

Robust Learning of Multi-Index Models under Logconcave Distributions via Weak Partial Learners

Ilias Diakonikolas*

University of Wisconsin-Madison
ilias@cs.wisc.edu

Daniel M. Kane†

University of California, San Diego
dakane@cs.ucsd.edu

Mingchen Ma‡

University of Wisconsin-Madison
mingchen@cs.wisc.edu

October 25, 2026

Abstract

We develop a new algorithmic framework for learning Boolean concept classes under arbitrary logconcave distributions. Our framework applies to broad families of multi-index models and yields significantly faster robust learners than were previously known in these settings. It also extends naturally to the settings of partial classification and testable learning. As consequences, we obtain substantially improved learners for several fundamental concept classes, including functions of halfspaces and multiclass linear classifiers, under logconcave marginals.

Our starting point is a conceptually new boosting framework based on *weak partial learners*. Classical boosting-based approaches either require distribution-free weak learners, which are often computationally intractable, or yield only weak approximation guarantees. We show instead that it suffices to design weak learners for *partial* classification problems, where labels may be undefined on a nontrivial fraction of the domain. This provides a robust and computationally tractable route to strong agnostic learning guarantees. The main technical challenge is that boosting requires a weak learner that succeeds under conditionings of the initial distribution that are not guaranteed to be logconcave. To address this, we develop a recursive moment-based partitioning procedure: given a logconcave distribution X and a conditional version X' , we partition the space into polynomially many convex regions so that, on most regions, the low-degree central moments of X and X' approximately agree. We show that in any good region, a well-behaved multi-index model will correlate with a low-degree polynomial, which provides a weak hypothesis on that region. Combining these weak hypotheses over the regions yields the required weak learner.

A key ingredient of our approach is an efficient method for detecting moment discrepancy. Specifically, we show how to find a low-dimensional subspace that captures any non-trivial discrepancy between X and X' , if one exists. Our approach combines spectral analysis of moment tensors with recent progress on isoperimetry for logconcave measures. This discrepancy-finding procedure may be of independent interest.

*Supported by NSF Medium Award CCF-2107079, ONR Award number N00014-25-1-2268, and a Guggenheim Fellowship.

†Supported by NSF Medium Award CCF-2107547.

‡Supported by NSF Award CCF-2144298 (CAREER).

1 Introduction

The main motivation of this work was to develop efficient learners for fundamental classes of Boolean functions—including multiclass linear classifiers and intersections of halfspaces—under mild distributional assumptions and potentially in the presence of adversarial label noise. In the process, we developed a new general learning algorithm that is both label noise-tolerant and significantly more efficient than state-of-the-art. Before we describe our contributions in detail, we start with some background on specific settings that motivated this work.

Multi-Index Models and Efficient Learnability A Boolean function is called a Multi-Index Model (MIM) if it depends only on the projection of its inputs onto a “low-dimensional” subspace. Formally, we have:

Definition 1.1 (Multi-Index Model). *A function $f : \mathbb{R}^d \rightarrow \mathcal{Y}$ is called a K -MIM if there exists a subspace $U \subseteq \mathbb{R}^d$ of dimension at most K such that f depends only on the projection onto U , i.e., $f(\mathbf{x}) = f(\text{proj}_U \mathbf{x})$, where $\text{proj}_U \mathbf{x}$ is the projection of $\mathbf{x}$ on U . Equivalently, f is of the form $f(\mathbf{x}) = g(\mathbf{W}\mathbf{x})$ for a link function $g : \mathbb{R}^K \rightarrow \mathcal{Y}$ and a $K \times d$ weight matrix $\mathbf{W}$.*

In several settings of interest, the dimension K of the hidden subspace is significantly smaller than the ambient dimension d and is often thought of as bounded above by a constant. Throughout this work, we focus on the case that the range of our functions is discrete, in most cases Boolean.

Multi-index models [FJS81, Hub85, Li91, HL93, XTLZ02, Xia08] are motivated by the typical assumption that real high-dimensional data are inherently structured, specifically that the relationship between the features is of a lower-dimensional nature. MIMs capture several well-studied concept classes, both discrete and real-valued, including halfspaces, functions of halfspaces, multiclass linear classifiers, low-dimensional polynomial threshold functions, and natural classes of neural networks. The efficient learnability of each of these concept classes has been a topic of intense investigation in the theoretical computer science and machine learning communities over several decades. More recently, there has been a flurry of research activity aiming at developing general algorithmic techniques that are applicable to broad families of MIMs—albeit under strong distributional assumptions. This progress notwithstanding, our understanding of the complexity of learning MIMs (both for specific special cases and in more general settings) remains limited. This motivates the general research direction explored in this work:

Can we develop efficient and noise-tolerant learners for broad classes of MIMs under mild distributional assumptions?

Our main result answers this question in the affirmative, when the underlying distribution is logconcave and the labels are adversarially corrupted. Additionally, our algorithmic results extend to *partial* classification and to the *testable learning* framework.

For concreteness, we define agnostic PAC learning under logconcave distributions.

Definition 1.2 (Agnostic Classification under Logconcave Marginals). *Let $\mathcal{Y}$ be finite and $\mathcal{H}$ be a class of functions from $\mathbb{R}^d$ to $\mathcal{Y}$. Given i.i.d. samples $(\mathbf{x}, y)$ from a distribution D on $\mathbb{R}^d \times \mathcal{Y}$ such that (a) the $\mathbf{x}$ -marginal, $D_{\mathbf{x}}$, of D is an unknown logconcave distribution, and (b) no assumptions are made on the labels, the goal is to output a hypothesis $h : \mathbb{R}^d \rightarrow \mathcal{Y}$ that, with high probability, has small 0-1 error $\text{err}_D(h) := \Pr_{(\mathbf{x}, y) \sim D}[h(\mathbf{x}) \neq y]$, as compared to the optimal 0-1 error of the class $\mathcal{H}$, i.e., $\text{opt}_D(\mathcal{H}) := \inf_{f \in \mathcal{H}} \text{err}_D(f)$.*

To demonstrate the limitations of current techniques, we next discuss the state-of-the-art for three basic learning tasks.

Example 1: Robustly Learning Halfspaces A halfspace is any function $h : \mathbb{R}^d \rightarrow \{\pm 1\}$ of the form $h(\mathbf{x}) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + t)$, where $\mathbf{w} \in \mathbb{R}^d$ and $t \in \mathbb{R}$. The task of learning halfspaces is one of the most extensively studied problems in machine learning [Ros58, Nov62, MP68, FS97, Vap98, STC00]. While halfspaces are efficiently learnable in the distribution-free (realizable) PAC model [Val84] (see, e.g., [MT94]), the algorithmic problem becomes computationally intractable in the presence of adversarial label noise [Dan16, Tie23]. This has motivated a long line of work on developing efficient robust learners under natural distributional assumptions. For the special case of *homogeneous* halfspaces (corresponding to zero threshold), [ABL17] gave a $\text{poly}(d/\epsilon)$ time learner with error $O(\text{opt}) + \epsilon$ under logconcave marginals, where opt is the optimal error in the class¹. The efficient learnability of *general* (i.e., not necessarily homogeneous) halfspaces under logconcave marginals has remained open since. For the special case of Gaussian marginals, [DKS18, DKTZ22, DKM24] gave an efficient constant-factor approximation. However, for the general logconcave case, the best known efficient algorithm achieves error $\tilde{\Omega}(\text{opt}^{1/2}) + \epsilon$; see, e.g., [DKS18]. On the other hand, the L_1 -polynomial regression algorithm [KKMS08] achieves optimal error of $\text{opt} + \epsilon$ with sample and computational complexity $d^{2^{\text{poly}(1/\epsilon)}}$. Concretely, this leads to the following open question.

Open Question 1. *Is there a robust learner for general halfspaces under logconcave marginals with error $O(\text{opt}) + \epsilon$ beating the complexity of polynomial regression?*

As a corollary of our general algorithmic approach, we answer this question in the affirmative (see Theorem 1.4).

Example 2: Learning Intersections of Halfspaces An intersection of K halfspaces is any function $f : \mathbb{R}^d \rightarrow \{\pm 1\}$ for which there exist K halfspaces $h_i(\mathbf{x}) = \text{sign}(\mathbf{w}^{(i)} \cdot \mathbf{x} + t_i)$ such that $f(\mathbf{x}) = 1$ if and only if $h_i(\mathbf{x}) = 1$ for all $i \in [K]$. In sharp contrast to the special case of a single halfspace, our understanding of the efficient learnability of this class remains limited—even for $K = 2$ and in the presence of clean labels. Specifically, even for this special case, it remains a major open question whether efficient learning is possible in the distribution-free setting. The best known distribution-free algorithm requires sub-exponential time [APS26] and prior work has given plausible evidence of hardness; see, e.g., [KS08, She09]. On the other hand, for $K = O(1)$, efficient learners have been developed with respect to natural distributions, including the Gaussian and the uniform distribution on the hypercube; see, e.g., [BK97, KOS04, AV06, KS04, KLT09, Vem10b, Vem10a, DMRT25a]. One of the strongest known results [Vem10b] (see also [KLT09]) gives an efficient (realizable) algorithm for any constant number of halfspaces under logconcave distributions *assuming all component halfspaces pass through the mean of the distribution*². The only known upper bound for the general case—even for clean labels—is the L_1 -polynomial regression method, which incurs sample and computational complexity $d^{2^{\text{poly}(K/\epsilon)}}$. Concretely, this leads to the following open question.

Open Question 2. *Is there a learner for intersections of two halfspaces under logconcave marginals beating the complexity of polynomial regression?*

As a corollary of our general approach, we answer this question in the affirmative (see Theorem 1.5). As a bonus, our algorithm succeeds for any function of $K = O(1)$ many halfspaces, and is robust to adversarial label noise (achieving *any* desired constant factor approximation).

¹A constant-factor approximation is best possible for fully-polynomial time algorithms, due to known intractability results even for Gaussian marginals [DKPZ21, DKR23].

²This is an important assumption without which their algorithmic approach fails.

Example 3: Robust Multiclass Linear Classification A Multiclass Linear Classifier (MLC) is any function $f : \mathbb{R}^d \rightarrow [K]$ of the form $f(\mathbf{x}) = \operatorname{argmax}_{i \in [K]} (\mathbf{w}^{(i)} \cdot \mathbf{x} + t_i)$, where $\mathbf{w}^{(i)} \in \mathbb{R}^d$ and $t_i \in \mathbb{R}$. Multiclass linear classification (MLC) is a textbook machine learning problem [SB14] generalizing the task of learning halfspaces (corresponding to $K = 2$). In the realizable PAC model, this task is efficiently solvable to any desired error ϵ with sample size $O(dK/\epsilon)$ via linear programming. On the other hand, despite extensive interest, see, e.g., [NDRT13, WLT18, LWS18, SKP⁺20, GWBL20, DMRT25b, DIM26], our understanding of the computational complexity of learning remains limited in the presence of adversarial label noise. The recent result [DIKZ25] gave an algorithm that succeeds under the Gaussian distribution, has complexity $\operatorname{poly}(d)2^{\operatorname{poly}(K/\epsilon)}$, and guarantees error of $C\operatorname{opt} + \epsilon$, for some universal constant $C > 1$. Note that the Gaussianity assumption is crucial. Moreover, the underlying complexity scales exponentially with both K and $1/\epsilon$. As in previous examples, for logconcave distributions, the only known algorithm is L_1 -regression which incurs complexity $d^{\operatorname{poly}(K/\epsilon)}$ even under the Gaussian distribution. Concretely, this leads to the following open question.

Open Question 3. *Is there a robust learner for K -MLC under logconcave marginals beating the complexity of polynomial regression?*

We answer this question in the affirmative (see Theorem 1.6). Specifically, we give two algorithms for this task: the first algorithm achieves error $\tilde{O}(K\operatorname{opt}) + \epsilon$ and runs in $\operatorname{poly}(d, K, 1/\epsilon)$ time, and the second achieves error $O(K)\operatorname{opt} + \epsilon$ with sample complexity $\operatorname{poly}(d, K, 1/\epsilon)$ and running time $\operatorname{poly}(d)2^{\operatorname{polylog}(K/\epsilon)}$.

1.1 Our Results

Before we state our algorithmic contributions in detail, we emphasize that all our upper bounds in this work are obtained via a *single unified algorithm*. In Section 1.1.1, we describe our general algorithmic result that is applicable to a broad family of MIMs, and its testable learning extension. In Section 1.1.2, we describe some concrete algorithmic implications for fundamental concept classes.

1.1.1 General Algorithmic Result

Our algorithm can be used to learn essentially any non-pathological K -MIM, although its sample and computational complexities will depend on some complexity measure of the underlying concept class. In particular, we can learn any function that admits L_1 -sandwiching polynomials (see Definition 7.8) with respect to any logconcave distribution. Results of [KKM13] imply that this applies to any function with bounded effective surface area.

Theorem 1.3 (Robust Learner for General MIMs, Informal). *Let $\mathcal{H}$ be a family of K -MIMs over $\mathbb{R}^d$, admitting L_1 -sandwiching polynomials achieving error $c > 0$ with degree $\ell = \ell(c)$. There is an algorithm such that for any distribution D over $\mathbb{R}^d \times \{\pm 1\}$ with $D_{\mathbf{x}}$ logconcave, and any $\epsilon \in (0, 1)$, it has the following behavior: the algorithm draws $\operatorname{poly}_{\ell}(d)\operatorname{poly}_{K,c}(1/\epsilon)$ samples from D , it runs in time $\operatorname{poly}_{\ell}(d)2^{\operatorname{polylog}_{K,c}(1/\epsilon)}$, and with high probability outputs a hypothesis $\hat{h}$ such that $\operatorname{err}_D(\hat{h}) \leq (1 + O(c))\operatorname{opt}_D(\mathcal{H}) + \epsilon$.*

The reader is referred to Theorem 7.9 in Section 7.3 for the detailed formal statement that also covers the partial learning setting. For our other generic families of functions covered by our main result, see Theorem 7.4 in Section 7.2.

Discussion We now provide a few remarks regarding Theorem 1.3. First, we note that our general MIM family includes a range of well-studied concept classes, including arbitrary functions of K halfspaces, K -dimensional constant degree PTFs, and others. Specifically, all of these classes admit sandwiching polynomials whose degree ℓ is only a function of K and the desired accuracy. Therefore, as long as $K = O(1)$, Theorem 1.3 implies a constant-factor robust learner for each of these classes with sample complexity $\text{poly}(d/\epsilon)$ and computational complexity $\text{poly}(d)2^{\text{polylog}(1/\epsilon)}$.

An additional feature of our result is that (for concept classes with the L_1 - sandwiching property) we can obtain *any* desired constant factor approximation by increasing the complexity by a polynomial factor. In particular, this allows us to obtain a Quasi-PTAS for the agnostic learning task (with quasi-polynomial dependence on ϵ in the runtime). It is also worth noting that our general algorithm is implementable in the Statistical Query (SQ) model [Kea98] with complexity (i.e., number of queries and inverse of accuracy) $\text{poly}_K(d/\epsilon)$. The quasi-polynomial dependence on ϵ in the runtime is due to a brute-force search over a space of dimension $\text{polylog}_K(1/\epsilon)$.

Extension to Testable Learning Interestingly, the algorithm establishing Theorem 1.3 also applies to the testable learning framework of [RV23], as long as the marginal logconcave distribution is known to the algorithm. In testable learning, the goal is to be able to test whether the underlying distributional assumptions for your algorithm are met (or met well enough) rather than having to assume them. In particular, a testable learner should consist of (see Definition 6.1) a tester-learner pair satisfying the following properties: (a) if the tester accepts, the learner outputs a hypothesis and a certificate that it achieves near-optimal error, and (b) it is highly unlikely that the tester rejects if the data satisfies the underlying distributional assumptions. The reader is referred to Theorem 6.2 in Section 6 for our generic testable learning result. This result provides significantly more efficient testable learners for a range of concept classes with near-optimal error guarantees.

Extension to Partial Learning Furthermore, our results also apply to agnostic partial learning, a stronger but natural extension of the standard agnostic learning setting. In agnostic partial learning, we receive samples $(\mathbf{x}, y)$ from a distribution D over $\mathbb{R}^d \times \{0, \pm 1\}$, with logconcave marginal, where a sample with $y = 0$ can be thought of as an example that goes unclassified. The goal of an agnostic partial learner is to output a hypothesis that does well over the examples that are classified in our data. In other words, even if only a small fraction of the examples are classified (whose marginal is logconcave conditioned on an event with a suitably small probability), our algorithm is still able to learn well with the same complexity.

1.1.2 Algorithmic Applications for Concrete Concept Classes

Our first algorithmic application is for the task of robustly learning general halfspaces under logconcave marginals, thereby answering Open Question 1 in the affirmative.

Theorem 1.4 (Robustly Learning General Halfspaces). *Let $\mathcal{H}$ be the class of halfspaces on $\mathbb{R}^d$. There are two algorithms for learning distributions D over $\mathbb{R}^d \times \{\pm 1\}$ with unknown logconcave $\mathbf{x}$ -marginal $D_{\mathbf{x}}$. Each draws $\text{poly}(d/\epsilon)$ samples from D , and has the following performance guarantees:*

1. *The first algorithm runs in time $\text{poly}(d/\epsilon)$ and with high probability outputs a hypothesis with error $\tilde{O}(\text{opt}) + \epsilon$, where $\text{opt} = \text{opt}_D(\mathcal{H})$.*
2. *The second algorithm runs in time $\text{poly}(d)2^{\text{polylog}(1/\epsilon)}$ and with high probability outputs a hypothesis with error $C\text{opt} + \epsilon$, where $C > 1$ is a universal constant.*

The reader is referred to Theorems 7.14 and 7.15 in Section 7.5 for more detailed formal statements that also cover the partial PAC learning setting.

Moreover, the testable version of Theorem 1.4 answers another basic open problem in the literature. Specifically, even for the special case of Gaussian marginals, the best known polynomial-time error guarantee for testable learning of halfspaces was $\tilde{\Omega}(\sqrt{\text{opt}}) + \epsilon$ [DKLZ24]. Our learner works for any fixed logconcave distribution and achieves near-optimal error of $\tilde{O}(\text{opt}) + \epsilon$ (or $O(\text{opt}) + \epsilon$ with a quasi-polynomial time dependence on $1/\epsilon$).

Our second algorithmic application is for the task of learning functions of halfspaces under logconcave marginals, thereby answering Open Question 2 in the affirmative.

Theorem 1.5 (Robustly Learning Functions of Halfspaces). *Let $\mathcal{H}_{K,d}$ be the concept class of functions of K halfspaces on $\mathbb{R}^d$. There is an algorithm that for every distribution D over $\mathbb{R}^d \times \{\pm 1\}$ with unknown logconcave $\mathbf{x}$ -marginal $D_{\mathbf{x}}$ and any given $c > 0$, draws $\text{poly}_{K,c}(d/\epsilon)$ samples from D , runs in time $\text{poly}_{K,c}(d)2^{\text{polylog}_{K,c}(1/\epsilon)}$, and with high probability outputs a hypothesis with error $(1+c)\text{opt} + \epsilon$, where $\text{opt} = \text{opt}_D(\mathcal{H}_{K,d})$.*

The reader is referred to Theorem 7.12 in Section 7.4 for a more detailed formal statement that also covers the partial PAC learning setting.

We emphasize that the upper bound of Theorem 1.5 is new even for the very special case of learning intersections of two halfspaces in the realizable PAC model (i.e., $K = 2$ and $\text{opt} = 0$). As a bonus, the upper bound applies to any function of any constant number of halfspaces and is robust to adversarial label noise. Specifically, for $K = O(1)$ we obtain a constant factor approximation to the optimal error with sample complexity $\text{poly}(d/\epsilon)$ and computational complexity $\text{poly}(d/\epsilon)2^{\text{polylog}(1/\epsilon)}$. Moreover, we can achieve a factor $(1+c)$ approximation, for any desired parameter $c > 0$, by increasing the complexity as a function of c .

Our third algorithmic application, obtained via a refinement of our generic technique, is for the task of robustly learning multiclass linear classifiers under logconcave marginals, thereby answering Open Question 3 in the affirmative.

Theorem 1.6 (Robustly Learning Multiclass Linear Classifiers). *There are two algorithms for learning the concept class $\mathcal{M}_{K,d}$ of K -MLC on $\mathbb{R}^d$. Each algorithm draws $\text{poly}(dK/\epsilon)$ samples from a distribution D over $\mathbb{R}^d \times [K]$ with unknown logconcave $\mathbf{x}$ -marginal $D_{\mathbf{x}}$, and has the following performance guarantees:*

1. *The first algorithm runs in time $\text{poly}(dK/\epsilon)$ and with high probability outputs a hypothesis with error $\tilde{O}(K)\tilde{O}(\text{opt}) + \epsilon$, where $\text{opt} = \text{opt}_D(\mathcal{M}_{K,d})$.*
2. *The second algorithm runs in time $\text{poly}(d)2^{\text{polylog}(K/\epsilon)}$ and outputs a hypothesis with error $O(K)\text{opt} + \epsilon$.*

The reader is referred to Theorem 7.21 in Section 7.6 for a more detailed formal statement that also covers the partial PAC learning setting. For comparison, the prior work [DIKZ25] gave an algorithm for the special case of the Gaussian distribution with complexity $\text{poly}(d)2^{\text{poly}(K/\epsilon)}$, and error of $O(\text{opt}) + \epsilon$. Our algorithm achieves a somewhat weaker robustness guarantee with substantially better complexity and under much weaker distributional assumptions.

1.2 Technical Overview

In this section, we give a detailed overview of the techniques developed in this paper.

Result	Function Class	Distribution	Time Complexity	Error	Notes
Theorem 7.9	L_1 -sandwiching K -MIMs	Logconcave	$d^{O(\ell(c))} 2^{\text{polylog}_{K,c}(1/\epsilon)}$	$(1+c)\text{opt} + \epsilon$	Partial, Testable
[KKMS08]	L_1 -Approximable Classes	General	$d^{O(\ell(\epsilon))}$	$\text{opt} + \epsilon$	
Theorem 7.12	Functions of K Halfspaces	Logconcave	$\text{poly}_{K,c}(d) 2^{\text{polylog}_{K,c}(1/\epsilon)}$	$(1+c)\text{opt} + \epsilon$	Partial, Testable
[KKM13]	Functions of K Halfspaces	Logconcave	$d^{2^{\text{poly}(K/\epsilon)}}$	$\text{opt} + \epsilon$	
[Vem10b]	Intersections of K Centered Halfspaces	Isotropic Logconcave	$O(dK/\epsilon)^K$	ϵ	Realizable
[DKS18]	Intersections of K General Halfspaces	Gaussian	$\text{poly}(dK/\epsilon) + (K/\epsilon)^{O(K^2)}$	$\tilde{O}(K^{4/11}\text{opt}^{1/11}) + \epsilon$	
[DIKZ25]	Intersections of K General Halfspaces	Gaussian	$\text{poly}(d) 2^{\text{poly}(K/\epsilon)}$	$K\tilde{O}(\text{opt}) + \epsilon$	
Theorem 7.21	K -MLC	Logconcave	$\text{poly}(dK/\epsilon)$	$\tilde{O}(K\text{opt}) + \epsilon$	Partial, Testable
[DIKZ25]	K -MLC	Gaussian	$\text{poly}(d) 2^{\text{poly}(K/\epsilon)}$	$O(\text{opt}) + \epsilon$	
Theorem 7.15	General LTFs	Logconcave	$\text{poly}(d/\epsilon)$	$\tilde{O}(\text{opt}) + \epsilon$	Partial, Testable
[ABL17]	Homogeneous LTFs	Logconcave	$\text{poly}(d/\epsilon)$	$O(\text{opt}) + \epsilon$	
[DKS18]	General LTFs	Logconcave	$\text{poly}(d/\epsilon)$	$\tilde{\Omega}(\text{opt}^{1/2}) + \epsilon$	$O(\text{opt}) + \epsilon$ Gaussian
[DKLZ24]	General LTFs	Gaussian	$\text{poly}(d/\epsilon)$	$\tilde{O}(\text{opt}^{1/2}) + \epsilon$	Testable

Table 1: Summary and comparison of some of our results on agnostic learning under logconcave distributions. All our theorems listed in the table have sample complexity polynomial in d/ϵ with degree depending on K, c , or $\ell(c)$.

Weakly Learnable Classes A natural method to address such learning problems is the method of moments. We note that for nice families of distributions, there always exists a low-degree polynomial p so that its correlation with the optimal hypothesis h^* , $\mathbf{E}[p(\mathbf{x})h^*(\mathbf{x})]$ is not too small. As we can easily compute moments of D , it is not hard to find such a p if it exists. Furthermore, thresholding this polynomial can be used as a weak learner—a method to produce a hypothesis with error rate slightly less than $1/2$. Furthermore, this weak learner can be made robust in two ways. For one, as long as the noise level over the input distribution D is not too large, $p(\mathbf{x})$ also has non-trivial correlation with the noisy label y , and thus yields a weak hypothesis over D . Furthermore, one can often somewhat relax the condition on the marginal distribution. In particular, so long as the marginal is “close” to logconcave in a reasonable sense, that should suffice to produce this weak learner. In particular, for our purposes, it will be enough to have a distribution whose *central moments* nearly match those of a logconcave distribution. These observations inspire us to consider a broad family of classes, formalized in Definition 2.7 and termed (k, Δ, β, C) -*weakly learnable* classes. Roughly speaking, a class is weakly learnable if given any distribution D with low noise rate, whose first k central moments approximately match those of some logconcave distribution, one can efficiently compute a hypothesis whose error is slightly better than $1/2$. As we will see in

Theorem 7.4, essentially any reasonable class of multi-index models satisfies this assumption for some setting of the parameters.

Agnostic Partial Learning and Boosting from Weak Partial Learners Given that we assume above that our function class has some kind of weak learner, it is natural to attempt to use a boosting algorithm to strengthen this to a stronger learner. Unfortunately, in order for boosting to work, one typically needs a weak learning algorithm that works not just over the original marginal distribution, but also over the marginals of distributions used by the boosting algorithm, which can be much less structured than a logconcave distribution. Fortunately, if we use a smooth boosting algorithm, see, e.g., [Ser03, Gav03, Fel10, KK09], these marginals will always consist of the original distribution conditioned on some not-too-unlikely event (see Fact 3.3). This motivates us to consider a stronger but natural generalization of the original problem that we term *agnostic partial learning* (see Definition 2.3). In agnostic partial learning, we receive samples $(\mathbf{x}, y)$ from a distribution D over $\mathbb{R}^d \times \{0, \pm 1\}$, with logconcave marginal, where a sample with $y = 0$ can be thought of as an example that goes unclassified. The goal of an agnostic partial learner is to output a hypothesis that does well over the examples that are classified in our data. We show that, to do boosting, it suffices to have a *weak partial learner* (see Definition 3.1). *That is a learner that will always return a hypothesis with error somewhat less than $1/2$ when given a partial classification problem with error not too big and the fraction of non-zero labels not too small.* In Theorem 3.2, we show that given a weak partial learner that works when the noise rate is less than $1/C$, we can use it to produce a hypothesis with error $(C/2)\text{opt} + \epsilon$.

Efficient Weak Partial Learning via Space Partitioning and Discrepancy Finder Unfortunately, even if our function class is itself weakly learnable, this does not mean we can immediately obtain a weak partial learner. This is because if we let $\mathbf{x}$ and $\mathbf{x}'$ be the marginals over all samples and over classified samples, respectively, we are not guaranteed that they have nearly matching moments. Without this property, it is not clear that our class being (k, Δ, β, C) -weakly learnable suffices to produce a weak hypothesis. As a result, simply running a weak learning algorithm on the classified data does not in general suffice. This would only work if $\mathbf{x}$ and $\mathbf{x}'$ had nearly matching central moments.

So what can we do if these moments do not match? In this case, there must be a direction $\mathbf{v}$ and an integer $i \leq k$ such that the expectations of $(\mathbf{v} \cdot \mathbf{x})^i$ and $(\mathbf{v} \cdot \mathbf{x}')^i$ substantially disagree. This means that the distributions of $(\mathbf{v} \cdot \mathbf{x})$ and $(\mathbf{v} \cdot \mathbf{x}')$ must be somewhat different. This, in turn, implies that if we can find such a direction $\mathbf{v}$ through some efficient algorithm—that we term *discrepancy finder* (see Definition 4.1)—then we can split the entire space into two halves such that the fraction of classified samples differs substantially across halves. After the split, we can recursively run the above procedure over each half until a direction with moment discrepancy cannot be found; or we reach a part, where the probability that $\mathbf{x}'$ falls into it is too small. By doing this repeatedly, we partition the whole space into many convex regions R , with $\Pr[y \neq 0 \mid R]$ starting to become more and more extreme. However, since $\mathbf{x}'$ is obtained from $\mathbf{x}$ by conditioning on some event of probability α , the ratio $\Pr[\mathbf{x}' \in R] / \Pr[\mathbf{x} \in R]$ will never exceed $1/\alpha$. This fact, along with a careful analysis of the partitioning scheme, allows us to show that we end up with a polynomial number of regions, so that most of the mass of $\mathbf{x}$ belongs to regions where the conditional moments of $\mathbf{x}$ and $\mathbf{x}'$ nearly match (see Lemma 4.5). As a result, by calling the two algorithmic modules—namely, the weak learner and the discrepancy finder—polynomially many times, we get a weak partial hypothesis over D , as long as the noise rate is not too high and the mass of non-zero labels is not too small (see Theorem 4.2).

Discrepancy Finder under Logconcave Distributions via Dimension Reduction Given the discussion thus far, we already have the ideas necessary to design an efficient robust learning algorithm in the Statistical Query model, where given the moments of $\mathbf{x}$ and $\mathbf{x}'$ we can existentially determine if there is a direction of significant mismatch. Unfortunately, computational difficulties arise in designing an actual algorithm for the discrepancy finder. In particular, given distributions $\mathbf{x}$ and $\mathbf{x}'$, the discrepancy finder is asked to either certify that $(\mathbf{v} \cdot \mathbf{x})$ and $(\mathbf{v} \cdot \mathbf{x}')$ approximately match their moments for each $\mathbf{v}$, or to find a direction $\mathbf{v}$, in which these moments differ substantially. While it is not difficult to approximate the expectation of $(\mathbf{v} \cdot \mathbf{x})^k - (\mathbf{v} \cdot \mathbf{x}')^k$ as an explicit polynomial in $\mathbf{v}$, finding an approximate maximum of this polynomial is believed to be computationally hard in general. Fortunately, the structure of the distributions in question will allow us to build such a discrepancy finder in quasi-polynomial time (see Theorem 5.1). We provide an overview of our algorithm to do so in the following paragraphs.

Our first observation is that if $\mathbf{x}$ is sampled from a logconcave distribution and $\mathbf{x}'$ is $\mathbf{x}$ conditioned on an event of probability at least α , then their moments cannot differ arbitrarily. In particular, we show in Lemma 5.7 that there must be a degree- k polynomial Q with relatively small L_2 norm over $\mathbf{x}$ such that for any degree- k polynomial p , $\mathbf{E}[p(\mathbf{x})Q(\mathbf{x})] = \mathbf{E}[p(\mathbf{x}) - p(\mathbf{x}')]^k$. How can we exploit this fact? For the purpose of motivation, we first cover the simpler case where $\mathbf{x}$ is drawn from a standard Gaussian distribution. In this case, we can expand $Q(\mathbf{x})$ in the Hermite basis, yielding a sequence of tensors $\mathbf{T}_0, \dots, \mathbf{T}_k$ with small Frobenius norm, so that for every $\mathbf{v} \in \mathbb{S}^{d-1}$ we can write $\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[Q(\mathbf{x})(\mathbf{v} \cdot \mathbf{x})^k]$ as a linear combination of $\mathbf{T}_k \cdot \mathbf{v}^{\otimes k}$ for $i \in [k]$. Then, viewing $\mathbf{T}_i$ as a bilinear form over $\mathbb{R}^d \times \mathbb{R}^{d^{i-1}}$, and using a singular-value decomposition, one can efficiently find a subspace V (of $\text{poly}(\log(1/\alpha)/\Delta)$ dimension) so that the moment-discrepancy in any direction $\mathbf{u}$ is not too much larger than the discrepancy in the direction $\mathbf{v} = \text{proj}_V(\mathbf{u})$. We can then spend time $2^{O(\dim(V))}$ (which is quasipoly($1/\alpha$) if Δ is constant) to do an exhaustive search over V for directions of large discrepancy.

Unfortunately, the above argument critically exploits many features of the Gaussian distributions (such as the fact that moments in orthogonal directions are independent) that do not hold for a general logconcave distribution. However, the basic idea of dimension-reduction can still be made to work. In particular, we wish to find a subspace V with a not-too-large dimension so that the maximum of the multi-linear form $\mathbf{E}[Q(\mathbf{x}) \prod_{i=1}^k (\mathbf{u}^{(i)} \cdot \mathbf{x})]$ over any unit vectors $\mathbf{u}^{(i)}$ is at most a constant multiple of the maximum of $\mathbf{E}[Q(\mathbf{x}) \prod_{i=1}^k (\mathbf{v}^{(i)} \cdot \mathbf{x})]$, with $\mathbf{v}^{(i)} \in V$, plus a small error Δ (see Lemma 5.5). We achieve this in two steps. First, we notice that $\mathbf{E}[Q(\mathbf{x})p(\mathbf{x})]$ (as a linear function of $p(\mathbf{x})$) can be written as a linear function of the expectations of p 's derivative $\nabla^i p(\mathbf{x})$. Making use of recent progress on the KLS conjecture, we can show that the size of the corresponding tensors cannot be too large. Combining this fact with a spectral analysis, we give an efficient algorithm (see Lemma 5.4) to find a subspace U so that if all of the vectors $\mathbf{u}^{(i)}$ s are orthogonal to U , then $\mathbf{E}[Q(\mathbf{x}) \prod_{i=1}^k (\mathbf{u}^{(i)} \cdot \mathbf{x})]$ is absolutely small.

The next step is to use this as a subroutine to recursively build our target subspace V . To do this, we decompose $\mathbf{u}^{(i)} = \mathbf{u}_U^{(i)} + \mathbf{u}_{U^\perp}^{(i)}$. This allows us to decompose $\mathbf{E}[Q(\mathbf{x}) \prod_{i=1}^k (\mathbf{u}^{(i)} \cdot \mathbf{x})]$ into 2^k terms $\mathbf{E}[Q(\mathbf{x}) \prod_{i=1}^k (\mathbf{v}^{(i)} \cdot \mathbf{x})]$, with each $\mathbf{v}^{(i)}$ either in U or in $U^\perp$. As a result, we only need to consider the case where one of the vectors is in U . Let $\mathbf{g}^{(1)}, \dots, \mathbf{g}^{(N)}$ be an orthonormal basis of U . If we apply the subroutine in the first step to $Q(\mathbf{x})(\mathbf{g}^{(i)} \cdot \mathbf{x})$ for each i , this allows us to get another subspace U' such that when each $\mathbf{u}^{(i)}$ is orthogonal to U' , then $\mathbf{E}[Q(\mathbf{x})(\mathbf{g}^{(i)} \cdot \mathbf{x}) \prod_{i=1}^{k-1} (\mathbf{u}^{(i)} \cdot \mathbf{x})]$ is small. In particular, this means that if $\mathbf{u}^{(1)} \in U$ and $\mathbf{u}^{(2)}, \dots, \mathbf{u}^{(k)}$ are orthogonal to U' , then $\mathbf{E}[Q(\mathbf{x}) \prod_{i=1}^k (\mathbf{u}^{(i)} \cdot \mathbf{x})]$ will be small. By repeating this approach for k iterations, and taking V as the span of the subspaces found in the process, we obtain that the maximum $\mathbf{E}[Q(\mathbf{x}) \prod_{i=1}^k (\mathbf{u}^{(i)} \cdot \mathbf{x})]$ will not be much larger if all the unit vectors are taken from V . This completes the overview of the

analysis of our learning algorithm.

Algorithmic Application (I): Generic Family and Learning via Sandwiching Polynomials

To make use of our algorithm template, it remains to show that there are natural function classes that satisfy our condition of being (k, Δ, β, C) -weakly learnable for some reasonable sets of parameters. In particular, we argue that essentially any non-pathological family of K -MIMs is (k, Δ, β, C) -weakly learnable for an appropriate choice of parameters. We note that the assumption of being a K -MIM here is critical, as nearly matching central moments will imply that nearly all moments along the critical directions will nearly match.

In order to construct such a weak learner, suppose that for the optimal hypothesis h^* in the class, we can find a polynomial p , such that $\mathbf{E}[p(\mathbf{x})] = 0$ and $p(\mathbf{x})$ is positive whenever $h^*(\mathbf{x}) = 1$. In this case, we have $\mathbf{E}[h^*(\mathbf{x})p(\mathbf{x})] = \mathbf{E}[(h^*(\mathbf{x}) + 1)p(\mathbf{x})] > 0$. Since this p then correlates with the indicator set of $h^*(\mathbf{x}) = 1$, it is not hard to see that some appropriate threshold of p will necessarily produce a weak learner.

We claim that very mild conditions on h^* will allow us to construct such a p . For example, supposing that $\mathbf{x}$ is isotropic, as long as there is a not-too-small ball B not-too-far from the origin so that h^* is -1 on B , we can construct (by standard techniques) a polynomial that approximates the indicator function of B . Subtracting this from a small constant, we produce a polynomial that is mean 0 and positive whenever h^* is. As we need something similar to hold for *every* logconcave $\mathbf{x}$ (not just isotropic ones), we will need the slightly stronger condition (see Definition 7.3) that every ellipsoid E contains a (not too much) smaller ellipsoid E' so that h^* is constant on E' . We note that this should hold for essentially any reasonable K -MIM.

Unfortunately, the parameters obtained through this natural approach will usually lead to poor approximation ratios. However, if we make the stronger assumption that h^* has sandwiching polynomials (i.e., polynomials p^u and p^l with $p^u(\mathbf{x}) \geq h^*(\mathbf{x}) \geq p^l(\mathbf{x})$ pointwise and $\mathbf{E}[p^u(\mathbf{x}) - p^l(\mathbf{x})] \leq c$ for any small $c > 0$; see Definition 7.8), then a much better error guarantee can be obtained (see Theorem 7.9). To see this, $\mathbf{E}[p^u(\mathbf{x}')y]$ will approximate $\mathbf{E}[h^*(\mathbf{x})y] = 1 - 2\text{opt} = c$ within error at most $\mathbf{E}[p^u(\mathbf{x}') - p^l(\mathbf{x}')] \approx \mathbf{E}[p^u(\mathbf{x}) - p^l(\mathbf{x})]$, which can be made as a small multiple of c . This means that we can find a weak hypothesis, as long as $\mathbf{E}[h^*(\mathbf{x}')y]$ is more than a small constant, which happens when opt is only slightly less than $1/2$. In particular, this implies that for every K -MIM that admits L_1 -sandwiching polynomial approximation, we can learn it in time $\text{poly}(d)2^{\text{polylog}(1/\epsilon)}$ to error $(1 + c)\text{opt} + \epsilon$, for any small constant $c > 0$, under any logconcave distribution. In particular, by leveraging the sandwiching polynomial construction from [KKM13], this immediately gives a QPTAS for many fundamental classes, such as functions of halfspaces and low-dimensional low-degree polynomial threshold functions (see Section 7.4).

Algorithmic Application (II): Halfspaces and Multiclass Linear Classifiers

As another important application, our algorithmic framework also gives a way to tackle more challenging multiclass classification problems. A natural way to solve a multiclass classification problem is to break it into $\binom{K}{2}$ binary classification problems to learn hypotheses $h_{ij}(\mathbf{x})$ that tell whether an example is more likely to have label i or label j and combine them together. We show in Lemma 7.22 that it suffices to design a semi-agnostic partial learning algorithm for the class of halfspaces over logconcave distributions. In particular, due to the structure of the class of halfspaces, we can use a simple Chow-parameter estimation method to obtain a weak hypothesis, as long as the moments of the marginal distribution match some logconcave marginal. In particular, in Theorem 7.14, we show that if the first four moments approximately match, then we can agnostically partially learn a halfspace to error $O(\text{opt}) + \epsilon$ in $\text{poly}(d)2^{\text{polylog}(1/\epsilon)}$ time, which in turn implies a quasi-polynomial

time algorithm for multiclass linear classification with error $O(K \cdot \text{opt}) + \epsilon$. The quasi-polynomial dependence on ϵ is due to the fact that we need to implement a discrepancy finder for the 4th moment. On the other hand, if we only use the first two moments, we can always find the moment discrepancy in polynomial time. For the halfspace learning problem (Theorem 7.15), this fact gives us a polynomial-time algorithm that achieves error $\tilde{O}(\text{opt}) + \epsilon$; this implies a polynomial-time algorithm for multiclass linear classification with error $\tilde{O}(K \cdot \text{opt}) + \epsilon$.

Testable Learning Finally, we point out that our algorithmic framework naturally fits the testable learning framework [RV23] (see Definition 6.1). Roughly speaking, given a known logconcave distribution $\mathbf{x} \sim D_B$ and sample access from an arbitrary distribution D over $\mathbb{R}^d \times \{0, \pm 1\}$, a testable learner is asked to either reject or output a hypothesis. We want to guarantee that the probability that the algorithm accepts but outputs a bad hypothesis is small; and that when $D_{\mathbf{x}} = D_B$, the algorithm will accept with high probability. Towards this end, we notice that our boosting algorithm will succeed with no extra assumption, as long as the weak partial learner succeeds. Furthermore, our weak partial learner is guaranteed to work, as long as we can partition the space into regions R so that most of the mass of examples with non-zero label, $\mathbf{x}'$, belong to regions on which the moments of $(\mathbf{x}' \mid R)$ and $(\mathbf{x} \mid R)$ approximately match. This implies that when we create the partition for the weak partial learner, we can replace the global marginal distribution $D_{\mathbf{x}}$ by D_B . If within the sample and time budget, the partition is not created successfully, we simply reject; otherwise, the partition is created successfully, and we obtain a weak partial hypothesis. In particular, if $D_{\mathbf{x}} = D_B$, we are likely to accept (see Theorem 6.2).

1.3 Organization

In Section 2, we begin with some preliminaries followed by a formal definition of an agnostic partial classifier and its resulting error metrics (Definitions 2.3 and 2.5). We then state our general algorithmic result (Theorem 2.8) efficiently translating our notion of weak learning (Definition 2.7) to agnostic partial learning. The subsequent sections develop the necessary machinery to prove Theorem 2.8. In Section 3, we define the notion of weak *partial* learning (Definition 3.1), cover the generic results on boosting from a weak partial learner, and show how it can be used to efficiently produce agnostic partial learners (Theorem 3.2). Section 4 develops our iterative spatial decomposition and shows (Theorem 4.2) how it can be used to produce a weak partial learner for function classes satisfying our weak learning condition (Definition 2.7). In Section 5, we provide our algorithm for computing directions of moment discrepancy for logconcave distributions. We combine the above results to prove Theorem 2.8, and state our testable learning extensions in Section 6. Finally, in Section 7 we describe our concrete applications, showing that our assumptions are satisfied by a wide variety of multi-index models, and in particular functions of halfspaces and multiclass linear classifiers.

2 Setup and Statement of the Main Result

Notation For $n \in \mathbb{Z}_+$, we let $[n] = \{1, 2, \dots, n\}$. We use small boldface characters for vectors and capital bold characters for matrices and tensors. For $\mathbf{x} \in \mathbb{R}^d$ and $i \in [d]$, $\mathbf{x}_i$ denotes the i -th coordinate of $\mathbf{x}$ and $\|\mathbf{x}\|_2 = \sqrt{\sum_{i=1}^d (\mathbf{x}_i^2)}$ is the ℓ_2 norm of $\mathbf{x}$. Let V be a subspace, we use $\text{proj}_V(\mathbf{x})$ to denote the projection of $\mathbf{x}$ onto subspace V , we will also use $\mathbf{x}_V$ for short to denote $\text{proj}_V(\mathbf{x})$. We use $\mathbb{S}^{d-1}$ to denote the unit sphere over $\mathbb{R}^d$ and use $\mathbb{B}^{d-1}(\mathbf{u}, r)$ to denote the ball in $\mathbb{R}^d$ centered at $\mathbf{u}$, with radius r . Let $\mathbf{A} \in \mathbb{R}^{d \times d}$, we define $\mathcal{E}(\mathbf{u}, \mathbf{A}) := \{\mathbf{x} \mid \|\mathbf{A}(\mathbf{x} - \mathbf{u})\|_2 \leq 1\}$ to be the ellipsoid

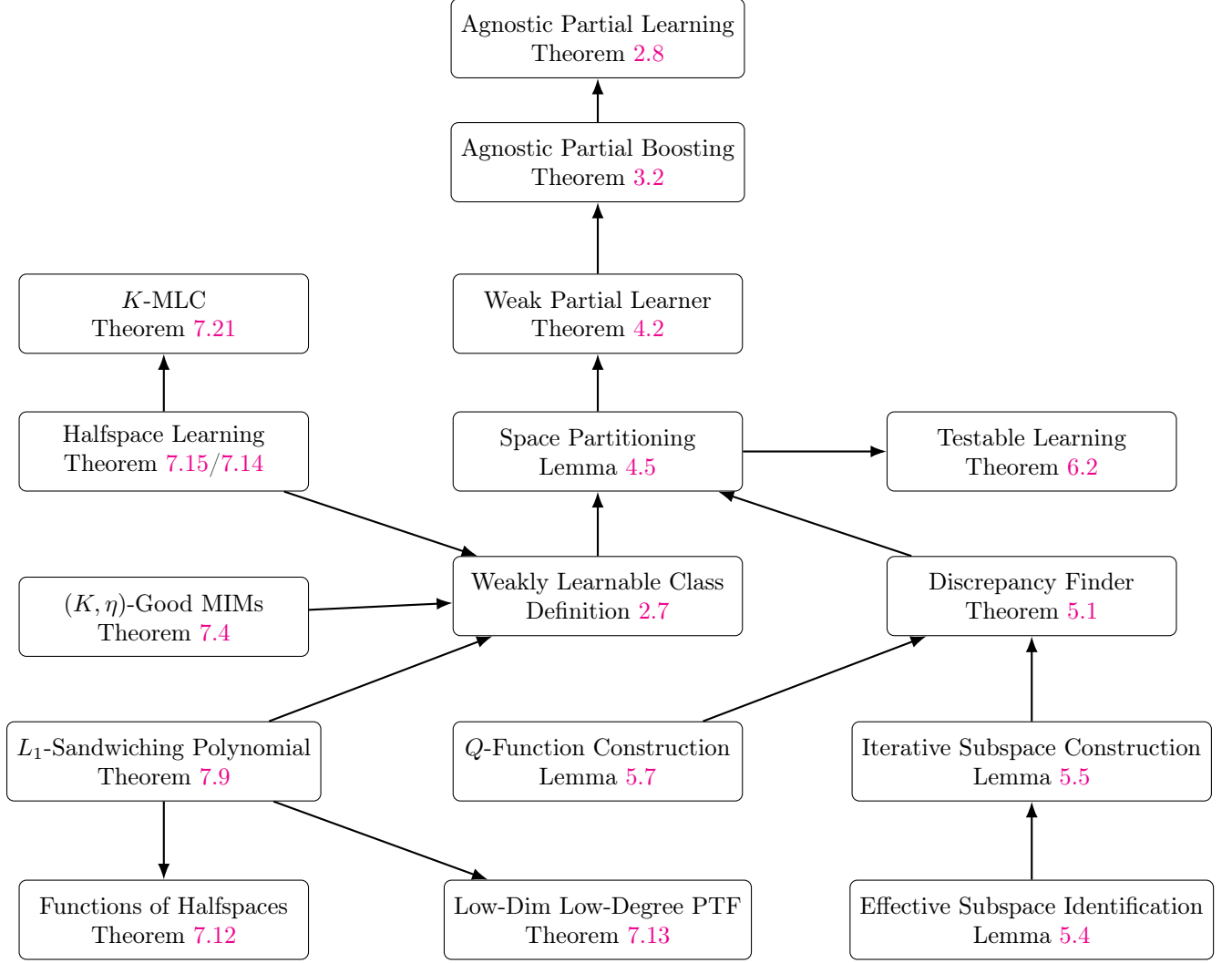


Figure 1: Overview of the proof structure and dependencies. Directed edges indicate logical dependencies: an arrow $A \rightarrow B$ means that the result at B relies on or invokes the result at A .

centered at $\mathbf{u}$, with shape $\mathbf{A}$. For an ℓ -order tensor $\mathbf{T}$, we use $\|\mathbf{T}\|_F = \sqrt{\sum_{\zeta} \mathbf{T}_{\zeta}^2}$ to denote its frobenius norm. For two tensors $\mathbf{T}, \mathbf{T}'$, we will use $\mathbf{T} \cdot \mathbf{T}'$ to denote $\sum_{\zeta} \mathbf{T}_{\zeta} \mathbf{T}'_{\zeta}$. For $\alpha = (\alpha_1, \dots, \alpha_d)$, a multi-index, we use $\mathbf{x}^{\alpha} = \prod_{i=1}^d \mathbf{x}_i^{\alpha_i}$, the monomial with multi-index α . For an event A over $\mathbb{R}^d$, we will use $\mathbb{1}\{A\}$ to denote the indicator of A i.e. If A is true, then $\mathbb{1}\{A\} = 1$ and otherwise $\mathbb{1}\{A\} = 0$. For D a distribution over $\mathbb{R}^d$ and $f : \mathbb{R}^d \rightarrow \mathbb{R}$, we use $\|f\|_{D,p} := (\mathbf{E}_{\mathbf{x} \sim D}[|f(\mathbf{x})|^p])^{1/p}$ to denote the ℓ_p norm of f under D . For a distribution D over $\mathbb{R}^d \times \mathcal{Y}$, we use $D(\mathbf{x}, y)$ to denote the density of D at $(\mathbf{x}, y)$ and use $D(y | \mathbf{x})$ the conditional distribution of y given $\mathbf{x}$ under D . And, we use $D_{\mathbf{x}}$ to denote the marginal of D on $\mathbf{x}$.

Basics on Logconcave Distributions The family of distributions D that we study in this paper are assumed to have logconcave marginal $D_{\mathbf{x}}$ on the feature space, defined as follows.

Definition 2.1 (Logconcave Distribution). *A distribution $D_{\mathbf{x}}$ over $\mathbb{R}^d$ with density function $f(\mathbf{x})$ is logconcave if the function $\log f(\mathbf{x})$ is concave. We say that a logconcave distribution $D_{\mathbf{x}}$ is isotropic*

logconcave if $\boldsymbol{\mu} := \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[\mathbf{x}] = \mathbf{0}$ and $\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[\mathbf{x}\mathbf{x}^\top] = \mathbf{I}$, where $\mathbf{I}$ is the identity matrix. For $c \in (0, 1)$, we say $D_{\mathbf{x}}$ is c -isotropic logconcave, if $\|\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[\mathbf{x}]\|_2 \leq c$ and $(1 - c)\mathbf{I} \leq \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(\mathbf{x} - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^\top] \leq (1 + c)\mathbf{I}$.

As shown in [LV07], any logconcave distribution can be efficiently transformed to nearly isotropic logconcave.

Fact 2.2 (see, e.g., Theorem 2.6 in [LV07]). *There is an algorithm with the following guarantee: given sample access to any logconcave distribution $D_{\mathbf{x}}$ on $\mathbb{R}^d$ and parameters $c, \delta \in (0, 1)$, the algorithm draws $M = \tilde{O}(dc^{-2} \log(1/\delta))$ samples from $D_{\mathbf{x}}$, runs in sample polynomial time, and outputs an affine transformation $L : \mathbb{R}^d \rightarrow \mathbb{R}^d$ such that when $\mathbf{x} \sim D_{\mathbf{x}}$, $L(\mathbf{x})$ follows a c -isotropic logconcave distribution with probability at least $1 - \delta$.*

Thanks to Fact 2.2, most of the technical lemmas in this work will focus on c -isotropic logconcave distributions, for suitably small c .

Basic Definitions for Partial Learning One of our key innovations is that of trying to produce weak learners for *partial* classifiers. Such a classifier can be thought of as a function $f : X \rightarrow \{\pm 1, 0\}$ (or more generally given as a distribution over $X \times \{\pm 1, 0\}$ where the output $f(x) = 0$ corresponds to f not classifying x and an output of ± 1 corresponding to that Boolean classification. Formally, we set out some of our basic terminology as follows:

Definition 2.3 (Agnostic Partial Learning). *Let $\mathcal{H}$ be a class of Boolean (i.e., $\{\pm 1\}$ -valued) functions over $\mathbb{R}^d$, and let D be a distribution over $\mathbb{R}^d \times \{\pm 1, 0\}$. Denote by D_f (called the effective distribution) the conditional distribution of D over the event $\mathcal{E} := \{(\mathbf{x}, y) \mid y \neq 0\}$. The error of a hypothesis $h : \mathbb{R}^d \rightarrow \{\pm 1\}$ is defined as $\text{err}_{D_f}(h) = \Pr_{(\mathbf{x}, y) \sim D_f}[h(\mathbf{x}) \neq y]$ and define $\text{opt}_{D_f}(\mathcal{H}) := \inf_{h \in \mathcal{H}} \text{err}_{D_f}(h)$.*

A (semi-)agnostic partial learner is one that given access to samples from such a distribution D attempts to compute a hypothesis h whose error is competitive with $\text{opt}_{D_f}(\mathcal{H})$, and in particular is bounded by $F(\text{opt}_{D_f}(\mathcal{H})) + \epsilon$ for some non-decreasing function $F : [0, 1] \rightarrow [0, 1]$ and $\epsilon > 0$.

We say that this problem is agnostic partial learning over some distribution D^* or class of distributions $\mathcal{C}$ on $\mathbb{R}^d$ if one is additionally guaranteed that the marginal of D over $\mathbb{R}^d$ is equal to D^* or lies in $\mathcal{C}$. In particular, our paper will be focused on the problem of agnostic partial learning of certain well behaved function classes $\mathcal{H}$ over the class of logconcave distributions.

We note that one important parameter of a partial classifier is the probability with which it assigns a non-zero value, a quantity we call the *weight* of the classifier. This is particularly important as classifiers with low weight will be difficult to learn at the very least because many samples will be required in order to see anything of interest. Formally, we define:

Definition 2.4 (Weight of a Partial Classifier). *For a distribution D on $\mathbb{R}^d \times \{\pm 1, 0\}$ we define its weight, denoted $W(D)$, to be $\Pr_{(\mathbf{x}, y) \sim D}[y \neq 0]$.*

Finally, while the notion of error for a hypothesis given above is perhaps the most natural for measuring the error of a partial classifier, it is sometimes useful to consider the probability over the distribution D as a whole that our hypothesis returns the wrong value. We call this the “global error” as defined below:

Definition 2.5 (Global Classification Error). *Let D be a distribution over $\mathbb{R}^d \times \{\pm 1, 0\}$. We define the global error of a hypothesis $h : \mathbb{R}^d \rightarrow \{\pm 1\}$ with respect to D as $\text{err}_D(h) := \Pr_{(\mathbf{x}, y) \sim D}[h(\mathbf{x}) \neq y, y \neq 0]$. The optimal global error of $\mathcal{H}$ with respect to D is $\text{opt}_D(\mathcal{H}) := \inf_{h \in \mathcal{H}} \text{err}_D(h)$.*

Remark 2.6. *We remark that for every boolean function h , $\text{err}_D(h) = W(D) \text{err}_{D_f}(h)$.*

Our Main Result Our main algorithmic result is applicable to a broad class of hypothesis classes that admit a weak partial learner so long as certain moment conditions on the effective marginal, D_f , are met. In practice, this weak learner will always be obtained by finding a low degree polynomial p which has non-trivial inner product with y , but in principle any (efficient) weak learning algorithm should suffice.

Note that for a partial classifier, one should consider applying the definition below to D_f .

Definition 2.7 ((k, Δ, β, C) -weakly learnable class). *Let $\mathcal{H}$ be a class of Boolean functions over $\mathbb{R}^d$. For $k \in \mathbb{N}, \Delta > 0, \beta \in (0, 1), C > 1$, we say that $\mathcal{H}$ is (k, Δ, β, C) -weakly learnable, if there is an algorithm $\mathcal{A}$ with the following performance guarantee. Suppose that for an input distribution D over $\mathbb{R}^d \times \{\pm 1\}$, the following conditions are satisfied:*

1. *There is some $1/2$ -isotropic logconcave distribution D'_x over $\mathbb{R}^d$ and a given affine transformation $L : \mathbb{R}^d \rightarrow \mathbb{R}^d$ such that $\tilde{D}_x$, the distribution of D_x under the affine transformation L satisfies for every $\mathbf{w} \in \mathbb{S}^{d-1}$ and all $i \in [k]$, $\left| \mathbf{E}_{\mathbf{x} \sim \tilde{D}_x}[(\mathbf{w} \cdot \mathbf{x})^i] - \mathbf{E}_{\mathbf{x} \sim D'_x}[(\mathbf{w} \cdot \mathbf{x})^i] \right| < \Delta$,*
2. $\text{opt}_D(\mathcal{H}) \leq 1/C$.

Then $\mathcal{A}$, on input i.i.d. samples from D , outputs a hypothesis $\hat{h}$ satisfying $\text{err}_D(\hat{h}) \leq 1/2 - \beta$ with probability at least 99%. We will call β the “advantage” of the (weak hypothesis) $\hat{h}$. The algorithm $\mathcal{A}$ will be called a (k, Δ, β, C) -weak learner for $\mathcal{H}$.

A number of prior works require that the function class either exhibit a weak learner or some kind of moment-based identifiability. However, previous results typically require that this identifiability either holds for all marginal distributions or only for the specific marginals being learned. To make our argument work for partial learners, we will need to be able to work with the effective marginal $(D_f)_x$, which is merely close to logconcave. Fortunately, if we merely assume that we nearly match central moments with a logconcave distribution (the requirement for $1/2$ -isotropic can be fixed by applying an affine transformation), this is a condition that we can both reduce to (as we show in Sections 4 and 5) and a condition that is sufficient in many instances (as we show in Section 7).

We are now ready to state our main algorithmic result. In particular, we show that any weakly learnable class of functions can be agnostically learned with respect to any logconcave distribution.

Theorem 2.8 (Weak Learning Implies Agnostic Partial Learning). *Suppose that $\mathcal{H}$ is a class of functions that is (k, Δ, β, C) -weakly learnable by an algorithm $\mathcal{A}$ with sample complexity m and time complexity T . Then there is an agnostic partial learning algorithm for $\mathcal{H}$ with the following guarantee: Given any $\epsilon \in (0, 1)$ and sample access to any distribution D over $\mathbb{R}^d \times \{0, \pm 1\}$ of weight $W(D) \geq \alpha$ with marginal D_x logconcave, the algorithm draws $(1/\beta^2) d^{O(k)} \text{poly}_{k, \Delta}(1/(\alpha\epsilon)) m$ samples from D , runs in time $(1/\beta^2) d^{O(k)} 2^{\text{poly}_{k, \Delta}(\log(1/(\alpha\epsilon)))} T$, and outputs a hypothesis $\hat{h}$ such that with probability at least 99%, $\text{err}_{D_f}(\hat{h}) \leq (C/2) \text{opt}_{D_f}(\mathcal{H}) + \epsilon$.*

In other words, so long as $\mathcal{H}$ is weakly-learnable by an efficient algorithm $\mathcal{A}$ we have an agnostic partial learner for $\mathcal{H}$ that works over any logconcave marginal. This learner will have fully polynomial sample complexity (though the degree of that polynomial may depend on the parameters of $\mathcal{H}$ ’s weak learnability), and time complexity that is only quasipoly($1/\epsilon$) worse. This can be used to learn partial classifiers even of small weight, though as we will see this algorithm substantially improves on the state of the art even for many classes of total functions. Additionally, this learner can be made testable in the sense of Theorem 6.2.

3 Boosting a Weak Partial Hypothesis

The first key ingredient of our approach is a boosting algorithm establishing that a weak *partial* learner (Definition 3.1) can always be efficiently converted into a (strong) agnostic partial learner (Theorem 3.2).

This can be viewed as a natural extension of the boosting techniques widely used in distribution-free learning to the distribution-specific setting. Traditionally boosting algorithms have typically required that the weak learner is distribution-free. However, if a smooth boosting algorithm, see, e.g. [Fel10, KK09] is used, we show that it suffices to have a *distribution-specific* weak *partial* learner. Technically, this result follows from a careful use of the boosting algorithm of [Fel10], but conceptually we think this is a significant advance, as it allows us to consider only effective marginal distributions which are not too far from the original marginal—a condition that we will later show how to exploit.

Formally, we define a weak partial learner as follows.

Definition 3.1 ((β, C) -weak partial learner). *For $0 < \alpha, \beta < 1$ and $C > 1$, we say that an algorithm $\mathcal{A}$ is an (β, C) -weak partial learner for a class $\mathcal{H}$ with respect to the distribution D^* if it satisfies the following guarantee: For every $\alpha \in (0, 1)$ and for every input distribution D over $\mathbb{R}^d \times \{\pm 1, 0\}$ with $\mathbf{x}$ -marginal $D_{\mathbf{x}} = D^*$, $W(D) > \alpha$ and $\text{opt}_{D_f}(\mathcal{H}) \leq 1/C$, then $\mathcal{A}$ draws samples from D and outputs a hypothesis $\hat{h} : \mathbb{R}^d \rightarrow \{\pm 1\}$, such that with probability at least 99%, $\text{err}_{D_f}(\hat{h}) \leq 1/2 - \beta$.*

Furthermore, we say that $\mathcal{A}$ has time and sample complexity $T(\alpha)$ and $m(\alpha)$ if the appropriate complexities are bounded by the given functions.

The main result of this section is the following theorem.

Theorem 3.2 (Weak Partial Learning Implies Agnostic Partial Learning). *Let D^* be a distribution over $\mathbb{R}^d$, $\mathcal{H}$ be a hypothesis class over $\mathbb{R}^d$, and $\mathcal{A}$ be a (β, C) -weak partial learner for $\mathcal{H}$ with respect to D^* with time and sample complexity $T(\alpha), m(\alpha)$. There is an algorithm with the following performance guarantee: for any $\epsilon \in (0, 1)$, any input distribution D over $\mathbb{R}^d \times \{0, \pm 1\}$ with marginal $D_{\mathbf{x}} = D^*$ and $W(D) \geq \alpha$, the algorithm draws $N = \tilde{O}(\beta^{-2}/\epsilon)m(\alpha\epsilon)$ samples from D , runs in time polynomial in N and $T(\alpha\epsilon)$, and outputs a hypothesis $\hat{h}$ such that with probability at least 99%, $\text{err}_{D_f}(\hat{h}) \leq (C/2)\text{opt} + \epsilon$.*

Essentially, Theorem 3.2 is saying that in order to run a boosting algorithm instead of needing to run our weak learner on *arbitrary* input distributions, it suffices to run it only on distributions that can be obtained from our original distribution D_f (itself a conditioning of the base distribution D) by a rejection sampling procedure that accepts with probability at least ϵ . This in turn means that a distribution-specific weak partial learner is sufficient instead of requiring a distribution-independent weak learner. This can be shown by analyzing the boosting algorithm of [Fel10].

Fact 3.3 (Corollary of Theorem 3.5 in [Fel10]). *Let $\mathcal{H}$ be a hypothesis class over $\mathbb{R}^d$ and let $\mathcal{A}$ be an oracle with the following behavior: for every distribution D over $\mathbb{R}^d \times \{\pm 1\}$ such that $\min_{h \in \mathcal{H}} \text{err}_D(h) \leq 1/2 - a$, given m examples from D , it outputs a hypothesis $\hat{h}$ such that $\text{err}_D(\hat{h}) \leq 1/2 - \beta$, with probability at least 99%. Then there is an algorithm $\mathcal{B}$ with the following performance guarantee: for every distribution D over $\mathbb{R}^d \times \{\pm 1\}$ and for every $\epsilon > 0$, $\mathcal{B}$ makes $\tilde{O}(\beta^{-2}/\epsilon)$ many calls to $\mathcal{A}$, it runs in time $\text{poly}(T, \beta^{-1}, \epsilon^{-1})$, and outputs a hypothesis $\hat{h}$ such that $\text{err}_D(\hat{h}) \leq \text{opt}_D(\mathcal{H})/(1 - 2a) + \epsilon$, with probability 99%. Furthermore, each call to $\mathcal{A}$ is defined over some distribution D' so that D' is given by D conditioned on some explicit event E of probability at least ϵ .*

Remark 3.4. The statement of Theorem 3.5 in [Fel10] requires access to a distribution-free weak learner. However, the proof requires a weaker guarantee. Specifically, given a fixed distribution D ,

to boost a weak learner over D , it suffices that the weak learner $\mathcal{A}$ succeeds over distributions whose marginals can be rejection sampled from $D_{\mathbf{x}}$ with an acceptance rate of at least ϵ . Since this is not explicitly stated in [Fel10], we provide the details of the proof in Appendix B for completeness.

Given Fact 3.3, the proof of Theorem 3.2 is straightforward.

Proof. Our algorithm will simulate running $\mathcal{B}$ from Fact 3.3 to learn the distribution D_f . In particular, when the oracle $\mathcal{A}$ is called on some distribution D' , we note that D' is given by D_f conditioned on an event E of probability at least ϵ . This in turn is D conditioned on an explicit event of probability at least $\alpha\epsilon$. Letting D^0 be obtained from the distribution D by setting $y = 0$ whenever E does not hold, we note that $W(D^0) \geq \alpha\epsilon$ and $D_f^0 = D'$. Therefore, the (β, C) -weak partial learner $\mathcal{A}$ will satisfy the hypotheses of the oracle from Fact 3.3 (with $1/2 - a = 1/C$). Thus, $\mathcal{B}$ can be simulated giving a final error of

$$\text{err}_{D_f}(h) \leq \text{opt}_{D_f}(\mathcal{H})/(1 - 2a) + \epsilon = (1/2)\text{opt}_{D_f}(\mathcal{H})/(1/2 - a) + \epsilon = (C/2)\text{opt}_{D_f}(\mathcal{H}) + \epsilon.$$

This completes our proof. $\square$

4 Efficient Weak Partial Learning of Weakly Learnable Classes

By Theorem 3.2, to agnostically partially learn a hypothesis class $\mathcal{H}$ under logconcave distributions, it suffices to design an efficient weak partial learner. Recall that our focus is on concept classes $\mathcal{H}$ that are efficiently (k, Δ, β, C) -weakly learnable by an algorithm $\mathcal{A}$. In particular, this means that so long as the effective distribution D_f has marginals whose low degree central moments approximately agree with those of some logconcave distribution (and in particular, if they agree with $D_{\mathbf{x}}$'s central moments), then $\mathcal{A}$ will learn a hypothesis with β -advantage over D_f .

It is not hard to see that simply running $\mathcal{A}$ over the distribution D_f is not a good strategy, as the moment information of $(D_f)_{\mathbf{x}}$ could significantly deviate from that of the requirement of $\mathcal{A}$, even if 90% of the examples have non-zero labels. We will need to find a way to reduce to the case where these central moments do approximately match. Our basic strategy will be to find a direction in which our moments fail to match if one exists, and use this to partition space into two halfspaces on which the masses of $D_{\mathbf{x}}$ and $(D_f)_{\mathbf{x}}$ substantially disagree. Doing this iteratively, we produce finer and finer partitions of $\mathbb{R}^d$ on which D_f becomes more and more concentrated. We show that eventually this procedure must terminate in a partition where almost all of the mass of D lies in regions where our moment matching assumption does apply. We can then use our (k, Δ, β, C) -weak learner on each region to get the desired result.

Unfortunately, to even get started on this project, we need to be able to either certify that $D_{\mathbf{x}}$ and $(D_f)_{\mathbf{x}}$ have nearly matching central moments or to find a direction that exhibits this moment mismatch. This is itself a complicated algorithmic problem that we will put off solving until Section 5. For the moment, we will be satisfied by formally defining the tool that we need:

Definition 4.1 (Moment Discrepancy Finder). *For $k \in \mathbb{Z}_+$, $\Delta > 0$, $c, \alpha \in (0, 1)$, we say that an algorithm $\mathcal{A}$ is a (k, Δ, c, α) -discrepancy finder if it satisfies the following performance guarantee: given sample access to a $1/2$ -isotropic logconcave distribution $D_{\mathbf{x}}$ over $\mathbb{R}^d$ and a distribution $D'_{\mathbf{x}}$, a conditional distribution of $D_{\mathbf{x}}$ over some event with probability mass at least α , if there exists $\mathbf{w} \in \mathbb{S}^{d-1}$ and $j \in [k]$ such that $|\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(\mathbf{w} \cdot \mathbf{x})^j] - \mathbf{E}_{\mathbf{x} \sim D'_{\mathbf{x}}}[(\mathbf{w} \cdot \mathbf{x})^j]| \geq \Delta$, then with probability at least 99%, the algorithm outputs $\mathbf{v} \in \mathbb{S}^{d-1}$ and $i \in [k]$ such that $|\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(\mathbf{v} \cdot \mathbf{x})^i] - \mathbf{E}_{\mathbf{x} \sim D'_{\mathbf{x}}}[(\mathbf{v} \cdot \mathbf{x})^i]| \geq c\Delta$.*

Our main result in this section is the following.

Theorem 4.2 (Weak Learning+Discrepancy Finder→Weak Partial Learning). *Let $\epsilon, \eta \in (0, 1)$, and let $\mathcal{A}_w$ be a (k, Δ, β, C) -weak learner for $\mathcal{H}$ with sample complexity M_w and time complexity T_w . Let $\mathcal{A}_m$ be a (k, Δ, c, α) -discrepancy finder, for an appropriate value of $\alpha = \eta \epsilon^{O_{k, \Delta}(1)}$ with sample complexity M_m and time complexity T_m . Then there is an algorithm $\mathcal{A}$ that satisfies the following: For every input distribution D over $\mathbb{R}^d \times \{\pm 1, 0\}$, with $D_{\mathbf{x}}$ logconcave, $W(D) > \epsilon$ and $\text{opt}_{D_f}(\mathcal{H}) < (1 - \eta)/C$, then $\mathcal{A}$ draws at most $N(M_w + M_m)$ samples from D , runs in time at most $N(T_w + T_m)$ time, where*

$$N = (c\Delta)^{-4k} (k \log k + \log(C/(c\Delta)))^{4k^2} (\eta\epsilon)^{-O((c\Delta)^{-2}(k \log k + \log(C/(c\Delta)))^{2k})}.$$

Algorithm $\mathcal{A}$ then outputs a hypothesis h such that $\text{err}_{D_f}(h) \leq 1/2 - \Omega(\eta\beta)$ with probability at least 99%.

As discussed, to prove Theorem 4.2, we will need to iteratively decompose $\mathbb{R}^d$ into smaller and smaller convex sets. In order to ensure that this procedure produces the desired results, we will need to assume that each split satisfies some conditions that force the distributions of D and D_f across the two halves to differ sufficiently. In particular, we will formalize the condition we need in the following definition:

Definition 4.3 (Good Split). *Let D and D' be two distributions over $\mathbb{R}$ such that $D' = D_A$ for some event A , with $\Pr_D[A] \geq \alpha$. We say that $t \in \mathbb{R}$ is a $(k, \tau, \gamma, \Delta)$ -good split for D, D' , if $|t| < T := O(k \log k + \log(1/\Delta\alpha))$, and for R either $\{z \in \mathbb{R} \mid z > t\}$ or $\{z \in \mathbb{R} \mid z < t\}$, one of the following properties hold:*

1. $|t| \leq \tau$ and $|\Pr_D[R] - \Pr_{D'}[R]| > \gamma$.
2. $|t| > \tau$, $\Pr_D[R] \leq (\Pr_{D'}[R])^4$ and $\Pr_{D'}[R] > |t|^{-2k}$.

Not only does a good split allow one to subdivide the line into two sets on which the probabilities of D and D' differ substantially, we show that if, in the case we are interested in, we can find a direction in which the moments of D and D_f disagree substantially, that we can produce a good split in this direction.

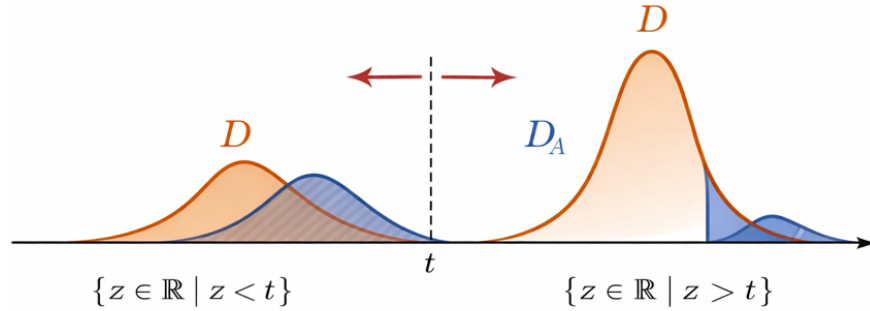


Figure 2: Illustration of a good split.

Lemma 4.4. *Let D be a $1/2$ -isotropic logconcave distribution over $\mathbb{R}$, A an event with probability $\alpha > 0$. Let D' be the conditional distribution of D over A . Suppose that $|\mathbf{E}_{z \sim D}[z^k] - \mathbf{E}_{z' \sim D'}[(z')^k]| > \Delta$, for some $\Delta > 0$ and $k \in \mathbb{N}$. Then there exists $t \in \mathbb{R}$ that is a $(k, \tau, \gamma, \Delta)$ -good split for D, D' , with*

$$\tau := \max\{(\Delta/20)^{-1/k}, 10(k \log k + \log(C/\Delta))\}, \gamma := \tau^{-k} \Delta/100,$$

where C is the constant from Fact A.2. Furthermore, there is an algorithm that on input k, Δ, α , oracle access to D and D' , and $\delta > 0$, draws $\text{poly}(1/\gamma, 1/\alpha, \log(1/\delta))$ samples from D , runs in sample-polynomial time, and outputs a $(k, \tau, \gamma, \Delta)$ -good split with probability at least $1 - \delta$.

Lemma 4.4 says that if the lower order moments of D and D' do not match, then a good split always exists, and we can efficiently find such a good split through a brute-force search. In particular, we want to remark that the existence of such a good split does not hold when D is an arbitrary distribution. In our proof, we show the existence of such a good split by making use of the fact that a logconcave distribution has a sub-exponential tail probability.

The basic strategy of the proof is as follows. We know that D and D' have some low order moment that has substantial disagreement. We note that as these moments can be expressed as suitable integrals of the cumulative density functions that there must be some t for which those cumulative densities differ. We have only to show that such t satisfy the particular conditions of a good split. We first note that since D has subexponential tails that the contribution to its moments from t with $|t| \gg k \log(k) + \log(1/\Delta\alpha)$ is small enough that we can ignore these points. We then need to carefully analyze the integral over the remaining points to show that unless some t satisfies either condition 1 or 2 that the moment discrepancy will be less than what is promised.

Proof of Lemma 4.4. We first show the existence of such a good split. To do this, suppose that for every $|t| < \tau$, we have $|\mathbf{Pr}_{z \sim D}[z \in R] - \mathbf{Pr}_{z' \sim D'}[z' \in R]| < 2\gamma$. We will show that we can find some $|t| \leq O(k \log k + \log(1/(\Delta\alpha)))$ that satisfies the second condition of a good split. We deal with the case where k is odd, and the case where k is even can be analyzed symmetrically. Notice that

$$\begin{aligned} \mathbf{E}_{z \sim D}[z^k] &= \int_0^\infty \mathbf{Pr}_{z \sim D}[z^k > s] ds - \int_0^\infty \mathbf{Pr}_{z \sim D}[z^k < -s] ds \\ &= \int_0^\infty k s^{k-1} \mathbf{Pr}_{z \sim D}[z > s] ds - \int_0^\infty k s^{k-1} \mathbf{Pr}_{z \sim D}[z < -s] ds. \end{aligned} \quad (1)$$

We first show that those $|z'| > T := 40(k \log k + \log(C/(\Delta\alpha)))$ only contribute little to the k th moment of z' . By Fact A.2, we know that there is some universal constant $C > 0$ such that for every $t > T$,

$$\mathbf{Pr}_{z' \sim D'}[|z'| > t] = \mathbf{Pr}_{z \sim D}[|z| > t, A]/\alpha \leq \mathbf{Pr}_{z \sim D}[|z| > t]/\alpha \leq C e^{-t/2}/\alpha.$$

This implies that

$$\int_T^\infty k s^{k-1} \mathbf{Pr}_{z' \sim D'}[|z| > s] ds \leq O\left(k T^{k-1} \exp(-T/2)/\alpha\right) \leq \Delta/100.$$

On the other hand, since for every $t, |t| < \tau$, $|\mathbf{Pr}_{z \sim D}[|z| > t] - \mathbf{Pr}_{z' \sim D'}[|z'| > t]| < \gamma$, we have

$$\begin{aligned} \left| \int_0^\tau k s^{k-1} \left(\mathbf{Pr}_{z \sim D}[z > s] - \mathbf{Pr}_{z' \sim D'}[z' > s] \right) ds \right| &\leq \tau^k \gamma \leq \Delta/100, \\ \left| \int_0^\tau k s^{k-1} \left(\mathbf{Pr}_{z \sim D}[z < -s] - \mathbf{Pr}_{z' \sim D'}[z' < -s] \right) ds \right| &\leq \tau^k \gamma \leq \Delta/100. \end{aligned}$$

By (1), we know that

$$\int_{|s| \in [\tau, T]} k t^{k-1} \left| \mathbf{Pr}_{z \sim D}[z > s] - \mathbf{Pr}_{z' \sim D'}[z' > s] \right| ds \geq \Delta/2. \quad (2)$$

Assuming that for every $t \in [\tau, T]$, $\mathbf{Pr}_{z' \sim D'}[|z'| > t] < t^{-2k}$, would imply that

$$\int_{\tau}^T ks^{k-1} \mathbf{Pr}_{z' \sim D'}[|z'| > s] ds \leq \int_{\tau}^T ks^{-(k+1)} ds / 10 = \tau^{-k} \leq \Delta / 10.$$

On the other hand,

$$\int_{\tau}^T ks^{k-1} \mathbf{Pr}_{z \sim D}[|z| > s] ds \leq C \int_{\tau}^T ks^{k-1} e^{-s/2} ds \leq 2C\tau^{k-1} \exp(-\tau/2) \leq \Delta / 10.$$

Combining these with the equivalent equations for negative s imply that the left hand side of (2) is at most $2\Delta/5$, which is a contradiction. This implies that there is some t with $|t| \in [\tau, T]$ such that $\mathbf{Pr}_{z' \sim D'}[|z'| > t] > t^{-2k}$. In particular, since $t > \tau > 10(k \log k + \log(C))$, we know that $\mathbf{Pr}_{z \sim D}[|z| > t] \leq Ce^{-t} \leq (\mathbf{Pr}_{z' \sim D'}[|z'| > t])^4$. This shows that there is always a $(k, \tau, \gamma, \Delta)$ -good split. In fact, reviewing these calculations, we see that there exists a t that forms a good split so that either in case 1, we have the difference in probabilities at least 2γ or in case 2, we have that $\mathbf{Pr}_{D'}[R] > (1.1)|t|^{-2k}$ and $\mathbf{Pr}_D[R] < (\mathbf{Pr}_{D'}[R]/1.1)^4$. This means that in order to find a good split, it suffices to estimate the tail probabilities $\mathbf{Pr}_{z \sim D}[z > t]$ and $\mathbf{Pr}_{z' \sim D'}[z' > t]$ to error $T^{-8k}/10$ for all t . However, a VC dimension argument says that this is accomplished by taking the empirical distribution over $O(T^{16k})$ samples. $\square$

We next give an overview of how to make use of the discrepancy finding algorithm together with Lemma 4.4. Recall that the (k, Δ, β, C) -weakly learning algorithms $\mathcal{A}_w$ work if the input distribution (after some suitable affine transformation) approximately matches each of its first k moments with some approximate isotropic logconcave distribution. Thus, unless there is such a large moment discrepancy, we can simply run this algorithm. If there is a discrepancy, this can be detected by our discrepancy finder, which will also demonstrate a particular direction v in which the moments fail to match. It will then be the case that $\langle v, D \rangle$ and $\langle v, D_f \rangle$ will be real random variables with a moment mismatch to which we can apply Lemma 4.4 to find a good split. This will allow us to subdivide $\mathbb{R}^d$ into two halfspaces so that the restrictions of D to each halfspace will have somewhat different weights.

From here we produce an iterative decomposition. We begin with a single subdivision consisting of all of $\mathbb{R}^d$. Then iteratively for each subdivision R that $(D_f)_{\mathbf{x}}$ assigns enough mass we run our discrepancy finding algorithm on D and D_f conditioned on R . If it finds no discrepancy, we run our weak learner on that region. Otherwise, our discrepancy finder has found a moment discrepancy along some direction and we can run Lemma 4.4 to find a good split to subdivide R . This idea is summarized in Algorithm 1.

We give an overview of the analysis of Algorithm 1, before giving the full proof. The analysis will follow two parts.

Firstly, we show that the number of rounds of the algorithm is bounded. For this, we note that for each round that we did not terminate, we must have subdivided at least one region in the previous round. Thus, the total number of rounds is at most $|\mathcal{U}_t|$. On the other hand, whenever we use a good split for a region R , the resulting regions R_1 and R_2 will have $\mathbf{Pr}_{D_{\mathbf{x}}}[R_i]$ not too much smaller than $\mathbf{Pr}_{D_{\mathbf{x}}}[R]$. Since $\mathbf{Pr}_{(\mathbf{x}, y) \sim D_f}[\mathbf{x} \in R]$ could not have been less than ξ (or we would not have considered R for subdivision), this means that the mass $D_{\mathbf{x}}$ assigns to each region must be bounded below. This in turn gives an upper bound on the number of regions and rounds.

In the second part, we need to show that the returned hypothesis has reasonably large advantage. We note that for any region R in our final partition with $\mathbf{Pr}_{(\mathbf{x}, y) \sim D_f}[\mathbf{x} \in R] \geq \xi$, we must have small central moment discrepancies and so in any such region where $\text{opt} < 1/C$, our weak learner will

Algorithm 1 K-MOMENTDECISIONTREE (Constructs A Decision Tree as A Weak Hypothesis)

- 1: **Input:** Sample access from D , minimum weight parameter $\eta > 0$, accuracy parameter $\epsilon \in (0, 1)$, moment discrepancy $\Delta > 0$ and advantage $\beta > 0$, access to (k, Δ, β, C) -weakly learning algorithm $\mathcal{A}_w$ and (k, Δ, c, α) -discrepancy finder $\mathcal{A}_m$ with $\alpha = \eta\epsilon^{C_{k,\Delta}}$, where $C_{k,\Delta}$ is a large enough constant given k and Δ .
 - 2: **Output:** $\hat{h}$: weak hypothesis for D_f and $\mathcal{U}$, a partition of $\mathbb{R}^d$
 - 3: Let τ, γ be parameters in the statement of Lemma 4.4 as a function of $c\Delta$ and k .
 - 4: Let $\xi = (\eta\epsilon/2)^{C'/\gamma^2}$ for C' a sufficiently large constant.
 - 5: Define T as in the statement of Definition 4.3 as a function of $c\Delta, k$ and $\alpha = \xi$.
 - 6: Let $\mathcal{U}_0 = \{\emptyset\}, \mathcal{U}_1 = \{\mathbb{R}^d\}, t = 1$.
 - 7: **while** $\mathcal{U}_t \neq \mathcal{U}_{t-1}$ **do**
 - 8: $\mathcal{U}_{t+1} = \emptyset$.
 - 9: **for** $R \in \mathcal{U}_t$ **do**
 - 10: Let D_R be the conditional distribution of D over R , normalizing to 1/2-isotropic position.
 - 11: **if** $(\Pr_{(\mathbf{x}, y) \sim D_f}[\mathbf{x} \in R] < \xi)$ **then** add R to $\mathcal{U}_{t+1}$
 - 12: **else**
 - 13: run $\mathcal{A}_m$ over D_R and $(D_R)_f$ to get a direction $\mathbf{u} \in \mathbb{S}^{d-1}$ such that $\left| \mathbf{E}_{(\mathbf{x}, y) \sim D_R}(\mathbf{u} \cdot \mathbf{x})^i - \mathbf{E}_{(\mathbf{x}, y) \sim (D_R)_f}(\mathbf{u} \cdot \mathbf{x})^i \right| > c\Delta$, for an $i \in [k]$.
 - 14: **if** $\mathbf{u}$ is not returned by $\mathcal{A}_m$ **then** add R to $\mathcal{U}_{t+1}$
 - 15: **else**
 - 16: find a $(i, \tau, \gamma, c\Delta)$ -good split for $(\mathbf{u} \cdot \mathbf{x}), (\mathbf{u} \cdot \mathbf{x}')$, where $\mathbf{x} \sim (D_R)_\mathbf{x}$ and $\mathbf{x}' \sim ((D_R)_f)_\mathbf{x}$ and add R_1, R_2 , the two regions split from R to $\mathcal{U}_{t+1}$.
 - 17: $t = t + 1$.
 - 18: **for** $R \in \mathcal{U}_t$ **do**
 - 19: If $\Pr_{(\mathbf{x}, y) \sim D_f}[\mathbf{x} \in R] \geq \xi$, run $\mathcal{A}_w$ on D_f conditioned on $\mathbf{x} \in R$. Estimate the error of this hypothesis to within $\beta/3$ and if the result is less than $1/2 - 2\beta/3$, let h_R be the returned hypothesis. If $\Pr_{(\mathbf{x}, y) \sim D_f}[\mathbf{x} \in R] < \xi$ or the error of the returned hypothesis is too large, let h_R be a random function.
 - 20: Return the weak hypothesis $\hat{h}(\mathbf{x}) = h_R(\mathbf{x})$ if $\mathbf{x} \in R$, for $R \in \mathcal{U}_t$, and the partition $\mathcal{U} = \mathcal{U}_t$.
-

return a hypothesis with advantage at least β . For the remaining regions, our random hypothesis will have error at most $1/2$. We therefore, need to establish that the total mass of the low weight regions is small. In particular, we need the total mass of these regions to be noticeably smaller than η , so that as long as $\text{opt} < (1 - \eta)/C$, there will still be some heavy regions with $\text{opt} < 1/C$. To do this we will do by instead bounding $\sum_{R \in \mathcal{U}_t} \Pr_{(\mathbf{x}, y) \sim D_f} [\mathbf{x} \in R]^{1-2s}$ for some small, positive constant s . This in turn can be bounded by considering the potential function

$$\Phi_t := \sum_{R \in \mathcal{U}_t} \left(\Pr_{(\mathbf{x}, y) \sim D_f} [\mathbf{x} \in R] \Pr_{(\mathbf{x}, y) \sim D} [\mathbf{x} \in R] \right)^{1/2-s}.$$

In particular, we show that this potential is non-increasing so long as s is at most a small multiple of γ^2 . As $\Phi_0 = 1$, this provides an appropriate bound on the small mass regions.

We summarize the above properties of the partition $\mathcal{U}$ output by Algorithm 1 as Lemma 4.5 and use it to prove Theorem 4.2.

Lemma 4.5. *For $k \in \mathbb{Z}_+$, $\epsilon, \eta, c \in (0, 1)$ and $\Delta > 0$, Let T, γ, τ , and ξ be as in Algorithm 1.*

If Algorithm 1 is run on a distribution D over $\mathbb{R}^d \times \{\pm 1, 0\}$, with $D_{\mathbf{x}}$ logconcave with $W(D) > \epsilon$, then upon termination, with probability at least 99%, the final partition $\mathcal{U}$ of $\mathbb{R}^d$ satisfies:

1. $|\mathcal{U}| \leq O(T^{2k}/(c\Delta\xi\epsilon)).$
2. For every $R \in \mathcal{U}$ with $\Pr_{(\mathbf{x}, y) \sim D_f} [\mathbf{x} \in R] > \xi$, for every $\mathbf{u} \in \mathbb{S}^{d-1}$ and for every $i \in [k]$

$$\left| \mathbf{E}_{(\mathbf{x}, y) \sim D_R} [(\mathbf{u} \cdot \mathbf{x})^i] - \mathbf{E}_{(\mathbf{x}, y) \sim (D_R)_f} [(\mathbf{u} \cdot \mathbf{x})^i] \right| \leq \Delta.$$
3. $\sum_{R \in \mathcal{B}} \Pr_{(\mathbf{x}, y) \sim D_f} [\mathbf{x} \in R] \leq \eta/2$, where $\mathcal{B} = \{R \in \mathcal{U} \mid \Pr_{(\mathbf{x}, y) \sim D_f} [\mathbf{x} \in R] \leq \xi\}.$

Proof of Lemma 4.5. We start by showing the first item in the statement that $|\mathcal{U}| \leq O(T/(c\Delta\xi\epsilon)).$ For each time step t , let $R \in \mathcal{U}_t$. If $R \notin \mathcal{U}_{t-1}$, it must be the case, where R is split from some $R' \in \mathcal{U}_{t-1}$ using a $(i, \tau, \gamma, c\Delta)$ good split, with $i \in [k]$. This implies that $\Pr_{(\mathbf{x}, y) \sim D_f} [\mathbf{x} \in R'] > \xi$ and $\Pr_{(\mathbf{x}, y) \sim D} [\mathbf{x} \in R'] > \epsilon\xi$.

By the definition of an $(i, \tau, \gamma, c\Delta)$ good split, either $\Pr_{D_{\mathbf{x}}}[R \mid R'] > \gamma$ or $\Pr_{D_{\mathbf{x}}}[R \mid R'] > T^{-2k}$. Since $c\Delta T^{-2k} < \gamma$, in the first case, we have $\Pr_{D_{\mathbf{x}}}[R] > c\Delta T^{-2k} \Pr_{D_{\mathbf{x}}}[R'] > c\Delta T^{-2k} \xi\epsilon$. In the second case, $\Pr_{D_{\mathbf{x}}}[R] \geq \epsilon \Pr_{(D_f)_{\mathbf{x}}}[R] > \epsilon T^{-2k} \Pr_{(D_f)_{\mathbf{x}}}[R'] > T^{-2k} \xi\epsilon$. Since $\sum_{R \in \mathcal{U}_t} \Pr_{(\mathbf{x}, y) \sim D} [\mathbf{x} \in R] = 1$ for each time step t , we know that $|\mathcal{U}| \leq O(T^{2k}/(c\Delta\xi\epsilon)).$

Next, we show the second item in the statement. Let $R \in \mathcal{U}$ be a region with $\Pr_{(D_f)_{\mathbf{x}}}[R] > \xi$. Notice that by the definition of $\mathcal{A}_m$, if there is a direction $\mathbf{u} \in \mathbb{S}^{d-1}$ and some $i \in [k]$ such that $\left| \mathbf{E}_{(\mathbf{x}, y) \sim D_R} [(\mathbf{u} \cdot \mathbf{x})^i] - \mathbf{E}_{(\mathbf{x}, y) \sim (D_R)_f} [(\mathbf{u} \cdot \mathbf{x})^i] \right| > \Delta$, then some direction $\mathbf{w}$ will be output and R will be split into two regions, which contradicts the fact that Algorithm 1 terminates.

Finally, we show the third item in the statement. To do this, we define a potential function

$$\Phi_t := \sum_{R \in \mathcal{U}_t} \left(\Pr_{(\mathbf{x}, y) \sim D} [\mathbf{x} \in R] \Pr_{(\mathbf{x}, y) \sim D_f} [\mathbf{x} \in R] \right)^{1/2-s}.$$

We show that for the choice of $s = \gamma^2/10$, $\Phi_{t+1} \leq \Phi_t$ for all t .

It suffices to show that for any $R \in \mathcal{U}_t$ that is split into regions R_1, R_2 ,

$$\left(\Pr_{(\mathbf{x}, y) \sim D} [\mathbf{x} \in R] \Pr_{(\mathbf{x}, y) \sim D_f} [\mathbf{x} \in R] \right)^{1/2-s} \geq \sum_{i=1,2} \left(\Pr_{(\mathbf{x}, y) \sim D} [\mathbf{x} \in R_i] \Pr_{(\mathbf{x}, y) \sim D_f} [\mathbf{x} \in R_i] \right)^{1/2-s}$$

To simplify the notation, we denote by $a := \Pr_{(\mathbf{x}, y) \sim D}[\mathbf{x} \in R_1 \mid \mathbf{x} \in R]$ and $b := \Pr_{(\mathbf{x}, y) \sim D_f}[\mathbf{x} \in R_1 \mid \mathbf{x} \in R]$. Notice that

$$\begin{aligned} & \sum_{i=1,2} \left(\Pr_{(\mathbf{x}, y) \sim D}[\mathbf{x} \in R_i] \Pr_{(\mathbf{x}, y) \sim D_f}[\mathbf{x} \in R_i] \right)^{\frac{1}{2}-s} \\ &= \left(\Pr_{(\mathbf{x}, y) \sim D}[\mathbf{x} \in R] \Pr_{(\mathbf{x}, y) \sim D_f}[\mathbf{x} \in R] \right)^{\frac{1}{2}-s} \left((ab)^{1/2-s} + ((1-a)(1-b))^{\frac{1}{2}-s} \right) \end{aligned}$$

We next show that under an $(i, \tau, \gamma, c\Delta)$ -good split, $\left((ab)^{1/2-s} + ((1-a)(1-b))^{\frac{1}{2}-s} \right) \leq 1$. We prove this by considering cases for a good split. In the first case in the definition of $(i, \tau, \gamma, \Delta)$ -good split, we have $|a - b| > \gamma$. In this case, we have

$$\begin{aligned} \left((ab)^{1/2-s} + ((1-a)(1-b))^{\frac{1}{2}-s} \right) &= \left(\left(\left(\frac{a+b}{2} \right)^2 - \left(\frac{a-b}{2} \right)^2 \right)^{\frac{1}{2}-s} + \left(\left(1 - \frac{a+b}{2} \right)^2 - \left(\frac{a-b}{2} \right)^2 \right)^{\frac{1}{2}-s} \right) \\ &\leq 2 \left(\frac{1}{4} - \left(\frac{a-b}{2} \right)^2 \right)^{\frac{1}{2}-s} \leq 2 \left(\frac{1-\gamma^2}{4} \right)^{\frac{1}{2}-s} \leq 1. \end{aligned}$$

Here in the first inequality, we use the fact that, fix the value of $(a-b)/2$, the summation is maximized when $(a+b)/2 = 1/2$, while in the second inequality, we use the condition that $|a-b| > \gamma$. In the last inequality, we use the choice of $s \leq \gamma^2/10$.

In the second case, by the definition of $(i, \tau, \gamma, c\Delta)$ -good split, we have $a \leq b^4$. Notice that $b < 1/10$, otherwise we are in the first case. We have

$$\begin{aligned} \left((ab)^{1/2-s} + ((1-a)(1-b))^{\frac{1}{2}-s} \right) &= \left((ab)^{\frac{1}{2}-s} + (1-a-b+ab)^{\frac{1}{2}-s} \right) \\ &\leq b^{5(1/2-s)} + 1 - (1/2-s)(a+b-ab) \\ &\leq 1 - (1/2-s)b + 2b^{5(1/2-s)} < 1. \end{aligned}$$

Here, the first inequality follows that for $z \in (0, 1)$, and $p < 1$, $(1-z)^p \leq 1 - pz$, and the second inequality follows the bound $a \leq b^4, b \leq 1/10$.

So, we have $\Phi_{t+1} \leq \Phi_t$ for all t . Using this, we know that

$$\begin{aligned} \Phi_0 = 1 &\geq \sum_{R \in \mathcal{B}_t} \left(\Pr_{(\mathbf{x}, y) \sim D}[\mathbf{x} \in R] \Pr_{(\mathbf{x}, y) \sim D_f}[\mathbf{x} \in R] \right)^{1/2-s} \\ &\geq \left(\Pr_{(\mathbf{x}, y) \sim D}[y \neq 0] \right)^{1/2-s} \sum_{R \in \mathcal{B}_t} \left(\Pr_{(\mathbf{x}, y) \sim D_f}[\mathbf{x} \in R] \right)^{1-2s} \\ &\geq \left(\Pr_{(\mathbf{x}, y) \sim D}[y \neq 0] \right)^{1/2-s} \sum_{R \in \mathcal{B}_t} \left(\Pr_{(\mathbf{x}, y) \sim D_f}[\mathbf{x} \in R] \right) \xi^{-2s}, \end{aligned}$$

which gives us

$$\sum_{R \in \mathcal{B}_t} \left(\Pr_{(\mathbf{x}, y) \sim D_f}[\mathbf{x} \in R] \right) \leq \left(\Pr_{(\mathbf{x}, y) \sim D}[y \neq 0] \right)^{s-1/2} \xi^{2s} \leq \eta/2,$$

where the last inequality holds when $\xi = (\eta\epsilon/2)^{1/2s} = (\eta\epsilon/2)^{O(1/\gamma^2)}$. $\square$

With Lemma 4.5 at hand, we can immediately prove Theorem 4.2.

Proof of Theorem 4.2. We first show the correctness of Algorithm 1 by showing that it outputs a hypothesis with $\Omega(\eta\beta)$ advantage with respect to D_f . Let $\mathcal{B} = \{R \in \mathcal{U} \mid \Pr_{(\mathbf{x},y) \sim D_f}[\mathbf{x} \in R] \leq \xi\}$. We denote by $\mathcal{G} = \mathcal{U} \setminus \mathcal{B}$ and denote by $\mathcal{M} \subseteq \mathcal{G}$ the set of R , where a random binary function h_R is assigned. We show that $\sum_{R \in \mathcal{M}} \Pr_{(\mathbf{x},y) \sim D_f}[\mathbf{x} \in R]$ is not too large. By Item 2 in Lemma 4.5, it holds for every $R \notin \mathcal{B}$ and $\mathbf{u} \in \mathbb{S}^{d-1}$ and for every $i \in [k]$, $\left| \mathbf{E}_{(\mathbf{x},y) \sim D_R}(\mathbf{u} \cdot \mathbf{x})^i - \mathbf{E}_{(\mathbf{x},y) \sim (D_R)_f}(\mathbf{u} \cdot \mathbf{x})^i \right| \leq \Delta$. If $\mathcal{A}$ fails to output a weak hypothesis over R , it must be the case, where we have $\Pr_{(\mathbf{x},y) \sim D_f}[h^*(\mathbf{x}) \neq y \mid \mathbf{x} \in R] > 1/C$, where h^* is the hypothesis in $\mathcal{H}$ that achieves $\text{opt}_{D_f}(\mathcal{H})$. Now suppose that $\sum_{R \in \mathcal{M}} \Pr_{(\mathbf{x},y) \sim D_f}[\mathbf{x} \in R] > (1 - \eta)$, then we have

$$\text{err}_{D_f}(h^*) \geq \sum_{R \in \mathcal{M}} \Pr_{(\mathbf{x},y) \sim D_f}[\mathbf{x} \in R] \Pr_{(\mathbf{x},y) \sim D_f}[h^*(\mathbf{x}) \neq y \mid \mathbf{x} \in R] > (1 - \eta)/C,$$

which gives a contradiction to the fact that $\text{opt}_{D_f} \leq (1 - \eta)/C$. By Item 3 of Lemma 4.5, $\sum_{R \in \mathcal{B}} \Pr_{(\mathbf{x},y) \sim D_f}[\mathbf{x} \in R] \leq \eta/2$. Thus, we have $\sum_{R \in \mathcal{G} \setminus \mathcal{M}} \Pr_{(\mathbf{x},y) \sim D_f}[\mathbf{x} \in R] > \eta/2$. Since for each $R \in \mathcal{G} \setminus \mathcal{M}$, we have $\Pr_{(\mathbf{x},y) \sim D_f}[h_R(\mathbf{x}) \neq y \mid R] < 1/2 - \beta/3$, (and the error on all other regions is $1/2$) we know that $\text{err}_{D_f}(\hat{h}) \leq 1/2 - \Omega(\eta\beta)$.

Next, we bound the sample complexity and the time complexity of Algorithm 1. Observe that every time Algorithm 1 makes a split, it adds one more region to the partition $\mathcal{U}_t$. This implies Algorithm 1 will terminate after making $|\mathcal{U}| - 1$ splits. By Item 1 of Lemma 4.5, it will run $\mathcal{A}_w, \mathcal{A}_m$ and the algorithm for finding a good split at most $O(T^{2k}/(c\Delta\xi\epsilon))$ times for regions R created by Algorithm 1. Since $\Pr_{(\mathbf{x},y) \sim D}[\mathbf{x} \in R] > \xi\epsilon$ and $\Pr_{(\mathbf{x},y) \sim D_f}[\mathbf{x} \in R] > \xi$, an example drawn from $D_R, (D_R)_f$ can be done by drawing $(\epsilon\xi)^{-1} = (\eta\epsilon)^{-O(\gamma^{-2})}$ examples from D . So, each call to $\mathcal{A}_w$ draws $(\epsilon\xi)^{-1}M_w$ sample from D and runs in $(\epsilon\xi)^{-1}T_w$ time. Similarly, each call to $\mathcal{A}_m$ draws $(\epsilon\xi)^{-1}M_m$ sample from D and runs in $(\epsilon\xi)^{-1}T_m$ time. Each call to the algorithm for finding a good split takes $\text{poly}(\xi^{-1})$ sample and time. Thus, the sample complexity of Algorithm 1 is $O(T^{2k}/(c\Delta\xi^2\epsilon))(M_w + M_m)$ and the time complexity of Algorithm 1 is $O(T^{2k}/(c\Delta\xi^2\epsilon))(T_w + T_m)$.

Recall that $\gamma = c\Delta(k \log k + \log(C/c\Delta))^{-k}$, $\xi = (\eta\epsilon)^{O(1/\gamma^2)}$ and $T = O(k \log k + \log(1/\Delta\xi))$. This gives $T^{2k}/(c\Delta\xi^2\epsilon) = N = (c\Delta)^{-4k}(k \log k + \log(C/(c\Delta)))^{4k^2}(\eta\epsilon)^{-O((c\Delta)^{-2}(k \log k + \log(C/(c\Delta)))^{2k})}$. $\square$

5 Finding Moment Discrepancies under Logconcave Distributions

By Theorem 4.2, an important step towards a weak learner for agnostic partial learning (Definition 2.7), is to design a moment discrepancy finder (Definition 4.1). For the simplest case of $k = 2$, one can leverage mean estimation and singular-value decomposition to achieve this goal. Unfortunately, for $k > 3$, this task appears challenging in general. For example, the straightforward way to achieve this goal requires approximating eigenvectors of higher-order moment tensors, which is NP-hard [HL13] in general.

We instead show that if the distribution $D_{\mathbf{x}}$ is logconcave and $(D_f)_{\mathbf{x}}$ is the conditional distribution over an event A with probability α there is a $(k, \Delta, 2^{-O(k^2)}, \alpha)$ -discrepancy finder that runs in $\text{poly}_{k,\Delta}(d)2^{\text{polylog}_{k,\Delta}(\alpha^{-1})}$ time.

Specifically, the main algorithmic result of this section is the following.

Theorem 5.1 (Moment Discrepancy Finder for Logconcave Distributions). *There is an algorithm with the following performance guarantee. Given sample access to a $1/2$ -isotropic logconcave distribution $D_{\mathbf{x}}$ over $\mathbb{R}^d$, oracle access to an event A with $\Pr_{D_{\mathbf{x}}}[A] = \alpha$, $k \in \mathbb{Z}_+$ and $\Delta \in (0, 1)$, the algorithm draws $M = d^{O(k)} \alpha^{-2} (\log(1/\alpha)/\Delta)^{2^{O(k)}}$ examples from $D_{\mathbf{x}}$, runs in time*

$$\text{poly}(M) + 2^{(\log(1/\alpha)/\Delta)^{2^{O(k)}}}$$

and satisfies the following:

1. With probability at least 99%, it either outputs $\perp$, or outputs a vector $\mathbf{v} \in \mathbb{S}^{d-1}$ such that $\left| \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(\mathbf{v} \cdot \mathbf{x})^k] - \mathbf{E}_{\mathbf{x} \sim (D_f)_{\mathbf{x}}}[(\mathbf{v} \cdot \mathbf{x})^k] \right| \geq 2^{-O(k^2)} \Delta$.
2. If there exists $\mathbf{w} \in \mathbb{S}^{d-1}$ and $k \in \mathbb{N}$ such that $\left| \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(\mathbf{w} \cdot \mathbf{x})^k] - \mathbf{E}_{\mathbf{x} \sim (D_f)_{\mathbf{x}}}[(\mathbf{w} \cdot \mathbf{x})^k] \right| \geq \Delta$, where $(D_f)_{\mathbf{x}}$ is the conditional distribution of $D_{\mathbf{x}}$ over A , the algorithm outputs $\mathbf{v} \in \mathbb{S}^{d-1}$ with probability at least 99%.

Before going to the technical analysis needed to prove Theorem 5.1, we provide an overview of the algorithm. Our first step is to show that the discrepancy between the higher moment tensors of D and D_f is in some sense small. For this, we note by the Carbery-Wright concentration inequality (Fact A.10), one can show that for any degree- k polynomial $p(\mathbf{x})$, $\left| \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[p(\mathbf{x})] - \mathbf{E}_{\mathbf{x} \sim (D_f)_{\mathbf{x}}}[p(\mathbf{x})] \right|$ is not too large. This implies (see Lemma 5.7), that there must be a degree- k polynomial $Q(\mathbf{x})$, with L_2 -norm at most $O(\log^k(1/\alpha))$ under $D_{\mathbf{x}}$, such that for every degree k polynomial $p(\mathbf{x})$, we have $\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[p(\mathbf{x})] - \mathbf{E}_{\mathbf{x} \sim (D_f)_{\mathbf{x}}}[p(\mathbf{x})] = \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[Q(\mathbf{x})p(\mathbf{x})]$. Given that the difference between the higher moments is in some sense “small”, our plan will be to find a low dimensional subspace V that captures any possible directions of large discrepancy. We can then conclude with an exhaustive search over V .

For the purpose of motivation, we begin by assuming that $D_{\mathbf{x}}$ is a standard Gaussian distribution. In this case, we can expand $Q(\mathbf{x})$ in the Hermite basis, yielding a sequence of tensors $\mathbf{T}_0, \dots, \mathbf{T}_k$, with small Frobenius norm so that for every $\mathbf{v} \in \mathbb{S}^{d-1}$, we can write $\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[Q(\mathbf{x})(\mathbf{v} \cdot \mathbf{x})^k]$ as a linear combination of the $\mathbf{T}_k \cdot \mathbf{v}^{\otimes i}$ for $i \in [k]$. If we flatten each tensor $\mathbf{T}_i$ and view it as a bilinear form $\mathbb{R}^d \times \mathbb{R}^{d^{i-1}} \rightarrow \mathbb{R}$, we can then find singular vectors of each bilinear form that correspond to large singular values. These singular vectors span a subspace V with dimension $\text{poly}_k(\log(1/\alpha)/\Delta)$ such that $\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[Q(\mathbf{x})(\mathbf{v} \cdot \mathbf{x})^k]$ can be approximated as a linear combination of $\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[Q(\mathbf{x})(\mathbf{v}' \cdot \mathbf{x})^i]$, where $\mathbf{v}'$ is the projection of the unit vector $\mathbf{v}$ onto V , plus an error of at most $\Delta/2$. Then in time exponential in $\dim(V)$, we can find such a direction in V near maximal moment discrepancy, which gives an algorithm with sample complexity $\text{poly}(1/\alpha)$ and time complexity quasi-polynomial with respect to $1/\alpha$.

However, the above approach heavily makes use of properties of Gaussian distributions, such as moments in orthogonal directions being independent, which do not hold for a general logconcave distribution. Our plan is still to make use of the above idea of dimension reduction, but with a relaxed goal. In particular, we would like to find a subspace V such that $\sup_{\mathbf{u}^1, \dots, \mathbf{u}^{(k)} \in \mathbb{S}^{d-1}} \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[Q(\mathbf{x}) \prod_{j=1}^k (\mathbf{u}^{(j)} \cdot \mathbf{x})]$ is at most some constant multiple of the maximum when $\mathbf{u}^1, \dots, \mathbf{u}^{(k)}$ are taken from V , plus $\Delta/2$. If such a subspace V is found, then we can brute force search over V , to find a direction with moment discrepancy. We achieve this goal through two steps.

In the first step, Lemma 5.4, we observe that given any polynomial $Q(\mathbf{x})$, the linear function $\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[Q(\mathbf{x})p(\mathbf{x})]$ can be written as $\sum_{i=0}^k \mathbf{T}_i \cdot \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[\nabla^i p(\mathbf{x})]$. By making use of the recent progress on KLS conjecture [KL25], we are able to show in Lemma 5.3 that the Frobenius norm of these tensors $\mathbf{T}_1, \dots, \mathbf{T}_k$ are not large. By flattening these tensors as a bilinear form and find singular

vectors of each bilinear form that correspond to large singular values, we are able to find a subspace V_1 , such that if $\mathbf{u}^1, \dots, \mathbf{u}^{(k)}$ are all orthogonal to V_1 , then $\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q(\mathbf{x}) \prod_{j=1}^k (\mathbf{u}^{(j)} \cdot \mathbf{x})]$ must be very small.

The second step is to recursively build the target subspace using the subroutine in the first step. Given any k unit vectors $\mathbf{u}^1, \dots, \mathbf{u}^{(k)}$, we can write each $\mathbf{u}^{(i)} = \mathbf{u}_{V_1}^{(i)} + \mathbf{u}_{V_1^\perp}^{(i)}$. So, we can expand $\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q(\mathbf{x}) \prod_{j=1}^k (\mathbf{u}^{(j)} \cdot \mathbf{x})]$ into a linear combination of 2^k terms, each of which is the expectation of $Q(\mathbf{x}) \prod_{j=1}^k (\mathbf{v}^{(j)} \cdot \mathbf{x})$, where $\mathbf{v}^{(j)}$ is either $\mathbf{u}_{V_1}^{(j)}$ or $\mathbf{u}_{V_1^\perp}^{(j)}$. We know if $\mathbf{v}^1, \dots, \mathbf{v}^{(k)}$ are all orthogonal to V_1 , then $\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q(\mathbf{x}) \prod_{j=1}^k (\mathbf{v}^{(j)} \cdot \mathbf{x})]$ must be very small. This implies that during the analysis, we can without loss of generality assume that $\mathbf{u}^{(1)} \in V_1$. Specifically, let $\mathbf{g}^{(1)}, \dots, \mathbf{g}^{(N)}$ be a basis for the subspace V_1 , we can write $\mathbf{u}^{(1)} = \sum_{i=1}^N a_i \mathbf{g}^{(i)}$. This implies that

$$\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q(\mathbf{x}) \prod_{j=1}^k (\mathbf{u}^{(j)} \cdot \mathbf{x})] = \sum_{i=1}^N a_i \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q(\mathbf{x}) (\mathbf{g}^{(i)} \cdot \mathbf{x}) \prod_{j=2}^k (\mathbf{u}^{(j)} \cdot \mathbf{x})].$$

Notice that if we apply the subroutine in the first step to each of the $Q(\mathbf{x}) (\mathbf{g}^{(i)} \cdot \mathbf{x})$ and denote by the span of the found subspaces as V_2 , then if $\mathbf{u}^{(j)} \perp V_1 \oplus V_2$ for $j = 2, \dots, k$, then $\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q(\mathbf{x}) \prod_{j=1}^k (\mathbf{u}^{(j)} \cdot \mathbf{x})]$ must be very small. As a result, if we recursively run this approach for k times, it must be the case that the final subspace V approximately maintains the maximum of $\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q(\mathbf{x}) \prod_{j=1}^k (\mathbf{u}^{(j)} \cdot \mathbf{x})]$.

We now give a roadmap of this section. In Section 5.1, we will provide technical lemmas that will be used in our analysis based on the recent progress on KLS conjecture. In Section 5.2, we will give formal analysis of the dimension reduction approach we discussed above. Finally, in Section 5.3, we will make use of the dimension reduction method to give our discrepancy finding algorithm and present the proof of Theorem 5.1.

5.1 Useful Technical Facts

To begin with, we have the following Poincaré inequality for isotropic logconcave distributions, which follows from recent progress on thin-shell bounds for isotropic logconcave distributions.

Lemma 5.2. *Let $D_{\mathbf{x}}$ be a $1/2$ -isotropic logconcave distribution over $\mathbb{R}^d$ and $p : \mathbb{R}^d \rightarrow \mathbb{R}$ be a degree- k polynomial. Then it holds $\mathbf{Var}_{\mathbf{x} \sim D_{\mathbf{x}}} [p(\mathbf{x})] \leq O((\log^2 d) \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\|\nabla p(\mathbf{x})\|_2^2])$.*

Proof of Lemma 5.2. We without loss of generality assume $D_{\mathbf{x}}$ is isotropic. To see this, let $\boldsymbol{\mu} = \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\mathbf{x}]$ and $\boldsymbol{\Sigma}$ be the covariance matrix of $\mathbf{x}$. Then $\mathbf{x}' = \boldsymbol{\Sigma}^{-1/2}(\mathbf{x} - \boldsymbol{\mu})$ is isotropic. Denote by $D_{\mathbf{x}'}$, the distribution of $\mathbf{x}'$ and write $g(\mathbf{x}') = p(\boldsymbol{\Sigma}^{1/2}(\mathbf{x}' + \boldsymbol{\mu}))$. We have $\mathbf{Var}_{\mathbf{x} \sim D_{\mathbf{x}}} [p(\mathbf{x})] = \mathbf{Var}_{\mathbf{x}' \sim D_{\mathbf{x}'}} [g(\mathbf{x}')] = \mathbf{E}_{\mathbf{x}' \sim D_{\mathbf{x}'}} [\|\nabla g(\mathbf{x}')\|_2^2] = \mathbf{E}_{\mathbf{x}' \sim D_{\mathbf{x}'}} [\|\boldsymbol{\Sigma}^{1/2} \nabla p(\boldsymbol{\Sigma}^{1/2}(\mathbf{x}' + \boldsymbol{\mu}))\|_2^2] \leq 2 \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\|\nabla p(\mathbf{x})\|_2^2]$.

By the Poincaré inequality [LV18], we know that there is a constant $C(D) > 0$ such that for every polynomial $p(\mathbf{x})$, we have

$$\mathbf{Var}_{\mathbf{x} \sim D_{\mathbf{x}}} [p(\mathbf{x})] \leq C(D) \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\|\nabla p(\mathbf{x})\|_2^2].$$

By Theorem 1.5 in [Mil09] and Theorem 1.1 in [Eld13], there is a universal constant $C > 0$ such that we have

$$C(D) \leq C \log d \sum_{i=1}^d \frac{\sigma_i^2}{i}. \quad (3)$$

Here, $\sigma_i^2 = \sup_{D_{\mathbf{x}}} \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [(\|\mathbf{x}\|_2 - \sqrt{i})^2]$, where the supremum is taken over all isotropic logconcave distributions over $\mathbb{R}^i$. By Theorem 1.1 in [KL25], there is a universal constant $c > 0$ such that $\sigma_i^2 < c, \forall i$. Thus, by (3), we get

$$\mathbf{Var}_{\mathbf{x} \sim D_{\mathbf{x}}} [p(\mathbf{x})] \leq O(\log^2 d) \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\|\nabla p(\mathbf{x})\|_2^2].$$

This completes the proof. $\square$

Using Lemma 5.2, we establish the following result that bounds the L_2 norm of a polynomial p in terms of the norm of its gradient under a logconcave distribution $D_{\mathbf{x}}$.

Lemma 5.3. *Let $D_{\mathbf{x}}$ be a $1/2$ -isotropic logconcave distribution over $\mathbb{R}^d$ and let $p : \mathbb{R}^d \rightarrow \mathbb{R}$ be a degree- k polynomial. Denote by $F(p) := (\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\nabla^i p(\mathbf{x})])_{i=0}^k$, where $\nabla^i p(\mathbf{x})$ is the i -th order derivative tensor of p . Then $\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [p^2(\mathbf{x})] \leq O(\log d)^{2k} \|F(p)\|_F^2$.*

Proof of Lemma 5.3. We prove Lemma 5.3 using an induction on k . If $k = 0$, we have $\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [p^2(\mathbf{x})] = \|F(p)\|_F^2$. Suppose the statement holds for $k - 1$, we show this for k . By Lemma 5.2, for some universal constant $C > 0$, we have

$$\begin{aligned} \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [p^2(\mathbf{x})] &= \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [p(\mathbf{x})^2] + \mathbf{Var}_{\mathbf{x} \sim D_{\mathbf{x}}} [p(\mathbf{x})] \\ &\leq \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\nabla^0 p(\mathbf{x})^2] + C \log^2 d \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\|\nabla p(\mathbf{x})\|_2^2] \\ &= \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\nabla^0 p(\mathbf{x})^2] + C \log^2 d \mathbf{E}_{\mathbf{v} \sim N(0, \mathbf{I})} \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [(\mathbf{v} \cdot \nabla p(\mathbf{x}))^2] \\ &\leq \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\nabla^0 p(\mathbf{x})^2] + C^{2k-1} \log^{2k} d \mathbf{E}_{\mathbf{v} \sim N(0, \mathbf{I})} [\|F(\mathbf{v} \cdot \nabla p(\mathbf{x}))\|_F^2] \\ &\leq O(\log d)^{2k} \|F(p)\|_F^2. \end{aligned}$$

Here the second inequality follows the induction on the degree $k - 1$ polynomial $\mathbf{v} \cdot \nabla p(\mathbf{x})$. $\square$

5.2 Novel Dimension-Reduction Approach for Logconcave Distributions

In this subsection, we design and analyze a new dimension-reduction approach to obtain our discrepancy finding algorithm. The key module of our dimension-reduction approach is the following result.

Lemma 5.4 (Effective Subspace Identification). *There is an algorithm that, given sample access to any $1/2$ -isotropic logconcave distribution $D_{\mathbf{x}}$ over $\mathbb{R}^d$, has the following performance guarantee. For any given parameters $m, k \in \mathbb{N}$, $\epsilon, \delta \in (0, 1)$, and for any degree m -polynomial $Q : \mathbb{R}^d \rightarrow \mathbb{R}$ with $|\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q(\mathbf{x})]| < \epsilon (dk)^{-O(k)} \|Q(\mathbf{x})\|_{D_{\mathbf{x}}, 2}$, the algorithm draws $M = \text{poly}(d^k, 2^{O(k \log k + m)}, 1/\epsilon, \log(1/\delta))$ samples from $D_{\mathbf{x}}$, runs in $\text{poly}(M)$ time, and outputs a subspace V of dimension $2^{O(k \log k)} \log^{2k}(d)/\epsilon^2$ such that with probability at least $1 - \delta$ the following holds: for every collection of unit vectors $\{\mathbf{u}^{(i)}\}_{i=1}^k$ that are perpendicular to V , we have $\left| \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} \left[Q(\mathbf{x}) \prod_{i=1}^k (\mathbf{u}^{(i)} \cdot \mathbf{x}) \right] \right| < \epsilon \|Q(\mathbf{x})\|_{D_{\mathbf{x}}, 2}$.*

Proof of Lemma 5.4. We show that Algorithm 2 satisfies the statement. To this end, we define the following linear map from P_k —the space of degree- k polynomials over $\mathbb{R}^d$ —to $\bigoplus_{i=0}^k \text{sym}(\mathbb{R}^d)$, the space of vectors, whose i th entry is an i th order symmetric tensor,

$$F(p) := \left(\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\nabla^i p(\mathbf{x})] \right)_{i=0}^k.$$

Algorithm 2 EFFECTIVESUBSPACEIDENTIFYING

- 1: **Input:** Sample access to $D_{\mathbf{x}}$, accuracy parameter $\epsilon \in (0, 1)$, confidence parameter $\delta \in (0, 1)$, degree m -polynomial $Q(\mathbf{x})$ with a small mean, degree parameter k
 - 2: **Output:** V , a low-dimensional subspace
 - 3: Let $\eta = \epsilon 2^{-O(k \log k)} \|Q(\mathbf{x})\|_{D_{\mathbf{x}}, 2}$ and $\eta' = \|Q(\mathbf{x})\|_{D_{\mathbf{x}}, 2} d^{-k} \epsilon / (2\sqrt{k!})$.
 - 4: For a degree- k polynomial p , define $F(p) := (\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\nabla^i p(\mathbf{x})])_{i=0}^k$.
 - 5: For each monomial $\mathbf{x}^\zeta$ of degree k , where ζ is the multi index, use empirical mean estimation to find $\mu_\zeta \in \mathbb{R}$ and a sequence of symmetric tensors Z_ζ such that $|\mu_\zeta - \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q(\mathbf{x}) \mathbf{x}^\zeta]| < \eta'/4$ and $\|Z_\zeta - F(\mathbf{x}^\zeta)\|_F \leq O(\eta' / \log^k(d))$.
 - 6: Find tensor $\hat{\mathbf{T}}_1, \dots, \hat{\mathbf{T}}_k$ such that $\|\hat{\mathbf{T}}_i\|_F \leq 2C \log^k(d)$, and $|\mu_\zeta - \sum_{i=1}^k \hat{\mathbf{T}}_i \cdot Z_\zeta^{(i)}| \leq \eta'/2$, for every ζ , with $|\zeta| = k$, using convex programming.
 - 7: For each $\hat{\mathbf{T}}_i$, treat it as a bilinear form over $\mathbb{R}^d \times \text{sym}(\mathbb{R}^{d^{i-1}})$ and compute the singular vectors $\mathbf{v}^{(i,1)}, \dots, \mathbf{v}^{(i,l)} \in \mathbb{R}^d$ of $\hat{\mathbf{T}}_i$ with corresponding singular value at least η .
 - 8: Return V , the subspace spanned by these singular vectors.
-

We claim that $F(p)$ is an invertible linear map. To see this, consider any polynomial $p(\mathbf{x}) = \sum_{i=0}^k \mathbf{A}_i \cdot \mathbf{x}^{\otimes i}$ such that $F(p) = 0$. Notice that $\nabla^k p(\mathbf{x}) = k! \mathbf{A}_k$. Since $F(p) = 0$, we know that $\nabla^k p(\mathbf{x}) = 0$, which implies $\mathbf{A}_k = 0$. By induction, we know that $\mathbf{A}_i = 0, i \in [k]$ and thus $p(\mathbf{x}) = 0$. So, $F(p)$ is an invertible linear map.

Consider the linear function over P_k , $G(p) := \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q(\mathbf{x}) p(\mathbf{x})] = \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q(\mathbf{x}) F^{-1} \circ F(p)]$ is a linear function over $\text{sym}(\mathbb{R}^{d^k})$. This implies that there exist tensors $\mathbf{T}_i \in \text{sym}(\mathbb{R}^{d^i})$, for $i = 0, \dots, k$ such that

$$\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q(\mathbf{x}) p(\mathbf{x})] = \sum_{i=1}^k \mathbf{T}_i \cdot \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\nabla^i p(\mathbf{x})].$$

In particular, $|\mathbf{T}_0| < \epsilon (dk)^{-O(k)} \|Q(\mathbf{x})\|_{D_{\mathbf{x}}, 2}$, because by setting $p(\mathbf{x}) = 1$, $|\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q(\mathbf{x})]| < \epsilon (dk)^{-O(k)} \|Q(\mathbf{x})\|_{D_{\mathbf{x}}, 2}$. This implies that for every monomial $\mathbf{x}^\zeta$, with degree at most k , $|\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\mathbf{x}^\zeta] \mathbf{T}_0| < \eta'$.

We claim that $\|\mathbf{T}_i\|_F \leq C^k \log^k(d) \|Q\|_{D_{\mathbf{x}}, 2}$ for $i \in [k]$. Suppose this is not correct, since $F(p)$ is an invertible linear map, we choose $p(\mathbf{x})$ such that $F(p) = (\mathbf{T}_i)_{i=0}^k$. By Lemma 5.3, we have

$$\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q(\mathbf{x}) p(\mathbf{x})] \leq \|Q\|_{D_{\mathbf{x}}, 2} \|p\|_{D_{\mathbf{x}}, 2} \leq C^k \log^k(d) \|F(p)\|_F \|Q\|_{D_{\mathbf{x}}, 2}.$$

On the other hand,

$$\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q(\mathbf{x}) p(\mathbf{x})] = \sum_{i=0}^k \mathbf{T}_i \cdot \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\nabla^i p(\mathbf{x})] = \sum_{i=0}^k \|\mathbf{T}_i\|_F^2.$$

Thus, we have $\|F(p)\|_F \leq C^k \log^k(d) \|Q\|_{D_{\mathbf{x}}, 2}$, which contradicts the fact that $\|F(p)\|_F > C^k \log^k(d) \|Q\|_{D_{\mathbf{x}}, 2}$.

Suppose that we have found $\hat{\mathbf{T}}_1, \dots, \hat{\mathbf{T}}_k$ such that for every $\hat{\mathbf{T}}_i$, $\|\hat{\mathbf{T}}_i\|_F \leq 2C^k \log^k(d)$ and for every degree k monomial $\mathbf{x}^\zeta$, where ζ is a multi-index, $|\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q(\mathbf{x}) \mathbf{x}^\zeta] - \sum_{i=1}^k \hat{\mathbf{T}}_i \cdot \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\nabla^i \mathbf{x}^\zeta]| \leq \eta'$. Notice that such a set of tensors always exists due to the existence of $\mathbf{T}_0, \dots, \mathbf{T}_k$. To complete the proof, we first show that we can efficiently find the desired subspace V through these tensors and then analyze the complexity of finding these tensors.

For each $\hat{\mathbf{T}}_i$, we treat it as a bilinear form over $\mathbb{R}^d \times \text{sym}(\mathbb{R}^{d^{i-1}})$ and compute the singular vectors $\mathbf{v}^{(i,1)}, \dots, \mathbf{v}^{(i,l)} \in \mathbb{R}^d$ of $\hat{\mathbf{T}}_i$ with corresponding singular value at least η . Since $\ell \leq \|\hat{\mathbf{T}}_i\|_F^2 / \eta^2$, and $\|\hat{\mathbf{T}}_i\|_F \leq 2C^k \log^k(d) \|Q(\mathbf{x})\|_{D_{\mathbf{x}},2}$, we know that the dimension of V is at most $2^{O(k \log k)} \log^{2k}(d) / \epsilon^2$. We claim that V satisfies the statement of Lemma 5.4.

Let $\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(k)} \perp V$ be unit vectors and let $p(\mathbf{x}) = \prod_{i=1}^k (\mathbf{u}^{(i)} \cdot \mathbf{x})$. We have

$$\begin{aligned} \left| \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q(\mathbf{x})p(\mathbf{x})] \right| &\leq \left| \sum_{i=1}^k \hat{\mathbf{T}}_i \cdot \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\nabla^i p(\mathbf{x})] \right| + \left| \sum_{i=1}^k \hat{\mathbf{T}}_i \cdot \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\nabla^i p(\mathbf{x})] - \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q(\mathbf{x})p(\mathbf{x})] \right| \\ &= \left| \sum_{i=1}^k \sum_{j=1}^k \mathbf{u}^{(j)} \cdot \hat{\mathbf{T}}_i \cdot \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\nabla^{i-1} p(\mathbf{x}) / (\mathbf{u}^{(j)} \cdot \mathbf{x})] \right| + \left| \sum_{i=1}^k \hat{\mathbf{T}}_i \cdot \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\nabla^i p(\mathbf{x})] - \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q(\mathbf{x})p(\mathbf{x})] \right| \\ &\leq \eta \sum_{i=1}^k \sum_{j=1}^k \left\| \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\nabla^{i-1} p(\mathbf{x}) / (\mathbf{u}^{(j)} \cdot \mathbf{x})] \right\|_F + \left| \sum_{i=1}^k \hat{\mathbf{T}}_i \cdot \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\nabla^i p(\mathbf{x})] - \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q(\mathbf{x})p(\mathbf{x})] \right|. \end{aligned}$$

We next bound the term $\left\| \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\nabla^{i-1} p(\mathbf{x}) / (\mathbf{u}^{(j)} \cdot \mathbf{x})] \right\|_F = \left\| \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\nabla^{i-1} \prod_{\ell \neq j} (\mathbf{u}^{(\ell)} \cdot \mathbf{x})] \right\|_F$. We have

$$\begin{aligned} \left\| \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\nabla^{i-1} \prod_{\ell \neq j} (\mathbf{u}^{(\ell)} \cdot \mathbf{x})] \right\|_F &\leq k! \left\| \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} \left[\sum_{\substack{S \subseteq [k] \setminus \{j\} \\ |S|=i-1}} \prod_{\ell \notin S} (\mathbf{u}^{(\ell)} \cdot \mathbf{x}) \text{sym} \bigotimes_{\ell \in S} \mathbf{u}^{(\ell)} \right] \right\|_F \\ &\leq k! \sum_{\substack{S \subseteq [k] \setminus \{j\} \\ |S|=i-1}} \left| \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\prod_{\ell \notin S} (\mathbf{u}^{(\ell)} \cdot \mathbf{x})] \right| \leq k! \binom{k-1}{i-1} (2(k-i))^{k-i}. \end{aligned}$$

Here, the first inequality follows from the fact that $\|\text{sym} \bigotimes_{\ell \in S} \mathbf{u}^{(\ell)}\|_F \leq 1$ and the second inequality follows from Fact A.9. This implies that

$$\begin{aligned} \sum_{i=1}^k \sum_{j=1}^k \left\| \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\nabla^{i-1} p(\mathbf{x}) / (\mathbf{u}^{(j)} \cdot \mathbf{x})] \right\|_F &\leq k! k \sum_{i=1}^k \binom{k-1}{i-1} (2(k-i))^{k-i} \\ &= k! k \sum_{i=0}^{k-1} \binom{k-1}{i} (2i)^i \\ &\leq k! k^2 (2k^2)^k = 2^{O(k \log(k))}. \end{aligned}$$

On the other hand, we notice that we can expand $p(\mathbf{x})$ as a linear combination of monomials, $p(\mathbf{x}) = \sum_{\zeta} c_{\zeta} \mathbf{x}^{\zeta}$. By linearity, we have

$$\begin{aligned} \left| \sum_{i=1}^k \hat{\mathbf{T}}_i \cdot \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\nabla^i p(\mathbf{x})] - \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q(\mathbf{x})p(\mathbf{x})] \right| &= \left| \sum_{\zeta} c_{\zeta} \left(\sum_{i=1}^k \hat{\mathbf{T}}_i \cdot \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\nabla^i \mathbf{x}^{\zeta}] - \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q(\mathbf{x})\mathbf{x}^{\zeta}] \right) \right| \\ &\leq \|c\|_2 \sqrt{\sum_{\zeta} \left(\sum_{i=1}^k \hat{\mathbf{T}}_i \cdot \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\nabla^i \mathbf{x}^{\zeta}] - \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q(\mathbf{x})\mathbf{x}^{\zeta}] \right)^2} \\ &\leq \sqrt{k!} d^k \eta'. \end{aligned}$$

Here, in the second inequality, we use the fact that the coefficient vector $c = (c_\zeta)$ is spanned from $\prod_{i=1}^k (\mathbf{u}^{(i)} \cdot \mathbf{x})$ and has ℓ_2 norm at most $\sqrt{k!}$. By the choice of η and η' , we have $|\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[Q(\mathbf{x})p(\mathbf{x})]| \leq \epsilon \|Q\|_{D_{\mathbf{x}}, 2}$.

To conclude the proof, it remains to upper bound the sample complexity of finding the tensors $\hat{\mathbf{T}}_1, \dots, \hat{\mathbf{T}}_k$. We know that for each monomial $\mathbf{x}^\zeta$, by Fact A.9 and Fact A.10, there is some constant $C > 0$ such that

$$\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q^2(\mathbf{x})(\mathbf{x}^\zeta)^2] \leq \sqrt{\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q^4(\mathbf{x})] \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [(\mathbf{x}^\zeta)^4]} \leq C^{2m} (8k)^{4k} \|Q\|_{D_{\mathbf{x}}, 2}^2.$$

By Chebyshev's inequality, we can estimate some $\mu_\zeta \in \mathbb{R}$ such that $|\mu_\zeta - \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[Q(\mathbf{x})\mathbf{x}^\zeta]| < \eta'/4$ with $d^{O(k)} 2^{O(k \log k + m)} / \epsilon^2$ examples from $D_{\mathbf{x}}$. Furthermore, denote by S_ζ the set of indices that are involved in $\mathbf{x}^\zeta$, for every i , we have

$$\begin{aligned} \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\|\nabla^i \mathbf{x}^\zeta\|_F^2] &\leq (k!)^2 \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} \left[\left\| \sum_{\substack{S \subseteq S_\zeta \\ |S|=i}} \prod_{\ell \notin S} (\mathbf{e}_\ell \cdot \mathbf{x})^{\text{sym}} \bigotimes_{\ell \in S} \mathbf{e}_\ell \right\|_F^2 \right] \\ &\leq (k!)^2 \sum_{\substack{S \subseteq S_\zeta \\ |S|=i}} \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\prod_{\ell \notin S} (\mathbf{e}_\ell \cdot \mathbf{x})^2] \\ &\leq 2^{k \log k} \binom{k}{i} (2C(k-i))^{2(k-i)}. \end{aligned}$$

By Chebyshev's inequality, we can estimate a tensor $Z_\zeta^{(i)}$ such that $\|Z_\zeta^{(i)} - \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[\nabla^i \mathbf{x}^\zeta]\|_F < \eta'/(4\sqrt{k})$ using $O(k^k \log^{2k}(d) \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[\|\nabla^i \mathbf{x}^\zeta\|_F^2]/(\eta')^2)$ examples from $D_{\mathbf{x}}$. This implies that $Z = (Z_\zeta^{(i)})_{i=0}^k$ is $\eta'/(4 \log^k(d))$ close to $F(\mathbf{x}^\zeta)$ in Frobenius norm.

By the triangle inequality, any tensors $\hat{\mathbf{T}}_1, \dots, \hat{\mathbf{T}}_k$ such that $\|\hat{\mathbf{T}}_i\|_F \leq 2C \log^k(d)$ for all $i \in [k]$, and $|\mu_\zeta - \sum_{i=0}^k \hat{\mathbf{T}}_i \cdot Z_\zeta^{(i)}| \leq \eta'/2$ must satisfy $|\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[Q(\mathbf{x})\mathbf{x}^\zeta] - \sum_{i=0}^k \hat{\mathbf{T}}_i \cdot \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[\nabla^i \mathbf{x}^\zeta]| \leq \eta'$, for every monomial $\mathbf{x}^\zeta$. Since $\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[Q(\mathbf{x})p(\mathbf{x})] = \sum_{i=0}^k \mathbf{T}_i \cdot \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[\nabla^i p(\mathbf{x})]$, we know that such a set of tensors exists due to $\mathbf{T}_1, \dots, \mathbf{T}_k$ and thus we can find $\hat{\mathbf{T}}_1, \dots, \hat{\mathbf{T}}_k$ through convex programming. Furthermore, the total number of examples used by Algorithm 2 is $d^{O(k)} 2^{O(k \log k + m)} / \epsilon^2$. $\square$

Using Lemma 5.4 as a subroutine, we give an algorithm that recursively calls Lemma 5.4 to build a low-dimensional subspace V that approximately preserves the maximum of the quantity $\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[Q(\mathbf{x}) \prod_{j=1}^k (\mathbf{u}^{(j)} \cdot \mathbf{x})]$, when $\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(k)}$ are taken from V . Specifically, we establish the following.

Lemma 5.5. *There is an algorithm with the following performance guarantee. Given sample access to a $1/2$ -isotropic logconcave distribution $D_{\mathbf{x}}$, parameters $m, k \in \mathbb{N}$, $\epsilon, \delta \in (0, 1)$, and any degree m -polynomial $Q : \mathbb{R}^d \rightarrow \mathbb{R}$, it draws $M = \text{poly}(d^k, C_k N_{k,m}(\epsilon), \log(1/\delta))$ examples from $D_{\mathbf{x}}$ over $\mathbb{R}^d$, runs in $\text{poly}(M)$ time, and outputs a subspace V of dimension $N_{k,m}(\epsilon)$ such that with probability at least $1 - \delta$,*

$$S_{k, \mathbb{R}^d}(Q) \leq C_k S_{k,V}(Q) + \epsilon \|Q\|_{D_{\mathbf{x}}, 2},$$

where for $k \in \mathbb{N}$, subspace $V \subseteq \mathbb{R}^d$ and a function Q , we denote

$$S_{k,V}(Q) := \sup_{0 \leq i \leq k} \sup_{\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(i)} \in V \cap \mathbb{S}^{d-1}} \left| \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q(\mathbf{x}) \prod_{j=1}^k (\mathbf{u}^{(j)} \cdot \mathbf{x})] \right|.$$

For $m, k \in \mathbb{N}$ and for $\epsilon \in (0, 1)$, $C_k = 2^{O(k^2)}$ and $N_{k,m}(\epsilon) = (2^{O(k+m)} \log^k(d)/\epsilon)^{2^{O(k)}}$.

Algorithm 3 LOWSUBSPACEFINDING(Find a low dimensional subspace that preserves discrepancy)

- 1: **Input:** Sample access from $D_{\mathbf{x}}$, accuracy parameter $\epsilon \in (0, 1)$, confidence parameter $\delta \in (0, 1)$, degree m -polynomial $Q(\mathbf{x})$, k -degree parameter
 - 2: **Output:** V , a low-dimensional subspace
 - 3: If $k = 0$, Return $\emptyset$
 - 4: Let $U^{(0)} = \emptyset$ and $\epsilon_0 = \epsilon$
 - 5: **for** $i = 1, \dots, k$ **do**
 - 6: Let $G_j = \{\mathbf{g}^{(j1)}, \dots, \mathbf{g}^{(jN_j)}\}$ be a standard basis for $U^{(j)}$, for $j \leq i-1$
 - 7: **for** $\mathbf{g} = (\mathbf{g}_1, \dots, \mathbf{g}_{i-1})$, where $\mathbf{g}_j \in G_j$ **do**
 - 8: Define $F_g(\mathbf{x}) = Q(\mathbf{x}) \prod_{j=1}^{i-1} (\mathbf{g}_j \cdot \mathbf{x})$
 - 9: Estimate $Q_g \in \mathbb{R}$ such that $|Q_g - \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [F_g(\mathbf{x})]| < 2^{-O(k \log k)} \epsilon_{i-1} \|F_g(\mathbf{x})\|_{D_{\mathbf{x}}, 2}$.
 - 10: Apply Algorithm 2 to $F_g(\mathbf{x}) - Q_g$ with dimension $k - i + 1$ and error parameter $\epsilon_{i-1}/2$ to get a subspace U_g
 - 11: Let $U^{(i)}$ be the subspace with dimension N_i spanned by all the U_g found in the i th iteration
 - 12: Let $\epsilon_i = (2^k C^m N_i)^{-1} \epsilon_{i-1}/2$.
 - 13: Return V the subspace spanned by $U^{(i)}$, $i \in [k]$.
-

Proof of Lemma 5.5. For the purpose of analysis, we will extend a degree m polynomial to a tensor-valued polynomial. Specifically, such a function P maps $\mathbb{R}^d$ to $\mathbb{R}^{d_1 \times \dots \times d_\ell}$ and each entry P_ζ of P is a degree- m polynomial. For an ℓ -order tensor $\mathbf{T} \in \mathbb{R}^{d_1 \times \dots \times d_\ell}$, define $\|\mathbf{T}\|_* := \sup_{\mathbf{v}^{(1)} \in \mathbb{S}^{d_1-1}, \dots, \mathbf{v}^{(\ell)} \in \mathbb{S}^{d_\ell-1}} |\mathbf{T} \cdot \otimes_{j=1}^\ell \mathbf{v}^{(j)}|$. For $k \in \mathbb{N}$ and subspace $V \subseteq \mathbb{R}^d$, and a ℓ -order tensor valued function P , define

$$S_{k,V}(P) = \sup_{0 \leq i \leq k} \sup_{\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(i)} \in V \cap \mathbb{S}^{d-1}} \left\| \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [P(\mathbf{x}) \prod_{j=1}^k (\mathbf{u}^{(j)} \cdot \mathbf{x})] \right\|_*.$$

Notice that in the i th iteration of the algorithm, if we write $Q^{(i-1)} = (F_{\mathbf{g}})$, for $\mathbf{g} = (\mathbf{g}_1, \dots, \mathbf{g}_{i-1})$, $\mathbf{g}_j \in G_j$ as a $i-1$ -order tensor, then, we can write $Q^{(i)}$ as a i -order tensor, for each $\mathbf{g} = (\mathbf{g}_1, \dots, \mathbf{g}_{i-1})$, $Q_{\mathbf{g}}^{(i)} = (F_{\mathbf{g}}(\mathbf{x})(g^{(i1)} \cdot \mathbf{x}), \dots, F_{\mathbf{g}}(\mathbf{x})(g^{(iN_i)} \cdot \mathbf{x}))$, where $g^{i1}, \dots, g^{iN_i}$ is the standard basis of $U^{(i)}$. In particular, under this notation, we have $Q(\mathbf{x}) = Q^{(0)}$. We will show by induction that for $j = 0, \dots, k$, the subspace $V^{(j)}$ spanned by $U^{(k)}, \dots, U^{(k-j+1)}$ satisfies

$$S_{j, \mathbb{R}^d}(Q^{(k-j)}) \leq C_j S_{j, V^{(j)}}(Q^{(k-j)}) + \epsilon_{k-j} \left\| Q^{(k-j)} \right\|_{F, D_{\mathbf{x}}, 2}.$$

Notice that if this is true, then for $j = k$, we have

$$S_{k, \mathbb{R}^d}(Q) \leq C_k S_{k, V}(Q) + \epsilon \|Q\|_{D_{\mathbf{x}}, 2}.$$

The base case $j = 0$ holds trivially because for every subspace V' , we have $S_{0,V'}(Q^k) = \|\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[Q^k]\|_*$. Suppose this holds for $j = k - 1$, show this for $j = k$.

Let $\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(i)}$ be i unit vectors in $\mathbb{R}^d$, where $i \leq k$. We write $\mathbf{u}^{(j)} = \mathbf{a}^{(j)} + \mathbf{b}^{(j)}$, where $\mathbf{a}^{(j)} = \text{proj}_{U^{(1)}} \mathbf{u}^{(j)}$ and $\mathbf{b}^{(j)} = \text{proj}_{(U^{(1)})^\perp} \mathbf{u}^{(j)}$. Then, we have

$$\left| \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q(\mathbf{x}) \prod_{j=1}^i (\mathbf{u}^{(j)} \cdot \mathbf{x})] \right| \leq \sum_{S \subseteq [i]} \left| \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q(\mathbf{x}) \prod_{j \notin S} (\mathbf{b}^{(j)} \cdot \mathbf{x}) \prod_{j \in S} (\mathbf{a}^{(j)} \cdot \mathbf{x})] \right|.$$

We upper bound each term in the above summation. Recall that $|Q_\emptyset - \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[Q(\mathbf{x})]| < 2^{-O(k \log k)} \epsilon_1 \|Q\|_{D_{\mathbf{x}},2}$. If $S = \emptyset$, then we have

$$\begin{aligned} \left| \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q(\mathbf{x}) \prod_{j \notin S} (\mathbf{b}^{(j)} \cdot \mathbf{x}) \prod_{j \in S} (\mathbf{a}^{(j)} \cdot \mathbf{x})] \right| &= \left| \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q(\mathbf{x}) \prod_{j=1}^i (\mathbf{b}^{(j)} \cdot \mathbf{x})] \right| \\ &= \left| \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q_\emptyset \prod_{j=1}^i (\mathbf{b}^{(j)} \cdot \mathbf{x}) + (Q(\mathbf{x}) - Q_\emptyset) \prod_{j=1}^i (\mathbf{b}^{(j)} \cdot \mathbf{x})] \right| \\ &\leq |Q_\emptyset| \left| \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\prod_{j=1}^i (\mathbf{b}^{(j)} \cdot \mathbf{x})] \right| + \left| \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [(Q(\mathbf{x}) - Q_\emptyset) \prod_{j=1}^i (\mathbf{b}^{(j)} \cdot \mathbf{x})] \right| \\ &\leq (2i)^i |Q_\emptyset| + \epsilon_0 \|Q\|_{D_{\mathbf{x}},2} / 2, \end{aligned}$$

where the last inequality follows by Fact A.9 and the definition of $U^{(1)}$.

In the rest of the proof, we focus on the case where $S \neq \emptyset$, which implies there is at least one $a^{(j)}$ that appears in the product $\prod_{j \notin S} (\mathbf{b}^{(j)} \cdot \mathbf{x}) \prod_{j \in S} (\mathbf{a}^{(j)} \cdot \mathbf{x})$. Without loss of generality, we assume it is $a^{(1)}$ and denote by $M_S(\mathbf{x}) = (\mathbf{a}^{(1)} \cdot \mathbf{x})^{-1} \prod_{j \notin S} (\mathbf{b}^{(j)} \cdot \mathbf{x}) \prod_{j \in S} (\mathbf{a}^{(j)} \cdot \mathbf{x})$ the remaining part of the product. We write $\mathbf{a}^{(1)} = \sum_{\mathbf{g} \in G_1}^{N_1} \mathbf{c}_{\mathbf{g}} \mathbf{g}$ under the basis G_1 . Then, we have

$$\begin{aligned} \left| \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q(\mathbf{x}) \prod_{j \notin S} (\mathbf{b}^{(j)} \cdot \mathbf{x}) \prod_{j \in S} (\mathbf{a}^{(j)} \cdot \mathbf{x})] \right| &= \left| \sum_{\mathbf{g} \in G_1} \mathbf{c}_{\mathbf{g}} \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q(\mathbf{x}) (\mathbf{g} \cdot \mathbf{x}) M_S(\mathbf{x})] \right| \\ &= \left| \sum_{\mathbf{g} \in G_1} [\mathbf{c} \cdot Q^{(1)}(\mathbf{x}) M_S(\mathbf{x})] \right|, \end{aligned}$$

where $\mathbf{c} = (\mathbf{c}_{\mathbf{g}})_{\mathbf{g} \in G_1}$. We notice that for every polynomial $p(\mathbf{x})$, we have

$$\sup_{\mathbf{v} \in U^{(1)} \cap \mathbb{S}^{d-1}} |Q(\mathbf{x})(\mathbf{v} \cdot \mathbf{x})p(\mathbf{x})| = \sup_{\mathbf{v} \in U^{(1)} \cap \mathbb{S}^{d-1}} |\mathbf{v} \cdot Q^{(1)}(\mathbf{x})p(\mathbf{x})| = \|Q^{(1)}(\mathbf{x})p(\mathbf{x})\|_*. \quad (4)$$

By induction, we have

$$\begin{aligned} \left| \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q(\mathbf{x})(\mathbf{a}^{(1)} \cdot \mathbf{x}) M_S(\mathbf{x})] \right| &\leq \left\| \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q^{(1)}(\mathbf{x}) M_S(\mathbf{x})] \right\|_* \leq S_{k-1,R^d}(Q^{(1)}) \\ &\leq C_{k-1} S_{k-1,V^{(k-1)}}(Q^{(1)}) + \epsilon_1 \sqrt{\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\|Q^{(1)}(\mathbf{x})\|_F^2]} \\ &\leq C_{k-1} S_{k,V}(Q) + \epsilon_1 \sqrt{\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\|Q^{(1)}(\mathbf{x})\|_F^2]}. \end{aligned}$$

Here, the last inequality follows from (4). By Fact A.10, for some universal constant $C > 0$,

$$\begin{aligned} \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} \left[\left\| Q^{(1)}(\mathbf{x}) \right\|_F^2 \right] &= \sum_{\mathbf{g} \in G_1} \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [(Q(\mathbf{x})(\mathbf{g}^{(i)} \cdot \mathbf{x}))^2] = \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q(\mathbf{x})^2 \sum_{\mathbf{g} \in G_1} (\mathbf{g} \cdot \mathbf{x})^2] \\ &\leq \sqrt{\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} Q(\mathbf{x})^4} \sqrt{\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} \left[\left(\sum_{i=1}^N (\mathbf{g}^{(i)} \cdot \mathbf{x})^2 \right)^2 \right]} \leq N_1^2 C^{2m} \|Q\|_{D_{\mathbf{x},2}}^2. \end{aligned}$$

Thus, we get

$$\left| \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q(\mathbf{x})(\mathbf{a}^{(1)} \cdot \mathbf{x}) M_S(\mathbf{x})] \right| \leq C_{k-1} S_{k,V}(Q) + \epsilon_1 N_1 C^m \|Q\|_{D_{\mathbf{x},2}}^2,$$

which implies

$$\begin{aligned} \left| \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q(\mathbf{x}) \prod_{j=1}^i (\mathbf{u}^{(j)} \cdot \mathbf{x})] \right| &\leq (2^k C_{k-1} + 2^{O(k \log k)}) S_{k,V}(Q) + (2^k C^m N_1 \epsilon_1 + \epsilon_0/2) \|Q(\mathbf{x})\|_{D_{\mathbf{x},2}}^2 \\ &\leq C_k S_{k,V}(Q) + \epsilon \|Q(\mathbf{x})\|_{D_{\mathbf{x},2}}^2, \end{aligned}$$

where we use our choice that $C_k = 2^k C_{k-1} + 2^{O(k \log k)} = 2^{O(k^2)}$ and $\epsilon_1 \leq (N_1 C^m)^{-1} \epsilon_0/2$.

To complete the proof, we bound the dimension of the subspace V and the complexity of the algorithm. Notice that in the i th iteration of the algorithm, we run Algorithm 2 $T_i = \prod_{j=1}^{i-1} \dim(U^{(j)})$ times with parameter $\epsilon_{i-1}/2$.

By Lemma 5.4, it follows that in the i th iteration, each time we run Algorithm 2, it draws $M_i = d^{O(k)} 2^{O(k \log k + m)} / \epsilon_{i-1}^2$ examples from D , runs in $\text{poly}(M_i)$ time, and outputs a subspace $U^{(g)}$ of dimension $2^{O(k \log k) \log^{2k}(d)} / \epsilon_{i-1}^2$. So, in the i th iteration, the total number of examples we draw from D is $T_i M_i$, the running time is $T_i \text{poly}(M_i)$ and $\dim(U^{(i)}) = T_i 2^{O(k \log k) \log^{2k}(d)} / \epsilon_{i-1}^2$.

Since $\epsilon_0 = \epsilon$ and $\epsilon_i = (2^k C^m N_i)^{-1} \epsilon_{i-1}/2$ for $i \in [k]$, we get the dimension of V is at most $N_{k,m}(\epsilon) = (2^{O(k+m)} \log^k(d)/\epsilon)^{2^{O(k)}}$ and the sample/time complexity of the algorithm is $\text{poly}(d^k, C_k N_{k,m}(\epsilon), \log(1/\delta))$. $\square$

Remark 5.6. We remark that the dimension of V found by Lemma 5.5 has a $\text{polylog}(d)$ factor dependence on the dimension d . Such a dependence appears because Lemma 5.3 has a $\text{polylog}(d)$ factor dependence. If the KLS conjecture holds, then we can get an improved version of Lemma 5.3 without the dependence on d , which implies we can remove the dependence of $\text{polylog}(d)$ in Lemma 5.5. However, for our application, we will see later that we can recursively apply Lemma 5.5 to the discovered subspace V to reduce the $\text{polylog}(d)$ dependence to $\log \log(d)$ dependence.

5.3 From Moment Discrepancy to Dimension Reduction

In this section, we will apply Lemma 5.5 to find a direction $\mathbf{w}$ along which $D_{\mathbf{x}}$ and $(D_f)_{\mathbf{x}}$ have mismatched moments. To begin with, we will specify the polynomial function $Q(\mathbf{x})$ we want to use.

Lemma 5.7. *There is an algorithm such that, given sample access to a logconcave distribution $D_{\mathbf{x}}$ over $\mathbb{R}^d$, oracle access to an event A with $\Pr_{D_{\mathbf{x}}}[A] = \alpha$, and parameters $k \in \mathbb{N}, \epsilon, \delta \in (0, 1)$, it draws $\text{poly}(d^k, 1/\epsilon^k, \log(1/\delta), 1/\alpha)$ samples from $D_{\mathbf{x}}$, runs in sample-polynomial time, and outputs a degree- k polynomial $Q : \mathbb{R}^d \rightarrow \mathbb{R}$ with $\|Q\|_{D_{\mathbf{x},2}} \leq O(\log^k(1/\alpha)) + \epsilon$, such that with probability at least $1 - \delta$, for every degree at most k polynomial $p : \mathbb{R}^d \rightarrow \mathbb{R}$, it holds $\left| \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q(\mathbf{x}) p(\mathbf{x})] - \mathbf{E}_{\mathbf{x} \sim (D_f)_{\mathbf{x}}} [p(\mathbf{x})] \right| \leq \epsilon \|p(\mathbf{x})\|_{D_{\mathbf{x},2}}$, where $(D_f)_{\mathbf{x}}$ is the conditional distribution of $D_{\mathbf{x}}$ over event A .*

Proof of Lemma 5.7. We first show the existence of such a polynomial. Define inner product over the space of polynomial functions over $\mathbb{R}^d$ as $\langle p, q \rangle_{D_{\mathbf{x}}} := \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [pq]$ as well as $\|\cdot\|_{D_{\mathbf{x}},2}$, the norm induced by the inner product, and denote by P_k the subspace of all polynomials with degree at most k over $\mathbb{R}^d$. Under the inner product $\langle p, q \rangle_{D_{\mathbf{x}}}$, define $Q^*(\mathbf{x}) := \text{proj}_{P_k} \mathbb{1}\{A\}/\alpha$. Notice that for every polynomial $p(\mathbf{x})$ with at most degree k ,

$$\mathbf{E}_{\mathbf{x} \sim (D_f)_{\mathbf{x}}} [p(\mathbf{x})] = \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [p(\mathbf{x}) \mathbb{1}\{A\}/\alpha] = \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q^*(\mathbf{x}) p(\mathbf{x})],$$

where the last equation uses the fact that $\langle Q^*(\mathbf{x}) - \mathbb{1}\{A\}/\alpha, p(\mathbf{x}) \rangle_{D_{\mathbf{x}}} = 0$. Furthermore, we

$$\|Q^*(\mathbf{x})\|_{D_{\mathbf{x}},2} = \sup_{p \in P_k, \|p\|_{D_{\mathbf{x}},2}=1} \langle p(\mathbf{x}), Q^*(\mathbf{x}) \rangle_{D_{\mathbf{x}}} = \sup_{p \in P_k, \|p\|_{D_{\mathbf{x}},2}=1} \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [p(\mathbf{x}) \mathbb{1}\{A\}/\alpha].$$

For $r \geq 2$, let $r' = (1 - r^{-1})^{-1}$. By Holder's inequality, we have

$$\begin{aligned} \left| \alpha^{-1} \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [p(\mathbf{x}) \mathbb{1}\{A\}] \right| &\leq \alpha^{-1} \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [p^r(\mathbf{x})]^{1/r} \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\mathbb{1}\{A\}^{r'}]^{1/r'} \\ &= \alpha^{-1/r} \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [p^r(\mathbf{x})]^{1/r} \leq \alpha^{-1/r} (Cr)^k. \end{aligned}$$

Here, in the last inequality, we use Fact A.10. By choosing $r = O(\log(1/\alpha))$, we get $\|Q^*(\mathbf{x})\|_{D_{\mathbf{x}},2} \leq O(\log^k(1/\alpha))$.

It suffices to find a polynomial $Q(\mathbf{x}) \in P_k$ such that $\|Q - Q^*\|_{D_{\mathbf{x}},2} < \epsilon$. To do this, solve the polynomial regression problem that finds a polynomial $Q \in P_k$ such that

$$\begin{aligned} \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [(Q(\mathbf{x}) - \mathbb{1}\{A\}/\alpha)^2] &\leq \min_{p(\mathbf{x}) \in P_k} \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [(p(\mathbf{x}) - \mathbb{1}\{A\}/\alpha)^2] + \epsilon \\ &= \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [(Q^*(\mathbf{x}) - \mathbb{1}\{A\}/\alpha)^2] + \epsilon, \end{aligned}$$

which needs $\text{poly}(d^k/(\alpha\epsilon))$ sample from $D_{\mathbf{x}}$, using standard sample complexity analysis of linear regression (see, e.g., [BM02].) Notice that

$$\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [(Q(\mathbf{x}) - Q^*(\mathbf{x}) + Q^*(\mathbf{x}) - \mathbb{1}\{A\}/\alpha)^2] = \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [(Q(\mathbf{x}) - Q^*(\mathbf{x}))^2] + \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [(Q^*(\mathbf{x}) - \mathbb{1}\{A\}/\alpha)^2].$$

This means the solution of the polynomial regression gives a polynomial $Q(\mathbf{x}) \in P_k$ such that $\|Q(\mathbf{x}) - Q^*(\mathbf{x})\|_{D_{\mathbf{x}},2} \leq \epsilon$ and $\|Q\|_{D_{\mathbf{x}},2} \leq O(\log^k(1/\alpha)) + \epsilon$.

Furthermore, for every polynomial $p(\mathbf{x}) \in P_k$, we have

$$\left| \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q(\mathbf{x}) p(\mathbf{x})] - \mathbf{E}_{\mathbf{x} \sim (D_f)_{\mathbf{x}}} [p(\mathbf{x})] \right| = \left| \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [(Q(\mathbf{x}) - Q^*(\mathbf{x})) p(\mathbf{x})] \right| \leq \epsilon \|p(\mathbf{x})\|_{D_{\mathbf{x}},2}.$$

□

With Lemma 5.5 and Lemma 5.7 at hand, we are able to directly prove Theorem 5.1.

Proof of Theorem 5.1. By Fact A.9, we know that for every $\mathbf{w} \in \mathbb{S}^{d-1}$, $\|(\mathbf{w} \cdot \mathbf{x})^i\|_{D_{\mathbf{x}},2} \leq (Ck)^k$, for $i \in [k]$. So, by applying Lemma 5.7 with $\epsilon_1 = 2^{-O(k^2)}\Delta$ over $D_{\mathbf{x}}$ and A we can find a degree k -polynomial $Q(\mathbf{x}')$ with $\|Q(\mathbf{x})\|_{D_{\mathbf{x}},2} \leq O(\log^k(1/\alpha))$ such that for every unit vector $\mathbf{w} \in \mathbb{S}^{d-1}$, and $i \in [k]$

$$\left| \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [Q(\mathbf{x})(\mathbf{w} \cdot \mathbf{x})^i] - \mathbf{E}_{\mathbf{x} \sim (D_f)_{\mathbf{x}}} [(\mathbf{w} \cdot \mathbf{x}')^i] \right| \leq 2^{-O(k^2)}\Delta.$$

Algorithm 4 MISMATCHFINDING(Finding a direction along which moment discrepancies)

- 1: **Input:** Sample access from D , event A , degree k , parameter Δ
 - 2: **Output:** $\mathbf{v} \in \mathbb{S}^{d-1}$ or $\perp$, indicating D and D_f have nearly matched moments
 - 3: Apply Lemma 5.7 to $D_{\mathbf{x}}$ and A with parameter $\epsilon_1 = 2^{-O(k^2)}\Delta$ and degree k to get a degree k -polynomial $Q(\mathbf{x})$.
 - 4: Run Algorithm 3 with input $Q(\mathbf{x}) - 1$, degree k and parameter $\epsilon = O(\Delta \log^{-k}(1/\alpha))$ to get a subspace U
 - 5: Let $D_{\mathbf{x}}^{(U)}$ be the distribution of $\text{proj}_U(\mathbf{x})$
 - 6: Apply Lemma 5.7 to $D_{\mathbf{x}}^{(U)}$ and A with parameter $\epsilon = 2^{-O(k^2)} \log^{-k}(1/\alpha)\Delta$ and degree k to get a degree k -polynomial $Q'(\mathbf{x})$.
 - 7: Run Algorithm 3 with input polynomial $Q'(\mathbf{x}) - 1$, degree k and parameter $\epsilon = 2^{-O(k^2)}\Delta$ to get a subspace V .
 - 8: Run a brute-force search over V to find a vector $\mathbf{v} \in \mathbb{S}^{d-1}$ such that $|\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(\mathbf{v} \cdot \mathbf{x})^i] - \mathbf{E}_{\mathbf{x} \sim D_f}[(\mathbf{v} \cdot \mathbf{x})^i]| > 2^{-O(k^2)}\Delta$, for some $i \in [k]$.
 - 9: If such a vector is not found, then return $\perp$, otherwise return $\mathbf{v}$.
-

Consider the k -symmetric linear form $f(\mathbf{v}^{(1)}, \dots, \mathbf{v}^{(k)}) = \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(Q(\mathbf{x}) - 1) \prod_{i=1}^k (\mathbf{v}^{(i)} \cdot \mathbf{x})]$. By [BS71], for every subspace V , $\sup_{\mathbf{v}^{(1)}, \dots, \mathbf{v}^{(k)} \in V \cap \mathbb{S}^{d-1}} f(\mathbf{v}^{(1)}, \dots, \mathbf{v}^{(k)}) = \sup_{\mathbf{v} \in V \cap \mathbb{S}^{d-1}} f(\mathbf{v}, \dots, \mathbf{v})$. By applying Lemma 5.5 on $Q(\mathbf{x}) - 1$ with distribution $D_{\mathbf{x}}$, we find a subspace U with dimension $(2^{O(k)} \log^k(1/\alpha) \log^k(d)/\Delta)^{2^{O(k)}}$, such that

$$\sup_{\mathbf{w} \in \mathbb{S}^{d-1}} \left| \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [(Q(\mathbf{x}) - 1)(\mathbf{w} \cdot \mathbf{x})^k] \right| \leq 2^{O(k^2)} \sup_{i \in [k]} \sup_{\mathbf{v} \in U \cap \mathbb{S}^{d-1}} \left| \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [(Q(\mathbf{x}) - 1)(\mathbf{v} \cdot \mathbf{x})^i] \right| + \Delta/100.$$

If there is some $\mathbf{w} \in \mathbb{S}^{d-1}$ such that $|\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(\mathbf{w} \cdot \mathbf{x})^k] - \mathbf{E}_{\mathbf{x} \sim (D_f)_{\mathbf{x}}}[(\mathbf{w} \cdot \mathbf{x})^k]| > \Delta$, there exists a unit vector $\mathbf{u} \in U$ and $i \in [k]$ such that $|\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(Q(\mathbf{x}) - 1)(\mathbf{u} \cdot \mathbf{x})^i]| \geq 2^{-O(k^2)}\Delta$ and thus

$$\left| \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [(\mathbf{u} \cdot \mathbf{x})^i] - \mathbf{E}_{\mathbf{x} \sim (D_f)_{\mathbf{x}}} [(\mathbf{u} \cdot \mathbf{x})^i] \right| \geq 2^{-O(k^2)}\Delta.$$

Denote by $D_{\mathbf{x}}^{(U)}$ the distribution of $\mathbf{a} = \text{proj}_U \mathbf{x}$ and $D_f^{(U)}$ the conditional distribution of $D_{\mathbf{x}}^{(U)}$ over A . For every $\mathbf{u} \in U$ and $i \in [k]$, we have

$$\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [(\mathbf{u} \cdot \mathbf{x})^i] - \mathbf{E}_{\mathbf{x} \sim (D_f)_{\mathbf{x}}} [(\mathbf{u} \cdot \mathbf{x})^i] = \mathbf{E}_{\mathbf{a} \sim D_{\mathbf{x}}^{(U)}} [(\mathbf{u} \cdot \mathbf{a})^i] - \mathbf{E}_{\mathbf{a} \sim (D_f^{(U)})_{\mathbf{x}}} [(\mathbf{u} \cdot \mathbf{a})^i].$$

Since $D_{\mathbf{x}}^{(U)}$ is a $1/2$ -isotropic logconcave distribution, by applying Lemma 5.7 to $D_{\mathbf{x}}^{(U)}$ with $\epsilon_2 = 2^{-O(k^2)}\Delta$, we can find a degree k -polynomial $Q'(\mathbf{a})$ with $\|Q'\|_{D_{\mathbf{x}}^{(U)}, 2} \leq O(\log^k(1/\alpha))$ such that for every unit vector $\mathbf{w} \in U$ and $i \in [k]$,

$$\left| \mathbf{E}_{\mathbf{a} \sim D_{\mathbf{x}}^{(U)}} [Q'(\mathbf{a})(\mathbf{w} \cdot \mathbf{a})^i] - \mathbf{E}_{\mathbf{a} \sim D_f^{(U)}} [(\mathbf{w} \cdot \mathbf{a})^i] \right| \leq 2^{-O(k^2)}\Delta.$$

By applying Lemma 5.5 on $Q'(\mathbf{a}) - 1$ and distribution $D_{\mathbf{x}}^{(U)}$ we find a subspace V with dimension $(2^{O(k^2)} \log^k(1/\alpha) \log^k \log(d)/\Delta)^{2^{O(k)}}$ such that

$$\sup_{\mathbf{w} \in U, \|\mathbf{w}\|_2=1} \left| \mathbf{E}_{\mathbf{a} \sim D_{\mathbf{x}}^{(U)}} [(Q'(\mathbf{a}) - 1)(\mathbf{w} \cdot \mathbf{a})^k] \right| \leq 2^{O(k^2)} \sup_{i \in [k]} \sup_{\mathbf{v} \in V, \|\mathbf{v}\|_2=1} \left| \mathbf{E}_{\mathbf{a} \sim D_{\mathbf{x}}^{(U)}} [(Q'(\mathbf{a}) - 1)(\mathbf{v} \cdot \mathbf{a})^i] \right| + 2^{-O(k^2)}\Delta.$$

Since there exists a unit vector $\mathbf{u} \in U$ such that $|\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(Q(\mathbf{x}) - 1)(\mathbf{u} \cdot \mathbf{x})^k]| \geq 2^{-O(k^2)}\Delta$, we know that there exists a unit vector $\mathbf{v} \in V$ such that $|\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(Q'(\mathbf{x}) - 1)(\mathbf{v} \cdot \mathbf{x})^i]| \geq 2^{-O(k^2)}\Delta$ and thus

$$\left| \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(\mathbf{v} \cdot \mathbf{x})^i] - \mathbf{E}_{\mathbf{x} \sim D_f}[(\mathbf{v} \cdot \mathbf{x})^i] \right| > 2^{-O(k^2)}\Delta.$$

By running a brute force search over V , we find such a vector $\mathbf{v}$. In particular, if no such vector $\mathbf{v}$ is output, then it must be the case that for every $\mathbf{w} \in \mathbb{S}^{d-1}$, $|\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(\mathbf{w} \cdot \mathbf{x})^k] - \mathbf{E}_{\mathbf{x} \sim D_f}[(\mathbf{w} \cdot \mathbf{x})^k]| < \Delta$.

Finally, we upper bound the sample and time complexity of the algorithm. In Algorithm 4, we run Algorithm 3 and Lemma 5.7 twice. Each run of Lemma 5.7 takes $\text{poly}(2^{(k^2)}d^k/(\alpha\Delta))$ sample and time. Each run of Algorithm 3 takes at most $d^{O(k)}(2^{O(k)}\log^k(1/\alpha)\log^k(d)/\Delta)^{2^{O(k)}}$ sample and time. Finally, since the final found subspace V has dimension $(2^{O(k^2)}\log^k(1/\alpha)\log^k(d)/\Delta)^{2^{O(k)}}$, the brute force search takes time $2^{O(k^2\log(1/\Delta)\dim(V))} = 2^{(\log(1/\alpha)\log\log(d)/\Delta)^{2^{O(k)}}}$. Thus, the algorithm draws $d^{O(k)}(\log(1/\alpha)\log(d)/\Delta)^{2^{O(k)}}$ sample and runs in $d^{O(k)}(\log(1/\alpha)\log(d)/\Delta)^{2^{O(k)}} + 2^{(\log(1/\alpha)\log\log(d)/\Delta)^{2^{O(k)}}}$ time. $\square$

6 Proof of Theorem 2.8 and Testable Learning Extensions

In this section, we put together results established in the previous sections to complete the proof of Theorem 2.8 and also provide our testable learning extensions.

Proof of Theorem 2.8. By Theorem 5.1, for any $\alpha, \epsilon \in (0, 1)$ and for any fixed constant $c(k, \Delta) \geq 1$, there is a $(k, \Delta, 2^{-O(k^2)}, (\alpha\epsilon)^{c(k, \Delta)})$ -discrepancy finder $\mathcal{A}_m$ for $D_{\mathbf{x}}$ and $(D_f)_{\mathbf{x}}$, with sample complexity $d^{O(k)}\text{poly}_{k, \Delta}((\alpha\epsilon)^{-1})$ and time complexity $d^{O(k)}2^{\text{polylog}_{k, \Delta}((\alpha\epsilon)^{-1})}$, whenever $W(D)$ —the weight of D over $\mathbb{R}^d \times \{0, \pm 1\}$ —is at least $(\alpha\epsilon)^{c(k, \Delta)}$.

Let $\epsilon' = 9\epsilon/10$. Let D be any distribution $\mathbb{R}^d \times \{0, \pm 1\}$ with weight $(\alpha\epsilon')$ and $D_{\mathbf{x}}$ a logconcave distribution. By Theorem 4.2, applied with $\eta = \epsilon/20$ and $\mathcal{A}_m$ as the discrepancy finder, it follows that whenever $\text{opt}_{D_f}(\mathcal{H}) \leq (1 - \eta)/C$, Algorithm 1 draws $d^{O(k)}\text{poly}_{k, \Delta}((\alpha\epsilon)^{-1})$ samples from D , runs in $\text{poly}(d^k)2^{\text{polylog}_{k, \Delta}((\alpha\epsilon)^{-1})}$ time and with probability at least 99% outputs a hypothesis $\hat{h}$ with $\text{err}_{D_f}(\hat{h}) \leq 1/2 - \Omega(\beta\eta)$. In other words, there is a $(\Omega(\epsilon\beta), C/(1 - \eta))$ -weak partial learner for $\mathcal{H}$ under $D_{\mathbf{x}}$ with sample complexity $d^{O(k)}\text{poly}_{k, \Delta}((\alpha\epsilon)^{-1})$ and time complexity $\text{poly}(d^k)2^{\text{polylog}_{k, \Delta}((\alpha\epsilon)^{-1})}$.

By Theorem 3.2, it follows that by calling the weak partial learner $\tilde{O}((\epsilon\beta)^{-2}/\epsilon')$ times, we obtain a hypothesis $\hat{h}$ with $\text{err}_{D_f}(\hat{h}) \leq C\text{opt}_{D_f}(\mathcal{H})/(2(1 - \eta)) + \epsilon'$. Without loss of generality, we assume that $(C/2)\text{opt}_{D_f}(\mathcal{H}) \leq 1/2$; otherwise, the guarantee is vacuous. This implies that

$$\text{err}_{D_f}(\hat{h}) \leq C\text{opt}_{D_f}(\mathcal{H})/(2(1 - \eta)) + \epsilon' \leq (C/2)\text{opt}_{D_f}(\mathcal{H}) + 2\eta + \epsilon' \leq (C/2)\text{opt}_{D_f}(\mathcal{H}) + \epsilon,$$

where the last inequality follows by our choice of ϵ' . This concludes the proof of Theorem 2.8. $\square$

Testable Learning The algorithm establishing Theorem 2.8 automatically gives an algorithm with the same complexity that succeeds in the testable learning framework [RV23], for any *fixed* logconcave distribution.

Before we state our relevant theorem, we start with the definition of testable learning.

Definition 6.1 (Testable Agnostic Partial Learning under Logconcave Distributions). *Let $\mathcal{H}$ be a class of Boolean (i.e., $\{\pm 1\}$ -valued) functions over $\mathbb{R}^d$. Let D be a distribution over $\mathbb{R}^d \times \{\pm 1, 0\}$ and D_f be the conditional distribution of D over the event $\mathcal{E} := \{(\mathbf{x}, y) \mid y \neq 0\}$. The error of*

$h : \mathbb{R}^d \rightarrow \{\pm 1\}$ is defined as $\text{err}_{D_f}(h) = \Pr_{(\mathbf{x}, y) \sim D_f}[h(\mathbf{x}) \neq y]$ and let $\text{opt}_{D_f}(\mathcal{H}) := \inf_{h \in \mathcal{H}} \text{err}_{D_f}(h)$. Given a set of i.i.d. examples S drawn from D and parameters $\epsilon, \delta \in (0, 1)$, a testable learner with respect to a given base distribution D_B over $\mathbb{R}^d$ can either reject S or accept and output a hypothesis $\hat{h} : \mathbb{R}^d \rightarrow \{\pm 1\}$. Further, the following conditions must be met:

1. (Soundness.) With probability at most δ , the learner accepts but outputs a hypothesis $\hat{h}$, $\text{err}_{D_f}(\hat{h}) > F(\text{opt}_{D_f}(\mathcal{H})) + \epsilon$, for some function $F : (0, 1) \rightarrow (0, 1)$.
2. (Completeness.) Whenever $D_{\mathbf{x}} = D_B$, for some known logconcave distribution D_B over $\mathbb{R}^d$, the learner will accept with probability at least $1 - \delta$.

For short, we say such a learner testably learns $\mathcal{H}$ with respect to D_B to error $F(\text{opt}_{D_f}(\mathcal{H})) + \epsilon$ with probability $1 - \delta$.

Our extension of Theorem 2.8 to the testable learning setting is as follows.

Theorem 6.2 (Testable Agnostic Partial Learning). *Consider the task of testable agnostic partial learning $\mathcal{H}$ under logconcave distributions. Suppose that $\mathcal{H}$ is (k, Δ, β, C) -weakly learnable by an algorithm $\mathcal{A}$ with sample complexity m and time complexity T . Then there is a testable agnostic partial learning algorithm for $\mathcal{H}$ with respect to a logconcave distribution D_B with the following guarantee: given any $\epsilon \in (0, 1)$ and sample access to any distribution D over $\mathbb{R}^d \times \{0, \pm 1\}$ of weight $W(D) \geq \alpha$ and sample access to D_B , the algorithm draws $(1/\beta^2) d^{O(k)} \text{poly}_{k, \Delta}(1/(\alpha\epsilon)) m$ samples from D and D_B , runs in time $(1/\beta^2) d^{O(k)} 2^{\text{poly}_{k, \Delta}(\log(1/(\alpha\epsilon)))} T$, and testably learns $\mathcal{H}$ with respect to D_B to error $(C/2)\text{opt}_{D_f}(\mathcal{H}) + \epsilon$ with probability at least 99%.*

Our algorithm for Theorem 6.2 will be essentially the same as the algorithm for Theorem 2.8 with the following modification. In the process of running Algorithm 1, we compute a partition of space so that (if our distributional hypotheses are satisfied) almost all of the mass of $(D_f)_{\mathbf{x}}$ lies in regions where the central moments of $(D_f)_{\mathbf{x}}$ and D_B approximately match (as is certified by our discrepancy finder). For the testable learning setting, as a high-level idea, we will essentially run Algorithm 1 identically, except every time we draw a sample from $D_{\mathbf{x}}$, we draw it from D_B , and we stop the algorithm after using an appropriate amount of time and samples. If a partition is generated successfully before we stop the algorithm, then it is likely that we can get a weak partial hypothesis using this partition; otherwise, we just reject. In particular, if $D_{\mathbf{x}} = D_B$, we will accept with high probability, as the input is the same as the one in the standard learning setting.

To formalize the above intuition, we will define a testable discrepancy finder as follows.

Definition 6.3 ((k, Δ, c) -Testable Discrepancy Finder). *For $k \in \mathbb{Z}_+$, $\Delta > 0$, $c \in (0, 1)$, we say that an algorithm $\mathcal{A}$ is a (k, Δ, c) -testable discrepancy finder for $D_{\mathbf{x}}$, if it satisfies the following performance guarantee: given sample access to a $1/2$ -isotropic logconcave distribution $D_{\mathbf{x}}$ over $\mathbb{R}^d$ and a distribution $D'_{\mathbf{x}}$ over $\mathbb{R}^d$, and accuracy parameter $\epsilon \in (0, 1/2)$ and weight parameter $\alpha \in (0, 1)$, it either rejects or accepts. With probability at least 99% over the sample access, if it accepts, it will output a distribution D_m over $\mathbb{R}^d$, with $d_{\text{TV}}(D_m, D'_{\mathbf{x}}) < \epsilon$, and a direction $\mathbf{v} \in \mathbb{S}^{d-1}$ and $i \in [k]$ such that $|\mathbb{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(\mathbf{v} \cdot \mathbf{x})^i] - \mathbb{E}_{\mathbf{x} \sim D_m}[(\mathbf{v} \cdot \mathbf{x})^i]| \geq c\Delta$, if there exists $\mathbf{w} \in \mathbb{S}^{d-1}$ and $j \in [k]$ such that $|\mathbb{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(\mathbf{w} \cdot \mathbf{x})^j] - \mathbb{E}_{\mathbf{x} \sim D_m}[(\mathbf{w} \cdot \mathbf{x})^j]| \geq \Delta$. Furthermore, if $D'_{\mathbf{x}}$ is a conditional distribution of $D_{\mathbf{x}}$ over some event with mass at least α , with probability at least 99%, the algorithm will accept.*

Lemma 6.4. *For every $1/2$ -isotropic logconcave distribution $D_{\mathbf{x}}$ over $\mathbb{R}^d$, for every $k \in \mathbb{Z}_+$, $\Delta \in (0, 1)$, there is a $(k, \Delta, 2^{-O(k^2)})$ testable discrepancy finder for $D_{\mathbf{x}}$ with sample complexity $d^{O(k)} \text{poly}_{k, \Delta}(1/(\alpha\epsilon))$ and time complexity $\text{poly}(d^k/\epsilon) 2^{\text{poly}_{k, \Delta}(1/\alpha)}$.*

Proof of Lemma 6.4. By Lemma 5.5, Lemma 5.7 and Theorem 5.1, suppose there is a degree k polynomial $Q(\mathbf{x})$ with $\|Q\|_{D_B,2} \leq O(\log^k(1/\alpha))$, such that $|\mathbf{E}_{\mathbf{x} \sim D_B}[Q(\mathbf{x})p(\mathbf{x})] - \mathbf{E}_{\mathbf{x} \sim D_m}[p(\mathbf{x})]| \leq 2^{-O(k^2)}\Delta$ for every degree k polynomial $p(\mathbf{x})$ of the form $p(\mathbf{x}) = \prod_{i=1}^k (\mathbf{w}^{(i)} \cdot \mathbf{x})$, where $\mathbf{w}^{(i)} \in \mathbb{S}^{d-1}$, then Algorithm 4 with such a function Q , will output in time $d^{O(k)}2^{\text{poly}(\log_k \Delta(1/\alpha))}$, a direction $\mathbf{v} \in \mathbb{S}^{d-1}$ and $i \in [k]$ such that $|\mathbf{E}_{\mathbf{x} \sim D_B}[(\mathbf{v} \cdot \mathbf{x})^i] - \mathbf{E}_{\mathbf{x} \sim D_m}[(\mathbf{v} \cdot \mathbf{x})^i]| \geq c\Delta$, if there exists $\mathbf{w} \in \mathbb{S}^{d-1}$ and $j \in [k]$ such that $|\mathbf{E}_{\mathbf{x} \sim D_B}[(\mathbf{w} \cdot \mathbf{x})^j] - \mathbf{E}_{\mathbf{x} \sim D_m}[(\mathbf{w} \cdot \mathbf{x})^j]| \geq \Delta$.

Given sample access to D'_x , we will first use $\text{poly}(1/(\alpha\epsilon))$ example to check if $\Pr_{\mathbf{x} \sim D'_x}[\|\mathbf{x}\|_2 > 100C \log(1/\alpha\epsilon)\sqrt{d}] < \alpha\epsilon/100$. If not, we will simply reject. Since D_B is a $1/2$ -logconcave distribution, we know from Fact C.2 that for $t > 1$, we have $\Pr_{\mathbf{x} \sim D_B}[\|\mathbf{x}\|_2 > Ct\sqrt{d}] < \exp(-t\sqrt{d})$, which implies that if D'_x is a conditional distribution of D_B over some event with mass α , then $\Pr_{\mathbf{x} \sim D'_x}[\|\mathbf{x}\|_2 > 100C \log(1/\alpha\epsilon)\sqrt{d}] < \alpha\epsilon/100$ and we will not reject with high probability at this step.

Notice that the distribution D_m defined as D'_x conditioned over examples with norm $\|\mathbf{x}\|_2 < r := 100C \log(1/\alpha\epsilon)\sqrt{d}$ satisfies $d_{TV}(D_m, D'_x) < \epsilon$. To complete the proof, it suffices to find a Q function discussed above. By Lemma 5.7, if D'_x is a conditional distribution of D_B over an event with mass α , such a function Q exists. So, it suffices to find such a polynomial Q using only sample access to D_m and D_B . Let $B_{d,k} = d^{O(k)}$. We will use linear program to find the coefficient $(q_\zeta)_\zeta$ of polynomial $Q(\mathbf{x}) = \sum_\zeta q_\zeta \mathbf{x}^\zeta$, with $\|Q\|_{D_B} \leq O(\log^k(1/\alpha))$ such that for each monomial $\mathbf{x}^\zeta$,

$$\left| \mathbf{E}_{\mathbf{x} \sim D_B}[Q(\mathbf{x})\mathbf{x}^\zeta] - \mathbf{E}_{\mathbf{x} \sim D_m}[\mathbf{x}^\zeta] \right| \leq B_{d,k}^{-1}2^{-O(k^2)}\Delta.$$

Notice that suppose this holds, then for every polynomial $p(\mathbf{x}) = \prod_{i=1}^k (\mathbf{w}^{(i)} \cdot \mathbf{x}) = \sum_\zeta a_\zeta \mathbf{x}^\zeta$, we have

$$\left| \mathbf{E}_{\mathbf{x} \sim D_B}[Q(\mathbf{x})p(\mathbf{x})] - \mathbf{E}_{\mathbf{x} \sim D_m}[p(\mathbf{x})] \right| \leq B_{d,k}^{-1}2^{-O(k^2)}\Delta \sum_\zeta |a_\zeta| \leq 2^{-O(k^2)}\Delta.$$

Here, we use the fact that $\sum_\zeta |a_\zeta| \leq d^{O(k)}$. The above linear program can be built by using samples from D_B and D_m to estimate $\mathbf{E}_{\mathbf{x} \sim D_B}[\mathbf{x}^\zeta]$ and $\mathbf{E}_{\mathbf{x} \sim D_m}[\mathbf{x}^\zeta]$ to error $B_{d,k}^{-1}2^{-O(k^2)}\Delta/4$. We remark that if such a function $Q(\mathbf{x})$ is not found, then we can just reject. Since D_B is logconcave, for each monomial, we can take the empirical mean of $\mathbf{x}^\zeta$ using $2^{-O(k^2)}\text{poly}(d^k/\Delta)$ examples to do this estimation. Since $\mathbf{x} \sim D_m$ has bounded norm r , we know that we can take $r^{O(k)}2^{O(k^2)}\text{poly}(d^k/\Delta)$ examples from D_m , we can estimate the expectation of $\mathbf{x}^\zeta$ to the desired accuracy. This completes the proof. $\square$

Proof of Theorem 6.2. We run the algorithm in Theorem 2.8 with a modified version of Algorithm 1. Let T, γ, τ , and ξ be as in Algorithm 1. Every time Algorithm 1 draws an example from D_x , instead, we draw it from D_B . Every time Algorithm 1 calls a discrepancy finder for $(D_R)_x$ and $((D_R)_f)_x$ over some region R , we will call a (k, Δ, c) -testable discrepancy finder for $(D_B)|_R$ with input distribution $((D_R)_f)_x$ with parameter weight parameter $\xi\epsilon$ and accuracy parameter $\epsilon' = (c\Delta\xi^2\epsilon^2)/T^{2k}$. We will terminate the algorithm mutually and reject if its running time exceeds $d^{O(k)}2^{\text{poly}_k \Delta(1/\epsilon)}$ or draws more than $d^{O(k)}\text{poly}_k \Delta(1/\epsilon)$ samples or it stops because the testable discrepancy finder rejects or a good split is not found.

If a partition $\mathcal{U}$ is output, we will check if $\sum_{R \in \mathcal{B}} \Pr_{(\mathbf{x}, y) \sim D_f}[\mathbf{x} \in R] \leq \eta/2$, where $\mathcal{B} = \{R \in \mathcal{U} \mid \Pr_{(\mathbf{x}, y) \sim D_f}[\mathbf{x} \in R] \leq \xi\}$. We will reject if it is not the case, and otherwise we will run the weak learner over each region R with $\Pr_{(\mathbf{x}, y) \sim D_f}[\mathbf{x} \in R] > \xi$ over the distribution output by the testable discrepancy finder. Since the distribution $(D_R)_m$ output by the testable discrepancy finder has TV distance at most ϵ' , over all such regions R , we only throw away at most ξ fraction of the mass. Furthermore, by the definition of the testable discrepancy finder, for every $R \in \mathcal{U}$, with $\Pr_{(\mathbf{x}, y) \sim D_f}[\mathbf{x} \in$

$R] > \xi$, it holds for every $\mathbf{u} \in \mathbb{S}^{d-1}$ and for every $i \in [k]$, $|\mathbf{E}_{\mathbf{x} \sim (D_B)_R}(\mathbf{u} \cdot \mathbf{x})^i - \mathbf{E}_{(\mathbf{x}, y) \sim (D_R)_m}(\mathbf{u} \cdot \mathbf{x})^i| \leq \Delta$, which implies that the modified version of Algorithm 1 will output a hypothesis with $\Omega(\eta\beta)$ advantage over D_f . Thus, the algorithm used for Theorem 2.8, will succeed. Furthermore, since the testable discrepancy finder will accept with a high probability when $D_{\mathbf{x}} = D_B$, we know that the whole testable learning algorithm will accept when $D_{\mathbf{x}} = D_B$ with a high probability. $\square$

7 Algorithmic Applications on Concrete Hypothesis Classes

In this section, we give algorithmic applications of Theorem 2.8 to several fundamental families of multi-index models.

Organization of This Section In Section 7.1, we will give basic technical lemmas that will be used for designing algorithms in this section. In Section 7.2, we introduce a generic family of concept classes that can be partially learned using Theorem 2.8. Specifically, we show that a wide range of “smooth” K -MIMs $\mathcal{H}$ have an efficient (ℓ, Δ, β, C) -weak learner, where the parameters ℓ, Δ, β, C only depend on geometric properties of the underlying class. This implies that any such family of smooth MIMs admits an agnostic partial learner under logconcave distributions with sample complexity $\text{poly}_{K, \Delta}(d, 1/\epsilon)$ and computational complexity $\text{poly}_{K, \Delta}(d)2^{\text{polylog}_{K, \Delta}(1/\epsilon)}$. We remark that the parameters (ℓ, Δ, β, C) for the weak learning algorithm above could be qualitatively quite large, if they are derived directly from the underlying geometric properties of the class. To rectify this, in Section 7.3, we show that MIMs that admit low-degree L_1 -sandwiching polynomials can be learned to any constant factor approximation using our framework. As a corollary of this result, in Section 7.4, we obtain such a constant factor approximation for agnostically learning functions of halfspaces and low-dimensional low-degree PTFs. In Section 7.5, we use a refinement of our framework to obtain improved learning algorithms for agnostic partial learning of halfspaces under logconcave distributions. Finally, in Section 7.6, we leverage these algorithms to obtain efficient and robust learners for multiclass linear classification.

7.1 Basic Tools for Weak Learning Algorithms

In this subsection, we give some basic tools that we will use in the following subsections. By Theorem 2.8, given a hypothesis class $\mathcal{H}$, to design an agnostic partial learner under logconcave distributions, it suffices to design an efficient (k, Δ, β, C) -weak learner for it.

Our weak learner makes use of the following simple framework. Given a distribution D over $\mathbb{R}^d \times \{\pm 1\}$, if we can find a real-valued function $g(\mathbf{x})$ such that $\mathbf{E}_{(\mathbf{x}, y) \sim D}[yg(\mathbf{x})]$ is non-trivial, then we can sample a random threshold $t \in [-1, 1]$ such that $\text{err}_D(\text{sign}(g(\mathbf{x})) - t)$ has small error with non-trivial probability.

To formally formulate this framework, we give the following lemma.

Lemma 7.1. *Let D be a distribution over $\mathbb{R}^d \times \{\pm 1\}$, and let $\mathcal{H}$ be a hypothesis class over $\mathbb{R}^d$. Let $h^* \in \mathcal{H}$ be such that $\text{err}_D(h^*) = \text{opt} = \inf_{h \in \mathcal{H}} \text{err}_D(h)$. For a real-valued function $p : \mathbb{R}^d \rightarrow \mathbb{R}$ and $\beta > 0$, the following holds.*

1. *If $\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[h^*(\mathbf{x})p(\mathbf{x})] > \beta$ and $\text{opt} < \beta^2(\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[p^2(\mathbf{x})])^{-1}/16$, then $\mathbf{E}_{(\mathbf{x}, y) \sim D}[yp(\mathbf{x})] > \beta/2$.*
2. *If $\mathbf{E}_{(\mathbf{x}, y) \sim D}[yp(\mathbf{x})] > \beta$, then the function $g(\mathbf{x}) : \mathbb{R}^d \rightarrow [-1, 1]$, defined by*

$$g(\mathbf{x}) = \text{proj}_{[-1, 1]} \left(\beta p(\mathbf{x}) \left(\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [p^2(\mathbf{x})] \right)^{-1/4} \right), \quad (5)$$

satisfies $\mathbf{E}_{(\mathbf{x},y) \sim D}[yg(\mathbf{x})] > \Omega(\beta^2/(\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[p^2(\mathbf{x})]))$.

Proof of Lemma 7.1. The first item of the statement holds because of the follows

$$\begin{aligned} \mathbf{E}_{(\mathbf{x},y) \sim D}[yp(\mathbf{x})] &= \mathbf{E}_{(\mathbf{x},y) \sim D}[h^*(\mathbf{x})p(\mathbf{x})] + \mathbf{E}_{(\mathbf{x},y) \sim D}[(y - h^*(\mathbf{x}))p(\mathbf{x})] \\ &\geq \mathbf{E}_{(\mathbf{x},y) \sim D}[h^*(\mathbf{x})p(\mathbf{x})] - 2 \mathbf{E}_{(\mathbf{x},y) \sim D}[\mathbb{1}\{y \neq h^*(\mathbf{x})\} |p(\mathbf{x})|] \\ &\geq \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[h^*(\mathbf{x})p(\mathbf{x})] - 2 \sqrt{\text{opt} \mathbf{E}_{(\mathbf{x},y) \sim D}[p(\mathbf{x})^2]} > \beta - \beta/4 = 3\beta/4. \end{aligned}$$

Here, in the first inequality, we use the fact that h^* takes values in $\{\pm 1\}$. The second inequality follows from Holder's inequality. The third inequality follows from the assumption that $\text{opt} < \beta^2(\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[p^2(\mathbf{x})])^{-1}/4$.

In the rest of the proof, we prove the second item of the statement. Let $W := 4(\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[p^2(\mathbf{x})])/\beta$. We have

$$\mathbf{E}_{(\mathbf{x},y) \sim D}[yg(\mathbf{x})] = \mathbf{E}_{(\mathbf{x},y) \sim D}[yg(\mathbf{x})\mathbb{1}\{|p(\mathbf{x})| < W\}] + \mathbf{E}_{(\mathbf{x},y) \sim D}[yg(\mathbf{x})\mathbb{1}\{|p(\mathbf{x})| > W\}].$$

By the choice of W , we have

$$\left| \mathbf{E}_{(\mathbf{x},y) \sim D}[yg(\mathbf{x})\mathbb{1}\{|p(\mathbf{x})| > W\}] \right| \leq \mathbf{Pr}_{\mathbf{x} \sim D_{\mathbf{x}}} [|p(\mathbf{x})| > W] \leq \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[p^2(\mathbf{x})]/W^2 \leq \beta^2/(16(\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[p^2(\mathbf{x})])).$$

Here the first inequality follows from the fact that $|yg(\mathbf{x})| \leq 1$ and the second inequality follows from Markov's inequality. Furthermore,

$$\begin{aligned} \mathbf{E}_{(\mathbf{x},y) \sim D}[yp(\mathbf{x})\mathbb{1}\{|p(\mathbf{x})| < W\}] &= \mathbf{E}_{(\mathbf{x},y) \sim D}[yp(\mathbf{x})] - \mathbf{E}_{(\mathbf{x},y) \sim D}[yp(\mathbf{x})\mathbb{1}\{|p(\mathbf{x})| > W\}] \\ &\geq \mathbf{E}_{(\mathbf{x},y) \sim D}[yp(\mathbf{x})] - \sqrt{\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[p^2(\mathbf{x})] \mathbf{Pr}_{\mathbf{x} \sim D_{\mathbf{x}}} [|p(\mathbf{x})| > W]} \\ &\geq \beta - \beta/2 = \beta/2, \end{aligned}$$

which implies by the definition of $g(\mathbf{x})$ that

$$\mathbf{E}_{(\mathbf{x},y) \sim D}[yg(\mathbf{x})\mathbb{1}\{|p(\mathbf{x})| < W\}] \geq 3\beta^2/(16(\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[p^2(\mathbf{x})])).$$

So, we conclude that $g(\mathbf{x}) : \mathbb{R}^d \rightarrow [-1, 1]$ satisfies $\mathbf{E}_{(\mathbf{x},y) \sim D}[yg(\mathbf{x})] > \Omega(\beta^2/(\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[p^2(\mathbf{x})]))$. $\square$

Lemma 7.1, shows that for an agnostic learning problem, suppose we have some real-valued function $p(\mathbf{x})$ such that the correlation $\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[h^*(\mathbf{x})p(\mathbf{x})]$ is non-trivial, where $h^*(\mathbf{x})$ is the hypothesis in the target class that achieves the optimal error, then as long as opt is reasonably small, we can find a function $g(\mathbf{x})$ with values in $[-1, 1]$ such that it has non-trivial correlation with respect to the noisy label y . Once such a $[-1, 1]$ valued function $g(\mathbf{x})$ is found, we can easily round it into a boolean function as a weak hypothesis over the noisy distribution D .

Fact 7.2. *Let X be a domain of examples and let D be a distribution over $X \times \{\pm 1\}$. Let $g(\mathbf{x}) : X \rightarrow [-1, 1]$ such that $\mathbf{E}_{\mathbf{x} \sim D}[yg(\mathbf{x})] > \gamma$, then $\mathbf{E}_{t \sim [-1, 1]}[\text{err}_D(\text{sign}(g(\mathbf{x}) - t))] < (1 - \gamma)/2$.*

Proof of Fact 7.2. To simplify the notation, let $g_t(\mathbf{x}) = \text{sign}(g(\mathbf{x}) - t)$, $t \in [-1, 1]$. Fix an example $(\mathbf{x}, y) \in X \times \{\pm 1\}$. If $y = 1$, then we have $\Pr_{t \sim [-1, 1]}[g_t(\mathbf{x}) \neq y] = (1 - g(\mathbf{x}))/2$, while if $y = -1$, then we have $\Pr_{t \sim [-1, 1]}[g_t(\mathbf{x}) \neq y] = (1 + g(\mathbf{x}))/2$. This implies that for every example $(\mathbf{x}, y)$, we have $\Pr_{t \sim [-1, 1]}[g_t(\mathbf{x}) \neq y] = (1 - yg(\mathbf{x}))/2$. Thus, we have

$$\mathbf{E}_{t \sim [-1, 1]}[\text{err}_D(g_t)] = \mathbf{E}_{(\mathbf{x}, y) \sim D} \left[\Pr_{t \sim [-1, 1]}[g_t(\mathbf{x}) \neq y] \right] = \mathbf{E}_{(\mathbf{x}, y) \sim D} \frac{1 - yg(\mathbf{x})}{2} \leq \frac{1 - \gamma}{2}.$$

□

7.2 Generic Family of MIMs

By Lemma 7.1 and Fact 7.2, to get an efficient weak learning algorithm for a class $\mathcal{H}$ under a distribution D , it suffices to show that when the moment information of the marginal distribution is close to that of some logconcave distribution, then we can efficiently find a polynomial with non-trivial correlation with respect to the optimal hypothesis.

Based on this observation, we show that a broad family of MIMs that satisfies a natural “smoothness” condition can be efficiently weakly learned. Specifically, we introduce the following generic family of MIMs.

Definition 7.3 ((K, η) -good Multi-Index Models). *We say that a family of Boolean functions $\mathcal{H}$ on $\mathbb{R}^d$ is a (K, η) -good class of Multi-Index Models if it satisfies the following properties:*

1. $\mathcal{H}$ is a family of K -Multi-index models (K -MIMs). That is, for every $h \in \mathcal{H}$ there is a K -dimensional subspace $V \subset \mathbb{R}^d$ such that $h(\mathbf{x}) = h(\text{proj}_V(\mathbf{x}))$. Equivalently, there is a class of Boolean functions $\mathcal{G}$ over $\mathbb{R}^K$ such that for every $h \in \mathcal{H}$ there exists $g \in \mathcal{G}$ and a matrix $\mathbf{W} \in \mathbb{R}^{K \times d}$ with $\mathbf{W}\mathbf{W}^\top = \mathbf{I}_{K \times K}$ so that $h(\mathbf{x}) = g(\mathbf{W}\mathbf{x})$.
2. Every $g \in \mathcal{G}$, such that $h(\mathbf{x}) = g(\mathbf{W}\mathbf{x})$ for some $h \in \mathcal{H}$, satisfies the following condition: For every ellipsoid $E = \mathcal{E}(\mathbf{u}, A)$, in $\mathbb{R}^K$, there is a ball $E' = \mathcal{E}(\mathbf{v}, A/\eta)$ such that $E' \subseteq E$ and $g(\mathbf{x})$ has a constant sign over E' .

Intuitively, any function in a (K, η) -good class of MIMs has a reasonably smooth decision boundary, so that the classification of points will not change too frequently within a small ball.

As a concrete example, consider the class of halfspaces—a basic class of 1-MIMs. Given any interval $I \subseteq \mathbb{R}$ and any one-dimensional linear threshold function, one can always find a subinterval $I' \subseteq I$, with half of the length, such that examples in I' have the same sign. This means the class of halfspaces is a $(1, 1/2)$ -good class of MIMs.

Our main learning result for (K, η) -good classes of MIMs is the following theorem.

Theorem 7.4 (Agnostic Partial Learning of Good MIMs). *Let $\mathcal{H}$ be a (K, η) -good class of MIMs over $\mathbb{R}^d$. There is an algorithm such that for every distribution D over $\mathbb{R}^d \times \{0, \pm 1\}$ with $D_{\mathbf{x}}$ logconcave and $W(D) \geq \alpha$, given $\epsilon \in (0, 1)$, it draws $(d\ell)^{O(\ell)} \text{poly}_{K, \eta}(1/(\alpha\epsilon))$ samples from D , runs in $(d\ell)^{O(\ell)} 2^{\text{poly}_{K, \eta}(\log(1/(\alpha\epsilon)))}$ time, where $\ell = \exp((K/\eta)^{O(K)})$, and outputs a hypothesis $\hat{h}$ such that with probability at least 99%, $\text{err}_{D_f}(\hat{h}) \leq \exp(O(\ell \log \ell)) \text{opt}_{D_f}(\mathcal{H}) + \epsilon$.*

By Theorem 2.8, to prove Theorem 7.4, it suffices to design an efficient weak learner for any (k, η) -good class of MIMs $\mathcal{H}$. To achieve this, we establish the following technical lemma, that gives a basic condition under which a general hypothesis class $\mathcal{H}$ can be efficiently weakly learned.

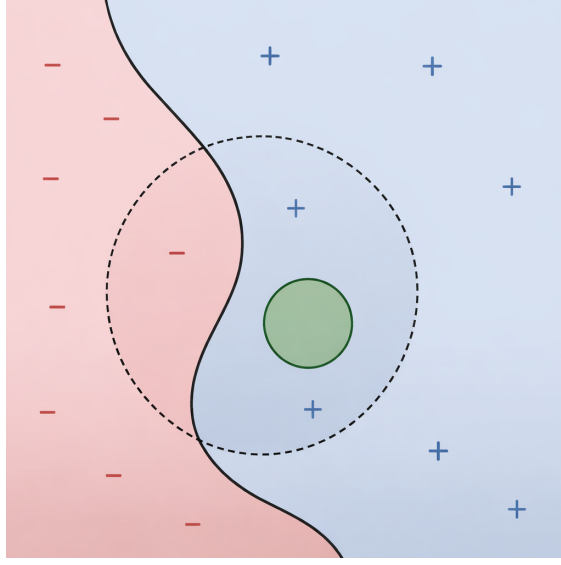


Figure 3: Generic Geometric Concept Class

Lemma 7.5. *Let $\mathcal{H}$ be a hypothesis class over $\mathbb{R}^d$ and $D'_\mathbf{x}$ be an arbitrary $1/2$ -isotropic logconcave distribution over $\mathbb{R}^d$. Suppose that for every $h \in \mathcal{H}$, the following statement is satisfied: there is a polynomial $p : \mathbb{R}^d \rightarrow \mathbb{R}$ with zero mean under $D'_\mathbf{x}$ such that either (i) if $h(\mathbf{x}) = 1$, then $p(\mathbf{x}) > \gamma$, or (ii) if $h(\mathbf{x}) = -1$, then $p(\mathbf{x}) < -\gamma$. If each polynomial p has the form $p(\mathbf{x}) = \sum_i c_i \prod_{j=1}^{\ell_i} (\mathbf{w}^{(ij)} \cdot \mathbf{x})$ with $\ell_i \leq \ell$, $\mathbf{w}^{(ij)} \in \mathbb{S}^{d-1}$, and $\max \left\{ \sum_i |c_i|, \sum_i |c_i| \left\| \prod_{j=1}^{\ell_i} (\mathbf{w}^{(ij)} \cdot \mathbf{x}) \right\|_{D_{\mathbf{x},2}^{\ell_i}} \right\} \leq B$, then there is an algorithm with the following guarantee. For any distribution D over $\mathbb{R}^d \times \{\pm 1\}$ such that for every $\mathbf{w} \in \mathbb{S}^{d-1}$ and $i \in [2\ell]$, $|\mathbf{E}_{\mathbf{x} \sim D_\mathbf{x}}[(\mathbf{w} \cdot \mathbf{x})^i] - \mathbf{E}_{\mathbf{x} \sim D'_\mathbf{x}}[(\mathbf{w} \cdot \mathbf{x})^i]| \leq \Delta \leq \gamma B^{-1} 2^{-O(\ell)}$, with $\text{opt}_D(\mathcal{H}) < O((\gamma/B)^2)$, the algorithm draws $\text{poly}(d^\ell, B/\gamma)$ examples from D , runs in time $\text{poly}(d^\ell, B/\gamma)$, and outputs a hypothesis $\hat{h}$ such that $\text{err}_D(\hat{h}) < 1/2 - \Omega((\gamma/B)^2)$ with probability at least 99%.*

Roughly speaking, the condition in Lemma 7.5 assumes that each function $h \in \mathcal{H}$ has a one-sided approximation by a mean-zero low-degree polynomial p , which can be written as a sum of products of linear functions with small L_2 -norms under a logconcave distribution $D'_\mathbf{x}$. Specifically, this means that $\mathbf{E}_{\mathbf{x} \sim D'_\mathbf{x}}[p(\mathbf{x})] = 0$ and whenever $h(\mathbf{x}) = 1$, it must be the case that $p(\mathbf{x}) > \gamma > 0$. The structure of $p(\mathbf{x})$ implies that if $\mathbf{x}$ is drawn from $D_\mathbf{x}$, a distribution that approximately matches moments with $D'_\mathbf{x}$, the mean and the L_2 norm of $p(\mathbf{x})$ will not change too much. This implies if we take $h(\mathbf{x})$ to be the optimal hypothesis, then, under $D_\mathbf{x}$, $h(\mathbf{x})$ and $p(\mathbf{x})$ must have non-trivial correlation—because $\mathbf{E}_{\mathbf{x} \sim D_\mathbf{x}}[h(\mathbf{x})p(\mathbf{x})] \approx \mathbf{E}_{\mathbf{x} \sim D_\mathbf{x}}[(h(\mathbf{x}) + 1)p(\mathbf{x})]$. By Lemma 7.1, this implies that there is some low-degree polynomial that has non-trivial correlation with the label y , if $\text{opt}_D(\mathcal{H})$ is sufficiently small. Therefore, we can run a simple polynomial regression subroutine to find such a good polynomial and turn it into a weak hypothesis.

Proof of Lemma 7.5. Let $h^* \in \mathcal{H}$ be the hypothesis, such that $\text{err}_D(h^*) = \text{opt}$ and let $p(\mathbf{x})$ be a degree ℓ polynomial in the statement for h^* . Without loss of generality, we assume that when

$h^*(\mathbf{x}) = 1$, $p(\mathbf{x}) > \gamma$. Since $\mathbf{E}_{\mathbf{x} \sim D'_\mathbf{x}}[p(\mathbf{x})] = 0$, we know that under $D_\mathbf{x}$, we have

$$\begin{aligned} \left| \mathbf{E}_{\mathbf{x} \sim D_\mathbf{x}}[p(\mathbf{x})] \right| &= \left| \sum_i c_i \mathbf{E}_{\mathbf{x} \sim D_\mathbf{x}} \left[\prod_{j=1}^{\ell_i} (\mathbf{w}^{(ij)} \cdot \mathbf{x}) \right] - \mathbf{E}_{\mathbf{x} \sim D'_\mathbf{x}} \left[\prod_{j=1}^{\ell_i} (\mathbf{w}^{(ij)} \cdot \mathbf{x}) \right] \right| \\ &\leq \sum_i |c_i| \left| \mathbf{E}_{\mathbf{x} \sim D_\mathbf{x}} \left[\prod_{j=1}^{\ell_i} (\mathbf{w}^{(ij)} \cdot \mathbf{x}) \right] - \mathbf{E}_{\mathbf{x} \sim D'_\mathbf{x}} \left[\prod_{j=1}^{\ell_i} (\mathbf{w}^{(ij)} \cdot \mathbf{x}) \right] \right| \\ &\leq (2e)^\ell \sum_i |c_i| \Delta \leq O(\gamma). \end{aligned}$$

Here, we use the fact that when $|\mathbf{E}_{\mathbf{x} \sim D_\mathbf{x}}[(\mathbf{w} \cdot \mathbf{x})^i] - \mathbf{E}_{\mathbf{x} \sim D'_\mathbf{x}}[(\mathbf{w} \cdot \mathbf{x})^i]| \leq \Delta$ for every $\mathbf{w} \in \mathbb{S}^{d-1}$ and $i \in [2\ell]$, $\left| \mathbf{E}_{\mathbf{x} \sim D_\mathbf{x}} \left[\prod_{j=1}^{\ell_i} (\mathbf{w}^{(ij)} \cdot \mathbf{x}) \right] - \mathbf{E}_{\mathbf{x} \sim D'_\mathbf{x}} \left[\prod_{j=1}^{\ell_i} (\mathbf{w}^{(ij)} \cdot \mathbf{x}) \right] \right| \leq (2e)^\ell \Delta$, when $\sum \ell_{ij} \leq \ell$.

This implies that

$$\begin{aligned} \mathbf{E}_{\mathbf{x} \sim D_\mathbf{x}}[h^*(\mathbf{x})p(\mathbf{x})] &= \mathbf{E}_{\mathbf{x} \sim D_\mathbf{x}}[(h^*(\mathbf{x}) + 1)p(\mathbf{x})] - \mathbf{E}_{\mathbf{x} \sim D_\mathbf{x}}[p(\mathbf{x})] \\ &\geq \mathbf{E}_{\mathbf{x} \sim D_\mathbf{x}}[(h^*(\mathbf{x}) + 1)p(\mathbf{x})] - \gamma/20. \end{aligned}$$

Notice that if $\Pr_{\mathbf{x} \sim D_\mathbf{x}}[h^*(\mathbf{x}) = 1] < 1/3$, then a constant polynomial $p^-(\mathbf{x}) \equiv -1$ satisfies the statement. So, we can assume that $\Pr_{\mathbf{x} \sim D_\mathbf{x}}[h^*(\mathbf{x}) = 1] > 1/3$. This implies that

$$\mathbf{E}_{\mathbf{x} \sim D_\mathbf{x}}[h^*(\mathbf{x})p(\mathbf{x})] \geq \gamma/3 - \gamma/20 \geq \gamma/6.$$

On the other hand, we have

$$\|p\|_{D_\mathbf{x}, 2} \leq \sum_i |c_i| \left\| \prod_{j=1}^{\ell_i} (\mathbf{w}^{(ij)} \cdot \mathbf{x}) \right\|_{D_\mathbf{x}, 2} \leq O \left(\sum_i |c_i| \left\| \prod_{j=1}^{\ell_i} (\mathbf{w}^{(ij)} \cdot \mathbf{x}) \right\|_{D'_\mathbf{x}, 2} \right) = O(B).$$

By Lemma 7.1, it follows that when $\text{opt} < O((\gamma/B)^2)$, we have $\mathbf{E}_{(\mathbf{x}, y) \sim D}[yp(\mathbf{x})] > \gamma/8$.

Using polynomial regression [KKMS08], we can draw $\text{poly}(d^\ell/\gamma)$ examples from D , run in $\text{poly}(d^\ell/\gamma)$ time, and compute a polynomial $\hat{p}$ such that $\mathbf{E}_{(\mathbf{x}, y) \sim D}[y\hat{p}(\mathbf{x})] > \Omega(\gamma)$ and $\mathbf{E}_{\mathbf{x} \sim D_\mathbf{x}}[\hat{p}^2(\mathbf{x})] \leq O(B)$. By Lemma 7.1 and Fact 7.2, we can round the computed polynomial $\hat{p}(\mathbf{x})$ into a Boolean function $\hat{h}(\mathbf{x})$ such that $\text{err}_D(\hat{h}) \leq 1/2 - \Omega((\gamma^2/B^2))$. This completes the proof of Lemma 7.5. $\square$

To prove Theorem 7.4, it suffices to show that any (K, η) -good class of MIMs $\mathcal{H}$ satisfies the assumption given by Lemma 7.5, which automatically gives a weak learning algorithm. Formally, we give Lemma 7.6 below.

Lemma 7.6. *Let $\mathcal{H}$ be a (K, η) -good MIM family over $\mathbb{R}^d$. There is an algorithm $\mathcal{A}$ that draws $(d\ell)^{O(\ell)}$ examples, runs in $(d\ell)^{O(\ell)}$ time and $(2\ell, \exp(-\ell \log \ell), \exp(-O(\ell \log \ell)), \exp(O(\ell \log \ell)))$ -weakly learns $\mathcal{H}$ with probability at least 99%, where $\ell = \exp(O(K^4/\alpha^4))$ and $\alpha = \eta^K 2^{-O(K \log K)}$.*

Before we proceed with the proof of the above lemma, we note that Theorem 7.4 is a direct corollary of Theorem 2.8 and Lemma 7.6.

We give the intuition behind Lemma 7.6. Since the value of a function $h \in \mathcal{H}$ at a point $\mathbf{x}$ only depends on the projection of $\mathbf{x}$ onto some K -dimensional subspace V , for a given logconcave distribution $D'_\mathbf{x}$, we will instead consider the distribution $\mathbf{x}' \sim D_\mathbf{x}^V$, the projection of $\mathbf{x}$ onto V ,

which is still logconcave. By the anti-concentration property of logconcave distributions, points within a ball centered at the mean of $D_{\mathbf{x}}^V$ have density at least $2^{-O(K)}$. By the goodness of $\mathcal{H}$, after a proper affine transformation, there is a ball of volume $\eta^{O(K)}$ in this region with negative sign and thus contains a box S within the ball with $(\eta/K)^{O(K)}$ volume. This implies that the box S has non-trivial probability mass under $D_{\mathbf{x}}^V$. We will construct a polynomial $p(\mathbf{x}')$ that always upper bounds the indicator $\mathbb{1}\{\mathbf{x}' \notin S\}$ and is close to it in L_1 norm under $D_{\mathbf{x}}^V$. This implies that the polynomial $p'(\mathbf{x}') = p(\mathbf{x}') - \mathbf{E}_{\mathbf{x}' \sim D_{\mathbf{x}}^V}[p(\mathbf{x}')] is slightly larger than 0, whenever $\mathbf{x}'$ is positive. To get such a polynomial, we will make use of the following result from [KKM13] which gives a low-degree construction of a pair of polynomials that “sandwich” any function of halfspaces.$

Theorem 7.7. [Theorem 15 in [KKM13]] *Let $D_{\mathbf{x}}$ be any logconcave distribution in $\mathbb{R}^d$, and let $f(\mathbf{x}) = g(h_1(\mathbf{x}), \dots, h_m(\mathbf{x})) : \mathbb{R}^d \rightarrow \{0, 1\}$ be a function of m halfspaces, where $h_i(\mathbf{x}) = \text{sign}(w^{(i)} \cdot \mathbf{x} + t_i)$. For every $\epsilon > 0$, there is a pair of polynomials p^u, p^l with degree $\exp(O(m/\epsilon)^4)$, such that for every $\mathbf{x} \in \mathbb{R}^d$, $p^l(\mathbf{x}) \leq f(\mathbf{x}) \leq p^u(\mathbf{x})$, and $\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [|p^u(\mathbf{x}) - p^l(\mathbf{x})|] \leq \epsilon$.*

Finally, since the constructed polynomial $p'(\mathbf{x}')$ is a low-degree polynomial over a K -dimensional subspace V , we can expand each monomial of p' in a form of $\prod_i (\mathbf{w}^{(i)} \cdot \mathbf{x})$, where $w^{(i)}$ is a standard basis vector of V . This implies that we can control the size of p' as a function that only depends on K . Therefore, $\mathcal{H}$ satisfies the condition of Lemma 7.5, and we obtain a weak learning algorithm for $\mathcal{H}$. We give the proof of Lemma 7.6 below.

Proof of Lemma 7.6. Let $L : \mathbb{R}^d \rightarrow \mathbb{R}^d$, be any invertible affine transformation. By the definition of a good MIM, we know that if $\mathcal{H}$ is (K, η) -good, then $\mathcal{H}_L := \{h(L\mathbf{x}) \mid h \in \mathcal{H}\}$ is also (K, η) -good. It suffices to give an algorithm such that for every input distribution D , if there is an $1/2$ -isotropic logconcave distribution that has low-degree moments approximately matching with $D_{\mathbf{x}}$ and $\text{opt}_D(\mathcal{H})$ is small, then we can get a weak hypothesis over D .

Let $D'_{\mathbf{x}}$ be any $1/2$ -isotropic logconcave distribution. Let $h(\mathbf{x}) \in \mathcal{H}$ such that $h(\mathbf{x}) = h(\text{proj}_V(\mathbf{x}))$, where V is a K -dimensional subspace. Write $h(\mathbf{x}) = g(\mathbf{W}\mathbf{x})$, for $g \in \mathcal{G}$ and $\mathbf{W} \in \mathbb{R}^{K \times d}$ such that $\mathbf{W}\mathbf{W}^\top = \mathbf{I}_{K \times K}$.

Denote by $D_{\mathbf{x}}^{(V)}$ the distribution of $\mathbf{x}' = \mathbf{W}\mathbf{x}$ for $\mathbf{x} \sim D'_{\mathbf{x}}$ and by $\boldsymbol{\mu} = \mathbf{E}_{\mathbf{x}' \sim D_{\mathbf{x}}^{(V)}}[\mathbf{x}']$. Notice that $D_{\mathbf{x}}^{(V)}$ is a $1/2$ -isotropic logconcave distribution over $\mathbb{R}^K$. We without loss of generality, assume that $\boldsymbol{\mu} = 0$.

Let $B = \mathbb{B}^{K-1}(0, 1/18)$. Since $\mathcal{H}$ is (K, η) -good, we know that there is some $B' = \mathbb{B}^{K-1}(\mathbf{v}, \eta/9)$ such that $B' \subseteq B$ and for every $\mathbf{x}' \in B'$, we have $g(\mathbf{x}') \equiv 1$ or $g(\mathbf{x}') \equiv -1$. Without loss of generality, we assume $g(\mathbf{x}') \equiv -1$. Let $S \subseteq B'$ be the largest box contained in B' . We will next lower bound $\Pr_{\mathbf{x}' \sim D_{\mathbf{x}}^{(V)}}[\mathbf{x}' \in S]$.

Notice that the largest box contained in $\mathbb{B}^{K-1}(0, 1/18)$ has volume $2^{-O(K \log K)}$, which implies that $\text{Vol}(S) \geq \eta^K 2^{-O(K \log K)}$. By Fact A.1, we know that for every $\mathbf{x}' \in S$, the density $D_{\mathbf{x}}^V(\mathbf{x}') \geq 2^{-8K}$. This implies that

$$\Pr_{\mathbf{x}' \sim D_{\mathbf{x}}^{(V)}}[\mathbf{x}' \in S] = \int_S D_{\mathbf{x}}^{(V)}(\mathbf{x}') d\mathbf{x}' \geq 2^{-8K} \text{Vol}(S) \geq \eta^K 2^{-O(K \log K)}.$$

Notice that the box S is an intersection of $2K$ halfspaces. By Theorem 7.7, there is a polynomial $p(\mathbf{x}')$ with degree $\ell = \exp(O(K^4/\alpha^4))$ such that

$$p(\mathbf{x}') \leq \mathbb{1}\{\mathbf{x}' \in S\}, \quad \mathbf{E}_{\mathbf{x}' \sim D_{\mathbf{x}}^{(V)}}[|\mathbb{1}\{\mathbf{x}' \in S\} - p(\mathbf{x}')|] \leq \alpha.$$

Since $\Pr_{\mathbf{x}' \sim D_{\mathbf{x}}^{(V)}}[\mathbf{x}' \in S] \geq \eta^K 2^{-O(K \log K)}$, by setting $\alpha = \Pr_{\mathbf{x}' \sim D_{\mathbf{x}}^{(V)}}[\mathbf{x}' \in S]/2$, the polynomial $p'(\mathbf{x}') = \mathbf{E}_{\mathbf{x}' \sim D_{\mathbf{x}}^{(V)}}[p(\mathbf{x}')] - p(\mathbf{x}')$ satisfies $p'(\mathbf{x}') > \alpha$ for every $\mathbf{x}' \notin S$. In particular, since $g(\mathbf{x}') = -1$ for every $\mathbf{x}' \in S$, we know that $p'(\mathbf{x}') > \alpha$, when $g(\mathbf{x}') = 1$.

In particular, we have $p'(\mathbf{W}\mathbf{x}) > \alpha$ whenever $h(\mathbf{x}) = 1$ and $\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}'}[p'(\mathbf{W}\mathbf{x})] = 0$. Let $\mathbf{w}^{(1)}, \dots, \mathbf{w}^{(K)}$ be the row vectors of $\mathbf{W}$. Then we have

$$p'(\mathbf{W}\mathbf{x}) = \sum_i c_i \prod_{j=1}^{\ell_i} (\mathbf{w}^{(j)} \cdot \mathbf{x}),$$

with $\ell_i \leq \ell$, for all i . Since $\mathbf{E}_{\mathbf{x}' \sim D_{\mathbf{x}}^{(V)}}[|p'(\mathbf{x}')|] \leq 2\alpha$, it follows from Fact A.12 that

$$\sum_i |c_i| \left\| \prod_{j=1}^{\ell_i} (\mathbf{w}^{(j)} \cdot \mathbf{x}) \right\|_{D_{\mathbf{x},2}'} \leq \exp(O(\ell \log \ell)) \mathbf{E}_{\mathbf{x}' \sim D_{\mathbf{x}}^{(V)}}[|p'(\mathbf{x}')|] = \exp(O(\ell \log \ell)).$$

Let $B = \exp(O(\ell \log \ell))$. By Lemma 7.5, we have that when the input distribution D has its marginal distribution $D_{\mathbf{x}}$ approximately matching the moments of $D_{\mathbf{x}}'$, we can weakly learn $\mathcal{H}$ in time $(d\ell)^{O(\ell)}$, when $|\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(\mathbf{w} \cdot \mathbf{x})^i] - \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}'}[(\mathbf{w} \cdot \mathbf{x})^i]| \leq \Delta \leq \alpha B^{-1} 2^{-O(\ell)} = \exp(-O(\ell \log \ell))$ and $\text{opt} \leq (\alpha/B)^2 = \exp(-O(\ell \log \ell))$. This completes our proof. $\square$

7.3 Weak Agnostic Learning via Sandwiching Polynomials

Although Theorem 7.4 gives agnostic partial learning algorithms for a broad class of MIMs under logconcave distributions, the parameters of the (k, Δ, β, C) -weak learner that we derive from the geometry of $\mathcal{H}$ are not good in general. As a result, the final learning algorithm only gives a reasonable hypothesis when opt is very small.

In this section, we show that if a hypothesis class $\mathcal{H}$ admits low-degree L_1 sandwiching polynomial approximation, then we are able to perform agnostic partial learning for such a hypothesis class to error $(1+c)\text{opt}$, for any desired constant $c > 0$, with much better complexity.

We formulate the class of functions we are interested in as follows.

Definition 7.8 (MIM Classes with Low-Degree L_1 -Sandwiching Polynomials). *Let $\mathcal{H}$ be a hypothesis class over $\mathbb{R}^d$ and let $\ell : (0, 1] \rightarrow \mathbb{Z}_+$ be an integer-valued function. We say that $\mathcal{H}$ is a K -MIM that admits degree- ℓ L_1 sandwiching polynomials, if the following conditions hold.*

1. $\mathcal{H}$ is a family of K -Multi-index models (K -MIMs). That is, for every $h \in \mathcal{H}$ there is a K -dimensional subspace $V \subset \mathbb{R}^d$ such that $h(\mathbf{x}) = h(\text{proj}_V(\mathbf{x}))$. Equivalently, there is a class of Boolean functions $\mathcal{G}$ over $\mathbb{R}^K$ such that for every $h \in \mathcal{H}$ there exists $g \in \mathcal{G}$ and a matrix $\mathbf{W} \in \mathbb{R}^{K \times d}$ with $\mathbf{W}\mathbf{W}^\top = \mathbf{I}_{K \times K}$ so that $h(\mathbf{x}) = g(\mathbf{W}\mathbf{x})$.
2. For any $c \in (0, 1)$, for every $g \in \mathcal{G}$, for every logconcave distribution $D_{\mathbf{x}}$ over $\mathbb{R}^K$, there is a pair of degree $\ell(c)$ -polynomials p^l, p^u such that for every $\mathbf{x}' \in \mathbb{R}^K$

$$p^l(\mathbf{x}') \leq g(\mathbf{x}') \leq p^u(\mathbf{x}'), \quad \mathbf{E}_{\mathbf{x}' \sim D_{\mathbf{x}}} [|p^u(\mathbf{x}') - p^l(\mathbf{x}')|] \leq c.$$

Our main result of this subsection is the following theorem.

Theorem 7.9 (L_1 -sandwiching Polynomial Approximation Implies QPTAS). *Let $\mathcal{H}$ be a hypothesis class over $\mathbb{R}^d$ and let $\ell : (0, 1] \rightarrow \mathbb{Z}_+$ be an integer-valued function. If $\mathcal{H}$ is a family of K -MIMs that admits degree- ℓ L_1 sandwiching polynomials, then there is an algorithm such that for every input distribution D over $\mathbb{R}^d \times \{0, \pm 1\}$ of weight at least α , with logconcave marginal, for every $c, \epsilon \in (0, 1)$, it draws $(d\ell)^{O(\ell)} \text{poly}_{K,c}(1/(\alpha\epsilon))$ examples from D , runs in $(d\ell)^{O(\ell)} 2^{\text{poly}_{K,c}(\log(1/(\alpha\epsilon)))}$ time, where $\ell = \ell(c/6)$, and outputs a hypothesis $\hat{h}$ such that with probability at least 99%, $\text{err}_{D_f}(\hat{h}) \leq (1 + c)\text{opt}_{D_f}(\mathcal{H}) + \epsilon$.*

By Theorem 2.8, to prove Theorem 7.9, the central technical lemma we need is that any MIM that admits low-degree sandwiching polynomials is weakly learnable.

Lemma 7.10. *Let $\mathcal{H}$ be a hypothesis class over $\mathbb{R}^d$ and let $\ell : (0, 1] \rightarrow \mathbb{Z}_+$ be an integer-valued function. If $\mathcal{H}$ is a family of K -MIMs that admits degree- ℓ L_1 sandwiching polynomials, then there is an algorithm such that for every $c \in (0, 1)$ and for every input distribution D over $\mathbb{R}^d \times \{\pm 1\}$ such that there is some $1/2$ -isotropic logconcave distribution D'_x over $\mathbb{R}^d$ that satisfies for every unit vector $\mathbf{w}$ and $i \in [2\ell]$, $|\mathbf{E}_{\mathbf{x} \sim D_x}[(\mathbf{w} \cdot \mathbf{x})^i] - \mathbf{E}_{\mathbf{x} \sim D'_x}[(\mathbf{w} \cdot \mathbf{x})^i]| < \Delta < \exp(-O(K\ell \log \ell))$ with $\text{opt}_D(\mathcal{H}) \leq (1 - c)/2$, where $\ell = \ell(c/3)$, it draws $\text{poly}((d\ell)^\ell/c^2)$ examples from D , runs in $\text{poly}((d\ell)^\ell/c^2)$ time and outputs a hypothesis $\hat{h}$ with error $\text{err}_D(\hat{h}) < 1/2 - c \exp(-O(\ell \log \ell))$ with probability at least 99%.*

We sketch the pseudo-code of the weak learning algorithm in Algorithm 5.

Algorithm 5 POLYNOMIALWEAKLEARNING(Weak learning using Polynomial)

- 1: **Input:** Sample access to D , diameter parameter $B > 0$, advantage parameter β , degree of polynomial ℓ .
 - 2: **Output:** $\hat{h}$: weak hypothesis over D .
 - 3: $h_+(\mathbf{x}) := 1$, $h_-(\mathbf{x}) := -1$, $\mathcal{C} := \{h_+, h_-\}$
 - 4: Draw $S \sim D^m$ a set of $m = \tilde{O}(d^\ell/\beta^2)$ i.i.d. examples from D
 - 5: Find polynomial $p(\mathbf{x})$, with $\mathbf{E}_{\mathbf{x} \sim D}[p^2(\mathbf{x})] \leq B$ that maximizes $\sum_{(\mathbf{x}, y) \in S} (yp(\mathbf{x}))$ through polynomial regression
 - 6: $g(\mathbf{x}) = \text{proj}_{[-1, 1]}(\beta p(\mathbf{x})(\mathbf{E}_{\mathbf{x} \sim D_x}[p^2(\mathbf{x})])^{-1}/4)$
 - 7: **for** $i \in [O(B^2/\beta^2)]$ **do**
 - 8: Sample t' uniformly from $[-1, 1]$ and define $h_i(\mathbf{x}) := \text{sign}(g(\mathbf{x}) - t')$
 - 9: $\mathcal{C} \leftarrow \mathcal{C} \cup \{h_i(\mathbf{x})\}$
 - 10: Draw $S' \sim D^m$ a set of $m = \tilde{O}(d/\beta^2)$ i.i.d. examples from D
 - 11: **return** $\hat{h} := \text{argmin}_{h \in \mathcal{C}} \sum_{(\mathbf{x}, y) \in S'} \mathbb{1}\{h(\mathbf{x}) \neq y\}$.
-

Intuitively, if we pick up h^* to be the optimal hypothesis in $\mathcal{H}$, then $\mathbf{E}_{(\mathbf{x}, y) \sim D}[h^*(\mathbf{x})y] > c$. Then, $\mathbf{E}_{(\mathbf{x}, y) \sim D}[p^u(\mathbf{x})y]$ is at most $\mathbf{E}_{\mathbf{x} \sim D_x}[p^u(\mathbf{x}) - p^l(\mathbf{x})]$ far from $\mathbf{E}_{(\mathbf{x}, y) \sim D}[h^*(\mathbf{x})y]$. Since the sandwiching polynomials only depend on the projection of $\mathbf{x}$ onto some k -dimensional subspace V , we can upper bound the sizes of these polynomials as a function of k . As a result, as long as D_x approximately matches its low-degree moment with some logconcave distribution D'_x , $\mathbf{E}_{\mathbf{x} \sim D_x}[p^u(\mathbf{x}) - p^l(\mathbf{x})]$ is close to $\mathbf{E}_{\mathbf{x} \sim D'_x}[p^u(\mathbf{x}) - p^l(\mathbf{x})]$. This implies that $\mathbf{E}_{(\mathbf{x}, y) \sim D}[p^u(\mathbf{x})y]$ is at least $\Omega(c)$, which gives us a weak hypothesis, whenever $\text{opt} < (1 - c)/2$.

Proof of Lemma 7.10. Let D'_x be a $1/2$ -isotropic logconcave distribution. Let $h^*(\mathbf{x})$ be the optimal hypothesis in $\mathcal{H}$ such that $\text{err}_D(h^*(\mathbf{x})) = \text{opt} = \text{opt}_D(\mathcal{H})$. Let V be a K dimensional subspace such that $h^*(\mathbf{x}) = h(\text{proj}_V(\mathbf{x}))$. Write $h^*(\mathbf{x}) = g(\mathbf{W}\mathbf{x})$, with $\mathbf{W}\mathbf{W}^\top = \mathbf{I}_{K \times K}$. We denote by $D_x^{(V)}$ the distribution of $\mathbf{x}' = \mathbf{W}\mathbf{x}$, $\mathbf{x} \sim D'_x$

We notice that $\mathbf{E}_{(\mathbf{x},y) \sim D}[g(\mathbf{x}')y] = \mathbf{E}_{(\mathbf{x},y) \sim D}[h^*(\mathbf{x})y] = 1 - 2\text{opt} > c$. Let $p^u(\mathbf{x}')$, $p^l(\mathbf{x}')$ be upper and lower polynomials for $g(\mathbf{x}')$

$$\mathbf{E}_{\mathbf{x}' \sim D_{\mathbf{x}}^{(V)}}[|p^u(\mathbf{x}') - p^l(\mathbf{x}')|] \leq c/3.$$

We write $p^u(\mathbf{x}') = \sum_{|\alpha| \leq \ell} a_\alpha(\mathbf{x}')^\alpha$ and $p^l(\mathbf{x}') = \sum_{|\alpha| \leq \ell} b_\alpha(\mathbf{x}')^\alpha$. Since $\mathbf{E}_{\mathbf{x}' \sim D_{\mathbf{x}}^{(V)}}[|p^u(\mathbf{x}')|] \leq 2$ and $\mathbf{E}_{\mathbf{x}' \sim D_{\mathbf{x}}^{(V)}}[|p^l(\mathbf{x}')|] \leq 2$, by Fact A.12, we have

$$\sum_{|\alpha| \leq k} \left(|a_\alpha|^2 \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(\mathbf{x}')^{2\alpha}] \right)^{1/2} \leq \exp(O(k\ell \log \ell)), \quad \sum_{|\alpha| \leq k} \left(|b_\alpha|^2 \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(\mathbf{x}')^{2\alpha}] \right)^{1/2} \leq \exp(O(K\ell \log \ell)).$$

Since $|\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(\mathbf{w} \cdot \mathbf{x})^i] - \mathbf{E}_{\mathbf{x} \sim D'_{\mathbf{x}}}[(\mathbf{w} \cdot \mathbf{x})^i]| < \Delta$, for $\mathbf{w} \in \mathbb{S}^{d-1}$ and $i \in [2\ell]$, we have

$$\begin{aligned} \left| \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [p^u(\mathbf{x}')] - \mathbf{E}_{\mathbf{x} \sim D'_{\mathbf{x}}} [p^u(\mathbf{x}')] \right| &= \left| \sum_{\alpha} a_{\alpha} (\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(\mathbf{x}')^{\alpha}] - \mathbf{E}_{\mathbf{x} \sim D'_{\mathbf{x}}}[(\mathbf{x}')^{\alpha}]) \right| \\ &\leq \sum_{\alpha} |a_{\alpha}| \left| (\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(\mathbf{x}')^{\alpha}] - \mathbf{E}_{\mathbf{x} \sim D'_{\mathbf{x}}}[(\mathbf{x}')^{\alpha}]) \right| \\ &\leq (2e)^{\ell} \sum_{\alpha} |a_{\alpha}| \Delta \leq c/6. \end{aligned}$$

Similarly, we have

$$\begin{aligned} \left| \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [p^l(\mathbf{x}')] - \mathbf{E}_{\mathbf{x} \sim D'_{\mathbf{x}}} [p^l(\mathbf{x}')] \right| &= \left| \sum_{\alpha} b_{\alpha} (\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(\mathbf{x}')^{\alpha}] - \mathbf{E}_{\mathbf{x} \sim D'_{\mathbf{x}}}[(\mathbf{x}')^{\alpha}]) \right| \\ &\leq \sum_{\alpha} |b_{\alpha}| \left| (\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(\mathbf{x}')^{\alpha}] - \mathbf{E}_{\mathbf{x} \sim D'_{\mathbf{x}}}[(\mathbf{x}')^{\alpha}]) \right| \\ &\leq (2e)^{\ell} \sum_{\alpha} |b_{\alpha}| \Delta \leq c/6. \end{aligned}$$

This implies that

$$\begin{aligned} \left| \mathbf{E}_{(\mathbf{x},y) \sim D} [h^*(\mathbf{x})y] - \mathbf{E}_{(\mathbf{x},y) \sim D} [p^u(\mathbf{W}\mathbf{x})y] \right| &\leq \mathbf{E}_{(\mathbf{x},y) \sim D} [p^u(\mathbf{W}\mathbf{x}) - p^l(\mathbf{W}\mathbf{x})] \\ &\leq \mathbf{E}_{\mathbf{x}' \sim D_{\mathbf{x}}^{(V)}} [|p^u(\mathbf{x}') - p^l(\mathbf{x}')|] + c/3 \leq 2c/3, \end{aligned}$$

which gives us $\mathbf{E}_{(\mathbf{x},y) \sim D}[yp^u(\mathbf{W}\mathbf{x})] > c/3$. By Fact A.10, we have $\mathbf{E}_{\mathbf{x} \sim D'_{\mathbf{x}}}[(p^u(\mathbf{x}))^2] \leq \exp(\ell \log \ell)$. Using polynomial regression [KKMS08], we can draw $\text{poly}((\ell d)^{\ell}/c^2)$ examples from D and run in $\text{poly}((\ell d)^{\ell})$ time to find a polynomial $\hat{p}$ such that $\mathbf{E}_{(\mathbf{x},y) \sim D}[y\hat{p}(\mathbf{x})] > c/6$ and $\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[\hat{p}^2(\mathbf{x})] \leq \exp(O(\ell \log \ell))$. By Lemma 7.1 and Fact 7.2, we can round the computed polynomial $\hat{p}(\mathbf{x})$ into a Boolean function $\hat{h}(\mathbf{x})$ such that $\text{err}_D(\hat{h}) \leq 1/2 - c \exp(-O(\ell \log \ell))$. $\square$

With Lemma 7.10, we can immediately obtain the proof of Theorem 7.9.

Proof of Theorem 7.9. Notice that for every invertible affine transformation $L : \mathbb{R}^d \rightarrow \mathbb{R}^d$, the class $\mathcal{H}_L := \{h(L\mathbf{x}) \mid h \in \mathcal{H}\}$ is also a family of K -MIMs that admits degree- ℓ L_1 sandwiching polynomials. This implies to show the class $\mathcal{H}$ is $(\ell, \exp(-O(K\ell \log \ell)), c \exp(-O(\ell \log \ell)), C)$ -weakly learnable,

we only need to show that given distribution D over $\mathbb{R}^d \times \{\pm 1\}$, such that $D_{\mathbf{x}}$ is close to some $1/2$ -isotropic logconcave distribution $D'_{\mathbf{x}}$, and $\text{opt}_D(\mathcal{H})$ is small, we can produce a weak hypothesis.

For every $h \in \mathcal{H}$, such that $h(\mathbf{x}) = g(\mathbf{W}\mathbf{x})$, with $g \in \mathcal{G}$, let $\mathbf{x}' = \mathbf{W}\mathbf{x}$. Then, we have $h(\mathbf{x}) = g(\mathbf{x}')$. By definition of ℓ , for every logconcave distribution $D'_{\mathbf{x}}$ over $\mathbb{R}^k$, and for every $a \in (0, 1)$, there is a pair of polynomial p^u and p^l with degree $\ell(a)$ such that

$$p^l(\mathbf{x}') \leq g(\mathbf{x}') \leq p^u(\mathbf{x}'), \quad \mathbf{E}_{\mathbf{x} \sim D'_{\mathbf{x}}} [|p^u(\mathbf{x}') - p^l(\mathbf{x}')|] \leq a.$$

For every $c \in (0, 1)$, by choosing $C = 2(1 + c)$, we have $C/2 = (1 + c)$. By Lemma 7.10, there is an algorithm $\mathcal{A}$ such that given a distribution D over $\mathbb{R}^d \times \{\pm 1\}$, with $\text{opt}_D(\mathcal{H}) < 1/C = (1 - c/(1 + c))/2$, it draws $\text{poly}((\ell d)^\ell / c^2)$ examples from D , runs in $\text{poly}((\ell d)^\ell / c^2)$ time and outputs a hypothesis $\hat{h}$ with error $\text{err}_D(\hat{h}) < 1/2 - c \exp(-O(\ell \log \ell))$ with probability at least 99%, whenever there is some $1/2$ -logconcave distribution $D'_{\mathbf{x}}$ over $\mathbb{R}^d$ that satisfies $|\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(\mathbf{w} \cdot \mathbf{x})^i] - \mathbf{E}_{\mathbf{x} \sim D'_{\mathbf{x}}}[(\mathbf{w} \cdot \mathbf{x})^i]| < \exp(-O(K\ell \log \ell))$. Here $\ell = \ell(c/(3(1 + c))) \leq \ell(c/6)$.

This means the class $\mathcal{H}$ is $(\ell, \exp(-O(K\ell \log \ell)), c \exp(-O(\ell \log \ell)), C)$ -weakly learnable. By Theorem 2.8, There is an algorithm such that for every distribution D over $\mathbb{R}^d \times \{0, \pm 1\}$, such that $D_{\mathbf{x}}$ is a logconcave distribution, and $W(D) \geq \alpha$, and for every $\epsilon, c \in (0, 1)$, it draws $(d\ell)^{O(\ell)} \text{poly}_{K,c}(1/(\alpha\epsilon))$ samples, runs in $(d\ell)^{O(\ell)} 2^{\text{poly}_{K,c}(\log(1/(\alpha\epsilon)))}$ time, where $\ell = \ell(c/6)$, and outputs a hypothesis $\hat{h}$ such that with probability at least 99%, $\text{err}_{D_f}(\hat{h}) \leq (1 + c)\text{opt}_{D_f}(\mathcal{H}) + \epsilon$. $\square$

7.4 Quasi-PTAS for Agnostic Partial Learning of Multi-index Models

As an important application of Theorem 7.9, we obtain a quasi-polynomial time approximation scheme (QPTAS) for a broad class of MIMs. The main tool we use is the polynomial approximation theory developed in [KKM13], which gives low-degree L_1 sandwiching polynomials for a broader class of MIMs. We state this theorem as follows.

Theorem 7.11 (Extension of Theorem 6 in [KKM13]). *Let $D_{\mathbf{x}}$ be an isotropic logconcave distribution over $\mathbb{R}^d$ and let $f : \mathbb{R}^d \rightarrow \{0, 1\}$. For every $\epsilon, \delta > 0$, there is a pair of polynomials p^l, p^u of degree $\ell = \exp(O(d^4/\delta^4 + d^2 \log^2(d/\epsilon)))$, such that*

$$p^l(\mathbf{x}) \leq f(\mathbf{x}) \leq p^u(\mathbf{x}), \quad \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [|p^u(\mathbf{x}) - p^l(\mathbf{x})|] \leq \epsilon + 2 \mathbf{Pr}_{\mathbf{x} \sim D_{\mathbf{x}}} [S_\delta],$$

where $S_\delta = \{\mathbf{u} \in \mathbb{R}^d \mid \exists \mathbf{v} \in \mathbb{R}^d : \|\mathbf{u} - \mathbf{v}\|_2 \leq \delta, f(\mathbf{u}) \neq f(\mathbf{v})\}$.

We remark that the original statement in [KKM13], only claims a slightly weaker guarantee—namely, that there is a degree- ℓ polynomial p such that $\|p - f\|_{D_{\mathbf{x}}, 1}$ is bounded. However, obtaining L_1 -sandwiching polynomials with qualitatively the same degree is implicitly implied from the proof. We refer to Appendix C for the details.

Given Theorem 7.11, the QPTAS can be obtained immediately as a corollary of Theorem 7.9. For example, by Theorem 7.7, the class of functions of K halfspaces,

$$\mathcal{H}_{K,d} = \left\{ h(\mathbf{x}) = g(h_1(\mathbf{x}), \dots, h_K(\mathbf{x})) \mid g : \{\pm 1\}^K \rightarrow \{\pm 1\}, h_i(\mathbf{x}) = \text{sign}(w^{(i)} \cdot \mathbf{x} + t_i), i \in [K] \right\}$$

admits $\exp(O(K^4/c^4))$ -degree L_1 -sandwiching polynomials, which immediately implies Theorem 7.12.

Theorem 7.12 (QPTAS for Agnostically Learning Functions of Halfspaces). *There is an algorithm that for any distribution D over $\mathbb{R}^d \times \{0, \pm 1\}$ with $D_{\mathbf{x}}$ logconcave and $W(D) \geq \alpha$ has the following performance guarantee. Given $K \in \mathbb{Z}_+$ and $\epsilon, c \in (0, 1)$, the algorithm draws $(d\ell)^{O(\ell)} \text{poly}_{K,c}(1/(\alpha\epsilon))$ samples from D , it runs in $(d\ell)^{O(\ell)} 2^{\text{poly}_{K,c}(\log(1/(\alpha\epsilon)))}$ time, where $\ell = \exp(O(K/c^4))$, and outputs a hypothesis $\hat{h}$ such that with probability at least 99%, $\text{err}_{D_f}(\hat{h}) \leq (1 + c)\text{opt}_{D_f}(\mathcal{H}_K) + \epsilon$.*

A similar bound can also be obtained for the class of low-dimensional and low-degree Polynomial Threshold Functions (PTFs), defined as

$$P_{K,m} = \left\{ h(\mathbf{x}) = \text{sign}(p(\mathbf{W}\mathbf{x})) \mid \mathbf{W} \in \mathbb{R}^{K \times d}, \mathbf{W}\mathbf{W}^\top = \mathbf{I}_{K \times K}, p : \mathbb{R}^K \rightarrow \mathbb{R}, \deg(p) \leq m \right\}.$$

Theorem 7.13 (QPTAS for Agnostically Learning Low-Dimensional PTFs). *There is an algorithm that for any distribution D over $\mathbb{R}^d \times \{0, \pm 1\}$ with $D_{\mathbf{x}}$ logconcave and $W(D) \geq \alpha$ has the following performance guarantee. Given $K, m \in \mathbb{Z}_+$ and $\epsilon, c \in (0, 1)$, the algorithm draws $(d\ell)^{O(\ell)} \text{poly}_{K,m,c}(1/(\alpha\epsilon))$ samples from D , it runs in $(d\ell)^{O(\ell)} 2^{\text{poly}_{K,m,c}(\log(1/(\alpha\epsilon)))}$ time, where $\ell = \exp(O(m2^{\tilde{O}(K)} \log^m(1/c)/c^4))$, and outputs a hypothesis $\hat{h}$ such that with probability at least 99%, $\text{err}_{D_f}(\hat{h}) \leq (1+c)\text{opt}_{D_f}(P_{K,m}) + \epsilon$.*

Theorem 7.13 can be seen as a direct corollary of Theorem 7.11 and the result by [Lot15], which gives an explicit bound on the volume of examples in $\mathbb{R}^d$ that are δ -close to the zero set of a low-degree polynomial. We refer to Appendix C for details.

7.5 Agnostically Learning Partial Linear Classifiers

In this section, we study the problem of agnostic partial learning halfspaces under logconcave distributions. Let $\mathcal{H} = \{h_{\mathbf{w},t}(\mathbf{x}) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + t) \mid \mathbf{w} \in \mathbb{S}^{d-1}, t \in \mathbb{R}\}$ denote the class of halfspaces on $\mathbb{R}^d$. By Theorem 7.12, $\mathcal{H}$ already admits an agnostic partial learner that achieves $(1+c)\text{opt} + \epsilon$ error, for any desired constant $c > 0$, with complexity $(\ell d)^\ell 2^{\text{poly}_c(\log(1/\epsilon))}$, with $\ell = \exp(O(1/c^4))$. The generic approach establishing this bound leads to a fairly large degree polynomial degree in the sample and computational complexities, as it requires sufficiently accurate sandwiching polynomials.

In this section, we give agnostic partial learners for halfspaces that achieve error $\tilde{O}(\text{opt}_{D_f}(\mathcal{H})) + \epsilon$ and $O(\text{opt}_{D_f}(\mathcal{H})) + \epsilon$ with quantitatively much better complexities. Formally, we establish the following result.

Theorem 7.14. *There is an algorithm such that for every distribution D over $\mathbb{R}^d \times \{0, \pm 1\}$ such that $D_{\mathbf{x}}$ is logconcave and $W(D) \geq \alpha$, for every $\epsilon \in (0, 1)$, it draws $O(d^4) \text{poly}(1/(\alpha\epsilon))$ samples, runs in $\text{poly}(d) 2^{\text{poly}(\log(1/(\alpha\epsilon)))}$ time and outputs a hypothesis $\hat{h}$ such that with probability at least 99%, $\text{err}_{D_f}(\hat{h}) \leq O(\text{opt}_{D_f}(\mathcal{H})) + \epsilon$.*

An implication of Theorem 7.14 is that to achieve $O(\text{opt}) + \epsilon$ error for halfspace learning, we can avoid using the framework of polynomial approximation, which requires the algorithm to compute higher moment tensor of the marginal distribution, but only make use of information from the first four moments of the marginal distribution. Although this observation gives us reasonably good sample complexity, that is polynomial with respect to all input parameters, since we need to invoke Theorem 5.1 for our learner, this prevents us from getting a fully-polynomial time algorithm.

Our second main result in this section shows that by making use of the first two moments of the marginal distribution, we can learn halfspaces within error $\log^4(1/\alpha) \tilde{O}(\text{opt}_{D_f}(\mathcal{H})) + \epsilon$ in fully-polynomial time.

Theorem 7.15. *There is an algorithm such that for every distribution D over $\mathbb{R}^d \times \{0, \pm 1\}$ such that $D_{\mathbf{x}}$ logconcave and $W(D) \geq \alpha$, for every $\epsilon \in (0, 1)$, it draws $O(d^2) \text{poly}(1/(\alpha\epsilon))$ samples, runs in $\text{poly}(d, 1/(\alpha\epsilon))$ time and outputs a hypothesis $\hat{h}$ such that with probability at least 99%, $\text{err}_{D_f}(\hat{h}) \leq O(\log^4(1/(\alpha\epsilon)) \text{opt}_{D_f}(\mathcal{H})) + \epsilon$.*

Remark 7.16. We remark that if $\text{opt}_{D_f}(\mathcal{H}) \leq \epsilon$, $\text{err}_{D_f}(\hat{h}) \leq \log^4(1/\alpha) \tilde{O}(\text{opt}_{D_f}(\mathcal{H})) + \epsilon$. On the other hand, if $\text{opt}_{D_f}(\mathcal{H}) > \epsilon$, we can choose to run the algorithm by with $\epsilon' = \Theta(\text{opt}_{D_f}(\mathcal{H}))$, which gives $\text{err}_{D_f}(\hat{h}) \leq \log^4(1/\alpha) \tilde{O}(\text{opt}_{D_f}(\mathcal{H})) + \epsilon$.

To prove the above two theorems, we will prove the following structural result (Lemma 7.17). Roughly speaking, if the first four moments of the marginal distribution $D_{\mathbf{x}}$ are close to those of some logconcave distribution, then for every halfspace, at least a constant fraction of the examples are constant far away from its decision boundary. On the other hand, if $D_{\mathbf{x}}$ only has its first two moments close to some logconcave distribution $D'_{\mathbf{x}}$, but it can be seen as a conditional distribution of $D'_{\mathbf{x}}$ over some event with mass α , then at least $\Omega(\log^{-4}(1/\alpha))$ fraction of the examples are far from its decision boundary.

Lemma 7.17. *Let $D_{\mathbf{x}}$ be a distribution over $\mathbb{R}^d$. Then the following holds.*

1. *Suppose that there is some $1/2$ -isotropic logconcave distribution $D'_{\mathbf{x}}$, such that for a unit vector $\mathbf{w} \in \mathbb{S}^{d-1}$ and $i \in [4]$, $|\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(\mathbf{w} \cdot \mathbf{x})^i] - \mathbf{E}_{\mathbf{x} \sim D'_{\mathbf{x}}}[(\mathbf{w} \cdot \mathbf{x})^i]| < 1/64$, then for every halfspace of the form $h_{(\mathbf{w}, t)}(\mathbf{x}) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + t)$, $\Pr_{\mathbf{x} \sim D_{\mathbf{x}}}[\mathbf{w} \cdot \mathbf{x} + t > 1/4] > 1/C$, for some universal constant $C > 1$.*
2. *Suppose that there is some $1/2$ -isotropic logconcave distribution $D'_{\mathbf{x}}$, such that for a unit vector $\mathbf{w} \in \mathbb{S}^{d-1}$, for $i \in [2]$, $|\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(\mathbf{w} \cdot \mathbf{x})^i] - \mathbf{E}_{\mathbf{x} \sim D'_{\mathbf{x}}}[(\mathbf{w} \cdot \mathbf{x})^i]| < 1/64$, and $D_{\mathbf{x}}$ is a conditional distribution of $D'_{\mathbf{x}}$ over some event A such that $\Pr_{\mathbf{x} \sim D'_{\mathbf{x}}}[A] = \alpha$, then for every halfspace of the form $h_{(\mathbf{w}, t)}(\mathbf{x}) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + t)$, $\Pr_{\mathbf{x} \sim D_{\mathbf{x}}}[\mathbf{w} \cdot \mathbf{x} + t > 1/4] > \Omega(\log^{-4}(1/\alpha))$.*

Proof of Lemma 7.17. Fix a halfspace $h_{(\mathbf{w}, t)}(\mathbf{x}) = \text{sign}(\mathbf{w} \cdot \mathbf{x} + t)$. Since $D'_{\mathbf{x}}$ is a mean-zero, $1/2$ isotropic logconcave distribution, by Fact A.9, we have $\mathbf{E}_{\mathbf{x} \sim D'_{\mathbf{x}}}[(\mathbf{w} \cdot \mathbf{x})^2] \leq 5/2$ and $\mathbf{E}_{\mathbf{x} \sim D'_{\mathbf{x}}}[(\mathbf{w} \cdot \mathbf{x})^4] \leq c$ for some constant $c > 1$. Let $\Delta = 1/64$. We consider two cases for the threshold t .

In the first case, we assume $|t| > 4$. By Chebyshev's inequality and the moment assumption on $D_{\mathbf{x}}$, we have $\Pr_{\mathbf{x} \sim D_{\mathbf{x}}}[\mathbf{w} \cdot \mathbf{x} > 3] \leq (2.5 + \Delta)/9 \leq 1/3$. This implies that with probability at least $2/3$, we have $|\mathbf{w} \cdot \mathbf{x}| < 3$. Since $|t| > 4$, if $|\mathbf{w} \cdot \mathbf{x}| < 3$, it must be the case, where $|\mathbf{w} \cdot \mathbf{x} + t| > 1$. This implies that with probability at least $2/3$, $|\mathbf{w} \cdot \mathbf{x} + t| > 1$.

In the second case, we assume $|t| < 4$. By Paley–Zygmund's inequality, we have for every $\theta \in (0, 1)$,

$$\Pr_{\mathbf{x} \sim D_{\mathbf{x}}} \left[|\mathbf{w} \cdot \mathbf{x} + t| > \sqrt{\theta \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(\mathbf{w} \cdot \mathbf{x} + t)^2]} \right] \geq (1 - \theta)^2 \frac{(\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(\mathbf{w} \cdot \mathbf{x} + t)^2])^2}{\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(\mathbf{w} \cdot \mathbf{x} + t)^4]}. \quad (6)$$

We will next lower bound $\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(\mathbf{w} \cdot \mathbf{x} + t)^2]$ and upper bound $\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(\mathbf{w} \cdot \mathbf{x} + t)^4]$ through the moment information of $\mathbf{w} \cdot \mathbf{x}$. Denote by $\mu_{\mathbf{w}} := \mathbf{E}_{\mathbf{x} \sim D'_{\mathbf{x}}}[\mathbf{w} \cdot \mathbf{x}]$. Since $|t| < 4$, we have

$$\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(\mathbf{w} \cdot \mathbf{x} + t)^2] = \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(\mathbf{w} \cdot \mathbf{x} - \mu_{\mathbf{w}})^2 + (t + \mu_{\mathbf{w}})^2 + 2(t + \mu_{\mathbf{w}})(\mathbf{w} \cdot \mathbf{x} - \mu_{\mathbf{w}})] \geq 1/8.$$

Since $\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(\mathbf{w} \cdot \mathbf{x})^2] \in [1/2 - \Delta, (5/2) + \Delta]$, by Holder's inequality, we have

$$\left| \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(\mathbf{w} \cdot \mathbf{x})^3] \right| \leq \sqrt{\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(\mathbf{w} \cdot \mathbf{x})^2] \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(\mathbf{w} \cdot \mathbf{x})^4]} \leq \sqrt{((5/2) + \Delta) \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(\mathbf{w} \cdot \mathbf{x})^4]}.$$

If $|\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(\mathbf{w} \cdot \mathbf{x})^4] - \mathbf{E}_{\mathbf{x} \sim D'_{\mathbf{x}}}[(\mathbf{w} \cdot \mathbf{x})^4]| < 1/64$, then we have

$$\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(\mathbf{w} \cdot \mathbf{x} + t)^4] = \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(\mathbf{w} \cdot \mathbf{x})^4 + t^4 + 6(\mathbf{w} \cdot \mathbf{x})^2 t^2 + 4t(\mathbf{w} \cdot \mathbf{x})^3 + 4t^3(\mathbf{w} \cdot \mathbf{x})] \leq C',$$

for some constant $C' > 1$. Combine with (6), with $\theta = 1/2$, we have $\Pr_{\mathbf{x} \sim D_{\mathbf{x}}}[\mathbf{w} \cdot \mathbf{x} + t > 1/4] \geq 1/C$, for some $C > 1$. This proves item 1.

To prove [item 2](#), by [Lemma 5.7](#), we know that there is a degree-4 polynomial $Q(\mathbf{x})$, with $\mathbf{E}_{\mathbf{x} \sim D'_x}[Q^2(\mathbf{x})] \leq O(\log^8(1/\alpha))$ such that

$$\mathbf{E}_{\mathbf{x} \sim D_x}[(\mathbf{w} \cdot \mathbf{x})^4] = \mathbf{E}_{\mathbf{x} \sim D'_x}[Q(\mathbf{x})(\mathbf{w} \cdot \mathbf{x})^4] \leq \sqrt{\mathbf{E}_{\mathbf{x} \sim D'_x}[Q^2(\mathbf{x})] \mathbf{E}_{\mathbf{x} \sim D'_x}[(\mathbf{w} \cdot \mathbf{x})^8]} \leq O(\log^4(1/\alpha)),$$

which implies

$$\mathbf{E}_{\mathbf{x} \sim D_x}[(\mathbf{w} \cdot \mathbf{x} + t)^4] = \mathbf{E}_{\mathbf{x} \sim D_x}[(\mathbf{w} \cdot \mathbf{x})^4 + t^4 + 6(\mathbf{w} \cdot \mathbf{x})^2 t^2 + 4t(\mathbf{w} \cdot \mathbf{x})^3 + 4t^3(\mathbf{w} \cdot \mathbf{x})] \leq O(\log^4(1/\alpha)).$$

Combine with [\(6\)](#), with $\theta = 1/2$, we have $\Pr_{\mathbf{x} \sim D_x}[|\mathbf{w} \cdot \mathbf{x} + t| > 1/4] \geq \Omega(\log^{-4}(1/\alpha))$. $\square$

With [Lemma 7.1](#) and [Lemma 7.17](#), we can immediately obtain weak learning algorithms for learning halfspaces. We summarize our weak learning algorithm as [Algorithm 6](#).

Algorithm 6 HALFSPACEWEAKLEARNING(Weak learning halfspaces under structured distribution)

- 1: **Input:** Sample access to D , diameter parameter $B > 0$, advantage parameter β .
 - 2: **Output:** $\hat{h}$: weak hypothesis over D .
 - 3: $h_+(\mathbf{x}) := 1, h_-(\mathbf{x}) := -1, \mathcal{C} := \{h_+, h_-\}$
 - 4: Draw $S \sim D^m$ a set of $m = \tilde{O}(d/\beta^2)$ i.i.d. examples from D
 - 5: Find $\mathbf{w} \in \mathbb{B}^{d-1}, t \in [-B, B]$ that maximize $\sum_{(\mathbf{x}, y) \in S} (y(\mathbf{w} \cdot \mathbf{x} + t))$ through linear regression and define $F(\mathbf{x}) := \mathbf{w} \cdot \mathbf{x} + t$.
 - 6: $g(\mathbf{x}) = \text{proj}_{[-1, 1]}(F(\mathbf{x})/(16B^2)) : \mathbb{R}^d \rightarrow [-1, 1]$
 - 7: **for** $i \in [O(B^2/\beta^2)]$ **do**
 - 8: Sample t' uniformly from $[-1, 1]$ and define $h_i(\mathbf{x}) := \text{sign}(g(\mathbf{x}) - t')$
 - 9: $\mathcal{C} \leftarrow \mathcal{C} \cup \{h_i(\mathbf{x})\}$
 - 10: Draw $S' \sim D^m$ a set of $m = \tilde{O}(d/\beta^2)$ i.i.d. examples from D
 - 11: **return** $\hat{h} := \text{argmin}_{h \in \mathcal{C}} \sum_{(\mathbf{x}, y) \in S'} \mathbb{1}\{h(\mathbf{x}) \neq y\}$.
-

We first show that there is a polynomial time $(4, 1/64, \Omega(1), O(1))$ -weak learner for the class of halfspaces, which immediately implies [Theorem 7.14](#) using [Theorem 2.8](#).

Lemma 7.18. *There is an algorithm $\mathcal{A}$ that draws $O(d)$ examples, runs in $\text{poly}(d)$ time and $(4, 1/64, \Omega(1), O(1))$ -weakly learns the class of halfspaces with probability 99%.*

Proof of Lemma 7.18. Since the class of halfspaces is closed under affine transformations, we show that by setting the input parameter $B = O(1)$, [Algorithm 6](#) outputs a weak hypothesis with $\Omega(1)$ advantage in $\text{poly}(d)$ time.

Let D be the input distribution over $\mathbb{R}^d \times \{\pm 1\}$. Let D'_x be a $1/2$ -isotropic logconcave distribution, such that for any unit vector $\mathbf{w} \in \mathbb{S}^{d-1}$ and, $|\mathbf{E}_{\mathbf{x} \sim D_x}[(\mathbf{w} \cdot \mathbf{x})^i] - \mathbf{E}_{\mathbf{x} \sim D'_x}[(\mathbf{w} \cdot \mathbf{x})^i]| < 1/64, i \in [4]$. Let $h_{\mathbf{w}^*, t^*} = \text{sign}(\mathbf{w}^* \cdot \mathbf{x} + t^*)$ be the halfspace such that $\text{err}_D(h_{\mathbf{w}^*, t^*}) = \text{opt} < 1/C$, for some constant $C > 1$. We first show that if $|t^*| > 4$, then a constant hypothesis $h_+(\mathbf{x}) = 1$ or $h_-(\mathbf{x}) = -1$ must have a constant advantage.

Without loss of generality, we assume that $t^* > 4$. By assumption and Chebyshev's inequality, we have

$$\Pr_{\mathbf{x} \sim D_x}[h_{\mathbf{w}^*, t^*}(\mathbf{x}) = -1] = \Pr_{\mathbf{x} \sim D_x}[\mathbf{w}^* \cdot \mathbf{x} < -4] \leq 1/4.$$

Since $\text{opt} < 1/C$, we know that $\Pr_{(\mathbf{x}, y) \sim D}[y = -1] \leq \Pr_{\mathbf{x} \sim D_x}[h_{\mathbf{w}^*, t^*}(\mathbf{x}) = -1] + \text{opt} < 2/5$. This implies that h_+ has $1/10$ advantage over D . So, in the rest of the proof, we will assume $|t^*| < 4$.

By Lemma 7.17, we know that $\Pr_{\mathbf{x} \sim D_{\mathbf{x}}}[\|\mathbf{w}^* \cdot \mathbf{x} + t^*\| > 1/4] > 1/C'$, for some universal constant C' . Let $F^*(\mathbf{x}) = \mathbf{w}^* \cdot \mathbf{x} + t^*$, then $\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[h_{w^*, t^*}(\mathbf{x})F^*(\mathbf{x})] > 1/(4C')$. By the assumption on the moment information of $D_{\mathbf{x}}$, and $|t^*| < 4$, we know that $\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(F^*(\mathbf{x}))^2] < O(1)$. By Lemma 7.1, we know that when $\text{opt} < 1/C$, for some constant $C > 1$, we have $\mathbf{E}_{(\mathbf{x}, y) \sim D}[yF^*(\mathbf{x})] > 2\beta$, for some $\beta > 1/C$. Thus, using standard sample complexity analysis of linear regression (see, e.g., [BM02],) by running a linear regression over $m = \tilde{O}(d/\beta^2)$ examples drawn from D , we are able to find an affine function $F(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + t$, $\mathbf{w} \in \mathbb{B}^{d-1}$, $|t| \leq 4$, such that $\mathbf{E}_{(\mathbf{x}, y) \sim D}[yF(\mathbf{x})] \geq \mathbf{E}_{(\mathbf{x}, y) \sim D}[yF^*(\mathbf{x})] - \beta \geq \beta$. By Lemma 7.1 and Fact 7.2, we know that using $F(\mathbf{x})$ and a random threshold t' sampled uniformly from $[-1, 1]$, we can find a boolean function $h_i(\mathbf{x})$ that has $\Omega(1)$ advantage with respect to D , with probability $\Omega(1)$.

Thus, by randomly sampling t' , $\Omega(1)$ times, Algorithm 6 finds such a weak hypothesis with high probability. Thus, by running a standard hypothesis testing approach, Algorithm 6 outputs a weak hypothesis as the statement. $\square$

Combining Lemma 7.18 and Theorem 2.8, we immediately get a quasi-polynomial time algorithm that agnostically learns a partial halfspace to $O(\text{opt}) + \epsilon$ error under any logconcave distribution and prove Theorem 7.14.

We remark that the quasi-polynomial time of the algorithm is due to the use of a quasi-polynomial time moment discrepancy finder Algorithm 4 for the 4th moment. If we have a faster algorithm for finding moment discrepancies, then the time complexity of the learning algorithm in Theorem 7.14 can be improved. On the other hand, since we can always efficiently find the moment discrepancies for the first two moments through mean estimation and singular value decomposition, we will give a polynomial time algorithm for agnostic partial learning halfspaces with $\tilde{O}(\text{opt}) + \epsilon$ error.

To begin with, we give the weak learning algorithm for agnostic learning halfspaces under assumptions on their first two moments.

Lemma 7.19. *Let D be a distribution over $\mathbb{R}^d \times \{\pm 1\}$ such that there is a $1/2$ -isotropic log-concave distribution $D'_{\mathbf{x}}$, such that $|\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(\mathbf{w} \cdot \mathbf{x})^i] - \mathbf{E}_{\mathbf{x} \sim D'_{\mathbf{x}}}[(\mathbf{w} \cdot \mathbf{x})^i]| < 1/64$, for $i \in [2]$ and $D_{\mathbf{x}}$ is a conditional distribution of $D'_{\mathbf{x}}$ over some event A such that $\Pr_{\mathbf{x} \sim D'_{\mathbf{x}}}[A] = \alpha$. Let $\text{opt} = \inf_{h \in \mathcal{H}} \text{err}_D(h)$, where $\mathcal{H}$ is the class of halfspaces. Suppose $\text{opt} < \log^{-4}(1/\alpha)$, then Algorithm 6 draws $\text{poly}(d, \log(1/\alpha))$ examples from D , runs in $\text{poly}(d, \log(1/\alpha))$ time and outputs a hypothesis $\hat{h}$ such that $\text{err}_D(\hat{h}) < 1/2 - \text{polylog}^{-1}(1/\alpha)$ with probability at least 99%.*

Proof of Lemma 7.19. Let $h_{\mathbf{w}^*, t^*}$ be the optimal halfspace under D . We first show that if $|t^*| > 4$, then a constant hypothesis $h_+(\mathbf{x}) = 1$ or $h_-(\mathbf{x}) = -1$ must have a constant advantage.

Without loss of generality, we assume that $t^* > 4$. By assumption and Chebyshev's inequality, we have

$$\Pr_{\mathbf{x} \sim D_{\mathbf{x}}}[h_{\mathbf{w}^*, t^*}(\mathbf{x}) = -1] = \Pr_{\mathbf{x} \sim D_{\mathbf{x}}}[\mathbf{w}^* \cdot \mathbf{x} < -4] \leq 1/4.$$

Since $\text{opt} < \log^{-4}(1/\alpha)$, we know that $\Pr_{(\mathbf{x}, y) \sim D}[y = -1] \leq \Pr_{\mathbf{x} \sim D_{\mathbf{x}}}[h_{\mathbf{w}^*, t^*}(\mathbf{x}) = -1] + \text{opt} < 2/5$. This implies that h_+ has $1/10$ advantage over D . So, in the rest of the proof, we will assume $|t^*| < 4$.

By Lemma 7.17, we know that $\Pr_{\mathbf{x} \sim D_{\mathbf{x}}}[\|\mathbf{w}^* \cdot \mathbf{x} + t^*\| > 1/4] > 1/\Omega(\log^4(1/\alpha))$. Let $F^*(\mathbf{x}) = \mathbf{w}^* \cdot \mathbf{x} + t^*$, then $\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[h_{w^*, t^*}(\mathbf{x})F^*(\mathbf{x})] > 1/\Omega(\log^4(1/\alpha))$. By the assumption on the moment information of $D_{\mathbf{x}}$, and $|t^*| < 4$, we know that $\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}}[(F^*(\mathbf{x}))^2] < O(1)$. By Lemma 7.1, we know that when $\text{opt} < 1/\Omega(\log^4(1/\alpha))$, we have $\mathbf{E}_{(\mathbf{x}, y) \sim D}[yF^*(\mathbf{x})] > 2\beta$, for some $\beta > 1/\Omega(\log^4(1/\alpha))$. Thus, using standard sample complexity analysis of linear regression (see, e.g., [BM02],) by running a linear regression over $m = \tilde{O}(d/\beta^2)$ examples drawn from D , we are able to find an affine function $F(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + t$, $\mathbf{w} \in \mathbb{B}^{d-1}$, $|t| \leq 4$, such that $\mathbf{E}_{(\mathbf{x}, y) \sim D}[yF(\mathbf{x})] \geq \mathbf{E}_{(\mathbf{x}, y) \sim D}[yF^*(\mathbf{x})] - \beta \geq \beta$. By

Lemma 7.1 and Fact 7.2, we know that using $F(\mathbf{x})$ and a random sampled threshold t' , we can find a boolean function $h_i(\mathbf{x})$ that has $1/\Omega(\log^4(1/\alpha))$ advantage with respect to D , with probability $1/\Omega(\log^4(1/\alpha))$.

Thus, by randomly sampling t' , $\Omega(\log^4(1/\alpha))$ times, we find such a weak hypothesis with high probability. Thus, by running a standard hypothesis testing approach, Algorithm 6 outputs a weak hypothesis as the statement. $\square$

Proof of Theorem 7.15. We will use Algorithm 6 as the weak learning algorithm $\mathcal{A}_w$ and singular-value decomposition as a moment discrepancy finding algorithm $\mathcal{A}_m$ used in Algorithm 1. We notice that given any $1/2$ -isotropic logconcave distribution $D_{\mathbf{x}}$ with weight $\alpha \in (0, 1)$, $\mathcal{A}_m$ is a $(2, 1/64, 1/2)$ discrepancy finder for the distribution $D_{\mathbf{x}}$ and $(D_f)_{\mathbf{x}}$ with sample complexity $\text{poly}(d/\alpha)$ and time complexity $\text{poly}(d/\alpha)$.

For every $\alpha \in (0, 1)$ and $\epsilon \in (0, 1)$, let D be any distribution over $\mathbb{R}^d \times \{\pm 1\}$, with $D_{\mathbf{x}}$ a logconcave distribution, $W(D) \geq \alpha\epsilon$ and $\text{opt}_{D_f}(\mathcal{H}) < 1/\Omega(\log^4(\alpha\epsilon))$. Denote by $\mathcal{U}$, the partition generated by Algorithm 1, after it terminates. By Lemma 4.5, we know that there is a constant $a > 0$, such that the choice of $\xi = (\alpha\epsilon)^a$ satisfies the following property.

Let $\mathcal{B} = \{R \in \mathcal{U} \mid \Pr_{(\mathbf{x}, y) \sim D_f}[\mathbf{x} \in R] \leq \xi\}$. Then, $\sum_{R \in \mathcal{B}} \Pr_{(\mathbf{x}, y) \sim D_f}[\mathbf{x} \in R] \leq 1/10$. We denote by $\mathcal{G} = \mathcal{U} \setminus \mathcal{B}$ and denote by $\mathcal{M} \subseteq \mathcal{G}$ the set of R , where a random binary function h_R is assigned. By definition, for each $R \in \mathcal{G}$, we have $\Pr_{(\mathbf{x}, y) \sim D_f}[\mathbf{x} \in R] > \xi$. This implies that the weight of D_R is at least $(\alpha\epsilon)^{1+a}$.

We show that $\sum_{R \in \mathcal{M}} \Pr_{(\mathbf{x}, y) \sim D_f}[\mathbf{x} \in R]$ is small. By Lemma 7.19, for each $R \in \mathcal{M}$, we have $\Pr_{(\mathbf{x}, y) \sim D_f}[h^*(\mathbf{x}) \neq y \mid \mathbf{x} \in R] > \Omega(\log^{-4}(1/(\alpha\epsilon)))$, where h^* is the hypothesis in H that achieves opt_{D_f} . Now suppose that $\sum_{R \in \mathcal{M}} \Pr_{(\mathbf{x}, y) \sim D_f}[\mathbf{x} \in R] > 1/2$, then we have

$$\text{err}_{D_f}(h^*) \geq \sum_{R \in \mathcal{M}} \Pr_{(\mathbf{x}, y) \sim D_f}[\mathbf{x} \in R] \Pr_{(\mathbf{x}, y) \sim D_f}[h^*(\mathbf{x}) \neq y \mid \mathbf{x} \in R] > \Omega(\log^{-4}(1/(\alpha\epsilon))),$$

which gives a contradiction to the fact that $\text{opt}_{D_f} < O(\log^{-4}(1/(\alpha\epsilon)))$. Thus, we have

$$\sum_{R \in \mathcal{G} \setminus \mathcal{M}} \Pr_{(\mathbf{x}, y) \sim D_f}[\mathbf{x} \in R] > 2/5.$$

Since for each $R \in \mathcal{G} \setminus \mathcal{M}$, we have $\Pr_{(\mathbf{x}, y) \sim D_f}[h_R(\mathbf{x}) \neq y \mid R] < 1/2 - \log^{-4}(1/(\alpha\epsilon))$, we know that $\text{err}_{D_f}(\hat{h}) \leq 1/2 - \log^{-4}(1/(\alpha\epsilon))$.

As a summary, for every $\alpha \in (0, 1)$ and $\epsilon \in (0, 1)$, let D be any distribution over $\mathbb{R}^d \times \{\pm 1\}$, with $D_{\mathbf{x}}$ a logconcave distribution, $W(D) \geq \alpha\epsilon$, and $\text{opt}_{D_f}(\mathcal{H}) < 1/\Omega(\log^4(\alpha\epsilon))$, Algorithm 1 runs in $\text{poly}(d/(\alpha\epsilon))$ time and output a hypothesis $\hat{h}$ with error $1/2 - \log^{-4}(1/(\alpha\epsilon))$.

By Theorem 3.2, we know that every distribution D over $\mathbb{R}^d \times \{0, \pm 1\}$, such that $D_{\mathbf{x}}$ is a logconcave distribution, and $\Pr_{(\mathbf{x}, y) \sim D}[y \neq 0] \geq \alpha$, for every $\epsilon \in (0, 1)$, we can call Algorithm 1 $\tilde{O}(1/\epsilon)$ times to get a hypothesis with error at most $O(\log^4(1/(\alpha\epsilon))\text{opt}_{D_f}(\mathcal{H}) + \epsilon)$. $\square$

7.6 Agnostic Multiclass Linear Classification via Partial Linear Classification

In this section, we consider the problem of agnostic partial learning of multiclass linear classifiers under logconcave distributions. In a multiclass classification problem, the label of an example is not from $\{\pm 1\}$, but from $[K]$ for some $K > 2$. A natural paradigm for solving a multiclass classification task is to reduce the problem to binary classification. Roughly speaking, for every pair of labels (i, j) , we want to learn a hypothesis h_{ij} to tell whether an example is more likely to have label i or

label j . By combining these binary classifiers, we obtain a reasonably robust multiclass classifier. We remark that even though $D_{\mathbf{x}}$ might be a well-behaved distribution, if we consider examples with label (i, j) , their marginal distribution can be significantly different from the underlying distribution on the entire space; e.g., a Gaussian or logconcave, under which efficient learning algorithms were previously known in the literature. Fortunately, our notion of agnostic *partial* learning provides a direct way to address this technical issue. As concrete applications, for multiclass linear classification, we give a fully-polynomial time algorithm that achieves error $\tilde{O}(K\text{opt}) + \epsilon$ over any logconcave distribution. Furthermore, we show that by allowing the runtime to become quasi-polynomial in K/ϵ , we can get an algorithm with error $O(K)\text{opt} + \epsilon$.

As an important application of agnostic partial linear classification, we give the first quasi-polynomial and polynomial time algorithm for agnostic multiclass linear classification under logconcave distribution. To begin with, we introduce the learning problem formally.

Definition 7.20 (Agnostic Partial Learning Multiclass Linear Classifiers). *Let $K \in \mathbb{Z}_+, K \geq 2$. Define $\mathcal{M}_{K,d} = \{h_W(\mathbf{x}) : \mathbb{R}^d \rightarrow [K] \mid h_W(\mathbf{x}) = \text{argmax}_{i \in [K]} (W_i \cdot (\mathbf{x}, 1)), W \in \mathbb{R}^{d+1 \times K}\}$ be the class of K -class linear classifier (K -MLC). Let D be a distribution over $\mathbb{R}^d \times [K] \cup \{0\}$ such that $D_{\mathbf{x}}$ is logconcave. Denote by D_f the conditional distribution of D over the event $\mathcal{E} := \{(\mathbf{x}, y) \mid y \neq 0\}$. Let $h : \mathbb{R}^d \rightarrow [K]$ be a multiclass classifier and define $\text{err}_{D_f}(h) = \Pr_{(\mathbf{x}, y) \sim D_f}[h(\mathbf{x}) \neq y]$. Denote by $\text{opt}_{D_f}(\mathcal{M}_{K,d}) = \inf_{h \in \mathcal{M}_{K,d}} \text{err}_{D_f}(h)$. Given a set of i.i.d. sample access to D , and parameter $\epsilon, \delta \in (0, 1)$, the goal of an algorithm is to output a hypothesis $\hat{h}$ such that with probability at least $1 - \delta$, $\text{err}_{D_f}(\hat{h}) \leq F(\text{opt}_{D_f}(\mathcal{M}_{K,d})) + \epsilon$ for some function $F : (0, 1) \rightarrow (0, 1)$.*

The main result of this section is the following theorem.

Theorem 7.21 (Agnostic Partial Learning K -MLC under Logconcave Distributions). *Consider the problem of agnostic partial learning multiclass linear classifiers under logconcave distributions.*

1. *There is an algorithm $\mathcal{A}$ such that for every input distribution D over $\mathbb{R}^d \times \{[K] \cup \{0\}\}$, with $W(D) \geq \alpha$ such that $D_{\mathbf{x}}$ is a logconcave distribution, it draws $\text{poly}(dK/(\alpha\epsilon))$ samples from D , runs in $\text{poly}(d)2^{\text{polylog}(K/(\alpha\epsilon))}$ time and outputs a hypothesis $\hat{h} : \mathbb{R}^d \rightarrow [K]$, such that with probability at least 99%, $\text{err}_{D_f}(\hat{h}) < O(K\text{opt}_{D_f}(\mathcal{M}_{K,d})) + \epsilon$, where $\mathcal{M}_{K,d}$ is the family of K -class linear classifiers.*
2. *There is an algorithm $\mathcal{A}$ such that for every input distribution D over $\mathbb{R}^d \times \{[K] \cup \{0\}\}$, with $W(D) \geq \alpha$ such that $D_{\mathbf{x}}$ is logconcave, it draws $\text{poly}(dK/(\alpha\epsilon))$ samples from D , runs in $\text{poly}(dK/(\alpha\epsilon))$ time and outputs a hypothesis $\hat{h} : \mathbb{R}^d \rightarrow [K]$, such that with probability at least 99%, $\text{err}_{D_f}(\hat{h}) < O(\log^4(K/(\alpha\epsilon)))(K\text{opt}_{D_f}(\mathcal{M}_{K,d})) + \epsilon$, where $\mathcal{M}_{K,d}$ is the class of K -class linear classifier.*

Our algorithm for learning multiclass linear classifiers is obtained via a simple reduction to the binary case. As already noted, our reduction crucially leverages our partial learning framework.

Lemma 7.22 (Reduction of MLC to Binary Linear Classification). *Suppose there is an algorithm $\mathcal{A}$ for the problem of agnostic partial learning the class of halfspaces $\mathcal{H}$, such that for every input distribution D over $\mathbb{R}^d \times \{\pm 1, 0\}$ and $\epsilon \in (0, 1)$, it draws $M(\epsilon)$ examples, runs in $T(\epsilon)$ time and outputs a hypothesis $\hat{h}_b : \mathbb{R}^d \rightarrow \{\pm 1\}$ with $\text{err}_D(\hat{h}_b) < C(\epsilon)(\text{opt}_D(\mathcal{H})) + \epsilon$, where $C(\epsilon) > 0$ is some constant that only depends on ϵ . Then there is an algorithm for the problem of agnostic partial learning of multiclass linear classifiers such that for every input distribution $D \times \{[K] \cup \{0\}\}$ and $\epsilon \in (0, 1)$, it draws $O(K^2 M(\epsilon))$ examples, runs in $O(K^2 T(\epsilon))$ time and outputs a hypothesis $\hat{h} : \mathbb{R}^d \rightarrow [K]$ such that $\text{err}_D(\hat{h}) \leq KC(\epsilon)(\text{opt}_D(\mathcal{M}_{K,d})) + K^2\epsilon$.*

Algorithm 7 MULTICLASSREDUCTION(Multiclass classification using binary classification)

- 1: **Input:** Sample access from D , accuracy parameter $\epsilon > 0$, halfspace learning algorithm $\mathcal{A}$.
 - 2: **Output:** $\hat{h} : \mathbb{R}^d \rightarrow [K]$ hypothesis over D .
 - 3: **for** $i, j \in [K], i \neq j$ **do**
 - 4: Define D_{ij} over $\mathbb{R}^d \times \{i, j, 0\}$ sampled as follows. Draw $(\mathbf{x}, y) \sim D$, if $y \in \{i, j\}$, return $(\mathbf{x}, y)$, otherwise, return $(\mathbf{x}, 0)$
 - 5: Run $\mathcal{A}$ over input distribution D_{ij} with accuracy parameter ϵ and get hypothesis $\hat{h}_{ij} : \mathbb{R}^d \rightarrow \{i, j\}$.
 - 6: Return hypothesis $\hat{h}(\mathbf{x}) = \operatorname{argmax}_{i \in [K]} s(\mathbf{x}, i)$, where $s(\mathbf{x}, i) = \sum_{j \neq i} \mathbb{1}\{\hat{h}_{ij}(\mathbf{x}) = i\}$.
-

Proof of Lemma 7.22. We show that Algorithm 7 satisfies the statement of Lemma 7.22. The sample complexity and the time complexity of Algorithm 7 follows directly. In the rest of the proof, we establish the correctness of Algorithm 7.

Let h_W be any multiclass linear classifier. For every $i \neq j \in [K]$, let $h_{ij}(\mathbf{x}) = \operatorname{sign}((W_i - W_j) \cdot (\mathbf{x}, 1))$ be a halfspace. Notice that for every $\mathbf{x}$ and $i \in [K]$, $h_W(\mathbf{x}) = i$ if and only if $h_{ij}(\mathbf{x}) = 1$, for all $j \neq i$. To simplify the notation, we will also treat the label of h_{ij} in $\{i, j\}$, where $h_{ij}(\mathbf{x}) = i$ if and only if the halfspace predicts $+1$.

For every $i \neq j$, we define distribution D_{ij} over $\mathbb{R}^d \times \{i, j, 0\}$ that is sampled in the following way. Draw $(\mathbf{x}, y) \sim D$, if $y \in \{i, j\}$, return $(\mathbf{x}, y)$, otherwise return $(\mathbf{x}, 0)$. Let $h^* = h_{W^*}$ be the multiclass linear classifier that achieves optimal error opt over D and let $h_{ij}^* = \operatorname{sign}((W_i^* - W_j^*) \cdot (\mathbf{x}, 1))$. For each pair of $i \neq j$, we have

$$\begin{aligned}
\operatorname{err}_{D_{ij}}(h_{ij}^*) &= \Pr_{(\mathbf{x}, y) \sim D} [h_{ij}^*(\mathbf{x}) \neq y, y \in \{i, j\}] \\
&= \sum_{r \in [K]} \Pr_{(\mathbf{x}, y) \sim D} [h_{ij}^*(\mathbf{x}) \neq y, h^*(\mathbf{x}) = r, y \in \{i, j\}] \\
&\leq \Pr_{(\mathbf{x}, y) \sim D} [h^*(\mathbf{x}) = i, y = j] + \Pr_{(\mathbf{x}, y) \sim D} [h^*(\mathbf{x}) = j, y = i] + \sum_{r \notin \{i, j\}} \Pr_{(\mathbf{x}, y) \sim D} [h^*(\mathbf{x}) = r, y \in \{i, j\}] \\
&= \operatorname{opt}_{ij} + \operatorname{opt}_{ji} + \sum_{r \notin \{i, j\}} (\operatorname{opt}_{r,i} + \operatorname{opt}_{r,j}),
\end{aligned}$$

where $\operatorname{opt}_{r,i} = \Pr_{(\mathbf{x}, y) \sim D} [h^*(\mathbf{x}) = r, y = i]$. This implies by running $\mathcal{A}$ over D_{ij} , it draws $M(\epsilon)$ examples, runs in $T(\epsilon)$ time, and outputs a hypothesis $\hat{h}_{ij}$ such that $\operatorname{err}_{D_{ij}}(\hat{h}_{ij}) \leq C(\epsilon)(\operatorname{opt}_{ij} + \operatorname{opt}_{ji} + \sum_{r \notin \{i, j\}} (\operatorname{opt}_{r,i} + \operatorname{opt}_{r,j})) + \epsilon$.

Using the $O(K^2)$ learned $\hat{h}_{ij}$, we define the function $\hat{h}$ as follows. For each $\mathbf{x}$ and each $i \in [K]$,

we define $s(\mathbf{x}, i) = \sum_{j \neq i} \mathbb{1}\{\hat{h}_{ij}(\mathbf{x}) = i\}$ and return $\hat{h}(\mathbf{x}) = \operatorname{argmax}_{i \in [K]} s(\mathbf{x}, i)$. We have

$$\begin{aligned}
\operatorname{err}_D(\hat{h}) &= \sum_{j=1}^K \mathbf{Pr}_{(\mathbf{x}, y) \sim D} [\hat{h}(\mathbf{x}) \neq y, y = j] \leq \sum_{j=1}^K \sum_{i \neq j} \mathbf{Pr}_{(\mathbf{x}, y) \sim D} [\hat{h}_{ji}(\mathbf{x}) \neq j, y = j] \\
&= \frac{1}{2} \sum_{(i,j): i \neq j} \mathbf{Pr}_{(\mathbf{x}, y) \sim D} [\hat{h}_{ij}(\mathbf{x}) \neq y, y \in \{i, j\}] \\
&\leq \frac{1}{2} \sum_{(i,j): i \neq j} \left(C(\epsilon)(\operatorname{opt}_{ij} + \operatorname{opt}_{ji} + \sum_{r \notin \{i,j\}} (\operatorname{opt}_{r,i} + \operatorname{opt}_{r,j})) + \epsilon \right) \\
&\leq \frac{1}{2} C(\epsilon) \left(\sum_{i=1}^K \sum_{j=1}^K \sum_{r \neq i} \operatorname{opt}_{r,i} + \sum_{i=1}^K \sum_{j=1}^K \sum_{r \neq j} \operatorname{opt}_{r,j} \right) + \frac{1}{2} K^2 \epsilon \\
&\leq KC(\epsilon)(\operatorname{opt}_D(\mathcal{M}_{K,d})) + K^2 \epsilon,
\end{aligned}$$

where the last inequality holds by the fact that $\sum_{(i,j): i \neq j} \operatorname{opt}_{ij} = \operatorname{opt}$. $\square$

We now make use of Lemma 7.22 to prove Theorem 7.21.

Proof of Theorem 7.21. Recall that for every hypothesis $h : \mathbb{R}^d \rightarrow [K]$ and for every D over $\mathbb{R}^d \times [K] \cup \{0\}$, the global error satisfies that $\operatorname{err}_{D_f}(h) = \operatorname{err}_D(h)/W(D)$. Since $W(D) \geq \alpha$, this implies that for some ϵ' that we will choose later, if we get a hypothesis such that $\operatorname{err}_D(\hat{h}) \leq KC(\epsilon')(\operatorname{opt}_D(\mathcal{M}_{K,d})) + K^2 \epsilon'$

$$\operatorname{err}_{D_f}(\hat{h}) \leq KC(\epsilon')(\operatorname{opt}_{D_f}(\mathcal{M}_{K,d})) + K^2 \epsilon' / \alpha.$$

By Lemma 7.22, by setting $\epsilon' = \alpha\epsilon/K^2$, suppose we have an algorithm that under every distribution D' over $\mathbb{R}^d \times \{0, \pm 1\}$, output a boolean hypothesis $h_b : \mathbb{R}^d \rightarrow \{\pm 1\}$, such that $\operatorname{err}_{D'}(h_b) \leq C(\epsilon')(\operatorname{opt}_{D'}(\mathcal{H})) + \epsilon'$, which means $\operatorname{err}_{D'_f}(h_b) \leq C(\epsilon')(\operatorname{opt}_{D'_f}(\mathcal{H})) + (\epsilon/K^2)$. then Algorithm 7, gives a hypothesis $\hat{h}$ such that

$$\operatorname{err}_{D_f}(\hat{h}) \leq KC(\epsilon')(\operatorname{opt}_{D_f}(\mathcal{M}_{K,d})) + \epsilon.$$

This implies that by running the algorithm in Theorem 7.14 and Theorem 7.15 $O(K^2)$ times, we can get a hypothesis $\hat{h}$, with error $\operatorname{err}(\hat{h}) = O(K \operatorname{opt}_{D_f}(\mathcal{M}_{K,d})) + \epsilon$ and $\operatorname{err}(\hat{h}) = O(\log^4(K/(\alpha\epsilon)))(K \operatorname{opt}_{D_f}(\mathcal{M}_{K,d})) + \epsilon$, respectively. This concludes the proof of Theorem 7.21. $\square$

References

- [ABL17] P. Awasthi, M. F. Balcan, and P. M. Long. The power of localization for efficiently learning linear separators with noise. *J. ACM*, 63(6):50:1–50:27, 2017.
- [AK18] L. M. Arutyunyan and E. D. Kosov. Deviation of polynomials from their expectations and isoperimetry. 2018.
- [APS26] J. Alman, S. Patel, and R. A. Servedio. Learning functions of halfspaces. 2026, 2603.08700. URL <https://arxiv.org/abs/2603.08700>.

- [AV06] R. Arriaga and S. Vempala. An algorithmic theory of learning: Robust concepts and random projection. *Machine learning*, 63:161–182, 2006.
- [BK97] A. Blum and R. Kannan. Learning an intersection of a constant number of halfspaces over a uniform distribution. *Journal of Computer and System Sciences*, 54(2):371–380, 1997.
- [BM02] P. L. Bartlett and S. Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of machine learning research*, 3(Nov):463–482, 2002.
- [BS71] J. Bochnak and J. Siciak. Polynomials and multilinear mappings in topological vector-spaces. *Studia Mathematica*, 39(1):59–76, 1971.
- [CW01] A. Carbery and J. Wright. Distributional and l^q norm inequalities for polynomials over convex bodies in $\mathbb{R}^n$. *Mathematical research letters*, 8(3):233–248, 2001.
- [Dan16] A. Daniely. Complexity theoretic limitations on learning halfspaces. In *Proceedings of the 48th Annual Symposium on Theory of Computing, STOC 2016*, pages 105–117, 2016.
- [DIKZ25] I. Diakonikolas, G. Iakovidis, D. M. Kane, and N. Zarifis. Robust learning of multi-index models via iterative subspace approximation. In *66th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2025, Sydney, Australia, December 14-17, 2025*, pages 2231–2239. IEEE, 2025.
- [DIM26] I. Diakonikolas, G. Iakovidis, and M. Ma. Polynomial-time robust multiclass linear classification under gaussian marginals. *arXiv preprint arXiv:2605.21428*, 2026.
- [DKLZ24] I. Diakonikolas, D. M. Kane, S. Liu, and N. Zarifis. Testable learning of general halfspaces with adversarial label noise. In *The Thirty Seventh Annual Conference on Learning Theory*, volume 247 of *Proceedings of Machine Learning Research*, pages 1308–1335. PMLR, 2024.
- [DKM24] I. Diakonikolas, D. Kane, and M. Ma. Active learning of general halfspaces: Label queries vs membership queries. *Advances in Neural Information Processing Systems*, 37:49180–49213, 2024.
- [DKPZ21] 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.
- [DKR23] I. Diakonikolas, D. Kane, and L. Ren. Near-optimal cryptographic hardness of agnostically learning halfspaces and relu regression under gaussian marginals. In *International Conference on Machine Learning*, pages 7922–7938. PMLR, 2023.
- [DKS18] I. Diakonikolas, D. M. Kane, and A. Stewart. Learning geometric concepts with nasty noise. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018*, pages 1061–1073, 2018.
- [DKTZ22] I. Diakonikolas, V. Kontonis, C. Tzamos, and N. Zarifis. Learning general halfspaces with adversarial label noise via online gradient descent. In *International Conference on Machine Learning*, pages 5118–5141. PMLR, 2022.

- [DMRT25a] I. Diakonikolas, M. Ma, L. Ren, and C. Tzamos. Learning intersections of two margin halfspaces under factorizable distributions. In N. Haghtalab and A. Moitra, editors, *The Thirty Eighth Annual Conference on Learning Theory, 30-4 July 2025, Lyon, France*, Proceedings of Machine Learning Research, pages 1472–1530. PMLR, 2025.
- [DMRT25b] I. Diakonikolas, M. Ma, L. Ren, and C. Tzamos. Statistical query hardness of multiclass linear classification with random classification noise. In *International Conference on Machine Learning*, pages 13693–13712. PMLR, 2025.
- [Eld13] R. Eldan. Thin shell implies spectral gap up to polylog via a stochastic localization scheme. *Geometric and Functional Analysis*, 23(2):532–569, 2013.
- [Fel10] V. Feldman. Distribution-specific agnostic boosting. In *Proceedings of Innovations in Computer Science*, pages 241–250, 2010.
- [FJS81] J. H. Friedman, M. Jacobson, and W. Stuetzle. Projection Pursuit Regression. *J. Am. Statist. Assoc.*, 76:817, 1981.
- [FS97] Y. Freund and R. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1):119–139, 1997.
- [Gav03] D. Gavinsky. Optimally-smooth adaptive boosting and application to agnostic learning. *Journal of Machine Learning Research*, 4(May):101–117, 2003.
- [GWBL20] S. Garg, Y. Wu, S. Balakrishnan, and Z. Lipton. A unified view of label shift estimation. In *Advances in Neural Information Processing Systems*, volume 33, pages 3290–3300. Curran Associates, Inc., 2020.
- [HL93] P. Hall and K.-C. Li. On almost Linearity of Low Dimensional Projections from High Dimensional Data. *The Annals of Statistics*, 21(2):867 – 889, 1993.
- [HL13] C. J. Hillar and L.-H. Lim. Most tensor problems are np-hard. *Journal of the ACM (JACM)*, 60(6):1–39, 2013.
- [Hub85] P. J. Huber. Projection Pursuit. *The Annals of Statistics*, 13(2):435 – 475, 1985.
- [Kea98] M. J. Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, 45(6):983–1006, 1998.
- [KK09] A. Kalai and V. Kanade. Potential-based agnostic boosting. In *Advances in Neural Information Processing Systems 22: 23rd Annual Conference on Neural Information Processing Systems 2009*, pages 880–888, 2009.
- [KKM13] D. Kane, A. Klivans, and R. Meka. Learning halfspaces under log-concave densities: Polynomial approximations and moment matching. In *Conference on Learning Theory*, pages 522–545. PMLR, 2013.
- [KKMS08] A. Kalai, A. Klivans, Y. Mansour, and R. Servedio. Agnostically learning halfspaces. *SIAM Journal on Computing*, 37(6):1777–1805, 2008.
- [KL25] B. Klartag and J. Lehec. Thin-shell bounds via parallel coupling. *arXiv preprint arXiv:2507.15495*, 2025.

- [KLT09] 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*, pages 588–600, 2009.
- [KOS04] A. Klivans, R. O’Donnell, and R. Servedio. Learning intersections and thresholds of halfspaces. *Journal of Computer & System Sciences*, 68(4):808–840, 2004.
- [KS04] A. Klivans and R. Servedio. Learning intersections of halfspaces with a margin. In *Proceedings of the 17th Annual Conference on Learning Theory*,, pages 348–362, 2004.
- [KS08] S. Khot and R. Saket. On hardness of learning intersection of two halfspaces. In *Proc. 40th Annual ACM Symposium on Theory of Computing (STOC)*, pages 345–354, 2008.
- [Lat11] R. Latafai. Order statistics and concentration of ℓ_1 norms for log-concave vectors. *Journal of Functional Analysis*, 261:681–696, 2011.
- [Li91] K.-C. Li. Sliced inverse regression for dimension reduction. *Journal of the American Statistical Association*, 86(414):316–327, 1991.
- [Lot15] M. Lotz. On the volume of tubular neighborhoods of real algebraic varieties. *Proceedings of the American Mathematical Society*, 143(5):1875–1889, 2015.
- [LV07] L. Lovász and S. Vempala. The geometry of logconcave functions and sampling algorithms. *Random Structures & Algorithms*, 30(3):307–358, 2007.
- [LV18] Y. T. Lee and S. S. Vempala. The kannan-lovász-simonovits conjecture. *arXiv preprint arXiv:1807.03465*, 2018.
- [LWS18] Z. Lipton, Y.-X. Wang, and A. Smola. Detecting and correcting for label shift with black box predictors. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 3122–3130. PMLR, 10–15 Jul 2018.
- [Mil09] E. Milman. On the role of convexity in isoperimetry, spectral gap and concentration. *Inventiones mathematicae*, 177(1):1–43, 2009.
- [MP68] M. Minsky and S. Papert. *Perceptrons: an introduction to computational geometry*. MIT Press, Cambridge, MA, 1968.
- [MT94] W. Maass and G. Turan. How fast can a threshold gate learn? In S. Hanson, G. Drastal, and R. Rivest, editors, *Computational Learning Theory and Natural Learning Systems*, pages 381–414. MIT Press, 1994.
- [NDRT13] N. Natarajan, I. S. Dhillon, P. K. Ravikumar, and A. Tewari. Learning with noisy labels. In C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26, 2013.
- [Nov62] A. Novikoff. On convergence proofs on perceptrons. In *Proceedings of the Symposium on Mathematical Theory of Automata*, volume XII, pages 615–622, 1962.
- [Pao12] G. Paouris. Small ball probability estimates for log-concave measures. *Transactions of the American Mathematical Society*, 364(1):287–308, 2012.

- [Ros58] F. Rosenblatt. The Perceptron: a probabilistic model for information storage and organization in the brain. *Psychological Review*, 65:386–407, 1958.
- [RV23] R. Rubinfeld and A. Vasilyan. Testing distributional assumptions of learning algorithms. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1643–1656, 2023.
- [SB14] S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning - From Theory to Algorithms*. Cambridge University Press, 2014.
- [Ser03] R. Servedio. Smooth boosting and learning with malicious noise. *Journal of Machine Learning Research*, 4:633–648, 2003.
- [Sha04] A. Shadrin. Twelve proofs of the markov inequality. *Approximation theory: a volume dedicated to Borislav Bojanov*, pages 233–298, 2004.
- [She09] A. Sherstov. The intersection of two halfspaces has high threshold degree. In *Proc. 50th IEEE Symposium on Foundations of Computer Science (FOCS)*, 2009.
- [SKP⁺20] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee. Learning from noisy labels with deep neural networks: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–29, 2020.
- [STC00] J. Shawe-Taylor and N. Cristianini. *An introduction to support vector machines*. Cambridge University Press, 2000.
- [Tie23] S. Tiegel. Hardness of agnostically learning halfspaces from worst-case lattice problems. In *The Thirty Sixth Annual Conference on Learning Theory, COLT 2023*, volume 195 of *Proceedings of Machine Learning Research*, pages 3029–3064. PMLR, 2023.
- [Val84] L. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.
- [Vap98] V. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, New York, 1998.
- [Vem10a] S. Vempala. Learning convex concepts from gaussian distributions with PCA. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pages 124–130, 2010.
- [Vem10b] S. Vempala. A random-sampling-based algorithm for learning intersections of halfspaces. *J. ACM*, 57(6):32:1–32:14, 2010.
- [WLT18] R. Wang, T. Liu, and D. Tao. Multiclass learning with partially corrupted labels. *IEEE Transactions on Neural Networks and Learning Systems*, 29(6):2568–2580, 2018.
- [Xia08] Y. Xia. A multiple-index model and dimension reduction. *Journal of the American Statistical Association*, 103(484):1631–1640, 2008.
- [XTLZ02] 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*, 64(3):363–410, 2002.

Appendix

A Probability Lemmas

Fact A.1 (Theorem 5.14 in [LV07]). *Let $f(\mathbf{x})$ be the density function of an isotropic logconcave distribution in $\mathbb{R}^d$. For every $\mathbf{x}$, with $\|\mathbf{x}\| \leq 1/9$, it holds $f(\mathbf{x}) \geq 2^{-9d\|\mathbf{x}\|} f(0)$ and $f(x) \geq 2^{-7d}$.*

Fact A.2 (Lemma 5.7 in [LV07]). *Let $D_{\mathbf{x}}$ be an isotropic logconcave distribution over $\mathbb{R}^d$ and let $\mathbf{w} \in \mathbb{S}^{d-1}$ be a unit vector. There is a constant $C > 0$ such that for every $t > 0$, $\Pr_{\mathbf{x} \sim D_{\mathbf{x}}}[\|\mathbf{w} \cdot \mathbf{x}\| > t] < Ce^{-t}$.*

Fact A.3 ([Pao12]). *Let $D_{\mathbf{x}}$ be an isotropic logconcave distribution over $\mathbb{R}^d$, then for $\epsilon \in (0, 1)$*

$$\Pr_{\mathbf{x} \sim D_{\mathbf{x}}} \left[\|\mathbf{x}\|_2 < \epsilon\sqrt{d} \right] \leq \exp(-\Omega(d)).$$

Fact A.4 (Theorem 13 in [LV18]). *Let $D_{\mathbf{x}}$ be an isotropic logconcave distribution over $\mathbb{R}^d$ and let $g : \mathbb{R}^d \rightarrow \mathbb{R}$ be an L -Lipschitz function, then for $t > 0$,*

$$\Pr_{\mathbf{x} \sim D_{\mathbf{x}}} \left[\left| g(\mathbf{x}) - \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [g(\mathbf{x})] \right| > Lt \right] \leq \exp(-\Omega(\psi_D t)),$$

where ψ_D is the Cheeger constant of $D_{\mathbf{x}}$.

As an implication of Fact A.4, we have the following tail bound for the L_p norm of a random vector drawn from an isotropic logconcave distribution.

Fact A.5 (Theorem 8 in [Lat11]). *Let $D_{\mathbf{x}}$ be an isotropic logconcave distribution over $\mathbb{R}^d$. For $p \geq 2$ and $t > 1$, it holds*

$$\Pr_{\mathbf{x} \sim D_{\mathbf{x}}} \left[\|\mathbf{x}\|_p > C(p)td^{1/p} \right] \leq \exp(-td^{1/p}),$$

where $C(p)$ is a universal constant that only depends on p .

Fact A.6 (Theorem 1.1 in [AK18]). *Let $D_{\mathbf{x}}$ be an isotropic logconcave distribution over $\mathbb{R}^d$ and $p(\mathbf{x}) : \mathbb{R}^d \rightarrow \mathbb{R}$ be a polynomial with degree at most k . Then, for $t > 1$, it holds*

$$\Pr_{\mathbf{x} \sim D_{\mathbf{x}}} [|p(\mathbf{x})| > \sqrt{e}(4t)^k \|p\|_2] \leq e^{-t}.$$

Fact A.7 (Theorem 8 in [CW01]). *There is a universal constant $C > 0$ such that for $D_{\mathbf{x}}$, an isotropic logconcave distribution over $\mathbb{R}^d$ and for $p(\mathbf{x}) : \mathbb{R}^d \rightarrow \mathbb{R}$, a polynomial with degree k and for $\epsilon > 0$, it holds*

$$\Pr_{\mathbf{x} \sim D_{\mathbf{x}}} [|p(\mathbf{x})| < \epsilon] \leq Ck \frac{\epsilon^{1/k}}{\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [p^2(\mathbf{x})]^{1/2k}}.$$

Fact A.8. *Let D be a distribution on $\mathbb{R}^d \times \{0, \pm 1\}$, such that $D_{\mathbf{x}}$ is an isotropic logconcave distribution. There is a constant C such that for $S(\mathbf{x}, y)$ an event over $\mathbb{R}^d \times \{0, \pm 1\}$ and $\mathbf{w} \in \mathbb{S}^{d-1}$ a unit vector, it holds*

$$\mathbf{E}_{(\mathbf{x}, y) \sim D} [\|\mathbf{w} \cdot \mathbf{x}\| \mathbf{1}\{S(\mathbf{x}, y)\}] \leq C \Pr_{(\mathbf{x}, y) \sim D} [S(\mathbf{x}, y)] \log \left(\frac{e}{\Pr_{(\mathbf{x}, y) \sim D} [S(\mathbf{x}, y)]} \right).$$

Proof of Fact A.8. Let $Z := \mathbf{w} \cdot \mathbf{x}$. Then, Z is also an isotropic logconcave distribution. For $z \in \mathbb{R}$, we denote by $z_+ := \max\{z, 0\}$. Notice that for every $z, s \in \mathbb{R}$, we have $|z| \leq s + (|z| - s)_+$. This implies for every $s \in \mathbb{R}$, we have

$$\begin{aligned} \mathbf{E}_{(\mathbf{x}, y) \sim D} [|\mathbf{w} \cdot \mathbf{x}| \mathbf{1}\{S(\mathbf{x}, y)\}] &\leq \mathbf{Pr}_{(\mathbf{x}, y) \sim D} [S(\mathbf{x}, y)]s + \mathbf{E}_{(\mathbf{x}, y) \sim D} [(|Z| - s)_+] \\ &= \mathbf{Pr}_{(\mathbf{x}, y) \sim D} [S(\mathbf{x}, y)]s + \int_s^\infty \mathbf{Pr}_Z[|Z| > r] dr \\ &\leq \mathbf{Pr}_{(\mathbf{x}, y) \sim D} [S(\mathbf{x}, y)]s + Ce^{-s}, \end{aligned}$$

where the last inequality follows Fact A.2. By taking $s = \log(C / \mathbf{Pr}_{(\mathbf{x}, y) \sim D} [S(\mathbf{x}, y)])$, we get

$$\mathbf{E}_{(\mathbf{x}, y) \sim D} [|\mathbf{w} \cdot \mathbf{x}| \mathbf{1}\{S(\mathbf{x}, y)\}] \leq C \mathbf{Pr}_{(\mathbf{x}, y) \sim D} [S(\mathbf{x}, y)] \log \left(\frac{e}{\mathbf{Pr}_{(\mathbf{x}, y) \sim D} [S(\mathbf{x}, y)]} \right).$$

□

Fact A.9. Let D be an isotropic logconcave distribution over $\mathbb{R}^d$ and $\mathbf{u}^{(1)}, \dots, \mathbf{u}^{(k)}$ be k unit vectors in $\mathbb{R}^d$. Then,

$$\left| \mathbf{E}_{\mathbf{x} \sim D} \left[\prod_{i=1}^k (\mathbf{u}^{(i)} \cdot \mathbf{x}) \right] \right| \leq (2k)^k.$$

Proof of Fact A.9. By Holder's inequality and Lemma 33 in [LV18], we have

$$\left| \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} \left[\prod_{i=1}^k (\mathbf{u}^{(i)} \cdot \mathbf{x}) \right] \right| \leq \prod_{i=1}^k \left(\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [|\mathbf{u}^{(i)} \cdot \mathbf{x}|^k]^{1/k} \right) \leq \prod_{\ell \notin S} 2k = (2k)^k.$$

□

Fact A.10 (Theorem 7 in [CW01]). There is a constant $C > 0$ such that for every logconcave distribution D and for every polynomial $p(\mathbf{x})$ with degree k , $0 < r < q < \infty$

$$(\mathbf{E}_{\mathbf{x} \sim D} [p(\mathbf{x})^{q/k}])^{1/q} \leq C \frac{\max(q, 1)}{\max(r, 1)} (\mathbf{E}_{\mathbf{x} \sim D} [p(\mathbf{x})^{r/k}])^{1/r}.$$

In particular $\mathbf{E}_{\mathbf{x} \sim D} [p(\mathbf{x})^\ell] \leq (C\ell)^{\ell k} \mathbf{E}_{\mathbf{x} \sim D} [p(\mathbf{x})^2]^\ell$, $\forall \ell > 2$, $\mathbf{E}_{\mathbf{x} \sim D} [p(\mathbf{x})^\ell] \leq (C\ell)^{\ell k} \mathbf{E}_{\mathbf{x} \sim D} [p(\mathbf{x})]^\ell$, $\forall \ell > 1$

Fact A.11. Let D be a $1/2$ -logconcave distribution over $\mathbb{R}$ and let $p(x) = \sum_0^\ell c_i x^i : \mathbb{R} \rightarrow \mathbb{R}$ be a degree ℓ polynomial, then

$$\sum_{i=0}^\ell |c_i| \leq (C\ell)^\ell \mathbf{E}_{x \sim D} [|p(x)|].$$

Proof of Fact A.11. Let $\mu = \mathbf{E}_{x \sim D}$ and let $D(x)$ be the density function of D . By Theorem 5.14 in [LV07], we know that $D(\mu) > 2^{-8}$ and for every $x = \mu + z$, $|z| < 1/9$, we have $D(x) > 2^{-9}$. Let $I = [\mu - 1/9, \mu + 1/9]$. This implies that

$$\int_I |p(x)| dx \leq 2^9 \int_I |p(x)| D(x) dx \leq 2^9 \mathbf{E}_{x \sim D} [|p(x)|].$$

We center I , by taking $y = x - \mu$, and $I' = [-1/9, 1/9]$. Then, we have

$$\int_{I'} |p(y + \mu)| dy = \int_I |p(x)| dx \leq 2^9 \mathbf{E}_{x \sim D} [|p(x)|].$$

Notice that $c_j = p^{(j)}(0)/(j!)$. By Markov's inequality [Sha04], we have $|p^{(j)}(0)| \leq (Ck)^k \max_{[-1,1]} |p(x)| \leq (C'k)^k \int_{[-1,1]} |p(x)| dx$. This gives

$$\sum_{i=0}^{\ell} |c_i| \leq (C\ell)^\ell \mathbf{E}_{x \sim D} [|p(x)|].$$

□

Fact A.12. Let $D_{\mathbf{x}}$ be a $1/2$ isotropic logconcave distribution on $\mathbb{R}^d$. Let $p(\mathbf{x}) = \sum_{|\alpha| \leq k} c_\alpha \mathbf{x}^\alpha$ be a polynomial of degree at most k , where each α is a multi index. Then

$$\sum_{|\alpha| \leq k} \left(|c_\alpha|^2 \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\mathbf{x}^{2\alpha}] \right)^{1/2} \leq C_{d,k} \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [|p(\mathbf{x})|],$$

where $C_{d,k} \leq \binom{d+k}{k} (Ck)^{Cdk} d^{k/2} 2^{7d}$.

Proof of Fact A.12. We first assume that $D_{\mathbf{x}}$ is isotropic for convenience. Let f denote the density of $D_{\mathbf{x}}$. By Fact A.1, we know that $f(\mathbf{x}) \geq 2^{-7d}$, when $\|\mathbf{x}\| \leq 1/9$. Let $r = d^{-1/2}/9$ and $Q_r := [-r, r]^d$. Since $Q_r \subseteq \mathbb{B}^{d-1}(1/9)$, for every $\mathbf{x} \in Q_r$, $f(\mathbf{x}) \geq 2^{-7d}$. So,

$$\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [|p(\mathbf{x})|] = \int_{\mathbb{R}^d} |p(\mathbf{x})| f(\mathbf{x}) d\mathbf{x} \geq 2^{-7d} \int_{Q_r} |p(\mathbf{x})| d\mathbf{x}.$$

This gives $\int_{Q_r} |p(\mathbf{x})| d\mathbf{x} \leq 2^{7d} \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [|p(\mathbf{x})|]$.

We next rescale $\mathbf{x}$ from Q_r to $[-1, 1]^d$. For $\mathbf{u} \in \mathbb{R}^d$, we define $q(\mathbf{u}) = p(r\mathbf{u})$. Then, we have $q(\mathbf{u}) = \sum_{|\alpha| \leq k} c_\alpha r^{|\alpha|} \mathbf{u}^\alpha$. Then, we have $\int_{[-1,1]^d} |q(\mathbf{u})| d\mathbf{u} = r^{-d} \int_{Q_r} |p(\mathbf{x})| d\mathbf{x}$.

Let matrix $M \in \mathbb{R}^{(k+1) \times (k+1)}$ such that $M_{ij} = \int_{-1,1} t^{i+j} dt$, $i, j = 0, \dots, k$. For $i = 0, \dots, k$, define $\psi_i(t) = \sum_{j=0}^k (M^{-1})_{ij} t^j$. Notice that

$$\int_{[-1,1]} \psi_i(t) t^j dt = \sum_{m=0}^k (M^{-1})_{im} \int_{[-1,1]} t^{m+j} dt = \sum_{m=0}^k (M^{-1})_{im} M_{mj} = \delta_{ij}.$$

For any multi-index $\alpha = (\alpha_1, \dots, \alpha_d)$ define $\Psi_\alpha(\mathbf{u}) := \prod_{m=1}^d \psi_{\alpha_m}(\mathbf{u}_m)$. Then, we have

$$\int_{[-1,1]^d} \mathbf{u}^\beta \Psi_\alpha(\mathbf{u}) d\mathbf{u} = \prod_{m=1}^d \int_{-1}^1 \mathbf{u}_m^{\beta_m} \psi_{\alpha_m}(\mathbf{u}_m) d\mathbf{u}_m = \delta_{\alpha\beta}.$$

This means $c_\alpha r^{|\alpha|} = \int_{[-1,1]^d} q(\mathbf{u}) \Psi_\alpha(\mathbf{u}) d\mathbf{u}$. This implies

$$|c_\alpha r^{|\alpha|}| \leq \|\Psi_\alpha(\mathbf{u})\|_\infty \int_{[-1,1]^d} |q(\mathbf{u})| d\mathbf{u}$$

Notice that for each i , $\|\psi_i\|_\infty \leq 2^{O(k \log k)}$, which implies that $\|\Psi_\alpha(\mathbf{u})\|_\infty \leq 2^{O(kd \log k)}$. Since there are $N = \binom{d+k}{k}$ multi-indices, we have

$$\sum_{|\alpha| \leq k} |c_\alpha| r^{|\alpha|} \leq N 2^{O(kd \log k)} r^{-d} 2^{7d} \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [|p(\mathbf{x})|].$$

By Fact A.9, we know that $\mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\mathbf{x}^{2\alpha}] \leq (4k)^{2k}$. This gives

$$\sum_{|\alpha| \leq k} \left(|c_\alpha|^2 \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\mathbf{x}^{2\alpha}] \right)^{1/2} \leq C'_{d,k} \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [|p(\mathbf{x})|],$$

where $C'_{d,k} \leq \binom{d+k}{k} (Ck)^{Cdk} d^{k/2} 2^{7d}$.

To conclude the proof, we now assume that $D_{\mathbf{x}}$ is $1/2$ -isotropic with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. Let $\mathbf{x}' = \boldsymbol{\Sigma}^{-1/2}(\mathbf{x} - \boldsymbol{\mu})$, so that $\mathbf{x}'$ follows an isotropic logconcave distribution $D'_{\mathbf{x}}$. We have

$$p(\mathbf{x}) = \sum_{|\alpha| \leq k} c_\alpha \mathbf{x}^\alpha = \sum_{|\alpha| \leq k} c_\alpha (\boldsymbol{\Sigma}^{1/2}(\mathbf{x}' + \boldsymbol{\mu}))^\alpha.$$

We write $g(\mathbf{x}') = \sum_{|\alpha| \leq k} c_\alpha (\boldsymbol{\Sigma}^{1/2}(\mathbf{x}' + \boldsymbol{\mu}))^\alpha = \sum_\beta a_\beta (\mathbf{x}')^\beta$. This means

$$\sum_{|\alpha| \leq k} \left(|c_\alpha|^2 \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\mathbf{x}^{2\alpha}] \right)^{1/2} = \sum_{|\alpha| \leq k} \left(|c_\alpha|^2 \mathbf{E}_{\mathbf{x}' \sim D'_{\mathbf{x}}} [(\boldsymbol{\Sigma}^{1/2}(\mathbf{x}' + \boldsymbol{\mu}))^{2\alpha}] \right)^{1/2} \leq 2^{O(k)} \sum_{|\beta| \leq k} \left(|a_\beta|^2 \mathbf{E}_{\mathbf{x}' \sim D'_{\mathbf{x}}} [(\mathbf{x}')^{2\beta}] \right)^{1/2}$$

This gives $\sum_{|\alpha| \leq k} \left(|c_\alpha|^2 \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [\mathbf{x}^{2\alpha}] \right)^{1/2} \leq 2^{O(k)} C'_{d,k} \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [|p(\mathbf{x})|]$ and concludes the proof. $\square$

B Omitted Details from Section 3

In this section, we give omitted details in Section 3, by presenting the proof of Theorem 3.3 from [Fel10]. We restate the fact below.

Fact B.1 (Corollary of Theorem 3.5 in [Fel10]). *Let $\mathcal{H}$ be a hypothesis class over $\mathbb{R}^d$ and let $\mathcal{A}$ be an algorithm with the following behavior: for every distribution D over $\mathbb{R}^d \times \{\pm 1\}$ such that $\min_{h \in \mathcal{H}} \text{err}_D(h) \leq 1/2 - a$, the algorithm draws m examples from D , runs in time T , and outputs a hypothesis $\hat{h}$ such that $\text{err}_D(h) \leq 1/2 - \beta$, with probability at least 99%. Then there is an algorithm $\mathcal{B}$ with the following performance guarantee: for every distribution D over $\mathbb{R}^d \times \{\pm 1\}$ and for every $\epsilon > 0$, $\mathcal{B}$ makes $\tilde{O}(\beta^{-2}/\epsilon)$ many calls to $\mathcal{A}$, it runs in time $\text{poly}(T, \beta^{-1}, \epsilon^{-1})$, and outputs a hypothesis $\hat{h}$ such that $\text{err}_D(\hat{h}) \leq \text{opt}_D(\mathcal{H})/(1 - 2a) + \epsilon$, with probability 99%. In particular, each call to $\mathcal{A}$ is defined over some distribution D' such that for every $\mathbf{z} \in \mathbb{R}^d$, $D'_{\mathbf{x}}(\mathbf{z}) = D_{\mathbf{x}}(\mathbf{z}) w(\mathbf{z})$, for some weight function $w : \mathbb{R}^d \rightarrow [0, 1/\epsilon]$.*

Proof of Fact 3.3. We will be consistent with the notations used in [Fel10]. For $a \in \mathbb{R}$, define $P_1(a) = a$ if $|a| \leq 1$ and $P_1(a) = \text{sign}(a)$ otherwise. Furthermore, define $R(a) = a^2$ if $|a| \leq 1$ and $2|a| - 1$ otherwise. We list several properties of P_1 and R , which are proved in [Fel10]. For every Boolean function f and distribution D , it holds

1. $\mathbf{E}_D[R(f)] = 1$ and for every real-valued function ϕ , $\mathbf{E}_D[R(f - \phi)] \geq 0$.
2. For $h, g : \mathbb{R}^d \rightarrow [-1, 1]$, if $\langle P_1(f - h), g \rangle \geq 2\gamma$, then $\mathbf{E}_D[R(f - (h + \gamma g))] \leq \mathbf{E}_D[R(f - h)] - 3\gamma^2$.

3. For $h : \mathbb{R}^d \rightarrow [-1, 1]$, $\mathbf{E}_D[R(f - P_1(h))] \leq \mathbf{E}_D[R(f - h)]$

We describe the algorithm used in [Fel10] here. Denote by $f(\mathbf{x}) = \mathbf{E}_D[y \mid \mathbf{x}]$, for each $\mathbf{x} \in \mathbb{R}^d$. According to the proof in [Fel10], we will also assume f is a Boolean function. We initialize $h_0 \equiv 0$. In each time step $t \geq 0$, we will maintain a function $h_t : \mathbb{R}^d \rightarrow [-1, 1]$ and maintain a distribution D_t defined as follows. For each $\mathbf{x} \in \mathbb{R}^d$, $(D_t)_{\mathbf{x}}(\mathbf{x}) = D_{\mathbf{x}}(\mathbf{x}) |P_1(f(\mathbf{x}) - h_t(\mathbf{x}))| / N_t$, where $N_t = \|P_1(f(\mathbf{x}) - h_t(\mathbf{x}))\|_{D,1}$ and $D_t(y \mid \mathbf{x}) = f(\mathbf{x})$. By running $\mathcal{A}$ over D_t , we get a weak hypothesis $g_t : \mathbb{R}^d \rightarrow \{\pm 1\}$ such that $\text{err}_{D_t}(g_t) \leq 1/2 - \beta$ and update $h_{t+1} = P_1(h_t + \beta N_t g_t)$. For some $\Delta' > 0$ determined later, we run the above procedure whenever $N_t \geq \Delta'$ or a weak hypothesis is not returned.

In particular, since $N_t > \text{err}_D(\text{sign}(h_t))$, when $N_t \leq \Delta'$, it must be the case where $\text{err}_D(\text{sign}(h_t)) \leq \Delta'$. Notice that this implies there is a weight function $w_t : \mathbb{R}^d \rightarrow [0, 1/\Delta']$ such that $(D_t)_{\mathbf{x}}(\mathbf{x}) = D_{\mathbf{x}}(\mathbf{x})w_t(\mathbf{x})$.

For each time step t , we have

$$\mathbf{E}_D[P_1(f - h_t)g_t] = \mathbf{E}_D[|P_1(f - h_t)|fg_t] = \mathbf{E}_{D_t}[fg_t]N_t \geq 2\beta N_t.$$

Since $N_t = \mathbf{E}_D[|P_1(f - h_t)|] \geq \mathbf{E}_D[R(f - h_t)]/3$. This gives

$$\begin{aligned} \mathbf{E}_{(\mathbf{x},y) \sim D}[R(f - h_{t+1})] &\leq \mathbf{E}_{(\mathbf{x},y) \sim D}[R(f - h_t)] - 3(\beta N_t)^2 \\ &\leq \mathbf{E}_{(\mathbf{x},y) \sim D}[R(f - h_t)](1 - \beta^2 N_t) \\ &\leq \mathbf{E}_{(\mathbf{x},y) \sim D}[R(f - h_t)](1 - \beta^2 \Delta') \end{aligned}$$

This implies that after $t = \beta^{-2}(\Delta')^{-1} \log(\Delta')^{-1}$ steps, we have $\mathbf{E}_{(\mathbf{x},y) \sim D}[R(f - h_t)] \leq \Delta'$. By the definition of R , this implies $\text{err}_D(\text{sign}(h_t)) \leq \mathbf{E}_{(\mathbf{x},y) \sim D}[R(f - h_t)] \leq \Delta'$.

To conclude the proof, it remains to bound Δ' . When a weak hypothesis is not returned, it must be the case that there is some $h^* \in \mathcal{H}$ such that $\mathbf{E}_{(\mathbf{x},y) \sim D_t}[yh^*] < 2a$ and equivalently $\mathbf{E}_{(\mathbf{x},y) \sim D}[h^*P_1(f - h_t)] < 2aN_t$. This gives

$$N_t = \mathbf{E}_{(\mathbf{x},y) \sim D}[P_1(f - h_t)] \leq 2\text{opt}_D(\mathcal{H}) + 2aN_t,$$

which means $N_t \leq 2\text{opt}_D(\mathcal{H})/(1 - 2a)$. This gives

$$\text{err}_D(\text{sign}(h_t)) = \mathbf{E}_{(\mathbf{x},y) \sim D}[|f - h_t|]/2 \leq \mathbf{E}_{(\mathbf{x},y) \sim D}[P_1(f - h_t)(f - \text{sign}(h_t))]/2 \leq (N_t + \epsilon)/2 \leq \text{opt}_D(\mathcal{H})/(1 - 2a) + \epsilon.$$

Therefore, by setting $\Delta' = \text{opt}_D(\mathcal{H})/(1 - 2a) + \epsilon$, this concludes the proof. $\square$

C Omitted Details from Section 7

C.1 Omitted Proof of Extended Theorem 6 in [KKM13]

In this section, we give the proof of the following extension of Theorem 6 in [KKM13].

Theorem C.1 (Extension of Theorem 6 in [KKM13]). *Let $D_{\mathbf{x}}$ be an isotropic logconcave distribution over $\mathbb{R}^d$ and let $f : \mathbb{R}^d \rightarrow \{0, 1\}$. For every $\epsilon, \delta \in (0, 1)$, there is a pair of polynomials p^l, p^u of degree $\ell = \exp(O(d^4/\delta^4) + d^2 \log^2(d/\epsilon))$, such that*

$$p^l(\mathbf{x}) \leq f(\mathbf{x}) \leq p^u(\mathbf{x}), \quad \mathbf{E}_{\mathbf{x} \sim D_{\mathbf{x}}} [|p^u(\mathbf{x}) - p^l(\mathbf{x})|] \leq \epsilon + 2 \mathbf{Pr}_{\mathbf{x} \sim D_{\mathbf{x}}} [S_{\delta}],$$

where $S_{\delta} = \{\mathbf{u} \in \mathbb{R}^d \mid \exists \mathbf{v} \in \mathbb{R}^d : \|\mathbf{u} - \mathbf{v}\|_2 \leq \delta, f(\mathbf{u}) \neq f(\mathbf{v})\}$.

We will make use of the following facts proved in [KKM13].

Fact C.2. Let $D_{\mathbf{x}}$ be an isotropic logconcave distribution over $\mathbb{R}^d$, for every $R > 0$, $\mathbf{Pr}_{\mathbf{x} \sim D_{\mathbf{x}}}[\|\mathbf{x}\|_2 > R] \leq 2^{O(d)} \exp(-R/32)$.

Fact C.3. Let $f(\mathbf{x}) : \mathbb{R}^d \rightarrow \mathbb{R}$ be c -smooth, which means there is some $c > 0$ such that for every k and for every $\mathbf{u} \in \mathbb{S}^{d-1}$, $\nabla_{\mathbf{u}}^k f$, the k -th order derivative of f in $\mathbf{u}$ direction satisfies $\|\nabla_{\mathbf{u}}^k f\|_{\infty} \leq c^k$. Then for every positive k, N , there is a degree $k2^N$ -polynomial p such that $|(f(\mathbf{x}))^N - p(\mathbf{x})| \leq E(\mathbf{x}) = \sum_{j=0}^{N-1} 2^N (2c \|\mathbf{x}\|_2)^{2^j k} / (2^j k)!$. Furthermore, for any isotropic logconcave distribution $D_{\mathbf{x}}$, $\|E(\mathbf{x})\|_{D_{\mathbf{x}},1} \leq 2^N 2^{O(d)} 2^{-k}$.

Proof of Theorem 7.11. We construct the polynomial p^u and p^l can be constructed by symmetry. Recall that $S_{\delta} = \{\mathbf{u} \in \mathbb{R}^d \mid \exists \mathbf{v} \in \mathbb{R}^d : \|\mathbf{u} - \mathbf{v}\|_2 \leq \delta, f(\mathbf{u}) \neq f(\mathbf{v})\}$. The construction of p^u will depend on several parameters that will be determined later: $c > 0$, a small constant, $k, N \in \mathbb{Z}_+$, large even integers, and $R > 0$ a suitably large number.

For $c > 0$, we define $\rho_c(\mathbf{x}) = \prod_{i=1}^d \text{sinc}(c\mathbf{x}_i/\sqrt{d})$, where $\text{sinc}(z) = \sin(z)/z, z \in \mathbb{R}$. By [KKM13], $\rho_c(\mathbf{x})$ is c -smooth, and when $\|\mathbf{x}\|_2 \leq c^{-1}\sqrt{d}$, the $\rho_c = \exp(-\Theta(c^2 \|\mathbf{x}\|_2^2/d))$. For $N \in \mathbb{Z}_+$, we define $r(\mathbf{x}) = C\rho_c(\mathbf{x})^N$, where $C > 0$ is a constant such that $\int_{\mathbb{R}^d} r(\mathbf{x}) = 1$. By [KKM13], $C = O(c^d N^{d/2})$. Follow the construction of [KKM13], we choose $c = 1/256$.

For $R > 0$, we define $f^*(\mathbf{x}) = 1$ if $\|\mathbf{x}\|_2 < R + \delta$ and $\mathbf{x} \in S_{\delta} \cup \{\mathbf{x} \mid f(\mathbf{x}) = 1\}$ and $f^*(\mathbf{x}) = 0$ otherwise. We define $s(\mathbf{x}) := r * f^*$.

$$s(\mathbf{x}) = \int_{\mathbb{R}^d} f^*(\mathbf{u}) r(\mathbf{x} - \mathbf{u}) d\mathbf{u}.$$

We notice that $s(\mathbf{x}) \geq 0$, for every $\mathbf{x}$. For every $\mathbf{x}$ with $\|\mathbf{x}\|_2 \leq R$ and $f(\mathbf{x}) = 1$, we have

$$s(\mathbf{x}) = \int_{\mathbf{u}: \|\mathbf{u}-\mathbf{x}\|_2 < \delta} f^*(\mathbf{u}) r(\mathbf{x} - \mathbf{u}) d\mathbf{u} + \int_{\mathbf{u}: \|\mathbf{u}-\mathbf{x}\|_2 \geq \delta} f^*(\mathbf{u}) r(\mathbf{x} - \mathbf{u}) d\mathbf{u}$$

Follow [KKM13], we choose $N = \max\{O(d^4/\delta^4), O(d^2 \log^2(d/\epsilon))\}$ as an even integer so that $\int_{\mathbf{u}: \|\mathbf{u}\|_2 > \delta} r(\mathbf{u}) d\mathbf{u} < \epsilon/16$. Such a choice implies $\int_{\mathbf{u}: \|\mathbf{u}-\mathbf{x}\|_2 < \delta} r(\mathbf{x} - \mathbf{u}) d\mathbf{u} > 1 - \epsilon/16$. We claim that for every $\mathbf{u}$ that is δ close to $\mathbf{x}$, $f^*(\mathbf{u}) = 1$. For any such $\mathbf{u}$, either $f(\mathbf{u}) = 1$ or $\mathbf{u} \in S_{\delta}$. Since $\|\mathbf{u}\| \leq R + \delta$, by definition of f^* , we have $f^*(\mathbf{u}) = 1$. This implies that $s(\mathbf{x}) \geq 1 - \epsilon/16$ if $\|\mathbf{x}\| \leq R$ and $f(\mathbf{x}) = 1$. So, for every $\mathbf{x}$ with $\|\mathbf{x}\| \leq R$, we have $f(\mathbf{x}) \leq s(\mathbf{x}) + \epsilon/16$.

We next design a polynomial $p(\mathbf{x})$ to approximate $s(\mathbf{x})$. Since $\rho_c(\mathbf{x})$ is c -smooth, by Fact C.3, for every $\mathbf{u} \in \mathbb{R}^d$, there is a polynomial $p_{\mathbf{u}}(\mathbf{x})$ with degree $k2^N$ such that

$$|p_{\mathbf{u}}(\mathbf{x}) - r(\mathbf{x} + \mathbf{u})| \leq CE(\mathbf{x}).$$

Define polynomial $p(\mathbf{x}) = \int_{\mathbb{B}(R+\delta)} p_{-\mathbf{u}}(\mathbf{x}) f^*(\mathbf{u}) d\mathbf{u}$ and we have

$$\begin{aligned} |p(\mathbf{x}) - s(\mathbf{x})| &\leq \left| \int_{\mathbb{B}(R+\delta)} (p_{-\mathbf{u}}(\mathbf{x}) - r(\mathbf{x} - \mathbf{u})) f^*(\mathbf{u}) d\mathbf{u} \right| + \left| \int_{\mathbb{R}^d \setminus \mathbb{B}(R+\delta)} r(\mathbf{x} - \mathbf{u}) f^*(\mathbf{u}) d\mathbf{u} \right| \\ &\leq CE(\mathbf{x})(R)^d. \end{aligned}$$

By choosing $k = O(N + d \log N + d \log R + \log(1/\epsilon))$, we have $\|CR^d E(\mathbf{x})\|_{D_{\mathbf{x}},1} \leq \epsilon/16$.

We choose $P(\mathbf{x}) := p(\mathbf{x}) + O(R)^d CE(\mathbf{x}) + \epsilon/16 + (\|\mathbf{x}\|/R)^{2t}$ for some $t > 0$ that will be chosen later. If $\mathbf{x} \in \mathbb{B}(R)$, then we have $P(\mathbf{x}) \geq s(\mathbf{x}) + \epsilon/16 \geq f(\mathbf{x})$. If $\mathbf{x} \notin \mathbb{B}(R)$, $P(\mathbf{x}) \geq 1$. So, $p^u := P(\mathbf{x}) \geq f(\mathbf{x})$ for every $\mathbf{x} \in \mathbb{R}^d$.

It remains to upper bound $\|p^u - f\|_{D_{\mathbf{x},1}}$. We have

$$\|p^u - f\|_{D_{\mathbf{x},1}} \leq \|s - f\|_{D_{\mathbf{x},1}} + O(R)^d C \|E(\mathbf{x})\|_{D_{\mathbf{x},1}} + \epsilon/16 + \|(\|\mathbf{x}\|/R)^{2t}\|_{D_{\mathbf{x},1}}.$$

We notice that unless $\|\mathbf{x}\| > R$ or $\mathbf{x} \in S_\delta$, we have $|s(\mathbf{x}) - f(\mathbf{x})| < \epsilon/16$. Otherwise $|s(\mathbf{x}) - f(\mathbf{x})| \leq 1$. This implies that $\|s - f\|_{D_{\mathbf{x},1}} \leq \Pr_{\mathbf{x} \in D_{\mathbf{x}}}[S_\delta] + \Pr_{\mathbf{x} \in D_{\mathbf{x}}}[\|\mathbf{x}\| > R] + \epsilon/16$. We choose $R = O(d + \log(1/\epsilon))$ and $t = O(\log(1/\epsilon))$ to make $\Pr_{\mathbf{x} \in D_{\mathbf{x}}}[\|\mathbf{x}\| > R] < \epsilon/16$ and $\|(\|\mathbf{x}\|/R)^{2t}\|_{D_{\mathbf{x},1}} < \epsilon/16$. This gives

$$\|p^u - f\|_{D_{\mathbf{x},1}} \leq \Pr_{\mathbf{x} \in D_{\mathbf{x}}}[S_\delta] + \epsilon/2.$$

This concludes the proof. $\square$

C.2 Omitted Details from Theorem 7.13

In this section, we provide omitted details from Theorem 7.13. To do this, we will show that under any isotropic logconcave distribution, the probability that a point is δ -close to the decision boundary is small. Such a result can be derived from the work [Lot15].

Fact C.4 (Theorem 1.1 in [Lot15]). *Let $p(\mathbf{x})$ be a degree m polynomial in $\mathbb{R}^d$ and let $S = \{\mathbf{x} \in \mathbb{R}^d \mid p(\mathbf{x}) = 0\}$. Let D' be the uniform distribution over a ball $\mathbb{B}^{d-1}(\mathbf{u}, R)$, then for every $\delta > 0$*

$$\Pr_{\mathbf{x} \sim D'}[\inf_{\mathbf{v} \in S} \|\mathbf{x} - \mathbf{v}\| < \delta] < 4 \sum_{i=0}^{d-1} \binom{d}{1+i} (2m\delta/R)^{1+i} (1 + \delta/R)^{d-1-i} \leq 8dm2^{d-1}(m+1)^{d-1}\delta/R.$$

Lemma C.5. *Let $p(\mathbf{x})$ be a degree m polynomial in $\mathbb{R}^d$ and let $D_{\mathbf{x}}$ be an isotropic logconcave distribution over $\mathbb{R}^d$. Let $S_\delta = \{\mathbf{u} \in \mathbb{R}^d \mid \exists \mathbf{v} \in \mathbb{R}^d : \|\mathbf{u} - \mathbf{v}\|_2 \leq \delta, \text{sign}(p(\mathbf{u})) \neq \text{sign}(p(\mathbf{v}))\}$, then $\Pr_{D_{\mathbf{x}}}[S_\delta] \leq (Cd)^d m \delta \log(1/\delta)^{m-1}$.*

Proof of Lemma C.5. By Theorem 1.1 in [Lot15], we have for any fixed $R > 0$, it holds

$$\begin{aligned} \Pr_{\mathbf{x} \sim D_{\mathbf{x}}}[\mathbf{x} \in S_\delta] &\leq \Pr_{\mathbf{x} \sim D_{\mathbf{x}}}[\|\mathbf{x}\| > R] + \Pr_{\mathbf{x} \sim D_{\mathbf{x}}}[\|\mathbf{x}\| < R, \mathbf{x} \in S_\delta] \leq \Pr_{\mathbf{x} \sim D_{\mathbf{x}}}[\|\mathbf{x}\| > R] + \sup_{\mathbf{x} \in \mathbb{R}^d} D_{\mathbf{x}}[\mathbf{x}] \text{Vol}(S_\delta \cap \mathbb{B}^{d-1}(R)) \\ &\leq \Pr_{\mathbf{x} \sim D_{\mathbf{x}}}[\|\mathbf{x}\| > R] + (CR)^{d-1}(m+1)^{d-1}dm\delta \end{aligned}$$

By choosing $R = \Theta(\sqrt{d} \log(1/\delta))$, we get $\Pr_{\mathbf{x} \sim D_{\mathbf{x}}}[\|\mathbf{x}\| > R] < \delta$, which gives $\Pr_{D_{\mathbf{x}}}[S_\delta] \leq (Cd)^d m \delta \log(1/\delta)^{m-1}$ for some universal constant $C > 1$. $\square$

Notice that to make $\Pr_{\mathbf{x} \sim D_{\mathbf{x}}}[S_\delta] < \epsilon$, Lemma C.5 implies that we only need to choose $\delta = \epsilon/((Cd)^d m \log(1/\epsilon)^m)$.