, this cannot work because the expected value of $f(\mathbf{x})$ must still be in $[-1, 1]$, while q is unbounded. To solve this, we first scale q down substantially and then truncate its extreme values. To do this, we define:

$$t(x) = \begin{cases} 1 & \text{if } x > 1 \\ -1 & \text{if } x < -1 \\ x & \text{otherwise.} \end{cases}$$

We then divide $\mathbb{R}^d$ into tiny boxes of side length δ for some very small δ. For each box B, we pick an $\mathbf{x} \in B$ and then (independently for each box) let f be 1 on B with probability $(t(q(\mathbf{x})/d) + 1)/2$ and -1 on B otherwise. We note that the expected value of f on B is $t(q(\mathbf{x})/d)$, where $\mathbf{x}$ is the representative element. As the difference between q at the representative element $\mathbf{x}$ of B and at any other point in B will be small if δ is (and if the box is not too far from the origin), it is not hard to see that the expectation over the randomness in defining f of $|\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[f(\mathbf{x})\mathrm{sign}(\mathbf{v} \cdot \mathbf{x})] - \mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[t(q(\mathbf{x})/d)\mathrm{sign}(\mathbf{v} \cdot \mathbf{x})]|$ goes to 0 with δ. As the variance of $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[f(\mathbf{x})\mathrm{sign}(\mathbf{v} \cdot \mathbf{x})]$ also goes to 0 with δ, if we take δ sufficiently small, then with high probability over the randomness in f, we have that $|\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[f(\mathbf{x})\mathrm{sign}(\mathbf{v}\cdot\mathbf{x})] - \mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[t(q(\mathbf{x})/d)\mathrm{sign}(\mathbf{v} \cdot \mathbf{x})]| < \epsilon/2$ for all unit vectors $\mathbf{v}$. Therefore, finding an ϵ additive approximation to the minimum misclassification error between f and an LTF is equivalent to finding a 2ϵ-additive approximation to the maximum value of $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[f(\mathbf{x})\mathrm{sign}(\mathbf{v} \cdot \mathbf{x})]$, which in turn is sufficient to find an ϵ-additive approximation of $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[t(q(\mathbf{x})/d)\mathrm{sign}(\mathbf{v} \cdot \mathbf{x})]$. We will show that this is computationally hard.

To start with, we note that $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[q(\mathbf{x})^2] = \|\mathbf{T}\|_2 = 1$. Therefore, by standard concentration bounds, we have that $\mathbf{Pr}_{\mathbf{x}\sim\mathcal{N}}[|q(\mathbf{x})| > d] = \exp(-\Omega(d^{2/5})) < \epsilon^3$. Therefore, by the Cauchy-Scwartz inequality, we have that

$$\begin{aligned}\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[|q(\mathbf{x})/d - t(q(\mathbf{x}))/d|] &\leq \sqrt{\mathbf{Pr}_{\mathbf{x}\sim\mathcal{N}}(|q(\mathbf{x})| > d)\ \mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[q(G\mathbf{x})^2]} \\ &\leq \epsilon/2\ .\end{aligned}$$

Thus, if one can approximate the maximum value of $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[t(q(\mathbf{x})/d)\mathrm{sign}(\mathbf{v} \cdot \mathbf{x})]$ to additive error ϵ, one can approximate the maximum value of $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[(q(\mathbf{x})/d)\mathrm{sign}(\mathbf{v} \cdot \mathbf{x})]$ to additive error $\epsilon/2$. However, we can compute this expectation by comparing the Hermite expansions for $q(\mathbf{x})/d$ and $\mathrm{sign}(\mathbf{v} \cdot \mathbf{x})$. In particular, the former only has non-vanishing terms in degree 5, where they are given by the tensor $\mathbf{T}/d$. The latter has its degree-5 Hermite tensor given by $c_5\mathbf{v}^{\otimes 5}$, where $c_5 = \mathbf{E}_{z\sim\mathcal{N}}[h_5(z)\mathrm{sign}(z)] = (3/2)\sqrt{1/(15\pi)}$. Therefore, we have that

$$\begin{aligned}\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[(q(\mathbf{x})/d)\mathrm{sign}(\mathbf{v} \cdot \mathbf{x})] &= (\mathbf{T}/d) \cdot (c_5\mathbf{v}^{\otimes 5}) \\ &= (c_5/d)\mathbf{T}(\mathbf{v}, \mathbf{v}, \mathbf{v}, \mathbf{v}, \mathbf{v}) \\ &= (c_5/d)p(\mathbf{v})\ .\end{aligned}$$

Thus, finding an $\epsilon/2$-additive approximation to the maximum value of $\mathbf{E}_{\mathbf{x}\sim\mathcal{N}}[(q(\mathbf{x})/d)\mathrm{sign}(\mathbf{v} \cdot \mathbf{x})]$ for unit vectors $\mathbf{v}$ is equivalent to finding an $O(d^{-9})$-additive approximation to the maximum value of $p(\mathbf{v})$ over unit vectors $\mathbf{v}$. We claim that doing this would give a constant-factor multiplicative approximation to the maximum value of $p(\mathbf{v})$, finishing our

reduction to the problem of Corollary VIII.2. To do this, we need to show that the maximum value of $p(\mathbf{v})$ is much larger than d^{-9}.

To show this, we note that because $\|\mathbf{T}\|_2 = 1$, the sum of the squares of the entries of $\mathbf{T}$ is 1. Since $\mathbf{T}$ has only d^5 entries, this means that it must have some entry with norm at least d^{-5}. Therefore, there must be unit vectors $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_5$ so that $\mathbf{T}(\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3, \mathbf{v}_4, \mathbf{v}_5) \geq d^{-5}$. However, this value is proportional to $\sum_{\epsilon_1,\ldots,\epsilon_5 \in \{\pm 1\}} \epsilon_1\epsilon_2 \cdots \epsilon_5 p(\epsilon_1\mathbf{v}_1 + \epsilon_2\mathbf{v}_2 + \ldots + \epsilon_5\mathbf{v}_5)$. As each term here is proportional to p of some unit vector (using the fact that p is homogeneous), this implies that there is some unit vector $\mathbf{v}$ with $|p(\mathbf{v})| \gg d^{-5}$. Replacing $\mathbf{v}$ by its negation if necessary, we have that the maximum value of $p(\mathbf{v})$ over unit vectors $\mathbf{v}$ is $\Omega(d^{-5})$. This completes our proof. □

Theorem VIII.4 (Hardness of Proper Learning for ReLUs)**.** *Suppose that there is an algorithm that given query access to a real-valued function f on $\mathbb{R}^d$ runs in $\mathrm{poly}(d)$ time and approximates the minimum L_2^2 error between f and a homogeneous ReLU (with respect to the standard Gaussian distribution) to additive error ϵ for some $\epsilon < d^{-4}$. Then there is a polynomial-time algorithm for the small set expansion problem.*

The proof of Theorem VIII.4 can be found in the full version of the paper.

ACKNOWLEDGMENT

I. D. supported by NSF Medium Award CCF-2107079, NSF Award CCF-1652862 (CAREER), and a DARPA Learning with Less Labels (LwLL) grant. D. K supported by NSF Medium Award CCF-2107547 and NSF Award CCF-1553288 (CAREER). V. K. supported in part by NSF Award CCF-2144298 (CAREER). C. T. supported by NSF Award CCF-2144298 (CAREER). N. Z. supported by NSF Medium Award CCF-2107079, and a DARPA Learning with Less Labels (LwLL) grant.

REFERENCES

[1] S. Chen, A. R. Klivans, and R. Meka, "Learning deep relu networks is fixed-parameter tractable," in *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, 2022.

[2] V. Feldman, "On the power of membership queries in agnostic learning," in *21st Annual Conference on Learning Theory - COLT 2008*, 2008, pp. 147–156. [Online]. Available: http://colt2008.cs.helsinki.fi/papers/129-Feldman.pdf

[3] L. Valiant, "A theory of the learnable," *Communications of the ACM*, vol. 27, no. 11, pp. 1134–1142, 1984.

[4] L. G. Valiant, "A theory of the learnable," in *Proc. 16th Annual ACM Symposium on Theory of Computing (STOC)*. ACM Press, 1984, pp. 436–445.

[5] D. Angluin, "Learning Regular Sets from Queries and Counterexamples," *Information and Computation*, vol. 75, no. 2, pp. 87–106, 1987.

[6] O. Goldreich and L. Levin, "A hard-core predicate for all one-way functions," in *Proceedings of the Twenty-First Annual Symposium on Theory of Computing*, Seattle, Washington, 1989, pp. 25–32.

[7] E. Kushilevitz and Y. Mansour, "Learning decision trees using the Fourier spectrum," *SIAM J. on Computing*, vol. 22, no. 6, pp. 1331–1348, Dec. 1993.

[8] J. Jackson, "An efficient membership-query algorithm for learning DNF with respect to the uniform distribution," *Journal of Computer and System Sciences*, vol. 55, no. 3, pp. 414–440, 1997.

[9] P. Gopalan, A. Kalai, and A. Klivans, "Agnostically learning decision trees," in *Proc. 40th Annual ACM Symposium on Theory of Computing (STOC)*, 2008, pp. 527–536.

[10] G. Blanc, J. Lange, M. Qiao, and L. Tan, "Properly learning decision trees in almost polynomial time," *J. ACM*, vol. 69, no. 6, pp. 39:1–39:19, 2022.

[11] F. Tramèr, F. Zhang, A. Juels, M. K. Reiter, and T. Ristenpart, "Stealing machine learning models via prediction apis," in *25th USENIX Security Symposium, USENIX Security 16, 2016*. USENIX Association, 2016, pp. 601–618.

[12] Y. Shi, Y. Sagduyu, and A. Grushin, "How to steal a machine learning classifier with deep learning," in *2017 IEEE International Symposium on Technologies for Homeland Security (HST)*, 2017, pp. 1–5.

[13] N. Papernot, P. D. McDaniel, I. J. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security, AsiaCCS 2017, Abu Dhabi, United Arab Emirates, April 2-6, 2017*. ACM, 2017, pp. 506–519.

[14] S. Milli, L. Schmidt, A. D. Dragan, and M. Hardt, "Model reconstruction from model explanations," in *Proceedings of the Conference on Fairness, Accountability, and Transparency, FAT 2019, Atlanta, GA, USA, 2019*. ACM, 2019, pp. 1–9.

[15] M. Jagielski, N. Carlini, D. Berthelot, A. Kurakin, and N. Papernot, "High accuracy and high fidelity extraction of neural networks," in *29th USENIX Security Symposium, USENIX Security 2020, 2020*. USENIX Association, 2020, pp. 1345–1362.

[16] D. Rolnick and K. P. Kording, "Reverse-engineering deep ReLU networks," in *Proceedings of the 37th International Conference on Machine Learning, ICML 2020*, ser. Proceedings of Machine Learning Research, vol. 119. PMLR, 2020, pp. 8178–8187.

[17] R. Jayaram, D. Woodruff, and Q. Zhang, "Span recovery for deep neural networks with applications to input obfuscation," in *8th International Conference on Learning Representations, ICLR 2020*, 2020.

[18] S. Chen, A. R. Klivans, and R. Meka, "Efficiently learning one hidden layer relu networks from queries," in *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021*, 2021, pp. 24 087–24 098.

[19] A. Daniely and E. Granot, "An exact poly-time membership-queries algorithm for extracting a three-layer relu network," in *The Eleventh International Conference on Learning Representations*, 2022.

[20] J. H. Friedman, M. Jacobson, and W. Stuetzle, "Projection Pursuit Regression," *J. Am. Statist. Assoc.*, vol. 76, p. 817, 1981.

[21] P. J. Huber, "Projection Pursuit," *The Annals of Statistics*, vol. 13, no. 2, pp. 435 – 475, 1985.

[22] K.-C. Li, "Sliced inverse regression for dimension reduction," *Journal of the American Statistical Association*, vol. 86, no. 414, pp. 316–327, 1991.

[23] P. Hall and K.-C. Li, "On almost Linearity of Low Dimensional Projections from High Dimensional Data," *The Annals of Statistics*, vol. 21, no. 2, pp. 867 – 889, 1993.

[24] Y. Xia, H. Tong, W. K. Li, and L. Zhu, "An adaptive estimation of dimension reduction space," *Journal of the Royal Statistical Society Series B: Statistical Methodology*, vol. 64, no. 3, pp. 363–410, 2002.

[25] Y. Xia, "A multiple-index model and dimension reduction," *Journal of the American Statistical Association*, vol. 103, no. 484, pp. 1631–1640, 2008.

[26] H. Ichimura, "Semiparametric least squares (sls) and weighted sls estimation of single-index models," *Journal of econometrics*, vol. 58, no. 1-2, pp. 71–120, 1993.

[27] M. Hristache, A. Juditsky, and V. Spokoiny, "Direct estimation of the index coefficient in a single-index model," *Annals of Statistics*, pp. 595–623, 2001.

[28] W. Härdle, M. Müller, S. Sperlich, A. Werwatz *et al.*, *Nonparametric and semiparametric models*. Springer, 2004, vol. 1.

[29] A. S. Dalalyan, A. Juditsky, and V. Spokoiny, "A new algorithm for estimating the effective dimension-reduction subspace," *The Journal of Machine Learning Research*, vol. 9, pp. 1647–1678, 2008.

[30] Y. Plan and R. Vershynin, "One-bit compressed sensing by linear programming," *Communications on pure and Applied Mathematics*, vol. 66, no. 8, pp. 1275–1297, 2013.

[31] S. Gopi, P. Netrapalli, P.and Jain, and A. Nori, "One-bit compressed sensing: Provable support and vector recovery," in *Proceedings of the 30th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research. PMLR, 2013, pp. 154–162.

[32] R. Rubinfeld, "Sublinear time algorithms," in *Proceedings of the international congress of mathematicians (ICM), Madrid, Spain, August 22–30, 2006. Volume III: Invited lectures*, 2006.

[33] E. Candes, X. Li, and M. Soltanolkotabi, "Phase retrieval via wirtinger flow: Theory and algorithms," *IEEE Transactions on Information Theory*, vol. 61, no. 4, pp. 1985–2007, 2015.

[34] P. Netrapalli, P. Jain, and S. Sanghavi, "Phase retrieval using alternating minimization," *Advances in Neural Information Processing Systems*, vol. 26, 2013.

[35] S. Vempala, "A random sampling based algorithm for learning the intersection of halfspaces," in *Proc. 38th IEEE Symposium on Foundations of Computer Science (FOCS)*, 1997, pp. 508–513.

[36] R. Arriaga and S. Vempala, "An algorithmic theory of learning: Robust concepts and random projection," in *Proceedings of the 40th Annual Symposium on Foundations of Computer Science (FOCS)*, New York, NY, 1999, pp. 616–623.

[37] A. Klivans, R. O'Donnell, and R. Servedio, "Learning intersections and thresholds of halfspaces," *Journal of Computer & System Sciences*, vol. 68, no. 4, pp. 808–840, 2004.

[38] A. R. Klivans, P. M. Long, and A. K. Tang, "Baum's algorithm learns intersections of halfspaces with respect to log-concave distributions," in *13th International Workshop, RANDOM 2009*, 2009, pp. 588–600.

[39] S. Vempala, "Learning convex concepts from gaussian distributions with PCA," in *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS*, 2010, pp. 124–130.

[40] S. Vempala and Y. Xiao, "Structure from local optima: Learning subspace juntas via higher order pca," in *STOC*, 2012.

[41] M. Janzamin, H. Sedghi, and A. Anandkumar, "Beating the perils of non-convexity: Guaranteed training of neural networks using tensor methods," *arXiv preprint arXiv:1506.08473*, 2015.

[42] R. Ge, J. D. Lee, and T. Ma, "Learning one-hidden-layer neural networks with landscape design," in *6th International Conference on Learning Representations, ICLR 2018*, 2018.

[43] R. Dudeja and D. Hsu, "Learning single-index models in gaussian space," in *Conference On Learning Theory, COLT 2018, Stockholm, Sweden, 6-9 July 2018*, ser. Proceedings of Machine Learning Research, vol. 75. PMLR, 2018, pp. 1887–1930. [Online]. Available: http://proceedings.mlr.press/v75/dudeja18a.html

[44] A. Bakshi, R. Jayaram, and D. P. Woodruff, "Learning two layer rectified neural networks in polynomial time," in *Conference on Learning Theory, COLT 2019*, 2019.

[45] R. Ge, R. Kuditipudi, Z. Li, and X. Wang, "Learning two-layer neural networks with symmetric inputs," in *7th International Conference on Learning Representations, ICLR 2019*, 2019.

[46] I. Diakonikolas, D. M. Kane, V. Kontonis, and N. Zarifis, "Algorithms and SQ lower bounds for pac learning one-hidden-layer ReLU networks," in *Conference on Learning Theory, COLT*. PMLR, 2020, pp. 1514–1539.

[47] S. Chen and R. Meka, "Learning polynomials in few relevant dimensions," in *Conference on Learning Theory, COLT 2020, 9-12 July 2020, Virtual Event [Graz, Austria]*, ser. Proceedings of Machine Learning Research, vol. 125. PMLR, 2020, pp. 1161–1227. [Online]. Available: http://proceedings.mlr.press/v125/chen20a.html

[48] A. Damian, J. Lee, and M. Soltanolkotabi, "Neural networks can learn representations with gradient descent," in *Conference on Learning Theory*. PMLR, 2022, pp. 5413–5452.

[49] A. Bietti, J. Bruna, C. Sanford, and M. J. Song, "Learning single-index models with shallow neural networks," *Advances in Neural Information Processing Systems*, vol. 35, pp. 9768–9783, 2022.

[50] S. Chen, Z. Dou, S. Goel, A. R. Klivans, and R. Meka, "Learning narrow one-hidden-layer relu networks," in *The Thirty Sixth Annual Conference on Learning Theory, COLT 2023*, ser. Proceedings of Machine Learning Research, vol. 195. PMLR, 2023, pp. 5580–5614. [Online]. Available: https://proceedings.mlr.press/v195/chen23a.html

[51] S. Chen and S. Narayanan, "A faster and simpler algorithm for learning shallow networks," *CoRR*, vol. abs/2307.12496, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2307.12496

[52] I. Diakonikolas and D. M. Kane, "Efficiently learning one-hidden-layer relu networks via schur polynomials," *CoRR*, vol. abs/2307.12840, 2023. [Online]. Available: https://doi.org/10.48550/arXiv.2307.12840

[53] D. Haussler, "Decision theoretic generalizations of the PAC model for neural net and other learning applications," *Information and Computation*, vol. 100, pp. 78–150, 1992.

[54] M. Kearns, R. Schapire, and L. Sellie, "Toward Efficient Agnostic Learning," *Machine Learning*, vol. 17, no. 2/3, pp. 115–141, 1994.

[55] A. Daniely, "Complexity theoretic limitations on learning halfspaces," in *Proceedings of the 48th Annual Symposium on Theory of Computing, STOC 2016*, 2016, pp. 105–117.

[56] I. Diakonikolas, D. Kane, P. Manurangsi, and L. Ren, "Hardness of learning a single neuron with adversarial label noise," in *Proceedings of the 25th International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2022.

[57] I. Diakonikolas, D. M. Kane, P. Manurangsi, and L. Ren, "Cryptographic hardness of learning halfspaces with massart noise," *CoRR*, vol. abs/2207.14266, 2022, conference version in NeurIPS'22. [Online]. Available: https://doi.org/10.48550/arXiv.2207.14266

[58] S. Tiegel, "Hardness of agnostically learning halfspaces from worst-case lattice problems," in *COLT*, 2023.

[59] P. Gopalan, A. Kalai, and A. R. Klivans, "A query algorithm for agnostically learning dnf?" in *21st Annual Conference on Learning Theory - COLT 2008, Helsinki, Finland, July 9-12, 2008*, 2008, pp. 515–516. [Online]. Available: http://colt2008.cs.helsinki.fi/papers/Gopalan-open-question.pdf

[60] A. B. Tsybakov, *Introduction to Nonparametric Estimation*. Springer Publishing Company, Incorporated, 2008.

[61] S. Vempala, "A random-sampling-based algorithm for learning intersections of halfspaces," *J. ACM*, vol. 57, no. 6, pp. 32:1–32:14, 2010.

[62] I. Diakonikolas, D. M. Kane, T. Pittas, and N. Zarifis, "The optimality of polynomial regression for agnostic learning under gaussian marginals in the sq model," in *Proceedings of The 34th Conference on Learning Theory, COLT*, 2021.

[63] I. Diakonikolas, D. M. Kane, and L. Ren, "Near-optimal cryptographic hardness of agnostically learning halfspaces and relu regression under gaussian marginals," in *ICML*, 2023.

[64] S. Goel, A. Gollakota, and A. R. Klivans, "Statistical-query lower bounds via functional gradients," in *Advances in Neural Information Processing Systems, NeurIPS*, 2020.

[65] I. Diakonikolas, D. M. Kane, and N. Zarifis, "Near-optimal SQ lower bounds for agnostically learning halfspaces and ReLUs under Gaussian marginals," in *Advances in Neural Information Processing Systems, NeurIPS*, 2020.

[66] A. T. Kalai and R. Sastry, "The isotron algorithm: High-dimensional isotonic regression." in *COLT*. Citeseer, 2009.

[67] S. M. Kakade, V. Kanade, O. Shamir, and A. Kalai, "Efficient learning of generalized linear and single index models with isotonic regression," *Advances in Neural Information Processing Systems*, vol. 24, 2011.

[68] A. Gollakota, P. Gopalan, A. R. Klivans, and K. Stavropoulos, "Agnostically learning single-index models using omnipredictors," *arXiv preprint arXiv:2306.10615*, 2023.

[69] I. Diakonikolas and D. M. Kane, "Small covers for near-zero sets of polynomials and learning latent variable models," in *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2020, pp. 184–195.

[70] P. Raghavendra and D. Steurer, "Graph expansion and the unique games conjecture," in *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010*. ACM, 2010, pp. 755–764.

[71] A. Klivans, R. O'Donnell, and R. Servedio, "Learning geometric concepts via Gaussian surface area," in *Proc. 49th IEEE Symposium on Foundations of Computer Science (FOCS)*, Philadelphia, Pennsylvania, 2008, pp. 541–550.

[72] D. M. Kane, "The gaussian surface area and noise sensitivity of degree-d polynomial threshold functions," *Computational Complexity*, vol. 20, no. 2, pp. 389–412, 2011.

[73] J. Neeman, "Testing surface area with arbitrary accuracy," in *Symposium on Theory of Computing, STOC 2014, 2014*. ACM, 2014, pp. 393–397.

[74] V. Kontonis, C. Tzamos, and M. Zampetakis, "Efficient truncated statistics with unknown truncation," in *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE, 2019, pp. 1578–1595.

[75] A. De, E. Mossel, and J. Neeman, "Robust testing of low dimensional functions," in *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing*. ACM, 2021, pp. 584–597.

[76] A. Kalai, A. Klivans, Y. Mansour, and R. Servedio, "Agnostically learning halfspaces," *SIAM Journal on Computing*, vol. 37, no. 6, pp. 1777–1805, 2008, special issue for FOCS 2005.

[77] F. Rosenblatt, "The Perceptron: a probabilistic model for information storage and organization in the brain," *Psychological Review*, vol. 65, pp. 386–407, 1958.
[78] I. Diakonikolas, D. M. Kane, and J. Nelson, "Bounded independence fools degree-2 threshold functions," in *FOCS*, 2010, pp. 11–20.
[79] V. Chernozhukov, D. Chetverikov, and K. Kato, "Detailed proof of Nazarov's inequality," *arXiv preprint arXiv:1711.10696*, 2017.
[80] D. J. Hsu, C. Sanford, R. A. Servedio, and E. Vlatakis-Gkaragkounis, "Near-optimal statistical query lower bounds for agnostically learning intersections of halfspaces with gaussian marginals," in *Conference on Learning Theory*, ser. Proceedings of Machine Learning Research, vol. 178. PMLR, 2022, pp. 283–312.
[81] I. Diakonikolas, P. Harsha, A. Klivans, R. Meka, P. Raghavendra, R. A. Servedio, and L. Y. Tan, "Bounding the average sensitivity and noise sensitivity of polynomial threshold functions," in *STOC*, 2010, pp. 533–542.
[82] I. Diakonikolas, R. Servedio, L.-Y. Tan, and A. Wan, "A regularity lemma, and low-weight approximators, for low-degree polynomial threshold functions," in *CCC*, 2010, pp. 211–222.
[83] I. Diakonikolas, P. Raghavendra, R. A. Servedio, and L. Y. Tan, "Average sensitivity and noise sensitivity of polynomial threshold functions," *SIAM J. Comput.*, vol. 43, no. 1, pp. 231–253, 2014.
[84] I. Diakonikolas, D. M. Kane, V. Kontonis, C. Tzamos, and N. Zarifis, "Agnostic proper learning of halfspaces under gaussian marginals," in *Proceedings of The 34th Conference on Learning Theory, COLT*, 2021.
[85] Y. Nesterov and V. Spokoiny, "Random gradient-free minimization of convex functions," *Foundations of Computational Mathematics*, vol. 17, pp. 527–566, 2017.
[86] M. Ledoux, "Semigroup proofs of the isoperimetric inequality in Euclidean and Gauss space," *Bull. Sci. Math.*, vol. 118, pp. 485–510, 1994.
[87] G. Pisier, "Probabilistic methods in the geometry of Banach spaces," in *Lecture notes in Math.* Springer, 1986, pp. 167–241.
[88] S. Mukherjee and Q. Wu, "Estimation of gradients and coordinate covariation in classification," *The Journal of Machine Learning Research*, vol. 7, pp. 2481–2514, 2006.
[89] S. Mukherjee, D. Zhou, and J. Shawe-Taylor, "Learning coordinate covariances via gradients." *Journal of Machine Learning Research*, vol. 7, no. 3, 2006.
[90] Q. Wu, J. Guinney, M. Maggioni, and S. Mukherjee, "Learning gradients: predictive models that infer geometry and statistical dependence," *Journal of Machine Learning Research*, vol. 11, pp. 2175–2198, 2010.
[91] V. Bogachev, *Gaussian measures*. Mathematical surveys and monographs, vol. 62, 1998.
[92] M. Ledoux, "Semigroup proofs of the isoperimetric inequality in euclidean and gauss space," *Bulletin des sciences mathématiques*, vol. 118, no. 6, pp. 485–510, 1994.
[93] B. Barak, F. G. S. L. Brandão, A. W. Harrow, J. A. Kelner, D. Steurer, and Y. Zhou, "Hypercontractivity, sum-of-squares proofs, and their applications," in *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, 2012*. ACM, 2012, pp. 307–326.