Iterative Chow Filtering for Learning with Distribution Shift

Gautam Chandrasekaran[*] UT Austin
Georgios Gkrinias[†] UT Austin
Adam R. Klivans[‡] UT Austin
Konstantinos Stavropoulos[§] Institute for Advanced Study
Arsen Vasilyan[¶] Aarhus University

Abstract

Recent work due to Goel et al. gave the first efficient algorithms for learning with distribution shift in the challenging PQ framework, where a learner receives labeled training examples and unlabeled test examples and must make correct predictions on the test set but is allowed to abstain from predicting on out-of-distribution points. Their results rely on $\mathcal{L}_2$ sandwiching approximations, a strong requirement that leads to poor bounds for several basic function classes such as DNF formulas. Here we show that the weaker notion of $\mathcal{L}_1$ sandwiching suffices for efficient PQ learning. As a consequence, we obtain the first quasipolynomial-time PQ learning algorithm for DNFs under the uniform distribution and essentially match the guarantees known for ordinary PAC learning. More broadly, our bounds provide exponential improvements for several classes including constant depth circuits and constant degree polynomial threshold functions. Our main technical ingredient is *Iterative Chow Filtering*, a new procedure that uses low-degree Chow parameters to identify and remove test points incompatible with the training distribution.

1 Introduction

A central challenge in modern machine learning is to obtain reliability in the presence of distribution shift. Arguably the most well-studied formalization of the problem is domain adaptation [BDBCP06, BCK+07, MMR09, BDBC+10, DLLP10], where a learner observes labeled examples from a training distribution, unlabeled examples from a test distribution, and must output a hypothesis with low test error (despite never observing test labels). In the simplest version (often referred to as covariate shift), the training and test labels are generated by the same unknown ground truth function f^*. Despite more than two decades of work, most known guarantees in this literature are information-theoretic and are not accompanied by efficient algorithms.

This gap has motivated a recent line of work on establishing efficient algorithms for learning with distribution shift in the TDS model due to Klivans et al. [KSV24a] and the more fine-grained PQ framework of Goldwasser et al. [GKKM20]. Most relevant to our work here is PQ learning, in which a learner is allowed to abstain on test points that appear out of distribution but must be accurate on the points on which it does make predictions. Formally, a PQ learner outputs a hypothesis h and a selector s such that, under the test distribution, the probability that $s(x) = 1$ and $h(x) \neq f^*(x)$ is small, while under the training distribution, the probability that $s(x) = 0$ is small. Thus, a PQ learner

[*] gautamc@cs.utexas.edu
[†] gkrinias@cs.utexas.edu
[‡] klivans@cs.utexas.edu
[§] kstavrop@amias.ias.edu
[¶] vasilyan@cs.au.dk

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

must provide accurate predictions on the portion of the test distribution that remains sufficiently similar to the training distribution.

Goel et al. [GSSV24] obtained the first efficient PQ learning algorithms for a variety of function classes, but several basic cases remained open. For example, no subexponential-time PQ algorithm was known for learning DNF formulas with respect to the uniform distribution on the hypercube, one of the most heavily studied problems in computational learning theory. This is in sharp contrast with the standard realizable PAC setting where the seminal result of Linial, Mansour, and Nisan [LMN93] gives a quasipolynomial-time algorithm for learning constant-depth circuits under the uniform distribution via low-degree polynomial approximations.

The obstacle in the PQ setting is that known algorithms require a stronger form of polynomial approximation. The result of [GSSV24], while also based on polynomial approximation, relies on $\mathcal{L}_2$ sandwiching: for the target function f^*, one must construct polynomials $p_{\mathrm{down}}, p_{\mathrm{up}}$ satisfying $p_{\mathrm{down}}(x) \le f^*(x) \le p_{\mathrm{up}}(x)$ pointwise, and such that the expected squared gap $(p_{\mathrm{up}}(x) - p_{\mathrm{down}}(x))^2$ under the training distribution is small. For DNFs, however, the best known $\mathcal{L}_2$ sandwiching bounds are exponentially worse than the corresponding bounds for ordinary polynomial approximation, with degree bounds depending polynomially on the size.

In this work, we show that $\mathcal{L}_1$ sandwiching suffices for efficient PQ learning. Here, the pointwise sandwiching requirement is the same, but the approximation guarantee only asks that the expected gap $p_{\mathrm{up}}(x) - p_{\mathrm{down}}(x)$ be small, rather than its square. This distinction is crucial: by the celebrated result of Braverman [Bra08], size-s constant-depth circuits have $\mathcal{L}_1$ sandwiching degree $\mathrm{poly}(\log s)$, matching the qualitative behavior of standard polynomial approximation. This essentially closes the gap between the known computational complexity of PAC learning and PQ learning for DNFs under the uniform distribution, yielding the first quasipolynomial-time PQ learning algorithm in this setting. More broadly, our result implies exponential runtime improvements for several well-studied function classes (see Table 1).

Our main technical ingredient is a new outlier-filtering procedure, which we call *Iterative Chow Filtering* (ICF). The procedure combines ideas from filtering methods in robust learning and learning with distribution shift with the classical notion of Chow parameters, allowing us to convert $\mathcal{L}_1$ sandwiching into the selective, pointwise coverage guarantees required by PQ learning.

1.1 Our Results

Our main result is a new computationally efficient algorithm for PQ learning that applies to any class admitting low-degree $\mathcal{L}_1$ sandwiching polynomials. In Table 1, we present a number of concrete applications of our main result by combining known sandwiching-degree bounds for the corresponding classes.

	Concept class $\mathcal{C}$	**Training Marginal**	**Our Work**	**Prior Work**
1	Degree-2 PTFs	Uniform on $\{\pm 1\}^d$	$\mathrm{poly}(d)$	$2^{O(d)}$
2	Constant-Degree PTFs	Standard Gaussian	$\mathrm{poly}(d)$	$2^{\mathrm{poly}(d)}$
3	Functions of a Constant Number of Halfspaces	Isotropic Log-Concave	$\mathrm{poly}(d)$	None
4	DNFs of Size s	Uniform on $\{\pm 1\}^d$	$d^{O(\log^2(s))}$	$d^{\widetilde{O}(\sqrt{s})}$
5	Circuits of Size s, Depth t	Uniform on $\{\pm 1\}^d$	$d^{(\log(s))^{O(t)}}$	$d^{\sqrt{s}\cdot(\log(s))^{O(t)}}$ only for formulas

Table 1: Runtime comparison between our PQ learning algorithm and the best previous results for PQ learning various classes up to constant error and rejection rate, even when both the training and the test labels are given by the same (unknown) function $f^* \in \mathcal{C}$, i.e., when there is only covariate shift.

More formally, we consider a feature space $\mathcal{X} \subseteq \mathbb{R}^d$ and binary labels $\mathcal{Y} = \{0, 1\}$. The learner has access to labeled samples from a training distribution $\mathcal{D}^{\mathrm{train}}$ over $\mathcal{X} \times \mathcal{Y}$, as well as unlabeled samples from the marginal $\mathcal{D}^{\mathrm{test}}_{\mathcal{X}}$ of a test distribution $\mathcal{D}^{\mathrm{test}}$. In the general case captured by our

results, neither distribution need be realizable. Since the learner has no access to test labels, the standard benchmark from domain adaptation [BDBCP06, BCK+07] is the joint optimal error

$$\lambda = \lambda(\mathcal{D}^{\text{train}}, \mathcal{D}^{\text{test}}; \mathcal{C}) = \min_{f \in \mathcal{C}} \Big(\text{err}_{\text{train}}(f) + \text{err}_{\text{test}}(f) \Big), \tag{1}$$

where $\text{err}_{\text{train}}(f) = \mathbf{Pr}_{(x,y)\sim\mathcal{D}^{\text{train}}}[f(x) \neq y]$ and $\text{err}_{\text{test}}(f)$ is defined analogously.

It will be useful to separate this benchmark into its training and test components. Let

$$\mathcal{C}_\lambda = \{f \in \mathcal{C} : \text{err}_{\text{train}}(f) + \text{err}_{\text{test}}(f) = \lambda\}.$$

We define

$$\lambda_{\text{train}} = \min_{f \in \mathcal{C}_\lambda} \text{err}_{\text{train}}(f) \qquad \text{and} \qquad \lambda_{\text{test}} = \lambda - \lambda_{\text{train}}.$$

Thus, λ_{train} and λ_{test} correspond to the training and test errors of a joint optimizer, with ties broken in favor of smaller training error.

The formal definition of PQ learning of a concept class $\mathcal{C} \subseteq \{\mathcal{X} \to \mathcal{Y}\}$ with respect to a distribution $\mathcal{D}$ over $\mathcal{X}$ is as follows.

Definition 1.1 (PQ Learning, modification of [GKKM20])**.** Let $\epsilon, \delta, \eta \in (0,1)$. An algorithm $\mathcal{A}$ is an (ϵ, δ, η)-PQ learner for $\mathcal{C}$ with respect to $\mathcal{D}$ if the following holds whenever $\mathcal{D}^{\text{train}}_{\mathcal{X}} = \mathcal{D}$. Given sufficiently many labeled samples from $\mathcal{D}^{\text{train}}$ and sufficiently many unlabeled samples from $\mathcal{D}^{\text{test}}_{\mathcal{X}}$, the algorithm outputs a classifier $h\colon \mathcal{X} \to \{0,1\}$ and a selector $s\colon \mathcal{X} \to \{0,1\}$ such that, with probability at least $1-\delta$:

1. *Selective accuracy.* The error on accepted test points satisfies

$$\mathbf{Pr}_{(x,y)\sim\mathcal{D}^{\text{test}}}[h(x) \neq y \wedge s(x) = 1] \leq \lambda_{\text{test}} + \frac{1}{\eta}\Big(\lambda_{\text{train}} + \min_{f\in\mathcal{C}} \text{err}_{\text{train}}(f)\Big) + \epsilon.$$

2. *Rejection rate.* The rejection probability under the training marginal satisfies

$$\mathbf{Pr}_{x\sim\mathcal{D}^{\text{train}}_{\mathcal{X}}}[s(x) = 0] \leq \eta.$$

Remark 1.1. *[GKKM20] showed that there are simple examples where any PQ learner must either have accuracy $\Omega(\sqrt{\lambda_{\text{train}}})$ or rejection rate $\Omega(\sqrt{\lambda_{\text{train}}})$.*

We now provide a definition for the sandwiching degree of the concept class $\mathcal{C}$ with respect to $\mathcal{D}$.

Definition 1.2 ($\mathcal{L}_1$ Sandwiching Degree)**.** We let $\mathcal{L}_1(\mathcal{C}, \mathcal{D}, \epsilon)$ denote the minimum natural number ℓ such that for any $f \in \mathcal{C}$, there are polynomials $p_{\text{up}}, p_{\text{down}}$ of degree at most ℓ such that:

$$p_{\text{down}}(x) \leq f(x) \leq p_{\text{up}}(x) \text{ for all } x \in \mathcal{X} \quad \text{and} \quad \mathbf{E}_{x\sim\mathcal{D}}[p_{\text{up}}(x) - p_{\text{down}}(x)] \leq \epsilon\,.$$

Our main result is the following theorem, whose formal version can be found in Theorem 4.1.

Theorem 1.1 (PQ Learning via $\mathcal{L}_1$ Sandwiching)**.** *Let any $\epsilon, \eta, \delta \in (0,1)$ and suppose $\mathcal{D}$ is hypercontractive. Let $\ell = \mathcal{L}_1(\mathcal{C}, \mathcal{D}, \epsilon\eta/c)$ for some appropriately large universal constant $c \geq 1$. There is an (ϵ, δ, η)-PQ learner for $\mathcal{C}$ with respect to $\mathcal{D}$ with sample complexity and runtime $\text{poly}(d^\ell, 1/\epsilon, 1/\eta, \log(1/\delta)^\ell)$.*

Relaxed Requirements. As mentioned earlier, our main result yields significant runtime improvements for several basic classes, including those in Table 1. This is because $\mathcal{L}_1$ sandwiching is a relaxed requirement compared to $\mathcal{L}_2$ sandwiching, but also because our result imposes no coefficient bounds on the sandwiching polynomials $p_{\text{up}}, p_{\text{down}}$. The latter improvement is important for the class of functions of a constant number of halfspaces under any fixed isotropic log-concave training distribution, where the available sandwiching approximators [KKM13] are not known to have bounded coefficients. In contrast, [GSSV24] require both an $\mathcal{L}_2$ sandwiching guarantee and explicit bounds on the coefficients of the sandwiching polynomials.[6]

[6]Note that both [GSSV24] and our main result require the additional assumption that the training marginal is hypercontractive. This condition is important for the concentration of polynomials and is satisfied by a wide range of distributions, including any log-concave distribution, as well as any product distribution over the Boolean hypercube.

Improved Error Guarantees. Our algorithm also achieves sharper error guarantees on the accepted portion of the test distribution, expressed in terms of the label-drift parameters λ_{train} and λ_{test}, together with the optimal training error. Compared with the previous best result of [GSSV24], which gives an $O(\lambda/\eta)$-type guarantee with a constant-factor loss in the leading term, our analysis obtains the sharpest known dependence on these parameters (as stated in Definition 1.1). In particular, this improves the best known error guarantees even for the fundamental case of PQ learning linear separators under Gaussian training marginals.

Tolerant TDS Learning. A related framework is TDS learning, where the learner may abstain on the entire test distribution, provided it detects the presence of distribution shift (it is easy to see that PQ learning implies TDS learning). Chandrasekaran et al. [CKK+24] showed that $\mathcal{L}_1$ sandwiching suffices for efficient TDS learning. Their result, however, does not extend to tolerant TDS learning, where the learner is required to accept whenever the training and test marginals are close in total variation distance. Our PQ learning theorem implies that $\mathcal{L}_1$ sandwiching suffices for tolerant TDS learning, but we can obtain even sharper error bounds by giving a direct white-box analysis of our algorithm for the tolerant setting (see Appendix D).

1.2 Our Techniques

We prove our main PQ-learning result in two steps. First, using the training data, we learn a hypothesis $\hat{f}\colon \mathcal{X} \to \{0,1\}$ with small training error via $\mathcal{L}_1$ polynomial regression. This is possible because the class $\mathcal{C}$ admits low-degree $\mathcal{L}_1$ sandwiching polynomials. In fact, the classical work of [KKMS08] shows that even the weaker existence of approximating polynomials suffices for efficient learning with near-optimal training error via $\mathcal{L}_1$ regression.

The main challenge is to identify the part of the test distribution on which $\hat{f}$ can be trusted. We construct a selector $s\colon \mathcal{X} \to \{0,1\}$ that accepts test points on which $\hat{f}$ has small error, while rejecting only a small fraction of the training distribution. The key ingredient is an error-transfer lemma: we identify a condition under which the training error of $\hat{f}$ transfers to the selected test distribution, and then design an efficient algorithm that enforces this condition.

Existing transfer lemmas for learning under distribution shift are insufficient for our purposes. They either require stronger approximation guarantees, such as $\mathcal{L}_2$ sandwiching rather than $\mathcal{L}_1$ sandwiching, or they do not yield an efficient method for constructing the selector.

The closest prior result is the following lemma of [CKK+24], proved in the context of TDS learning.

Lemma 1.1 (Informal, from [CKK+24])**.** *Suppose that $\mathcal{C}$ has $\mathcal{L}_1$ sandwiching polynomials of degree ℓ with respect to the training distribution, and let $\hat{f}\colon \mathcal{X} \to \{0,1\}$. Suppose also that for every monomial χ of degree at most ℓ,*

$$\mathop{\mathbf{E}}_{\mathcal{D}^{\mathrm{train}}}[\hat{f}(x)\chi(x)] \approx \mathop{\mathbf{E}}_{\mathcal{D}^{\mathrm{test}}}[\hat{f}(x)\chi(x)] \quad \textit{and} \quad \mathop{\mathbf{E}}_{\mathcal{D}^{\mathrm{train}}}[\chi(x)] \approx \mathop{\mathbf{E}}_{\mathcal{D}^{\mathrm{test}}}[\chi(x)].$$

Then, the test error of $\hat{f}$ is approximately upper bounded by $\lambda(\mathcal{C}) + \mathrm{err}_{\mathrm{train}}(\hat{f})$.

The quantities $\mathbf{E}[\hat{f}(x)\chi(x)]$ are the so-called Chow parameters of $\hat{f}$ [Cho61, OS08, DK19]. Thus, this lemma shows that, under $\mathcal{L}_1$ sandwiching, matching the low-degree Chow parameters of $\hat{f}$ and of the constant function suffices for error transfer in the (weaker) model of TDS learning, where a learner can abstain on the entire test set.

However, requiring additive Chow matching between a *selection* of the test distribution and the training distribution is too much to hope for. Consider $\mathcal{X} = \mathbb{R}$, $\hat{f}(x) = \mathbf{1}\{x \geq 0\}$, $\mathcal{D}^{\mathrm{train}} = \mathcal{U}[-1,1]$, and $\mathcal{D}^{\mathrm{test}} = \mathcal{U}[0,1]$. Any selector s that rejects only a small fraction of the training distribution must accept almost all of $[0,1]$. Hence,

$$\mathop{\mathbf{E}}_{\mathcal{D}^{\mathrm{test}}}[\hat{f}(x)s(x)] \approx 1, \qquad \text{whereas} \qquad \mathop{\mathbf{E}}_{\mathcal{D}^{\mathrm{train}}}[\hat{f}(x)] = \frac{1}{2}.$$

Thus, no valid selector can generally make the selected test distribution Chow-match the training distribution.

We overcome this obstruction with a multiplicative Chow-transfer lemma.

Lemma 1.2 (Informal, see Lemma 4.1)**.** *Suppose that $\mathcal{C}$ has $\mathcal{L}_1$ sandwiching polynomials of degree ℓ with respect to the training distribution, and let $\hat{f}\colon \mathcal{X} \to \{0,1\}$. Also suppose that, for some $R > 1$, every nonnegative polynomial p of degree at most ℓ satisfies*

$$\mathop{\mathbf{E}}_{\mathcal{D}^{\mathrm{test}}}[\hat{f}(x)p(x)] \lesssim R \cdot \mathop{\mathbf{E}}_{\mathcal{D}^{\mathrm{train}}}[\hat{f}(x)p(x)] \quad \text{and} \quad \mathop{\mathbf{E}}_{\mathcal{D}^{\mathrm{test}}}[(1-\hat{f}(x))p(x)] \lesssim R \cdot \mathop{\mathbf{E}}_{\mathcal{D}^{\mathrm{train}}}[(1-\hat{f}(x))p(x)].$$

Then, the test error of $\hat{f}$ is approximately upper bounded by $R \cdot \left(\lambda(\mathcal{C}) + \mathrm{err}_{\mathrm{train}}(\hat{f})\right)$.

Instead of requiring additive matching of Chow parameters, this lemma only requires multiplicative domination of the Chow-type expectations of $\hat{f}$ and $1-\hat{f}$ against nonnegative low-degree polynomial tests. We enforce this condition on the selected test distribution using an efficient algorithm, which we call Iterative Chow Filtering (ICF).

ICF proceeds by repeatedly searching for a violated Chow-domination inequality. Suppose, for example, that there is a polynomial p such that the expectation of $\hat{f}(x)p(x)$ under the current test distribution is much larger than the expectation of $\hat{f}(x)|p(x)|$ under the training distribution. Then, for an appropriate threshold τ, the event

$$\hat{f}(x)|p(x)| > \tau$$

has significantly larger mass under the current test distribution than under the training distribution. We reject all points in this event and recurse, applying the same argument symmetrically with $1-\hat{f}$ in place of $\hat{f}$. Each step removes disproportionately more test mass than training mass, so the total rejected training mass remains small.

Iterative filtering is a standard paradigm in robust learning [KLS09, DKS18, DKK+19a, DKK+19b, KSV25, KSTV26] and has also appeared in prior work on PQ learning [GSSV24]. The key difference is that all previous filtering procedures are based on discrepancies in polynomial expectations of only the marginal distributions. Our procedure here instead filters using discrepancies in Chow expectations, namely polynomial expectations weighted by the learned Boolean hypothesis $\hat{f}$ or by $1-\hat{f}$. This Chow-based filtering step is what allows us to obtain PQ learning from $\mathcal{L}_1$ sandwiching alone.

1.3 Related Work

PQ Learning. The framework of PQ learning was introduced by [GKKM20]. In its original formulation, they showed that, without assumptions on the training distribution, an agnostic empirical risk minimization oracle yields PQ learning via an oracle-efficient reduction. This was later strengthened by [KK21], who showed that distribution-free PQ learning is equivalent to the reliable agnostic learning framework of [KKM12]. Reliable agnostic learning lies between realizable PAC learning and agnostic learning: it is easier than agnostic learning but harder than realizable PAC learning. However, reliable agnostic learning is already hard for very basic classes. For example, reliable agnostic learning of conjunctions is as hard as PAC learning DNFs, a long-standing open problem widely believed to be computationally intractable.

The recent work of [GSSV24] showed that, in distribution-specific settings where the *training distribution* is fixed to be, for example, the standard Gaussian distribution or the uniform distribution on the Boolean hypercube, PQ learning is tractable for a broad range of classes. In particular, they showed that any class admitting low-degree $\mathcal{L}_2$ sandwiching polynomials with respect to the training distribution can be efficiently learned in the PQ model. Their result captures several classes, especially in light of recent $\mathcal{L}_2$ sandwiching degree bounds of [KSV26]. Nevertheless, prior to our work, no subexponential-time PQ learning algorithm was known for DNFs when the training distribution is uniform over the Boolean hypercube. More broadly, constant-depth circuits are known to have quasipolynomial $\mathcal{L}_1$ sandwiching degree with respect to the uniform training distribution [Baz09, Raz09, Bra08, Tal17, HH23], but the best known bound on their $\mathcal{L}_2$ sandwiching degree is exponentially worse, namely polynomial in the circuit size [OS03, KSV24a].

Sandwiching Polynomials. The notion of sandwiching polynomials originates in the pseudorandomness literature; see [HH23] and references therein. In this context, most known results concern the uniform distribution over the Boolean hypercube [Baz09, Raz09, Bra08, Tal17, DGJ+10, DKN10],

although sandwiching bounds were also shown for certain continuous distributions [GOWZ10, Kan11, KKM13].

Recent work in learning theory has highlighted the role of sandwiching approximation in challenging learning settings, including testable learning [RV23, GKK23, STW24], learning with distribution shift [KSV24a, GSSV24], and learning under contamination [KSV25, KSTV26]. This has motivated further work establishing sandwiching-degree bounds for broader concept classes, under broader families of distributions, and with higher-order error guarantees, namely $\mathcal{L}_r$ sandwiching for $r > 1$ [KSV26].

Nevertheless, substantial gaps remain between the best known $\mathcal{L}_1$ and $\mathcal{L}_2$ sandwiching-degree bounds for several basic classes. For constant-depth circuits under the uniform distribution, the best known bounds exhibit an exponential separation. For degree-2 polynomial threshold functions under the uniform distribution on the Boolean hypercube, and for constant-degree polynomial threshold functions under the Gaussian distribution, the $\mathcal{L}_1$ sandwiching degree is polynomial in the inverse approximation error, whereas the best known $\mathcal{L}_2$ sandwiching-degree bounds scale polynomially with the dimension d [KSV26].

TDS Learning. The framework of TDS learning was introduced by [KSV24a]. PQ learning is stronger than TDS learning: in particular, it implies tolerant TDS learning, a stronger variant in which the learner is guaranteed to accept under sufficiently small distribution shift. Although PQ learning was defined before TDS learning, the first distribution-specific efficient algorithms for learning with distribution shift were developed in the TDS framework.

The first general result for TDS learning showed that $\mathcal{L}_2$ sandwiching implies efficient TDS learning [KSV24a]. This was later strengthened by [CKK+24], who showed that $\mathcal{L}_1$ sandwiching already suffices. Our work shows that $\mathcal{L}_1$ sandwiching also suffices for tolerant TDS learning and PQ learning. Some positive results for special concept classes in the TDS model have not yet been extended to PQ learning, or even to tolerant TDS learning [KSV24a, KSV24b, CKK+24]. Nevertheless, existing TDS learning results have so far often admitted extensions to the PQ setting through suitable filtering procedures. Our ICF algorithm provides an additional technical tool for carrying out such extensions.

Chow Parameters. The notion of Chow parameters originates with Chow's characterization of Boolean threshold functions [Cho61]. Since then, Chow parameters have played a role in several areas of theoretical computer science, including circuit complexity and threshold logic [Win71, GHR92], game theory and the design of voting systems [Ban65, DS79], and learning theory [BDD98, BDJ+98, OS08, CKSV25]. A common use of Chow parameters is through structural theorems showing that certain Boolean-valued functions are determined, either exactly or robustly, by their low-degree Chow parameters [Cho61, OS08, DDFS14, DK19].

More recently, work in testable learning [GKSV25] and learning with distribution shift [CKK+24] has highlighted a different role for Chow-type expectations: not merely as parameters that characterize the underlying function, but as a tool for certifying the performance and reliability of learning algorithms. In this work, we show that such Chow-type expectations can also be used to obtain state-of-the-art guarantees in the PQ learning model.

2 Preliminaries

Notation. Let $\mathbb{N} = \{0, 1, \dots\}$ denote the set of natural numbers and $\mathbb{N}^* = \mathbb{N} \setminus \{0\}$. For any $d, \ell \in \mathbb{N}$, let $\mathcal{P}_{d,\ell}$ denote the set of d-variate polynomials, whose degree is at most ℓ. For any multiset S, we let $\mathcal{U}(S)$ denote the uniform distribution on S. For any real-valued function f, we denote $\bar{f} = 1 - f$.

Definition 2.1 (Hypercontractivity)**.** Let any $A \geq 1$. A distribution $\mathcal{D}$ on $\mathbb{R}^d$ is said to be A-hypercontractive (with respect to polynomials) if for any polynomial $p\colon \mathbb{R}^d \to \mathbb{R}$ and $t \geq 2$:

1. $\mathbf{E}_{x\sim\mathcal{D}}[|p(x)|^t] \leq (At)^{t \deg(p)} \mathbf{E}_{x\sim\mathcal{D}}[|p(x)|]^t$ and
2. the absolute value of any degree-1 monomial has finite expectation under $\mathcal{D}$.

Hypercontractive distributions are a wide class of probability distributions, which includes Gaussians, all log-concave distributions on $\mathbb{R}^d$ (see [Bob01]), as well as product distributions over $\{\pm 1\}^d$.

3 Iterative Chow Filtering

In this section, we present the ICF algorithm (Algorithm 1), which is the main tool for our results in PQ learning and tolerant TDS learning. The algorithm takes as input a classifier $\hat{f}\colon \mathcal{X} \to \{0,1\}$, a sample S' from a test marginal $\mathcal{D}'$ and a reference sample from a training marginal $\mathcal{D}$, and outputs a selector $s\colon \mathcal{X} \to \{0,1\}$. The goal is that the selector keeps test points whose low-degree polynomial behavior, restricted to the classifier-relevant regions specified by $\mathcal{F} = \{\hat{f}, 1-\hat{f}\}$, is consistent with $\mathcal{D}$, and rejects points that seem to be outliers. Specifically, ICF tries to enforce the following:

1. For each $f \in \mathcal{F}$ and for every nonnegative low-degree polynomial p the following holds: If the expectation of $f(x)p(x)$ under $\mathcal{D}$ is small, then the corresponding expectation under the part of $\mathcal{D}'$ selected by s, should be small as well.
2. The probability that s rejects a sample drawn from $\mathcal{D}$ (i.e., $\mathbf{Pr}_{x\sim\mathcal{D}}[s(x)=0]$) is small.

Concretely, we prove the following theorem. For the proof see Theorem B.1 in Appendix B.

Theorem 3.1 (Iterative Chow Filtering)**.** *Let any $\ell \in \mathbb{N}^*$, $A \geq 1$, $R > 1$, $\beta > 0$, $\epsilon, \delta \in (0,1)$ and let $\hat{f}\colon \mathbb{R}^d \to \{0,1\}$. Let $\mathcal{D}$ be any A-hypercontractive distribution on $\mathbb{R}^d$ and let $\mathcal{D}'$ be any distribution on $\mathbb{R}^d$. Let S and S' be multisets of i.i.d. examples drawn from $\mathcal{D}$ and $\mathcal{D}'$ respectively. If $|S|, |S'| \geq \mathrm{poly}(R, \beta, A^\ell, (d+1)^\ell, 1/\epsilon, \log(1/\delta)^\ell)$, then Algorithm 1, given $\hat{f}, S, S', \ell, R, \beta, \epsilon$ as input, runs in time $\mathrm{poly}\left(|S|, |S'|, (d+1)^\ell\right)$ and outputs a succinct polynomial-time-computable description of a function $s\colon \mathbb{R}^d \to \{0,1\}$ that satisfies the following guarantees.*

1. *Let any $p, q \in \mathcal{P}_{d,\ell}$ such that $\max\{\mathbf{E}_{x\sim\mathcal{D}}[p(x)^2], \mathbf{E}_{x\sim\mathcal{D}}[q(x)^2]\} \leq \beta$. Then, with probability at least $1-\delta$ over S and S', we have that*

$$\mathbf{E}_{x\sim\mathcal{D}'}[\hat{f}(x)p(x)s(x)] \leq (R+\epsilon)(1+\epsilon)\,\mathbf{E}_{x\sim\mathcal{D}}[\hat{f}(x)|p(x)|] + 2\epsilon\,,$$
$$\mathbf{E}_{x\sim\mathcal{D}'}[\bar{\hat{f}}(x)q(x)s(x)] \leq (R+\epsilon)(1+\epsilon)\,\mathbf{E}_{x\sim\mathcal{D}}[\bar{\hat{f}}(x)|q(x)|] + 2\epsilon\,.$$

2. *With probability at least $1-\delta$ over S and S', it holds $\mathbf{Pr}_{x\sim\mathcal{D}}[s(x)=0] \leq (1+\epsilon)/R$.*

The hyperparameter R in Theorem 3.1 determines how selective the filtering is, that is, larger values of R imply that the rejection rate of s under $\mathcal{D}$ will be smaller. The cost of larger choices of R is that the first (Chow-transfer) guarantee of Theorem 3.1 holds with a worse (larger) multiplicative factor.

ICF starts with a set of test points S' and performs a preliminary filtering step which ensures boundedness (see line 3). It proceeds with iteratively solving the optimization problem of line 5 to find some $f \in \mathcal{F}$ and some low-degree polynomial p such that $f(x)p(x)$ is large over the current set of accepted test points. Then, it uses f, p and the current set of accepted test points to find a rejection threshold τ and discards test points such that $f(x)|p(x)| > \tau$ (see line 9). After several filtering iterations, once the optimal value of the optimization problem in line 5 becomes sufficiently small, ICF outputs a selector, which is the intersection of all filtering rules accumulated over all iterations.

Proof Overview. The main idea behind the analysis of Algorithm 1 is that whenever the terminating condition of line 6 does not hold, there exists some $f \in \mathcal{F}$ and some low-degree polynomial p such that the expectation $f(x)|p(x)|$ under the currently accepted part of $\mathcal{D}'$ is R times larger than the corresponding expectation under $\mathcal{D}$. We show that in that case, there exists some threshold $\tau \geq 0$ such that $\mathcal{D}'$ assigns R times more mass on the currently accepted test points with $f(x)|p(x)| > \tau$ compared to $\mathcal{D}$. This implies that if we reject the test points with $f(x)|p(x)| > \tau$, then the mass of rejected points under $\mathcal{D}$ is at most $1/R$ times the corresponding mass under $\mathcal{D}'$ (which implies that the rejection rate of s will be at most $1/R$). Finally, we show that in each iteration the algorithm removes at least a Δ-fraction of the initial test sample, so the algorithm will eventually terminate.

Comparison with Prior Filtering Algorithms. ICF shares a similar filtering framework with the Iterative Polynomial Filtering (IPF) algorithm of [KSTV26] and the outlier removal method of [GSSV24]. The main difference is that ICF uses Chow-type expectations and handles localized quantities such as $f(x)p(x)$ and $f(x)|p(x)|$, whereas the algorithms of [KSTV26] and [GSSV24] handle purely polynomial quantities that do not depend on a classifier f.

Algorithm 1 Iterative Chow Filtering

Input: Classifier $\hat{f}$, sample S, S' from $\mathcal{D}, \mathcal{D}'$ respectively, maximum polynomial degree ℓ, multiplicative slack $R > 1$, polynomial variance bound $\beta > 0$, additive error $\epsilon \in (0, 1)$.

Output: A succinct description of a selector function $s \colon \mathbb{R}^d \to \{0, 1\}$.

Let $B \leftarrow 2\sqrt{\frac{2R(d+1)^\ell \beta}{\epsilon}}$; $\Delta \leftarrow \frac{\epsilon^2}{B(2R+\epsilon)}$; $\mathcal{F} \leftarrow \{\hat{f}, \bar{\hat{f}}\}$.

For $f \in \mathcal{F}$, define $\mathcal{P}(f) = \{p \in \mathcal{P}_{d,\ell} : \mathbf{E}_{x\sim\mathcal{U}(S)}[p(x)^2] \le 2\beta, \mathbf{E}_{x\sim\mathcal{U}(S)}[f(x)|p(x)|] \le \frac{2\epsilon}{2R+\epsilon}\}$.

Let $S_0 \leftarrow \{x \in S' : \max_{f\in\mathcal{F}} \max_{p\in\mathcal{P}(f)} f(x)|p(x)| \le B\}$.

for $i \in [|S_0| + 1]$ **do**

  Let $(f_i^\star, p_i^\star)$ be a solution and $\mu_i^\star$ the optimal value of the following optimization problem:

$$\max_{f\in\mathcal{F}} \max_{p\in\mathcal{P}(f)} \frac{1}{|S'|} \sum_{x\in S_{i-1}} f(x)p(x)$$

  if $\mu_i^\star \le \epsilon$ **then return** $s \colon \mathbb{R}^d \to \{0, 1\}$ such that for all $x \in \mathbb{R}^d$, it holds

$$s(x) = \mathbf{1}\left\{\max_{f\in\mathcal{F}} \max_{p\in\mathcal{P}(f)} f(x)|p(x)| \le B\right\} \prod_{j=1}^{i-1} \mathbf{1}\{f_j^\star(x)|p_j^\star(x)| \le \tau_j^\star\}\,.$$

  else

    Let $\tau_i^\star$ be the minimum nonnegative real number such that

$$\frac{|S_{i-1}|}{|S'|} \Pr_{x\sim\mathcal{U}(S_{i-1})}[f_i^\star(x)|p_i^\star(x)| > \tau_i^\star] \ge R \Pr_{x\sim\mathcal{U}(S)}[f_i^\star(x)|p_i^\star(x)| > \tau_i^\star] + \Delta\,.$$

    $S_i \leftarrow S_{i-1} \setminus \{x \in S_{i-1} : f_i^\star(x)|p_i^\star(x)| > \tau_i^\star\}$. ▷ Filtering step

  end if

end for

4 PQ Learning

In this section, we state our main result for PQ learning. Recall that the goal in PQ learning is to find a classifier $\hat{f} \colon \mathcal{X} \to \{0, 1\}$ and a selector $s \colon \mathcal{X} \to \{0, 1\}$ such that (i) $\hat{f}$ achieves small error on the part of $\mathcal{D}^{\text{test}}$ selected by s, and (ii) the rejection rate of s is small under $\mathcal{D}_{\mathcal{X}}^{\text{train}}$.

Algorithm Sketch. The steps of our PQ learner are as follows (see also Theorem C.1).

- First, we learn a classifier $\hat{f}$ via $\mathcal{L}_1$ polynomial regression on a training sample from $\mathcal{D}^{\text{train}}$.
- Then, we run ICF with input $\hat{f}$ and examples from $\mathcal{D}_{\mathcal{X}}^{\text{train}}$ and $\mathcal{D}_{\mathcal{X}}^{\text{test}}$ to obtain a selector s.
- Finally, we output $(\hat{f}, s)$.

Since $\hat{f}$ is learned using labeled examples from the training distribution, we can easily guarantee that $\hat{f}$ achieves low training error. The main difficulty in proving the correctness of our algorithm, lies in identifying a condition that will allow us to transfer the error from the training distribution to the selected part of the test distribution. To this end, we establish the following transfer lemma.

Lemma 4.1 (Chow Transfer)**.** *Let $\mathcal{X} \subseteq \mathbb{R}^d$. Let $\mathcal{D}^{\text{train}}, \mathcal{D}^{\text{test}}$ be distributions on $\mathcal{X} \times \{0, 1\}$ and let $\mathcal{D}$ and $\mathcal{D}'$ be their $\mathcal{X}$-marginals respectively. Let $\epsilon > 0$, $R > 1$, let $\mathcal{C} \subseteq \{0, 1\}^{\mathcal{X}}$, let any $f^\star \in \mathcal{C}$ such that $\text{err}_{\text{train}}(f^\star) = \lambda_{\text{train}}$ and $\text{err}_{\text{test}}(f^\star) = \lambda_{\text{test}}$ and let $p_{\text{up}}, p_{\text{down}}$ be $\epsilon/(4R)$-approximate $\mathcal{L}_1$ sandwiching polynomials for $f^\star$ under $\mathcal{D}$.*[7] *Suppose that for some $\hat{f} \colon \mathcal{X} \to \{0, 1\}$ and $s \colon \mathcal{X} \to \{0, 1\}$, we have*

1. $\mathbf{E}_{x\sim\mathcal{D}'}[\hat{f}(x)\bar{p}_{\text{down}}(x)s(x)] \le R\,\mathbf{E}_{x\sim\mathcal{D}}[\hat{f}(x)\bar{p}_{\text{down}}(x)] + \epsilon/4$, *and*
2. $\mathbf{E}_{x\sim\mathcal{D}'}[\bar{\hat{f}}(x)p_{\text{up}}(x)s(x)] \le R\,\mathbf{E}_{x\sim\mathcal{D}}[\bar{\hat{f}}(x)p_{\text{up}}(x)] + \epsilon/4$.

[7] Recall that this means that $p_{\text{down}}(x) \le f^\star(x) \le p_{\text{up}}(x)$ for all $x \in \mathcal{X}$ and $\mathbf{E}_{x\sim\mathcal{D}}[p_{\text{up}}(x) - p_{\text{down}}(x)] \le \epsilon/(4R)$.

Then, it holds $\mathbf{Pr}_{(x,y)\sim\mathcal{D}^{\mathrm{test}}}[\hat{f}(x) \neq y \wedge s(x) = 1] \leq \lambda_{\mathrm{test}} + R(\lambda_{\mathrm{train}} + \mathrm{err}_{\mathrm{train}}(\hat{f})) + \epsilon$.

In words, Lemma 4.1 says the following: If we find a selector s such that the Chow-type expectations of $\hat{f}$ and $\bar{\hat{f}}$ against appropriate choices of polynomials with respect to the selected part of $\mathcal{D}^{\mathrm{test}}$ are bounded by their analogues with respect to $\mathcal{D}^{\mathrm{train}}$ up to a multiplicative factor of R, then the error of $\hat{f}$ on the selected part of $\mathcal{D}^{\mathrm{test}}$ is roughly bounded by $R(\lambda + \mathrm{err}_{\mathrm{train}}(\hat{f}))$.

By the first guarantee of Theorem 3.1, a selector with the aforementioned property can be obtained via our ICF algorithm.[8] Moreover, by the second guarantee of Theorem 3.1 any selector obtained via the ICF algorithm, ensures low rejection rate (roughly at most $1/R$) with respect to $\mathcal{D}^{\mathrm{train}}_{\mathcal{X}}$. Hence, choosing $R = 1/\eta$ would imply that the error of $\hat{f}$ on the selected part of $\mathcal{D}^{\mathrm{test}}$ is at most $(\lambda + \mathrm{err}_{\mathrm{train}}(\hat{f}))/\eta$ and the rejection rate of the selector is at most η. This provides a high level idea of the correctness of our PQ learning algorithm. We now prove Lemma 4.1.

Proof of Lemma 4.1. Let $\epsilon' = \epsilon/(4R)$. We have

$$\begin{aligned}
\mathop{\mathbf{Pr}}_{x\sim\mathcal{D}'}[\hat{f}(x) \neq f^\star(x) \wedge s(x) = 1] &= \mathop{\mathbf{E}}_{x\sim\mathcal{D}'}[\hat{f}(x)\bar{f}^\star(x)s(x)] + \mathop{\mathbf{E}}_{x\sim\mathcal{D}'}[\bar{\hat{f}}(x)f^\star(x)s(x)] \\
&\leq \mathop{\mathbf{E}}_{x\sim\mathcal{D}'}[\hat{f}(x)\bar{p}_{\mathrm{down}}(x)s(x)] + \mathop{\mathbf{E}}_{x\sim\mathcal{D}'}[\bar{\hat{f}}(x)p_{\mathrm{up}}(x)s(x)] \\
&\leq R\left(\mathop{\mathbf{E}}_{x\sim\mathcal{D}}[\hat{f}(x)\bar{p}_{\mathrm{down}}(x)] + \mathop{\mathbf{E}}_{x\sim\mathcal{D}}[\bar{\hat{f}}(x)p_{\mathrm{up}}(x)]\right) + \epsilon/2\,,
\end{aligned}$$

where we used the fact that $p_{\mathrm{down}} \leq f^\star \leq p_{\mathrm{up}}$ and the assumption of Lemma 4.1. Now, we upper bound each of $\mathbf{E}_{x\sim\mathcal{D}}[\hat{f}(x)\bar{p}_{\mathrm{down}}(x)]$ and $\mathbf{E}_{x\sim\mathcal{D}}[\bar{\hat{f}}(x)p_{\mathrm{up}}(x)]$ separately. We have

$$\begin{aligned}
\mathop{\mathbf{E}}_{x\sim\mathcal{D}}[\hat{f}(x)\bar{p}_{\mathrm{down}}(x)] &= \mathop{\mathbf{E}}_{x\sim\mathcal{D}}[\hat{f}(x)\bar{f}^\star(x)] + \mathop{\mathbf{E}}_{x\sim\mathcal{D}}[\hat{f}(x)(\bar{p}_{\mathrm{down}}(x) - \bar{f}^\star(x))] \\
&\leq \mathop{\mathbf{E}}_{x\sim\mathcal{D}}[\hat{f}(x)\bar{f}^\star(x)] + \mathop{\mathbf{E}}_{x\sim\mathcal{D}}[p_{\mathrm{up}}(x) - p_{\mathrm{down}}(x)] \leq \mathop{\mathbf{E}}_{x\sim\mathcal{D}}[\hat{f}(x)\bar{f}^\star(x)] + \epsilon'\,,
\end{aligned}$$

where we used the fact that $\bar{p}_{\mathrm{down}} - \bar{f}^\star \leq p_{\mathrm{up}} - p_{\mathrm{down}}$ and $\mathbf{E}_{x\sim\mathcal{D}}[p_{\mathrm{up}}(x) - p_{\mathrm{down}}(x)] \leq \epsilon'$. Similarly,

$$\mathop{\mathbf{E}}_{x\sim\mathcal{D}}[\bar{\hat{f}}(x)p_{\mathrm{up}}(x)] = \mathop{\mathbf{E}}_{x\sim\mathcal{D}}[\bar{\hat{f}}(x)f^\star(x)] + \mathop{\mathbf{E}}_{x\sim\mathcal{D}}[\bar{\hat{f}}(x)(p_{\mathrm{up}}(x) - f^\star(x))] \leq \mathop{\mathbf{E}}_{x\sim\mathcal{D}}[\bar{\hat{f}}(x)f^\star(x)] + \epsilon'\,.$$

Putting everything together, we have

$$\begin{aligned}
\mathop{\mathbf{Pr}}_{x\sim\mathcal{D}'}[\hat{f}(x) \neq f^\star(x) \wedge s(x) = 1] &\leq R\left(\mathop{\mathbf{E}}_{x\sim\mathcal{D}}[\hat{f}(x)\bar{f}^\star(x)] + \mathop{\mathbf{E}}_{x\sim\mathcal{D}}[\bar{\hat{f}}(x)f^\star(x)] + 2\epsilon'\right) + \epsilon/2 \\
&= R\left(\mathop{\mathbf{Pr}}_{x\sim\mathcal{D}}[\hat{f}(x) \neq f^\star(x)] + 2\epsilon'\right) + \epsilon/2 \\
&\leq R\left(\mathop{\mathbf{Pr}}_{(x,y)\sim\mathcal{D}^{\mathrm{train}}}[f^\star(x) \neq y] + \mathop{\mathbf{Pr}}_{(x,y)\sim\mathcal{D}^{\mathrm{train}}}[\hat{f}(x) \neq y]\right) + \epsilon\,,
\end{aligned}$$

where we applied the triangle inequality in the last line. Finally, another application of the triangle inequality implies that

$$\begin{aligned}
\mathop{\mathbf{Pr}}_{(x,y)\sim\mathcal{D}^{\mathrm{test}}}[\hat{f}(x) \neq y \wedge s(x) = 1] &\leq \mathop{\mathbf{Pr}}_{(x,y)\sim\mathcal{D}^{\mathrm{test}}}[f^\star(x) \neq y] + \mathop{\mathbf{Pr}}_{x\sim\mathcal{D}'}[\hat{f}(x) \neq f^\star(x) \wedge s(x) = 1] \\
&\leq \lambda_{\mathrm{test}} + R\left(\lambda_{\mathrm{train}} + \mathop{\mathbf{Pr}}_{(x,y)\sim\mathcal{D}^{\mathrm{train}}}[\hat{f}(x) \neq y]\right) + \epsilon\,,
\end{aligned}$$

which concludes the proof. □

We now state our main theorem for PQ learning. For the proof and applications to standard classes see Appendix C.

[8] The first guarantee of Theorem 3.1 is applied on the polynomials p_{up} and $\bar{p}_{\mathrm{down}}$ of Lemma 4.1. Those polynomials only depend on $\mathcal{D}^{\mathrm{train}}$, $\mathcal{D}^{\mathrm{test}}$ and $\mathcal{C}$ and, although they are unknown to the algorithm, Theorem 3.1 can still be applied for them, since it holds for any fixed pair of low-degree polynomials.

Theorem 4.1 (PQ Learning via $\mathcal{L}_1$ Sandwiching)**.** *Let any $\epsilon, \delta, \eta \in (0,1)$ and $A \geq 1$. Let $\mathcal{X} \subseteq \mathbb{R}^d$ and let $\mathcal{D}$ be any A-hypercontractive distribution on $\mathcal{X}$. Let $\mathcal{C} \subseteq \{0,1\}^{\mathcal{X}}$ be a concept class and let $\ell = \mathcal{L}_1(\mathcal{C}, \mathcal{D}, \epsilon\eta/8)$. Then, there exists an (ϵ, δ, η)-PQ learner for $\mathcal{C}$ with respect to $\mathcal{D}$ with sample complexity and runtime* $\mathrm{poly}((d+1)^\ell, A^\ell, 1/\epsilon, 1/\eta, \log(1/\delta)^\ell)$.

Limitations and Future Work. Our results rely on the existence of $\mathcal{L}_1$ sandwiching polynomials with respect to a hypercontractive training distribution. Whether such sandwiching conditions are necessary for efficient algorithms in the PQ setting remains an open question. Moreover, although our error bounds are sharp and are tight in certain special cases, including the construction of [GKKM20], it is unclear whether these bounds can be improved in other special cases, even by computationally inefficient algorithms.

Acknowledgments and Disclosure of Funding

All authors were supported in part by the NSF AI Institute for Foundations of Machine Learning (IFML). Konstantinos Stavropoulos was also supported by the 2025 Apple Scholars in AI/ML PhD fellowship and in part by the Institute for Advanced Study and an anonymous donor. This work was done in part while Konstantinos Stavropoulos and Arsen Vasilyan were at The University of Texas at Austin.

References

[Ban65] John F. Banzhaf. Weighted voting doesn't work: A mathematical analysis. *Rutgers Law Review*, 19:317–343, 1965.

[Baz09] Louay MJ Bazzi. Polylogarithmic independence can fool dnf formulas. *SIAM Journal on Computing*, 38(6):2220–2272, 2009.

[BCK+07] John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman. Learning bounds for domain adaptation. *Advances in neural information processing systems*, 20, 2007.

[BDBC+10] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79:151–175, 2010.

[BDBCP06] Shai Ben-David, John Blitzer, Koby Crammer, and Fernando Pereira. Analysis of representations for domain adaptation. *Advances in neural information processing systems*, 19, 2006.

[BDD98] Shai Ben-David and Eli Dichterman. Learning with restricted focus of attention. *Journal of Computer and System Sciences*, 56:277–298, 1998.

[BDJ+98] Andreas Birkendorf, Eli Dichterman, Jeffrey Jackson, Nathaniel Klasner, and Hans Ulrich Simon. On restricted-focus-of-attention learnability of boolean functions. *Machine Learning*, 30:89–123, 1998.

[Bob01] Sergey G Bobkov. Some generalizations of prokhorov's results on khinchin-type inequalities for polynomials. *Theory of Probability & Its Applications*, 45(4):644–647, 2001.

[Bra08] Mark Braverman. Polylogarithmic independence fools ac0 circuits. *J. ACM*, 57(5), June 2008.

[Cho61] C. K. Chow. On the characterization of threshold functions. In *Proceedings of the 2nd Annual Symposium on Switching Circuit Theory and Logical Design (SWCT 1961)*, FOCS '61, page 34–38, USA, 1961. IEEE Computer Society.

[CKK+24] Gautam Chandrasekaran, Adam Klivans, Vasilis Kontonis, Konstantinos Stavropoulos, and Arsen Vasilyan. Efficient discrepancy testing for learning with distribution shift. *Advances in Neural Information Processing Systems*, 37:137263–137308, 2024.

[CKLS25] Gautam Chandrasekaran, Adam Klivans, Lin Lin Lee, and Konstantinos Stavropoulos. Learning neural networks with distribution shift: Efficiently certifiable guarantees. In *The Thirteenth International Conference on Learning Representations*, 2025.

[CKSV25] Gautam Chandrasekaran, Adam R Klivans, Konstantinos Stavropoulos, and Arsen Vasilyan. A fully polynomial-time algorithm for robustly learning halfspaces over the hypercube. *arXiv preprint arXiv:2511.07244*, 2025.

[DDFS14] Anindya De, Ilias Diakonikolas, Vitaly Feldman, and Rocco A. Servedio. Nearly optimal solutions for the chow parameters problem and low-weight approximation of halfspaces. *J. ACM*, 61(2), April 2014.

[DGJ+10] Ilias Diakonikolas, Parikshit Gopalan, Ragesh Jaiswal, Rocco A Servedio, and Emanuele Viola. Bounded independence fools halfspaces. *SIAM Journal on Computing*, 39(8):3441–3462, 2010.

[DK19] Ilias Diakonikolas and Daniel M. Kane. Degree-d chow parameters robustly determine degree-d ptfs (and algorithmic applications). In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, page 804–815, New York, NY, USA, 2019. Association for Computing Machinery.

[DKK+19a] Ilias Diakonikolas, Gautam Kamath, Daniel Kane, Jerry Li, Ankur Moitra, and Alistair Stewart. Robust estimators in high-dimensions without the computational intractability. *SIAM Journal on Computing*, 48(2):742–864, 2019.

[DKK+19b] Ilias Diakonikolas, Gautam Kamath, Daniel Kane, Jerry Li, Jacob Steinhardt, and Alistair Stewart. Sever: A robust meta-algorithm for stochastic optimization. In *International Conference on Machine Learning*, pages 1596–1606. PMLR, 2019.

[DKN10] Ilias Diakonikolas, Daniel M Kane, and Jelani Nelson. Bounded independence fools degree-2 threshold functions. In *2010 IEEE 51st Annual Symposium on Foundations of Computer Science*, pages 11–20. IEEE, 2010.

[DKS18] Ilias Diakonikolas, Daniel M Kane, and Alistair Stewart. Learning geometric concepts with nasty noise. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1061–1073, 2018.

[DLLP10] Shai Ben David, Tyler Lu, Teresa Luu, and Dávid Pál. Impossibility theorems for domain adaptation. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 129–136. JMLR Workshop and Conference Proceedings, 2010.

[DS79] Pradeep Dubey and Lloyd S. Shapley. Mathematical properties of the banzhaf power index. *Mathematics of Operations Research*, 4:99–131, 1979.

[Fer14] Dietmar Ferger. Optimal constants in the marcinkiewicz–zygmund inequalities. *Statistics & Probability Letters*, 84:96–101, 2014.

[GHR92] Mikael Goldmann, Johan Håstad, and Alexander Razborov. Majority gates vs. general weighted threshold gates. *Computational Complexity*, 2:277–300, 1992.

[GKK23] Aravind Gollakota, Adam R. Klivans, and Pravesh K. Kothari. A moment-matching approach to testable learning and a new characterization of rademacher complexity. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023, page 1657–1670, New York, NY, USA, 2023. Association for Computing Machinery.

[GKKM20] Shafi Goldwasser, Adam Tauman Kalai, Yael Kalai, and Omar Montasser. Beyond perturbations: Learning guarantees with arbitrary adversarial test examples. *Advances in Neural Information Processing Systems*, 33:15859–15870, 2020.

[GKSV25] Surbhi Goel, Adam R Klivans, Konstantinos Stavropoulos, and Arsen Vasilyan. Testing noise assumptions of learning algorithms. *arXiv preprint arXiv:2501.09189*, 2025.

[GOWZ10] Parikshit Gopalan, Ryan O'Donnell, Yi Wu, and David Zuckerman. Fooling functions of halfspaces under product distributions. In *2010 IEEE 25th Annual Conference on Computational Complexity*, pages 223–234. IEEE, 2010.

[GSSV24] Surbhi Goel, Abhishek Shetty, Konstantinos Stavropoulos, and Arsen Vasilyan. Tolerant algorithms for learning with arbitrary covariate shift. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 124979–125018. Curran Associates, Inc., 2024.

[HH23] Pooya Hatami and William Hoza. Theory of unconditional pseudorandom generators. In *Electron. Colloquium Comput. Complex., TR23-019*, volume 1, page 1, 2023.

[HS18] Prahladh Harsha and Srikanth Srinivasan. On polynomial approximations to ac. *Random Structures & Algorithms*, 54(2):289–303, 2018.

[Kan11] Daniel M. Kane. k-independent gaussians fool polynomial threshold functions. In *2011 IEEE 26th Annual Conference on Computational Complexity*, pages 252–261, 2011.

[KK21] Adam Tauman Kalai and Varun Kanade. Efficient learning with arbitrary covariate shift. In Vitaly Feldman, Katrina Ligett, and Sivan Sabato, editors, *Proceedings of the 32nd International Conference on Algorithmic Learning Theory*, volume 132 of *Proceedings of Machine Learning Research*, pages 850–864. PMLR, 16–19 Mar 2021.

[KKM12] Adam Tauman Kalai, Varun Kanade, and Yishay Mansour. Reliable agnostic learning. *Journal of Computer and System Sciences*, 78(5):1481–1495, 2012. JCSS Special Issue: Cloud Computing 2011.

[KKM13] Daniel M. Kane, Adam R. Klivans, and Raghu Meka. Learning halfspaces under log-concave densities: Polynomial approximations and moment matching. In Shai Shalev-Shwartz and Ingo Steinwart, editors, *COLT 2013 - The 26th Annual Conference on Learning Theory, June 12-14, 2013, Princeton University, NJ, USA*, volume 30 of *JMLR Workshop and Conference Proceedings*, pages 522–545. JMLR.org, 2013.

[KKMS08] Adam Tauman Kalai, Adam R. Klivans, Yishay Mansour, and Rocco A. Servedio. Agnostically learning halfspaces. *SIAM Journal on Computing*, 37(6):1777–1805, 2008.

[KLS09] Adam R Klivans, Philip M Long, and Rocco A Servedio. Learning halfspaces with malicious noise. *Journal of Machine Learning Research*, 10(12), 2009.

[KSTV26] Adam Klivans, Konstantinos Stavropoulos, Kevin Tian, and Arsen Vasilyan. The power of iterative filtering for supervised learning with (heavy) contamination. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026.

[KSV24a] Adam Klivans, Konstantinos Stavropoulos, and Arsen Vasilyan. Testable learning with distribution shift. In Shipra Agrawal and Aaron Roth, editors, *Proceedings of Thirty Seventh Conference on Learning Theory*, volume 247 of *Proceedings of Machine Learning Research*, pages 2887–2943. PMLR, 30 Jun–03 Jul 2024.

[KSV24b] Adam R Klivans, Konstantinos Stavropoulos, and Arsen Vasilyan. Learning intersections of halfspaces with distribution shift: Improved algorithms and sq lower bounds. *The Thirty Seventh Annual Conference on Learning Theory*, 2024.

[KSV25] Adam Klivans, Konstantinos Stavropoulos, and Arsen Vasilyan. Learning constant-depth circuits in malicious noise models. In Nika Haghtalab and Ankur Moitra, editors, *Proceedings of Thirty Eighth Conference on Learning Theory*, volume 291 of *Proceedings of Machine Learning Research*, pages 3253–3263. PMLR, 30 Jun–04 Jul 2025.

[KSV26] Adam R Klivans, Konstantinos Stavropoulos, and Arsen Vasilyan. Sandwiching polynomials for geometric concepts with low intrinsic dimension. *arXiv preprint arXiv:2602.24178*, 2026.

[LMN93] Nathan Linial, Yishay Mansour, and Noam Nisan. Constant depth circuits, fourier transform, and learnability. *J. ACM*, 40(3):607–620, July 1993.

[MMR09] Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *Proceedings of The 22nd Annual Conference on Learning Theory (COLT 2009)*, Montréal, Canada, 2009.

[OS03] Ryan O'Donnell and Rocco A. Servedio. New degree bounds for polynomial threshold functions. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '03, page 325–334, New York, NY, USA, 2003. Association for Computing Machinery.

[OS08] Ryan O'Donnell and Rocco A. Servedio. The chow parameters problem. In *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing*, STOC '08, page 517–526, New York, NY, USA, 2008. Association for Computing Machinery.

[Raz09] Alexander Razborov. A simple proof of bazzi's theorem. *ACM Transactions on Computation Theory (TOCT)*, 1(1):1–5, 2009.

[RV23] Ronitt Rubinfeld and Arsen Vasilyan. Testing distributional assumptions of learning algorithms. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023, page 1643–1656, New York, NY, USA, 2023. Association for Computing Machinery.

[STW24] Lucas Slot, Stefan Tiegel, and Manuel Wiedmer. Testably learning polynomial threshold functions. *Advances in Neural Information Processing Systems*, 37:3781–3831, 2024.

[Tal17] Avishay Tal. Tight Bounds on the Fourier Spectrum of AC0. In Ryan O'Donnell, editor, *32nd Computational Complexity Conference (CCC 2017)*, volume 79 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 15:1–15:31, Dagstuhl, Germany, 2017. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[VW09] Aad Vaart and Jon Wellner. A note on bounds for vc dimensions. *Institute of Mathematical Statistics collections*, 5:103–107, 01 2009.

[Win71] R. O. Winder. Chow parameters in threshold logic. *Journal of the ACM*, 18:265–289, 1971.

A Extended Preliminaries

A.1 Notation

Let $\mathbb{N} = \{0, 1, \dots\}$ denote the set of natural numbers and $\mathbb{N}^* = \mathbb{N} \setminus \{0\}$. Let $\mathbb{Z}$ denote the set of integers. For any $d, \ell \in \mathbb{N}$, let $\mathcal{P}_{d,\ell}$ denote the set of d-variate polynomials, whose degree is at most ℓ. For any polynomial p, let $\deg(p)$ denote its degree. For any $d \in \mathbb{N}^*$, let $\mathcal{N}_d$ denote the standard normal distribution on $\mathbb{R}^d$. For any multiset S, we let $\mathcal{U}(S)$ denote the uniform distribution on S. For any real-valued function f, we denote $\bar{f} = 1 - f$. For any real-valued functions f, g, we use $f \lesssim g$ (resp. $f \gtrsim g$) to indicate that there exists some universal constant $C > 0$ such that $f \leq Cg$ (resp. $Cf \geq g$). For any pair of distributions $\mathcal{D}, \mathcal{D}'$ on some set, we let $d_{\mathrm{TV}}(\mathcal{D}, \mathcal{D}')$ denote the total variation distance between $\mathcal{D}$ and $\mathcal{D}'$. For any pair of training and test distributions $\mathcal{D}^{\mathrm{train}}$ and $\mathcal{D}^{\mathrm{test}}$ on $\mathcal{X} \times \mathcal{Y}$, we denote their marginals on $\mathcal{X}$ by $\mathcal{D}^{\mathrm{train}}_{\mathcal{X}}$ and $\mathcal{D}^{\mathrm{test}}_{\mathcal{X}}$ respectively. Finally, for a given concept class $\mathcal{C} \subseteq \mathcal{Y}^{\mathcal{X}}$, we denote $\mathsf{opt}_{\mathrm{train}} = \inf_{f \in \mathcal{C}} \mathbf{Pr}_{(x,y)\sim\mathcal{D}^{\mathrm{train}}}[f(x) \neq y]$.

A.2 Learning Theory

Definition A.1. A halfspace in $\mathbb{R}^d$ is any function $h \colon \mathbb{R}^d \to \{\pm 1\}$ of the form $h(x) = \mathrm{sign}(w \cdot x + b)$ for some unit vector $w \in \mathbb{R}^d$ and $b \in \mathbb{R}$.

Definition A.2. Let any $\ell \in \mathbb{N}$. A degree-ℓ polynomial threshold function (PTF) in $\mathbb{R}^d$ is any function $h \colon \mathbb{R}^d \to \{\pm 1\}$ of the form $h(x) = \mathrm{sign}(p(x))$ for some $p \in \mathcal{P}_{d,\ell}$.

Definition A.3. Let any set $\mathcal{X}$, $r \in \mathbb{N}^*$, $\mathcal{H}_1, \dots, \mathcal{H}_r \subseteq \{0,1\}^{\mathcal{X}}$ and let $V = \max_{i\in[r]} \mathrm{VCdim}(\mathcal{H}_i)$. The OR of $\mathcal{H}_1, \dots, \mathcal{H}_r$ is defined as $\bigvee_{i=1}^r \mathcal{H}_i = \{h_1 \vee \cdots \vee h_r : h_1 \in \mathcal{H}_1, \dots, h_r \in \mathcal{H}_r\}$ and the *AND of* $\mathcal{H}_1, \dots, \mathcal{H}_r$ is defined as $\bigwedge_{i=1}^r \mathcal{H}_i = \{h_1 \wedge \cdots \wedge h_r : h_1 \in \mathcal{H}_1, \dots, h_r \in \mathcal{H}_r\}$.

Fact A.1. *Let any set $\mathcal{X}$, $r \in \mathbb{N}^*$, $\mathcal{H}_1, \dots, \mathcal{H}_r \subseteq \{0,1\}^{\mathcal{X}}$ and let $V = \max_{i\in[r]} \mathrm{VCdim}(\mathcal{H}_i)$. Then, it holds $\mathrm{VCdim}(\bigcup_{i=1}^r \mathcal{H}_i) \lesssim V \log(V) + \log(r)$.*

Fact A.2 ([VW09] and references therein)**.** *Let any set $\mathcal{X}$, $r \in \mathbb{N}^*$, $\mathcal{H}_1, \dots, \mathcal{H}_r \subseteq \{0,1\}^{\mathcal{X}}$ and let $V = \max_{i\in[r]} \mathrm{VCdim}(\mathcal{H}_i)$. Then, it holds $\mathrm{VCdim}(\bigvee_{i=1}^r \mathcal{H}_i) \lesssim Vr\log(r)$ and $\mathrm{VCdim}(\bigwedge_{i=1}^r \mathcal{H}_i) \lesssim Vr\log(r)$.*

Fact A.3. *For any $d, \ell \in \mathbb{N}^*$, the VC dimension of polynomial threshold functions of degree at most ℓ in $\mathbb{R}^d$ is at most $(d+1)^\ell$.*

Fact A.4. *Let any set $\mathcal{X}$, any distribution $\mathcal{D}$ on $\mathcal{X}$ and any class $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ such that $\mathrm{VCdim}(\mathcal{H}) < \infty$. Let S be a multiset of n i.i.d. samples drawn from $\mathcal{D}$. Then, there exists some universal constant $C > 0$ such that for any $\delta \in (0,1)$, with probability at least $1 - \delta$ over S, it holds*

$$\sup_{h\in\mathcal{H}} \left| \mathop{\mathbf{E}}_{x\sim\mathcal{U}(S)}[h(x)] - \mathop{\mathbf{E}}_{x\sim\mathcal{D}}[h(x)] \right| \leq C\sqrt{\frac{\mathrm{VCdim}(\mathcal{H})\log(en) + \log(1/\delta)}{n}}\,.$$

A.3 Sandwiching Approximators

In this section, we present known literature results on the existence of $\mathcal{L}_1$ sandwiching polynomials for various concept classes.

Fact A.5 ([Bra08, Tal17, HS18])**.** *Let any $\epsilon \in (0,1)$. Let $\mathcal{C}$ be the class of depth-t, size-s AC^0 circuits in $\{\pm 1\}^d$. Then, $\mathcal{L}_1(\mathcal{C}, \mathcal{U}(\{\pm 1\}^d), \epsilon) \leq O(\log(s))^{O(t)} \log(1/\epsilon)$.*

Fact A.6 ([GOWZ10])**.** *Let any $\epsilon \in (0,1)$. Let $\mathcal{C}$ be the class of depth-t, size-s decision trees of halfspaces in $\mathbb{R}^d$. Then, $\mathcal{L}_1(\mathcal{C}, \mathcal{N}_d, \epsilon) \leq \tilde{O}(t^4 s^2/\epsilon^2)$ and $\mathcal{L}_1(\mathcal{C}, \mathcal{U}(\{\pm 1\}^d), \epsilon) \leq \tilde{O}(t^4 s^2/\epsilon^2)$.*

Fact A.7 ([DKN10])**.** *Let any $\epsilon \in (0,1)$. Let $\mathcal{C}$ be the class of degree-2 PTFs in $\mathbb{R}^d$. Then, $\mathcal{L}_1(\mathcal{C}, \mathcal{N}_d, \epsilon) \leq \tilde{O}(1/\epsilon^8)$ and $\mathcal{L}_1(\mathcal{C}, \mathcal{U}(\{\pm 1\}^d), \epsilon) \leq \tilde{O}(1/\epsilon^9)$.*

Fact A.8 ([Kan11, STW24])**.** *Let any $\epsilon \in (0,1)$. Let $\mathcal{C}$ be the class of degree-k PTFs in $\mathbb{R}^d$. Then, $\mathcal{L}_1(\mathcal{C}, \mathcal{N}_d, \epsilon) \leq O_k(\epsilon^{-4k\cdot 7^k})$.*[9]

Fact A.9 ([KKM13])**.** *Let any $\epsilon \in (0,1)$. Let $\mathcal{C}$ be the class of arbitrary functions of k halfspaces in $\mathbb{R}^d$ and let $\mathcal{D}$ be any log-concave distribution on $\mathbb{R}^d$. Then, $\mathcal{L}_1(\mathcal{C}, \mathcal{D}, \epsilon) \leq \exp((\log(\log(k)/\epsilon))^{O(k)}/\epsilon^4)$.*

[9] Here $O_k(\cdot)$ is hiding a multiplicative factor that scales as an arbitrary function of k.

B Iterative Chow Filtering

In this section, we provide the proof of our main ICF theorem. In fact, here we state Algorithm 2 and Theorem B.1, which are the more general versions of Algorithm 1 and Theorem 3.1 respectively. The difference of Theorem B.1 from Theorem 3.1 is that the guarantee of Theorem B.1 holds for any finite collection of classifiers $\mathcal{F} \subset \{\mathbb{R}^d \to \{0,1\}\}$ instead of $\mathcal{F} = \{\hat{f}, \bar{\hat{f}}\}$. Moreover, Theorem B.1 provides an additional upper bound on the rejection rate of the selector, which is crucial to obtaining our tolerant TDS learning result (see Appendix D). We now prove Theorem B.1 (which Theorem 3.1 follows trivially from).

Algorithm 2 Iterative Chow Filtering

Input: Family of classifiers $\mathcal{F} \subset \{\mathbb{R}^d \to \{0,1\}\}$, sample S, S' from $\mathcal{D}, \mathcal{D}'$ respectively, maximum polynomial degree ℓ, multiplicative slack $R > 1$, polynomial variance bound $\beta > 0$, additive error $\epsilon \in (0,1)$.

Output: A succinct description of a selector function $s\colon \mathbb{R}^d \to \{0,1\}$.

1: Let $B \leftarrow 2\sqrt{\frac{2R(d+1)^\ell \beta}{\epsilon}}$; $\Delta \leftarrow \frac{\epsilon^2}{B(2R+\epsilon)}$; $t \leftarrow 0$.
2: For $f \in \mathcal{F}$, define $\mathcal{P}(f) = \{p \in \mathcal{P}_{d,\ell} : \mathbf{E}_{x\sim\mathcal{U}(S)}[p(x)^2] \le 2\beta, \mathbf{E}_{x\sim\mathcal{U}(S)}[f(x)|p(x)|] \le \frac{2\epsilon}{2R+\epsilon}\}$.
3: Let $S_0 \leftarrow \{x \in S' : \max_{f\in\mathcal{F}} \max_{p\in\mathcal{P}(f)} f(x)|p(x)| \le B\}$.
4: **for** $i \in [|S_0| + 1]$ **do**
5: Let $(f_i^\star, p_i^\star)$ be a solution and $\mu_i^\star$ the optimal value of the following optimization problem:

$$\max_{f\in\mathcal{F}} \max_{p\in\mathcal{P}(f)} \frac{1}{|S'|} \sum_{x\in S_{i-1}} f(x)p(x)$$

6: **if** $\mu_i^\star \le \epsilon$ **then**
7: Set $t \leftarrow i-1$ and **break** the loop.
8: **else**
9: Let $\tau_i^\star$ be the minimum nonnegative real number such that

$$\frac{|S_{i-1}|}{|S'|} \mathop{\mathbf{Pr}}_{x\sim\mathcal{U}(S_{i-1})}[f_i^\star(x)|p_i^\star(x)| > \tau_i^\star] \ge R \mathop{\mathbf{Pr}}_{x\sim\mathcal{U}(S)}[f_i^\star(x)|p_i^\star(x)| > \tau_i^\star] + \Delta\,.$$

10: $S_i \leftarrow S_{i-1} \setminus \{x \in S_{i-1} : f_i^\star(x)|p_i^\star(x)| > \tau_i^\star\}$. ▷ Filtering step
11: **end if**
12: **end for**
13: **return** $s\colon \mathbb{R}^d \to \{0,1\}$ such that for all $x \in \mathbb{R}^d$, it holds

$$s(x) = \mathbf{1}\left\{\max_{f\in\mathcal{F}} \max_{p\in\mathcal{P}(f)} f(x)|p(x)| \le B\right\} \prod_{i=1}^{t} \mathbf{1}\{f_i^\star(x)|p_i^\star(x)| \le \tau_i^\star\}\,.$$

We first prove the following proposition, which ensures that all steps of Algorithm 2 are well-defined.

Proposition B.1. *In the setting of Algorithm 2, for any iteration i such that $\mu_i^\star > \epsilon$, there exists some $\tau \in [0, B]$ that satisfies the condition of line 9.*

Proof. Let any iteration i such that $\mu_i^\star > \epsilon$. Suppose, for contradiction, that for all $\tau \in [0, B]$, it holds

$$\frac{|S_{i-1}|}{|S'|} \mathop{\mathbf{Pr}}_{x\sim\mathcal{U}(S_{i-1})}[f_i^\star(x)|p_i^\star(x)| > \tau] < R \mathop{\mathbf{Pr}}_{x\sim\mathcal{U}(S)}[f_i^\star(x)|p_i^\star(x)| > \tau] + \Delta\,.$$

Now, we may integrate over $\tau \in [0, B]$ both sides of the above inequality, since the corresponding functions of τ have a finite number of discontinuities.

$$\frac{|S_{i-1}|}{|S'|} \int_{\tau=0}^{B} \mathop{\mathbf{Pr}}_{x\sim\mathcal{U}(S_{i-1})}[f_i^\star(x)|p_i^\star(x)| > \tau]\,\mathrm{d}\tau < R \int_{\tau=0}^{B} \mathop{\mathbf{Pr}}_{x\sim\mathcal{U}(S)}[f_i^\star(x)|p_i^\star(x)| > \tau]\,\mathrm{d}\tau + \Delta B\,.$$

Using the tail-sum formula for nonnegative random variables, we get that

$$\begin{aligned}\mathop{\mathbf{E}}_{x\sim\mathcal{U}(S_{i-1})}[f_i^\star(x)|p_i^\star(x)|] &= \int_{\tau=0}^{\infty} \mathop{\mathbf{Pr}}_{x\sim\mathcal{U}(S_{i-1})}[f_i^\star(x)|p_i^\star(x)| > \tau]\,\mathrm{d}\tau \\ &= \int_{\tau=0}^{B} \mathop{\mathbf{Pr}}_{x\sim\mathcal{U}(S_{i-1})}[f_i^\star(x)|p_i^\star(x)| > \tau]\,\mathrm{d}\tau\,,\end{aligned}$$

where the second equality follows from the fact that $p_i^\star \in \mathcal{P}(f_i^\star)$, $S_{i-1} \subseteq S_0$ and $f(x)|p(x)| \le B$ for any $f \in \mathcal{F}$, $p \in \mathcal{P}(f)$ and $x \in S_0$ due to line 3. Similarly, we have

$$\begin{aligned}\mathop{\mathbf{E}}_{x\sim\mathcal{U}(S)}[f_i^\star(x)|p_i^\star(x)|] &= \int_{\tau=0}^{\infty} \mathop{\mathbf{Pr}}_{x\sim\mathcal{U}(S)}[f_i^\star(x)|p_i^\star(x)| > \tau]\,\mathrm{d}\tau \\ &\ge \int_{\tau=0}^{B} \mathop{\mathbf{Pr}}_{x\sim\mathcal{U}(S)}[f_i^\star(x)|p_i^\star(x)| > \tau]\,\mathrm{d}\tau\,,\end{aligned}$$

where the inequality follows from the fact that the integrated function is nonnegative. Putting everything together, we have

$$\begin{aligned}\mu_i^\star &= \frac{|S_{i-1}|}{|S'|}\mathop{\mathbf{E}}_{x\sim\mathcal{U}(S_{i-1})}[f_i^\star(x)p_i^\star(x)] \\ &\le \frac{|S_{i-1}|}{|S'|}\mathop{\mathbf{E}}_{x\sim\mathcal{U}(S_{i-1})}[f_i^\star(x)|p_i^\star(x)|] \\ &< R\mathop{\mathbf{E}}_{x\sim\mathcal{U}(S)}[f_i^\star(x)|p_i^\star(x)|] + \Delta B \\ &\le R\cdot\frac{2\epsilon}{2R+\epsilon} + B\cdot\frac{\epsilon^2}{B(2R+\epsilon)} && \text{(since } p_i^\star \in \mathcal{P}(f_i^\star)\text{)} \\ &= \epsilon\end{aligned}$$

Hence, we reached a contradiction since we showed that $\mu_i^\star \le \epsilon$. This concludes the proof. □

We now prove the following proposition, which bounds the number of iterations of Algorithm 2 independently of the number of input examples. This is crucial in order to show that the selector returned by Algorithm 2 belongs to a class of bounded VC dimension, so that uniform convergence arguments can be applied to show generalization.

Proposition B.2. *In the setting of Algorithm 2, it holds $t \le \lfloor 1/\Delta \rfloor$.*

Proof. If $t \neq 0$, then for any $i \in [t]$, it holds that

$$\frac{|S_{i-1}|}{|S'|}\mathop{\mathbf{Pr}}_{x\sim\mathcal{U}(S_{i-1})}[f_i^\star(x)|p_i^\star(x)| > \tau_i^\star] \ge R\mathop{\mathbf{Pr}}_{x\sim\mathcal{U}(S)}[f_i^\star(x)|p_i^\star(x)| > \tau_i^\star] + \Delta\,,$$

which implies that $|\{x \in S_{i-1} : f_i^\star(x)|p_i^\star(x)| > \tau_i^\star\}| \ge \Delta|S'|$, that is, in each iteration, at least $\Delta|S'|$ elements of S_0 (that had not been discarded in any previous iteration) are discarded. Thus, we have $\Delta|S'|t \le |S_0| \le |S'|$, which implies that $t \le \lfloor 1/\Delta \rfloor$. □

Theorem B.1 (Iterative Chow Filtering)**.** *Let any $\ell, k \in \mathbb{N}^*$, $A \ge 1$, $R > 1$, $\beta > 0$, $\epsilon, \delta \in (0,1)$ and let $\mathcal{F}$ be a class of k functions $f_1, \ldots, f_k \colon \mathbb{R}^d \to \{0,1\}$. Let $\mathcal{D}$ be any A-hypercontractive distribution on $\mathbb{R}^d$ and let $\mathcal{D}'$ be any distribution on $\mathbb{R}^d$. Let S and S' be multisets of independent examples drawn from $\mathcal{D}$ and $\mathcal{D}'$ respectively. If $|S|, |S'| \ge \mathrm{poly}(R, \beta, A^\ell, (d+1)^\ell, 1/\epsilon, \log(k/\delta)^\ell)$, then Algorithm 2, given $\mathcal{F}, S, S', \ell, R, \beta, \epsilon$ as input, runs in time $k \cdot \mathrm{poly}(|S|, |S'|, (d+1)^\ell)$ and outputs a succinct $k\sqrt{\beta R^3/\epsilon^5}\mathrm{poly}(|S|, (d+1)^\ell)$-time-computable description of a function $s \colon \mathbb{R}^d \to \{0,1\}$ that satisfies the following guarantees.*

1. *Let any $p_1, \ldots, p_k \in \mathcal{P}_{d,\ell}$ such that $\max_{i\in[k]} \mathbf{E}_{x\sim\mathcal{D}}[p_i(x)^2] \le \beta$. Then, with probability at least $1-\delta$ over S and S', we have that for any $i \in [k]$, it holds*

$$\mathop{\mathbf{E}}_{x\sim\mathcal{D}'}[f_i(x)p_i(x)s(x)] \le (R+\epsilon)(1+\epsilon)\mathop{\mathbf{E}}_{x\sim\mathcal{D}}[f_i(x)|p_i(x)|] + 2\epsilon\,.$$

2. *With probability at least $1-\delta$ over S and S', it holds*

$$\Pr_{x\sim\mathcal{D}}[s(x)=0] \le \min\left\{\frac{1+\epsilon}{R}, \frac{d_{\mathrm{TV}}(\mathcal{D},\mathcal{D}')+\epsilon}{R-1}\right\}.$$

Proof. Let

$$s, B, \Delta, t, S_0, \dots, S_t, \mu_1^\star, \dots, \mu_{t+1}^\star, f_1^\star, \dots, f_{t+1}^\star, p_1^\star, \dots, p_{t+1}^\star, \tau_1^\star, \dots, \tau_t^\star, \mathcal{P}(f_1), \dots, \mathcal{P}(f_k)$$

be defined as in the description of Algorithm 2.

First, we show that Algorithm 2 is efficient. For any $f \in \mathcal{F}$, the set $\mathcal{P}(f)$ can be described by $O(|S|)$ linear constraints plus one convex quadratic constraint over the coefficient vectors whose dimension is at most $(d+1)^\ell$. Thus, line 3 and line 5 can be implemented via convex programming in time $k \cdot \mathrm{poly}(|S|, |S'|, (d+1)^\ell)$.[10] Moreover, line 9 and line 10 can be implemented with a single pass of the input points. Finally, for any $x \in \mathbb{R}^d$, the value of the first indicator function in the expression of $s(x)$ can be computed via convex programming in time $k \cdot \mathrm{poly}\left(|S|, (d+1)^\ell\right)$ and the value of each of the other t indicator functions can be computed in time $(d+1)^\ell$. So, $s(x)$ can be computed in time $k\sqrt{\beta R^3/\epsilon^5}\mathrm{poly}\left(|S|, (d+1)^\ell\right)$.

Proof of Item 1.

We proceed by proving Item 1 of Theorem B.1. Let any $f \in \mathcal{F}$ and $p \in \mathcal{P}_{d,\ell}$ such that $\mathbf{E}_{x\sim\mathcal{D}}[p(x)^2] \le \beta$. Let $\gamma = \epsilon/(R+\epsilon)$.

First, we assume that $\mathbf{E}_{x\sim\mathcal{D}}[f(x)|p(x)|] \le \gamma$. Then, if $|S| \gtrsim R^2\beta(d+1)^{2\ell}\left(CA\log(k/\delta)\right)^{4\ell+1}/\epsilon^2$ for some sufficiently large constant $C > 0$, then, by Lemma E.3, we have that with probability at least $1-\delta/(2k)$ over S, it holds $p \in \mathcal{P}(f)$. We condition on that happening for the rest of the paragraph. Then, by definition of the selector s, it holds $s(x) = 1$ for all $x \in S_t$ and $s(x) = 0$ for all $x \in S' \setminus S_t$. Thus, we have

$$\begin{aligned}
\mathop{\mathbf{E}}_{x\sim\mathcal{U}(S')}[f(x)p(x)s(x)] &= \frac{1}{|S'|}\sum_{x\in S'} f(x)p(x)s(x) \\
&= \frac{1}{|S'|}\sum_{x\in S_t} f(x)p(x) \\
&\le \frac{1}{|S'|}\sum_{x\in S_t} f_{t+1}^\star(x)p_{t+1}^\star(x) \\
&\le \epsilon\,.
\end{aligned}$$

The penultimate inequality follows from the fact that $p \in \mathcal{P}(f)$ and from the fact that the pair $(f_{t+1}^\star, p_{t+1}^\star)$ is a maximizer of the optimization problem defined in line 5 of Algorithm 2 and the final inequality follows from the fact that $t+1$ is the last iteration of Algorithm 2 (thus, the terminating condition $\mu_{t+1}^\star \le \epsilon$ of line 6 of Algorithm 2 holds). Now, consider the class

$$\mathcal{H} = \left\{ x \mapsto \mathbf{1}\left\{\max_{g\in\mathcal{F}}\max_{q\in\mathcal{P}(g)} g(x)|q(x)| \le B\right\}\prod_{i=1}^{\lfloor 1/\Delta\rfloor}\mathbf{1}\{f_i(x)|p_i(x)| \le \tau_i\} : \begin{array}{l} f_i \in \mathcal{F} \\ p_i \in \mathcal{P}_{d,\ell} \\ \tau_i \ge 0 \end{array} \right\}.$$

Notice that $\mathcal{H}$ does not depend on S'. Also, due to Proposition B.2, we have that $t \le \lfloor 1/\Delta\rfloor$, which implies that $s \in \mathcal{H}$. Moreover, each function in $\mathcal{H}$ is a logical AND of $\lfloor 1/\Delta\rfloor + 1$ functions, the first of which is a fixed function (after conditioning on S) and the other $\lfloor 1/\Delta\rfloor$, by Lemma E.4, belong to a class of VC dimension at most $O\left((d+1)^\ell\ell\log(d)\log(k)\right)$. Thus, by Fact A.2, we have that $\mathrm{VCdim}(\mathcal{H}) \le O\left((d+1)^\ell\ell\log(d)\log(k)\right)\lfloor 1/\Delta\rfloor\log(\lfloor 1/\Delta\rfloor)$. Moreover, since $\sup_{h\in\mathcal{H}}\sup_{x\in\mathbb{R}^d} h(x)f(x)|p(x)| \le B$, if $|S'| \gtrsim B^4\mathrm{VCdim}(\mathcal{H})\log(1/\epsilon)\log(k/\delta)/\epsilon^8$, Lemma E.5 (uniform convergence) implies that with probability at least $1-\delta/(2k)$ over S', it holds

$$\mathop{\mathbf{E}}_{x\sim\mathcal{D}'}[f(x)p(x)s(x)] \le \mathop{\mathbf{E}}_{x\sim\mathcal{U}(S')}[f(x)p(x)s(x)] + \epsilon^2 \le \epsilon + \epsilon^2\,.$$

Thus, f and p satisfy the condition of Item 1 with probability at least $1-\delta/k$.

[10] In lines 3 and 5, the algorithm solves one convex optimization problem for each $f \in \mathcal{F}$.

Now, we assume that $\mathbf{E}_{x\sim\mathcal{D}}[f(x)|p(x)|] > \gamma$. We define the polynomial $\widetilde{p} = \gamma p/\mathbf{E}_{x\sim\mathcal{D}}[f(x)|p(x)|]$. Then, $\mathbf{E}_{x\sim\mathcal{D}}[\widetilde{p}(x)^2] < \beta$ and $\mathbf{E}_{x\sim\mathcal{D}}[f(x)\widetilde{p}(x)] = \gamma$. If $|S|$ is sufficiently large, by Lemma E.3, we have that with probability at least $1-\delta/(2k)$ over S, it holds $\widetilde{p} \in \mathcal{P}(f)$. We condition on this high probability event for the rest of the argument. Following the same steps as in the previous case, we have that with probability at least $1-\delta/(2k)$ over S', it holds $\mathbf{E}_{x\sim\mathcal{D}'}[f(x)\widetilde{p}(x)s(x)] \le \epsilon + \epsilon^2$, which implies that

$$\mathop{\mathbf{E}}_{x\sim\mathcal{D}'}[f(x)p(x)s(x)] \le (R+\epsilon)(1+\epsilon)\mathop{\mathbf{E}}_{x\sim\mathcal{D}}[f(x)|p(x)|]\,.$$

Thus, f and p again satisfy the condition of Item 1 with probability at least $1-\delta/k$. Finally, a union bound over all pairs (f_i, p_i) for $i \in [k]$, implies that with probability at least $1-\delta$, it holds

$$\mathop{\mathbf{E}}_{x\sim\mathcal{D}'}[f_i(x)p_i(x)s(x)] \le (R+\epsilon)(1+\epsilon)\mathop{\mathbf{E}}_{x\sim\mathcal{D}}[f_i(x)|p_i(x)|] + 2\epsilon$$

for all $i \in [k]$.

Proof of Item 2. Now, we will prove Item 2 of Theorem B.1. In the trivial case $t = 0$, we have $s(x) = \mathbf{1}\{\max_{f\in\mathcal{F}}\max_{p\in\mathcal{P}(f)} f(x)|p(x)| \le B\}$. So, we have

$$\mathop{\mathbf{Pr}}_{x\sim\mathcal{D}}[s(x)=0] = \mathop{\mathbf{Pr}}_{x\sim\mathcal{D}}\left[\max_{f\in\mathcal{F}}\max_{p\in\mathcal{P}(f)} f(x)|p(x)| > B\right].$$

We now bound the right hand side in the above equality. Let $\mathcal{P} = \{p \in \mathcal{P}_{d,\ell} : \mathbf{E}_{x\sim\mathcal{U}(S)}[p(x)^2] \le 2\beta\}$ and recall that $B = 2\sqrt{2R(d+1)^\ell\beta/\epsilon}$. Hence, if $|S| \gtrsim (d+1)^{2\ell}(4A\log(12/\delta))^{4\ell+1}$, by Lemma E.2, we have that with probability at least $1-\delta/3$ over S, it holds

$$\mathop{\mathbf{Pr}}_{x\sim\mathcal{D}}\left[\max_{f\in\mathcal{F}}\max_{p\in\mathcal{P}(f)} f(x)|p(x)| > B\right] \le \mathop{\mathbf{Pr}}_{x\sim\mathcal{D}}\left[\max_{p\in\mathcal{P}}|p(x)| > B\right] \le \epsilon/(2R)\,,$$

which directly implies Item 2 of Theorem B.1.

We now consider the case $t \ge 1$. Then, by definition of the selector we have

$$\mathop{\mathbf{Pr}}_{x\sim\mathcal{D}}[s(x)=0] \le \mathop{\mathbf{Pr}}_{x\sim\mathcal{D}}\left[\max_{f\in\mathcal{F}}\max_{p\in\mathcal{P}(f)} f(x)|p(x)| > B\right] + \sum_{i=1}^{t}\mathop{\mathbf{Pr}}_{x\sim\mathcal{D}}[f_i^\star(x)|p_i^\star(x)| > \tau_i^\star]\,.$$

We bound the right term of the right hand side in the above inequality. If $|S| \gtrsim (d+1)^\ell \ell\log(d)\log(k)\log(R/\Delta)\log(1/\delta)R^2/\Delta^2$, by Lemma E.4 and Fact A.4 (uniform convergence), we have that with probability at least $1-\delta/3$ over S, for all $i \in [t]$, it holds

$$\mathop{\mathbf{Pr}}_{x\sim\mathcal{D}}[f_i^\star(x)|p_i^\star(x)| > \tau_i^\star] \le \mathop{\mathbf{Pr}}_{x\sim\mathcal{U}(S)}[f_i^\star(x)|p_i^\star(x)| > \tau_i^\star] + \frac{\Delta}{R}\,.$$

Hence, summing over all $i \in [t]$, we get that

$$\begin{aligned}
\sum_{i=1}^{t}\mathop{\mathbf{Pr}}_{x\sim\mathcal{D}}[f_i^\star(x)|p_i^\star(x)| > \tau_i^\star] &\le \sum_{i=1}^{t}\left(\mathop{\mathbf{Pr}}_{x\sim\mathcal{U}(S)}[f_i^\star(x)|p_i^\star(x)| > \tau_i^\star] + \frac{\Delta}{R}\right)\\
&\le \frac{1}{R}\sum_{i=1}^{t}\frac{|S_{i-1}|}{|S'|}\mathop{\mathbf{Pr}}_{x\sim\mathcal{U}(S_{i-1})}[f_i^\star(x)|p_i^\star(x)| > \tau_i^\star]\\
&= \frac{1}{R|S'|}\sum_{i=1}^{t}\sum_{x\in S_{i-1}}\mathbf{1}\{f_i^\star(x)|p_i^\star(x)| > \tau_i^\star\}\\
&= \frac{1}{R|S'|}\sum_{i=1}^{t}\sum_{x\in S_{i-1}\setminus S_i}\mathbf{1}\{f_i^\star(x)|p_i^\star(x)| > \tau_i^\star\}\,.
\end{aligned}$$

In the second line, we used the fact that the condition of line 9 of Algorithm 2 is satisfied. In the fourth line, we used the fact that for all $i \in [t]$, we have $\sum_{x\in S_i}\mathbf{1}\{f_i^\star(x)|p_i^\star(x)| > \tau_i^\star\} = 0$ by definition of

S_i (see line 10 of Algorithm 2). Using the fact that $\mathbf{1}\{f_i^\star(x)|p_i^\star(x)| > \tau_i^\star\} \le \mathbf{1}\{s(x) = 0\}$ for all $i \in [t]$ and $x \in \mathbb{R}^d$, we get

$$\begin{aligned}
\sum_{i=1}^{t} \mathop{\mathbf{Pr}}_{x\sim\mathcal{D}}[f_i^\star(x)|p_i^\star(x)| > \tau_i^\star] &\le \frac{1}{R|S'|} \sum_{i=1}^{t} \sum_{x\in S_{i-1}\setminus S_i} \mathbf{1}\{s(x) = 0\} \\
&= \frac{1}{R|S'|} \sum_{x\in S_0} \mathbf{1}\{s(x) = 0\} \\
&\le \frac{1}{R|S'|} \sum_{x\in S'} \mathbf{1}\{s(x) = 0\} \\
&= \frac{1}{R} \mathop{\mathbf{Pr}}_{x\sim\mathcal{U}(S')}[s(x) = 0]\,.
\end{aligned}$$

Trivially, the quantity above is upper bounded by $1/R$, which implies that $\mathbf{Pr}_{x\sim\mathcal{D}}[s(x) = 0] \le (1+\epsilon)/R$. We will now provide an additional upper bound for the same quantity. If $|S'| \gtrsim \mathrm{VCdim}(\mathcal{H}) \log(1/\epsilon) \log(1/\delta)/\epsilon^2$, using Fact A.4 (uniform convergence), we get that with probability at least $1 - \delta/3$, it holds

$$\mathop{\mathbf{Pr}}_{x\sim\mathcal{U}(S')}[s(x) = 0] \le \mathop{\mathbf{Pr}}_{x\sim\mathcal{D}'}[s(x) = 0] + \epsilon/2 \le d_{\mathrm{TV}}(\mathcal{D},\mathcal{D}') + \mathop{\mathbf{Pr}}_{x\sim\mathcal{D}}[s(x) = 0] + \epsilon/2\,.$$

Thus, we have

$$\mathop{\mathbf{Pr}}_{x\sim\mathcal{D}}[s(x) = 0] \le \frac{1}{R}\left(d_{\mathrm{TV}}(\mathcal{D},\mathcal{D}') + \mathop{\mathbf{Pr}}_{x\sim\mathcal{D}}[s(x) = 0] + \epsilon\right)\,,$$

which implies (after rearranging) that

$$\mathop{\mathbf{Pr}}_{x\sim\mathcal{D}}[s(x) = 0] \le \frac{d_{\mathrm{TV}}(\mathcal{D},\mathcal{D}') + \epsilon}{R-1}\,.$$

Finally, a union bound implies that the condition of Item 2 holds with probability at least $1 - \delta$. This concludes the proof of Theorem B.1. □

C PQ Learning

In this section, we prove Theorem 4.1, which we restate here as Theorem C.1 for convenience.

Theorem C.1 (PQ Learning via $\mathcal{L}_1$ Sandwiching)**.** *Let any $\epsilon, \delta, \eta \in (0,1)$ and $A \ge 1$. Let $\mathcal{X} \subseteq \mathbb{R}^d$ and let $\mathcal{D}$ be any A-hypercontractive distribution on $\mathcal{X}$. Let $\mathcal{C} \subseteq \{0,1\}^{\mathcal{X}}$ be a concept class and let $\ell = \mathcal{L}_1(\mathcal{C}, \mathcal{D}, \epsilon\eta/8)$. Then, there exists an (ϵ,δ,η)-PQ learner for $\mathcal{C}$ with respect to $\mathcal{D}$ with sample complexity and runtime* $\mathrm{poly}((d+1)^\ell, A^\ell, 1/\epsilon, 1/\eta, \log(1/\delta)^\ell)$.

By combining the above result with existing sandwiching degree bounds from prior work (see Appendix A.3), we obtain the following corollaries.

Corollary C.1. *Let any $\epsilon, \delta, \eta \in (0,1)$. Let $\mathcal{C}$ be the class of depth-t, size-s AC^0 circuits in $\{\pm1\}^d$ and let $\ell = O(\log(s))^{O(t)} \log(1/(\epsilon\eta))$. Then, there exists an (ϵ,δ,η)-PQ learner for $\mathcal{C}$ with respect to $\mathcal{U}(\{\pm1\}^d)$ with sample complexity and runtime* $\mathrm{poly}((d+1)^\ell, 1/\epsilon, 1/\eta, \log(1/\delta)^\ell)$.

Corollary C.2. *Let any $\epsilon, \delta, \eta \in (0,1)$. Let $\mathcal{C}$ be the class of depth-t, size-s decision trees of halfspaces in $\mathbb{R}^d$ and let $\ell = \tilde{O}(t^4 s^2/(\epsilon\eta)^2)$. Then, there exists an (ϵ,δ,η)-PQ learner for $\mathcal{C}$ with respect to $\mathcal{N}_d$ (resp. $\mathcal{U}(\{\pm1\}^d)$) with sample complexity and runtime* $\mathrm{poly}((d+1)^\ell, 1/\epsilon, 1/\eta, \log(1/\delta)^\ell)$.

Corollary C.3. *Let any $\epsilon, \delta, \eta \in (0,1)$. Let $\mathcal{C}$ be the class of degree-2 PTFs in $\mathbb{R}^d$ and let $\ell = \tilde{O}(1/(\epsilon\eta)^8)$ (resp. $\ell = \tilde{O}(1/(\epsilon\eta)^9)$). Then, there exists an (ϵ,δ,η)-PQ learner for $\mathcal{C}$ with respect to $\mathcal{N}_d$ (resp. $\mathcal{U}(\{\pm1\}^d)$) with sample complexity and runtime* $\mathrm{poly}((d+1)^\ell, 1/\epsilon, 1/\eta, \log(1/\delta)^\ell)$.

Corollary C.4. *Let any $\epsilon, \delta, \eta \in (0,1)$. Let $\mathcal{C}$ be the class of degree-k PTFs in $\mathbb{R}^d$ and let $\ell = O_k((\epsilon\eta)^{-4k\cdot 7^k})$. Then, there exists an (ϵ,δ,η)-PQ learner for $\mathcal{C}$ with respect to $\mathcal{N}_d$ with sample complexity and runtime* $\mathrm{poly}((d+1)^\ell, 1/\epsilon, 1/\eta, \log(1/\delta)^\ell)$.

Corollary C.5. *Let any $\epsilon, \delta, \eta \in (0,1)$. Let $\mathcal{C}$ be the class of arbitrary functions of k halfspaces in $\mathbb{R}^d$, let $\mathcal{D}$ be any isotropic log-concave distribution on $\mathbb{R}^d$ and let $\ell = \exp((\log(\log(k)/(\epsilon\eta)))^{O(k)}/(\epsilon\eta)^4)$. Then, there exists an (ϵ, δ, η)-PQ learner for $\mathcal{C}$ with respect to $\mathcal{D}$ with sample complexity and runtime* $\mathrm{poly}((d+1)^\ell, 1/\epsilon, 1/\eta, \log(1/\delta)^\ell)$.

We will now prove Theorem C.1.

Proof of Theorem C.1. Let $\mathcal{D}^{\mathrm{train}}$ be any training distribution on $\mathcal{X} \times \{0,1\}$ whose marginal on $\mathcal{X}$ is $\mathcal{D}$. Let $\mathcal{D}^{\mathrm{test}}$ be any test distribution on $\mathcal{X} \times \{0,1\}$ and let $\mathcal{D}'$ denote its marginal on $\mathcal{X}$. Let $\mathsf{opt}_{\mathrm{train}} = \inf_{f \in \mathcal{C}} \mathbf{Pr}_{(x,y)\sim\mathcal{D}^{\mathrm{train}}}[f(x) \neq y]$. The algorithm does the following:

1. It draws a set S_{train} of $\mathrm{poly}\left((d+1)^\ell, 1/\epsilon, 1/\eta, \log(1/\delta)\right)$ i.i.d. examples from $\mathcal{D}^{\mathrm{train}}$.
2. It runs the degree-ℓ $\mathcal{L}_1$ polynomial regression algorithm of Theorem E.1 with input S_{train} to get a classifier $\hat{f}\colon \mathbb{R}^d \to \{0,1\}$.
3. It draws sets S, S' of $\mathrm{poly}((d+1)^\ell, A^\ell, 1/\epsilon, 1/\eta, \log(1/\delta)^\ell)$ i.i.d. examples from $\mathcal{D}$ and $\mathcal{D}'$ respectively.
4. It runs Algorithm 1 with input $\hat{f}, S, S', \ell, R' = 1/\eta + \epsilon/96, \beta = 4(2A)^{2\ell}, \epsilon' = \epsilon\eta/96$ to get a selector $s\colon \mathcal{X} \to \{0,1\}$.
5. It returns $(\hat{f}, s)$.

First, since $|S_{\mathrm{train}}|$ is sufficiently large, by Theorem E.1, we can ensure that with probability at least $1 - \delta/2$, it holds $\mathbf{Pr}_{(x,y)\sim\mathcal{D}^{\mathrm{train}}}[\hat{f}(x) \neq y] \leq \mathsf{opt}_{\mathrm{train}} + \epsilon\eta/2$.

Now, let any $f^\star \in \mathcal{C}$ such that $\mathrm{err}_{\mathrm{train}}(f^\star) = \lambda_{\mathrm{train}}$ and $\mathrm{err}_{\mathrm{test}}(f^\star) = \lambda_{\mathrm{test}}$ and let $p_{\mathrm{up}}, p_{\mathrm{down}}$ be ϵ''-approximate $\mathcal{L}_1$ sandwiching polynomials for $f^\star$ under $\mathcal{D}$ of degree at most ℓ, where $\epsilon'' = \epsilon\eta/8$. Then, p_{up} and $\bar{p}_{\mathrm{down}}$ are nonnegative. Moreover, using the fact that $\mathcal{D}$ is A-hypercontractive and the fact that $\mathbf{E}_{x\sim\mathcal{D}}[p_{\mathrm{up}}(x) - f^\star(x)] \leq \epsilon'' \leq 1$, we get that

$$\begin{aligned}
\mathop{\mathbf{E}}_{x\sim\mathcal{D}}[p_{\mathrm{up}}(x)^2] &\leq (2A)^{2\ell} \mathop{\mathbf{E}}_{x\sim\mathcal{D}}[p_{\mathrm{up}}(x)]^2 \\
&\leq (2A)^{2\ell} \left(\mathop{\mathbf{E}}_{x\sim\mathcal{D}}[p_{\mathrm{up}}(x) - f^\star(x)] + \mathop{\mathbf{E}}_{x\sim\mathcal{D}}[f^\star(x)] \right)^2 \\
&< 4(2A)^{2\ell}
\end{aligned}$$

and, similarly, $\mathbf{E}_{x\sim\mathcal{D}}[\bar{p}_{\mathrm{down}}(x)^2] < 4(2A)^{2\ell}$. Since $|S|$ and $|S'|$ are sufficiently large, applying Theorem 3.1 (with additive error ϵ', multiplicative slack R' and polynomial variance bound $\beta = 4(2A)^{2\ell}$), we get that with probability at least $1 - \delta/2$ we have that $\mathbf{Pr}_{x\sim\mathcal{D}}[s(x) = 0] \leq (1+\epsilon')/R'$ and

$$\begin{aligned}
\mathop{\mathbf{E}}_{x\sim\mathcal{D}'}[\hat{f}(x)\bar{p}_{\mathrm{down}}(x)s(x)] &\leq (R' + \epsilon')(1+\epsilon') \mathop{\mathbf{E}}_{x\sim\mathcal{D}}[\hat{f}(x)\bar{p}_{\mathrm{down}}(x)] + 2\epsilon' \\
&\leq R' \mathop{\mathbf{E}}_{x\sim\mathcal{D}}[\hat{f}(x)\bar{p}_{\mathrm{down}}(x)] + 8R'\epsilon'
\end{aligned}$$

and

$$\begin{aligned}
\mathop{\mathbf{E}}_{x\sim\mathcal{D}'}[\bar{\hat{f}}(x)p_{\mathrm{up}}(x)s(x)] &\leq (R' + \epsilon')(1+\epsilon') \mathop{\mathbf{E}}_{x\sim\mathcal{D}}[\bar{\hat{f}}(x)p_{\mathrm{up}}(x)] + 2\epsilon' \\
&\leq R' \mathop{\mathbf{E}}_{x\sim\mathcal{D}}[\bar{\hat{f}}(x)p_{\mathrm{up}}(x)] + 8R'\epsilon' \,,
\end{aligned}$$

where we used the fact that $\mathbf{E}_{x\sim\mathcal{D}}[\hat{f}(x)\bar{p}_{\mathrm{down}}(x)] \leq \mathbf{E}_{x\sim\mathcal{D}}[\bar{p}_{\mathrm{down}}(x)] = \mathbf{E}_{x\sim\mathcal{D}}[f^\star(x) - p_{\mathrm{down}}(x)] + \mathbf{E}_{x\sim\mathcal{D}}[\bar{f}^\star(x)] \leq 2$ and, similarly, $\mathbf{E}_{x\sim\mathcal{D}}[\bar{\hat{f}}(x)p_{\mathrm{up}}(x)] \leq 2$ in the last inequality of each of the above lines. Thus, for $R' = 1/\eta + \epsilon/96$ and $\epsilon' = \epsilon\eta/96$, we have that $\mathbf{Pr}_{x\sim\mathcal{D}}[s(x) = 0] \leq \eta$ and

$$\begin{aligned}
\mathop{\mathbf{E}}_{x\sim\mathcal{D}'}[\hat{f}(x)\bar{p}_{\mathrm{down}}(x)s(x)] &\leq \eta^{-1} \mathop{\mathbf{E}}_{x\sim\mathcal{D}}[\hat{f}(x)\bar{p}_{\mathrm{down}}(x)] + \epsilon/8 \\
\mathop{\mathbf{E}}_{x\sim\mathcal{D}'}[\bar{\hat{f}}(x)p_{\mathrm{up}}(x)s(x)] &\leq \eta^{-1} \mathop{\mathbf{E}}_{x\sim\mathcal{D}}[\bar{\hat{f}}(x)p_{\mathrm{up}}(x)] + \epsilon/8 \,.
\end{aligned}$$

Hence, by Lemma 4.1 (with $R = 1/\eta$), given that $\mathbf{Pr}_{(x,y)\sim\mathcal{D}^{\mathrm{train}}}[\hat{f}(x) \neq y] \leq \mathsf{opt}_{\mathrm{train}} + \epsilon\eta/2$, we have

$$\begin{aligned}\mathop{\mathbf{Pr}}_{(x,y)\sim\mathcal{D}^{\mathrm{test}}}[\hat{f}(x) \neq y \wedge s(x) = 1] &\leq \lambda_{\mathrm{test}} + \eta^{-1}\left(\lambda_{\mathrm{train}} + \mathop{\mathbf{Pr}}_{(x,y)\sim\mathcal{D}^{\mathrm{train}}}[\hat{f}(x) \neq y]\right) + \epsilon/2 \\ &\leq \lambda_{\mathrm{test}} + \eta^{-1}\left(\lambda_{\mathrm{train}} + \mathsf{opt}_{\mathrm{train}}\right) + \epsilon\,.\end{aligned}$$

A union bound implies that, with probability at least $1 - \delta$, we have

1. $\mathbf{Pr}_{(x,y)\sim\mathcal{D}^{\mathrm{test}}}[\hat{f}(x) \neq y \wedge s(x) = 1] \leq \lambda_{\mathrm{test}} + (\lambda_{\mathrm{train}} + \mathsf{opt}_{\mathrm{train}})/\eta + \epsilon$ and
2. $\mathbf{Pr}_{x\sim\mathcal{D}}[s(x) = 0] \leq \eta$,

which concludes the proof. □

D Tolerant TDS Learning

In this section, we provide our result for tolerant TDS learning. As before, we consider a feature space $\mathcal{X} \subseteq \mathbb{R}^d$, binary labels $\mathcal{Y} = \{0, 1\}$ and the learner is assumed to have access to labeled examples from a training distribution $\mathcal{D}^{\mathrm{train}}$ over $\mathcal{X} \times \mathcal{Y}$, as well as unlabeled examples from the marginal $\mathcal{D}^{\mathrm{test}}_{\mathcal{X}}$ of a test distribution $\mathcal{D}^{\mathrm{test}}$. The formal definition of tolerant TDS learning of a concept class $\mathcal{C} \subseteq \{\mathcal{X} \to \mathcal{Y}\}$ with respect to a distribution $\mathcal{D}$ over $\mathcal{X}$ is as follows.

Definition D.1 (Tolerant TDS Learning [GSSV24, KSV24a])**.** Let $\delta \in (0, 1)$ and $\theta \in [0, 1]$. An algorithm $\mathcal{A}$ is a TDS learner for $\mathcal{C}$ with respect to $\mathcal{D}$ up to error γ, tolerance θ and failure probability δ if the following holds whenever $\mathcal{D}^{\mathrm{train}}_{\mathcal{X}} = \mathcal{D}$. Given sufficiently many labeled examples from $\mathcal{D}^{\mathrm{train}}$ and sufficiently many unlabeled examples from $\mathcal{D}^{\mathrm{test}}_{\mathcal{X}}$, the algorithm either accepts and outputs a classifier $h\colon \mathcal{X} \to \mathcal{Y}$ or rejects. Moreover:

1. *Soundness.* If $\mathcal{A}$ accepts, then $\mathbf{Pr}_{(x,y)\sim\mathcal{D}^{\mathrm{test}}}[h(x) \neq y] \leq \gamma$ with probability at least $1 - \delta$.
2. *Completeness.* If $d_{\mathrm{TV}}(\mathcal{D}^{\mathrm{train}}_{\mathcal{X}}, \mathcal{D}^{\mathrm{test}}_{\mathcal{X}}) \leq \theta$, then $\mathcal{A}$ accepts with probability at least $1 - \delta$.

The error benchmark γ may depend on the particular instance of the problem, i.e., the distributions $\mathcal{D}^{\mathrm{train}}, \mathcal{D}^{\mathrm{test}}$, but its value is not necessarily known to the learner.

We first note that our result for PQ learning already implies that $\mathcal{L}_1$ sandwiching suffices for tolerant TDS learning due to the following result of [GSSV24].

Proposition D.1 (PQ Learning Implies Tolerant TDS Learning [GSSV24])**.** *Let $\gamma, \delta \in (0, 1)$. Let $\mathcal{A}$ be an algorithm that PQ-learns $\mathcal{C}$ with respect to $\mathcal{D}$ up to error γ, rejection rate η and failure probability δ.*[11] *Then, for any $\epsilon \in (0, 1)$ and $\theta \in [0, 1)$, there exists an algorithm that TDS-learns $\mathcal{C}$ with respect to $\mathcal{D}$ up to error $\gamma + \eta + \theta + \epsilon$, tolerance θ and failure probability 2δ that calls $\mathcal{A}$ once and uses $O(\log(1/\delta)/\epsilon^2)$ additional examples and evaluations of the selector given by $\mathcal{A}$.*

Corollary D.1. *Let any $\epsilon, \delta, \eta \in (0, 1)$ and $\theta \in [0, 1)$. Let $\mathcal{X} \subseteq \mathbb{R}^d$ and let $\mathcal{D}$ be any A-hypercontractive distribution on $\mathcal{X}$. Let $\mathcal{C} \subseteq \{0, 1\}^{\mathcal{X}}$ be a concept class and let $\ell = \mathcal{L}_1(\mathcal{C}, \mathcal{D}, \epsilon\eta/9)$. Then, there exists a TDS learner for $\mathcal{C}$ with respect to $\mathcal{D}$ up to error $\lambda_{\mathrm{test}} + (\lambda_{\mathrm{train}} + \mathsf{opt}_{\mathrm{train}})/\eta + \eta + \theta + \epsilon$, tolerance θ and failure probability δ with sample complexity and runtime $\mathrm{poly}((d+1)^{\ell}, A^{\ell}, 1/\epsilon, 1/\eta, \log(1/\delta)^{\ell})$.*

Notice that setting $\eta = \theta + \epsilon$ in Corollary D.1, yields a TDS learner with error $\lambda_{\mathrm{test}} + \frac{\lambda_{\mathrm{train}} + \mathsf{opt}_{\mathrm{train}}}{\theta+\epsilon} + 2\theta + 2\epsilon$ and tolerance θ. To obtain an error bound with improved dependence in $\lambda_{\mathrm{train}} + \mathsf{opt}_{\mathrm{train}}$, we provide a TDS learner that does not use a PQ learner as a black box, but directly invokes the ICF algorithm.

Our TDS learner (see Theorem D.1 for a detailed description) first learns a classifier $\hat{f}$ via $\mathcal{L}_1$ polynomial regression on a sample from $\mathcal{D}^{\mathrm{train}}$, then runs ICF using $\hat{f}$ and unlabeled data from

[11] Here, we mean that $\mathcal{A}$ outputs some $h, s\colon \mathcal{X} \to \{0, 1\}$ such that, with probability at least $1 - \delta$, it holds $\mathbf{Pr}_{(x,y)\sim\mathcal{D}^{\mathrm{test}}}[h(x) \neq y \wedge s(x) = 1] \leq \gamma$ and $\mathbf{Pr}_{x\sim\mathcal{D}^{\mathrm{train}}_{\mathcal{X}}}[s(x) = 0] \leq \eta$.

$\mathcal{D}^{\mathrm{train}}_{\mathcal{X}}$ and $\mathcal{D}^{\mathrm{test}}_{\mathcal{X}}$ as input to obtain a selector and, finally accepts or rejects depending on the selector's empirical rejection rate on unlabeled data from $\mathcal{D}^{\mathrm{test}}_{\mathcal{X}}$. A key technical ingredient in the analysis of our algorithm is the improved bound of Theorem B.1 on the rejection rate of the selector, which upper bounds the rejection rate by a multiple of the total variation distance between the training and test marginals.

Finally, we remark that our tolerant TDS learner is similar in spirit to the tolerant TDS learner of [GSSV24]. The difference is that [GSSV24] use $\mathcal{L}_2$ polynomial regression to learn the classifier $\hat{f}$. Second, instead of ICF, their algorithm uses their spectral outlier removal method (which enforces their $\mathcal{L}_2$ sandwiching assumptions) to obtain a selector.

Theorem D.1 (Tolerant TDS Learning via $\mathcal{L}_1$ Sandwiching)**.** *Let any $\epsilon, \delta \in (0,1)$, $\theta \in [0,1)$, $R > 1$ and $A \geq 1$. Let $\mathcal{X} \subseteq \mathbb{R}^d$ and let $\mathcal{D}$ be any A-hypercontractive distribution on $\mathcal{X}$. Let $\mathcal{C} \subseteq \{0,1\}^{\mathcal{X}}$ be a concept class and let $\ell = \mathcal{L}_1(\mathcal{C}, \mathcal{D}, \epsilon/(16R))$. Then, there is an algorithm that TDS-learns $\mathcal{C}$ with respect to $\mathcal{D}$ up to error $\lambda_{\mathrm{test}} + R(\lambda_{\mathrm{train}} + \mathsf{opt}_{\mathrm{train}}) + R(R-1)^{-1}\theta + \epsilon$, tolerance θ and failure probability δ in time $\mathrm{poly}((d+1)^{\ell}, A^{\ell}, R(R-1)^{-1}/\epsilon, \log(1/\delta)^{\ell})$.*

Remark D.1. *Setting $R = 1 + \max\{\sqrt{\theta/2}, \epsilon/9\}$ above, we get that for any class $\mathcal{C}$, there exists an algorithm that TDS-learns $\mathcal{C}$ with respect to $\mathcal{D}$ up to error $\lambda + \mathsf{opt}_{\mathrm{train}} + 4\sqrt{\theta} + \epsilon$, tolerance θ and failure probability δ in time $\mathrm{poly}((d+1)^{\ell}, A^{\ell}, 1/\epsilon, \log(1/\delta)^{\ell})$, where $\ell = \mathcal{L}_1(\mathcal{C}, \mathcal{D}, \epsilon/64)$.*

By combining Remark D.1 with existing sandwiching degree bounds from prior work (see Appendix A.3), we obtain the following corollaries.

Corollary D.2. *Let any $\epsilon, \delta \in (0,1)$ and $\theta \in [0,1)$. Let $\mathcal{C}$ be the class of depth-t, size-s AC^0 circuits in $\{\pm 1\}^d$ and let $\ell = O(\log(s))^{O(t)} \log(1/(\epsilon))$. Then, there is an algorithm that TDS-learns $\mathcal{C}$ with respect to $\mathcal{U}(\{\pm 1\}^d)$ up to error $\lambda + \mathsf{opt}_{\mathrm{train}} + 4\sqrt{\theta} + \epsilon$, tolerance θ and failure probability δ in time $\mathrm{poly}((d+1)^{\ell}, 1/\epsilon, \log(1/\delta)^{\ell})$.*

Corollary D.3. *Let any $\epsilon, \delta \in (0,1)$ and $\theta \in [0,1)$. Let $\mathcal{C}$ be the class of depth-t, size-s decision trees of halfspaces in $\mathbb{R}^d$ and let $\ell = \tilde{O}(t^4 s^2/\epsilon^2)$. Then, there is an algorithm that TDS-learns $\mathcal{C}$ with respect to $\mathcal{N}_d$ (resp. $\mathcal{U}(\{\pm 1\}^d)$) up to error $\lambda + \mathsf{opt}_{\mathrm{train}} + 4\sqrt{\theta} + \epsilon$, tolerance θ and failure probability δ in time $\mathrm{poly}((d+1)^{\ell}, 1/\epsilon, \log(1/\delta)^{\ell})$.*

Corollary D.4. *Let any $\epsilon, \delta \in (0,1)$ and $\theta \in [0,1)$. Let $\mathcal{C}$ be the class of degree-2 PTFs in $\mathbb{R}^d$ and let $\ell = \tilde{O}(1/\epsilon^8)$ (resp. $\ell = \tilde{O}(1/\epsilon^9)$). Then, there is an algorithm that TDS-learns $\mathcal{C}$ with respect to $\mathcal{N}_d$ (resp. $\mathcal{U}(\{\pm 1\}^d)$) up to error $\lambda + \mathsf{opt}_{\mathrm{train}} + 4\sqrt{\theta} + \epsilon$, tolerance θ and failure probability δ in time $\mathrm{poly}((d+1)^{\ell}, 1/\epsilon, \log(1/\delta)^{\ell})$.*

Corollary D.5. *Let any $\epsilon, \delta \in (0,1)$ and $\theta \in [0,1)$. Let $\mathcal{C}$ be the class of degree-k PTFs in $\mathbb{R}^d$ and let $\ell = O_k(\epsilon^{-4k \cdot 7^k})$. Then, there is an algorithm that TDS-learns $\mathcal{C}$ with respect to $\mathcal{N}_d$ up to error $\lambda + \mathsf{opt}_{\mathrm{train}} + 4\sqrt{\theta} + \epsilon$, tolerance θ and failure probability δ in time $\mathrm{poly}((d+1)^{\ell}, 1/\epsilon, \log(1/\delta)^{\ell})$.*

Corollary D.6. *Let any $\epsilon, \delta \in (0,1)$ and $\theta \in [0,1)$. Let $\mathcal{C}$ be the class of arbitrary functions of k halfspaces in $\mathbb{R}^d$, let $\mathcal{D}$ be any isotropic log-concave distribution on $\mathbb{R}^d$ and let $\ell = \exp((\log(\log(k)/\epsilon))^{O(k)}/\epsilon^4)$. Then, there is an algorithm that TDS-learns $\mathcal{C}$ with respect to $\mathcal{D}$ up to error $\lambda + \mathsf{opt}_{\mathrm{train}} + 4\sqrt{\theta} + \epsilon$, tolerance θ and failure probability δ in time $\mathrm{poly}((d+1)^{\ell}, 1/\epsilon, \log(1/\delta)^{\ell})$.*

We will now prove Theorem D.1.

Proof of Theorem D.1. Let $\mathcal{D}^{\mathrm{train}}$ be any training distribution on $\mathcal{X} \times \{0,1\}$ whose marginal on $\mathcal{X}$ is $\mathcal{D}$. Let $\mathcal{D}^{\mathrm{test}}$ be any test distribution on $\mathcal{X} \times \{0,1\}$ and let $\mathcal{D}'$ denote its marginal on $\mathcal{X}$. Let $\mathsf{opt}_{\mathrm{train}} = \inf_{f \in \mathcal{C}} \mathbf{Pr}_{(x,y) \sim \mathcal{D}^{\mathrm{train}}}[f(x) \neq y]$. The algorithm does the following:

1. It draws a set S_{train} of $\mathrm{poly}\left((d+1)^{\ell}, R/\epsilon, \log(1/\delta)\right)$ i.i.d. examples from $\mathcal{D}^{\mathrm{train}}$ and a set X_{test} of $O(\log(1/\delta)/\epsilon^2)$ i.i.d. examples from $\mathcal{D}'$.
2. It runs the degree-ℓ $\mathcal{L}_1$ polynomial regression algorithm of Theorem E.1 with input S_{train} to get a classifier $\hat{f} \colon \mathbb{R}^d \to \{0,1\}$.

3. It draws sets S, S' of $\mathrm{poly}((d+1)^\ell, A^\ell, 1/\epsilon, R(R-1)^{-1}, \log(1/\delta)^\ell)$ i.i.d. examples from $\mathcal{D}$ and $\mathcal{D}'$ respectively.
4. It runs Algorithm 1 with input $\hat{f}, S, S', \ell, R, \beta = 4(2A)^{2\ell}, \epsilon' = (R-1)\epsilon/(128R^2)$ to get a selector $s\colon \mathcal{X} \to \{0,1\}$.
5. If $\mathbf{Pr}_{x\sim\mathcal{U}(X_{\mathrm{test}})}[s(x)=0] \geq R(R-1)^{-1}\theta + \epsilon/4$ it returns REJECT, otherwise it returns $(\mathsf{ACCEPT}, \hat{f})$.

Proof of Soundness. We assume that the algorithm accepts. So, from Hoeffding's inequality, since $|X_{\mathrm{test}}| \gtrsim \log(1/\delta)/\epsilon^2$, with probability at least $1-\delta/3$, it holds

$$\mathop{\mathbf{Pr}}_{x\sim\mathcal{D}'}[s(x)=0] \leq \mathop{\mathbf{Pr}}_{x\sim\mathcal{U}(X_{\mathrm{test}})}[s(x)=0] + \epsilon/4 \leq \frac{R\theta}{R-1} + \epsilon/2\,.$$

Moreover, since $|S_{\mathrm{train}}|$ is sufficiently large, by Theorem E.1, we can ensure that with probability at least $1-\delta/3$, it holds $\mathbf{Pr}_{(x,y)\sim\mathcal{D}^{\mathrm{train}}}[\hat{f}(x)\neq y] \leq \mathsf{opt}_{\mathrm{train}} + \epsilon/(4R)$.

Now, let any $f^\star \in \mathcal{C}$ such that $\mathrm{err}_{\mathrm{train}}(f^\star) = \lambda_{\mathrm{train}}$ and $\mathrm{err}_{\mathrm{test}}(f^\star) = \lambda_{\mathrm{test}}$ and let $p_{\mathrm{up}}, p_{\mathrm{down}}$ be ϵ''-approximate $\mathcal{L}_1$ sandwiching polynomials for $f^\star$ under $\mathcal{D}$ of degree at most ℓ, where $\epsilon'' = \epsilon/(16R)$. Then, p_{up} and $\bar{p}_{\mathrm{down}}$ are nonnegative. Moreover, using the same steps as in the proof of Theorem C.1, we get that $\mathbf{E}_{x\sim\mathcal{D}}[p_{\mathrm{up}}(x)^2] < 4(2A)^{2\ell}$ and $\mathbf{E}_{x\sim\mathcal{D}}[\bar{p}_{\mathrm{down}}(x)^2] < 4(2A)^{2\ell}$. Since $|S|$ and $|S'|$ are sufficiently large, applying Theorem 3.1 (with additive error ϵ', multiplicative slack R and polynomial variance bound $\beta = 4(2A)^{2\ell}$), we get that with probability at least $1-\delta/3$ we have that

$$\begin{aligned}\mathop{\mathbf{E}}_{x\sim\mathcal{D}'}[\hat{f}(x)\bar{p}_{\mathrm{down}}(x)s(x)] &\leq (R+\epsilon')(1+\epsilon')\mathop{\mathbf{E}}_{x\sim\mathcal{D}}[\hat{f}(x)\bar{p}_{\mathrm{down}}(x)] + 2\epsilon' \\ &\leq R\mathop{\mathbf{E}}_{x\sim\mathcal{D}}[\hat{f}(x)\bar{p}_{\mathrm{down}}(x)] + 8R\epsilon'\end{aligned}$$

and

$$\begin{aligned}\mathop{\mathbf{E}}_{x\sim\mathcal{D}'}[\bar{\hat{f}}(x)p_{\mathrm{up}}(x)s(x)] &\leq (R+\epsilon')(1+\epsilon')\mathop{\mathbf{E}}_{x\sim\mathcal{D}}[\bar{\hat{f}}(x)p_{\mathrm{up}}(x)] + 2\epsilon' \\ &\leq R\mathop{\mathbf{E}}_{x\sim\mathcal{D}}[\bar{\hat{f}}(x)p_{\mathrm{up}}(x)] + 8R\epsilon'\,,\end{aligned}$$

where we used the fact that $\mathbf{E}_{x\sim\mathcal{D}}[\hat{f}(x)\bar{p}_{\mathrm{down}}(x)] \leq 2$ and $\mathbf{E}_{x\sim\mathcal{D}}[\bar{\hat{f}}(x)p_{\mathrm{up}}(x)] \leq 2$ in the last inequality of each of the above lines. Thus, for $\epsilon' = (R-1)\epsilon/(128R^2)$, we have that

$$\begin{aligned}\mathop{\mathbf{E}}_{x\sim\mathcal{D}'}[\hat{f}(x)\bar{p}_{\mathrm{down}}(x)s(x)] &\leq R\mathop{\mathbf{E}}_{x\sim\mathcal{D}}[\hat{f}(x)\bar{p}_{\mathrm{down}}(x)] + \epsilon/16 \\ \mathop{\mathbf{E}}_{x\sim\mathcal{D}'}[\bar{\hat{f}}(x)p_{\mathrm{up}}(x)s(x)] &\leq R\mathop{\mathbf{E}}_{x\sim\mathcal{D}}[\bar{\hat{f}}(x)p_{\mathrm{up}}(x)] + \epsilon/16\,.\end{aligned}$$

Hence, using Lemma 4.1 (with additive error $\epsilon/4$), given that $\mathbf{Pr}_{(x,y)\sim\mathcal{D}^{\mathrm{train}}}[\hat{f}(x)\neq y] \leq \mathsf{opt}_{\mathrm{train}} + \epsilon/(4R)$, we have

$$\begin{aligned}\mathop{\mathbf{Pr}}_{(x,y)\sim\mathcal{D}^{\mathrm{test}}}[\hat{f}(x)\neq y \wedge s(x)=1] &\leq \lambda_{\mathrm{test}} + R\left(\lambda_{\mathrm{train}} + \mathop{\mathbf{Pr}}_{(x,y)\sim\mathcal{D}^{\mathrm{train}}}[\hat{f}(x)\neq y]\right) + \epsilon/4 \\ &\leq \lambda_{\mathrm{test}} + R\left(\lambda_{\mathrm{train}} + \mathsf{opt}_{\mathrm{train}}\right) + \epsilon/2\,.\end{aligned}$$

Taking a union bound and applying the law of total probability, we conclude that with probability at least $1-\delta$, we have

$$\begin{aligned}\mathop{\mathbf{Pr}}_{(x,y)\sim\mathcal{D}^{\mathrm{test}}}[\hat{f}(x)\neq y] &\leq \mathop{\mathbf{Pr}}_{x\sim\mathcal{D}'}[s(x)=0] + \mathop{\mathbf{Pr}}_{(x,y)\sim\mathcal{D}^{\mathrm{test}}}[\hat{f}(x)\neq y \wedge s(x)=1] \\ &\leq \lambda_{\mathrm{test}} + R(\lambda_{\mathrm{train}} + \mathsf{opt}_{\mathrm{train}}) + \frac{R\theta}{R-1} + \epsilon\,.\end{aligned}$$

Proof of Completeness. We assume that $d_{\mathrm{TV}}(\mathcal{D},\mathcal{D}') \leq \theta$. Since $|S|$ and $|S'|$ are sufficiently large, by Theorem B.1, with probability at least $1-\delta/2$ we have that $\mathbf{Pr}_{x\sim\mathcal{D}}[s(x)=0] \leq (\theta+\epsilon')/(R-1)$. Also, from Hoeffding's inequality, since $|X_{\mathrm{test}}| \gtrsim \log(1/\delta)/\epsilon^2$, we have that with probability at least

$1-\delta/2$, it holds $\mathbf{Pr}_{x\sim\mathcal{U}(X_{\mathrm{test}})}[s(x)=0] \le \mathbf{Pr}_{x\sim\mathcal{D}'}[s(x)=0]+\epsilon/8$. Hence, with probability at least $1-\delta$, it holds

$$\mathop{\mathbf{Pr}}_{x\sim\mathcal{U}(X_{\mathrm{test}})}[s(x)=0] \le \mathop{\mathbf{Pr}}_{x\sim\mathcal{D}}[s(x)=0]+\theta+\epsilon/8 \le \theta+\frac{\theta+\epsilon'}{R-1}+\epsilon/8 < \frac{R\theta}{R-1}+\epsilon/4\,.$$

Thus, if $d_{\mathrm{TV}}(\mathcal{D},\mathcal{D}') \le \theta$, then the algorithm accepts with probability at least $1-\delta$. □

E Additional Tools

Theorem E.1 (Implicit in [KKMS08]). *Let any $d\in\mathbb{N}^*$, $\ell\in\mathbb{N}$, let $\mathcal{X}\subseteq\mathbb{R}^d$, let $\mathcal{D}$ be a distribution on $\mathcal{X}\times\{0,1\}$ and let $\mathcal{D}_{\mathcal{X}}$ be the marginal of $\mathcal{D}$ on $\mathcal{X}$. Let any $\epsilon,\delta\in(0,1)$. Let $\mathcal{C}\subseteq\{0,1\}^{\mathcal{X}}$ such that for all $f\in\mathcal{C}$ there exists a polynomial $p\in\mathcal{P}_{d,\ell}$ such that $\mathbf{E}_{x\sim\mathcal{D}_{\mathcal{X}}}[|p(x)-f(x)|]\le\epsilon/2$. Then, the degree-$\ell$ $\mathcal{L}_1$ polynomial regression algorithm of [KKMS08] agnostically learns $\mathcal{C}$ up to error $\inf_{f\in\mathcal{C}}\mathbf{Pr}_{(x,y)\sim\mathcal{D}}[f(x)\neq y]+\epsilon$ and failure probability δ with $\mathrm{poly}\left((d+1)^{\ell},1/\epsilon,\log(1/\delta)\right)$ sample complexity and runtime.*

Lemma E.1 (Lemma A.1 from [CKLS25]). *Let any $d\in\mathbb{N}^*,\ell\in\mathbb{N}$ and $A\ge 1$. Let $\mathcal{D}$ be an A-hypercontractive distribution on $\mathbb{R}^d$. Let S be a multiset of m independent examples from $\mathcal{D}$. Then, for any $\delta\in(0,1)$, if $m\gtrsim(d+1)^{2\ell}\left(4A\log_2(4/\delta)\right)^{4\ell+1}$, then with probability at least $1-\delta$, we have that for all $p\in\mathcal{P}_{d,\ell}$, it holds*

$$\frac{1}{2}\mathop{\mathbf{E}}_{x\sim\mathcal{D}}[p(x)^2] \le \mathop{\mathbf{E}}_{x\sim\mathcal{U}(S)}[p(x)^2] \le 2\mathop{\mathbf{E}}_{x\sim\mathcal{D}}[p(x)^2]\,.$$

Lemma E.2 (Lemma A.2 from [KSTV26]). *Let any $d\in\mathbb{N}^*,\ell\in\mathbb{N}$, $A\ge 1$ and $\beta>0$. Let $\mathcal{D}$ be an A-hypercontractive distribution on $\mathbb{R}^d$. Let S be a multiset of m independent examples from $\mathcal{D}$ and let $\mathcal{P}=\{p\in\mathcal{P}_{d,\ell}:\mathbf{E}_{x\sim\mathcal{U}(S)}[p(x)^2]\le 2\beta\}$. Then, for any $\epsilon,\delta\in(0,1)$, if $m\gtrsim(d+1)^{2\ell}\left(4A\log_2(4/\delta)\right)^{4\ell+1}$, then with probability at least $1-\delta$, it holds*

$$\mathop{\mathbf{Pr}}_{x\sim\mathcal{D}}\left[\max_{p\in\mathcal{P}}|p(x)| > 2\sqrt{\frac{\beta(d+1)^{\ell}}{\epsilon}}\right] \le \epsilon\,.$$

Lemma E.3. *Let any $d\in\mathbb{N}^*,\ell\in\mathbb{N}$, $A\ge 1$, $\epsilon,\beta>0$ and $R\ge 1$. Let $\mathcal{D}$ be an A-hypercontractive distribution on $\mathbb{R}^d$. Let any $p\in\mathcal{P}_{d,\ell}$ and $f\colon\mathbb{R}^d\to\{0,1\}$ such that $\mathbf{E}_{x\sim\mathcal{D}}[f(x)|p(x)|]\le\epsilon/(R+\epsilon)$ and $\mathbf{E}_{x\sim\mathcal{D}}[p(x)^2]\le\beta$. Let S be a multiset of m independent examples from $\mathcal{D}$. Then, for any $\delta\in(0,1)$, if $m\gtrsim R^4\beta(d+1)^{2\ell}\left(4A\log_2(8/\delta)\right)^{4\ell+1}/\epsilon^4$, then with probability at least $1-\delta$, we have that $\mathbf{E}_{x\sim\mathcal{U}(S)}[f(x)|p(x)|]\le 2\epsilon/(2R+\epsilon)$ and $\mathbf{E}_{x\sim\mathcal{U}(S)}[p(x)^2]\le 2\beta$.*

Proof. If m is sufficiently large, by Lemma E.1, we have that with probability at least $1-\delta/2$, it holds $\mathbf{E}_{x\sim\mathcal{U}(S)}[p(x)^2]\le 2\beta$. It remains to show that $\mathbf{E}_{x\sim\mathcal{U}(S)}[f(x)|p(x)|]\le 2\epsilon/(2R+\epsilon)$ with probability at least $1-\delta/2$. For any $z\in\mathbb{R}^d$, let $\psi(z)=f(z)|p(z)|-\mathbf{E}_{x\sim\mathcal{D}}[f(x)|p(x)|]$. Then, for any $i\ge 1$ and for any $\varepsilon>0$, it holds[12]

$$\begin{aligned}
\mathop{\mathbf{Pr}}_{S\sim\mathcal{D}^m}\left[\left|\mathop{\mathbf{E}}_{x\sim\mathcal{U}(S)}[\psi(x)]\right|>\varepsilon\right] &\le \frac{1}{\varepsilon^{2i}}\mathop{\mathbf{E}}_{S\sim\mathcal{D}^m}\left[\left|\mathop{\mathbf{E}}_{x\sim\mathcal{U}(S)}[\psi(x)]\right|^{2i}\right]\\
&\le \left(\frac{4i}{\varepsilon^2 m}\right)^i \mathop{\mathbf{E}}_{x\sim\mathcal{D}}[|\psi(x)|^{2i}]\\
&\le 2\left(\frac{16i}{\varepsilon^2 m}\right)^i \mathop{\mathbf{E}}_{x\sim\mathcal{D}}[|p(x)|^{2i}]\\
&\le 2(2Ai)^{2\ell i}\left(\frac{16i}{\varepsilon^2 m}\right)^i \mathop{\mathbf{E}}_{x\sim\mathcal{D}}[|p(x)|]^{2i}\\
&\le 2\left(\frac{16(2A)^{2\ell}i^{2\ell+1}\beta}{\varepsilon^2 m}\right)^i.
\end{aligned}$$

[12] The notation $S\sim\mathcal{D}^m$ means that S is a multiset of m independent samples from $\mathcal{D}$.

In the first line, we used Markov's inequality. In the second line, we used the Marcinkiewicz–Zygmund inequality and Hölder's inequality (see [Fer14]). In the third line, we used the fact that $(a+b)^{2k} \leq 4^k \max\{a^{2k}, b^{2k}\}$ for any $a, b, k \geq 0$. Let $\varepsilon = 2\epsilon/(2R+\epsilon) - \epsilon/(R+\epsilon) > \epsilon^2/6R^2$. If $m \gtrsim R^4\beta(d+1)^{2\ell}\left(4A\log_2(8/\delta)\right)^{4\ell+1}/\epsilon^4$, then, for $i = \lceil \log_2(4/\delta) \rceil$, we get that the term in the last line is upper bounded by $\delta/2$, which concludes the proof. □

Lemma E.4. *Let any $d, \ell \in \mathbb{N}^*$. Let $\mathcal{F} \subset \{\mathbb{R}^d \to \{0,1\}\}$ be a finite class and let $\mathcal{H} = \{x \mapsto \mathbf{1}\{f(x)|p(x)| > \tau\} : p \in \mathcal{P}_{d,\ell}, \tau \geq 0, f \in \mathcal{F}\}$. Then,* $\mathrm{VCdim}(\mathcal{H}) \lesssim (d+1)^\ell \ell \log(d) + \log(|\mathcal{F}|)$.

Proof. For any $f \in \mathcal{F}$, let $\mathcal{H}_f = \{x \mapsto \mathbf{1}\{f(x)|p(x)| > \tau\} : p \in \mathcal{P}_{d,\ell}, \tau \geq 0\}$. We will first show that $\mathrm{VCdim}(\mathcal{H}_f) \lesssim (d+1)^\ell$ for any $f \in \mathcal{F}$. Let $\mathcal{G} = \{x \mapsto \mathbf{1}\{|p(x)| > \tau\} : p \in \mathcal{P}_{d,\ell}, \tau \geq 0\}$. Notice that for any $x \in \mathbb{R}^d$, $p \in \mathcal{P}_{d,\ell}$ and $\tau \geq 0$, it holds $\mathbf{1}\{|p(x)| > \tau\} = \mathbf{1}\{p(x) > \tau\} \vee \mathbf{1}\{p(x) < -\tau\}$, namely, the function $x \mapsto \mathbf{1}\{|p(x)| > \tau\}$ is the logical OR of two PTFs of degree at most ℓ. Thus, by Fact A.2, it holds $\mathrm{VCdim}(\mathcal{G}) \lesssim (d+1)^\ell$. Finally, notice that $\mathcal{H}_f = \{fg : g \in \mathcal{G}\}$, which implies that $\mathrm{VCdim}(\mathcal{H}_f) \leq \mathrm{VCdim}(\mathcal{G}) \lesssim (d+1)^\ell$. Since $\mathcal{H} = \bigcup_{f\in\mathcal{F}} \mathcal{H}_f$, by Fact A.1, we deduce that $\mathrm{VCdim}(\mathcal{H}) \lesssim (d+1)^\ell \ell \log(d) + \log(|\mathcal{F}|)$. □

Lemma E.5. *Let any $M > 0$. Let any set $\mathcal{X}$, any distribution $\mathcal{D}$ on $\mathcal{X}$, any $f\colon \mathcal{X} \to \mathbb{R}$ and any $\mathcal{G} \subseteq \{0,1\}^{\mathcal{X}}$ such that $\sup_{g\in\mathcal{G}} \sup_{x\in\mathcal{X}} g(x)|f(x)| \leq M$ and $\mathrm{VCdim}(\mathcal{G}) < \infty$. Let S be a multiset of n i.i.d. samples drawn from $\mathcal{D}$. Then, there exists some universal constant $C > 0$ such that for any $\delta \in (0,1)$, with probability at least $1-\delta$ over S, it holds*

$$\sup_{g\in\mathcal{G}} \left| \mathop{\mathbf{E}}_{x\sim\mathcal{U}(S)}[f(x)g(x)] - \mathop{\mathbf{E}}_{x\sim\mathcal{D}}[f(x)g(x)] \right| \leq CM \left(\frac{\mathrm{VCdim}(\mathcal{G})\log(n) + \log(1/\delta)}{n} \right)^{1/4}.$$

Proof. Let any $\Delta > 0$ such that $k := M/\Delta$ is an integer. For any $i \in \mathbb{Z}$ and $x \in \mathcal{X}$, we define $I_i(x) := \mathbf{1}\{i\Delta \leq f(x) < (i+1)\Delta\}$. Then, for any $g \in \mathcal{G}$, we have

$$\begin{aligned}
\left| \mathop{\mathbf{E}}_{x\sim\mathcal{U}(S)}[f(x)g(x)] - \mathop{\mathbf{E}}_{x\sim\mathcal{D}}[f(x)g(x)] \right| &= \left| \sum_{i=-k}^{k} \mathop{\mathbf{E}}_{x\sim\mathcal{U}(S)}[f(x)g(x)I_i(x)] - \mathop{\mathbf{E}}_{x\sim\mathcal{D}}[f(x)g(x)I_i(x)] \right| \\
&\leq \sum_{i=-k}^{k} \left| \mathop{\mathbf{E}}_{x\sim\mathcal{U}(S)}[f(x)g(x)I_i(x)] - \mathop{\mathbf{E}}_{x\sim\mathcal{D}}[f(x)g(x)I_i(x)] \right| \\
&\leq \sum_{i=-k}^{k} \left| i\Delta \left(\mathop{\mathbf{E}}_{x\sim\mathcal{U}(S)}[g(x)I_i(x)] - \mathop{\mathbf{E}}_{x\sim\mathcal{D}}[g(x)I_i(x)] \right) \right| \\
&\quad + \Delta \mathop{\mathbf{E}}_{x\sim\mathcal{U}(S)}[g(x)I_i(x)] + \Delta \mathop{\mathbf{E}}_{x\sim\mathcal{D}}[g(x)I_i(x)] \\
&\leq 2\Delta + \sum_{i=-k}^{k} \left| i\Delta \left(\mathop{\mathbf{E}}_{x\sim\mathcal{U}(S)}[g(x)I_i(x)] - \mathop{\mathbf{E}}_{x\sim\mathcal{D}}[g(x)I_i(x)] \right) \right| \\
&\leq 2\Delta + M \sum_{i=-k}^{k} \left| \mathop{\mathbf{E}}_{x\sim\mathcal{U}(S)}[g(x)I_i(x)] - \mathop{\mathbf{E}}_{x\sim\mathcal{D}}[g(x)I_i(x)] \right|.
\end{aligned}$$

Since the VC dimension of the class $\{x \mapsto \mathbf{1}\{a \leq f(x) < b\} : a, b \in \mathbb{R}\}$ is at most 2, by Fact A.2, we have that for any $g \in \mathcal{G}$, integer i and $\Delta > 0$, the function $g(x)I_i(x)$ belongs to a class of VC dimension at most $O\left(\mathrm{VCdim}(\mathcal{G})\right)$. Thus, by Fact A.4, we have that with probability at least $1-\delta$, it holds

$$\sup_{g\in\mathcal{G}} \left| \mathop{\mathbf{E}}_{x\sim\mathcal{U}(S)}[f(x)g(x)] - \mathop{\mathbf{E}}_{x\sim\mathcal{D}}[f(x)g(x)] \right| \leq O\left(\Delta + \frac{M^2}{\Delta} \sqrt{\frac{\mathrm{VCdim}(\mathcal{G})\log(n) + \log(1/\delta)}{n}} \right).$$

Taking Δ to minimize the right hand side, concludes the proof. □

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We justify all of our claims through appropriate proofs both in the main content and in the appendix.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We included a dedicated discussion on the limitations of our work in the last paragraph of the main content.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Our assumptions are made clear in Definitions 1.2 and 2.1 and we provide complete proofs for all our results in the main content and in the appendix.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [N/A]

   Justification: This paper does not include experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [N/A]

Justification: This paper does not include experiments.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://neurips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

Answer: [N/A]

Justification: This paper does not include experiments.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [N/A]

Justification: This paper does not include experiments.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [N/A]

   Justification: This paper does not include experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: Our work fully complies with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: Our work is of theoretical nature and we do not identify any direct potential negative societal impact.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: Our paper does not pose any such risks.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [N/A]

    Justification: We do not use any existing assets.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: We do not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: We used LLMs only for minor editing suggestions.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.