Approximate Replicability in Learning

Max Hopkins[*], Russell Impagliazzo[†], Christopher Ye[‡]

October 24, 2025

Abstract

Replicability, introduced by (Impagliazzo et al. STOC '22), is the notion that algorithms should remain stable under a resampling of their inputs (given access to shared randomness). While a strong and interesting notion of stability, the cost of replicability can be prohibitive: there is no replicable algorithm, for instance, for tasks as simple as threshold learning (Bun et al. STOC '23). Given such strong impossibility results we ask: under what approximate notions of replicability is learning possible?

In this work, we propose three natural relaxations of replicability in the context of PAC learning:

1. **Pointwise:** the learner must be consistent on any *fixed* input, but not across all inputs simultaneously.
2. **Approximate:** the learner must output hypotheses that classify most of the distribution consistently.
3. **Semi:** the algorithm is fully replicable, but may additionally use shared unlabeled samples.

In all three cases, for constant replicability parameters, we obtain sample-optimal agnostic PAC learners: 1) and 2) are achievable for "free" using $\Theta(d/\alpha^2)$ samples, while 3) requires $\Theta(d^2/\alpha^2)$ labeled samples.

[*]Institute for Advanced Study, Princeton. nmhopkin@ias.edu. Supported by NSF Award DMS-2424441

[†]University of California, San Diego. rimpagliazzo@ucsd.edu. Supported by NSF Award AF: Medium 2212136

[‡]University of California, San Diego. czye@ucsd.edu. Supported by NSF Award AF: Medium 2212136 and HDR TRIPODS Phase II grant 2217058 (EnCORE Institute).

Contents

1 Introduction

Replicability is the notion that statistical methods should remain stable under fresh samples from the same population. Recently, [ILPS22] introduced a formal framework of replicability in learning, stating that for (most) random seeds, running a "replicable" algorithm over fresh samples should produce the same result:

$$\forall \mathcal{D} : \Pr_{S,S' \sim \mathcal{D}^m, r \in R}[A(S;r) = A(S';r)] \geqslant 1 - \rho.$$

Unfortunately, [ILPS22]'s replicability, while powerful, is highly restrictive. [BGH+23], for instance, showed the notion is quantitatively equivalent to approximate differential privacy, ruling out a broad family of tasks as basic as learning 1D-Thresholds. Moreover, even for tasks where replicability is possible (such as mean estimation), achieving constant replicability often comes with quadratic overhead [BGH+23, HIK+24], rendering the notion infeasible in high dimensional settings.

In this work, we introduce three weakened notions of replicability, *pointwise* replicability, *approximate* replicability, and *semi*-replicability, that remain powerful enough to be of potential use in application, yet apply to a substantially broader domain of problems. In particular, we show in these settings that any standard (non-replicable) learning algorithm can be 'boosted' to one satisfying weak replicability, along with corresponding lower bounds on the overhead required to do so.

Pointwise Replicability Our first notion of study is a basic relaxed variant of replicability we call *pointwise* replicability, which asks that the behavior of our algorithm is replicable on any *fixed* point in the domain (rather than over all such points simultaneously):

Definition 1.1 (Pointwise Replicability)**.** *We say an algorithm A is ρ-pointwise replicable if:*

$$\forall \mathcal{D}, \forall x \in X : \Pr_{S,S' \sim \mathcal{D}^m, r \sim R}[A(S;r)(x) \neq A(S';r)(x)] < \rho$$

This definition is inspired by the notion of private prediction of Dwork and Feldman [DF18], where privacy is only required over revealing the label of a single queried point in the domain (rather than revealing the full model). Similarly, one can think of our notion as *replicable prediction*: given a test point x, such an algorithm will almost always give the same predicted label. We give a simple transform which turns any non-replicable algorithm into a pointwise replicable predictor at the cost of running the original algorithm $\mathrm{poly}(\rho^{-1})$ times.

Approximate Replicability Our second notion, approximate replicability, asks that the outputs remain the same on "most" (rather than all) points in the output domain simultaneously.

Definition 1.2 (Approximate Replicability)**.** *We say an algorithm A is (ρ, γ)-approximately replicable if:*

$$\forall \mathcal{D} : \Pr_{S,S' \sim \mathcal{D}^m, r \sim R}\left[\Pr_{x \sim \mathcal{D}_{\mathcal{X}}}[A(S;r)(x) \neq A(S';r)(x)] > \gamma\right] < \rho$$

where $\mathcal{D}_{\mathcal{X}}$ is the marginal distribution over unlabeled data.

This definition is closest to other approximate notions proposed in the literature, and, in fact, was explicitly raised in [CMY23] as a potential direction of study. This notion can easily be generalized beyond PAC learning settings by equipping the output space with an appropriate metric (e.g. the same metric used to evaluate correctness). For example, if the task is ℓ_p mean estimation, an algorithm is (ρ, γ)-approximately replicable if it outputs two estimates within γ in ℓ_p-norm. In this particular instance (and others with similarly unique solutions), it is clear one can achieve (ρ, α)-approximate replicability essentially for "free" since any

two estimates within α of the true mean are also within 2α of each other, yielding a large class of problems where the natural algorithm is immediately approximately replicable.

Approximate replicability becomes more interesting for complex tasks like PAC learning where two optimal hypotheses can disagree on much of the domain.[1] Nevertheless, building on our pointwise replicable algorithm, we show any learner can be transformed into an approximately replicable one without too much overhead.

Semi-Replicable Learning Our final relaxation of replicability, the semi-replicable model, diverges from the above in still requiring identical outputs, but gives the algorithm access to an additional pool of *shared, unlabeled* samples (in addition to shared randomness).

Definition 1.3 (Semi-Replicability)**.** *We say an algorithm is ρ-semi replicable if:*

$$\forall \mathcal{D} : \Pr_{S_U \sim \mathcal{D}_{\mathcal{X}}^m, S, S' \sim \mathcal{D}^m, r \in R}[A(S; S_U, r) = A(S'; S_U, r)] \geqslant 1 - \rho$$

where $\mathcal{D}_{\mathcal{X}}$ is the marginal distribution over unlabeled data.

The semi-replicable model is inspired by the well-studied notion of *semi-privacy* [BNS16a] and has several interesting motivations. First, the notion is reasonably practical; public unlabeled data is plentiful and can easily be shared between different research groups. For instance, fine-tuning machine learning models to a specific application (where we require algorithmic stability) can begin from a large, shared public model.

The model can also be thought of as a natural interpolation between *distribution-dependent* PAC learning where the unlabeled data distribution is known (and every learnable class is easily shown to be replicably learnable by seminal results of [BI91] and [ILPS22, BGH+23]) and the standard *distribution-free* PAC model where many learnable classes are not replicably learnable at all. We observe one does not need to know the entire underlying marginal — just a few public unlabeled samples (shared data about the marginal distribution) suffices to allow replicable learning for every learnable class (albeit, in this case, with a necessary quadratic blowup).

1.1 Our Contributions

We now give a more detailed exposition of our main contributions. We start with some brief background, and refer the reader to Section 2 for further details. See Table 1 for a quick informal summary of our results.

We primarily study (binary) classification in the context of PAC learning [Val84]. Given sample access to a labeled distribution $\mathcal{D}$ over $(x, y) \in \mathcal{X} \times \{\pm 1\}$, we are asked to produce a hypothesis $h : \mathcal{X} \to \{\pm 1\}$. The classification error of h, denoted $\mathrm{err}_{\mathcal{D}}(h)$, is the probability a point is mislabeled, i.e. $\Pr_{\mathcal{D}}(h(x) \neq y)$. An algorithm is a (α, β)-learner for a hypothesis class $\mathcal{H}$ if for every distribution $\mathcal{D}$, $\Pr(\mathrm{err}_{\mathcal{D}}(A(S)) > \mathrm{OPT} + \alpha) < \beta$ where $\mathrm{OPT} := \inf_{h \in \mathcal{H}} \mathrm{err}_{\mathcal{D}}(h)$ is the optimal error of the class. In this standard setting, learnability of a class is tightly parameterized by its VC Dimension. In particular, a seminal result [VC74, BEHW89] shows that any class $\mathcal{H}$ with VC Dimension d can be (α, β)-learned with $O\left(\frac{d+\log(1/\beta)}{\alpha^2}\right)$ samples.

Pointwise Replicability Our first main result is a simple transformation turning any standard learner into a pointwise replicable one:

Theorem 1.4 (Informal Theorem 3.1)**.** *Let A be an (agnostic) (α, β)-learner on $m(\alpha, \beta)$ samples. There exists an (agnostic) ρ-pointwise replicable (α, β)-learner with sample complexity $\tilde{O}\left(\frac{m(\alpha\rho, \alpha\rho)\log(1/\beta)}{\rho^2}\right) =$*

[1] In fact, in this case even in the realizable PAC setting where all optimal hypotheses are close, approximate replicability is still not immediate from accuracy since we'd like to ensure replicability over *all* input distributions (including non-realizable ones).

	Upper Bound	Lower Bound	Notes
Pointwise	$\frac{d}{\rho^4\alpha^2}$ [1.4]	$\frac{d}{\rho^2\alpha^2}$ [1.5]	Agnostic
Approximate	$\frac{d}{\rho^2\gamma^2\alpha^2} + \frac{d}{\rho^2\alpha^4}$ [1.6] $\frac{d}{\rho^2\gamma^4\alpha^2}$ [1.7]	$\frac{d}{(\rho+\gamma)^2\alpha^2}$ [1.9]	Agnostic
	$\frac{d}{\rho^2\min(\alpha,\gamma)}$ [1.7]		Proper, Realizable
	$\frac{1}{\rho^2\alpha^2} + \frac{1}{\gamma^2}$ [1.8]	$\frac{1}{(\rho+\gamma)^2\alpha^2}$ [1.9]	Proper, Agnostic Threshold Learner
Semi-Replicable	$\frac{d^2}{\rho^2\alpha^2}$ labeled samples $\frac{d}{\alpha}$ shared samples [1.10]	$\frac{d^2}{\rho^2\alpha^2}$ labeled samples [1.11] $\frac{d}{\alpha}$ shared samples [1.12]	Proper*, Agnostic

Table 1: Table of our results. Polylogarithmic factors and dependence on β omitted for simplicity. All algorithms incur an additional $\mathrm{polylog}(1/\beta)$-factor in sample complexity. *Our algorithm for semi-replicable learning invokes an improper learning algorithm to optimize sample complexity. Invoking a proper version instead results in a proper learner losing a $\log(1/\alpha)$-factor in sample complexity.

$\tilde{O}\left(\frac{d\log(1/\beta)}{\rho^4\alpha^2}\right)$. *Furthermore, our algorithm runs in time linear in sample complexity with* $O\left(\frac{\log(1/\beta)}{\rho^2}\right)$ *oracle calls to* A. *In the realizable setting, our algorithm has sample complexity* $\tilde{O}\left(\frac{d\log(1/\beta)}{\rho^3\alpha}\right)$.

We remark that if the error parameter β is only required to be a small constant (i.e. an $(\alpha, 0.001)$-learner), we in fact obtain an agnostic learner algorithm with sample complexity $\tilde{O}\left(\frac{d}{\rho^2\alpha^2}\right)$. The following lower bound shows this is essentially optimal.

Theorem 1.5. *Any* ρ*-pointwise replicable* $(0.01\alpha, 0.0001)$*-learner requires* $\Omega\left(\frac{d}{\rho^2\alpha^2\sqrt{\log(1/\rho)}}\right)$ *samples. Furthermore, any pointwise-replicable learner requires shared randomness. Any* ρ*-pointwise replicable realizable* $(0.01\alpha, 0.0001)$*-learner requires* $\Omega\left(\frac{d}{\rho^2\alpha\sqrt{\log(1/\rho)}}\right)$ *samples.*

Approximate Replicability Building on our pointwise replicable procedure, we give a second simple transform turning any standard learner into an approximately replicable one:

Theorem 1.6. *Let* A *be an (agnostic)* (α, β)*-learner on* $m(\alpha, \beta)$ *samples. There exists an (agnostic)* (ρ, γ)*-approximately replicable* (α, β)*-learner with sample complexity* $O\left(m(\alpha,\alpha)\left(\frac{\log^3(1/\beta)}{\rho^2\gamma^2} + \frac{\log^4(1/\beta)}{\rho^2\alpha^2}\right)\right) = \tilde{O}\left(\frac{d\log^3(1/\beta)}{\rho^2\gamma^2\alpha^2} + \frac{d\log^4(1/\beta)}{\rho^2\alpha^4}\right)$. *Furthermore, our algorithm runs in time linear in sample complexity with* $O\left(\frac{\log^3(1/\beta)}{\rho^2\gamma^2} + \frac{\log^4(1/\beta)}{\rho^2\alpha^2}\right)$ *oracle calls to* A.

The above result has unfortunately poor dependence on α when γ, ρ are constant, requiring $\frac{d}{\alpha^4}$ samples even for $(0.1, 0.1)$-approximate replicability. Toward this end, we give an alternate algorithm achieving

optimal α-dependence at the cost of an additional polynomial factor in γ, in particular showing that under constant replicability parameters, approximate replicable learning is not much more expensive than non-replicable learning.

Theorem 1.7. *Let A be an (agnostic) (α, β)-learner on $m(\alpha, \beta)$ samples. There exists an (agnostic) (ρ, γ)-approximately replicable (α, β)-learner with sample complexity $\tilde{O}\left(\left(\frac{d\log(1/\beta)}{\gamma^4\alpha^2}\right)\left(\frac{1}{\rho^2} + \log(1/\beta)\right)\right)$. In the realizable case, there is a proper (ρ, γ)-approximately replicable (α, β)-learner with sample complexity $O\left(\frac{d+\log(1/\min(\rho,\beta))}{\rho^2\min(\alpha,\gamma)}\right)$.*

Unfortunately, in terms of their γ, ρ, and α-dependencies, both of our generic approximately replicable agnostic learners are likely sub-optimal. They are also improper, relying on a simple aggregation of multiple hypotheses in the class. Toward resolving these issues, we give a simple direct algorithm for the fundamental class of 1-dimensional thresholds on $\mathbb{R}$, that is hypotheses of the form $h(x) \mapsto 1[x > a]$. Thresholds pose a fundamental obstacle to strict replicability: there is no replicable algorithm for learning thresholds [BGH+23, ALMM19]. Nevertheless, we give a proper, approximately replicable learner for the class with optimal sample complexity when $\rho = \Theta(\gamma)$.

Proposition 1.8. *Let $\rho, \gamma, \alpha > 0$ and $0 < \beta < \rho$. There is a proper (ρ, γ)-approximately replicable (α, β)-learner for thresholds with sample complexity $O\left(\frac{\log(1/\beta)}{\gamma^2} + \frac{\log^3(1/\alpha\beta)}{\rho^2\alpha^2}\right)$.*

Finally, we give our main lower bound showing the above is essentially optimal when $\rho = \Theta(\gamma)$.

Theorem 1.9. *Let $\rho < 0.001$. Any (ρ, γ)-approximately replicable $(\alpha, 0.001)$-learner requires sample complexity $\Omega\left(\frac{d}{\alpha^2(\rho+\gamma)^2\sqrt{\log(1/\rho)}}\right)$.*

Semi-Replicable Learning In our final model, semi-replicability, we give upper and lower bounds that are optimal (up to polylogarithmic factors) in both the shared and labeled sample complexities.

Theorem 1.10. *There is a ρ-semi-replicable (α, β)-learner for with shared (unlabeled) sample complexity $O\left(\frac{d}{\alpha}\log(1/\beta)\right)$ and labeled sample complexity $O\left(\frac{d^2}{\alpha^2\rho^2}\log^5\frac{1}{\rho\beta}\right)$.*

We give a lower bound for the number of labeled samples (shared or otherwise) showing Theorem 1.10 is tight up to log factors regardless of the number of unlabeled samples.

Theorem 1.11. *Any ρ-semi-replicable $(\alpha, 0.0001)$-learner for any class $\mathcal{H}$ with VC dimension d requires sample complexity $\Omega\left(\frac{d^2}{\rho^2\alpha^2\log^6 d}\right)$. This bound holds regardless of unlabeled sample complexity.*

Finally, we give a lower bound for the shared sample complexity that holds regardless of the overall sample complexity.

Theorem 1.12. *There exists a class $\mathcal{H}$ such that any 0.0001-semi-replicable $(\alpha, 0.0001)$-learner for $\mathcal{H}$ requires $\Omega\left(\frac{d}{\alpha}\right)$ shared samples. The lower bound holds even if the shared samples are labeled.*

On our way to proving this result, we also show that a similar lower bound holds for semi-private learners. In particular, we exhibit a class where even semi-private learners requires $\Omega\left(\frac{d}{\alpha}\right)$ public samples [ABM19]. It remains an interesting open question to determine whether *every* class with infinite Littlestone Dimension and VC dimension d requires $\Omega\left(\frac{d}{\alpha}\right)$ shared (resp. public) samples or if there are some classes that can be learned with fewer shared (resp. public) samples.

Shared Randomness It is well known that replicability requires shared randomness: for tasks as simple as bias estimation it is not possible to achieve high probability replication without shared random bits [ILPS22], and this impossibility result is now known to extend to a broad class of problems [CMY23]. While our algorithms critically use shared randomness, a priori it is not clear this is necessary. Our final result is to show that all three models indeed require at least one shared random bit.

For pointwise and semi-replicability, the necessity of shared randomness is fairly straightforward, and follows already from the reductions we give to prove sample lower bounds, e.g. by encoding a (strictly replicable) bias estimation instance onto the label of a single element. For approximate replicability, we show directly using the Poincare-Miranda theorem that even an algorithm that is consistent on a $\frac{1}{2}$-fraction of the domain requires shared randomness.

Theorem 1.13. *There is no deterministic* $(0.01, 0.5)$*-approximately replicable* $(0.05, 0.01)$*-learner for any class with VC dimension* $d \geqslant 2$*.*

1.2 Technical Overview

We give an overview of our techniques.

1.2.1 Replicable Prediction

We begin by describing a simple procedure to turn any learning algorithm A into a replicable predictor. For any x, let $p_x := \Pr_S(A(S)(x) = 1)$ denote the probability that a hypothesis learned by A classifies x as a positive example. Our algorithm runs A on T fresh independent samples to obtain hypotheses $h_1, \ldots, h_T$. Then, given any x, let $\hat{p}_x := \frac{1}{T}\sum_{t=1}^{T} \frac{1+h_t(x)}{2}$ denote an empirical estimate of p_x. Our algorithm then draws a random threshold r and outputs the hypothesis

$$g : x \mapsto 2 \cdot \mathbb{1}[\hat{p}_x > r] - 1.$$

Since α is drawn uniformly from $[0, 1]$ and $\hat{p}_x$ is an unbiased estimator of p_x, we immediately get that the expected error of our algorithm g over the choice of random threshold r and input samples S matches the expected error of A. Markov's inequality then ensures g is α-accurate 99% probability.

We now argue this procedure is pointwise replicable. Fix an x. If we run A on $T = O\left(\frac{1}{\rho^2}\right)$ fresh datasets, we can show that $\mathrm{Var}(\hat{p}_x) \leqslant \rho^2$. Since g fails to be replicable when r falls between two estimates $\hat{p}_x$, the probability that this occurs (over the randomness of $\hat{p}_x, r$) is (via Jensen's inequality) at most ρ. Thus, if A is an $(\alpha, 0.01)$-accurate learner like ERM on e.g. $\frac{d}{\alpha^2}$ samples, we obtain 99% confidence ρ-pointwise replicable algorithm on $\frac{d}{\rho^2\alpha^2}$ samples which our corresponding lower bound shows is tight.

Finally, we'd like to amplify the above pointwise replicable algorithm to arbitrary β confidence. In typical statistical learning settings this is easy: one simply repeats the algorithm $\log(1/\beta)$ times and selects the best option. This works in the replicable setting since one can ensure the output list of hypotheses is the same across both rounds. This is of course not true in the pointwise setting, but we will see that a variant of the process still works up to a further quadratic loss in ρ.

Toward this end, let's start with the standard amplification algorithm. Run the constant success algorithm above $O(\log(1/\beta))$ times to obtain hypothesis $g_1, \ldots, g_d$. Since each hypothesis is α-accurate with high constant probability, at least one of the resulting hypotheses will be α-accurate with probability at least $1 - \beta$. Ideally, we'd now like to test the error of each hypothesis and *replicably* select an α-optimal hypothesis from $g_1, \ldots, g_d$. If over two runs, we are guaranteed $g_i^{(1)}, g_i^{(2)}$ are sufficiently close in classification distance, then we can use replicable hypothesis selection algorithms to select an index [GKM21, HLYY25].

Unfortunately, even though g_i are likely to be consistent on any *fixed* x (as guaranteed by pointwise replicability), since g_i classify x based on $\hat{p}_x$ that are arbitrarily correlated (as averages over hypotheses), we

cannot apply a strong enough concentration bound to argue that g_i are consistent over most x (let alone all x) with high probability. To circumvent this issue, we boost the accuracy of A to $\rho\alpha$, so that the first hypothesis is α-accurate with probability $1-\rho$. In particular, with $1-\rho$ probability, multiple runs of the algorithm output the first hypothesis, which we argued was ρ-pointwise replicable. Since we boost the accuracy of A to $\rho\alpha$, this yields a final sample complexity of $\frac{d}{\rho^4\alpha^2}$.

Lower Bound To obtain a lower bound, we reduce from replicable bias estimation: given sample access to a Rademacher distribution $\mathcal{D}^*$ (supported on $\{\pm 1\}$), determine if the mean is greater than α or less than $-\alpha$. [ILPS22] show any ρ-replicable algorithm for α-bias estimation (even with constant error probability) requires $\Omega\left(\frac{1}{\rho^2\alpha^2}\right)$ samples. We will build a reduction that 'hides' one such instance into a set of d shattered points, improving the lower bound to the tight $\frac{d}{\rho^2\alpha^2}$.

Now, fix any class $\mathcal{H}$ with VC Dimension d and let $\{x_1,\dots,x_d\}$ be a set of shattered elements. Without loss of generality, let d be odd. We reduce bias estimation to PAC learning as follows. Let $\mathcal{D}$ be a uniform distribution over $\{x_1,\dots,x_d\}$ that labels x_i according to $\mathrm{Rad}(\alpha)$ for a randomly selected subset of $\frac{d-1}{2}$ elements, $\mathrm{Rad}(-\alpha)$ for the remaining $\frac{d-1}{2}$, and $\mathcal{D}^*$ for the remaining element x^*. Given sample access to $\mathcal{D}^*$, we can easily create a dataset of m samples from $\mathcal{D}$, by drawing m elements from $\{x_1,\dots,x_d\}$ independently and uniformly, and labeling each one according to the element's corresponding Rademacher distribution. Note that we only need (in expectation) $\frac{m}{d}$ samples from $\mathcal{D}^*$ to simulate m samples from $\mathcal{D}$. Let A be a ρ-pointwise replicable (α,β)-learner for $\mathcal{H}$. We construct S as described above, and then given $h \leftarrow A(S)$, we return $h(x^*)$.

Since for any fixed x, and in particular x^*, our algorithm is consistent with probability ρ, we immediately obtain a ρ-replicable algorithm. Thus, it suffices to argue correctness. Assume without loss of generality $\mathcal{D}^* \sim \mathrm{Rad}(\alpha)$. Suppose we obtain an 0.01α-accurate hypothesis h on this distribution. There are $\frac{d+1}{2}$ (randomly chosen) elements in the domain whose labels are distributed according to $\mathrm{Rad}(\alpha)$. For every element labeled -1, h incurs an excess error of $\frac{\alpha}{d}$ over the optimal hypothesis. In particular, if h is 0.01α-accurate, h can label at most $\frac{0.01\alpha}{\alpha/d} = 0.01d$ elements -1. Since the elements appear identical to A and the position of x^* is randomly selected, we argue this implies the probability that $h(x^*) = -1$ is at most $\frac{0.01d}{(d+1)/2} \leqslant 0.02$. Thus, we obtain an algorithm that solves bias estimation with error probability at most 0.02 and takes (with high probability) at most $O(m/d)$ samples from $\mathcal{D}^*$, violating [ILPS22] if $m \leqslant o(\frac{d}{\rho^2\alpha^2})$.

Finally, note that this reduction also immediately implies that pointwise replicability requires shared randomness, since replicable bias estimation is also well known to require shared randomness [ILPS22, CMY23, DPWV23]. In particular, we can take a single element $x \in \mathcal{X}$ and use samples from a bias estimation instance to label x.

1.2.2 Approximate Replicability

Naively, approximate replicability seems to be a somewhat weaker guarantee than pointwise replicability: while the latter requires replicability on *every* fixed x, approximate replicability only asks for an average-case guarantee over the domain. Indeed, this intuition holds true to an extent — a simple Markov argument shows that any γ-pointwise replicable algorithm is automatically $(0.01,\gamma)$-approximately replicable. This allows us to immediately translate a γ-pointwise replicable algorithm (with constant error β) above into a $(0.01,\gamma)$-approximately replicable $(\alpha,0.01)$-learner using $O\left(\frac{d}{\gamma^2\alpha^2}\right)$ samples. Of course, our goal is to obtain a (ρ,γ)-approximately replicable (α,β)-learner. Naively using the above argument, we can achieve this with a $\rho\gamma$-pointwise replicable (α,β)-learner that takes $O\left(\frac{d}{\rho^4\gamma^4\alpha^2}\right)$ samples. We describe how this can be improved below.

Better Approximately Replicable Learning I: Boosting Replicability Our first improvement comes from a more efficient method to reduce the replicability error ρ. Using the above argument, we begin with a $(0.01, \gamma)$-approximately replicable (α, β)-learner A with $O\left(\frac{d+\log(1/\beta)}{\gamma^4\alpha^2}\right)$ samples. By definition, our algorithm, over the randomness of r and the samples S, S', produces γ-close hypotheses with 99% probability. This means that for most random strings r, the hypotheses produced by $A(\cdot; r)$ lie mostly in a single "cluster" of radius $O(\gamma)$. Call such random strings "good". Our goal will be to replicably identify a good r, then collect a few additional samples to find one of the hypotheses in its corresponding cluster.

Toward this end, observe that taking $O(\log(1/\rho))$ random strings r, we can of course guarantee with probability $1-\rho$ at least one is good, but we must ensure two executions of the learner identifies the *same* good string. In particular, it is enough to give a replicable procedure to decide whether each sampled string contains a hypothesis cluster. We give a new procedure for this task. We show this can be done simply by producing few hypotheses (around $O\left(\frac{1}{\rho^2}\right)$) and running a randomized replicable test over their pairwise distances. This also allows us to find a hypothesis in the cluster, which we can output as our final solution. Thus, we obtain an approximately replicable learner with $O\left(\frac{d+\log(1/\beta)}{\rho^2\gamma^4\alpha^2}\right)$ samples.

Better Approximately Replicable Learning II: Boosting Correctness We now describe a method to improve the dependence on γ (at the cost of dependence on α) by giving an alternative way to reduce the error β. Toward this end, we begin with our $(0.01, \gamma)$-approximately replicable $(\alpha, 0.01)$-learner with $O\left(\frac{d}{\gamma^2\alpha^2}\right)$ samples.

To boost correctness, we again follow the basic repetition paradigm with a new wrinkle. Run the algorithm on $O(\log(1/\beta))$ fresh samples to obtain hypotheses $g_1, \ldots, g_d$, so that at least one is α-accurate with high probability. Similar to before, our goal is to use a variant of replicable hypothesis selection to select an α-optimal hypothesis. Indeed were the output g_i's over two runs actually to be the same, applying the best known replicable hypothesis selection algorithm [GKM21, HLYY25] would only require $O\left(\frac{\log(1/\beta)}{\rho^2\alpha^2}\right)$ samples.

Of course this is not the case; we are only promised each $g_i^{(1)}, g_i^{(2)}$ are γ-close. To handle this, we need a new variant of replicable hypothesis selection with a stronger notion of replicability that produces the same output whenever the input distributions are *similar*, rather than identical. Towards this, we say a hypothesis selection algorithm is (ρ, τ)-*robustly replicable* if under the assumption that $g_i^{(1)}, g_i^{(2)}$ are τ-close for all i, the algorithm selects the same hypothesis with probability $1-\rho$. We build upon the analysis of replicable hypothesis selection algorithms to show that the algorithm of [HLYY25] is in fact $(\rho, \rho\alpha)$-robustly replicable. Thus, setting $\gamma \ll \frac{\alpha}{\log(1/\beta)}$, we can use $\frac{d\text{polylog}(1/\beta)}{\alpha^2\min(\gamma,\alpha)^2}$ samples to obtain $O(\log(1/\beta))$ hypotheses that are $\ll \alpha$-close to each other. Then, our robustly replicable hypothesis selection algorithm selects the same index with high constant probability, producing the desired $(0.01, \gamma)$-approximately replicable (α, β)-learner. Boosting replicability ρ as above yields our approximately replicable learner with $O\left(\frac{d\text{polylog}(1/\beta)}{\rho^2\gamma^2\min(\gamma,\alpha)^2}\right)$ samples.

Sample Lower Bound Like in the pointwise replicable setting, we'd like to prove a sample lower bound for approximate replicability via reducing from bias estimation, but doing so is no longer as straightfoward as 'planting' a single randomized domain point. In particular, while pointwise replicability guaranteed our answer is replicable on *every* $x \in \mathcal{X}$ (and therefore also on whichever element we embed the bias estimation instance), in approximate replicability the algorithm only needs to be consistent on a γ-fraction of the domain, so it is entirely possible the algorithm is never consistent on the embedded instance.

Morally, we handle this by making sure the embedded x^* used for bias estimation is completely indistinguishable from the other elements to the algorithm, and therefore that x^* usually lies in the $1-\gamma$ fraction of elements that do replicate. This differs in a subtle but important way from our pointwise lower bound,

which also uses a form of indistinguishability for x^*. The difference is that under the pointwise guarantee, we only needed indistinguishability to guarantee correctness (and thus only when the label of x^* is sufficiently biased), while here we need to ensure x^* remains indistinguishable for *every* input distribution to the bias estimation problem.

To do this, we will critically rely on the fact that replicable bias estimation is hard even in the following *average-case* setting: even when the adversary selects $\mathrm{Rad}(p)$ with $p \in [-\alpha, \alpha]$ chosen uniformly at random (and this distribution is known to the learner), replicable bias estimation *still* requires $\Omega\left(\frac{1}{\rho^2\alpha^2}\right)$ samples. Thus, assuming that our input distribution to the bias estimation is drawn from this meta-distribution,[2] we can draw $d-1$ other dummy input distributions from the same meta-distribution, thereby hiding the true distribution from the algorithm.

More formally, consider the following reduction. Let $\mathcal{D}$ denote an (unknown) distribution over $\{\pm 1\}$ to which we have sample access. Let $\{x_1, \ldots, x_d\}$ denote a set of d shattered elements, and consider a uniform distribution over the d elements. We define the label distribution as follows. Randomly select an index $r \in [d]$ and generate labels according to $\mathcal{D}$. For every other index, generate labels according to a (shared) distribution sampled from the hard meta-distribution. Note that this process defines a meta-distribution over input distributions to our learning algorithm, and this meta-distribution is a d-wise product of the hard meta-distribution for bias estimation.

Suppose we have an (ρ, γ)-approximately replicable $(0.01\alpha, \beta)$-learner taking m samples. If we have a randomized algorithm that is correct and replicable on every distribution, a standard minimax-style argument shows there is also a deterministic algorithm that is correct and replicable on a random distribution (drawn from the product meta-distribution). In particular, even if every element's label (including the one embedding the bias estimation problem) is drawn from the hard meta-distribution, we have an algorithm that is correct and replicable with respect to "most" of the label distributions.

Now, consider the random variable err_i denoting the error of the hypothesis on x_i. Over the product meta-distribution, err_i are distributed identically, and therefore at most α on average. In particular, $\mathrm{err}_i = O(\alpha)$ with high constant probability. Furthermore, since (over the meta-distribution) all elements of the domain are indistinguishable, the approximately replicable learner is consistent on the relevant label with probability $1 - O(\rho + \gamma)$, by union bounding over the event that the algorithm produces a γ-consistent hypothesis, and that the relevant element lies in the consistent region of the hypotheses. Thus, we obtain a $O(\alpha)$-accuracy algorithm for bias estimation with $O(\rho + \gamma)$-replicability. Since our algorithm only takes $O(m/d)$ samples from the relevant element with high probability, we conclude $m = \Omega\left(\frac{d}{(\rho+\gamma)^2\alpha^2}\right)$.

Threshold Learning As we can see above, our algorithm for general VC classes loses various factors in the sample complexity over our lower bound. Moreover, the algorithm is *improper*, outputting a threshold over hypotheses in $\mathcal{H}$ rather than a function in $\mathcal{H}$ itself. It is very interesting to ask whether either of these weaknesses can be resolved. Next, we briefly sketch an elementary argument showing that at least for 1-D Thresholds (which we recall are not learnable in the fully replicable model), there is a simple quantile-estimation based algorithm that (essentially) achieves both.

Toward this end, let $\mathcal{D}$ be the data distribution and $\mathcal{D}_{\mathcal{X}}$ the induced distribution on unlabeled samples. We begin by approximately learning α-quantiles of $\mathcal{D}_{\mathcal{X}}$, which allows us to obtain $\frac{1}{\alpha}$ thresholds such that there is a quantile within α of any threshold. By taking $\frac{1}{\min(\rho\alpha,\gamma)^2}$ samples, we can learn each quantile up to error $\min(\rho\alpha, \gamma)$.

Now, our goal is to replicably select one of the $\frac{1}{\alpha}$ empirical quantiles: since every pair of quantiles is within γ, this guarantees an approximately replicable algorithm. Since we have learned each quantile up to error $\ll \rho\alpha$, we can apply robustly replicable hypothesis selection (see Theorem 4.5 and discussion above) to

[2] Here we mean the adversary's distribution over distributions.

select the same index with high probability. In particular, this requires an additional *additive* term of $\frac{1}{\rho^2\alpha^2}$ samples.

Shared Randomness Recall for replicable prediction, we showed shared randomness is required by reduction to the bias estimation problem. Unfortunately, this reduction is not meaningful for approximate replicability — in particular such a reduction only rules out $(\rho, 0)$-approximate replicability, which is essentially equivalent to replicability itself. The more interesting case is of course to show that (ρ, γ)-approximate replicability requires shared randomness for some larger $\gamma > 0$. Below, we sketch an argument showing randomness is indeed necessary even for γ as large as $\frac{1}{2}$.

Following a similar reduction to our other lower bounds, it suffices to consider an algorithm that accomplishes the following task:

$$\text{Given sample access to } \mathrm{Rad}(p_1) \times \mathrm{Rad}(p_2), \text{ output } v \in \{\pm 1\}^2 \text{ such that}$$
$$(1)\ \max_i\{|p_i|\mathbb{1}[v_i \neq \mathrm{sign}(p_i)]\} \leqslant \alpha \text{ and } (2)\ \Pr(\left\|v^{(1)} - v^{(2)}\right\|_0 \geqslant 1) < 0.01.$$

Above, (1) corresponds to correctly deciding the bias of each distribution, while (2) corresponds to an $(0.01, 0.5)$-approximately replicable algorithm. The idea is now as follows. Suppose there is an approximately replicable deterministic algorithm A. We may define for every $y \in \{\pm 1\}^2$ the sets

$$B_y := \{p \text{ s.t. } \Pr_{S \sim \mathrm{Rad}(p)^m}(A(S) = y) \geqslant 0.1\}.$$

That is, B_y contain the set of distributions where A outputs y with reasonable probability. Since there is at least one output with probability $\geqslant \frac{1}{4}$, the collection of B_y cover the cube $[-1, 1]^2$. Observe that if $B_x \cap B_y$ for $\|x - y\|_0 = 2$ then A is not approximately replicable for the distribution parameterized by $p \in B_x \cap B_y$, as with probability at least 0.02 the algorithm produces x, y on two distinct runs. Thus, it suffices to show such an intersection exists.

Suppose we wish to cover the unit square with 4 sets, one corresponding to each corner. Our goal is to show that at least one pair of opposite sets must intersect. Figure 1 gives an illustration.

Formally, consider the sets

$$\begin{aligned} B_{\text{left}} &= B_{(-1,-1)} \cup B_{(-1,1)} \\ B_{\text{right}} &= B_{(1,-1)} \cup B_{(1,1)} \\ B_{\text{up}} &= B_{(-1,1)} \cup B_{(1,1)} \\ B_{\text{down}} &= B_{(-1,-1)} \cup B_{(1,-1)} \end{aligned}$$

and the map

$$F : p \mapsto (\mathrm{dist}(p, B_{\text{left}}) - \mathrm{dist}(p, B_{\text{right}}), \mathrm{dist}(p, B_{\text{up}}) - \mathrm{dist}(p, B_{\text{down}}))\,.$$

By correctness, we know $p \in B_{\text{left}}$ on the left border of $[-1, 1]^2$, $p \in B_{\text{right}}$ on the right border, $p \in B_{\text{up}}$ on the up border, and $p \in B_{\text{down}}$ on the down border. Furthermore, as F is continuous, the Poincare-Miranda Theorem states that there exists p^* with $F(p^*) = (0, 0)$. Without loss of generality, we can assume $p^* \in B_{(1,1)}$ and therefore $p^* \notin B_{(-1,-1)}$. Since $p^* \in B_{\text{right}} \cap B_{\text{up}}$, we must have $p^* \in B_{\text{left}} \cap B_{\text{down}}$, or $p^* \in B_{(-1,1)} \cap B_{(1,-1)}$, obtaining a contradiction.

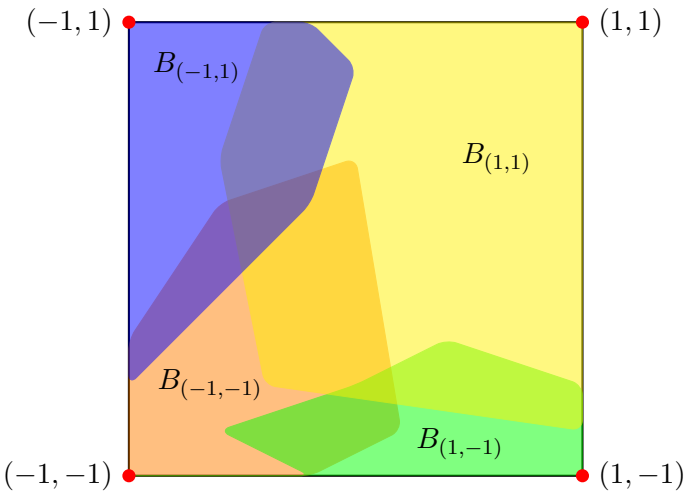

Figure 1: An example of sets B_y covering $[-1,1]^2$. The four regions must cover the whole square and correctness constraints require B_{left} (which consists of the blue $B_{(-1,1)}$ and orange $B_{(-1,-1)}$ regions) to contain the full left border. Similar constraints hold on $B_{\text{right}}, B_{\text{up}}, B_{\text{down}}$. Then, regardless of how the square is colored, at least one pair of opposing regions (either yellow and orange or blue and green) must intersect. In the given example, yellow $B_{(1,1)}$ and orange $B_{(-1,-1)}$ intersect.

1.2.3 Semi-Replicable Learning

Our algorithm for semi-replicable learning follows the same basic two-step framework introduced for semi-private learning [ABM19, HKLM22]: first we use $\frac{d}{\alpha}$ shared unlabeled samples to compute a small set of roughly $(\frac{d}{\alpha})^{O(d)}$ hypotheses $C \subset \mathcal{H}$ that almost certainly contains a good classifier, and, second, we use a replicable learner for finite classes from [BGH+23] to learn C. The latter requires roughly $\log^2 |C|$ labeled samples, which gives the desired bound. Thus perhaps the more interesting result lies in showing this simple procedure is sample-optimal, both in terms of the shared and unshared sample complexities.

Shared Sample Complexity Our first lower bound, on the shared sample complexity of semi-replicable learning, is actually proved via a reduction to the so-called *semi-private* learning model, along with a new lower bound for this setting built on [ABM19]. Intuitively, a semi-private algorithm has access to public dataset D_{pub} and private dataset D_{priv} and must adhere to privacy constraints only on the private dataset (the exact notion of approximate DP used below is standard, but not important for the proof overview so we omit the exact definition, see Definition 2.7):

An algorithm A is (ε, δ)-semi-private if given any public dataset D_{pub}, $A(D_{\text{pub}}, \cdot)$ is (ε, δ)-private.

Following existing tools [GKM21, BGH+23], it is easy to transform any semi-replicable algorithm into a 'semi-private' one with roughly matching sample complexity. Thus, if we can show that no semi-private algorithm exists using $o(\frac{d}{\alpha})$-public samples, it will imply a corresponding lower bound for semi-replicable algorithms on $o(\frac{d}{\alpha})$-shared samples. Toward this end, we will rely heavily on a result of [ABM19] who prove any semi-private algorithm for a class with infinite Littlestone dimension must use $\Omega\left(\frac{1}{\alpha}\right)$ public samples. Our main goal is to improve this bound to $\Omega\left(\frac{d}{\alpha}\right)$ for a specific class of infinite Littlestone Dimension and VC Dimension d: the class of conjunctions of d thresholds, i.e. $x \mapsto \bigwedge_{i=1}^{d} h_i(x_i)$.

As in approximate replicability, the main idea in this improvement is to hide one "relevant" threshold among d dummy thresholds, forcing the algorithm to use an additional multiplicative factor of d samples. In

particular, given one hard distribution over datasets $(D_{\text{pub}}, D_{\text{priv}})$, we construct an instance that consists of a uniform distribution over d distinct thresholds. In particular, any (α, β)-algorithm that solves the problem over d thresholds must be accurate on a large constant fraction of the d thresholds. Since the thresholds are drawn from the same hard distribution, they are indistinguishable to the algorithm, so that the error of our algorithm on the single "relevant" threshold is $O(\alpha)$. However, any semi-private algorithm for thresholds requires $\Omega\left(\frac{1}{\alpha}\right)$ public samples, so that our original algorithm requires $\Omega\left(\frac{d}{\alpha}\right)$ public samples to be semi-private and accurate on $\Omega(d)$ thresholds.

Labeled Sample Complexity To obtain a lower bound on the labeled sample complexity, we take inspiration for the lower bound for replicably learning VC classes in [BGH+23] and reduce from the d-dimensional sign-one-way marginals problem. Let $\mathcal{D}_p$ denote a Rademacher product distribution (supported on $\{\pm 1\}^d$ with mean p). A solution $v \in \{\pm 1\}^d$ is α-accurate if $\frac{1}{d}\sum_i \mathbb{1}[v_i \neq \text{sign}(p_i)]|p_i| \leqslant \alpha$. A standard reduction shows that $\hat{v} = (h(x_1), \ldots, h(x_d))$ is α-accurate if and only if $\text{err}_{\mathcal{D}}(h) \leqslant \alpha$ [BGH+23, HLYY25]. Since our learning algorithm is replicable, we obtain a replicable algorithm for d-dimensional sign-one-way marginals, which [HLYY25] recently showed requires $\Omega\left(\frac{d^2}{\rho^2\alpha^2}\right)$ samples, giving the desired bound.

1.3 Conclusion and Open Questions

Replicability is an interesting and highly desirable property of learning algorithms, but comes at a major cost due to its extreme requirements (making many classes quadratically harder to learn, and others like 1-D thresholds simply impossible). In this work, we have shown that several natural relaxations of replicability are feasible for *any* learnable class, often at very little cost for constant replicability parameters.

Our work leaves open several interesting questions, most obviously settling the sample complexity of pointwise and approximate replicability, but also whether either notion can be achieved *properly* for any class. Below, we discuss a few challenges and potential directions toward resolving the former of these questions.

For replicable prediction, we obtain a ρ-pointwise replicable $(\alpha, 0.01)$-learner with $\tilde{O}\left(\frac{d}{\rho^2\alpha^2}\right)$. Our lower bound in fact applies to small constant β, so this is the optimal sample complexity even to obtain small constant β. Unfortunately, since our algorithm computes labels for each domain element x by averaging many hypotheses, the labels for different domain elements can be arbitrarily correlated. As a result, we are only able to bound the expected error of our algorithm, and only obtain high probability bounds via Markov's inequality, costing us an extra factor of ρ^2. Typically, one can apply replicable amplification to reduce the complexity back to just ρ^2, but all known methods require running the replicable algorithm many times and strongly rely on the fact that all repeated hypotheses are exactly the same over two runs, which of course fails under pointwise replicability.

On the topic of approximate replicability, it is interesting to note that even a simpler variant remains unsolved: approximately replicable bias estimation.

> Given samples to a binary product distribution with mean $p \in [-1, 1]^n$, return $v \in \{\pm 1\}^n$ satisfying,
> (1) $\max_i \mathbb{1}[v_i \neq \text{sign}(p_i)]|p_i| \leqslant \alpha$ and (2) $\Pr(\left\|v^{(1)} - v^{(2)}\right\|_0 > \gamma) < \rho$.

In other words, the algorithm (with shared randomness and given independently sampled data) must output (with $1-\rho$ probability) hypotheses that disagree on at most a γ-fraction of coordinates. On the lower bound side, we obtain a $\Omega\left(\frac{d}{(\rho+\gamma)^2\alpha^2}\right)$ lower bound, while [HIK+24] show that $\Omega\left(\frac{d}{\gamma\alpha^2}\right)$ samples are necessary to obtain a $(0.01, \gamma)$-approximately replicable algorithm. On the other hand, the best algorithms for this problem need $O\left(\min\left(\frac{d^2}{\rho^2\alpha^2}, \frac{d\cdot\text{polylog}(1/\rho)}{\gamma^2\alpha^2}\right)\right)$ samples, leaving a substantial gap in sample complexity.

1.4 Further Related Work

Replicable Learning Replicable learning was introduced by [GKM21, ILPS22] and has played a central role in recent years in the study of various notions of stability [KKMV23, MSS23, BGH+23], notably including differential privacy [BGH+23]. Replicable algorithms are now known for a variety of classical problems, including mean estimation [ILPS22, HIK+24, VWDP+24], distribution testing [LY24, DGK+25, AAC+25], clustering [EKM+23], PAC Learning [BGH+23, KKL+24, LMS25], and reinforcement learning [EHKS23, KVYZ23, HLYY25, EHK+25]. [ILPS22, KKVZ24] study replicable quantile estimation over finite domains, while our threshold algorithm requires approximately replicable quantile estimation over arbitrary domains. Conversely, various works have also shown strong lower bounds implying replicability is either more costly than standard learning [HIK+24, HLYY25] or impossible (e.g. for learning infinite Littlestone classes) [BGH+23]).

Finally, related to our lower bounds on shared randomness, several recent works have studied the number of random bits needed to achieve replicability [CMY23, DPWV23, VWDP+24, CCMY24, HM25]. [CMY23] in particular also use Poincare-Miranda for this purpose, though list replicability and approximate replicability are incomparable: a list replicable algorithm may have few very different outputs, while an approximately replicable algorithm may product (infinitely) many similar outputs.

Approximate Replicable Learning Several relaxations of replicability have been studied or proposed in the literature [KVYZ23, CMY23, KKMV23, HIK+24]. Our notion of approximate replicability, for instance, was proposed as an open direction in [CMY23]. Concretely, approximately replicable reinforcement learning was studied in [KVYZ23] and bias estimation in [HIK+24]. While [KKMV23] does not explicitly study approximate replicability, their notion of indistinguishability between output distributions could be interpreted as a related notion of approximate replicability.

Our work also relates more broadly to the study of notions of approximate stability in learning, starting with the seminal work of [BE02] who require output hypotheses to have similar error under small perturbations to the training set. These notions are typically incomparable to the ones we study — they are noticeably weaker in the sense they only require similar loss rather than similar output hypotheses, but do not require shared randomness as we show our models must.

Private Prediction Private Prediction was introduced by Dwork and Feldman [DF18] where they give private predictors for PAC learning and convex regression. Subsequent works [BTT18, DF20] improve the sample complexity of agnostic PAC learning and [vdMH20] gave an experimental study of private prediction algorithms. Using known connections between differential privacy and replicability, it is possible to translate any private predictor into a replicable one at roughly quadratic overhead. Thus such bounds would scale quadratically with the VC dimension, while we achieve linear dependence directly.

The literature has also covered interesting extensions of the private prediction model such as 'everlasting' prediction, which allows an infinite stream of queries preserving privacy [NNSY23, Ste24]. It would be very interesting to show an analogous result in the replicable setting that decays slower than the trivial union bound argument against multiple such prediction queries.

Semi-Private Learning Semi-private learning, introduced by [BNS16a], studies how access to public samples may be helpful for private learning. Semi-privacy has been studied in various settings, including estimation and distribution learning [BKS22, BDBC+23], query release [LVS+21, BCM+20], feature learning [KMR+24], stochastic optimization [UMB+24], and PAC learning [BTT18, BBD+24]. [ABM19] settles the public sample complexity of semi-private learning for constant VC classes.

2 Preliminaries

Let $[n] = \{1, \dots, n\}$. We use $\wedge$ to denote logical AND and $\vee$ to define logical OR. For any distribution $\mathcal{D}$, let $\mathcal{D}^m$ denote the product of $\mathcal{D}$ taken m times (i.e. taking m i.i.d. samples from $\mathcal{D}$). Let $\mathcal{M}$ denote a meta-distribution, i.e. a distribution over distributions. Let $\mathrm{Rad}(p)$ denote the Rademacher distribution (i.e. a distribution supported on $\{\pm 1\}$) with mean p i.e. $\Pr(X = 1) = \frac{1+p}{2}$. If p is a vector, $\mathrm{Rad}(p)$ denotes a product of Rademacher distributions where each coordinate has mean p_i.

Statistical Learning Let f, g be functions $\mathcal{X} \to \{\pm 1\}$. We typically use x to denote samples from the domain $\mathcal{X}$ and y to denote labels i.e. $\{\pm 1\}$. For any distribution $\mathcal{D}$ over $\mathcal{X} \times \{\pm 1\}$, the distribution error of f is $\mathrm{err}_{\mathcal{D}}(f) := \Pr_{(x,y)\sim\mathcal{D}}(f(x) \neq y)$. The classification distance of f, g is $\mathrm{dist}_{\mathcal{D}}(f, g) := \Pr_{x\sim\mathcal{D}}(f(x) \neq g(x))$. For any sample $S \in (\mathcal{X} \times \{\pm 1\})^m$, the empirical error of f is $\mathrm{err}_S(f) := \frac{1}{m}\sum_{i=1}^m \mathbb{1}[f(x_i) \neq y_i]$. The empirical classification distance of f, g is $\mathrm{dist}_S(f, g) := \frac{1}{m}\sum_{i=1}^m \mathbb{1}[f(x_i) \neq g(x_i)]$.

Definition 2.1 (VC Dimension)**.** *A hypothesis class $\mathcal{H} \subset \mathcal{X} \to \{\pm 1\}$ has VC-dimension d if d is the largest integer such that there exists a set of d samples $\{x_1, \dots, x_d\}$ for which any labeling of the d samples can be realized by a hypothesis $h \in \mathcal{H}$. That is, $|(h(x_1), \dots, h(x_d)) \text{ s.t. } h \in \mathcal{H}| = 2^d$. We say such a set is shattered.*

Theorem 2.2 (Uniform Convergence, e.g. [BEHW89])**.** *Let $\mathcal{H}$ be a binary class of functions with domain $\mathcal{X}$ with VC Dimension d. Then, for any distribution $\mathcal{D}$ over $\mathcal{X}$ and all $m > 0$,*

$$\Pr_{x_1,\dots,x_m\sim\mathcal{D}}\left(\sup_{h\in\mathcal{H}}\left|\frac{1}{m}\sum_{i=1}^m \mathbb{1}[h(x_i) = 1] - \Pr_{z\sim\mathcal{D}}(h(z) = 1)\right| \geqslant \gamma\right) \leqslant 4(2m)^d e^{-\gamma^2 m/8}.$$

Definition 2.3 (PAC Learning [Val84])**.** *An algorithm A is an agnostic (α, β)-learner for $\mathcal{H}$ if for any distribution $\mathcal{D}$ over $\mathcal{X} \times \{\pm 1\}$, $\Pr_{S\sim\mathcal{D}^m}(\mathrm{err}_{\mathcal{D}}(A(S)) > \mathrm{OPT} + \alpha) < \beta$ where $\mathrm{OPT} := \inf_{h\in\mathcal{H}} \mathrm{err}_{\mathcal{D}}(h)$ and m is the sample complexity of A.*

We say that A is a realizable (α, β)-learner for $\mathcal{H}$ if for any $\mathcal{D}$ over pairs $(x, h(x))$ where $x \sim \mathcal{D}_{\mathcal{X}}$ and $h \in \mathcal{H}$, $\Pr_{S\sim\mathcal{D}^m}(\mathrm{err}_{\mathcal{D}}(A(S)) > \alpha) < \beta$.

We note that any class of VC dimension d has an agnostic (α, β)-learner with $O\left(\frac{d+\log\frac{1}{\beta}}{\alpha^2}\right)$ samples [Hau92] and an (improper) realizable (α, β)-learner with $O\left(\frac{d+\log(1/\beta)}{\alpha}\right)$ samples [Han16, Lar23].

Replicability We use A to denote algorithms, while $A(S; r)$ denotes executing A on samples S with internal randomness r.

Definition 2.4 (Replicability [ILPS22])**.** *A randomized algorithm is ρ-replicable if for every distribution $\mathcal{D}$,*

$$\Pr_{S_1,S_2\sim\mathcal{D}^m,r}(A(S_1; r) \neq A(S_2; r)) < \rho.$$

Lemma 2.5 (Correlated Sampling, Lemma 7.5 of [RY20])**.** *Let $\mathcal{X}$ be a finite domain. There is a randomized algorithm* CorrSamp *(with internal randomness $\xi \sim \mathcal{D}_R$) such that given any distribution over $\mathcal{X}$ outputs a random variable over $\mathcal{X}$ satisfying the following:*

1. *(Marginal Correctness) For all distributions $\mathcal{D}$ over $\mathcal{X}$,*

$$\Pr_{\xi\sim\mathcal{D}_R}(\textsc{CorrSamp}(\mathcal{D}, \xi) = x) = \Pr_{X\sim\mathcal{D}}(X = x).$$

2. *(Error Guarantee) For all distributions $\mathcal{D}, \mathcal{D}'$ over $\mathcal{X}$,*

$$\Pr_{\xi \sim \mathcal{D}_R}\left(\textsc{CorrSamp}(\mathcal{D}, \xi) \neq \textsc{CorrSamp}(\mathcal{D}', \xi)\right) \leqslant 2 \cdot \mathrm{tvd}(\mathcal{D}, \mathcal{D}').$$

Furthermore, the algorithm runs in expected time $\tilde{O}(|\mathcal{X}|)$.

The learnability of a class using replicable algorithms is characterized by what is known as Littlestone Dimension. The Littlestone dimension is a combinatorial parameter that characterizes regret bounds in Online Learning [Lit88, BDPSS09].

Definition 2.6 (Littlestone Dimension)**.** *Let $\mathcal{H}$ be a hypothesis class over $\mathcal{X}$. A mistake tree is a binary decision tree whose internal nodes are labeled by elements of $\mathcal{X}$. Any root-to-leaf path in a mistake tree can be described as a sequence of examples $(x_1, y_1), \ldots, (x_d, y_d)$ where x_i is the label of the i-th node in the path and $y_i = 1$ if the $(i+1)$-th node in the path is the right child of the i-th node and otherwise $y_i = -1$. We say that a tree T is shattered by $\mathcal{H}$ if for any in T, there is $h \in \mathcal{H}$ such that $h(x_i) = y_i$, for all $i \leqslant d$. The Littlestone dimension of $\mathcal{H}$, denoted by $\mathrm{LDim}(\mathcal{H})$, is the depth of largest complete tree that is shattered by $\mathcal{H}$.*

Differential Privacy We define differential privacy.

Definition 2.7. *Let $A : \mathcal{X}^m \to \mathcal{Y}$ be an algorithm. Two datasets $S, S' \in \mathcal{X}^m$ are neighboring if are identical in all but one sample. A is (ε, δ)-private if for every pair of neighboring datasets S, S' and every set of outputs $Y \subset \mathcal{Y}$,*

$$\Pr(A(S) \in Y) \leqslant e^{\varepsilon} \Pr(A(S') \in Y) + \delta.$$

Distribution Divergence and Distance In the following, let X, Y be random variables. Unless otherwise specified, the random variables are over domain $\mathcal{X}$.

Definition 2.8. *The* total variation distance *between X and Y is*

$$\mathrm{tvd}(X, Y) = \frac{1}{2} \sum_{x \in \mathcal{X}} |\Pr(X = x) - \Pr(Y = x)|.$$

3 Replicable Prediction

In this section, we examine the special case of pointwise replicability in binary classification (over $\{\pm 1\}$), which we call *replicable prediction*. In particular, we show that any standard PAC-learner can be amplified to a (pointwise) replicable predictor with only polynomial overhead in ρ.

Theorem 3.1 (Formal Theorem 1.4)**.** *Let A be an (agnostic) (α, β)-learner on $m(\alpha, \beta)$ samples. There exists an (agnostic) ρ-pointwise replicable (α, β)-learner with sample complexity*

$$O\left(\frac{m(\alpha\rho, \alpha\rho)\log(1/\beta)}{\rho^2} + \frac{\log(1/\beta)\log\log(1/\beta)}{\alpha^2}\right).$$

Furthermore, our algorithm runs in time linear in sample complexity with $O\left(\frac{\log(1/\beta)}{\rho^2}\right)$ oracle calls to A.

Proof. We first give a simple procedure that produces an (agnostic) ρ-pointwise replicable (α, β) PAC-learner with polynomial sample dependence on β (i.e. we first obtain an algorithm with small constant error probability). Later, we use this as a sub-routine to obtain a ρ-pointwise replicable and (α, β)-accurate learner.

Algorithm 1: BASICPOINTWISEREPLICABILITY$(\mathcal{A}, \alpha, \beta, \rho)$

Input : PAC-learner $\mathcal{A}$ with sample complexity $m(\alpha, \beta)$.
Parameters : α accuracy, β error probability, and ρ replicability
Output : ρ-pointwise replicable (α, β)-accurate PAC-learner $\tilde{\mathcal{A}}$.

$T \leftarrow \frac{C}{\rho^2}$ for a sufficiently large constant C.
Run $\mathcal{A}$ on T fresh samples of size $m(\alpha\beta/2, \alpha\beta/2)$, obtaining hypotheses $h_1, \ldots, h_T : \mathcal{X} \to \{\pm 1\}$.
Draw random threshold $r \in [0, 1]$.
For each $x \in \mathcal{X}$, let $\hat{p}_x \leftarrow \frac{1}{T}\sum_{i=1}^{T} h_i(x)$.
return $h : x \mapsto \mathbf{1}[\hat{p}_x \geqslant \alpha_x]$.

We first argue this procedure is ρ-pointwise replicable. For any fixed x, let $p_x := \mathbb{E}_S[\Pr(A(S)(x) = 1]$ denote true expectation of $A(\cdot)(x)$ and observe that for large enough C, $\mathrm{Var}(\hat{p}_x) \leqslant \rho^2$ since $\hat{p}_x \sim \frac{\mathrm{Binom}(T, p_x)}{T}$. Then, the probability that α_x falls between $\hat{p}_x^{(1)}, \hat{p}_x^{(2)}$ over two runs of the algorithm is

$$\begin{aligned}
\Pr_{\alpha_x, \hat{p}_x^{(1)}, \hat{p}_x^{(2)}} \left(\alpha_x \in \left(\min\left(\hat{p}_x^{(1)}, \hat{p}_x^{(2)}\right), \max\left(\hat{p}_x^{(1)}, \hat{p}_x^{(2)}\right)\right)\right)^2 &= \mathbb{E}_{\hat{p}_x^{(1)}, \hat{p}_x^{(2)}}\left[\left|\hat{p}_x^{(1)} - \hat{p}_x^{(2)}\right|\right]^2 \\
&\leqslant \mathbb{E}_{\hat{p}_x^{(1)}, \hat{p}_x^{(2)}}\left[\left(\hat{p}_x^{(1)} - \hat{p}_x^{(2)}\right)^2\right] \\
&= \mathrm{Var}_{\hat{p}_x^{(1)}, \hat{p}_x^{(2)}}\left[\hat{p}_x\right] \leqslant \rho^2.
\end{aligned}$$

Here, in the first line we have used for any fixed $\hat{p}_x^{(1)}, \hat{p}_x^{(2)}$, the probability r lies between $\hat{p}_x^{(1)}, \hat{p}_x^{(2)}$ is exactly the width of the interval between the two endpoints. In the second line, we have used Jensen's inequality, and the final line is simply the definition of variance. Thus, for any fixed x, the probability that the algorithm outputs different $h(x)$ on two different runs is at most ρ.

We now argue this procedure has small expected error. This follows from the fact that our algorithm (denoted $\tilde{A}_1$) is an unbiased estimator, or in particular that

$$\begin{aligned}
\mathbb{E}_{S\sim D^{nT}, r}[\mathrm{err}_{\mathcal{D}}(\tilde{A}_1)] &= \mathop{\mathbb{E}}_{S\sim D^{nT}, r, (x,y)\sim D}\left[\mathbf{1}[\tilde{A}_1(S; r)(x) \neq y]\right] \\
&= \mathop{\mathbb{E}}_{(x,y)\sim D}\left[\mathop{\mathbb{E}}_{S\sim D^{nT}, r}\left[\mathbf{1}[\tilde{A}_1(S; r)(x) \neq y]\right]\right] \\
&= \mathop{\mathbb{E}}_{(x,y)\sim D}\left[\Pr_{S\sim D^{nT}, r}\left[\tilde{A}_1(S; r)(x) \neq y\right]\right],
\end{aligned}$$

so it is enough to observe that for any fixed x:

$$\Pr_{S\sim D^{nT}, r\in[0,1]}\left[\tilde{A}_1(S; r)(x) = 1\right] = \Pr_{S_1, \ldots, S_T \sim D^n, r\in[0,1]}\left[\frac{1}{T}\sum_{i=1}^{T} A(S_i)(x) > r\right] = p_x.$$

In particular, we have that expected error of $\tilde{A}_1$ is the expected error of A and therefore at most $\alpha\beta$. Applying Markov's Inequality to the non-negative random variable $\mathrm{err}_{\mathcal{D}}(\tilde{A}_1(S; r)) - \mathrm{OPT}$, we have that

$$\Pr_{S\sim D^{nT}, r}(\mathrm{err}_{\mathcal{D}}(\tilde{A}_1(S; r)) > \mathrm{OPT} + \alpha) < \beta.$$

The above discussion also yields the following intermediate result.

Lemma 3.2. *Let A be an (agnostic) (α, β)-learner on $m(\alpha, \beta)$ samples. There exists an (agnostic) ρ-pointwise replicable (α, β)-learner with sample complexity*

$$O\left(\frac{m(\alpha\beta, \alpha\beta)}{\rho^2}\right).$$

For constant β, we obtain an algorithm with $O\left(\frac{m(\alpha,\alpha)}{\rho^2}\right)$ samples. However, we would like to efficiently amplify our algorithm to arbitrary confidence with only the typical $\mathrm{polylog}(1/\beta)$ dependence in sample complexity. Consider the following procedure toward this end.

Algorithm 2: BOOSTPOINTWISEERROR$(\mathcal{A}, \alpha, \beta, \rho)$

Input : ρ-pointwise-replicable (α, ρ)-accurate PAC-learner A.
Parameters : α accuracy, β error probability, and ρ replicability
Output : $(\rho + 2\beta)$-pointwise replicable $(2\alpha, \beta)$-accurate PAC-learner.

1 Run A on $O(\log(1/\beta))$ fresh samples and denote the output list $\{\tilde{h}_i\}$.
2 Test the error of each output on $m_{\text{test}} := O\left(\frac{\log\log(1/\beta)}{\alpha^2}\right)$ fresh samples
3 Output the first hypothesis with empirical error at most $\frac{\alpha}{2}$ greater than the minimum in the list.

We begin by arguing for correctness. Since A produces an α-accurate hypothesis with probability 0.99, the probability that none of the output hypotheses are α-accurate is at most $0.01^{\log(1/\beta)} < \beta$. By a standard Chernoff bound, the empirical error of each hypothesis is within $\frac{\alpha}{2}$ of the distributional error. Since we output a hypothesis with empirical error at most $\frac{\alpha}{2}$ greater than the minimum in the list, the algorithm outputs a 2α-accurate hypothesis with probability at least $1 - \beta$. Here we remark that if the optimal hypothesis has error $\leqslant \alpha$ (i.e. the realizable setting), then we only need $O\left(\frac{\log\log(1/\beta)}{\alpha}\right)$ samples to distinguish hypotheses with error at most 2α from those with error at least 3α. In particular, we instead return the first hypothesis with empirical error at most 2α (i.e. distributional error at most 3α).

Now, we argue that this procedure is 2ρ-pointwise replicable. Over both runs, the first hypothesis is α-accurate with probability $1 - 2\rho$. By a union bound, both runs of the algorithm return the first hypothesis with probability at least $1 - 2\rho$. Finally, we note that for all fixed x, the outputs of the first hypothesis agree on x with probability ρ since A is ρ-pointwise replicable.

The theorem follows by applying Algorithm 2 to a ρ-pointwise replicable (α, ρ)-learner obtained from Lemma 3.2 to obtain a 2ρ-pointwise replicable $(2\alpha, \beta)$-accurate algorithm.

□

3.1 Lower Bounds for Replicable Prediction

In this section, we give lower bounds for replicable prediction showing that pointwise replicability has at least quadratic overhead in the replicability parameter ρ over standard learning, similar to the overhead seen in classical bias estimation [ILPS22]. Our lower bounds apply to any VC class.

Theorem 1.5. *Any ρ-pointwise replicable $(0.01\alpha, 0.0001)$-learner requires $\Omega\left(\frac{d}{\rho^2\alpha^2\sqrt{\log(1/\rho)}}\right)$ samples. Furthermore, any pointwise-replicable learner requires shared randomness. Any ρ-pointwise replicable realizable $(0.01\alpha, 0.0001)$-learner requires $\Omega\left(\frac{d}{\rho^2\alpha\sqrt{\log(1/\rho)}}\right)$ samples.*

We begin with the agnostic lower bound. See Theorem 3.4 for the realizable lower bound. Our lower bound follows by reduction to bias estimation, also known as one-way marginals.

Definition 3.3 (One-Way Marginals)**.** *Let $\mathcal{D}_p$ be a product of d Rademacher distributions with expectations $p = (p_1, \ldots, p_d)$. A vector $v \in \{\pm 1\}^d$ is an α-accurate solution to the one-way marginals problem for $\mathcal{D}_p$ if $\max_{i=1}^{d}(\mathrm{sign}(p_i) - v_i)p_i \leqslant 2\alpha$.*

An algorithm (α, β)-accurately solves the sign-one-way marginals problem with sample complexity m if given any product of d Rademacher distributions $\mathcal{D}$ and m i.i.d. samples from $\mathcal{D}$, with probability at least $1 - \beta$ the algorithm outputs an α-accurate solution to the sign-one-way marginals problem for $\mathcal{D}$.

We will use bias estimation to refer to the 1-dimensional one-way marginals problem i.e. given sample access to $\mathrm{Rad}(p)$, return 1 if $p \geqslant \alpha$ and -1 if $p \leqslant -\alpha$.

Proof. Since $\mathcal{H}$ is has VC dimension d, there is a set of d elements shattered by $\mathcal{H}$. Denote them $\{x_1, \ldots, x_d\}$. We will use the following family of input distributions, parameterized by mean vectors $p \in [-1, +1]^d$. Fix a mean vector $p \in [-1, +1]^d$. First, a domain element x_i is sampled uniformly. Then, if the domain element x_i is sampled, the label is distributed according to $\mathrm{Rad}(p_i)$. Suppose there is a pointwise replicable PAC-learner A for $\mathcal{H}$ using m samples. Assume without loss of generality that $m \geqslant d$. In the proof, we in fact assume A learns a class with VC dimension $2d + 1$.

Given A, we design a ρ-replicable algorithm for bias estimation. To this end, suppose we are given sample access to a Rademacher distribution with mean $p \in [-1, +1]$ and are required to output 1 if $p \leqslant -\alpha$ and -1 if $p \geqslant \alpha$. From [ILPS22], it is known that any algorithm for this task (that succeeds with error probability $\beta \leqslant 0.1$) requires $\Omega\left(\frac{1}{\rho^2\alpha^2}\right)$ samples and at least 1 bit of shared randomness, even when it is known that $p \in [-\alpha, \alpha]$. We give an algorithm for bias estimation in Algorithm 3.

Algorithm 3: BIASESTIMATORPOINTWISE(A, α, ρ)

Input : ρ-pointwise replicable $(\alpha, 0.0001)$-accurate PAC-learner A. Sample access to unknown Rademacher distribution $\mathcal{D}$ with mean $p \in [-\alpha, \alpha]$.
Parameters : α accuracy and ρ replicability
Output : 2ρ-replicable algorithm for bias estimation.

Initiate S as an empty multi-set.
Initiate counter $c \leftarrow 0$.
Sample $r \in [2d + 1]$ uniformly.
Split $[2d + 1] \backslash \{r\}$ into two random equal subsets B_+, B_-.
for $j \in [m]$ **do**
    Sample $x_j \in \mathcal{X}$ uniformly.
    if $x_j = x_r$ **then**
        Draw a sample $y_j \sim \mathcal{D}$ and add (x_j, y_j) to S.
        $c \leftarrow c + 1$.
        if $c > 2\frac{m}{d}\sqrt{\log(1/\rho)}$ **then**
            return 1.
    else
        if $x_j \in B_+$ **then**
            Draw $y_j \sim \mathrm{Rad}(\alpha)$.
        if $x_j \in B_-$ **then**
            Draw $y_j \sim \mathrm{Rad}(-\alpha)$.
        Add (x_j, y_j) to S.
return $A(S)(x_r)$.

We claim our algorithm is ρ-replicable. We condition on the event that the sample limit is not reached. Let m_r denote the number of times x_r is sampled so that $\mu := \mathbb{E}[m_r] = \frac{m}{2d+1}$. By a standard Chernoff bound,

$$\Pr\left(m_r > \frac{m}{d}\left(1 + \sqrt{100\log(1/\rho)}\right)\right) < \rho.$$

Since B_-, B_+ are sampled with shared randomness, in both runs of the algorithm A is given sample access to the same distribution over $(\{x_1, \ldots, x_d\} \times \{\pm 1\})^m$ where labels are parameterized according to some mean vector $\overline{p} \in [-1, 1]^{2d+1}$. In particular, the label of x_r is then identical with probability at least $1 - \rho$ by pointwise replicability, so algorithm is 2ρ-replicable in total via a union bound.

Next, we claim our algorithm is accurate. Suppose without loss of generality that $p = \alpha$. (A similar argument holds when $p = -\alpha$ and when $p \in (-\alpha, \alpha)$ there is no correctness constraint). We analyze the error of any hypothesis $h : \mathcal{X} \to \{\pm 1\}$.

$$\begin{aligned}
\mathrm{err}_{\mathcal{D}}(h) &= \frac{1}{2d+1}\sum_{i=1}^{2d+1} \Pr_{(x,y)}(y \neq h(x) | x = x_i) \\
&= \frac{1}{2d+1}\sum_{i=1}^{2d+1} \mathbb{1}[h(x_i) = -1] \cdot \frac{1+p_i}{2} + \mathbb{1}[h(x_i) = 1] \cdot \frac{1-p_i}{2}
\end{aligned}$$

where p_i is the mean of the labels of x_i. Thus, the excess error of h against the optimal hypothesis h^* is

$$\mathrm{err}_{\mathcal{D}}(h) - \mathrm{err}_{\mathcal{D}}(h^*) = \frac{\alpha}{2d+1}\left(\sum_{i \in B_+ \cup \{r\}} \mathbb{1}[h(x_i) = -1] + \sum_{i \in B_-} \mathbb{1}[h(x_i) = 1]\right).$$

We condition on the event that $h \leftarrow A(S)$ is 0.01α-accurate and the sample limit is not reached. The first occurs with probability 0.9999 by the correctness of A, and by the above we then have

$$\frac{1}{2d+1}\sum_{i \in B_+ \cup \{r\}} \mathbb{1}[h(x_i) = -1] \leqslant 0.01.$$

In particular, out of the $d+1$ elements in $B_+ \cup \{r\}$, at most a 0.02-fraction of them have $h(x) = -1$. Now, for any fixed B_-, over the random choice of r, B_+, the samples are identically distributed so that A cannot distinguish different choices of r in $B_+ \cup \{r\}$. In particular, define $E_+ := \{i \in B_+ \cup \{r\} \text{ s.t. } h(x_i) = -1\}$ so that $\frac{|E_+|}{d+1} \leqslant 0.02$. Then, over the random choices of our algorithm and the samples observed from the unknown Rademacher distribution $\mathcal{D}$, we have

$$\Pr(r \in E_+) \leqslant \max_{B_-} \Pr_{r, B_+}(r \in E_+ | B_-) \leqslant 0.02.$$

Thus, conditioned on the event that the output hypothesis h is 0.01α-accurate, the probability that $h(x_r) = -1$ is at most 0.02. By a union bound, we output -1 with probability at most 0.03.

Finally, the sample complexity of our algorithm is immediate. In particular, we have $\frac{m}{d}\sqrt{\log(1/\rho)} = \Omega\left(\rho^{-2}\alpha^{-2}\right)$ which implies the desired lower bound.

To see that the algorithm requires shared randomness, note that the only stage Algorithm 3 uses shared randomness is in sampling B_-, B_+. However, to learn $\mathcal{H}$ with respect to a distribution which is supported over a single element, there is no need to sample these sets, so the reduction uses no shared randomness. Therefore, A must use shared randomness to learn $\mathcal{H}$. □

Above, we have shown a lower bound for pointwise replicable algorithms that learn in the agnostic setting. Below, we give a lower bound that applies in the realizable setting.

Theorem 3.4. *Any ρ-pointwise replicable $(\alpha, 0.0001)$-learner in the realizable setting requires sample complexity*

$$\Omega\left(\frac{d}{\rho^2\alpha\sqrt{\log(1/\rho)}}\right).$$

We give a reduction from a simpler version of bias estimation, where correctness is only required on the two constant distributions over $\{\pm 1\}$.

Definition 3.5. *An algorithm solves the $\{\pm 1\}$-bias estimation problem if given sample access to $\mathrm{Rad}(p)$, the algorithm returns 1 with probability $2/3$ when $p = 1$ and returns -1 with probability $2/3$ when $p = -1$.*

Without replicability, there is a simple 1-sample algorithm for $\{\pm 1\}$-bias estimation. First, we note the following lower bound for replicable algorithms solving this problem.

Theorem 3.6. *Any ρ-replicable algorithm for $\{\pm 1\}$-bias estimation requires $\Omega\left(\frac{1}{\rho^2}\right)$ samples.*

We defer the proof to Section A, as it is a standard modification of known replicable bias estimation lower bounds [ILPS22, HIK+24].

Proof of Theorem 3.4. We reduce realizable learning to $\{\pm 1\}$-bias estimation. Assume without loss of generality that $\mathcal{H}$ has VC Dimension $d+1$ and d is odd. As before, since $\mathcal{H}$ has VC dimension $d+1$, let $\{x_0, x_1, \ldots, x_d\}$ denote $d+1$ shattered elements. In contrast to the agnostic setting, we use a family of input distribution, parameterized by mean vectors $p \in \{\pm 1\}^{d+1}$. First, we describe the domain distribution $\mathcal{D}_{\mathcal{X}}$ which gives $\mathcal{D}_{\mathcal{X}}(x_0) = 1 - 100\alpha$ and $\mathcal{D}_{\mathcal{X}}(x_i) = \frac{100\alpha}{d}$ for $i > 0$. As before, when x_i is sampled, the label is distributed according to $\mathrm{Rad}(p_i)$. Suppose there is a pointwise replicable realizable PAC-learner A for $\mathcal{H}$ using m samples. Assume without loss of generality that $m \geqslant d$.

Given A, we describe our algorithm for ρ-replicable $\{\pm 1\}$-bias estimation. Our algorithm is analogous to Algorithm 3 except in Line 3, we sample $r \in [d]$ uniformly; in Line 4, we split $[d]\backslash r$ into random equal subsets; in Algorithm 3 we sample from $\mathcal{D}_{\mathcal{X}}$ given above instead of uniformly; in Line 10, we instead set the sample limit to $200\frac{\alpha m}{d}\sqrt{\log(1/\rho)}$. Finally, we consistently assign the label -1 to element x_0, and assume the algorithm is aware of this (i.e. without loss of generality, assume the algorithm outputs $h(x_0) = -1$).

We claim our algorithm is replicable, by again conditioning on the event that the sample limit is not reached. As before if m_r is the number of times x_r is sampled, $\mathbb{E}[m_r] = \frac{100\alpha m}{d}$ so that

$$\Pr\left(m_r > \frac{10\alpha m}{d}\left(1 + \sqrt{100\log(1/\rho)}\right)\right) < \rho,$$

so that the sample limit is not reached. Then, since A is pointwise-replicable and both samples are sampled from the same distribution (recall B_+, B_- are sampled with shared randomness), we ensure $A(S)(x_r)$ is consistent with probability $1 - \rho$, thus guaranteeing 2ρ-replicability.

Accuracy follows from a similar argument. Without loss of generality, assume we have sample access to $\mathcal{D} \sim \mathrm{Rad}(1)$. As before, conditioned on any choice of B_-, the choice of r in $B_+ \cup \{r\}$ is indistinguishable to A, so that the probability that $r \in E_+ := \{i \in B_+ \cup \{r\} \text{ s.t. } h(x_i) = -1\}$ is at most $\frac{|E_+|}{(d-1)/2}$. If we condition on A outputting an accurate hypothesis, we have $\frac{100\alpha}{d}|E_+| < \alpha$ or $|E_+| \leqslant \frac{d}{100}$. In particular, the probability that x_r is classified incorrectly is at most 0.03. Thus, we obtain a 2ρ-replicable algorithm that solves the $\{\pm 1\}$-bias estimation problem with error probability 0.03 and $O\left(\frac{m\alpha\sqrt{\log(1/\rho)}}{d}\right)$ By Theorem 3.6, this implies

$$m = \Omega\left(\frac{d}{\alpha\rho^2\sqrt{\log(1/\rho)}}\right).$$

□

4 Approximate Replicability

In this section, we give algorithms and lower bounds for approximately replicable PAC learning.

4.1 From Pointwise to Approximate

Our first step is to show that any γ-pointwise replicable learner is automatically $(\rho, \frac{\gamma}{\rho})$-approximately replicable.

Proposition 4.1. *Suppose A is a γ-pointwise replicable learner. Then, A is also $(\rho, \frac{\gamma}{\rho})$-approximately replicable.*

Proof. Suppose A is ρ-pointwise replicable and fix an arbitrary distribution $\mathcal{D}$. By γ-pointwise replicability, we have

$$\begin{aligned}
\mathbb{E}_{S,S'\sim D^n, r\sim R}\left[\Pr_{x\sim D}\left[A(S;r)(x) \neq A(S';r)(x)\right]\right] &= \mathbb{E}_{S,S'\sim D^n, r\sim R}\left[\mathbb{E}_{x\sim D}\left[\mathbb{1}\left(A(S;r)(x) \neq A(S';r)(x)\right)\right]\right] \\
&= \mathbb{E}_{x\sim D}\left[\mathbb{E}_{S,S'\sim D^n, r\sim R}\left[\mathbb{1}\left(A(S;r)(x) \neq A(S';r)(x)\right)\right]\right] \\
&= \mathbb{E}_{x\sim D}\left[\Pr_{S,S'\sim D^n, r\sim R}\left(A(S;r)(x) \neq A(S';r)(x)\right)\right] \\
&\leqslant \gamma.
\end{aligned}$$

Thus, by Markov's inequality,

$$\Pr_{S,S'\sim D^n, r\sim R}\left(\Pr_{x\sim D}\left(A(S;r)(x) \neq A(S';r)(x)\right) > \frac{\gamma}{\rho}\right) < \rho.$$

□

Combining Lemma 3.2 and Proposition 4.1, we immediately obtain an approximately replicable PAC-learner with constant success probability. In particular, we take a $\rho\gamma$-pointwise replicable learner to obtain a (ρ, γ)-approximately replicable learner.

Proposition 4.2. *Let A be an (agnostic) (α, β)-learner on $m(\alpha, \beta)$ samples. There is an (agnostic) (ρ, γ)-approximately replicable (α, β)–learner $\tilde{A}$ on $O\left(\frac{m(\alpha\beta, \alpha\beta)}{\rho^2\gamma^2}\right)$ samples.*

4.2 Boosting Correctness and Replicability

Proposition 4.2 gives a good approximately replicable PAC-learner when β is constant. We'd now like to boost it to a learner with arbitrarily small error probability paying only polylogarithmic dependence.

The key tool to do this is a new analysis of replicable hypothesis selection, a mechanism for selecting the best option in a set of candidates studied in [GKM21, HLYY25]. In particular, we will show that existing algorithms for hypothesis selection can be made replicable even under the weaker assumption that across our two 'independent' runs of the algorithm, each candidate may have slightly different rewards.

Definition 4.3 (α-Optimal Hypothesis Selection)**.** *Given hypotheses $f_1, \ldots, f_n : X \to Y$ and sample access to distribution $\mathcal{D}$ on $X \times Y$, output index i such that $\operatorname{err}_{\mathcal{D}}(f_i) \leqslant \min_j \operatorname{err}_{\mathcal{D}}(f_j) + \alpha$.*

Definition 4.4. *An algorithm A solving the α-Optimal Hypothesis Selection problem is (ρ, τ)-robustly replicable if, given sample access to distribution $\mathcal{D}$ and hypotheses $f_1, \ldots, f_n, g_1, \ldots, g_n : X \to Y$ satisfying $\mathrm{dist}_{\mathcal{D}}(f_i, g_i) < \tau$ for all $i \in [n]$,*

$$\Pr_{S,S' \sim \mathcal{D}^m, r \sim R} \left(A(S, f_1, \ldots, f_n; r) \neq A(S', g_1, \ldots, g_n; r) \right) < \rho$$

where m is the sample complexity of A.

We give a robustly replicable algorithm for hypothesis selection.

Theorem 4.5. *Let $\rho, \alpha, \beta > 0$. Let $0 < \tau \ll \frac{\rho\alpha}{\log(n/\beta)}$ for some sufficiently small constant. There is a $(\rho + \beta, \tau)$-robustly replicable algorithm solving the α-Optimal Hypothesis Selection problem with sample complexity $m = O\left(\frac{\log^2(n/\beta)}{\tau^2}\right)$.*

We defer the proof to Section 4.3 and show how to use Theorem 4.5 to reduce the error β.

Proposition 4.6. *Let $\gamma \ll \frac{\rho\alpha}{\log(1/\beta)}$ for some sufficiently small constant. Let A be a (ρ, γ)-approximately replicable agnostic $(\alpha, 0.01)$-learner on $m(\alpha, 0.01, \rho, \gamma)$ samples.*

There is a (ρ, γ)-approximately replicable agnostic (α, β)-learner with sample complexity

$$O\left(m\left(\alpha, 0.01, \frac{\rho}{\log(1/\beta)}, \gamma\right)\log(1/\beta) + \frac{\log^2(1/\beta)}{\gamma^2}\right).$$

Combined with Proposition 4.2 and algorithms for agnostic PAC learning, we obtain Theorem 1.6 as an immediate corollary.

Theorem 1.6. *Let A be an (agnostic) (α, β)-learner on $m(\alpha, \beta)$ samples. There exists an (agnostic) (ρ, γ)-approximately replicable (α, β)-learner with sample complexity $O\left(m(\alpha, \alpha)\left(\frac{\log^3(1/\beta)}{\rho^2\gamma^2} + \frac{\log^4(1/\beta)}{\rho^2\alpha^2}\right)\right) = \tilde{O}\left(\frac{d\log^3(1/\beta)}{\rho^2\gamma^2\alpha^2} + \frac{d\log^4(1/\beta)}{\rho^2\alpha^4}\right)$. Furthermore, our algorithm runs in time linear in sample complexity with $O\left(\frac{\log^3(1/\beta)}{\rho^2\gamma^2} + \frac{\log^4(1/\beta)}{\rho^2\alpha^2}\right)$ oracle calls to A.*

Proof. Consider Algorithm 4.

Algorithm 4: BOOSTERRORAPPROX$(A, \alpha, \beta, \rho, \gamma)$

Input : (ρ, γ)-approximately replicable $(\alpha, 0.01)$-accurate agnostic PAC leaner.
Parameters : β error probability.
Output : (ρ, γ)-approximately replicable (α, β)-accurate agnostic PAC leaner.

Fix $D \leftarrow 200\log(1/\beta)$ and $\gamma \ll \frac{\rho\alpha}{\log(D/\beta)}$ for some sufficiently small constant.
for $i \in [D]$ **do**
  Compute $h_i \leftarrow A(S)$ on a fresh dataset of $m\left(\frac{\alpha}{2}, 0.01, \frac{\rho}{2D}, \gamma\right)$ samples.
return HYPOTHESISSELECTION $\left(\{h_i\}_{i=1}^D, \frac{\alpha}{2}, \frac{\beta}{2}, \frac{\rho}{2}, \gamma\right)$.

Note that the sample complexity is immediate.

We will separately argue the correctness and replicability of the algorithm. We begin with correctness. Condition on the following events:

1. At least one index h_i is an $\alpha/2$-accurate hypothesis.

2. The execution of **HypothesisSelection** (Theorem 4.5) chooses an $\alpha/2$-optimal hypothesis.

We bound the probability that either event fails. The second event fails with probability at most $\beta/2$. Suppose we run A independently D times on fresh samples. This produces D hypotheses $h_1, \ldots, h_D$. For each $i \in [D]$, h_i is $\alpha/2$-accurate (i.e. has excess error at most $\alpha/2$) with probability 0.99. Thus, the probability no hypothesis is $\alpha/2$-accurate is at most $(1-0.99)^D < e^{-0.99D} < \beta^2$. A union bound ensures that either event fails with probability at most β. Now, since Theorem 4.5 chooses an $\alpha/2$-optimal hypothesis, we return a hypothesis h_i such that

$$\mathrm{err}_{\mathcal{D}}(h_i) \leqslant \min_{j=1}^{n} \mathrm{err}_{\mathcal{D}}(h_j) + \frac{\alpha}{2} \leqslant \mathrm{OPT} + \alpha.$$

We now argue replicability. For each $i \in [D]$, let h_i, h_i' denote the hypotheses output by A on two separate runs of our algorithm. We condition on the event that $\mathrm{dist}_{\mathcal{D}}(h_i, h_i') \leqslant \gamma$ for all $i \in [D]$. Since each invocation of A is $(\rho/2D, \gamma)$-replicable, this event holds with probability at least $1 - \rho/2$. Then, since we have fixed $\gamma \ll \frac{\rho\alpha}{\log(D/\beta)}$ for some sufficiently small constant, Theorem 4.5 selects the same index across both runs as the algorithm is $(\rho/2, \gamma)$-robustly replicable. In particular, since the same index $\hat{i}$ is returned across both runs with probability $1 - \rho$, we have the the output hypothesis disagree on at most a γ-fraction of the distribution $\mathcal{D}$, as desired. □

Boosting Replicability Above, we obtain an algorithm that incurs sub-optimal dependence on α even when the replicability parameters ρ, γ are constant. Below, we give an alternative approach that achieves better dependency on α at the cost of polynomial factors in γ. Recall that a γ-pointwise-replicable algorithm automatically implies a $(0.01, \gamma)$-approximately replicable algorithm. In particular, Theorem 3.1 gives a $(0.01, \gamma)$-approximately replicable (α, β)-learner with $\tilde{O}\left(\frac{d}{\gamma^4\alpha^2}\right)$ samples. We show how to boost the replicability parameter ρ and obtain the following theorem.

Theorem 1.7. *Let A be an (agnostic) (α, β)-learner on $m(\alpha, \beta)$ samples. There exists an (agnostic) (ρ, γ)-approximately replicable (α, β)-learner with sample complexity $\tilde{O}\left(\left(\frac{d\log(1/\beta)}{\gamma^4\alpha^2}\right)\left(\frac{1}{\rho^2} + \log(1/\beta)\right)\right)$. In the realizable case, there is a proper (ρ, γ)-approximately replicable (α, β)-learner with sample complexity $O\left(\frac{d+\log(1/\min(\rho,\beta))}{\rho^2\min(\alpha,\gamma)}\right)$.*

We do this in two steps. First, we give a replicable procedure for deciding when a particular random string is approximately replicable for a given input distribution. Then, using this procedure, we select an (approximately) replicable random string and one of the γ-clustered hypotheses generated by the string.

Toward this end, we first formalize the notion of an approximately replicable random string. We first need the notion of a hypothesis cluster.

Definition 4.7 (Hypothesis Cluster)**.** *Let $\mathcal{P}$ be a distribution over hypotheses $f : X \to \{0, 1\}$. $\mathcal{P}$ has a (v, γ)-cluster with respect to $\mathcal{D}$ if $\Pr_{f,g\sim\mathcal{P}}(\mathrm{dist}_{\mathcal{D}}(f, g) < \gamma) > v$. When the distribution $\mathcal{D}$ is clear, we say $\mathcal{P}$ has a (v, γ)-cluster.*

Definition 4.8 (Approximately Replicable Random String)**.** *A random string r is (v, γ)-replicable for A with respect to $\mathcal{D}$ if*

$$\Pr_{S,S'}(\mathrm{dist}_{\mathcal{D}}(A(S; r), A(S'; r) < \gamma) > v.$$

When the algorithm A and the distribution $\mathcal{D}$ is clear, we just say r is (v, γ)-replicable.

Equivalently, r is (v, γ)-replicable if the distribution $A(\cdot; r)$ has a (v, γ)-cluster.

Our goal is now to obtain a replicable algorithm for deciding when r is replicable.

Theorem 4.9. *Let $v, \varepsilon, \gamma, \beta > 0$. There is a ρ-replicable algorithm that given sample access to $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$ and $\mathcal{P}$ over hypotheses satisfies:*

1. *(Completeness) If $\mathcal{P}$ has a $(v + 2\varepsilon, \gamma/2)$-cluster, accept (output 1) with probability $1 - \beta$.*
2. *(Soundness) If $\mathcal{P}$ has no $(v - 2\varepsilon, 2\gamma)$-cluster, reject (output 0) with probability $1 - \beta$.*

The algorithm requires $m_1 := O\left(\frac{1}{\rho^2\varepsilon^2} + \frac{\log(1/\beta)}{\varepsilon^2}\right)$ samples from $\mathcal{P}$ and $O\left(\frac{m_1 \log(m_1/\beta)}{\gamma^2}\right)$ samples from $\mathcal{D}$.

We defer the proof of Theorem 4.9 to Section 4.3. Now, given that we have replicably selected a replicable r, we know that $A(\cdot; r)$ has a $(v - \varepsilon, 2\gamma)$-cluster and would like to return such a hypothesis.

Definition 4.10 (Hypothesis Cluster Detection)**.** *Given a distribution $\mathcal{P}$ over hypotheses $h : \mathcal{X} \to \{0, 1\}$, distribution $\mathcal{D}$ over $\mathcal{X}$, and $v, \gamma, \varepsilon > 0$ output a set of hypotheses F satisfying:*

1. *(Completeness) If there exists some f' such that $\Pr_{g\sim\mathcal{P}}(\mathrm{dist}_{\mathcal{D}}(f', g) < \gamma) > v + \varepsilon$ then there is some $f \in F$ such that $\mathrm{dist}_{\mathcal{D}}(f, f') < 7\gamma$,*
2. *(Soundness) For each $f \in F$, $\Pr_{g\sim\mathcal{P}}(\mathrm{dist}_{\mathcal{D}}(f, g) < 3\gamma) > v - \varepsilon$.*
3. *(Separability) For every distinct $f_1, f_2 \in F$, $\mathrm{dist}_{\mathcal{D}}(f_1, f_2) \geqslant \gamma$.*

Lemma 4.11. *There is an algorithm solving Hypothesis Cluster Detection with $O\left(\frac{\log(1/\beta)}{\varepsilon^2}\right)$ samples from $\mathcal{P}$ and $O\left(\frac{\log(1/\varepsilon\beta)}{\gamma^2}\right)$ samples from $\mathcal{D}$ with probability $1 - \beta$.*

Proof. Consider Algorithm 5.

Algorithm 5: ClusterDetection$(\mathcal{A}, r, \gamma, v, \varepsilon, \beta)$

Input : Approximately Replicable Agnostic PAC-learner $\mathcal{A}$, (v, γ)-replicable random string r.
Parameters : γ distance, v frequency, ε frequency error, and β error probability
Output : Set F of hypotheses $\mathcal{X} \to \{0, 1\}$.

Sample $n = O\left(\frac{\log(1/\beta)}{\varepsilon^2}\right)$ hypotheses from $\mathcal{P} \sim A_{\cdot;r}$, denoted $f_1, \ldots, f_n$.
Draw $m = O\left(\frac{\log(n/\beta)}{\gamma^2}\right)$ samples $\mathcal{D}$, denoted $T = (x_1, \ldots, x_m)$.
For all $i, j \in [n]$, compute $\mathrm{dist}_T(f_i, f_j) = \frac{1}{m}\sum_{k=1}^{m} \mathbb{1}[f_i(x_k) \neq f_j(x_k)]$.
Let $F \leftarrow \varnothing$ and $F_{\text{cand}} \leftarrow \{f_1, \ldots, f_n\}$.
for $i \in [n]$ **do**
  Let $N(i) \leftarrow \{f_j \in F_{\text{cand}} \text{ s.t. } \mathrm{dist}_T(f_i, f_j) < 2\gamma\}$
  if $\frac{|N(i)|}{n} \geqslant v$ **then**
    Update $F \leftarrow F \cup \{f_i\}$ and $F_{\text{cand}} \leftarrow F_{\text{cand}} \backslash N(i)$.
return F.

We claim the algorithm satisfies the desired properties. Note that for any f, g, $\mathbb{E}_T\left[\mathrm{dist}_T(f, g)\right] = \mathrm{dist}_{\mathcal{D}}(f, g)$ and by Chernoff bounds we have for all i, j,

$$\Pr\left(|\mathrm{dist}_S(f_i, f_j) - \mathrm{dist}_{\mathcal{D}}(f_i, f_j)| > \gamma\right) \leqslant \frac{\beta}{2n^2}.$$

By a union bound, any of these events occur with probability at most $\frac{\beta}{2}$. Thus, we can condition on the event that $|\text{dist}_S(f_i, f_j) - \text{dist}_D(f_i, f_j)| \leqslant \gamma$ for all i, j. We now prove the correctness of our algorithm. The third condition is satisfied by construction (since $\text{dist}_{\mathcal{D}}(f_i, f_j) < \gamma$ implies $\text{dist}_T(f_i, f_j) < 2\gamma$). For all i, let $F^{(i)}_{\text{cand}}$ denote the set F_{cand} at the start of the i-th iteration.

We start with the remaining conditions. Let $N_t(f') := \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}\left[\text{dist}_{\mathcal{D}}(f_i, f') < t\right]$. By a Chernoff bound, we have $|N_\gamma(f') - \Pr_{g\sim\mathcal{P}}(\text{dist}_{\mathcal{D}}(f', g) < \gamma)| < \varepsilon$ with probability $1 - \frac{\beta}{4}$. Similarly, we condition on $|N_{3\gamma}(f') - \Pr_{g\sim\mathcal{P}}(\text{dist}_{\mathcal{D}}(f', g) < 3\gamma)| < \varepsilon$. All events hold with probability $1 - \beta$.

For the soundness condition, let $f \in F$ so that there are at least vn functions f_i in $F^{(i)}_{\text{cand}} \subseteq \{f_1, \ldots, f_n\}$ such that $\text{dist}_T(f_i, f) < 2\gamma$ and therefore $\text{dist}_{\mathcal{D}}(f_i, f_j) < 3\gamma$. Then, we have

$$\Pr_{g\sim\mathcal{P}}(\text{dist}_{\mathcal{D}}(f', g) < 3\gamma) > N_{3\gamma}(f') - \varepsilon \geqslant v - \varepsilon.$$

For the completeness condition, suppose there exists f' satisfying $\Pr_{g\sim\mathcal{P}}(\text{dist}_{\mathcal{D}}(f', g) < \gamma) > v + \varepsilon$. Then, we have $N_\gamma(f') \geqslant v$, i.e. there are at least vn-functions in $f_1, \ldots, f_n$ satisfying $\text{dist}_{\mathcal{D}}(f', f_i) < \gamma$ and therefore $\text{dist}_T(f', f_i) < 2\gamma$. We now consider two cases. If there are vn functions in $F^{(i)}_{\text{cand}}$ satisfying $\text{dist}_T(f', f_i) < 2\gamma$, we include $f_i \in F$. Otherwise, there exists some $i' < i$ such that $|N(i')| \geqslant vn$ with $f_{i'} \in F$, and there exists some function f_j in $N(i')$ such that $\text{dist}_T(f_i, f_j) < 2\gamma$. By the triangle inequality, we have $\text{dist}_T(f_i, f_{i'}) < \text{dist}_T(f_i, f_j) + \text{dist}_T(f_j, f_{i'}) < 4\gamma$. Thus

$$\text{dist}_D(f', f_{i'}) \leqslant \text{dist}_T(f', f_{i'}) + \gamma \leqslant \text{dist}_T(f', f_i) + \text{dist}_T(f_i, f_{i'}) + \gamma < 7\gamma.$$

□

Finally, we describe how to boost the replicability parameter of an approximately replicable PAC learner using Theorem 4.9 and Lemma 4.11.

Proposition 4.12. *Let $\rho, \gamma, \alpha, \beta > 0$. Suppose A is a $(0.01, \gamma)$-approximately replicable (agnostic) (α, β)-learner on $m(\alpha, \beta, 0.01, \gamma)$ samples. There exists a (ρ, γ)-approximately replicable (agnostic) (α, β)-learner on with sample complexity*

$$O\left(m(\alpha, \beta, 0.01, \gamma)\left(\frac{1}{\rho^2} + \log(1/\beta)\right) + \frac{\log(1/\beta)}{\gamma^2} + \frac{\log(1/\beta)\log^3(1/\rho)}{\rho^2}\right).$$

Proof. We present Algorithm 6.

To analyze our algorithm, we condition on the following events.

1. At least one random string r_i is $\left(0.9, \frac{\gamma}{12}\right)$-replicable over $\mathcal{D}$.
2. Every execution of **ReplicableStableTester** is correct.
3. Every execution of **ReplicableStableTester** is replicable.
4. Every execution of **ClusterDetection** is correct.
5. Every execution of A produces an α-accurate hypothesis.

First, we argue that the above events happen with high probability. Since A is $\left(0.01, \frac{\gamma}{12}\right)$-replicable, we have that

$$\mathbb{E}_r\left[\mathbb{E}_{S_1, S_2\sim\mathcal{D}^{n_3}}\left[\mathbb{1}\left(\text{dist}_{\mathcal{D}}(A(S_1; r), A(S_2, r)) > \frac{\gamma}{12}\right)\right]\right] < 0.01.$$

By Markov's inequality, a randomly chosen string is $(0.9, \gamma)$-replicable with probability at least 0.9. Thus, if we choose R random strings independently, none are $(0.9, \gamma)$-replicable with probability at most $0.1^R < \rho/2$

Algorithm 6: BOOSTREPLICABILITY$(A, \alpha, \beta, \rho, \gamma)$

Input : $(0.01, \gamma)$-approximately replicable (α, β)-accurate agnostic PAC leaner with sample complexity $m(\alpha, \beta, 0.01, \gamma)$.

Parameters : ρ replicability parameter.

Output : (ρ, γ)-approximately replicable (α, β)-accurate agnostic PAC leaner.

Let A be $\left(0.01, \frac{\gamma}{12}\right)$-approximately replicable and (α, β_1)-accurate where $\beta_1 \ll \frac{\beta\rho^2}{R} + \frac{\beta}{R\log(R/\beta)}$ for some sufficiently small constant.

Fix $R \leftarrow O(\log(1/\rho))$ and randomly sample R random strings $r_1, \ldots, r_R$.

for $i \in [R]$ **do**

  if REPLICABLESTABLETESTER $\left(A, r_i, 0.8, \frac{\gamma}{6}, 0.01, \frac{\beta}{3R}, \frac{\rho}{2R}\right)$ **then**

    return CLUSTERDETECTION $\left(A, r_i, \frac{\gamma}{3}, 0.75, 0.01, \frac{\beta}{3R}\right)$

return $h \leftarrow A(S)$ on some fresh dataset $S \sim \mathcal{D}^{n_3}$.

for $R = O(\log(1/\rho))$. Note that REPLICABLESTABLETESTER requires $O\left(\frac{1}{\rho^2} + \log(R/\beta)\right)$ calls to A and CLUSTERDETECTION requires $O(\log(R/\beta))$ calls to A. Thus, by a union bound we can ensure that every output hypothesis is α-accurate. We can thus conclude events 2, 4, and 5 hold with probability $1 - \beta$ (for correctness) and events 1, 3 hold with probability at least $1 - \rho$. Note that correctness of the algorithm follows immediately from the last event.

We now argue that our procedure is (ρ, γ)-replicable. Since at least one random string r_i is $\left(0.9, \frac{\gamma}{24}\right)$-replicable, this random string r_i has a $\left(0.9, \frac{\gamma}{12}\right)$-cluster. Since each execution of REPLICABLESTABLETESTER is correct, at least one index i must accept by the completeness condition (since $v + \varepsilon < 0.81$ and $\frac{\gamma}{24} = \frac{\gamma}{12 \cdot 2}$). By the soundness condition, the first index that accepts satisfies that r_i has a $\left(0.79, \frac{\gamma}{6}\right)$-cluster. Then, our call to CLUSTERDETECTION returns (via the completeness condition) a function f such that the soundness condition guarantees f has a $\left(0.74, \frac{\gamma}{2}\right)$-cluster. Furthermore, by the replicability of REPLICABLESTABLETESTER (condition 3), we observe that REPLICABLESTABLETESTER accepts for the first time on the same index i across two runs of the algorithm. Thus, both runs select the same random string r_i and there exist two hypotheses h_1, h_2 such that

$$\Pr_{S \sim \mathcal{D}^m}\left(\mathrm{dist}_{\mathcal{D}}(h_1, A(S; r)) < \frac{\gamma}{2}\right), \Pr_{S \sim \mathcal{D}^m}\left(\mathrm{dist}_{\mathcal{D}}(h_2, A(S; r)) < \frac{\gamma}{2}\right) > 0.74$$

so that there exists some S where $\mathrm{dist}_{\mathcal{D}}(A(S; r), h_1), \mathrm{dist}_{\mathcal{D}}(A(S; r), h_2) < \frac{\gamma}{2}$. By the triangle inequality, $\mathrm{dist}_{\mathcal{D}}(h_1, h_2) < \gamma$ as desired.

To bound the sample complexity, we compute

$$O\left(m(\alpha, \beta, 0.01, \gamma)\left(\frac{1}{\rho^2} + \log(1/\beta)\right) + \frac{\log(1/\beta)}{\gamma^2} + \frac{\log(1/\beta)\log^3(1/\rho)}{\rho^2}\right)$$

where the first term comes from calls to $\mathcal{P}$ in REPLICABLESTABLETESTER and CLUSTERDETECTION, while the second term comes from samples to $\mathcal{D}$ in CLUSTERDETECTION and the final term from samples to $\mathcal{D}$ in REPLICABLESTABLETESTER. □

Finally, we are ready to prove Theorem 1.7 in the agnostic case. The realizable case is given in Theorem 4.15.

Proof of Theorem 1.7. The result follows from combining Theorem 1.4, Proposition 4.1, and Proposition 4.12.

First, we begin with a $(0.01, \gamma)$-approximately replicable (α, β)-learner with sample complexity

$$O\left(\frac{m(\alpha\gamma, \alpha\gamma)\log(1/\beta)}{\gamma^2} + \frac{\log(1/\beta)\log\log(1/\beta)}{\alpha^2}\right) = O\left(\frac{(d + \log(1/\alpha\gamma))\log(1/\beta)}{\gamma^4\alpha^2}\right).$$

Then, via Proposition 4.12, we obtain a (ρ, γ)-approximately replicable (α, β)-learner with sample complexity

$$O\left(\left(\frac{m(\alpha\gamma, \alpha\gamma)\log(1/\beta)}{\gamma^2} + \frac{\log(1/\beta)\log\log(1/\beta)}{\alpha^2}\right)\left(\frac{1}{\rho^2} + \log(1/\beta)\right) + \frac{\log(1/\beta)\log^3(1/\rho)}{\rho^2}\right)$$
$$= \tilde{O}\left(\frac{d\log(1/\beta) + \log^2(1/\beta)}{\rho^2\gamma^4\alpha^2}\right).$$

□

4.3 Hypothesis Selection and Clustering

We prove Theorem 4.5. Our algorithm closely follows [GKM21, HLYY25], using correlated sampling to sample from a distribution over the hypotheses given by the so-called exponential mechanism from privacy [MT07].

Proof. Consider the algorithm given by Algorithm 7.

Algorithm 7: HypothesisSelection$(\{f_i\}_{i=1}^n, \alpha, \beta, \rho, \tau)$

Input : Hypotheses $\{f_i\}_{i=1}^n$ and sample access to $\mathcal{D}$.
Parameters : α accuracy, β error probability, and (ρ, τ) robust replicability
Output : α-optimal hypothesis f_i.

1 Set $t \leftarrow \frac{2\log(2n/\beta)}{\alpha}$.
2 Set $m \leftarrow \frac{\Theta(1)}{\tau^2}\log\left(\frac{n}{\beta}\right)$. Draw samples $S \sim \mathcal{D}^m$ and denote $S \leftarrow ((x_i, y_i))_{i=1}^m$.
3 For each $i \in [n]$, empirically estimate $\mathrm{err}_S(f_i) \leftarrow \frac{1}{m}\sum_{\ell=1}^m \mathbb{1}(f_i(x_\ell) \neq y_\ell)$.
4 Define the distribution $\hat{\mathcal{D}}$ on $[n]$ using the exponential mechanism, i.e. i is drawn with probability proportional to $\exp(-t \cdot \mathrm{err}_S(f_i)))$.
5 **return** CorrSamp$(\hat{\mathcal{D}}; r)$ with shared internal randomness r.

We separately argue the correctness and robust replicability of the algorithm. Condition on the event

$$|\mathrm{err}_S(f_i) - \mathrm{err}_{\mathcal{D}}(f_i)| \ll \frac{\tau}{\log(n/\beta)}$$

for all i, which happens with probability at least $1 - \frac{\beta}{2}$ by a Chernoff and union bound.

We begin by arguing correctness. Denote by $i^* \in [n]$ the index of an optimal hypothesis i.e., $i^* = \arg\min_i \mathrm{err}_{\mathcal{D}}(f_i)$. Then, the probability that a hypothesis f_i is sampled is at most $\exp(-t(\mathrm{err}_S(f_i) - \mathrm{err}_S(f_{i^*}))$. If $\mathrm{err}_{\mathcal{D}}(f_i) > \mathrm{err}_{\mathcal{D}}(f_{i^*}) + \alpha$ then by the above conditioning, $\mathrm{err}_S(f_i) \geqslant \mathrm{err}_{\mathcal{D}}(f_i) - \frac{\tau}{4} > \mathrm{err}_{\mathcal{D}}(f_{i^*}) + \alpha - \frac{\tau}{4} \geqslant \mathrm{err}_S(f_{i^*}) + \alpha - \frac{\tau}{2}$. Thus, by our assumption $\tau < \alpha$, $\mathrm{err}_S(f_i) \geqslant \mathrm{err}_S(f_{i^*}) + \frac{\alpha}{2}$. Finally the probability any such i is output is at most

$$\exp\left(-t\alpha/2\right) \leqslant \frac{\beta}{2n}$$

so that by a union bound, the probability that any hypothesis with $\mathrm{err}_{\mathcal{D}}(f_i) > \mathrm{err}_{\mathcal{D}}(f_{i^*}) + \alpha$ is chosen is at most $\frac{\beta}{2}$. Combined with the conditioned event, the probability of an error is at most β.

Next, we analyze replicability. Let S_1, S_2 denote the samples observed by two independent runs of the algorithm A and r the shared internal randomness. Let $P(i) = \Pr(A(S_1; r) = i)$ and $Q(i) = \Pr(A(S_2; r) = i)$. Then by the definition of the exponential mechanism

$$P(i) = \frac{\exp(-t \cdot \mathrm{err}_{S_1}(f_i))}{\sum_j \exp(-t \cdot \mathrm{err}_{S_1}(f_j))}$$
$$Q(i) = \frac{\exp(-t \cdot \mathrm{err}_{S_2}(g_i))}{\sum_j \exp(-t \cdot \mathrm{err}_{S_2}(g_j))}$$

We bound the total variation distance between P and Q. Fix some $i \in [n]$. Then,

$$\begin{aligned} P(i) - Q(i) &= \frac{\exp(-t \cdot \mathrm{err}_{S_1}(f_i))}{\sum_j \exp(-t \cdot \mathrm{err}_{S_1}(f_j))} - \frac{\exp(-t \cdot \mathrm{err}_{S_2}(g_i))}{\sum_j \exp(-t \cdot \mathrm{err}_{S_2}(g_j))} \\ &= \left(\frac{\exp(-t \cdot \mathrm{err}_{S_2}(g_i))}{\sum_j \exp(-t \cdot \mathrm{err}_{S_2}(g_j))} \left(\frac{\exp(-t \cdot \mathrm{err}_{S_1}(f_i))}{\exp(-t \cdot \mathrm{err}_{S_2}(g_i))} \frac{\sum_j \exp(-t \cdot \mathrm{err}_{S_2}(g_j))}{\sum_j \exp(-t \cdot \mathrm{err}_{S_1}(f_j))} - 1 \right) \right). \end{aligned}$$

Combining our conditioned event and the triangle inequality, we bound

$$\begin{aligned} |\mathrm{err}_{S_1}(f_i) - \mathrm{err}_{S_2}(g_i)| &\leqslant |\mathrm{err}_{S_1}(f_i) - \mathrm{err}_{\mathcal{D}}(f_i)| + |\mathrm{err}_{\mathcal{D}}(f_i) - \mathrm{err}_{\mathcal{D}}(g_i)| + |\mathrm{err}_{\mathcal{D}}(g_i) - \mathrm{err}_{S_2}(g_i)| \\ &\leqslant 2\tau + |\mathrm{err}_{\mathcal{D}}(f_i) - \mathrm{err}_{\mathcal{D}}(g_i)| \\ &\leqslant 2\tau + \tau \\ &\leqslant 3\tau. \end{aligned}$$

In particular, the total variation distance can be bounded by

$$\frac{1}{2} \sum_i |P(i) - Q(i)| = \frac{1}{2} \sum_i Q(i) \left(e^{4t(3\tau)} - 1 \right) = O(t\tau) = O\left(\frac{\log(n/\beta)\tau}{\alpha} \right)$$

where we use that $e^x < 1 + O(x)$ for small $x > 0$. Since we have assumed that $\tau \ll \rho\alpha/\log(n/\beta)$, the total variation distance is bounded by ρ. In particular, by Correlated Sampling (see Lemma 2.5) we can ensure that the outputs of the algorithms differ by at most probability $O(\rho)$. Combined with a union bound on the conditioned events, the algorithms fail to be replicable with probability at most $O(\rho + \beta)$. □

We now prove Theorem 4.9.

Proof. Consider Algorithm 8.

The claimed sample complexity follows exactly from the bounds on m_1 and m_2.

First, we claim that our algorithm is replicable. Note that each $\mathrm{dist}_{T_i}(f_i, g_i)$ is an independent sample from $X \sim \mathrm{dist}_T(f, g)$ where $f, g \sim \mathcal{P}$ and $T \sim \mathcal{D}^{m_2}$. In particular, each $\mathbb{1}[\mathrm{dist}_{T_i}(f_i, g_i) < \gamma]$ is an independent flip of a coin with bias $\Pr(X < \gamma)$. Let $\hat{v}, \hat{v}'$ be the empirical estimates of $\Pr(X < \gamma)$ computed in two runs. Our algorithm is replicable whenever v' does not lie between $\hat{v}, \hat{v}'$. Let $v^* := \Pr(X < \gamma)$. Then,

Algorithm 8: REPLICABLESTABLETESTER$(\mathcal{A}, r, v, \gamma, \varepsilon, \beta, \rho)$

Input : Approximately Replicable Agnostic PAC-learner A, random string r.
Parameters : ε accuracy, β error, and ρ replicability
Output : Decide if r is (v, γ)-stable.

1 Sample $m_1 = O\left(\frac{1}{\rho^2\varepsilon^2} + \frac{\log(1/\beta)}{\varepsilon^2}\right)$ hypotheses from $\mathcal{P} \sim A(\cdot; r)$, denoted $f_1, g_1, \ldots, f_{m_1}, g_{m_1}$.
2 For all $i \in [m_1]$, draw $m_2 = O\left(\frac{\log(m_1/\beta)}{\gamma^2}\right)$ samples from $\mathcal{D}$, denoted $T_i = (x_{i,1}, \ldots, x_{i,m_2})$.
3 For all $i \in [m_1]$, compute $\mathrm{dist}_{T_i}(f_i, g_i) = \frac{1}{m_2}\sum_{k=1}^{m_2} \mathbb{1}[f_i(x_{i,k}) \neq g_i(x_{i,k})]$.
4 Let $\hat{v} := \frac{1}{m_1}\sum_{i=1}^{m_1} \mathbb{1}[\mathrm{dist}_{T_i}(f_i, g_i) < \gamma]$.
5 Draw a random threshold $v' \in (v - \varepsilon, v + \varepsilon)$.
6 Accept (output 1) if $\hat{v} > v'$, Reject (output 0) otherwise.

we may compute

$$\begin{aligned}
\Pr_{v',\hat{v},\hat{v}'}\left(v' \in (\hat{v}, \hat{v}')\right) &= \mathbb{E}_{v',\hat{v},\hat{v}'}\left[\mathbb{1}\left[v' \in (\hat{v}, \hat{v}')\right]\right] \\
&= \mathbb{E}_{\hat{v},\hat{v}'}\left[\mathbb{E}_{v'}\left[\mathbb{1}\left[v' \in (\hat{v}, \hat{v}')\right]|\hat{v}, \hat{v}'\right]\right] \\
&= \mathbb{E}_{\hat{v},\hat{v}'}\left[\frac{|\hat{v} - \hat{v}'|}{2\varepsilon}\right] \\
&\leqslant \frac{1}{\varepsilon}\mathbb{E}_{\hat{v}}\left[|\hat{v} - v^*|\right] \\
&\leqslant \frac{1}{2\varepsilon}\sqrt{\mathbb{E}_{\hat{v}}\left[(\hat{v} - v^*)^2\right]} \\
&= O\left(\frac{1}{\varepsilon\sqrt{m_1}}\right) \leqslant \rho.
\end{aligned}$$

In the first equality we rewrite probability as expectation of indicator. In the second, we apply the total probability rule. In the third, we note that the probability v' lies between $\hat{v}, \hat{v}'$ is proportional to the length of the interval between the two estimates (normalized by $\frac{1}{2\varepsilon}$). In the first inequality, we apply the triangle inequality since $|\hat{v} - \hat{v}'| \leqslant |\hat{v} - v^*| + |v^* - \hat{v}'|$, and in the second, Jensen's inequality. Finally, we conclude with a standard concentration bound (on the estimate of a coin's bias from m_2 independent samples) and our definition of m_1. Thus, our algorithm is ρ-replicable.

We now argue correctness. We condition on the event that for all i,

$$|\mathrm{dist}_{\mathcal{D}}(f_i, g_i) - \mathrm{dist}_{T_i}(f_i, g_i)| < \frac{\gamma}{2}$$

which holds with probability $1 - \beta/2$ following a standard Chernoff bound and our definition of m_2. Similarly, let $v^* := \Pr_{f,g\sim\mathcal{P}}(\mathrm{dist}_{\mathcal{D}}(f, g) > 2\gamma)$ and $\hat{v}^* := \frac{1}{m_1}\sum_{i=1}^{m_1} \mathbb{1}\left[\mathrm{dist}_{\mathcal{D}}(f_i, g_i) > 2\gamma\right]$. Thus, with probability $1 - \beta/2$ we have $|\hat{v}^* - v^*| \leqslant \varepsilon$.

Now, suppose $\mathcal{P}$ has a $(v + 2\varepsilon, \gamma/2)$-cluster. By our condition events, we have that $\hat{v} \geqslant v + \varepsilon > v'$ and the algorithm must accept. Otherwise, suppose $\mathcal{P}$ has no $(v - 2\varepsilon, 2\gamma)$-cluster. Again we have $\hat{v} \leqslant v - \varepsilon < v'$ and the algorithm rejects. □

4.4 Approximately Replicable Learning for Thresholds

In this section, we give a proper approximately replicable learning algorithm for thresholds with near-optimal sample complexity by combining our robust hypothesis selection method from the previous section with a basic high probability quantile estimation.

Definition 4.13 (Thresholds)**.** *The class of thresholds consist of all functions* $\mathcal{H} := \{f_t : x \mapsto \mathbb{1}[x > t] \mid t \in \mathbb{R}\}$.

Our learner satisfies the following formal guarantees.

Proposition 1.8. *Let* $\rho, \gamma, \alpha > 0$ *and* $0 < \beta < \rho$*. There is a proper* (ρ, γ)*-approximately replicable* (α, β)*-learner for thresholds with sample complexity* $O\left(\frac{\log(1/\beta)}{\gamma^2} + \frac{\log^3(1/\alpha\beta)}{\rho^2\alpha^2}\right)$.

In Section 4.6 we show this algorithm is optimal (up to log factors) when $\rho = \Theta(\gamma)$.

Our algorithm proceeds in two stages: first we estimate quantiles of the marginal distribution $\mathcal{D}_{\mathcal{X}}$. Then, given that the quantiles give a sufficiently granular splitting of the marginal distribution (and they are learned up to sufficient accuracy), we use robustly replicable hypothesis selection to select an α-optimal quantile. In slightly more detail, if we learn the marginal $\mathcal{D}_{\mathcal{X}}$'s α-quantiles up to $\gamma \ll \rho\alpha$ accuracy, we are guaranteed (1) that at least one quantile is α-optimal, and (2) every quantile is γ-close in classification distance across multiple runs of the algorithm. Therefore, robustly replicable hypothesis selection (Theorem 4.5) replicably outputs an $O(\alpha)$-optimal hypothesis.

Proof. We now formally describe our algorithm.

Algorithm 9: THRESHOLDLEARNER$(\alpha, \beta, \rho, \gamma)$

Input : Sample access to distribution $\mathcal{D}$ over $\mathbb{R} \times \{0, 1\}$.
Parameters : (ρ, γ) replicability parameters and (α, β)-accuracy parameters.
Output : (ρ, γ)-approximately replicable (α, β)-accurate agnostic PAC leaner.

Fix $K \leftarrow \frac{3}{\alpha}$ and $\tau \ll \min\left(\gamma, \frac{\rho\alpha}{\log(K/\beta)}\right)$ for some sufficiently small constant.
Draw $m \leftarrow O\left(\frac{\log(1/\beta)}{\tau^2}\right)$ samples from $\mathcal{D}$, denoted $S = (x_1, \ldots, x_m)$.
Sort S (with ties broken arbitrarily).
for $0 \leqslant i \leqslant K$ **do**
    Set $\hat{t}_i \leftarrow x_{\frac{im}{K}}$ and $h_i : x \mapsto \mathbb{1}[x \geqslant t_i]$.
Set $h_{-1} : x \mapsto 0$ and $h_{K+1} : x \mapsto 1$.
return HYPOTHESISSELECTION$\left(\{h_i\}_{i=-1}^{K+1}, \frac{\alpha}{2}, \frac{\beta}{2}, \frac{\rho}{3}, \tau\right)$.

Let $\tau, K > 0$ to be parameters to be fixed later. We will assume that K is a multiple of $\frac{1}{\alpha}$. For simplicity, we may assume m is a multiple of $\frac{1}{\alpha}$ (since we will have $m > \frac{1}{\alpha}$, this comes at most a constant factor increase in sample complexity). Let $F(x)$ denote the cumulative distribution function of the marginal distribution $\mathcal{D}_{\mathcal{X}}$, i.e. $F(x) := \Pr_{X \sim \mathcal{D}_{\mathcal{X}}}(X \leqslant x)$. For all $0 \leqslant i \leqslant K$, let $t_i := F(i/K)$ denote the i/K-th quantile of $\mathcal{D}_{\mathcal{X}}$. Since $\hat{t}_i$ are the empirical i/K-th quantiles of our sample S drawn i.i.d. from $\mathcal{D}_{\mathcal{X}}$, we use the Dvoretsky-Kiefer-Wolfowitz (DKW) Inequality to bound the accuracy of our estimates.

Theorem 4.14 (Dvoretsky-Kiefer-Wolfowitz Inequality [DKW56, Mas90])**.** *Let* $\mathcal{D}$ *be a distribution on* $\mathbb{R}$ *with cumulative distribution function* F*. Let* $X_1, \ldots, X_m$ *be drawn i.i.d. from* $\mathcal{D}$ *and* $F_m(x) := \frac{1}{m}\sum_{i=1}^{m} \mathbb{1}[X_i \leqslant x]$*. Then*

$$\Pr\left(\sup_{x \in \mathbb{R}} |F_n(x) - F(x)| > \tau\right) < 2e^{-m\tau^2}.$$

In particular, by our choice of m, we have that with probability $1 - \beta$, the estimated quantiles $\hat{t}_i$ satisfy

$$\Pr_{z \sim \mathcal{D}}\left(z \in (\min(t_i, \hat{t}_i), \max(t_i, \hat{t}_i))\right) < \frac{\tau}{20}. \tag{1}$$

We claim that our algorithm is correct. Let $h^* : x \mapsto \mathbb{1}[x \geqslant t^*]$ be the optimal hypothesis. We claim that there exists a hypothesis h_i such that $\Pr_z(h_i(z) \neq h^*(z)) \leqslant \frac{1}{K} + 0.1\tau$. By construction (if we set $\hat{t}_{-1} \leftarrow -\infty$ and $\hat{t}_{K+1} \leftarrow +\infty$) there exists some $-1 \leqslant i \leqslant K$ for which $\hat{t}_i \leqslant t^* \leqslant \hat{t}_{i+1}$. If $\hat{t}_i = \hat{t}_{i+1}$, then h^* is in fact included in our hypothesis set. Otherwise, $\hat{t}_i < \hat{t}_{i+1}$ and we have

$$\begin{aligned}\Pr_z(h_i(z) \neq h^*(z)) &\leqslant \Pr_z(h_i(z) \neq h_{i+1}(z)) \\ &\leqslant \Pr_z(\hat{t}_i < z < \hat{t}_{i+1}) \\ &\leqslant \frac{1}{K} + \frac{\tau}{10}\end{aligned}$$

where the final inequality follows from the definition that $\hat{t}_i, \hat{t}_{i+1}$ are separated by $\frac{m}{K}$ samples and (1). Thus, setting $K = \frac{3}{\alpha}$ and $\tau < \alpha$, there exists a hypothesis h_i that is within $\frac{\alpha}{2}$ of h^* i.e. we ensure that we obtain a $\frac{\alpha}{2}$-optimal hypothesis among the set of candidate hypotheses. In particular, applying Theorem 4.5, we output a hypothesis h_j with

$$\mathrm{err}_{\mathcal{D}}(h_j) \leqslant \mathrm{err}_{\mathcal{D}}(h_i) + \frac{\alpha}{2} \leqslant \mathrm{OPT} + \frac{\alpha}{2}.$$

Now, we argue that our algorithm is approximately replicable. Consider two runs of the algorithm, and let $\{\hat{t}_i^{(1)}\}, \{\hat{t}_i^{(2)}\}$ denote the estimated quantiles in each run of the algorithm, and $\{h_i^{(1)}\}, \{h_i^{(2)}\}$ the corresponding hypotheses. Conditioned on (1), we have that with probability at least $1 - \beta$, the following holds for all i:

$$\begin{aligned}\Pr_{z\sim\mathcal{D}}\left(h_i^{(1)}(z) \neq h_i^{(2)}(z)\right) &= \Pr_z\left(z \in (\min(\hat{t}_i^{(1)}, \hat{t}_i^{(2)}), \max(\hat{t}_i^{(1)}, \hat{t}_i^{(2)}))\right) \\ &= \int \left|\mathbb{1}[x \leqslant \hat{t}_i^{(1)}] - \mathbb{1}[x \leqslant \hat{t}_i^{(2)}]\right| \mathrm{d}x \\ &\leqslant \int \left|\mathbb{1}[x \leqslant \hat{t}_i^{(1)}] - \mathbb{1}[x \leqslant t_i]\right| + \left|\mathbb{1}[x \leqslant t_i] - \mathbb{1}[x \leqslant \hat{t}_i^{(2)}]\right| \mathrm{d}x \\ &= \sum_{j=1}^{2} \Pr_z\left(z \in (\min(t_i, \hat{t}_i^{(j)}), \max(t_i, \hat{t}_i^{(j)}))\right) \\ &< \tau.\end{aligned}$$

In particular, we satisfy the assumption of Theorem 4.5 that $\mathrm{dist}_{\mathcal{D}}(h_i^{(1)}, h_i^{(2)}) < \tau$ for all i. Since HypothesisSelection is $\left(\frac{\rho}{2}, \tau\right)$-robustly replicable, we obtain via a union bound that the same index is selected with probability at least $1 - \beta - \frac{\rho}{2} \geqslant 1 - \rho$. The proof concludes by setting $\tau \leqslant \gamma$.

To bound the sample complexity, note that quantile estimation requires $\tau < \min(\alpha, \gamma)$ and applying Theorem 4.5 requires $\tau \ll \frac{\rho\alpha}{\log(K/\beta)}$. Thus, the final sample complexity is

$$O\left(\frac{\log(1/\beta)\log^2(1/\alpha\beta)}{\rho^2\alpha^2} + \frac{\log(1/\beta)}{\gamma^2} + \frac{\log^3(1/\alpha\beta)}{\rho^2\alpha^2}\right) = O\left(\frac{\log(1/\beta)}{\gamma^2} + \frac{\log^3(1/\alpha\beta)}{\rho^2\alpha^2}\right).$$

□

4.5 Approximately Replicable Realizable Learning

In this section, we give a simple algorithm with improved sample complexity in the 'realizable' PAC learning setting, giving the second half of Theorem 1.7. Here correctness is only required when the distribution $\mathcal{D}$ is realizable, but we still ensure approximate replicability over *all* input distributions.

Theorem 4.15. *Let A be a realizable (α, β)-learner with sample complexity $m(\alpha, \beta)$. There is a (ρ, γ)-approximately replicable realizable (α, β)-learner with sample complexity*

$$O\left(\frac{d + \log(1/\min(\rho, \beta))}{\rho^2 \min(\alpha, \gamma)}\right).$$

Proof. Our algorithm proceeds in two steps. First, we replicably estimate OPT to determine whether $\mathcal{D}$ is realizable. If not, we can replicably output an arbitrary hypothesis (since we have no correctness requirement in this case). Otherwise, since every accurate solution is close to the optimal hypothesis, it suffices to run a (non-replicable) realizable PAC learner. We formalize this in Algorithm 10.

Algorithm 10: REALIZABLEAPPROXIMATEREPLICABILITY(A)

Input : Realizable PAC learner A. Sample access to $\mathcal{D}$.
Parameters : (ρ, γ)-approximate replicability, α accuracy, β error.
Output : (ρ, γ)-approximately replicable realizable (α, β)-learner.

Step 1: Replicably Estimate OPT**.**
Collect $m_1 = O\left(\frac{d+\log(1/\min(\rho,\beta))}{\rho^2\alpha}\right)$ samples from $\mathcal{D}$, denoted S.
Draw a random threshold $r \in [0.1\alpha, 0.2\alpha]$.
Compute $\hat{\mathrm{OPT}} := \min_{h\in\mathcal{H}} \mathrm{err}_S(h)$.
if $\hat{\mathrm{OPT}} > r$ **then**
  return $h : x \to 1$.
Step 2: Running a Realizable Learner.
return $\hat{h} \leftarrow \arg\min_{h\in\mathcal{H}} \mathrm{err}_S(h)$.

We prove the correctness and replicability of our algorithm. We begin with the following lemma, which states that Step 1 replicably decides whether the distribution is realizable.

Lemma 4.16. *At the end of Step 1, the following hold:*

1. *If* $\mathrm{OPT} = 0$*, then* $\hat{\mathrm{OPT}} < r$ *with probability* $1 - \beta$*.*
2. *If* $\mathrm{OPT} > \alpha$*, then* $\hat{\mathrm{OPT}} \geqslant r$ *with probability* $1 - \beta$*.*
3. *The output is* 11ρ*-replicable.*

Proof. The argument is similar to the general sample complexity for realizable learning with ERM. In this argument, we require a strengthening of the guarantees of realizable PAC learning.

Let S consist of a dataset of m i.i.d. samples from $\mathcal{D}$.

$$E(h) := ((\mathrm{err}_{\mathcal{D}}(h) < \alpha) \wedge |\mathrm{err}_S(h) - \mathrm{err}_{\mathcal{D}}(h)| > \rho\alpha) \vee \left(\mathrm{err}_{\mathcal{D}}(h) \geqslant \alpha \wedge \mathrm{err}_S(h) < \frac{\alpha}{3}\right)$$
$$B(S) := \{\exists h \text{ s.t. } E(h)\}.$$

In particular, we ensure that every hypotheses either has large empirical error (and $\hat{\mathrm{OPT}} \geqslant r$) or any near-optimal hypothesis is estimated up to error $\rho\alpha$. Let S' consist of m i.i.d. samples from $\mathcal{D}$ (independent of S) and $\sigma \in \{\pm 1\}^m$ an independent random vector. Note that to prove the lemma, it suffices to show $\Pr(B(S)) < \min(\rho, \beta)$. In particular, if $\mathrm{OPT} = 0$, we have $\hat{\mathrm{OPT}} < \rho\alpha < 0.1\alpha < r$. If $\mathrm{OPT} > \alpha$, we have $\hat{\mathrm{OPT}} > \frac{\alpha}{3} > 0.2\alpha > r$. Finally, $\hat{\mathrm{OPT}}$ (if it lies in $[0.1\alpha, 0.2\alpha]$, lies in an interval of width $\rho\alpha$, so that r falls in this interval with probability at most 10ρ. We union bound over the probability of $B(S)$.

We now bound the probability of $B(S)$. Given S, S', σ, define T to consist of an example from S when $\sigma = +1$ and S' when $\sigma = -1$. Define the following events:

$$\begin{aligned} E'(S,S',h) &:= \left(\left(|\mathrm{err}_S(h) - \mathrm{err}_{S'}(h)| > \frac{\rho\alpha}{2}\right) \wedge (\mathrm{err}_S(h) < \alpha)\right) \\ &\quad \vee \left(\left(\mathrm{err}_{S'}(h) > \frac{\alpha}{2}\right) \wedge \left(\mathrm{err}_S(h) < \frac{\alpha}{3}\right)\right) \\ B'(S,S') &:= \{\exists h \text{ s.t. } E'(h)\} \\ B''(S,S',\sigma) &:= \{\exists h \text{ s.t. } E'(T,T',h) \text{ where } T,T' \text{ correspond to } S,S',\sigma\}. \end{aligned}$$

Claim 4.17. *When $m \gg \frac{1}{\rho^2\alpha}$, for some sufficiently large constant, $\Pr(B'(S,S')|B(S)) > \frac{1}{2}$.*

Proof. Suppose $B(S)$ holds i.e. $E(h)$ holds for some h. We split into two cases. Suppose $\mathrm{err}_{\mathcal{D}}(h) \leqslant \alpha$ and $|\mathrm{err}_S(h) - \mathrm{err}_{\mathcal{D}}(h)| > \rho\alpha$. Then, $\mathbb{E}[\mathrm{err}_{S'}(h)] = \mathrm{err}_{\mathcal{D}}(h) \leqslant \alpha$. By a standard Chernoff bound,

$$Pr\left(|\mathrm{err}_{S'}(h) - \mathrm{err}_{\mathcal{D}}(h)| > \frac{\rho\alpha}{2}\right) < 2\exp\left(-\frac{\rho^2\alpha m}{3}\right) < \frac{1}{2}.$$

By the triangle inequality, we have

$$|\mathrm{err}_S(h) - \mathrm{err}_{S'}(h)| > \frac{\rho\alpha}{2}.$$

On the other hand, suppose $\mathrm{err}_{\mathcal{D}}(h) > \alpha$ and $\mathrm{err}_S(h) < \frac{\alpha}{3}$. Following an identical argument, we have $\mathbb{E}[\mathrm{err}_{S'}(h)] > \alpha$ and

$$\Pr\left(\mathrm{err}_{S'}(h) < \frac{\alpha}{2}\right) < \exp\left(-\frac{m\alpha}{12}\right) < \frac{1}{2}.$$

□

Thus, we have

$$\Pr(B(S)) = \frac{\Pr(B'(S,S') \cap B(S))}{\Pr(B'(S,S')|B(S))} \leqslant \frac{\Pr(B'(S,S'))}{\Pr(B'(S,S')|B(S))} \leqslant 2\Pr(B'(S,S')).$$

Observe that since S, S' and T, T' are identically distributed, $\Pr(B'(S,S')) = \Pr(B''(S,S',\sigma))$. We thus hope to bound $\Pr(B''(S,S',\sigma))$. It suffices to bound the probability of $B''(S,S',\sigma)$ conditioned on any fixed S, S'.

Claim 4.18. *For any fixed S, S', h,*

$$\Pr_{\sigma}\left(E'(T,T',h)|S,S'\right) < \exp\left(-\Omega\left(\rho^2\alpha m\right)\right).$$

Proof. Consider the predictions on S, S', denoted $\{h(x_i)\}$ and $\{h(x_i')\}$. Note that σ independently distributes each sample to T, T' uniformly. Consider the set of indices where exactly one of $h(x_i), h(x_i')$ is correct. We may assume this set has size $k \geqslant \rho\alpha m$ (otherwise $E'(T,T',h)$ cannot happen). Then, the difference in the errors $\mathrm{err}_T(h), \mathrm{err}_{T'}(h)$ is distributed as $X \sim \frac{\mathrm{Binom}(k,0.5)}{m}$. Consider two cases. Suppose $k < 10\alpha m$, so that a standard Chernoff bound implies

$$\begin{aligned} \Pr_{\sigma}(E(T,T',h)|S,S') &< \Pr(|X - \mathbb{E}[X]| > \rho\alpha m) \\ &= \Pr\left(|X - \mathbb{E}[X]| > \frac{2\rho\alpha m}{k}\frac{k}{2}\right) \\ &< \exp\left(-\Omega\left(\frac{\rho^2\alpha^2m^2}{k}\right)\right) \\ &< \exp\left(-\Omega\left(\rho^2\alpha m\right)\right). \end{aligned}$$

On the other hand, when $k \geqslant 10\alpha m$, we bound the probability that $\mathrm{err}_T(h) < \alpha$. By a Chernoff Bound, we have

$$\Pr_{\sigma}(E(T,T',h)|S,S') < \Pr(\mathrm{err}_T(h) < \alpha) < \exp\left(-\Omega\left(k\right)\right) < \exp\left(-\Omega\left(\alpha m\right)\right).$$

□

Finally, for any fixed S, S', we bound the probability that there exists any hypothesis h with $E(T,T',h)$ holds. Now, for a fixed S, S', we can apply Sauer's Lemma to union bound over all labellings of $S \cup S'$, rather than all hypotheses in $\mathcal{H}$.

Lemma 4.19 (Sauer's Lemma)**.** *Let $\mathcal{H}$ be a class with VC Dimension d and $S \in \mathcal{X}^m$. Let $\Pi(\mathcal{H})$ denote the set of labellings of S under $\mathcal{H}$. Then,*

$$|\Pi(\mathcal{H})| = m^{O(d)}.$$

Then, for any S, S', we have

$$\Pr_{\sigma}\left(B''(S,S',\sigma)|S,S'\right) < m^{O(d)}e^{-\Omega\left(\rho^2\alpha m\right)} < \min(\beta,\rho)$$

for $m \gg \frac{1}{\rho^2\alpha}\left(d + \log(1/\min(\rho,\beta))\right)$. This completes the proof of Lemma 4.16. □

We now continue with the proof of Theorem 4.15. Correctness is clear, since we proceed to Step 2 and return a hypothesis with empirical error at most 0.1α (since $\mathrm{err}_S(h^*) < \rho\alpha < 0.1\alpha$. We thus argue replicability. Conditioned on the success of Step 1, we immediately have replicability if the algorithm returns $x \to 1$. Thus, we assume that the algorithm proceeds to Step 2, and thus assume $\mathrm{OPT} < \alpha$. In particular, there exists a hypothesis with $\mathrm{err}_{\mathcal{D}}(h) < \alpha$ so that $\mathrm{err}_S(h) \leqslant 1.1\alpha$. In two runs of the algorithm, we obtain $\hat{h}_1, \hat{h}_2$ both with empirical error at most 1.1α. Following identical arguments as in Lemma 4.16, we can prove that both hypotheses have true distributional error at most 2α. Then, we conclude with the triangle inequality

$$\Pr(\hat{h}_1(x) \neq \hat{h}_2(x)) \leqslant \Pr(\hat{h}_1(x) \neq y) \leqslant 2\alpha$$

since $\hat{h}_1(x) = \hat{h}_2(x)$ when both are correct. □

4.6 Lower Bounds for Approximate Replicability

In this section, we argue approximate replicable learning of VC classes can be reduced to the bias estimation problem, a useful distribution estimation task used in a number of prior replicability lower bounds. As in replicable prediction, our goal is to "plant" one instance of bias estimation at a single randomized point. However, we now need to ensure the instance remains indistinguishable for *every* input distribution to the bias estimation problem.

Theorem 1.9. *Let $\rho < 0.001$. Any (ρ,γ)-approximately replicable $(\alpha, 0.001)$-learner requires sample complexity $\Omega\left(\frac{d}{\alpha^2(\rho+\gamma)^2\sqrt{\log(1/\rho)}}\right)$.*

As discussed in the technical overview, we will take advantage of the fact that replicable bias estimation is still hard in the average-case setting. In particular, even when the adversary selects $\mathrm{Rad}(p)$ with $p \in [-\alpha, \alpha]$ uniformly at random (and the algorithm knows this distribution), the sample complexity lower bound $\Omega\left(\frac{1}{\rho^2\alpha^2}\right)$ still holds. We will use this to obtain a lower bound on approximately replicable PAC learning.

Before proceeding to the proof, we define some relevant terms. Let $\{x_1, \ldots, x_d\}$ denote a set of d shattered elements. Given any meta-distribution $\mathcal{M}$ over Rademacher distributions (i.e. a distribution over distributions over $\{\pm 1\}$), define $\mathcal{M}^{\otimes d}$ to be the meta-distribution that samples distributions as follows:

1. Sample d distributions $\mathcal{D}_1, \ldots, \mathcal{D}_d \sim \mathcal{M}$ independently.
2. Define $\mathcal{D}^{\otimes d}$ to be the following distribution:
   (a) Sample $i \in [d]$ uniformly.
   (b) Sample $y \sim \mathcal{D}_i$ and return (x_i, y).

For the rest of the proof, the reader can assume $\mathcal{M}$ is the meta-distribution given by sampling $p \in [-\alpha, \alpha]$ uniformly and providing samples from $\mathrm{Rad}(p)$, although our reduction will hold for any meta-distribution $\mathcal{M}$. We will prove lower bounds for algorithms that are correct and replicable against a meta-distribution. Let's first be careful in defining what this means.

Definition 4.20 (Distributional Accuracy)**.** *Let $\mathcal{M}$ consist of a distribution over input labeled distributions. An algorithm A is (α, β)-accurate with respect to $\mathcal{M}$ if*

$$\Pr_{\mathcal{D}\sim\mathcal{M}, S\sim\mathcal{D}^m}(\mathrm{err}_{\mathcal{D}}(A(S)) > \alpha) < \beta.$$

While the above notion is defined for PAC learners, they naturally extend to distributions over $\{\pm 1\}$ as follows.

Definition 4.21 (Distributional Accuracy)**.** *Let $\mathcal{M}$ consist of a distribution over distributions over $\{\pm 1\}$. An algorithm $A : \{\pm 1\}^m \to \{\pm 1\}$ is (α, β)-accurate with respect to $\mathcal{M}$ if*

$$\Pr_{\mathcal{D}\sim\mathcal{M}, S\sim\mathcal{D}^m}((\mathrm{sign}(v) - p)p > 2\alpha) < \beta.$$

where $\mathcal{D} \sim \mathrm{Rad}(p)$ and $v = A(S)$ is the output of A.

Next, we define distributional replicability, and for PAC learners, distributional approximate replicability.

Definition 4.22 (Distributional (Approximate) Replicability)**.** *Let $\mathcal{M}$ consist of a distribution over input labeled distributions. An algorithm A is ρ-replicable with respect to $\mathcal{M}$ if*

$$\Pr_{\mathcal{D}\sim\mathcal{M}, S_1, S_2\sim\mathcal{D}^m}(A(S_1) \neq A(S_2)) < \rho.$$

An algorithm is (ρ, γ)-approximately replicable with respect to $\mathcal{M}$ if

$$\Pr_{\mathcal{D}\sim\mathcal{M}, S_1, S_2\sim\mathcal{D}^m}(\mathrm{dist}_{\mathcal{D}}(A(S_1), A(S_2)) > \gamma) < \rho.$$

Our goal is now to relate distributional replicability and accuracy of algorithms to replicability and accuracy via the MiniMax theorem. Consider the following games, defined $\mathcal{G}_{m,\gamma}, \mathcal{G}_{m,\alpha}, \mathcal{G}_{m,\alpha,\gamma}$.

1. The algorithm player (randomly) chooses an algorithm: $A : (\mathcal{X} \times \{\pm 1\})^m \to \{\mathcal{X} \to \{\pm 1\}\}$.
2. The adversary player (randomly) chooses a distribution $\mathcal{D}$ over $\mathcal{X} \times \{\pm 1\}$.
3. Dataset T, S_1, S_2 of size m_u, m_s, m_s are drawn i.i.d. from $\mathcal{D}_{\mathcal{X}}$.
4. The algorithm player wins $\mathcal{G}_{m,\gamma}$ if $\mathrm{dist}_{\mathcal{D}}(A(T, S_1), A(T, S_2)) \leqslant \gamma$, $\mathcal{G}_{m,\alpha}$ if $\mathrm{err}_{\mathcal{D}}(A(T, S_1)) \leqslant \alpha$, and $\mathcal{G}_{m,\alpha,\gamma}$ if both occur. Otherwise, in each game, the adversary player wins. The value of the game, denoted $\mathcal{G}_{m,\alpha,\gamma}(A, \mathcal{D})$, is 1 when the algorithm player wins and 0 when the adversary player wins.

When the sample complexity of the algorithm m and parameters α, γ are clear, we simply write $\mathcal{G}$. Note that any randomized algorithm can be viewed as a randomized strategy for the algorithm player while a meta-distribution can be viewed as a randomized strategy for the adversary player. We are now ready to prove Theorem 1.9.

Proof of Theorem 1.9. Our overall proof will follow two steps: (1) from an approximately replicable learner for $\mathcal{H}$, we obtain a distributionally approximately replicable learner for $\mathcal{H}$, and (2) from the distributionally approximately replicable learner we obtain a distributionally replicable algorithm for bias estimation, which requires $\Omega\left(\frac{1}{\rho^2\alpha^2}\right)$ samples. We begin with step (1).

Suppose we have an (ρ, γ)-approximately replicable (α, β)-learner for $\mathcal{H}$ with sample complexity m. Then, from the easy direction of the MiniMax Theorem, we have

$$1 - (\rho + \beta) \leqslant \max_{A} \min_{\mathcal{D}} \mathbb{E}_A[\mathcal{G}(A, \mathcal{D})] \leqslant \min_{\mathcal{M}} \max_{A_0} \mathbb{E}_{\mathcal{D}\sim\mathcal{M}}[\mathcal{G}(A_0, \mathcal{D})] \leqslant \min_{\mathcal{M}^{\otimes d}} \max_{A_0} \mathbb{E}_{\mathcal{D}\sim\mathcal{M}}[\mathcal{G}(A_0, \mathcal{D})].$$

Note that the final inequality follows since restricting the choice of the adversary to meta-distributions of the form $\mathcal{M}^{\otimes d}$ can only help the algorithm player. Thus, we claim that for any meta-distribution $\mathcal{M}^{\otimes d}$, we have a deterministic algorithm A_0 that is (α, β)-accurate and (ρ, γ)-approximately replicable with respect to $\mathcal{M}^{\otimes d}$. This follows as

$$\begin{aligned}
1 - (\rho + \beta) &\leqslant \min_{\mathcal{M}^{\otimes d}} \max_{A_0} \mathbb{E}_{\mathcal{D}\sim\mathcal{M}^{\otimes d}} [\mathcal{G}(A_0, \mathcal{D})] \\
&= \min_{\mathcal{M}^{\otimes d}} \max_{A_0} \\
&\qquad \Pr_{\mathcal{D}\sim\mathcal{M}^{\otimes d}, S_1, S_2 \sim \mathcal{D}^m} \left(\left(\max_i \mathrm{err}_{\mathcal{D}}(A_0(T, S_i)) < \alpha \right) \wedge \left(\mathrm{dist}_{\mathcal{D}}(A_0(T, S_1), A_0(T, S_2)) < \gamma \right) \right).
\end{aligned}$$

In the above, we take the maximum over strategies for the algorithm player, denoted A_0, which are is the set of all deterministic algorithms. This completes step (1). We now proceed with step (2) and argue that from A_0 we can obtain an algorithm that is accurate and replicable with respect to $\mathcal{M}$.

Lemma 4.23. *Let $\mathcal{M}$ be any meta-distribution over Rademacher distributions. Suppose A_0 is a (ρ, γ)-approximately replicable (α, β)-accurate learner with respect to $\mathcal{M}^{\otimes d}$ with sample complexity m. Then, there is a $(2\rho + \gamma)$-replicable $(100\alpha, 0.1)$-accurate algorithm A_1 with respect to $\mathcal{M}$ with sample complexity $O\left(\frac{m}{d}\left(\sqrt{\log(1/\rho)}\right)\right)$.*

In the above lemma, we assume that $\mathcal{M}$ is explicitly known. We reiterate that for replicable bias estimation, the hard meta-distribution $\mathcal{M}$ is explicitly known, i.e. sample from $\mathrm{Rad}(p)$ where $p \in [-\alpha, \alpha]$ is chosen uniformly.

Proof of Lemma 4.23. We describe our algorithm in Algorithm 11.

The sample complexity of our algorithm is immediate.

By construction, since $\mathcal{D} \sim \mathcal{M}$, the dataset S consists of m samples drawn from a distribution sampled from the meta-distribution $\mathcal{M}^{\otimes d}$. Since A_0 is distributionally approximately replicable, with probability at least $1 - \rho$, A_0 outputs γ-close hypotheses (i.e. hypotheses that disagree on at most a γ-fraction of $\{x_1, \ldots, x_d\}$). Since A_0 is distributionally accurate, with probability at least $1 - \beta$, A_0 outputs α-accurate hypotheses. We condition on these events. In addition, we condition on the event that the sample limit is not reached. Let m_r denote the number of samples x_r drawn. Since the expectation $\mathbb{E}[m_r] = \frac{m}{d}$, a standard Chernoff bound yields that $m_r > 10\frac{m}{d}\sqrt{\log(1/\rho)}$ with probability at most ρ.

Conditioned on the above, we argue our algorithm is accurate and replicable with respect to $\mathcal{M}$. We begin with accuracy. Let us analyze the error of the output hypothesis $h \leftarrow A(S)$. Let $\mathcal{D}'$ be the distribution over

Algorithm 11: APXREPLICABILITYHARDNESSAMPLIFICATION$(A_0, \mathcal{M})$

Input : Approximately replicable PAC learner A_0. Sample access to $\mathcal{M}$ and $\mathcal{D}$.
Parameters : ρ replicability, α accuracy, β error probability.
Output : ρ-replicable (α, β)-accurate algorithm with respect to $\mathcal{M}$.

Sample $r \in [d]$ uniformly.
for $i \in [d] \backslash r$ **do**
  Sample $\mathcal{D}_i \sim \mathcal{M}$ independently.
Initiate S as an empty multi-set and counter $c \leftarrow 0$.
for $j \in [m]$ **do**
  Sample $x_j \sim \mathcal{X}$ uniformly.
  if $x_j = x_r$ **then**
    Sample $y_j \sim \mathcal{D}$ and add (x_j, y_j) to S.
    Increment $c \leftarrow c + 1$.
    if $c \geqslant 10\frac{m}{d}\sqrt{\log(1/\rho)}$ **then**
      return $h : x \mapsto 1$.
  else
    Sample $y_j \sim \mathcal{D}_i$ where $x_j = x_i$ and add (x_j, y_j) to S.
return $h(x_r)$ where $h \leftarrow A_0(S)$.

labeled samples (x, y) where the marginal is given by the uniform distribution on $\{x_1, \ldots, x_d\}$, each domain element x_i is labeled according to $\mathcal{D}_i$, and x_r is labeled according to $\mathcal{D}$, so that S consists of m samples from $\mathcal{D}'$. For all i, let $p_i = \mathbb{E}_{\mathcal{D}'}[y|x = x_i] = \mathbb{E}[\mathcal{D}_i]$ be the expected label of x_i under $\mathcal{D}'$, i.e. the mean of $\mathcal{D}_i$. Let h^* be the optimal hypothesis for $\mathcal{D}'$. Then, if h is α-accurate,

$$\begin{aligned}\alpha &\geqslant \mathrm{err}_{\mathcal{D}'}(h) - \mathrm{err}_{\mathcal{D}'}(h^*) \\ &= \frac{1}{d}\sum_{i=1}^{d} \mathbb{1}[h(x_i) \neq h^*(x_i)]|p_i|.\end{aligned}$$

Define $\mathrm{err}_i := \mathbb{1}[h(x_i) \neq h^*(x_i)]|p_i|$. Then, for any fixed $\mathcal{D}'$, we note that $r \sim [d]$ is still uniformly distributed, as $\mathcal{D}_i$ for $i \neq r$ and $\mathcal{D}$ are all sampled i.i.d. from $\mathcal{M}$. Since A_0 only observes samples $S \sim \mathcal{D}'$, we note that err_i are identically distributed. In particular, conditioned on $\frac{1}{d}\sum_{i=1}^{d} \mathrm{err}_i \leqslant \alpha$, we have

$$\mathbb{E}_{\mathcal{D}\sim\mathcal{M}, \mathcal{D}'\sim\mathcal{M}^{\otimes d}, S\sim\mathcal{D}'}[\mathrm{err}_r] \leqslant \alpha.$$

By Markov's Inequality,

$$\Pr_{\mathcal{D}\sim\mathcal{M}, \mathcal{D}'\sim\mathcal{M}^{\otimes d}, r\sim[d]}(\mathrm{err}_r > 100\alpha) < 0.01.$$

Union bounding with the above events, we obtain a $(100\alpha, 0.1)$-accurate algorithm with respect to $\mathcal{M}$.

Finally, we argue replicability. Consider two outputs of the algorithm h_1, h_2. Conditioned on the success of A, h_1, h_2 disagree in at most γd domain elements $\{x_i\}$. Let $E = \{i \text{ s.t. } h_1(x_i) \neq h_2(x_i)\}$ be the set of domain elements where h_1, h_2 disagree. Again, since for any fixed $\mathcal{D}'$, r is uniformly distributed in $[d]$, the probability that $r \in E$ is at most γ. Thus, we conclude that our algorithm is $(2\rho + \gamma)$-replicable with respect to $\mathcal{M}$ by a union bound (since A outputs γ-close hypotheses with probability ρ and we do not exceed the sample bound with probability ρ). □

It remains to prove a lower bound for algorithms that are distributionally replicable and accurate with respect to some meta-distribution $\mathcal{M}$. Towards this, we fix $\mathcal{M}$ to be the specific hard meta-distribution mentioned above. For any fixed accuracy threshold α and replicability parameter ρ, define $\mathcal{M}_\alpha$ to be the meta-distribution that uniformly samples a mean p from $[-\alpha, \alpha]$ and produces samples from $\mathrm{Rad}(p)$. Then, $\mathcal{M}_\alpha^{\otimes d}$ is the meta-distribution defined above by picking label distributions from $\mathcal{M}_\alpha$ independently for each domain element. From [ILPS22, HIK+24], it is known that any algorithm that is $(\alpha, 0.1)$-accurate and ρ-replicable with respect to $\mathcal{M}_\alpha$ requires $\Omega\left(\frac{1}{\alpha^2\rho^2}\right)$ samples. In particular, we have

$$\frac{m}{d}\sqrt{\log(1/\rho)} = \Omega\left(\frac{1}{\alpha^2(\rho+\gamma)^2}\right)$$

which yields the desired lower bound. □

4.7 Shared Randomness

We have seen that relaxing the definition of replicability to approximate replicability removes the necessity of shared randomness for basic tasks such as mean estimation, but our more involved methods for PAC learning rely heavily on a shared random string. Moreover, our sample lower bound from Theorem 1.9 does not show this is necessary, as the reduction itself uses shared randomness. In this section, we give a direct argument showing shared randomness is provably necessary for approximately replicable agnostic PAC learning.

To obtain a lower bound against deterministic algorithms, we instead give a *deterministic reduction* from sign-one-way marginals. In order to make this reduction useful, we give the first impossibility result for deterministic approximately replicable algorithms.

Definition 4.24. *An algorithm A with m samples solving the one-way marginals problem (Definition 5.11) is (ρ, γ)-approximately replicable if for every distribution $\mathcal{D}$,*

$$\Pr_{S_1, S_2 \sim \mathcal{D}^m, r}\left(\|A(S_1; r) - A(S_2; r)\|_0 > \gamma n\right) < \rho.$$

First, we show that no deterministic approximately replicable algorithm can solve the sign-one-way marginals problem. In fact, we argue this even for the case $d = 2$.

Theorem 4.25. *There is no deterministic algorithm that $(0.01, 0.5)$-approximately replicable and $(0.1, 0.01)$-accurately solves the sign-one-way marginals problem.*

Proof. Suppose such an algorithm A exists. Note that A has range $\mathcal{Y} = \{(-1,-1), (-1,1), (1,-1), (1,1)\}$. Consider a distribution $\mathcal{D}$ with mean $p = (p_1, p_2)$ which consists of a product of two Rademacher distributions. For all p, define $S_p := \{y \in \mathcal{Y} \text{ s.t. } \Pr_S(A(S) = y) \geqslant 0.1\}$ to be the set of canonical outcomes of A, i.e., A outputs y with probability at least 0.1. Since $|\mathcal{Y}| \leqslant 4$, S_p is non-empty for all $\mathcal{D}$. Now, suppose S_p has two distinct elements x, y. We claim $\|x - y\|_0 \leqslant 1$. Otherwise,

$$\Pr_{T_1, T_2}\left(\|A(T_1) - A(T_2)\|_0 > 1\right) \geqslant 2\Pr(A(T) = x)\Pr(A(T) = y) \geqslant 0.02,$$

violating the approximate replicability constraint. Then, we have $|S_p| \leqslant 2$ for all p and the two elements can disagree in at most one coordinate. We claim that there is some p^* for which S_{p^*} contains x, y where $\|x - y\|_0 = 2$, which violates approximate replicability.

Towards this, for each $y \in \mathcal{Y}$, let $P_y : [-1, 1]^2 \to [0, 1]$ denote the function $\Pr_{T \sim \mathcal{D}_p}(A(T) = y)$. Note that

$$P_y(p) = \sum_{T \text{ s.t. } A(T) = y} \Pr_{T \sim \mathcal{D}_p}(T)$$

is continuous since $\Pr(T)$ is a product of $p_1, (1-p_1), p_2, (1-p_2)$ and therefore continuous in p. In particular, if we denote $B_y := \{p \text{ s.t. } P_y(p) \geqslant 0.1\}$, we have that B_y is closed for all y. Define the sets,

$$\begin{aligned} B_{\text{left}} &= B_{(-1,-1)} \cup B_{(-1,1)} \\ B_{\text{right}} &= B_{(1,-1)} \cup B_{(1,1)} \\ B_{\text{up}} &= B_{(-1,1)} \cup B_{(1,1)} \\ B_{\text{down}} &= B_{(-1,-1)} \cup B_{(1,-1)}. \end{aligned}$$

Then, define the continuous map

$$F : p \mapsto (\text{dist}(p, B_{\text{left}}) - \text{dist}(p, B_{\text{right}}), \text{dist}(p, B_{\text{up}}) - \text{dist}(p, B_{\text{down}})).$$

We would like to apply the Poincare-Miranda Theorem.

Theorem 4.26. *Consider n continuous, real-value functions $f_1, \ldots, f_n : [-1, 1]^n \to \mathbb{R}$. Assume for each x_i, f_i is non-positive when $x_i = -1$ and non-negative when $x_i = +1$. Then, there is a point $p \in [-1, 1]^n$ when all f_i are equal to 0.*

Note that when $p_1 = -1$, then since A is (α, β)-accurate, with probability $1 - \beta$, we have

$$\alpha \geqslant \frac{1}{2}(1 + v_1).$$

Rearranging, $v_1 \leqslant 2\alpha - 1 < 0$ i.e. $v_1 = -1$ so that $A(T) \in \{(-1, -1), (-1, 1)\}$ with probability at least $1 - \beta \geqslant 0.99$ so that $F(p)_1 < 0$. Similarly, when $p_1 > 0$, we have $A(T) \in \{(1, -1), (1, 1)\}$ with probability at least $1 - \beta \geqslant 0.99$ so that $F(p)_1 > 0$. A similar argument holds for the second coordinate, so we may apply the Poincare-Miranda theorem and obtain a point p^* where $F(p^*) = 0$.

Now, since S_{p^*} is not empty, assume without loss of generality that $(-1, -1) \in S_{p^*}$. Then, since $F(p^*) = 0$ we have that $\text{dist}(p^*, B_{\text{right}}) = \text{dist}(p^*, B_{\text{up}}) = 0$. Assume without loss of generality that $p^* \notin B_{(1,1)}$, otherwise we are done. Then, since $B_{(1,1)}$ is closed, we have $\text{dist}(p^*, B_{(1,-1)}) = \text{dist}(p^*, B_{(1,-1)}) = 0$ and therefore $p^* \in B_{(1,-1)} \cap B_{(-1,1)}$. In particular, $(-1, 1), (1, -1) \in S_{p^*}$, as desired. □

Using this lower bound, we show that any approximately replicable PAC learner must also use shared randomness.

Theorem 1.13. *There is no deterministic $(0.01, 0.5)$-approximately replicable $(0.05, 0.01)$-learner for any class with VC dimension $d \geqslant 2$.*

Proof. Consider a set of two shattered elements $\{x_1, x_2\}$ and the uniform distribution over $\mathcal{X}$. We reduce from 2-dimensional sign-one-way marginals. Given $p \in [0, 1]^2$, the label of x_i is 1 with probability p_i and -1 otherwise. Suppose there is a deterministic approximately replicable learner, denoted A and we have sample access to a 2-coin problem instance. We construct samples for A by uniformly sampling x from $\mathcal{X}$ and labeling using samples from the corresponding coin. Given the output $h \leftarrow A$, we return $v \leftarrow (h(x_1), h(x_2))$. The excess error of $h \leftarrow A$ is exactly $\frac{1}{2}\sum_i 2|p_i|\mathbb{1}[h(x_i) \neq h^*(x_i)| = \sum_i |p_i|\mathbb{1}[h(x_i) \neq h^*(x_i)]$ which is (up to a factor of 2) the error of the solution v. Furthermore, whenever h is approximately replicable, either $h(x_1)$ or $h(x_2)$ is consistent, so that v is approximately replicable as well. Thus, we obtain a deterministic $(0.01, 0.5)$-approximately replicable $(0.1, 0.01)$-accurate algorithm for the sign-one-way marginals problem. Furthermore, our reduction is deterministic, so A must be randomized from Theorem 4.25. □

5 Semi-Replicable Learning

In this section, we give our algorithms and lower bounds for semi-replicable learning.

Definition 5.1 (Replicable Semi-Supervised Learning)**.** *Let $\mathcal{H}$ be a concept class. A (randomized) algorithm A is a ρ-semi-replicable (α, β)-learner for $\mathcal{H}$ with shared unlabeled sample complexity m_u and labeled sample complexity m_s if for any distribution $\mathcal{D}$, A satisfies the following:*

1. *Let $S \sim \mathcal{D}^{m_s}$, $U \sim \mathcal{D}_{\mathcal{X}}^{m_u}$ and r denote the internal randomness of A. Then*

$$\Pr_{S,U,r}\left(\mathrm{err}_{\mathcal{D}}(A(S;U,r)) \geqslant \mathrm{OPT}_{\mathcal{H}}(\mathcal{D}) + \alpha\right) < \beta.$$

2. *Let $S_1, S_2 \sim \mathcal{D}^{m_s}$ be drawn independently and $U \sim \mathcal{D}_{\mathcal{X}}^{m_u}$. Let r denote the internal randomness of A. Then*

$$\Pr_{S_1,S_2,U,r}\left(A(S_1;U,r) \neq A(S_2;U,r)\right) < \rho.$$

For convenience, we say A has sample complexity (m_u, m_s).

Theorem 1.10. *There is a ρ-semi-replicable (α, β)-learner for with shared (unlabeled) sample complexity $O\left(\frac{d}{\alpha}\log(1/\beta)\right)$ and labeled sample complexity $O\left(\frac{d^2}{\alpha^2\rho^2}\log^5\frac{1}{\rho\beta}\right)$.*

We will need the following replicable learner for finite hypothesis classes of [BGH+23].

Theorem 5.2 (Theorem 5.13 of [BGH+23])**.** *Let $\mathcal{H}$ be a finite concept class. There is a ρ-replicable (α, β)-learner for $\mathcal{H}$ with sample complexity*

$$O\left(\frac{d^2\log^2|\mathcal{H}| + \log\frac{1}{\rho\beta}}{\alpha^2\rho^2}\log^3\frac{1}{\rho}\right).$$

Proof of Theorem 1.10. The argument is essentially the same as the standard upper bound for the semi-private model [ABM19, HKLM22]. We first use the shared unlabeled samples to create a small shared subset of hypotheses $\mathcal{H}_0$ that contains a hypothesis $\alpha/2$-close to optimal with high probability, then learn from this class replicably using Theorem 5.2.

Step 1: Constructing an α-non-uniform cover with unlabeled samples.

Let $m(\alpha, \beta) = O\left(\frac{d+\log\frac{1}{\beta}}{\alpha}\right)$ denote the sample complexity of an (improper) realizable learner A for hypothesis class $\mathcal{H}$. By taking $m_u = m(\alpha/2, \beta/2)$ unlabeled samples U and constructing the set of hypotheses

$$C(U) = \{A(h(U)) \text{ s.t. } h \in \mathcal{H}\}$$

obtained by running A on all possibly labellings of U under $\mathcal{H}$, Claim 2.3 of [HKLM22] shows that with probability at least $1 - \beta/2$, there exists $h' \in C(U)$ such that $\mathrm{err}_{\mathcal{D}}(h') \leqslant \mathrm{OPT}_{\mathcal{H}}(\mathcal{D}) + \alpha/2$.

Step 2: Replicably learning from the α-non-uniform cover.

Consider now the finite hypothesis class $C(U)$. Note that since the concept class $\mathcal{H}$ has VC dimension d, there are at most $m_u^{O(d)}$ possible labellings of U (see Lemma 4.19) so that

$$|C(U)| \leqslant m_u^{O(d)}.$$

Then, we draw sufficiently many samples so that applying Theorem 5.2, we have a ρ-replicable $(\alpha/2, \beta/2)$-learner for $C(U)$. Thus the supervised sample complexity is

$$m_s = O\left(\frac{\log^2 |C(U)| + \log \frac{1}{\rho\beta}}{\alpha^2\rho^2} \log^3 \frac{1}{\rho}\right) = O\left(\frac{d^2 \log^2 \frac{d+\log(1/\beta)}{\alpha} + \log \frac{1}{\rho\beta}}{\alpha^2\rho^2} \log^3 \frac{1}{\rho}\right).$$

We now show the algorithm has the claimed correctness and replicability properties.

Correctness. By a union bound, with probability at least $1 - \beta$ we have that the $(\alpha/2, \beta/2)$-learner of $C(U)$ outputs a hypothesis h such that

$$\mathrm{err}_{\mathcal{D}}(h) \leqslant \mathrm{OPT}_{C(U)} + \alpha/2 \leqslant \mathrm{OPT}_{\mathcal{H}} + \alpha.$$

Replicability. Since the unsupervised samples U and internal randomness of A are shared between two runs of the algorithm, both have the finite concept class $C(U)$ after Step 1 of the algorithm. Replicability then follows from Theorem 5.2. □

We now argue that Theorem 1.10 is optimal.

5.1 Lower Bounds for Shared Sample Complexity

First, we give a bound on the shared sample complexity.

Theorem 1.12. *There exists a class $\mathcal{H}$ such that any 0.0001-semi-replicable $(\alpha, 0.0001)$-learner for $\mathcal{H}$ requires $\Omega\left(\frac{d}{\alpha}\right)$ shared samples. The lower bound holds even if the shared samples are labeled.*

Lower Bound for Infinite Littlestone Dimension As a first step, we show that $\frac{1}{\alpha}$ shared samples are necessary. This is essentially immediate from a corresponding lower bound for semi-privacy and the now standard replicable → private transformation of [GKM21, BGH+23], but we include the details for completeness.

Theorem 5.3. *Let $\mathcal{H}$ be any class with infinite Littlestone Dimension (for example, thresholds over $\mathbb{R}$). Then, any 0.0001-semi-replicable $(\alpha, 0.0001)$-learner for $\mathcal{H}$ requires $\Omega\left(\frac{1}{\alpha}\right)$ shared samples. The lower bound holds even when the shared samples are labeled.*

We will obtain our lower bound via a reduction to semi-private learning. We remark that our lower bound holds even when the shared samples are labeled, as is the case in semi-private learning, where lower bounds hold even when the public samples are labeled. In this setting, an algorithm A given m_{pub} public samples and m_{priv} private samples is (ε, δ)-semi-private if it is private with respect to the private samples. Formally, for any public data set $T \sim \mathcal{D}^{m_{\mathrm{pub}}}$ and any two neighboring datasets $S, S' \sim \mathcal{D}^{m_{\mathrm{priv}}}$ (see Definition 2.7) we have for any subset Y of the outputs,

$$\Pr(A(T; S) \in Y) \leqslant e^{\varepsilon} \Pr(A(T; S') \in Y) + \delta.$$

In particular, $A(T; \cdot)$ is private for every public dataset T. We say A is an $(\alpha, \beta, \varepsilon, \delta)$-agnostic semi-private learner if it is (ε, δ)-private and (α, β)-accurate. Lower bounds for semi-private learning follow from a data reduction lemma and lower bounds for private learning.

Lemma 5.4 (Lemma 4.4 of [ABM19])**.** *Let $0 < \alpha \leqslant 0.01, \varepsilon > 0, \delta > 0$. Suppose there is an $(\alpha, \frac{1}{18}, \varepsilon, \delta)$-agnostic semi-private learner for a hypothesis class $\mathcal{H}$ with private sample size n_{priv} and public sample size n_{pub}. Then, there is a $(100 n_{\mathrm{pub}}\alpha, \frac{1}{16}, \varepsilon, \delta)$-private learner that learns any distribution realizable by $\mathcal{H}$ with input sample size $\lceil \frac{n_{\mathrm{priv}}}{10 n_{\mathrm{pub}}} \rceil$.*

Theorem 5.5 (Theorem 1 of [ALMM19]). *Let $\mathcal{H}$ be a class with Littlestone Dimension n. Let A be a $(0.01, 0.01)$-accurate algorithm for $\mathcal{H}$ satisfying (ε, δ)-privacy with $\varepsilon = 0.1$ and $\delta = O\left(\frac{1}{m^2 \log m}\right)$. Then,*

$$m \geqslant \Omega(\log^* n).$$

In particular, any class that is privately learnable has finite Littlestone Dimension.

In particular, whenever $n_{\text{pub}} = o(\alpha^{-1})$, we must have $n_{\text{priv}} = \Omega\left(n_{\text{pub}} \log^* n\right)$. We show that any semi-replicable algorithm can be converted into a semi-private learner.

Lemma 5.6. *Let $\beta, \rho > 0$ be sufficiently small constants. Let A be a ρ-semi-replicable (α, β)-learner for $\mathcal{H}$ with sample complexity (m_u, m_s).*

Then, there is a $(\alpha, \beta \log(1/\beta), \varepsilon, \delta)$-semi-private learner for $\mathcal{H}$ using $m_{\text{pub}} = O\left(m_u \log(1/\beta)\right)$ public samples and $m_{\text{priv}} = O\left(m_s \left(\frac{\log(1/\delta)}{\varepsilon} + \log(1/\beta)\right) \log(1/\beta)\right)$ private samples.

Proof. Suppose A is a ρ-semi-replicable (α, β)-learner with sample complexity (m_u, m_s). We follow the framework of replicability-privacy reductions of [GKM21, BGH+23]. A key tool we require is private selection.

Theorem 5.7 (Private Selection [KKMN09, BNS16b, BDRS18]). *There exists some $c > 0$ such that for every $\varepsilon, \delta > 0$ and $m \in \mathbb{N}$, there is an (ε, δ)-private algorithm* PrivateSelection *that on input $S \in \mathcal{X}^m$, outputs with probability 1 an element $x \in \mathcal{X}$ that occurs in S at most $\frac{c \log(1/\delta)}{\varepsilon}$ times fewer than the true mode of S. Moreover, the algorithm runs in $\text{poly}(m, \log(|\mathcal{X}|))$-time.*

We now describe our algorithm.

Algorithm 12: SemiPrivateReduction$(A, \alpha, \beta, \varepsilon, \delta)$

Input : (α, β, ρ)-semi-replicable learner A, sample access to $\mathcal{D}$.
Parameters : (ε, δ) privacy parameters and (ε, δ)-accuracy parameters.
Output : $(\alpha, \beta, \varepsilon, \delta)$-semi-private leaner.

Set $k_1 \leftarrow O(\log(1/\beta))$ and $k_2 \leftarrow O\left(\frac{\log(1/\delta)}{\varepsilon} + \log(1/\beta)\right)$.
for $i \in [k_1]$ **do**
  Sample $T_i \sim \mathcal{D}^{m_u}$ and random string r_i.
  for $j \in [k_2]$ **do**
    Sample $S_{i,j} \sim \mathcal{D}^{m_s}$ using fresh samples and compute $y_{i,j} \leftarrow A((T_i, S_{i,j}); r_i)$.
return PrivateSelection$(\{y_{i,j}\}, \varepsilon, \delta)$.

The stated algorithm has public sample complexity $m_{\text{pub}} = O\left(k_1 m_u\right) = O\left(m_u \log(1/\beta)\right)$ and private sample complexity $m_{\text{priv}} = O\left(k_1 k_2 m_s\right) = O\left(m_s \left(\frac{\log(1/\delta)}{\varepsilon} + \log(1/\beta)\right) \log(1/\beta)\right)$.

We argue that the algorithm is semi-private. Note that $\{T_i\}$ consists of public samples and $\{S_{i,j}\}$ consists of private samples. Consider two neighboring datasets S, S' of private samples. Since fresh samples are drawn for each (i, j), note that there is exactly one index (i, j) where $S_{i,j}, S'_{i,j}$ are neighboring instead of identical. Then, $A((T_i, S_{i,j}), r_i)$ differ in exactly one run. We thus inherit (ε, δ)-differential privacy from the guarantees of Theorem 5.7.

We now argue correctness via two events (as in [BGH+23]). Consider the set of output hypotheses $\{y_{i,j}\}$. There exists a constant c such that the following two conditions hold:

1. With probability $1 - \frac{\beta}{2}$, some hypothesis appears $t_1 := 2c\left(\frac{\log(1/\delta)}{\varepsilon} + \log(1/\beta)\right)$ times.

2. With probability $1 - \frac{\beta \log(1/\beta)}{2}$, any element appearing $t_2 := c\left(\frac{\log(1/\delta)}{\varepsilon} + \log(1/\beta)\right)$ times is correct.

Towards the first item, note that

$$\mathbb{E}_{r,T\sim\mathcal{D}^{m_u}} \mathbb{E}_{S,S'\sim\mathcal{D}^{m_s}} \left[\mathbb{1}\left[A((T,S);r) \neq A((T,S');r)\right]\right] \leqslant \rho.$$

By Markov's inequality, with probability $\frac{1}{2}$, a randomly chosen (r,T) satisfy

$$\mathbb{E}_{S,S'\sim\mathcal{D}^{m_s}} \left[\mathbb{1}\left[A((T,S);r) \neq A((T,S');r)\right]\right] \leqslant 2\rho.$$

In particular, with probability $1 - \frac{\beta}{4}$, some r_i, T_i satisfy the desired condition. Now, by a Chernoff bound, we can guarantee that with probability at least $1 - \frac{\beta}{4}$, the canonical element corresponding to r_i, T_i appears at least t_1 times.

Towards the second item, since A is β-accurate, we observe that in expectation at most a β-fraction of hypotheses are incorrect. Then, Markov's inequality implies that at most $b := O\left(\frac{k_1 k_2}{\log(1/\beta)}\right) = O(k_2)$ hypotheses are incorrect with probability $1 - \beta\log(1/\beta)$. For a small enough (constant) choice of β, we have $b < t_2$, thus proving the claim.

Thus, the mode occurs at least t_1 times, and via **PrivateSelection** we select an element that occurs at least t_2 times, which we guarantee is correct. □

Proof of Theorem 5.3. Let β, ρ be sufficiently small constants. Suppose we have a ρ-semi-replicable (α,β)-learner for thresholds over $[n]$ with sample complexity (m_u, m_s). Then, via Lemma 5.6, we have a $(\alpha, \beta\log(1/\beta), \varepsilon, \delta)$-semi-private learner for thresholds over n with $m_{\text{pub}} = O\left(m_u \log(1/\beta)\right)$ public samples and $m_{\text{priv}} = O\left(m_s\left(\frac{\log(1/\delta)}{\varepsilon} + \log(1/\beta)\right)\log(1/\beta)\right)$ private samples. In particular, since β is small constant, applying Lemma 5.4 we obtain an $(100 m_{\text{pub}}\alpha, 0.01, \varepsilon, \delta)$-private learner using $m := O\left(\frac{m_{\text{priv}}}{m_{\text{pub}}}\right) = O\left(\frac{m_s}{m_u}\left(\frac{\log(1/\delta)}{\varepsilon} + \log(1/\beta)\right)\right)$ samples. Now, if $m_u = o\left(\frac{1}{\alpha}\right) = o\left(\frac{1}{\alpha\log(1/\beta)}\right)$, we obtain a $(0.01, 0.01, 0.1, \frac{1}{\text{poly}(m)})$-private learner (by setting $\varepsilon < 0.1$, $\delta < \frac{1}{\text{poly}(m_s)}$). Thus, Theorem 5.5 implies that

$$\frac{m_s}{m_u}\log(m_s) = \Omega\left(\log^* n\right).$$

Thus $m_s \log(m_s) = \Omega\left(\log^* n\right)$. In particular, any semi-private learner for $\mathcal{H}$ with infinite Littlestone dimension must have $m_u = \Omega\left(\frac{1}{\alpha}\right)$. □

Lower Bound for Large VC Dimension Towards proving Theorem 1.12, we fix a class $\mathcal{H}$ with VC dimension d and infinite Littlestone Dimension and show that any semi-private learner for $\mathcal{H}$ requires $\Omega\left(\frac{d}{\alpha}\right)$ public samples. This follows from a similar hardness amplification lemma as in [BDRS18].

First, we define the notion of an empirical learner and note that it suffices to prove hardness against empirical learners. This is similar to Lemma 5.8 of [BNSV15]. For a semi-private learner, we define the total complexity $m = m_{\text{priv}} + m_{\text{pub}}$ as the sum of the private and public sample complexities.

Definition 5.8. *Algorithm A is an (α,β)-accurate empirical learner for a hypothesis class $\mathcal{H}$ over $\mathcal{X}$ with (total) sample complexity m if for every $h \in \mathcal{H}$ and for every database $D = ((x_i, h(x_i)), ..., (x_m, h(x_m)))$, algorithm A outputs a hypothesis $\hat{h} \in \mathcal{H}$ satisfying $\Pr_A(\text{err}_D(\hat{h}) \leqslant \alpha) \geqslant 1 - \beta$. Here, $\text{err}_D(\hat{h}) = \frac{1}{m}\sum_{i=1}^m \mathbb{1}[h(x_i) \neq \hat{h}(x_i)]$.*

Lemma 5.9 (Lemma 5.8 of [BNSV15])**.** *Any (α,β)-accurate algorithm for empirically learning the class of thresholds with m samples is also a $(2\alpha, \beta+\beta')$-accurate PAC learner for thresholds when given at least $\max(m, 4\log(2/\beta')/\alpha)$ samples.*

We say an hypothesis h is α-consistent with dataset S if its empirical error $\mathrm{err}_S(h)$ is at most α.

Lemma 5.10. *Let $\alpha, \beta, \varepsilon, \delta > 0$. Let $\mathcal{H}$ be the class of thresholds over $[n]$, and assume there is a minimum element 0 such that $h(0) = 1$ for every $h \in \mathcal{H}$. Let $\mathcal{H}^{\wedge d} \subset \{[n]^d \to \{\pm 1\}\}$ be the class of conjunctions of d thresholds: the set of functions $(x_1, \ldots, x_d) \to \bigwedge_{i=1}^{d} h_i(x_i)$ where $h_i \in \mathcal{H}$. Then any $(\alpha, \beta, \varepsilon, \delta)$-semi-private learner for $\mathcal{H}^{\wedge d}$ with $o\left(\frac{d}{\alpha}\right)$ public examples requires $\Omega\left(d \log^* n\right)$ private examples.*

Proof. Suppose A_0 is an $(\alpha, \beta, \varepsilon, \delta)$-semi-private learner for $\mathcal{H}^d$ with m_{pub} public examples and m_{priv} private examples. We give an algorithm A_1 that is a semi-private learner for the class of thresholds.

Algorithm 13: SEMIPRIVATEHARDNESSAMPLIFICATION(A_0)

Input : $(\alpha, \beta, \varepsilon, \delta)$-semi-private learner A_0. Databases D_{pub} of n_{pub} public examples (denoted $(x_j^{(\mathrm{pub})}, y_j^{(\mathrm{pub})})$) and D_{priv} of n_{priv} private examples (denoted $(x_j^{(\mathrm{priv})}, y_j^{(\mathrm{priv})})$). Sample access to $\mathcal{D}$.

Parameters :

Output : Semi-private learner for thresholds.

Initiate S as an empty multiset.
Sample $r \in [d]$ uniformly.
For $i \in [d]$, let $f_i(x)$ map x to vector $v \in \mathbb{R}^d$ with $v[i] = x$ and $v[j] = 0$ for $j \neq i$.
for $i \in [d]$ **do**
  if $i = r$ **then**
    for $j \in [n_{\mathrm{pub}}]$ **do**
      Let $z_{(i,j)}^{(\mathrm{pub})} \leftarrow \left(f_r\left(x_j^{(\mathrm{pub})}\right), y_j^{(\mathrm{pub})}\right)$.
    for $j \in [n_{\mathrm{priv}}]$ **do**
      Let $z_{(i,j)}^{(\mathrm{priv})} \in \mathbb{R}^d$ with $z_{(i,j)}^{\prime(\mathrm{priv})}[i] = x_j^{(\mathrm{priv})}$ and $z_{(i,j)}^{(\mathrm{priv})}[k] = 0$. Add $(z_{(i,j)}^{(\mathrm{priv})}, y_j^{(\mathrm{priv})})$ to S.
  else
    Let $D'_{\mathrm{pub}}, D'_{\mathrm{priv}}$ denote a fresh sample from $\mathcal{D}$. Note that we take a fresh sample for each i.
    for $j \in [n_{\mathrm{pub}}]$ **do**
      Let $z_{(i,j)}^{\prime(\mathrm{pub})} \in \mathbb{R}^d$ with $z_{(i,j)}^{\prime(\mathrm{pub})}[i] = x_j^{\prime(\mathrm{pub})}$ and $z_{(i,j)}^{\prime(\mathrm{pub})}[k] = 0$. Add $(z_{(i,j)}^{\prime(\mathrm{pub})}, y_j^{\prime(\mathrm{pub})})$ to S.
    for $j \in [n_{\mathrm{priv}}]$ **do**
      Let $z_{(i,j)}^{\prime(\mathrm{priv})} \in \mathbb{R}^d$ with $z_{(i,j)}^{\prime(\mathrm{priv})}[i] = x_j^{\prime(\mathrm{priv})}$ and $z_{(i,j)}^{\prime(\mathrm{priv})}[k] = 0$. Add $(z_{(i,j)}^{\prime(\mathrm{priv})}, y_j^{(\mathrm{priv})})$ to S.
Compute $h_0 \leftarrow A_0(S)$
Return $h : x \mapsto h_0(f_r(x))$.

First, observe that SEMIPRIVATEHARDNESSAMPLIFICATION is (ε, δ)-semi-private. Note that a single change in D_{priv} leads to a single change in the multi-set S. Since A_0 is (ε, δ)-semi-private, so is the output h_0. Finally, the output h, and therefore SEMIPRIVATEHARDNESSAMPLIFICATION, is (ε, δ)-semi-private by post-processing.

We now argue the correctness of SEMIPRIVATEHARDNESSAMPLIFICATION. Consider the execution of SEMIPRIVATEHARDNESSAMPLIFICATION on databases $D_{\mathrm{pub}}, D_{\mathrm{priv}}$ of size $n_{\mathrm{pub}}, n_{\mathrm{priv}}$, sampled from $\mathcal{D}$. First, we argue that A_0 is applied on a multiset S labeled by some hypothesis $h \in \mathcal{H}^{\wedge d}$. For each i, let $D_{\mathrm{pub}}^{(i)}, D_{\mathrm{priv}}^{(i)}$ denote the datasets sampled from $\mathcal{D}$, which are labeled by some hypothesis $h_i \in \mathcal{H}$. Define $h : x \to$

$\bigwedge_{i=1}^{d} h_i(x_i)$. Then, for all examples $(x_j, y_j) \in D_{\text{priv}}, D_{\text{pub}}$,

$$h(x) = h_1(0) \wedge \ldots \wedge h_{i-1}(0) \wedge h_i(x_j) \wedge h_{i+1}(0) \wedge \ldots \wedge h_d(0) = h_i(x_j) = y_j$$

as desired. Then, A_0 with probability at least $1 - \beta$ finds an α-consistent hypothesis h_0 with respect to S. We now analyze the error of A_0. Let $\mathcal{D}_i$ denote the distribution on databases of size n obtained by applying f_i to each example in $D_{\text{pub}}, D_{\text{priv}} \sim \mathcal{D}$. Note that $S = \bigcup_{i=1}^{d} S_i$ where $S_i \sim \mathcal{D}_i$. Then, we may write the empirical error of h_0 as

$$\alpha \geqslant \text{err}_S(h) = \frac{1}{d} \sum_{i=1}^{d} \text{err}_{S_i}(h_0).$$

Then, there are at most βd indices on which $\text{err}_{S_i}(h_0) \geqslant \frac{\alpha}{\beta}$. Since r is uniformly chosen at random, and the points on every axis are distributed in exactly the same way, the probability that r is one of the βd indices occurs with probability at most β. Thus, with probability $1 - 2\beta$ we have that $\text{err}_{S_r}(h_0) \leqslant \frac{\alpha}{\beta}$. Finally, we bound the error of h on $D_{\text{pub}}, D_{\text{priv}}$. Note that for any point (x_j, y_j) (either a public or private sample),

$$h(x_j) = h_0(f_r(x_j)) = h_0(z_{r,j})$$

so that $\text{err}_D(h) = \text{err}_{S_r}(h_0) \leqslant \frac{\alpha}{\beta}$.

In particular, we obtain a $\left(\frac{\alpha}{\beta}, 2\beta, \varepsilon, \delta\right)$-semi-private empirical learner for thresholds with $\frac{m_{\text{pub}}}{d}$ public samples and $\frac{m_{\text{priv}}}{d}$ private samples. Applying Lemma 5.9, we obtain a $\left(\frac{2\alpha}{\beta}, 3\beta, \varepsilon, \delta\right)$-semi-private learner for thresholds with $\frac{m_{\text{pub}}}{d}$ public samples and $\frac{m_{\text{priv}}}{d} + \frac{4\log(2/\beta)}{\alpha}$ private samples. Suppose that $m_{\text{pub}} = o\left(\frac{d}{\alpha}\right)$, so that we obtain a semi-private learner with $o\left(\frac{1}{\alpha}\right)$ public examples. Then, from Lemma 5.4 and Theorem 5.5 we have that

$$\frac{m_{\text{priv}}}{d} + \frac{4\log(2/\beta)}{\alpha} \geqslant \log^* n.$$

In particular, for sufficiently large n, we have $m_{\text{priv}} = \Omega\left(d \log^* n\right)$, as desired. □

Finally, we are ready to prove Theorem 1.12.

Proof of Theorem 1.12. Let β, ρ be sufficiently small constants. Consider the class of thresholds over $[n]^d$, $\mathcal{H}^{\wedge d}$. Suppose we have a ρ-replicable (α, β)-semi-replicable learner for $\mathcal{H}^{\wedge d}$ with sample complexity (m_u, m_s). Applying Lemma 5.6, we obtain a $(\alpha, \beta\log(1/\beta), \varepsilon, \delta)$-semi-private learner for $\mathcal{H}^{\wedge d}$ with $O\left(m_u \log(1/\beta)\right)$ public samples and $O\left(m_s\left(\frac{\log(1/\delta)}{\varepsilon} + \log(1/\beta)\right)\log(1/\beta)\right)$ private samples. Finally, from Lemma 4.23, we may conclude that if $m_u = o\left(\frac{1}{\alpha\log(1/\beta)}\right) = o\left(\frac{1}{\alpha}\right)$, we must have $m_s \log m_s = \Omega\left(d \log^* n\right)$. Again, this implies that over an infinite domain, thresholds over $\mathbb{R}$ require $\Omega\left(\frac{d}{\alpha}\right)$ shared sample complexity. □

5.2 Lower Bound for Supervised Sample Complexity

Finally, we give our near-matching lower bound on the supervised sample complexity.

Theorem 1.11. *Any ρ-semi-replicable $(\alpha, 0.0001)$-learner for any class $\mathcal{H}$ with VC dimension d requires sample complexity $\Omega\left(\frac{d^2}{\rho^2\alpha^2\log^6 d}\right)$. This bound holds regardless of unlabeled sample complexity.*

Our lower bound, will follow from a reduction to the sign-one-way marginals problem.

Definition 5.11 (Sign-One-Way Marginals)**.** *Let $\mathcal{D}_p$ be a product of d Rademacher distributions with expectations $p = (p_1, \ldots, p_d)$. A vector $v \in \{\pm 1\}^d$ is an α-accurate solution to the sign-one-way marginals problem for $\mathcal{D}_p$ if $\frac{1}{d}\sum_i^d v_i p_i \geqslant \frac{1}{d}\sum_{i=1}^d |p_i| - \alpha$.*

An algorithm (α, β)-accurately solves the sign-one-way marginals problem with sample complexity m if given any product of d Rademacher distributions $\mathcal{D}$ and m i.i.d. samples from $\mathcal{D}$, with probability at least $1 - \beta$ the algorithm outputs an α-accurate solution to the sign-one-way marginals problem for $\mathcal{D}$.

[BGH+23] show that any 0.0005-replicable algorithm $(0.02, 0.002)$-accurately solving the sign-one-way marginals problem requires sample complexity $\tilde{\Omega}(d)$. [HLYY25] showed that any ρ-replicable $(\alpha, 0.001)$-accurate algorithm requires $\tilde{\Omega}(d\rho^{-2}\alpha^{-2})$ samples.

Theorem 5.12 ([HLYY25])**.** *Let $\alpha < 0.001$ be small constants. Any ρ-replicable algorithm $(\alpha, 0.001)$-accurately solving the sign-one-way marginals problem requires sample complexity $\Omega\left(\frac{d}{\rho^2\alpha^2\log^6 d}\right)$.*

We now present a lower bound for semi-supervised learning given Theorem 5.12. Let $\mathcal{H}$ be a VC class of dimension d. Consider a set of d shattered elements $\{x_1, \ldots, x_d\}$. We provide a generic lemma that says any algorithm that PAC learns $\mathcal{H}$ with respect to these distributions solves the sign-one-way marginals problem.

Proof of Theorem 1.11. Assume without loss of generality $m = \Omega(d \log d)$.[3] Consider a set of d shattered elements $\{x_1, \ldots, x_d\}$. Suppose A is a ρ-replicable $(\alpha, 0.0001)$-semi-supervised learner for $\mathcal{H}$ with sample complexity m. We design a ρ-replicable $(2\alpha, 0.0002)$-accurate learner for the sign-one-way marginals problem with sample complexity $O(md)$.

Suppose we receive m' samples from $\mathcal{D}_p$ for some expectation vector p. We let the distribution $\mathcal{D}$ over $\mathcal{X}$ be uniform. Thus, whenever A demands an unsupervised sample, we can easily generate one uniformly from $\mathcal{D}$.

Whenever a supervised sample is required by A, we uniformly sample $x_i \in \mathcal{X}$ and label it according to a fresh sample from the i-th coordinate of $\mathcal{D}_p$. Note that with m samples from $\mathcal{D}$, any element of $\mathcal{X}$ is sampled $\frac{m}{d}$ times in expectation, and more than Cm/d times for some large constant C with probability at most $\exp(-\Omega(m/d)) < \frac{0.0001}{d}$ since we have assumed $m = \Omega(d \log d)$. By a union bound, no element is sampled more than Cm/d times with probability at least $1 - 0.0001$. We condition on this event. In particular, we as long as we have $m' = O(m/d)$ samples from $\mathcal{D}_p$, we can successfully label every sampled point from the domain $\mathcal{D}$. Let S denote the m uniform samples from $\mathcal{X}$ labeled with the corresponding coordinates from $\mathcal{D}_p$. Note that also the samples of S are independent.

Thus, let $f \leftarrow A(S)$ be the output of the semi-supervised learner. We condition on the event $\mathrm{err}_{\mathcal{D}} f \leqslant \mathrm{err}_{\mathcal{D}}(f_{\mathrm{OPT}}) + \alpha$, which fails with probability at most 0.0001 and output the vector v with $v_i = f(x_i)$ for the sign-one-way marginals problem. It is clear that any output hypothesis f has error

$$\frac{1}{d}\left(\sum_{f(x_i)=+1}\frac{1-p_i}{2} + \sum_{f(x_i)=-1}\frac{1+p_i}{2}\right) = \frac{1}{d}\left(\sum_{i=1}^d \frac{1-|p_i|}{2} + \sum_{f(x_i)\neq \mathrm{sign}(p_i)} |p_i|\right).$$

Thus, the optimal hypothesis is $f_{\mathrm{OPT}}(x_i) = \mathrm{sign}(p_i)$ and our hypothesis satisfies

$$\frac{1}{d}\sum_{f(x_i)\neq \mathrm{sign}(p_i)} |p_i| = \frac{1}{d}\sum_{v_i \neq \mathrm{sign}(p_i)} |p_i| \leqslant \alpha.$$

[3] We will show that whenever $m = \Omega(d \log d)$ that $m = \Omega\left(\frac{d^2}{\rho^2\alpha^2\log^6 d}\right)$. Any algorithm on $O(d \log d)$ samples implies one between $O(d \log d)$ and $O\left(\frac{d^2}{\rho^2\alpha^2\log^6 d}\right)$, giving a contradiction for large enough d.

Thus, the vector v satisfies

$$\frac{1}{d}\sum_{i=1}^{d}|p_i| - v_i p_i = \frac{1}{d}\sum_{v_i \neq \text{sign}(p_i)} 2|p_i| \leqslant 2\alpha.$$

In particular, we obtain a $(2\alpha, 0.0002)$-accurate learner for the sign-one-way marginals problem. Furthermore, our algorithm for the sign-one-way marginals problem is ρ-replicable since the semi-supervised learner A is ρ-replicable. From Theorem 5.12, we know that $m = \Omega\left(\frac{d}{\rho^2\alpha^2\log^6 d}\right)$ so that $m' = \Omega\left(\frac{d^2}{\rho^2\alpha^2\log^6 d}\right)$. □

Acknowledgements

We would like to thank Arsen Vasilyan for many fruitful discussions throughout the development of this work both regarding approaches to improving the sample complexity of our algorithms, as well as toward designing proper approximately replicable learners. We also thank Mark Bun, Rex Lei, Satchit Sivakumar, and Shay Moran for helpful conversations regarding relaxed notions of replicability and their connection to differential privacy.

References

[AAC+25] Anders Aamand, Maryam Aliakbarpour, Justin Y Chen, Shyam Narayanan, and Sandeep Silwal. On the structure of replicable hypothesis testers. *arXiv preprint arXiv:2507.02842*, 2025. 12

[ABM19] Noga Alon, Raef Bassily, and Shay Moran. Limits of private learning with access to public data. In *Advances in Neural Information Processing Systems 32*, pages 10342–10352, 2019. 4, 10, 12, 39, 40

[ALMM19] Noga Alon, Roi Livni, Maryanthe Malliaris, and Shay Moran. Private pac learning implies finite littlestone dimension. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 852–860, 2019. 4, 41

[BBD+24] Adam Block, Mark Bun, Rathin Desai, Abhishek Shetty, and Steven Z Wu. Oracle-efficient differentially private learning with public data. *Advances in Neural Information Processing Systems*, 37:113191–113233, 2024. 12

[BCM+20] Raef Bassily, Albert Cheu, Shay Moran, Aleksandar Nikolov, Jonathan Ullman, and Steven Wu. Private query release assisted by public data. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pages 695–703, 2020. 12

[BDBC+23] Shai Ben-David, Alex Bie, Clément L Canonne, Gautam Kamath, and Vikrant Singhal. Private distribution learning with public data: The view from sample compression. *Advances in Neural Information Processing Systems*, 36:7184–7215, 2023. 12

[BDPSS09] Shai Ben-David, Dávid Pál, and Shai Shalev-Shwartz. Agnostic online learning. In *COLT*, volume 3, page 1, 2009. 14

[BDRS18] Mark Bun, Cynthia Dwork, Guy N. Rothblum, and Thomas Steinke. Composable and versatile privacy via truncated CDP. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC*, pages 74–86. ACM, 2018. 41, 42

[BE02] Olivier Bousquet and André Elisseeff. Stability and generalization. *The Journal of Machine Learning Research*, 2:499–526, 2002. 12

[BEHW89] Anselm Blumer, Andrzej Ehrenfeucht, David Haussler, and Manfred K Warmuth. Learnability and the vapnik-chervonenkis dimension. *Journal of the ACM (JACM)*, 36(4):929–965, 1989. 2, 13

[BGH+23] Mark Bun, Marco Gaboardi, Max Hopkins, Russell Impagliazzo, Rex Lei, Toniann Pitassi, Jessica Sorrell, and Satchit Sivakumar. Stability is stable: Connections between replicability, privacy, and adaptive generalization. *arXiv preprint arXiv:2303.12921*, 2023. 1, 2, 4, 10, 11, 12, 39, 40, 41, 45

[BI91] Gyora M Benedek and Alon Itai. Learnability with respect to fixed distributions. *Theoretical Computer Science*, 86(2):377–389, 1991. 2

[BKS22] Alex Bie, Gautam Kamath, and Vikrant Singhal. Private estimation with public data. *Advances in neural information processing systems*, 35:18653–18666, 2022. 12

[BNS16a] Amos Beimel, Kobbi Nissim, and Uri Stemmer. Private learning and sanitization: Pure vs. approximate differential privacy. *Theory Comput.*, 12(1):1–61, 2016. 2, 12

[BNS16b] Mark Bun, Kobbi Nissim, and Uri Stemmer. Simultaneous private learning of multiple concepts. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*, pages 369–380, 2016. 41

[BNSV15] Mark Bun, Kobbi Nissim, Uri Stemmer, and Salil P. Vadhan. Differentially private release and learning of threshold functions. In *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS*, pages 634–649. IEEE Computer Society, 2015. 42

[BTT18] Raef Bassily, Om Thakkar, and Abhradeep Thakurta. Model-agnostic private learning via stability. *arXiv preprint arXiv:1803.05101*, 2018. 12

[CCMY24] Zachary Chase, Bogdan Chornomaz, Shay Moran, and Amir Yehudayoff. Local borsuk-ulam, stability, and replicability. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 1769–1780, 2024. 12

[CMY23] Zachary Chase, Shay Moran, and Amir Yehudayoff. Replicability and stability in learning. *arXiv preprint arXiv:2304.03757*, 2023. 1, 5, 6, 12

[DF18] Cynthia Dwork and Vitaly Feldman. Privacy-preserving prediction. In *Conference On Learning Theory*, pages 1693–1702. PMLR, 2018. 1, 12

[DF20] Yuval Dagan and Vitaly Feldman. Pac learning with stable and private predictions. In *Conference on Learning Theory*, pages 1389–1410. PMLR, 2020. 12

[DGK+25] Ilias Diakonikolas, Jingyi Gao, Daniel Kane, Sihan Liu, and Christopher Ye. Replicable distribution testing. *arXiv preprint arXiv:2507.02814*, 2025. 12

[DK16] Ilias Diakonikolas and Daniel M Kane. A new approach for testing properties of discrete distributions. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 685–694. IEEE, 2016. 51

[DKW56] Aryeh Dvoretzky, Jack Kiefer, and Jacob Wolfowitz. Asymptotic minimax character of the sample distribution function and of the classical multinomial estimator. *The Annals of Mathematical Statistics*, pages 642–669, 1956. 29

[DPWV23] Peter Dixon, A Pavan, Jason Vander Woude, and NV Vinodchandran. List and certificate complexities in replicable learning. *arXiv preprint arXiv:2304.02240*, 2023. 6, 12

[EHK+25] Eric Eaton, Marcel Hussing, Michael Kearns, Aaron Roth, Sikata Bela Sengupta, and Jessica Sorrell. Replicable reinforcement learning with linear function approximation. *arXiv preprint arXiv:2509.08660*, 2025. 12

[EHKS23] Eric Eaton, Marcel Hussing, Michael Kearns, and Jessica Sorrell. Replicable reinforcement learning. *Advances in Neural Information Processing Systems*, 36:15172–15185, 2023. 12

[EKM+23] Hossein Esfandiari, Amin Karbasi, Vahab Mirrokni, Grigoris Velegkas, and Felix Zhou. Replicable clustering. *Advances in Neural Information Processing Systems*, 36:39277–39320, 2023. 12

[GKM21] Badih Ghazi, Ravi Kumar, and Pasin Manurangsi. User-level differentially private learning via correlated sampling. In *Advances in Neural Information Processing Systems 34*, pages 20172–20184, 2021. 5, 7, 10, 12, 20, 26, 40, 41

[Han16] Steve Hanneke. The optimal sample complexity of pac learning. *The Journal of Machine Learning Research*, 17(1):1319–1333, 2016. 13

[Hau92] David Haussler. Decision theoretic generalizations of the pac model for neural net and other learning applications. *Information and computation*, 100(1):78–150, 1992. 13

[HIK+24] Max Hopkins, Russell Impagliazzo, Daniel Kane, Sihan Liu, and Christopher Ye. Replicability in high dimensional statistics. *arXiv preprint arXiv:2406.02628*, 2024. 1, 11, 12, 19, 37, 51

[HKLM22] Max Hopkins, Daniel M. Kane, Shachar Lovett, and Gaurav Mahajan. Realizable learning is all you need. In *Conference on Learning Theory (COLT)*, Proceedings of Machine Learning Research. PMLR, 2022. 10, 39

[HLYY25] Max Hopkins, Sihan Liu, Christopher Ye, and Yuichi Yoshida. From generative to episodic: Sample-efficient replicable reinforcement learning. *arXiv preprint arXiv:2507.11926*, 2025. 5, 7, 11, 12, 20, 26, 45

[HM25] Max Hopkins and Shay Moran. The role of randomness in stability. *arXiv preprint arXiv:2502.08007*, 2025. 12

[ILPS22] Russell Impagliazzo, Rex Lei, Toniann Pitassi, and Jessica Sorrell. Reproducibility in learning. In *54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 818–831. ACM, 2022. 1, 2, 5, 6, 12, 13, 16, 17, 19, 37

[KKL+24] Alkis Kalavasis, Amin Karbasi, Kasper Green Larsen, Grigoris Velegkas, and Felix Zhou. Replicable learning of large-margin halfspaces. *arXiv preprint arXiv:2402.13857*, 2024. 12

[KKMN09] Aleksandra Korolova, Krishnaram Kenthapadi, Nina Mishra, and Alexandros Ntoulas. Releasing search queries and clicks privately. In *Proceedings of the 18th international conference on World wide web*, pages 171–180, 2009. 41

[KKMV23] Alkis Kalavasis, Amin Karbasi, Shay Moran, and Grigoris Velegkas. Statistical indistinguishability of learning algorithms. Personal communication, January 2023. 12

[KKVZ24] Alkis Kalavasis, Amin Karbasi, Grigoris Velegkas, and Felix Zhou. On the computational landscape of replicable learning. *Advances in Neural Information Processing Systems*, 37:105887–105927, 2024. 12

[KMR+24] Walid Krichene, Nicolas E Mayoraz, Steffen Rendle, Shuang Song, Abhradeep Thakurta, and Li Zhang. Private learning with public features. In *International Conference on Artificial Intelligence and Statistics*, pages 4150–4158. PMLR, 2024. 12

[KVYZ23] Amin Karbasi, Grigoris Velegkas, Lin Yang, and Felix Zhou. Replicability in reinforcement learning. *Advances in Neural Information Processing Systems*, 36:74702–74735, 2023. 12

[Lar23] Kasper Green Larsen. Bagging is an optimal pac learner. In *The Thirty Sixth Annual Conference on Learning Theory*, pages 450–468. PMLR, 2023. 13

[Lit88] Nick Littlestone. Learning quickly when irrelevant attributes abound: A new linear-threshold algorithm. *Machine learning*, 2(4):285–318, 1988. 14

[LMS25] Kasper Green Larsen, Markus Engelund Mathiasen, and Clement Svendsen. Improved replicable boosting with majority-of-majorities. *arXiv preprint arXiv:2501.18388*, 2025. 12

[LVS+21] Terrance Liu, Giuseppe Vietri, Thomas Steinke, Jonathan Ullman, and Steven Wu. Leveraging public data for practical private query release. In *International Conference on Machine Learning*, pages 6968–6977. PMLR, 2021. 12

[LY24] Sihan Liu and Christopher Ye. Replicable uniformity testing. *Advances in Neural Information Processing Systems*, 37:32039–32075, 2024. 12

[Mas90] Pascal Massart. The tight constant in the dvoretzky-kiefer-wolfowitz inequality. *The annals of Probability*, pages 1269–1283, 1990. 29

[MSS23] Shay Moran, Hilla Schefler, and Jonathan Shafer. The bayesian stability zoo. *Advances in Neural Information Processing Systems*, 36:61725–61746, 2023. 12

[MT07] Frank McSherry and Kunal Talwar. Mechanism design via differential privacy. In *Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, FOCS '07, page 94–103, USA, 2007. IEEE Computer Society. 26

[NNSY23] Moni Naor, Kobbi Nissim, Uri Stemmer, and Chao Yan. Private everlasting prediction. *Advances in Neural Information Processing Systems*, 36:54785–54804, 2023. 12

[RY20] Anup Rao and Amir Yehudayoff. *Communication Complexity: and Applications*. Cambridge University Press, 2020. 13

[Ste24] Uri Stemmer. Private truly-everlasting robust-prediction. *arXiv preprint arXiv:2401.04311*, 2024. 12

[UMB+24] Enayat Ullah, Michael Menart, Raef Bassily, Cristóbal Guzmán, and Raman Arora. Public-data assisted private stochastic optimization: Power and limitations. *Advances in Neural Information Processing Systems*, 37:20383–20427, 2024. 12

[Val84] Leslie G Valiant. A theory of the learnable. In *Proceedings of the sixteenth annual ACM symposium on Theory of computing*, pages 436–445. ACM, 1984. 2, 13

[VC74] Vladimir Vapnik and Alexey Chervonenkis. Theory of pattern recognition, 1974. 2

[vdMH20] Laurens van der Maaten and Awni Hannun. The trade-offs of private prediction. *arXiv preprint arXiv:2007.05089*, 2020. 12

[VWDP+24] Jason Vander Woude, Peter Dixon, Aduri Pavan, Jamie Radcliffe, and NV Vinodchandran. Replicability in learning: Geometric partitions and kkm-sperner lemma. *Advances in Neural Information Processing Systems*, 37:78996–79028, 2024. 12

A Omitted Proofs

We give the omitted proof of Theorem 3.6. The proof is a simple modification of the lower bound from [HIK+24].

Proof of Theorem 3.6. We will construct a meta-distribution $\mathcal{M}$ over biases $p \in [-1, 1]$ such that any ρ-replicable algorithm with error probability less than 0.01 with respect to $\mathcal{M}$ requires $\Omega\left(\frac{1}{\rho^2}\right)$ samples. Let A be a ρ-replicable algorithm for $\{\pm 1\}$-bias estimation with succeeds with probability 0.9 on m samples. First, following [HIK+24], we fix a random string r such that $A(;r)$ has error probability at most 0.04 and is 4ρ-replicable with respect to $\mathcal{M}$. For any bias p, let $\mathrm{acc}(p) := \Pr_{S\sim \mathrm{Rad}(p)^m}(A(S;r) = 1)$ denote the acceptance probability of A given samples from $\mathrm{Rad}(p)$. Then, since $\mathrm{acc}(p)$ is a continuous function (see Claim 3.10 of [HIK+24]) there exists p^* such that $\mathrm{acc}(p^*) = \frac{1}{2}$.

Our goal is now to prove that there is in fact a (sufficiently wide) interval I^* where $\mathrm{acc}(p) \in (1/3, 2/3)$. To do so, we follow the mutual information framework of [HIK+24]. First, we need a standard fact that says any correct algorithm must see samples correlated with the answer (see e.g. [DK16]).

Claim A.1. *Let $X \sim \mathrm{Rad}(0)$ and A be a random variable possibly correlated with X. If there exists a (randomized) function f so that $f(A) = X$ with at least 51% probability, then $I(X : A) \geqslant 2 \cdot 10^{-4}$.*

Next, we argue that samples from coins with similar bias are indistinguishable.

Claim A.2 (Claim 3.9 of [HIK+24])**.** *Let $m \geqslant 0$ be an integer and $-1 \leqslant a < b \leqslant 1$. Let $X \sim \mathrm{Rad}(0)$ and Y be distributed according to $\mathrm{Rad}(a)^m$ if $X = 1$ and $\mathrm{Rad}(b)^m$ if $X = -1$. Then,*

$$I(X : Y) = O\left(\frac{m(b-a)^2}{\min\left(1+a, 1+b, 1-a, 1-b\right)}\right).$$

Assume without loss of generality that $p^* \leqslant 0$. Our goal is to find an interval $I^* = [p^*, q^*]$ such that $|\mathrm{acc}(p^*) - \mathrm{acc}(q^*)| \leqslant 0.1$. Consider the following experiment: let X, Y be distributed as in Claim A.2 where $a = p^*, b = q^*$. Suppose $|\mathrm{acc}(p^*) - \mathrm{acc}(q^*)| \geqslant 0.1$. Then, for some universal constant C', there is a $C'm$ sample algorithm (that runs $A(;r)$ on C' fresh samples) which successfully guesses X with 51% probability. We will choose q^* such that no $C'm$ sample algorithm can guess X with 51% probability, so that $|\mathrm{acc}(p^*) - \mathrm{acc}(q^*)| \geqslant 0.1$. We proceed with case analysis on p^*.

1. $p^* = -1$. Note that $Y \sim \mathrm{Rad}(p^*)^{C'm}$ always yields -1 samples, while $Y \sim \mathrm{Rad}(q^*)^{C'm}$ yields all -1 samples with probability $\left(1 - \frac{q^*+1}{2}\right)^m \geqslant \exp\left(-C'm(q^*+1)\right) \geqslant 0.999$ for $q^* \leqslant -1 + \frac{1}{C'm}$ for some sufficient constant factor.

2. $p^* > 0$. From Claim A.2, we have

$$I(X : Y) = O\left(\frac{m(q^*-p^*)^2}{1+p^*}\right) < 2 \cdot 10^{-4}$$

   if $q^* \leqslant p^* + c\sqrt{\frac{1+p^*}{m}}$ for some sufficiently small constant $c > 0$.

Then, consider $\mathcal{M}$ that samples uniformly from the following set of points defined $p_0 = -1, p_1 = \frac{c}{m} - 1$ and $p_i = p_i + c\sqrt{\frac{1+p_i}{m}}$ for sufficiently small constant c until $p_k > 0$ for some integer k. By symmetry, add all points $-p_i$ as well. Observe that we have thus obtained $O(\sqrt{m})$ points B that intersect any interval I^*. In particular, for any random string r, $A(;r)$ is not 0.1-replicable on the $\mathrm{Rad}(p)$ for $p \in I^* \cap B$. Then, if we define $\mathcal{M}$ to pick a bias uniformly from B, $A(;r)$ is not $\frac{c}{\sqrt{m}}$-replicable with respect to $\mathcal{M}$. Since we assumed $A(;r)$ is 4ρ-replicable, we obtain $m = \Omega\left(\frac{1}{\rho^2}\right)$. □