Refereed Learning

Ran Canetti* Ephraim Linder* Connor Wagaman*

October 6, 2025

Abstract

We initiate an investigation of learning tasks in a setting where the learner is given access to two competing provers, only one of which is honest. Specifically, we consider the power of such learners in assessing purported properties of opaque models. Following prior work that considers the power of competing provers in different settings, we call this setting *refereed learning*.

After formulating a general definition of refereed learning tasks, we show refereed learning protocols that obtain a level of accuracy that far exceeds what is obtainable at comparable cost without provers, or even with a single prover. We concentrate on the task of choosing the better one out of two *black-box* models, with respect to some ground truth. While we consider a range of parameters, perhaps our most notable result is in the high-precision range: For all $\varepsilon > 0$ and ambient dimension d, our learner makes only one query to the ground truth function, communicates only $(1 + \frac{1}{\varepsilon^2}) \cdot \mathrm{poly}(d)$ bits with the provers, and outputs a model whose loss is within a multiplicative factor of $(1 + \varepsilon)$ of the best model's loss. Obtaining comparable loss with a *single* prover would require the learner to access the ground truth at almost all of the points in the domain. To obtain this bound, we develop a technique that allows the learner to sample, using the provers, from a distribution that is not efficiently samplable to begin with. We find this technique to be of independent interest.

We also present lower bounds that demonstrate the optimality of our protocols in a number of respects, including prover complexity, number of samples, and need for query access.

*Boston University, {canetti,ejlinder,wagaman}@bu.edu.

Contents

1 Introduction

Modern machine learning tasks require increasingly large amounts of data and computational power. As a result, model training has shifted from a task that almost anyone can perform on their own to a task that requires the assistance of external agents that are resource-abundant and have better access to the underlying data. Furthermore, one is often presented with opaque models that purport to approximate some ground truth, but are not accompanied with a rigorous or trustworthy performance guarantee. Moreover, such models are often given only as black-boxes via a controlled query/response mechanism, and without disclosing their oft-extensive training processes.

This state of affairs naturally raises the need to verify claims of performance of such models, without fully trusting the parties making the claims, and with significantly fewer resources than those needed to train comparable models to begin with. Verifying such claims appears hard: There is a rich literature on efficient verification of *computations* performed by powerful-but-untrusted parties (e.g., [Kil92, Mic94, GKR15]; see [WB15] for a survey); however, such mechanisms appear to be ill-suited for the task of verifying the performance of even explicitly described ML models, let alone opaque ones. There is also a growing body of work on using powerful-but-untrusted intermediaries to learn properties of unknown ground-truth functions (e.g., [GRSY21, CK21]), as well as efficient verification of claims made by powerful provers regarding properties of huge combinatorial objects in general [EKR04, RVW13]. However, these works mostly focus on verifiable execution of a specific learning (or property testing) algorithm, rather than evaluating the performance of a given, opaque model without knowledge of the process used to train it. Perhaps closest to our task are the works of Herman and Rothblum [HR22, HR24b, HR24a] that allow verifying claims about properties of *distributions* that are accessible only via obtaining samples. However, this is still a far cry from assessing properties of black-box models.

Some natural properties of black-box models can of course be assessed using standard methods. For instance, the loss of a given model w.r.t. some sample distribution and loss function can be approximated by computing the empirical loss on a large enough sample. However, this method can be prohibitively costly both in samples from the ground truth and in queries to the model. One can use a technique from [GRSY21] to push much of the burden to an external powerful-but-untrusted prover, but even this method incurs high cost: to obtain an *additive* bound of η on the error, the learner needs to both communicate η^{-2} unlabeled samples to the prover and have the prover query and report the ground truth values at these points, and obtain η^{-1} labeled samples which are hidden from the prover in order to verify the prover's responses.

We would like to do better—in terms of the error, in terms of the access to the ground truth, and in terms of the communication with the prover. However, as evidenced by lower bounds proven in [GRSY21], this appears hard—at least within the present framing of the problem.

1.1 This work

We show that the quality of black-box models can be assessed with significantly better accuracy and with significantly lower cost, if the learner can interact with *two powerful and competing provers,* one of which is honest. The provers' power can be manifested either in terms of their computational power, or in access to the ground truth, or in knowledge of the models, or any combination of these. This model can be naturally viewed as an extension of the *refereed delegation of computation* model [FST88, FK97, CRR11, CRR13, KR14] to our setting. Following the cue of these works, we coin the term refereed learning to denote the type of learning performed in this model. (See Section 1.2

for more discussion on refereed delegation and other related works.)

We first define refereed learning for a general setting. Next we focus on applying refereed learning to the following specific task, where we showcase the power of the refereed learning framework via concrete protocols. The learner-verifier[1] is presented with *two* candidate models that purport to compute the same ground truth, and is tasked with choosing the model that incurs the smaller overall loss with respect to some sample distribution and loss function. (The restriction to two candidates is not essential, but it helps make the model more concrete. It also facilitates envisioning each one of the provers as "trying to promote" a different one of the two models.)

The salient parameters we consider are (a) the overall loss of the output model relative to the better of the two competing ones; (b) the number of learner queries to the ground truth and samples from it (both labeled and unlabeled ones); (c) the number of learner queries to the candidate models; (d) the computational complexity of the learner; (e) the computational complexity and query and sample complexity of the provers. We first sketch and briefly discuss definitions of refereed learning, then present our results, and finally discuss the new tools we develop to obtain these results.

1.1.1 Defining refereed learning

We start with general refereed learning. Consider a learner-verifier $\mathcal{V}$ and two provers $\mathcal{P}_0, \mathcal{P}_1$ that are presented with a ground truth function $f : \{0,1\}^d \to \mathcal{Y}$, a distribution $\mathcal{D}$ over $\{0,1\}^d$, and k models (a.k.a. hypotheses) $h_1, \ldots, h_k \in \mathcal{H}$, where $\mathcal{H}$ is some family of hypotheses. We will typically assume that $h_1, \ldots, h_k$ and f are accessed via queries and, depending on the setting, that $\mathcal{D}$ is accessed either via samples or via queries to its probability mass function, $Q_{\mathcal{D}}$, which maps $x \mapsto \Pr_{X\sim\mathcal{D}}[X = x]$. To measure the learner's performance we use a scoring function $\mathcal{S}$ (which assigns a score to each potential output of the learner, with respect to some sample distribution $\mathcal{D}$, hypotheses $h_1, \ldots, h_k$ and function f) and a target function $\mathcal{T}$.

Definition 1.1 (**Refereed learning, general case (informal)**). *A protocol $[\mathcal{P}_0, \mathcal{P}_1, \mathcal{V}]$ is an (α, η, β)-*refereed learning *protocol, with respect to a family $\mathcal{H}$ of hypotheses, a scoring function $\mathcal{S}$ and target function $\mathcal{T}$, if for all $b \in \{0,1\}$, $h_1, \ldots, h_k \in \mathcal{H}$, and $\mathcal{P}^*_{1-b}$, the learner output ρ satisfies*

$$\Pr[\mathcal{S}(\rho, \mathcal{D}, f, h_1, \ldots, h_k) \leq \alpha\mathcal{T}(\mathcal{D}, f, h_1, \ldots, h_k) + \eta] \geq 1 - \beta.$$

In this general definition, we choose to *minimize* the score $\mathcal{S}$. This choice is motivated by the learning theory formulation of this problem, used in the specific definition below, where the goal is to select the model (hypothesis) that minimizes a *loss* function quantifying the deviation of the hypothesis from some ground truth. Concretely, if the learner is restricted to selecting between two models (hypotheses) h_0 and h_1, its output bit ρ indicates the selected hypothesis, the general scoring function $\mathcal{S}$ measures the loss of the chosen model with respect to some metric ℓ on the domain $\mathcal{Y}$, and the target function $\mathcal{T}$ is the smaller of the losses of h_0 and h_1. (Specifically, we let the loss of some hypothesis h w.r.t. sample distribution $\mathcal{D}$ and ground truth f be $\mathcal{L}_{\mathcal{D}}(f, h \mid \ell) = \mathbb{E}_{x\sim\mathcal{D}}\left[\ell\big(f(x), h(x)\big)\right]$.) That is:

Definition 1.2 (**Refereed learning, loss minimization (informal)**). *A protocol $[\mathcal{P}_0, \mathcal{P}_1, \mathcal{V}]$ is an (α, η, β)-*refereed learning protocol for loss minimization, *with respect to a family $\mathcal{H}$ of hypotheses and metric ℓ, if for all $b \in \{0,1\}$, $h_0, h_1 \in \mathcal{H}$, and $\mathcal{P}^*_{1-b}$, the learner output ρ satisfies*

$$\Pr\left[\mathcal{L}_{\mathcal{D}}(f, h_\rho \mid \ell) \leq \alpha \min_{s\in\{0,1\}} \mathcal{L}_{\mathcal{D}}(f, h_s \mid \ell) + \eta\right] \geq 1 - \beta.$$

[1] We use the terms *learner* and *verifier* interchangeably. Indeed, the learner now doubles as a verifier.

Bounding the expected score and loss. An alternative formulation would instead bound the *expected* score (resp. loss). That is, the requirement would be that $\mathbb{E}[\mathcal{S}(\rho, \mathcal{D}, f, h_1, \ldots, h_k))] \leq \alpha\mathcal{T}(\mathcal{D}, f, h_1, \ldots, h_k) + \eta$ (in the general case), or $\mathbb{E}[\mathcal{L}_{\mathcal{D}}(f, h_\rho \mid \ell)] \leq \alpha \min_{s\in\{0,1\}} \mathcal{L}_{\mathcal{D}}(f, h_s \mid \ell) + \eta$ (in the case of minimizing the loss). While the two formulations are incomparable in general, our protocols satisfy both with similar parameters.

On strategic provers. The above definition posits that at least one of the provers follows the protocol. We note, however, that refereed learning protocols appear to preserve their guarantees even when both provers are strategic with opposing goals. Some supporting evidence for the implication is the use of protocols developed in the refereed delegation of computation model in real-world applications where truth-telling is economically incentivized. Similar phenomena are manifested in the context of debate systems. See more details in Section 1.2.

1.1.2 Our results

Protocols. We give protocols for both the additive and multiplicative error settings. In the additive error setting, we show how to use the two provers to obtain additive error similar to that obtained by the [GRSY21] protocol mentioned above, while significantly reducing the learner's interaction with both the ground truth and the models: this interaction now consists of only a *single query.*

We then show, via simple extension, that even the provers can use significantly fewer queries at the cost of obtaining an error bound that is both additive and multiplicative. Specifically, to obtain additive loss at most η and multiplicative loss at most $1 + \varepsilon$, our learner only makes a single query to either the ground truth or one of the models, draws $\left(1 + \frac{1}{\varepsilon^2}\right) \cdot \frac{1}{\eta}$ *unlabeled* sample points from the underlying distribution, and has the provers query each model on all of the unlabeled sample points, and query the ground truth on $1 + \frac{1}{\varepsilon^2}$ of them.

Next we design protocols for the low-loss setting, which turns out to be significantly more challenging. Here we would like to guarantee that the learner makes the right choice even when the models' losses are close to each other, up to a *multiplicative* factor of $1 + \varepsilon$ for an arbitrarily small $\varepsilon > 0$. Indeed, in such a setting, the number of samples needed to even observe the difference can be close to the entire sample space. Still, determining which of the two competing models incurs smaller loss may have significant ramifications in applications that require high precision (e.g., using ML models for medical predictions based on imaging or other multi-dimensional measurements, or for financial applications where even tiny error margins become significant over time).

We first concentrate on the zero-one metric[2] on $\mathcal{Y}$ and the uniform underlying distribution. In this setting, we design a protocol where the learner is guaranteed (except with some arbitrarily small constant probability) to select a model whose loss is at most a multiplicative factor of $(1 + \varepsilon)$ worse than the better model's. Moreover, the learner in this protocol is efficient: it makes a single query to the ground truth function $f : \{0, 1\}^d \to \mathcal{Y}$ and has $(1 + \frac{1}{\varepsilon^2})\,\mathrm{poly}(d)$ communication with the provers.

We then extend these results to handle arbitrary metric loss functions as well as arbitrary sample distributions. Our results also take into account the cost of obtaining the desired level of numerical precision. The protocol for this more general setting is guaranteed to select a model whose loss is at

[2]Define the *zero-one metric* ℓ_{zo} by $\ell_{\mathsf{zo}}(y, y') = 1$ if $y \neq y'$ and $\ell_{\mathsf{zo}}(y, y) = 0$. Let the *zero-one loss* between functions f and h w.r.t. distribution $\mathcal{D}$ be $\mathbb{E}_{x\sim\mathcal{D}}\left[\ell_{\mathsf{zo}}(f(x), h(x))\right] = \Pr\left[f(x) \neq h(x)\right]$.

most a multiplicative factor of $3 + \varepsilon$ worse than the best model. Moreover, the learner makes a single query to either f, the two models, or the distribution, has the same runtime as before, and only incurs a small cost of $\lambda \cdot \operatorname{poly} d$ in communication with the provers, where λ is a bound on the allowed numerical precision. We also show how to handle arbitrary precision at the cost of incurring a tiny additive error, with essentially no overhead in communication and runtime.

The prover complexity in the protocols with purely multiplicative error may well depend on the hypothesis class, on the competing models, and on the prover's knowledge of both. In the extreme case where the prover has no apriori knowledge of the ground truth or the models, an exponential number of queries to the model is needed to follow our protocol. We demonstrate that this exponential overhead is inherent for a general solution. Despite this strong lower bound, we also demonstrate a setting where the provers can use some apriori knowledge on the models to gain computational efficiency.

Lower bounds. We complement our protocols by establishing lower bounds that justify some of the complexity parameters of our protocols. We show that in any refereed learning protocol where the learner obtains additive error at most η, and either (a) accesses the ground truth only via labeled samples taken from the given distribution, or else (b) has no knowledge of the underlying distribution of samples other than the samples obtained, the number of samples that the learner obtains from the ground truth must be at least $\frac{1}{\eta}$. In other words, unless the learner both queries the ground truth and also obtains additional information on the underlying distribution (say, the value of its PMF at certain points), $\eta \to 0$ is unattainable.

We also show that the prover's exponential runtime in any general-purpose refereed learning protocol with purely multiplicative error is inherent. The argument for the case where the models can be accessed only as black boxes is straightforward and unconditional. We then demonstrate the need for exponential computational power even for a general solution where the provers have a full whitebox view of the models. Specifically, we show that a refereed learning protocol with a purely multiplicative error guarantee can be used to solve computational problems (such as 3SAT) that are assumed to be exponentially hard. This means that, assuming the hardness of these problems, a refereed learning protocol cannot in general be executed in polynomial time, even in the case where all parties have full white-box access to the model.

1.1.3 Our techniques

Protocols for purely additive and for mixed errors. As a warm-up, we describe our refereed learning protocols for the setting with additive error $\eta > 0$ and zero-one metric on $\mathcal{Y}$. (We describe these protocols in more detail in Section 7.) The natural way for the learner to bound the additive error by η without using the provers is to draw $O(1/\eta^2)$ samples from $\mathcal{D}$, query f, h_0, h_1 at these points, and pick the hypothesis with the smaller empirical loss. When multiplicative loss $\alpha = 1 + \varepsilon$ with $\varepsilon > 0$ is also allowed then the learner can choose to draw only $(1 + \frac{1}{\varepsilon^2}) \cdot \frac{1}{\eta}$ samples from $\mathcal{D}$, find the set S of samples on which h_0 and h_1 disagree, and then pick the hypothesis with the smaller empirical loss *over the samples in* S.

The learner can then use the provers as follows: instead of directly querying f, h_0, h_1 on the sampled points, the learner can send the sample points to the provers and have the provers make the queries and report back the obtained values. If the provers disagree on any returned value, the learner picks *one* such value, makes the query itself, and proceeds with the values provided by the prover that reported the correct value.

Protocols for purely multiplicative error. Next, consider the more challenging setting where only multiplicative error is allowed, i.e., $\eta = 0$. Directly extending the above protocol from the mixed error case would be meaningless, since the number of samples required exceeds the domain size when η approaches 0. An alternative approach would be to compute the empirical loss of the two hypotheses over a *sufficiently large sample* from the "disagreement set" $S = \{x \mid h_0(x) \neq h_1(x)\}$; however, if the set S is sparse then the learner cannot efficiently sample from it. Furthermore, it is not immediately clear how the learner can use the provers to provide it with correctly distributed samples from S. In particular, the above method of settling discrepancies between the provers regarding the value of either one of h_0, h_1, f at a given point does not seem to be useful for the purpose of obtaining a random sample from S. To get around this issue we devise a *certifiable uniform sampling* protocol, described below, that allows the learner to obtain samples from S that are guaranteed to be correctly distributed. Given this protocol, the learner simply picks the model with the smaller empirical loss over $O(1 + \frac{1}{\varepsilon^2})$ random samples from S. Finally, the learner can offload all queries but one to the provers, as described earlier.

Certifiable uniform sampling. In order to design a refereed learning protocol with a computationally efficient learner and bounded communication cost, we construct a protocol that allows the learner to efficiently generate, with the help of the provers, uniform samples from a set S *which can be both exponentially large and exponentially sparse*. In particular, while the provers are assumed to know S, the learner does not need to compute S itself.

As an aside, we note that the ability to use the two provers to *sample* from a distribution that the learner is unable to sample from by itself may be of more general interest beyond the current framework of refereed learning.

The certifiable sampling protocol uses two sub-protocols: (I) a protocol, *Certifiable Sum,* which allows the learner to efficiently obtain the size of S with only a membership query oracle for S; (II) a protocol, *Certifiable Index,* which allows the learner to efficiently obtain the lexicographically i^{th} element in S. In each protocol, the learner runs in time $\operatorname{poly} d$. Now, to obtain m uniformly random samples from S, the learner simply executes the first protocol to obtain $|S|$, then samples m uniform indices $i_1, \ldots, i_m \in \{1, \ldots, |S|\}$, and executes the second protocol to obtain elements $S_{i_1}, \ldots, S_{i_m}$. These two sub-protocols are described next.

The certifiable sum protocol. This protocol allows the learner, assisted by provers, to evaluate quantities of the form $s = \sum_{x \in \{0,1\}^d} t(x)$, given only query access to the function t, in time that is polynomial in d. Now, to compute the size of the set S we set t to be the indicator function $t(x) = 1$ if $x \in S$, and $t(x) = 0$ otherwise.

The protocol has two symmetric stages (one for each prover); for clarity, we just describe one stage. At a high level, each stage of the protocol starts by having one prover make a claim about the total sum s along with a claim about the sums s_0, s_1 on two disjoint halves of the domain. The learner then asks the other prover to identify a half of the domain on which the other prover is lying (if there is such a half). This process continues recursively, for d rounds, until the learner is left with a single point x^* and the prover's claim that $t(x^*) = y$. The learner can then check this claim in a single query to t. The key observation is that if a malicious prover misreports the value of the sum, then it must incorrectly report the value of the sum on at least one half of the domain. Thus, if the second prover is following the protocol, then the lying prover is bound to be caught in one of the d rounds. Indeed, if the prover ever misreports the sum on one half of the domain, that prover

will necessarily lie about the sum of t on (at least) one half of the remaining domain, and the final query of $t(x^*)$ will reveal that the cheating prover is lying; an honest prover's claims, by contrast, will always be found true. See Lemma 3.2 for the full description.

The certifiable index protocol. This protocol allows the verifier to efficiently obtain the i^{th} element of a set $S \in \{0,1\}^d$ with ordering $\prec$, as long as the verifier has membership query access to S and query access to the ordering $\prec$. At a high level, the verifier asks each prover for a candidate i^{th} element x^* and then checks that x^* is in S, and uses certifiable sum to confirm that there are $i-1$ elements in S that are smaller than x^*. More formally, the verifier obtains the claimed i^{th} element x^* from one of the provers, and then executes certifiable sum with the function $t(x) = \mathbb{1}[x \in S \wedge x \prec x^*]$. If certifiable sum returns $i-1$ then the verifier outputs x^* as the i^{th} element of S. Otherwise, it repeats the protocol with the other prover's claimed i^{th} element. (Indeed, x^* is the i^{th} element of S if and only if there are exactly $i-1$ points x such that $t(x) = 1$.)

Extending to general sample distributions. The protocol for certifiable uniform sampling is only relevant when the underlying distribution is the uniform distribution over $\{0,1\}^d$. We extend this protocol to handle sampling from arbitrary distributions $\mathcal{D}$ over $\{0,1\}^d$. This extended protocol allows the learner to efficiently generate certified samples from $\mathcal{D}$, given query access to the distribution's probability mass function $Q_{\mathcal{D}}$. This is done by using the certifiable sum protocol with a function t that depends on the distribution via Birgé's decomposition [Bir87]. (Birgé's decomposition states that a a monotone distribution over $[N]$ can be approximated by a piecewise constant distribution over $O(\log N)$ many buckets.) See Lemma 3.1 for a formal statement of the certifiable sample protocol, and Theorem 4.2 for a formal statement of the resulting refereed learning protocol.

Extending to general metric loss functions. Next we deal with the case of a general (i.e., not zero-one) metric ℓ on the range $\mathcal{Y}$. This case is more challenging since different points in $\mathcal{Y}$ may have very different contributions to the overall loss. In particular, there may be a single point x^* with $\ell(h_1(x^*), f(x^*)) \gg \ell(h_0(x^*), f(x^*))$, and thus a naive learner which, as in the zero-one case, samples from the set $S = \{x \mid \ell(h_0(x), h_1(x)) > 0\}$ and outputs the hypothesis with better loss on the sample, will require many samples before obtaining the point x^* and thus may select the worse model.

To circumvent this issue, we define a rescaled version of $\mathcal{D}$, denoted $\mathcal{D}_\ell^{h_0,h_1}$, that places more mass on points x where $\ell\big(h_0(x), h_1(x)\big)$ is large. By the triangle inequality, if $\ell\big(h_0(x), h_1(x)\big)$ is large, then either $\ell\big(h_0(x), f(x)\big)$ or $\ell\big(h_1(x), f(x)\big)$ must be large as well. Roughly, this technique resolves the above difficulty since if there is such a point x^*, then the rescaled distribution will place proportionally more mass on x^*. More generally, we show that under the rescaled distribution $\mathcal{D}_\ell^{h_0,h_1}$, with high probability the worse hypothesis accounts for more than half of the combined empirical loss of h_0 and h_1 on f, computed over only $O\big(1 + \frac{1}{\varepsilon^2}\big)$ samples. Leveraging the techniques we develop for the zero-one case, we show that the learner, with the help of the provers, can efficiently provide itself with query access to the probability mass function of $\mathcal{D}_\ell^{h_0,h_1}$, and thus can execute the certifiable sample protocol to efficiently generate samples from $\mathcal{D}_\ell^{h_0,h_1}$. The full treatment appears in Section 4.2.

Efficient refereed learning for juntas. We complement the above general-purpose refereed learning protocol, where the provers need exponential time, by demonstrating a refereed learning protocol for a natural learning task where the provers can be efficient. Specifically, we consider the case where h_0 and h_1 are promised to be *j-juntas* (i.e., Boolean functions that each depend only on some set of $j \approx \log d$ input coordinates), and the active index sets J_0 and J_1 of h_0 and h_1 are given as input to all parties. In this setting, we show that the provers can be implemented efficiently. The idea is the following: In the general case, the task that determines the prover runtime is computing the set $S = \{x \mid h_0(x) \neq h_1(x)\}$; when h_0 and h_1 are promised to be juntas with $j \approx \log d$ active indices, the provers can compute S in time $\operatorname{poly} d$. Since the distribution is uniform, the certifiable sample protocol used in the general case can also be implemented efficiently, and thus the provers can be made to run in time $\operatorname{poly} d$. See Proposition 6.1 for a formal statement and construction of the protocol.

Lower bounds. As described earlier, we show several different impossibility results, where each result demonstrates the optimality of a different aspect of the protocols. The first result (Theorem 5.1) demonstrate that without query access to the ground truth f, a learner would in general need a prohibitive number of queries. The argument is straightforward: fix $\{h_0, h_1\}$, sample $b \sim \{0,1\}$ and let $f \leftarrow h_b$. Consider the cheating prover that executes the honest protocol, except it "pretends" that $f = h_{1-b}$. As long as the learner does not obtain any sample x with $f(x) \neq h_{1-b}(x)$, it cannot refute the malicious prover's claim that h_{1-b} has zero loss, and thus cannot determine if it should accept h_0 or h_1. The proof that query access to $Q_{\mathcal{D}}$,the PMF of $\mathcal{D}$, is necessary (Theorem 5.3) follows a similar outline.

Turning to the complexity of the provers, we first observe that, when the provers only have black-box access to h_0 and h_1, the provers may need to query h_0 and h_1 at $\Omega(2^d)$ points to find the hypothesis with better loss, up to a multiplicative factor. (Indeed, for all $z \in \{0,1\}^d$ let $h_z = \mathbb{1}[x = z]$. Sample hypotheses $z, z' \sim \{0,1\}^d$ and ground truth $f \sim \{h_z, h_{z'}\}$. Any algorithm $\mathcal{A}$ which gets query access to $h_z, h_{z'}$, and f cannot distinguish whether $f = h_z$ or $f = h_{z'}$ until it queries either z or z'. Since these are uniformly random points, $\mathcal{A}$ must make $\Omega(2^d)$ queries.) In Theorem 5.4 we then extend this bound to the case where the provers are given an explicit description of h_0 and h_1. This bound proceeds by reduction from 3-SAT: Any refereed learning protocol which guarantees any purely multiplicative bound on the loss of the computed hypothesis can be used to distinguish satisfiable formulas from unsatisfiable ones.

1.2 Related work

This work combines ideas, formalisms, and techniques from a number of different areas. Here we briefly review some of the main works that inspired the present one, as well as works that may appear related but differ in some substantial ways.

The idea of considering the computational power of a model that involves a weak verifier and two or more provers, one of which is assumed to be honest, goes back to the works of Feige et al. [FST88, FK97], who also observe that the provers can be viewed as *competing*. Later, Canetti et al. [CRR11, CRR13] and Kol and Raz [KR14] have demonstrated several protocols for delegating arbitrary computations to untrusted servers in that model. These *refereed delegation* protocols are both simpler and significantly more efficient than ones designed for a single untrusted prover. In fact, one of the protocols in [CRR13], which is sufficiently efficient to be practical, is currently

in commercial use within an application where provers are competing strategic agents, and the protocol is used to incentivize the agents to be truthful [AAT+25].

We note, however, that known refereed delegation protocols do not appear to be directly applicable to our setting. In particular, these protocols are geared towards verifying fully specified deterministic computations with inputs that are readable by the verifier in full. In contrast, in our setting the learner is presented with a black box model whose code is unknown and whose sample space is potentially huge.

Ergun et al. and later Rothblum et al. [EKR04, RVW13] consider a weak verifier that uses the power of a *single* untrusted prover to decide whether some huge mathematical object, which is accessible to the verifier only via queries, has some claimed property, or else is far from having the property. Herman and Rothblum [HR22, HR24b, HR24a] consider the case where the object in question is a distribution, and the verifier's access to the distribution is via obtaining samples. Goldwasser et al. [GRSY21] consider a (single) prover that wishes to convince a suspicious verifier that a given concept has some desirable properties. In all, to the best of our knowledge, this is the first work that considers the power of the two-prover model in the context of learning, testing, or verifying the properties of black-box objects, models being a special case.

A related and very vibrant area of research is that of *debate systems* where a panel of competing AI agents debate in order to help a weak referee (either a human or another AI agent) obtain a meaningful decision, often with respect to AI safety and alignment. See, e.g., [ICA18, GCW+24]. However, both the methods and the specific goals in these works are very different than the one here.

1.3 Organization

Section 2 introduces a framework for refereed learning and provides a formal definition of refereed learning protocols. Section 3 develops several key tools which are used to construct refereed learning protocols. Section 4 leverages these tools to construct refereed learning protocols for the zero-one loss (Theorem 4.2) and for general metric loss functions (Theorem 4.4). Section 5 proves several lower bounds which justify the learner's access model and the runtime of the provers. We conclude with Sections 6 and 7. Section 6 extends the earlier protocols to the setting where the distribution and loss are measured to arbitrary precision, as well as an application to junta functions where the provers can be implemented efficiently. Section 7 presents some simple protocols for the additive and additive/multiplicative error settings.

2 Framework for refereed learning

In this section we formally define a *refereed learning protocol*. First, we provide a definition in terms of a "score" and "target" function, and second, we provide a definition for the special case where the score and target correspond to loss minimization.

Throughout this paper we use the following standard notation: for a protocol $[\mathcal{P}_0, \mathcal{P}_1, \mathcal{V}]$, let $[\mathcal{P}_0(A), \mathcal{P}_1(B), \mathcal{V}(C)](D)$ be a random variable denoting the output of $[\mathcal{P}_0, \mathcal{P}_1, \mathcal{V}]$ when prover $\mathcal{P}_0$ has input A, prover $\mathcal{P}_1$ has input B, learner-verifier $\mathcal{V}$ has input C, and all have input D. We define the *communication complexity* of a protocol $[\mathcal{P}_0, \mathcal{P}_1, \mathcal{V}]$ as the number of bits sent between $\mathcal{P}_0$, $\mathcal{P}_1$, and $\mathcal{V}$. Additionally, for an algorithm $\mathcal{A}$ and function f, let $\mathcal{A}^f$ denote $\mathcal{A}$ with query access to f—that is, $\mathcal{A}$ can specify a query x and receive response $f(x)$. The *query complexity* of $\mathcal{A}^f$ is the

number of queries made by $\mathcal{A}$ to f. For a distribution $\mathcal{D}$, let $\mathcal{A}^{\mathcal{D}}$ denote $\mathcal{A}$ with access to samples drawn from $\mathcal{D}$. The *sample complexity* of $\mathcal{A}^{\mathcal{D}}$ is the number of samples $\mathcal{A}$ draws from $\mathcal{D}$. We say a prover $\mathcal{P}$ is *honest* if it runs the algorithm that is specified by the protocol; a *malicious* prover (denoted $\mathcal{P}^*$) may deviate from the algorithm specified by the protocol.

In order to define refereed learning, we first define a score and a target. A *score* $\mathcal{S}$ (parameterized by $k, d \in \mathbb{N}$ and a set $\mathcal{Y}$) is a function that sends tuples $(\rho, f, h_1, \ldots, h_k, \mathcal{D}) \mapsto \mathbb{R}$, where ρ is an output of the learner, $f, h_1, \ldots, h_k$ are functions $\{0,1\}^d \to \mathcal{Y}$ for some fixed range $\mathcal{Y}$, and $\mathcal{D}$ is a distribution over $\{0,1\}^d$. Additionally, a *target* $\mathcal{T}$ (parametrized by $k, d \in \mathbb{N}$ and a set $\mathcal{Y}$) is a function that sends tuples $(f, h_1, \ldots, h_k, \mathcal{D}) \mapsto \mathbb{R}$.

In order to encode various ways in which the parties can access the ground truth, the sample distribution, and the hypotheses, we formalize an *access model* as three oracles, one for each prover and one for the verifier. An oracle allows some sample requests and queries (namely, it returns either a sample or the value of the relevant function applied to the query, as appropriate) while disallowing others. We let $\mathcal{A}^{\mathcal{O}(f,h_1,\ldots,h_k,\mathcal{D})}$ denote algorithm $\mathcal{A}$ with access to $f, h_1, \ldots, h_k$ and $\mathcal{D}$, controlled by oracle $\mathcal{O}$.

Definition 2.1 (Refereed learning protocol—general case)**.** *Fix score $\mathcal{S}$ and target $\mathcal{T}$ with respect to parameters $k, d \in \mathbb{N}$ and range $\mathcal{Y}$. Let $\mathcal{H} \subseteq \{h : \{0,1\}^d \to \mathcal{Y}\}$ and $\mathbb{D} \subseteq \{\mathcal{D} \mid \mathrm{supp}(\mathcal{D}) \subseteq \{0,1\}^d\}$. Fix oracle access models $\mathcal{O}_0, \mathcal{O}_1$ and $\mathcal{O}_{\mathcal{V}}$, slack parameters $\alpha \geq 1$ and $\eta \geq 0$, and soundness error $\beta \geq 0$. A protocol $[\mathcal{P}_0, \mathcal{P}_1, \mathcal{V}]$ is a* (k, α, η, β)-refereed learning protocol (RLP) for $\mathcal{H}$ and $\mathbb{D}$ with respect to $\mathcal{S}, \mathcal{T}$, and oracles $\mathcal{O}_0, \mathcal{O}_1$ and $\mathcal{O}_{\mathcal{V}}$, *if for all distributions $\mathcal{D} \in \mathbb{D}$, functions $f : \{0,1\}^d \to \mathcal{Y}$ and $h_1, \ldots, h_k \in \mathcal{H}$, the following holds:*

- *For all $b \in \{0,1\}$ and $\mathcal{P}^*_{1-b}$, the output $\rho \leftarrow \left[\mathcal{P}_b^{\mathcal{O}_b(f,h_1,\ldots,h_k,\mathcal{D})}, \mathcal{P}^*_{1-b}, \mathcal{V}^{\mathcal{O}_{\mathcal{V}}(f,h_1,\ldots,h_k,\mathcal{D})}\right]$ satisfies*

$$\Pr\Big[\mathcal{S}(\rho, f, h_1, \ldots, h_k, \mathcal{D}) \leq \alpha \cdot \mathcal{T}(f, h_1, \ldots, h_k, \mathcal{D}) + \eta\Big] \geq 1 - \beta,$$

where the randomness is over the coins of the verifier, the honest prover, and the oracles.

The definition above is quite general and can be applied to settings beyond learning. In this work, we focus on using the refereed setting for learning and adopt the following, more concrete definition. As compared to Definition 2.1, in Definition 2.3 we replace the score and target with a metric loss function $\mathcal{L}$ and only consider the case of $k = 2$.

Definition 2.2 (Metric loss function, zero-one metric)**.** *Fix a range $\mathcal{Y}$ with metric*[3] *$\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$. Additionally, for all $d \in \mathbb{N}$ and all functions $f, h : \{0,1\}^d \to \mathcal{Y}$ and distributions $\mathcal{D}$ over $\{0,1\}^d$, define the* metric loss between f and h with respect to ℓ and $\mathcal{D}$ *as $\mathcal{L}_{\mathcal{D}}(f, h \mid \ell) = \mathbb{E}_{x \sim \mathcal{D}}\left[\ell\big(f(x), h(x)\big)\right]$. We will omit the dependence on ℓ when it is clear from context. Additionally, we define the* zero-one *metric ℓ_{zo} by $\ell_{\mathsf{zo}}(y, y') = 1$ if $y \neq y'$ and $\ell_{\mathsf{zo}}(y, y) = 0$.*

Definition 2.3 (Refereed learning protocol—loss minimization)**.** *Fix range $\mathcal{Y}$ with metric ℓ, dimension $d \in \mathbb{N}$, and $\mathcal{H} \subseteq \{h : \{0,1\}^d \to \mathcal{Y}\}$ and $\mathbb{D} \subseteq \{\mathcal{D} \mid \mathrm{supp}(\mathcal{D}) \subseteq \{0,1\}^d\}$. Fix oracle access models $\mathcal{O}_0, \mathcal{O}_1$ and $\mathcal{O}_{\mathcal{V}}$, slack parameters $\alpha \geq 1$ and $\eta \geq 0$, and soundness error $\beta \geq 0$. A protocol $[\mathcal{P}_0, \mathcal{P}_1, \mathcal{V}]$ is a* (α, η, β)-refereed learning protocol (RLP) for $\mathcal{H}$ and $\mathbb{D}$ with respect to ℓ and oracles $\mathcal{O}_0, \mathcal{O}_1$ and $\mathcal{O}_{\mathcal{V}}$, *if for all distributions $\mathcal{D} \in \mathbb{D}$, functions $f : \{0,1\}^d \to \mathcal{Y}$ and $h_0, h_1 \in \mathcal{H}$, the following holds:*

[3] A metric ℓ satisfies non-negativity (i.e., $\ell(y, y') > 0$ if and only if $y \neq y'$), symmetry, and the triangle inequality.

- *For all $b \in \{0,1\}$ and $\mathcal{P}^*_{1-b}$, the bit $\rho \leftarrow \left[\mathcal{P}_b^{\mathcal{O}_b(f,h_0,h_1,\mathcal{D})}, \mathcal{P}^*_{1-b}, \mathcal{V}^{\mathcal{O}_\mathcal{V}(f,h_0,h_1,\mathcal{D})}\right]$ satisfies*

$$\Pr\Big[\mathcal{L}_\mathcal{D}(h_\rho, f \mid \ell) \le \alpha \cdot \min_{s\in\{0,1\}} \mathcal{L}_\mathcal{D}(h_s, f \mid \ell) + \eta\Big] \ge 1-\beta,$$

where the randomness is over the coins of the verifier, the honest prover, and the oracles.

Definition 2.3 allows the distribution, metric, and functions to take on arbitrary real values. To handle issues with describing and sending arbitrary real-values, we focus on the setting where the distribution $\mathcal{D}$ and the metric ℓ are "λ-precise":

Definition 2.4 (Set $\mathbb{Q}_\lambda$, λ-precise, distribution family $\mathbb{D}_\lambda$)**.** *Fix $\lambda \in \mathbb{N}$ and define $\mathbb{Q}_\lambda \subseteq \mathbb{Q}$ as the set of all rationals $\frac{p}{q}$ such that $\max\{|p|,|q|\} \le 2^\lambda$. A metric ℓ is* λ-precise *if its image satisfies $\mathsf{im}(\ell) \subseteq \mathbb{Q}_\lambda$. A probability distribution $\mathcal{D}$ over $\{0,1\}^d$ with probability mass function $Q_\mathcal{D}$ is* λ-precise *if its image satisfies $\mathsf{im}(Q_\mathcal{D}) \subseteq \mathbb{Q}_\lambda$. Let $\mathbb{D}_{d,\lambda}$ denote the set of λ-precise distributions over $\{0,1\}^d$. When d is clear from context we will write $\mathbb{D}_\lambda$ instead of $\mathbb{D}_{d,\lambda}$.*

Finally, in order to succinctly refer to the set of all functions $\{0,1\}^d \to \mathcal{Y}$ and all distributions over $\mathcal{D}$, we define the following families:

Definition 2.5 (Families $\mathfrak{F}$ and $\mathfrak{D}$)**.** *For all $d \in \mathbb{N}$ and sets $\mathcal{Y}$, let $\mathfrak{F}_{d,\mathcal{Y}} = \left\{f : \{0,1\}^d \to \mathcal{Y}\right\}$ and $\mathfrak{D}_d = \left\{\mathcal{D} \mid \mathrm{supp}(\mathcal{D}) \subseteq \{0,1\}^d\right\}$. When d and $\mathcal{Y}$ are clear from context we write $\mathfrak{F}$ and $\mathfrak{D}$.*

3 Tools for refereed learning

In this section we develop several key tools for designing refereed learning protocols.

3.1 Certifiable sample and certifiable sum

Our first tool, Lemma 3.1, allows the verifier to efficiently sample from a distribution that is close to $\mathcal{D}$ given query access to its *probability mass function* $Q_\mathcal{D}$ (defined by $Q_\mathcal{D}(x) = \Pr_{X\sim\mathcal{D}}[X = x]$). For all distributions P and Q over domain $\mathcal{X}$, the *total-variation distance* (TV) distance between P and Q is $d_{\mathrm{TV}}(P,Q) = \sup_{A\subseteq\mathcal{X}} |P(A) - Q(A)|$.

Lemma 3.1 (Certifiable sample)**.** *There exists a protocol $[\mathcal{P}_0, \mathcal{P}_1, \mathcal{V}]$ such that for all $d \in \mathbb{N}$, distributions $\mathcal{D}$ over $\{0,1\}^d$ with probability mass function $Q_\mathcal{D}$, distance $\delta \in (0,1)$, and sample size $m \in \mathbb{N}$, there exists a distribution $\widehat{\mathcal{D}}$ with $d_{\mathrm{TV}}\big(\widehat{\mathcal{D}}, \mathcal{D}\big) \le \delta$ such that:*

1. *For all $b \in \{0,1\}$ and $\mathcal{P}^*_{1-b}$, the output $\left[\mathcal{P}_b^{Q_\mathcal{D}}, \mathcal{P}^*_{1-b}, \mathcal{V}^{Q_\mathcal{D}}\right](d,\delta,m)$ consists of m samples $x_1,\ldots,x_m \sim \widehat{\mathcal{D}}$.*
2. *The verifier's runtime and protocol's communication complexity are $(m + \frac{1}{\delta}\log\frac{1}{\delta}) \cdot \mathrm{poly}\, d$.*

In order to prove Lemma 3.1 we leverage our second important tool, Lemma 3.2, which gives a protocol for the verifier to efficiently determine the answer to arbitrary functions of the form $\sum_{x\in\{0,1\}^d} t(x)$ given only query access to t.

Lemma 3.2 (Certifiable sum)**.** *Fix $\lambda \in \mathbb{N}$. There exists a protocol $[\mathcal{P}_0, \mathcal{P}_1, \mathcal{V}]$ such that for all dimensions $d \in \mathbb{N}$ and functions $t : \{0,1\}^d \to \mathbb{Q}_\lambda$ the following holds:*

1. *For all* $b \in \{0,1\}$ *and* $\mathcal{P}^*_{1-b}$ *we have* $\left[\mathcal{P}^t_b, \mathcal{P}^*_{1-b}, \mathcal{V}^t\right](d) = \sum_{x\in\{0,1\}^d} t(x)$.

2. *The runtime of the verifier and the communication complexity of the protocol are both* $\lambda \cdot \operatorname{poly} d$.

3. *The verifier makes* 2 *queries to* t.

We defer the proof of Lemma 3.2 and complete the proof of Lemma 3.1 below.

Proof of Lemma 3.1. Let $S = \operatorname{supp}(\mathcal{D})$ and $N = |S|$. Let S_i denote the i^{th} element of S when its elements are ordered according to their probability masses. At a high level, the verifier in the certifiable sample protocol leverages certifiable sum and a protocol called certifiable index (which we present in Claim 3.5) to certifiably construct a monotone distribution $\mathcal{D}'$ over $[N]$ that captures the "shape" of $\mathcal{D}$—that is, sampling $i \sim \mathcal{D}'$ and outputting $x = S_i$ yields a distribution $\widehat{\mathcal{D}}$ such that $d_{\text{TV}}(\widehat{\mathcal{D}}, \mathcal{D}) \le \delta$. Thus, to sample such an x, the verifier simply samples $i \sim \mathcal{D}'$ and then executes certifiable index to obtain S_i.

In order to bound the runtime and communication complexity of the verifier, we use a well-known decomposition result from [Bir87] which states that a monotone distribution over $[N]$ can be approximated to error δ by a histogram with $O(\log(N)/\delta)$ buckets of exponentially increasing cardinality. We present the formulation of Birgé's decomposition presented in [Can20, Appendix D.4].

Definition 3.3 (Oblivious decomposition, flattened histogram [Can20])**.** *For all* $N \in \mathbb{N}$ *and* $\delta > 0$*, the corresponding* oblivious decomposition *of* $[N]$ *is the partition* $(I_1, \ldots, I_\ell)$ *of disjoint intervals, where* $\ell = \Theta\left(\frac{\log N}{\delta}\right)$*, and* $|I_k| \le \lfloor (1+\delta)^k \rfloor$ *for all* $k \in [\ell]$.

Additionally, define the flattened histogram $\Phi_\delta[\mathcal{D}]$ *as follows:*

$$\forall k \in [\ell], \forall i \in I_k,\ \ \Phi_\delta[\mathcal{D}](i) = \frac{\mathcal{D}(I_k)}{|I_k|}.$$

Fact 3.4 (Birgé's decomposition [Bir87, Can20])**.** *If* $\mathcal{D}$ *is a monotone non-increasing distribution over* $[N]$ *then* $d_{\text{TV}}(\mathcal{D}, \Phi_\delta[\mathcal{D}]) \le \delta$ *for all* $\delta > 0$.

For a set S, a total ordering $\prec$, and an algorithm $\mathcal{A}$, let $\mathcal{A}^{S,\prec}$ denote $\mathcal{A}$ with membership query access to S and query access to $\mathbb{1}[\cdot \prec \cdot]$ (the function that takes as input (x, x') and returns 1 if $x \prec x'$ and 0 otherwise).

Claim 3.5 (Certifiable index)**.** *For all* $d \in \mathbb{N}$ *let* $\prec_d$ *be a total ordering on* $\{0,1\}^d$*. There exists a protocol* $[\mathcal{P}_0, \mathcal{P}_1, \mathcal{V}]$ *such that for all* $d \in \mathbb{N}$*, all sets* $S \subseteq \{0,1\}^d$ *ordered according to* $\prec_d$*, and all* $i \in [|S|]$ *the following holds:*

1. *For all* $b \in \{0,1\}$ *and all* $\mathcal{P}^*_{1-b}$ *we have* $\left[\mathcal{P}_b^{\prec_d}(S), \mathcal{P}^*_{1-b}, \mathcal{V}^{S,\prec_d}\right](d, i) = S_i$*, the* i^{th} *element in* S *(where the initial element in* S *has index* 1*).*

2. *The runtime of the verifier and communication complexity of the protocol is* $\operatorname{poly} d$.

We defer the proof of Claim 3.5 and continue with the proof of Lemma 3.1. Since probabilities may require arbitrary precision to express completely, the provers cannot hope to send the exact probabilities. In order to circumvent this issue and bound the communication cost of certifiable sample, we will first perform a preproccessing step that provides the verifier with query access to the probability mass function $Q_{\mathcal{D}_\lambda}$ of the distribution $\mathcal{D}_\lambda$ defined by $\mathcal{D}_\lambda(x) = \frac{\lfloor \mathcal{D}(x) \rfloor_\lambda}{\sum_{x\in\{0,1\}^d} \lfloor \mathcal{D}(x) \rfloor_\lambda}$ where

$\lambda > d$ is an integer and $\lfloor y \rfloor_\lambda$ denotes $2^{-\lambda} \cdot \lfloor 2^\lambda \cdot y \rfloor$ for all $y \in \mathbb{R}$—that is, $\lfloor y \rfloor_\lambda$ denotes the nearest multiple of $2^{-\lambda}$ that is at most y.

Before presenting the certifiable sample protocol, we introduce an ordering $\prec$ which will ensure the distribution $\mathcal{D}'$ is monotone so that we can apply Birgé's decomposition (Fact 3.4). Define the ordering $\prec$ on $\{0,1\}^d$ by $x \prec y$ if $\mathcal{D}_\lambda(x) < \mathcal{D}_\lambda(y)$, with ties broken according to the lexicographical ordering on $\{0,1\}^d$.

$$\left[\mathcal{P}_0^{Q_\mathcal{D}}, \mathcal{P}_1^{Q_\mathcal{D}}, \mathcal{V}^{Q_\mathcal{D}}\right](d, \delta, m)$$

1. $\mathcal{V}$: Set $\lambda \leftarrow d + \log \frac{4+\delta}{\delta}$. Obtain $T_\lambda = \sum_{x \in \{0,1\}^d} \lfloor \mathcal{D}(x) \rfloor_\lambda$ using Lemma 3.2 with $t(x) = \lfloor \mathcal{D}(x) \rfloor_\lambda$ and $\lambda \leftarrow \lambda$. Provide query access to $Q_{\mathcal{D}_\lambda}$ using query access to $Q_\mathcal{D}$ and T_λ.
2. $\mathcal{P}_0, \mathcal{P}_1$: Let $S \leftarrow \mathrm{supp}(\mathcal{D}_\lambda)$ and order S according to $\prec$.
3. $\mathcal{V}$: Obtain $N = |S|$ via certifiable sum (Lemma 3.2) using $t(x) = \mathbb{1}[x \in S]$ and $\lambda \leftarrow d$.
4. $\mathcal{V}$: Let $(I_1, \ldots, I_\ell)$ be the partition of $[N]$ into disjoint intervals given by Definition 3.3. Let $\mathcal{D}'$ denote[a] the distribution over $[N]$ given by setting $\mathcal{D}'(j) = \mathcal{D}_\lambda(S_j)$ for all $j \in [N]$.
5. $\mathcal{V}$: For each $k \in [\ell]$ let $S[I_k]$ denote $\{S_i : i \in I_k\}$, and obtain $L_k = S[I_k]_1$ and $R_k = S[I_k]_{|I_k|}$ via the certifiable index protocol (Claim 3.5).[b]
6. $\mathcal{V}$: For each $k \in [\ell]$ obtain[c] $p_k = \mathcal{D}_\lambda(S[I_k])$ via certifiable sum (Lemma 3.2) with $t_k(x) = \mathcal{D}_\lambda(x) \cdot \mathbb{1}[x \in S[I_k]]$ and $\lambda \leftarrow \lambda$.
7. $\mathcal{V}$: Construct a distribution $\widehat{\mathcal{D}}'$ as follows: for all $k \in [\ell]$ and $i \in I_k$, set $\widehat{\mathcal{D}}'(i) = \frac{p_k}{|I_k|}$.
8. $\mathcal{V}$: Sample indices $i_1, \ldots, i_m \sim \widehat{\mathcal{D}}'$ and run certifiable index (Claim 3.5) to obtain elements $S_{i_1}, \ldots, S_{i_m}$. Output $x_1, \ldots, x_m \leftarrow S_{i_1}, \ldots, S_{i_m}$.

[a] The verifier need not explicitly construct $\mathcal{D}'$.
[b] These are the leftmost and rightmost elements of $S[I_k]$.
[c] L_k and R_k are used to provide query access to $\mathbb{1}[x \in S[I_k]]$—see the proof for more details.

Protocol 1: certifiable sample

To prove Item 1, we will argue that the distributions $\widehat{\mathcal{D}}'$ and $\Phi_\delta[\mathcal{D}']$ (Definition 3.3) are equivalent. Then, we will argue that sampling $i \sim \mathcal{D}'$ and outputting $x \leftarrow S_i$ yields the same distribution as sampling $x \sim \mathcal{D}_\lambda$. Finally, we will complete the proof by leveraging the guarantee of Birgé's decomposition for monotone distributions (Fact 3.4), and by arguing the $\mathcal{D}_\lambda$ and $\mathcal{D}$ are close in total variation distance. First, by Lemma 3.2, the verifier correctly obtains $T_\lambda = \sum_{x \in \{0,1\}^d} \lfloor \mathcal{D}(x) \rfloor_\lambda$, and thus can provide itself with query access to $Q_{\mathcal{D}_\lambda}$ by first querying $Q_\mathcal{D}(x)$ to obtain $\mathcal{D}(x)$ and then computing $Q_{\mathcal{D}_\lambda}(x) = \frac{\lfloor \mathcal{D}(x) \rfloor_\lambda}{T_\lambda}$.

Claim 3.6. *If either $\mathcal{P}_0$ or $\mathcal{P}_1$ is honest, then the distribution $\widehat{\mathcal{D}}'$ constructed by $\mathcal{V}$ in Protocol 1 is equivalent to the flattened histogram $\Phi_\delta[\mathcal{D}']$ defined in Definition 3.3.*

Proof. Assume without loss of generality that $\mathcal{P}_0$ is honest. Then by Lemma 3.2, the verifier obtains the correct value of $N = |S|$. Similarly, by Claim 3.5, for all $k \in [\ell]$ the verifier obtains the correct endpoints $L_k, R_k \in S$ of the set $S[I_k]$. Given the endpoints of $S[I_k]$ and query access to the probability mass function $Q_{\mathcal{D}_\lambda}$ of distribution $\mathcal{D}_\lambda$, the verifier can provide itself with query

access to $t(x) = \mathcal{D}_\lambda(x) \cdot \mathbb{1}[x \in S[I_k]]$. Thus, by Lemma 3.2, the verifier obtains the correct values of $p_k = \mathcal{D}_\lambda(S[I_k])$ for each $k \in [\ell]$. It follows that for all $k \in [\ell]$ and $i \in [k]$ we have

$$\widehat{\mathcal{D}}'(i) = \frac{\mathcal{D}_\lambda(S[I_k])}{|I_k|} = \frac{\mathcal{D}'(I_k)}{|I_k|} = \Phi_\delta[\mathcal{D}'](i). \qquad \square$$

By definition of $\mathcal{D}'$, sampling $j \sim \mathcal{D}'$ and outputting $x \leftarrow S_j$ is equivalent to sampling $x \sim \mathcal{D}_\lambda$. Moreover, since $\mathcal{D}'$ is a monotone distribution over $[N]$, Fact 3.4 immediately implies that $d_{\mathrm{TV}}(\mathcal{D}', \Phi_\delta[\mathcal{D}']) \le \delta$. Let $\widehat{\mathcal{D}}_\lambda$ be the distribution given by sampling $j \sim \widehat{\mathcal{D}}'$ and outputting $x \leftarrow S_j$. By Claim 3.6, we have $d_{\mathrm{TV}}(\mathcal{D}', \widehat{\mathcal{D}}') \le \delta$ and hence $d_{\mathrm{TV}}(\mathcal{D}_\lambda, \widehat{\mathcal{D}}_\lambda) \le \delta$ as well. To complete the proof of Item 1, it remains to argue that $d_{\mathrm{TV}}(\mathcal{D}_\lambda, \mathcal{D}) \le \delta$, and hence $d_{\mathrm{TV}}(\widehat{\mathcal{D}}_\lambda, \mathcal{D}) \le 2\delta$.

Claim 3.7. *Fix $d, \lambda \in \mathbb{N}$ with $\lambda > d$ and a distribution $\mathcal{D}$ over $\{0,1\}^d$. Let $\mathcal{D}_\lambda$ be the distribution defined by $\mathcal{D}_\lambda(x) = \frac{\lfloor \mathcal{D}(x) \rfloor_\lambda}{\sum_{x \in \{0,1\}^d} \lfloor \mathcal{D}(x) \rfloor_\lambda}$, where $\lfloor y \rfloor_\lambda$ denotes $2^{-\lambda} \cdot \lfloor 2^\lambda \cdot y \rfloor$ for all $y \in \mathbb{R}$. Then $\mathcal{D}_\lambda$ is λ-precise and $d_{\mathrm{TV}}(\mathcal{D}_\lambda, \mathcal{D}) \le 2^{d+1-\lambda}$.*

Proof. Observe that for each $y \in [0,1]$ we have $|y - \lfloor y \rfloor_\lambda| \le 2^{-\lambda}$, and therefore $|\lfloor \mathcal{D}(x) \rfloor_\lambda - \mathcal{D}(x)| \le 2^{-\lambda}$ for each $x \in \{0,1\}^d$. It follows that $\sum_{x \in \{0,1\}^d} |\lfloor \mathcal{D}(x) \rfloor_\lambda - \mathcal{D}(x)| \le 2^{d-\lambda}$ and hence, since $\sum_{x \in \{0,1\}^d} \mathcal{D}(x) = 1$, that $\sum_{x \in \{0,1\}^d} \lfloor \mathcal{D}(x) \rfloor_\lambda \in \left[1 \pm 2^{d-\lambda}\right]$. Let $T_\lambda = \sum_{x \in \{0,1\}^d} \lfloor \mathcal{D}(x) \rfloor_\lambda$. Then $T_\lambda \in \left[1 \pm 2^{d-\lambda}\right]$. Let $a \in [\pm 2^{d-\lambda}]$ be such that $T_\lambda = 1 + a$. Then

$$\begin{aligned}
d_{\mathrm{TV}}(\mathcal{D}_\lambda, \mathcal{D}) &= \frac{1}{2} \sum_{x \in \{0,1\}^d} |\mathcal{D}_\lambda(x) - \mathcal{D}(x)| \\
&= \frac{1}{2 \cdot T_\lambda} \sum |\lfloor \mathcal{D}(x) \rfloor_\lambda - T_\lambda \cdot \mathcal{D}(x)| \\
&\le \frac{1}{2 \cdot T_\lambda} \Big(\sum |\lfloor \mathcal{D}(x) \rfloor_\lambda - (1+a) \cdot \mathcal{D}(x)| \Big) \\
&\le \frac{1}{2 \cdot T_\lambda} \Big(\sum |\lfloor \mathcal{D}(x) \rfloor_\lambda - \mathcal{D}(x)| + 2^{d-\lambda} \sum \mathcal{D}(x) \Big) \\
&\le \frac{1}{2 \cdot T_\lambda} \left(2^{d-\lambda} + 2^{d-\lambda} \right) \\
&\le \frac{2^{d-\lambda}}{1 - 2^{d-\lambda}} \le 2^{d+1-\lambda}.
\end{aligned}$$

To see that $\mathcal{D}_\lambda$ is λ-precise, observe that $\mathcal{D}_\lambda(x) = \frac{\lfloor 2^\lambda \mathcal{D}(x) \rfloor}{\sum \lfloor 2^\lambda \mathcal{D}(x) \rfloor}$ has both numerator and denominator that are non-negative integers at most 2^λ. $\square$

By Claim 3.7 and our choice of λ in Protocol 1, we have $d_{\mathrm{TV}}(\widehat{\mathcal{D}}_\lambda, \mathcal{D}) \le 2\delta$, and hence setting $\delta \leftarrow \delta/2$ completes the proof of Item 1. To see why Item 2 holds, recall that by Lemma 3.2 and Claim 3.5, each call to certifiable sum with $\lambda = d + \log \frac{4+\delta}{\delta}$ uses $\lambda \operatorname{poly} d = \log \frac{1}{\delta} \cdot \operatorname{poly} d$ bits of communication and verifier runtime, and each call to certifiable index, uses a verifier that runs in time $\operatorname{poly}(d)$ and uses $\operatorname{poly}(d)$ bits of communication. Since $N \le 2^d$, and certifiable sum is called once for each of the $\ell = \Theta\left(\frac{\log N}{\delta}\right)$ partitions, computing the distribution $\widehat{\mathcal{D}}'$ takes the verifier $\frac{1}{\delta} \log \frac{1}{\delta} \cdot \operatorname{poly} d$ time, and uses $\frac{1}{\delta} \log \frac{1}{\delta} \cdot \operatorname{poly} d$ bits of communication. Similarly, since the verifier draws m samples and calls certifiable index for each sample, the total runtime of the verifier and communication complexity of the protocol is $\left(\frac{1}{\delta} \log \frac{1}{\delta} + m\right) \cdot \operatorname{poly} d$. $\square$

In the remainder of the section we complete the proof of our second tool, the certifiable sum protocol (Lemma 3.2), and leverage it to complete the proof of the certifiable index protocol (Claim 3.5). At a high level, the protocol works as follows: For all $b \in \{0,1\}$ let $C_b = \left\{x \in \{0,1\}^d \mid x_1 = b\right\}$. First, $\mathcal{P}_0$ claims that the sum over $\{0,1\}^d$ is $\widehat{T}$, and that the sum over C_0 and C_1 is $\widehat{T}_0$ and $\widehat{T}_1$ respectively. Since C_0 and C_1 are disjoint and $C_0 \cup C_1 = \{0,1\}^d$, we must have $\widehat{T}_0 + \widehat{T}_1 = \widehat{T}$. The key observation is that if $\widehat{T} \neq T$ (the correct value of the sum), then either $\widehat{T}_0 \neq T_0$ or $\widehat{T}_1 \neq T_1$ (the correct values of the sum on C_0 and C_1). The verifier can then ask $\mathcal{P}_1$ for the bit b such that $\widehat{T}_b \neq T_b$, and repeat the above steps on C_b. After d rounds there is only a single point remaining in the subcube, and hence the verifier can check if the claim made by $\mathcal{P}_0$ is correct by making a single query to t. The protocol to certify a claim made by $\mathcal{P}_1$ is identical but has the roles of $\mathcal{P}_0$ and $\mathcal{P}_1$ switched.

Proof of Lemma 3.2. Our protocol consists of two phases. In the first phase the verifier uses $\mathcal{P}_0$ to verify the claim made by $\mathcal{P}_1$, and in the second phase the verifier uses $\mathcal{P}_1$ to verify the claim made by $\mathcal{P}_0$. The final protocol executes both phases and returns the first verified claim (this will be correct if at least one prover is honest). We first define the following notation: for all $t : \{0,1\}^d \to \mathbb{Q}_\lambda$ and points $z \in \{0,1\}^j$ for some $j < d$, let

$$t_z(x) = \begin{cases} t(x) & \text{if } x_i = z_i, \text{ for all } i \in [j]; \\ 0 & \text{otherwise,} \end{cases}$$

and define $T_z = \sum_{x \in \{0,1\}^d} t_z(x)$. Fix $b \in \{0,1\}$, and define the protocol $\left[\mathcal{P}_0(t), \mathcal{P}_1(t), \mathcal{V}^t\right]_b$ in Protocol 2.

$[\mathcal{P}_0^t, \mathcal{P}_1^t, \mathcal{V}^t]_b(d)$

1. $\mathcal{P}_b$: Let $z \leftarrow \emptyset$. Send $(\widehat{T}_z, \widehat{T}_{z0}, \widehat{T}_{z1}) \leftarrow (T_z, T_{z0}, T_{z1})$ to $\mathcal{V}$.
2. $\mathcal{V}$: If $\widehat{T}_z \neq \widehat{T}_{z0} + \widehat{T}_{z1}$ then output $\perp$. Otherwise, send $(\widehat{T}_z, \widehat{T}_{z0}, \widehat{T}_{z1})$ to $\mathcal{P}_{1-b}$.
3. $\mathcal{P}_{1-b}$: If there exists $j \in \{0,1\}$ such that $T_{zj} \neq \widehat{T}_{zj}$ then send j to $\mathcal{V}$. Otherwise send 0.
4. $\mathcal{V}$: If $|z| < d-1$ then send j to $\mathcal{P}_b$ and repeat the protocol with $\widehat{T}_z \leftarrow \widehat{T}_{zj}$ and $z \leftarrow zj$. Otherwise, accept if and only if $t(zj) = \widehat{T}_{zj}$.

Protocol 2: certifiable sum

Assume without loss of generality that $\mathcal{P}_0$ is honest. First, we prove that for all $\mathcal{P}_1^*$ the protocol $\left[\mathcal{P}_0^t, \mathcal{P}_1^*, \mathcal{V}^t\right]_0$ accepts. The proof proceeds by induction on d. For the base case, suppose $d = 1$. Then, since $\mathcal{P}_0$ is honest, $\widehat{T} = T$, $\widehat{T}_0 = T_0$, and $\widehat{T}_1 = T_1$. Thus, for all $j \in \{0,1\}^d$ we have $t(j) = \widehat{T}_j$, and thus the verifier always accepts. Now, suppose the claim holds up to some $d \in \mathbb{N}$ and consider the case of $d+1$. By the same argument as the base case, after the first round of the protocol we have $\widehat{T}_j = T_j$. Since $t_j(x) = 0$ whenever $x_1 \neq j$, round two of the protocol is identical to executing the protocol with function $t'_j : \{0,1\}^d \to \mathbb{Q}_\lambda$ given by $t'_j(x) = t_j(jx)$. Since the domain of t'_j is d, the inductive hypothesis implies that the verifier will accept.

Next, we prove that $\left[\mathcal{P}_0^t, \mathcal{P}_1^*, \mathcal{V}^t\right]_1$ rejects for all $\mathcal{P}_1^*$ that send $\widehat{T} \neq T$ in the first round. As before, the proof follows by induction on d. Consider the base case of $d = 1$. If $\widehat{T} \neq T$ and $\widehat{T}_0 + \widehat{T}_1 = \widehat{T}$,

then either $\widehat{T}_0 \neq T_0$ or $\widehat{T}_1 \neq T_1$. Since $\mathcal{P}_0$ is assumed to be honest, it will send $j \in \{0,1\}$ such that $\widehat{T}_j \neq T_j = t(j)$. Since $t(j) \neq \widehat{T}_j$ the verifier will reject. Now, suppose the claim holds up to some $d \in \mathbb{N}$, and consider the case of $d+1$. If $\widehat{T} \neq T$ in the first round, then $\widehat{T}_j \neq T_j$ for some $j \in \{0,1\}$. Since $\mathcal{P}_0$ is assumed to be honest, it will send $\mathcal{V}$ the $j \in \{0,1\}$ such that $\widehat{T}_j \neq T_j$. Observe that the next round of the protocol is identical to executing the protocol with the function $t'_j : \{0,1\}^d \to \mathbb{Q}_\lambda$ given by $t'_j(x) = t_j(jx)$, and with $\widehat{T}' \neq T' = \sum_{x \in \{0,1\}^d} t'_j(x)$. By the inductive hypothesis, the verifier rejects.

To complete the proof, define the protocol $\left[\mathcal{P}_0^t, \mathcal{P}_1^t, \mathcal{V}^t\right]$ as follows: for each $b \in \{0,1\}$ run protocols $\left[\mathcal{P}_0^t, \mathcal{P}_1^t, \mathcal{V}^t\right]_b$, and return the $\widehat{T}$ from the first round of an accepting execution. By the above arguments, the protocol $\left[\mathcal{P}_0^t, \mathcal{P}_1^t, \mathcal{V}^t\right]$ always outputs T.

To see why the communication complexity holds, observe that the numerator in $\sum_{x \in \{0,1\}^d} t(x)$ can be at most $2^{d+2\lambda}$, and that the denominator can be at most $2^{\lambda \cdot d}$. Thus, $\widehat{T}_z$ will require at most $O(d\lambda)$ bits to send. Since the protocol uses $2d$ rounds, and in each round $O(d\lambda)$ bits are required to transmit $(\widehat{T}_z, \widehat{T}_{z0}, \widehat{T}_{z1})$, it immediately follows that the overall communication complexity is $\lambda \operatorname{poly} d$. The runtime and query complexity of the verifier follows by inspection of Protocol 2. □

Proof of Claim 3.5. Let $\prec$ denote $\prec_d$. At a high level, our protocol works as follows: First, the verifier requests $\widehat{x} = S_i$ from one of the provers, and using its membership query oracle to S, the verifier checks that $\widehat{x} \in S$. Then, the verifier runs the certifiable sum protocol with $t(x) = \mathbb{1}[x \in S \wedge x \prec \widehat{x}]$ to compute $s = |\{x : x \prec \widehat{x}\}|$. If $s \neq i-1$ then the verifier rejects. Fix $b \in \{0,1\}$, and define the protocol $[\mathcal{P}_0, \mathcal{P}_1, \mathcal{V}]_b$ as follows:

$\left[\mathcal{P}_0^{\prec}(S), \mathcal{P}_1^{\prec}(S), \mathcal{V}^{S,\prec}\right]_b(d,i)$

1. $\mathcal{P}_b$: Send $\widehat{x} = S_i$ to $\mathcal{V}$.
2. $\mathcal{V}$: If $\widehat{x} \notin S$ then output reject. Otherwise, send $\widehat{x}$ to $\mathcal{P}_{1-b}$.
3. $\mathcal{V}$: Provide query access to $A = \{x \in S : x \prec \widehat{x}\}$[a] using oracle for S and $\prec$. Execute certifiable sum (Lemma 3.2) with $t = \mathbb{1}[x \in A]$ and $\lambda \leftarrow d$. Accept if $\sum_{x \in \{0,1\}^d} t(x) = |A| = i-1$ and reject otherwise.

[a] The verifier need not construct A explicitly

Protocol 3: certifiable index

Without loss of generality assume $\mathcal{P}_0$ is honest. Then $\widehat{x} = S_i$ and hence $|A| = i-1$. By Lemma 3.2, protocol $[\mathcal{P}_0, \mathcal{P}_1^*, \mathcal{V}]_0$ outputs accept. Next, consider $[\mathcal{P}_0, \mathcal{P}_1^*, \mathcal{V}]_b$. If $\mathcal{P}_1^*$ sends $\widehat{x} \neq S_i$, then, if $\widehat{x} \notin S$, the protocol rejects in Step 2. On the other hand, if $\widehat{x} \in S$ then $|A| = |\{x \in S \mid x \prec \widehat{x}\} \neq i-1$. By Lemma 3.2, the sum $\sum_{x \in \{0,1\}^d} t(x) = |A| \neq i-1$, and hence the protocol outputs reject. To complete the proof, define the protocol $[\mathcal{P}_0, \mathcal{P}_1, \mathcal{V}]$ as follows: for each $b \in \{0,1\}$ run protocols $[\mathcal{P}_0, \mathcal{P}_1, \mathcal{V}]_b$, and return the $\widehat{x}$ from the first round of an accepting execution. By the above argument, the protocol $[\mathcal{P}_0, \mathcal{P}_1, \mathcal{V}]$ always outputs S_i. The communication complexity and runtime guarantees follow by inspection of Protocol 3 and from the guarantees of Lemma 3.2. □

3.2 Refereed query delegation

In this section, we show that a protocol where all parties are given query access to a deterministic oracle $\mathcal{O}$ can be modified to a protocol where the verifier offloads nearly all of its queries to the provers and only makes a single query to $\mathcal{O}$. The modification essentially preserves the guarantees of the original protocol (up to a small cost in communication complexity). At a high level, the modification works as follows: each time the verifier would make a query to the oracle, it instead has each prover make that query to the oracle. Each prover then sends the query answer to the verifier. If the query answers match, the verifier continues the protocol with this answer; if the query answers do not match, then the verifier issues a single query to the oracle to figure out the true query answer, and then continues the protocol using only the query answers from the correct prover going forward.

Lemma 3.8 (Refereed query delegation)**.** *Fix a deterministic oracle $\mathcal{O}$. Suppose there exists a protocol $[\mathcal{P}_0, \mathcal{P}_1, \mathcal{V}]$ that for all inputs $\kappa \in \mathbb{R}$ and $\lambda \in \mathbb{N}$ has communication complexity $C(\kappa, \lambda)$, verifier runtime $T_{\mathcal{V}}(\kappa, \lambda)$, verifier query complexity $q_{\mathcal{V}}(\kappa, \lambda)$, prover runtime $T_{\mathcal{P}}(\kappa, \lambda)$, and prover query complexity $q_{\mathcal{P}}(\kappa, \lambda)$. Additionally, assume all queries to $\mathcal{O}$ and their answers can be specified using at most λ bits.*

Then there exists a protocol $\left[\widetilde{\mathcal{P}}_0, \widetilde{\mathcal{P}}_1, \widetilde{\mathcal{V}}\right]$ such that for all $b \in \{0,1\}$ and $\widetilde{\mathcal{P}}^*_{1-b}$ there exists a $\mathcal{P}^*_{1-b}$ such that

$$\left[\widetilde{\mathcal{P}}^{\mathcal{O}}_b, \widetilde{\mathcal{P}}^*_{1-b}, \widetilde{\mathcal{V}}^{\mathcal{O}}\right](\kappa, \lambda) = \left[\mathcal{P}^{\mathcal{O}}_b, \mathcal{P}^*_{1-b}, \mathcal{V}^{\mathcal{O}}\right](\kappa, \lambda).$$

Moreover, $\left[\widetilde{\mathcal{P}}_0, \widetilde{\mathcal{P}}_1, \widetilde{\mathcal{V}}\right]$ has communication complexity $C(\kappa, \lambda) + 2\lambda \cdot q_{\mathcal{V}}(\kappa, \lambda)$, verifier runtime $T_{\mathcal{V}}(\kappa, \lambda)$, prover runtime $T_{\mathcal{P}}(\kappa, \lambda) + q_{\mathcal{V}}(\kappa, \lambda)$, prover query complexity $q_{\mathcal{P}}(\kappa, \lambda) + q_{\mathcal{V}}(\kappa, \lambda)$, and verifier query complexity at most 1.

Proof. Consider the following protocol:

$\left[\widetilde{\mathcal{P}}^{\mathcal{O}}_0, \widetilde{\mathcal{P}}^{\mathcal{O}}_1, \widetilde{\mathcal{V}}^{\mathcal{O}}\right](\kappa, \lambda)$

1. Simulate $[\mathcal{P}^{\mathcal{O}}_0, \mathcal{P}^{\mathcal{O}}_1, \mathcal{V}^{\mathcal{O}}](\kappa, \lambda)$, except answer $\mathcal{V}$'s queries to $\mathcal{O}$ using the following procedure:
   (a) $\widetilde{\mathcal{V}}$ asks both $\widetilde{\mathcal{P}}_0$ and $\widetilde{\mathcal{P}}_1$ to answer the query using their access to $\mathcal{O}$.
   (b) If both provers agree, continue the protocol using this query answer.
   (c) If the provers disagree, $\widetilde{\mathcal{V}}$ makes a single query to $\mathcal{O}$ to determine the lying prover $(1-b)$, and then uses the answers from $\widetilde{\mathcal{P}}_b$ for all subsequent queries to $\mathcal{O}$.

Protocol 4: refereed query delegation

The communication complexity, runtime, and query complexity follow immediately from the fact that Protocol 4 simply runs $[\mathcal{P}_0, \mathcal{P}_1, \mathcal{V}]$ once, and has each prover make at most $q_{\mathcal{V}}(\kappa, \lambda)$ additional queries to $\mathcal{O}$.

Consider the protocol above run with (malicious) prover $\widetilde{\mathcal{P}}^*_{1-b}$. We show there exists a prover $\mathcal{P}^*_{1-b}$ such that

$$\left[\widetilde{\mathcal{P}}^{\mathcal{O}}_b, \widetilde{\mathcal{P}}^*_{1-b}, \widetilde{\mathcal{V}}^{\mathcal{O}}\right](\kappa, \lambda) = \left[\mathcal{P}^{\mathcal{O}}_b, \mathcal{P}^*_{1-b}, \mathcal{V}^{\mathcal{O}}\right](\kappa, \lambda).$$

Let $\mathcal{P}^*_{1-b}$ be the prover that is identical to $\widetilde{\mathcal{P}}^*_{1-b}$ (except it simulates query requests from $\mathcal{V}$ and does not actually send query answers to $\mathcal{V}$).[4] We see that, if $\mathcal{V}$ is simulated using queries to $\mathcal{O}$ that are answered correctly, then Protocol 4 has exactly the same distribution of outputs as $\left[\mathcal{P}^{\mathcal{O}}_b, \mathcal{P}^*_{1-b}, \mathcal{V}^{\mathcal{O}}\right](\kappa, \lambda)$.

To complete the proof, we show that verifier $\widetilde{\mathcal{V}}$ correctly answers all of $\mathcal{V}$'s queries to $\mathcal{O}$ using at most 1 query to $\mathcal{O}$. First recall that the honest prover $\widetilde{\mathcal{P}}_b$ always answers queries truthfully. Additionally, $\widetilde{\mathcal{V}}$ determines which prover is telling the truth at the first instance on which $\widetilde{\mathcal{P}}^*_{1-b}$ and $\widetilde{\mathcal{P}}^{\mathcal{O}}_b$ disagree by making a single query to $\mathcal{O}$. After this query $\widetilde{\mathcal{V}}$ only uses answers provided by the honest prover, and hence all of the answers it provides to $\mathcal{V}$ are correct. □

4 Refereed learning protocols

In this section we prove our two main results. In Section 4.1 we construct a $(1+\varepsilon)$-refereed learning protocol for the zero-one loss function; in Section 4.2 we construct a $(3+\varepsilon)$-refereed learning protocol for metric loss functions.

Throughout this section we use the following simpler version of Definition 2.3, which only has a multiplicative slack term and fixes the oracle access model. Recall that $Q_{\mathcal{D}}$ denotes the probability mass function of a distribution $\mathcal{D}$.

Definition 4.1 (Refereed learning protocol—multiplicative error with fixed oracles)**.** *Fix a range $\mathcal{Y}$ with metric ℓ, dimension $d \in \mathbb{N}$, slack $\alpha \geq 1$, and soundness error $\beta \geq 0$. Let $\mathcal{H} \subseteq \left\{h : \{0,1\}^d \to \mathcal{Y}\right\}$ and $\mathbb{D} \subseteq \left\{\mathcal{D} \mid \mathrm{supp}(\mathcal{D}) \subseteq \{0,1\}^d\right\}$. A protocol $[\mathcal{P}_0, \mathcal{P}_1, \mathcal{V}]$ is an (α, β)*-refereed learning protocol for $\mathcal{H}$ and $\mathbb{D}$ with respect to ℓ, *if for all distributions $\mathcal{D} \in \mathbb{D}$, functions $f : \{0,1\}^d \to \mathcal{Y}$ and $h_0, h_1 \in \mathcal{H}$, the following holds:*

- *For all $b \in \{0,1\}$ and $\mathcal{P}^*_{1-b}$, the bit $\rho \leftarrow \left[\mathcal{P}_b^{h_0,h_1,Q_{\mathcal{D}}}, \mathcal{P}^*_{1-b}, \mathcal{V}^{f,h_0,h_1,Q_{\mathcal{D}}}\right]$ satisfies*

$$\Pr\left[\mathcal{L}_{\mathcal{D}}(h_\rho, f \mid \ell) \leq \alpha \cdot \min_{s \in \{0,1\}} \mathcal{L}_{\mathcal{D}}(h_s, f \mid \ell)\right] \geq 1 - \beta,$$

where the randomness is over the coins of the verifier, the honest prover, and the oracles.

The provers in Definition 4.1 do not have query access to f, and in the protocols of Theorems 4.2 and 4.4 the verifier makes a number of queries to f that depends only on α. In Section 4.3 we describe how, if we also give the provers query access to f, the verifier can offload all but a single query to the provers—that is, we show how the protocols can be modified so that the verifier makes at most one query to either h_0, h_1, f, or $Q_{\mathcal{D}}$.

4.1 Refereed learning for zero-one loss

We first consider refereed learning protocols for the special case of the *zero-one metric* defined by $\ell_{\mathsf{zo}}(y, y') = 1$ if $y \neq y'$ and 0 otherwise. Theorem 4.2 states that for all $\varepsilon, \beta > 0$ there exists an (α, β)-refereed learning protocol for $\alpha = 1 + \varepsilon$, with respect to ℓ_{zo}.

[4] Morally, think of $\mathcal{P}^*_{1-b}$ and $\widetilde{\mathcal{P}}^*_{1-b}$ as identical. However, a subtlety arises that the behavior of $\widetilde{\mathcal{P}}^*_{1-b}$ may depend on the queries it answers for $\mathcal{V}$—e.g., $\widetilde{\mathcal{P}}^*_{1-b}$ may execute strategy "a" when asked an oracle query by $\mathcal{V}$, and strategy "b" when $\mathcal{V}$ does not ask any oracle queries. To ensure that $\mathcal{P}^*_{1-b}$ executes the correct malicious prover strategy, it simulates $\widetilde{\mathcal{P}}^*_{1-b}$ with the queries from $\mathcal{V}$.

Theorem 4.2 (Refereed learning protocol for zero-one loss)**.** *Fix range $\mathcal{Y}$ and $\lambda \in \mathbb{N}$. There exists a protocol $[\mathcal{P}_0, \mathcal{P}_1, \mathcal{V}]$ that, for all inputs $d \in \mathbb{N}$ and $\varepsilon, \beta > 0$, is a $(1+\varepsilon, \beta)$-refereed learning protocol for $\mathfrak{F}$ and $\mathbb{D}_\lambda$ with respect to ℓ_{zo}. The protocol has communication complexity and verifier runtime $\lambda\big(1+\frac{1}{\varepsilon}\big)^2 \log\big(1+\frac{1}{\varepsilon}\big) \log\frac{1}{\beta} \cdot \operatorname{poly} d = \widetilde{O}_{\lambda,\beta}\big(\big(1+\frac{1}{\varepsilon}\big)^2 \cdot \operatorname{poly} d\big)$, and the verifier makes $O\big(\big(1+\frac{1}{\varepsilon^2}\big)\log\frac{1}{\beta}\big)$ queries to f.*

Proof of Theorem 4.2. We define the protocol $[\mathcal{P}_0, \mathcal{P}_1, \mathcal{V}]$ in Protocol 5.

$$\Big[\mathcal{P}_0^{h_0,h_1,Q_{\mathcal{D}}}, \mathcal{P}_1^{h_0,h_1,Q_{\mathcal{D}}}, \mathcal{V}^{f,h_0,h_1,Q_{\mathcal{D}}}\Big](d,\varepsilon,\beta)$$

1. $\mathcal{P}_0, \mathcal{P}_1$: Let $S \leftarrow \big\{x \in \{0,1\}^d \mid h_0(x) \neq h_1(x)\big\}$.
2. $\mathcal{V}$: Obtain $p_S = \mathcal{D}(S)$ using certifiable sum (Lemma 3.2) with $t(x) = \mathbb{1}[x \in S] \cdot \mathcal{D}(x)$ and $\lambda \leftarrow \lambda$. If $p_S = 0$ then output $\rho \sim \{0,1\}$.
3. $\mathcal{V}$: Set $\delta \leftarrow \frac{\varepsilon}{4(2+\varepsilon)}$ and $m \leftarrow \frac{\log 1/\beta}{\delta^2}$. Execute certifiable sample (Lemma 3.1) with distribution $\mathcal{D}|_S$ to draw m samples $x_1, \ldots, x_m \sim \widehat{\mathcal{D}}$ with distance parameter δ.
4. $\mathcal{V}$: Query f on $x_1, \ldots, x_m$ and output $\rho = \arg\min_{s\in\{0,1\}} |\{i \in [m] : h_s(x_i) \neq f(x_i)\}|$.

Protocol 5: refereed learning for zero-one loss

First, we argue that Protocol 5 satisfies the soundness condition of Definition 4.1. Assume without loss of generality that $\mathcal{L}_{\mathcal{D}}(h_1, f \mid \ell_{\mathsf{zo}}) > (1+\varepsilon) \cdot \mathcal{L}_{\mathcal{D}}(h_0, f \mid \ell_{\mathsf{zo}})$. Since ℓ_{zo} is the zero-one metric, this is equivalent to the assumption that $\Pr_{\mathcal{D}}[h_1(x) \neq f(x)] > (1+\varepsilon) \cdot \Pr_{\mathcal{D}}[h_0(x) \neq f(x)]$. To prove that the verifier in Protocol 5 outputs the correct bit, we prove Claim 4.3, which roughly states that for $x \sim \mathcal{D}|_S$, we have $h_1(x) \neq f(x)$ with probability at least $\frac{1}{2} + \frac{\varepsilon}{1+\varepsilon}$.

Claim 4.3. *Fix $d \in \mathbb{N}$, functions $f, h_0, h_1 : \{0,1\}^d \to \mathcal{Y}$, distribution $\mathcal{D}$ over $\{0,1\}^d$, and $\varepsilon > 0$. Assume $\Pr_{\mathcal{D}}[h_1(x) \neq f(x)] > (1+\varepsilon) \cdot \Pr_{\mathcal{D}}[h_0(x) \neq f(x)]$, and let $S = \{x : h_0(x) \neq h_1(x)\}$. Then,*

$$\Pr_{x\sim\mathcal{D}|_S}[h_0(x) \neq f(x)] < \frac{1}{2} - \frac{\varepsilon}{2(2+\varepsilon)}.$$

Proof. By the hypothesis on h_0, h_1, and f, and the law of total probability,

$$\begin{aligned}
\varepsilon \cdot \Pr_{x\sim\mathcal{D}}[h_0(x) \neq f(x)] &< \Pr_{x\sim\mathcal{D}}[h_1(x) \neq f(x)] - \Pr_{x\sim\mathcal{D}}[h_0(x) \neq f(x)] \qquad (1)\\
&= \left(\Pr_{x\sim\mathcal{D}}[h_1(x) \neq f(x) \mid x \in S] - \Pr_{x\sim\mathcal{D}}[h_0(x) \neq f(x) \mid x \in S]\right) \cdot \Pr_{x\sim\mathcal{D}}[x \in S]\\
&+ \left(\Pr_{x\sim\mathcal{D}}[h_1(x) \neq f(x) \mid x \notin S] - \Pr_{x\sim\mathcal{D}}[h_0(x) \neq f(x) \mid x \notin S]\right) \cdot \Pr_{x\sim\mathcal{D}}[x \notin S].
\end{aligned}$$

Since $h_0(x) = h_1(x)$ for all $x \notin S$, the difference in the second term of the sum is

$$\Pr_{x\sim\mathcal{D}}[h_1(x) \neq f(x) \mid x \notin S] - \Pr_{x\sim\mathcal{D}}[h_0(x) \neq f(x) \mid x \notin S] = 0.$$

so rearranging (1) yields

$$\Pr_{x\sim\mathcal{D}}[h_1(x) \neq f(x) \mid x \in S] - \Pr_{x\sim\mathcal{D}}[h_0(x) \neq f(x) \mid x \in S] > \frac{\varepsilon \cdot \Pr_{x\sim\mathcal{D}}[h_0(x) \neq f(x)]}{\Pr_{x\sim\mathcal{D}}[x \in S]}. \qquad (2)$$

By the definition of S, the events $h_0(x) \neq f(x)$ and $h_1(x) \neq f(x)$ are disjoint, and hence

$$\begin{aligned}
1 &= \Pr_{\mathcal{D}|_S}[h_1(x) \neq f(x)] + \Pr_{\mathcal{D}|_S}[h_0(x) \neq f(x)] \\
&= \Pr_{\mathcal{D}|_S}[h_1(x) \neq f(x)] - \Pr_{\mathcal{D}|_S}[h_0(x) \neq f(x)] + 2\Pr_{\mathcal{D}|_S}[h_0(x) \neq f(x)] \\
&> \frac{\varepsilon \cdot \Pr_{x\sim\mathcal{D}}[h_0(x) \neq f(x)]}{\Pr_{x\sim\mathcal{D}}[x \in S]} + 2\Pr_{\mathcal{D}|_S}[h_0(x) \neq f(x)] \\
&\geq (2+\varepsilon)\Pr_{x\sim\mathcal{D}|_S}[h_0(x) \neq f(x)]
\end{aligned}$$

where the second to last inequality follows from (2), and the last equality follows by applying the law of total probability to the numerator. Rearranging terms yields the desired conclusion. □

To complete the proof of Theorem 4.2, observe that by Lemma 3.2, the verifier correctly obtains $p_S = \mathcal{D}(S)$. Since $\mathcal{D}|_S(x) = \frac{\mathcal{D}(x)\cdot\mathbb{1}[x\in S]}{\mathcal{D}(S)}$, the verifier can provide query access to $Q_{\mathcal{D}|_S}$, the probability mass function of $\mathcal{D}|_S$. Thus, by Lemma 3.1, we have $d_{\mathrm{TV}}\big(\widehat{\mathcal{D}}, \mathcal{D}|_S\big) \leq \delta$ and therefore, by Claim 4.3 and the definition of S, we have

$$\Pr_{x\sim\widehat{\mathcal{D}}}[h_0(x) \neq f(x)] < \frac{1}{2} - \frac{\varepsilon}{2(2+\varepsilon)} + \delta.$$

To see why the verifier outputs 0 with probability at least $1-\beta$, let $\widehat{p} = \frac{1}{m}\sum_{i\in[m]}\mathbb{1}[h_0(x_i) \neq f(x_i)]$. Since $\mathbb{E}[\widehat{p}] = \Pr[h_0(x) \neq f(x)]$, Hoeffding's inequality and our setting of m in Protocol 5 implies

$$\Pr\left[|\widehat{p} - \Pr[h_0(x) \neq f(x)]| \geq \delta\right] \leq 2\exp\left(-2m/\delta^2\right) < \beta.$$

It follows that $\widehat{p} < \frac{1}{2} - \frac{\varepsilon}{2(2+\varepsilon)} + 2\delta \leq \frac{1}{2}$ with probability at least $1-\beta$, and therefore $\mathcal{V}$ will output $\rho = 0$ with probability at least $1-\beta$. The runtime and communication complexity follow by inspection of Protocol 5, and from the guarantees of Lemmas 3.1 and 3.2. □

4.2 Refereed learning for metric loss functions

We next consider refereed learning protocols for any metric loss function (Definition 2.2). Theorem 4.4 states that for all $\varepsilon, \beta > 0$ there exists an (α, β)-refereed learning protocol for $\alpha = 3+\varepsilon$, with respect to the chosen metric loss function. While the slack parameter $\alpha = 3+\varepsilon$ is worse (as compared to $\alpha = 1+\varepsilon$), this general protocol has the same time, communication, and query complexity guarantees as the protocol in Theorem 4.2.

Theorem 4.4 (Refereed learning protocol for general loss functions)**.** *Fix range $\mathcal{Y}$, $\lambda \in \mathbb{N}$, and λ-precise metric ℓ on $\mathcal{Y}$. There exists a protocol $[\mathcal{P}_0, \mathcal{P}_1, \mathcal{V}]$ that, for all inputs $d \in \mathbb{N}$ and $\varepsilon, \beta > 0$, is a $(3+\varepsilon, \beta)$-refereed learning protocol for $\mathfrak{F}$ and $\mathbb{D}_\lambda$ with respect to ℓ. The protocol has communication complexity and verifier runtime $\lambda\big(1+\frac{1}{\varepsilon^2}\big)\log\big(1+\frac{1}{\varepsilon}\big)\log\frac{1}{\beta}\cdot\operatorname{poly} d = \widetilde{O}_{\lambda,\beta}\big(\big(1+\frac{1}{\varepsilon^2}\big)\cdot\operatorname{poly} d\big)$, and the verifier makes $O\big(\big(1+\frac{1}{\varepsilon^2}\big)\log\frac{1}{\beta}\big)$ queries to f.*

Proof of Theorem 4.4. In order to construct our protocol, we first introduce a scaled version of the distribution $\mathcal{D}$ called $\mathcal{D}_\ell^{h_0,h_1}$. Intuitively, $\mathcal{D}_\ell^{h_0,h_1}$ assigns more probability mass to points x where $\ell\big(h_0(x), h_1(x)\big)$ is large. Since whenever $\ell\big(h_0(x), h_1(x)\big)$ is large, it must be the case that either $\ell\big(h_0(x), f(x)\big)$ or $\ell\big(h_1(x), f(x)\big)$ is large, sampling points x with higher probability where $\ell\big(h_0(x), h_1(x)\big)$ is larger allows the verifier to distinguish h_0 and h_1 more easily.

Definition 4.5 (Loss-rescaled distribution)**.** *For all $d \in \mathbb{N}$, distributions $\mathcal{D}$ over $\{0,1\}^d$, sets $\mathcal{Y}$ with metric ℓ, and functions $h_0, h_1 : \{0,1\}^d \to \mathcal{Y}$, define the* loss-rescaled distribution $\mathcal{D}_\ell^{h_0,h_1}$ *via the density*

$$\mathcal{D}_\ell^{h_0,h_1}(x) := \mathcal{D}(x) \cdot \frac{\ell\big(h_0(x), h_1(x)\big)}{\mathbb{E}_{x\sim\mathcal{D}}\left[\ell\big(h_0(x), h_1(x)\big)\right]}.$$

We define the protocol $[\mathcal{P}_0, \mathcal{P}_1, \mathcal{V}]$ in Protocol 6.

$$\left[\mathcal{P}_0^{h_0,h_1,Q_\mathcal{D}}, \mathcal{P}_1^{h_0,h_1,Q_\mathcal{D}}, \mathcal{V}^{f,h_0,h_1,Q_\mathcal{D}}\right](d,\varepsilon,\beta)$$

1. $\mathcal{V}$: Execute certifiable sum (Lemma 3.2) with $t(x) := \ell(h_0(x), h_1(x)) \cdot \mathcal{D}(x)$ and $\lambda \leftarrow 2\lambda$ to compute $\mu \leftarrow \mathbb{E}_{x\sim\mathcal{D}}\left[\ell\big(h_0(x), h_1(x)\big)\right]$. If $\mu = 0$ then output $\rho \sim \{0,1\}$.
2. $\mathcal{V}$: Set $\delta \leftarrow \frac{\varepsilon}{4(2+\varepsilon)}$ and $m \leftarrow \frac{\log 1/\beta}{\delta^2}$. Execute certifiable sample (Lemma 3.1) with distribution $\mathcal{D}_\ell^{h_0,h_1}$ to draw m samples $x_1, \ldots, x_m \sim \widehat{\mathcal{D}}$ with distance parameter δ.
3. $\mathcal{V}$: Query f on $x_1, \ldots, x_m$ and for each $b \in \{0,1\}$ let

$$\widehat{R}_b \leftarrow \frac{1}{m} \sum_{i\in[m]} \frac{\ell\big(h_b(x_i), f(x_i)\big)}{\ell\big(h_0(x_i), f(x_i)\big) + \ell\big(h_1(x_i), f(x_i)\big)}.$$

4. $\mathcal{V}$: Output $\rho \leftarrow \arg\min_{b\in\{0,1\}} \widehat{R}_b$

Protocol 6: refereed learning for metric loss

In order to prove the soundness condition, we argue that the bit ρ output by the verifier satisfies $\mathcal{L}_\mathcal{D}(h_\rho, f) \le (3+\varepsilon)\mathcal{L}_\mathcal{D}(h_{1-\rho}, f)$ with probability at least $1-\beta$. First, recall that by Lemmas 3.1 and 3.2, the verifier $\mathcal{V}$ obtains the expectation $\mathbb{E}_{x\sim\mathcal{D}}\left[\ell(h_0(x), h_1(x))\right]$, and m samples $x_1, \ldots, x_m$ from a distribution $\widehat{\mathcal{D}}$ such that $d_{\mathrm{TV}}\big(\widehat{\mathcal{D}}, \mathcal{D}_\ell^{h_0,h_1}\big) \le \delta$. We assume without loss of generality that $\mathbb{E}_{x\sim\mathcal{D}}\left[\ell\big(h_0(x), h_1(x)\big)\right] \neq 0$ since otherwise $h_0 = h_1$.

The proof of correctness proceeds in two major steps. First, in Claim 4.6 we will show that for all $b \in \{0,1\}$ the statistic defined by

$$r_b(x) := \frac{\ell\big(h_b(x), f(x)\big)}{\ell\big(h_0(x), f(x)\big) + \ell\big(h_1(x), f(x)\big)} \quad \text{and} \quad R_b := \mathop{\mathbb{E}}_{x\sim\mathcal{D}_\ell^{h_0,h_1}}[r_b(x)], \tag{3}$$

is less than $\frac{1}{2} - \varepsilon$ whenever $\mathcal{L}_\mathcal{D}(h_{1-b}, f \mid \ell) > (3+\varepsilon) \cdot \mathcal{L}_\mathcal{D}(h_b, f \mid \ell)$. Then, we will argue that the estimate $\widehat{R}_b$ computed by $\mathcal{V}$ in Protocol 6 is concentrated around $\mathbb{E}\left[\widehat{R}_b\right]$, and that R_b and $\mathbb{E}\left[\widehat{R}_b\right]$ are close together. Combining the above with the fact that $R_0 + R_1 = 1$ suffices to complete the proof.

Claim 4.6. *Fix a set $\mathcal{Y}$ with metric ℓ, dimension $d \in \mathbb{N}$, distribution $\mathcal{D}$ over $\{0,1\}^d$, functions $f, h_0, h_1 : \{0,1\}^d \to \mathcal{Y}$, and $\varepsilon > 0$. For all $b \in \{0,1\}$, if $\mathcal{L}_\mathcal{D}(h_{1-b}, f \mid \ell) > (3+\varepsilon) \cdot \mathcal{L}_\mathcal{D}(h_b, f \mid \ell)$ then*

$$R_b < \frac{1}{2} - \frac{\varepsilon}{2(2+\varepsilon)},$$

where R_b is defined in (3).

Proof. To avoid cluttered expressions, let $\ell_b(x) := \ell(h_b(x), f(x))$, let $\Delta(x) := \ell(h_0(x), h_1(x))$ and let $\mu := \mathbb{E}_{x\sim\mathcal{D}}[\Delta(x)]$. First, by the definition of R_b and $\mathcal{D}_\ell^{h_0,h_1}$, we have

$$R_b = \mathop{\mathbb{E}}_{x\sim\mathcal{D}_\ell^{h_0,h_1}}\left[\frac{\ell_b(x)}{\ell_0(x)+\ell_1(x)}\right] = \mathop{\mathbb{E}}_{x\sim\mathcal{D}}\left[\frac{\ell_b(x)}{\ell_0(x)+\ell_1(x)}\cdot\frac{\Delta(x)}{\mu}\right] \le \frac{\mathbb{E}_{x\sim\mathcal{D}}\left[\ell_b(x)\right]}{\mu},$$

where the last inequality follows since $\Delta(x) \le \ell_0(x) + \ell_1(x)$ by the triangle inequality. Next, we apply the assumption that $\mathcal{L}_\mathcal{D}(h_{1-b}, f \mid \ell) > (3+\varepsilon)\cdot\mathcal{L}_\mathcal{D}(h_b, f \mid \ell)$ to show that μ can be lower bounded as

$$\mu = \mathop{\mathbb{E}}_{x\sim\mathcal{D}}\left[\Delta(x)\right] \ge \mathop{\mathbb{E}}_{x\sim\mathcal{D}}\left[\ell_{1-b}(x) - \ell_b(x)\right] > (2+\varepsilon)\cdot\mathop{\mathbb{E}}_{x\sim\mathcal{D}}\left[\ell_b(x)\right].$$

Substituting this bound on μ into our bound on R_b gives us $R_b < \frac{1}{2+\varepsilon} = \frac{1}{2} - \frac{\varepsilon}{2(2+\varepsilon)}$. □

Now, since $r_b(x) \in [0,1]$ and each x_i in Protocol 6 is sampled from $\widehat{\mathcal{D}}$ independently, Hoeffding's inequality implies that

$$\Pr\left[\left|\widehat{R}_b - \mathbb{E}\left[\widehat{R}_b\right]\right| \ge \delta\right] \le 2\exp\left(-2m\delta^2\right) < \frac{\beta}{2}. \tag{4}$$

Next, we bound the distance between R_b and $\mathbb{E}\left[\widehat{R}_b\right]$. Recall that $R_b = \mathbb{E}_{x\sim\mathcal{D}_\ell^{h_0,h_1}}\left[r_b(x)\right]$, and that $\mathbb{E}\left[\widehat{R}_b\right] = \mathbb{E}_{x\sim\widehat{\mathcal{D}}}\left[r_b(x)\right]$. Since $r_b(x) \in [0,1]$ and $d_{\mathrm{TV}}\left(\widehat{\mathcal{D}}, \mathcal{D}_\ell^{h_0,h_1}\right) \le \delta$, we have

$$\begin{aligned}
\left|\mathbb{E}\left[\widehat{R}_b\right] - R_b\right| &= \left|\mathop{\mathbb{E}}_{x\sim\widehat{\mathcal{D}}}\left[r_b(x)\right] - \mathop{\mathbb{E}}_{x\sim\mathcal{D}_\ell^{h_0,h_1}}\left[r_b(x)\right]\right| \\
&= \left|\sum_{x\in\{0,1\}^d}\left(\widehat{\mathcal{D}}(x) - \mathcal{D}_\ell^{h_0,h_1}(x)\right)\cdot r_b(x)\right| \\
&\le \left|\sum_{x\in\{0,1\}^d}\left(\widehat{\mathcal{D}}(x) - \mathcal{D}_\ell^{h_0,h_1}(x)\right)\cdot\left(r_b(x) - \frac{1}{2}\right)\right| + \left|\sum_{x\in\{0,1\}^d}\left(\widehat{\mathcal{D}}(x) - \mathcal{D}_\ell^{h_0,h_1}(x)\right)\cdot\frac{1}{2}\right| \\
&\le d_{\mathrm{TV}}(\widehat{\mathcal{D}}, \mathcal{D}_\ell^{h_0,h_1}) \le \delta.
\end{aligned}$$

Combining (4) and the above bound yields $\left|\widehat{R}_b - R_b\right| \le 2\delta$ for each $b \in \{0,1\}$ with probability at least $1-\beta$. To complete the proof, suppose $\mathcal{L}_\mathcal{D}(h_{1-s}, f) > (3+\varepsilon)\mathcal{L}_\mathcal{D}(h_s, f)$ for some $s \in \{0,1\}$. By Claim 4.6 we have $R_s < \frac{1}{2} - \frac{\varepsilon}{2(2+\varepsilon)}$, and since $R_0 + R_1 = 1$, this implies that $R_{1-s} > \frac{1}{2} + \frac{\varepsilon}{2(2+\varepsilon)}$. If $|\widehat{R}_b - R_b| \le 2\delta$ for each $b \in \{0,1\}$ then by our choice of δ we have $\widehat{R}_s < \frac{1}{2} < \widehat{R}_{1-s}$. Since $\rho = \arg\min_{b\in\{0,1\}}\widehat{R}_b$, we see that the verifier outputs $\rho = s$ with probability at least $1-\beta$. The runtime, communication complexity, and query complexity guarantees follow from Lemmas 3.1 and 3.2, and by inspection of Protocol 6. □

4.3 Offloading queries to the provers

The protocols in the proofs of Theorems 4.2 and 4.4 do not have the provers make any queries to f. However, if we give the provers query access to f, then we can use the "refereed query

delegation" technique of Lemma 3.8 to offload all verifier queries to the provers, while keeping the complexity of the protocol essentially unchanged. The resulting protocols incur an additional factor of $d + \lambda + \log|\mathcal{Y}|$ in communication complexity; however, the verifier in this modified protocol only makes at most 1 query to either f, h_0, h_1, or $Q_{\mathcal{D}}$.

5 Lower bounds for refereed learning

In this section we prove several lower bounds for refereed learning protocols with "white-box" access to h_0 and h_1—that is, the protocols receive a representation of h_0 and h_1 as input. Since the white-box versions of $\mathcal{P}_0, \mathcal{P}_1$, and $\mathcal{V}$ can simulate their black-box counterparts, a lower bound against white-box refereed learning implies the same lower bound against the black-box version.[5] In what follows we show that for simple classes of functions and distributions, even white-box refereed learning protocols require: (1) query access to f (Section 5.1), (2) query access to $Q_{\mathcal{D}}$ (Section 5.2), and (3) exponential-time provers (Section 5.3).

Lower bounds for weaker verifier access models. In Theorems 5.1 and 5.3, we consider refereed learning protocols with additive and multiplicative error ($\alpha \geq 1$ and $\eta \in (0,1)$), and show that if instead of query access to f and $Q_{\mathcal{D}}$ the verifier either (1) has query access to $Q_{\mathcal{D}}$ but only has access to f via random labeled samples $(x, f(x))$, or (2) has query access to f, but only has access to $Q_{\mathcal{D}}$ via samples $x \sim \mathcal{D}$, then it requires sample complexity at least $\frac{1}{\eta}$. This immediately implies that when $\eta \to 0$ (the setting of Theorems 4.2 and 4.4), every refereed learning protocol requires verifier query access to f and $Q_{\mathcal{D}}$.

Time complexity lower bound. In Theorem 5.4 we focus on the setting of $\eta = 0$, and show how a refereed learning protocol can be used to decide 3-SAT with a constant factor overhead in running time. Subject to standard computational hardness assumptions, Theorem 5.4 justifies the exponential running time of the provers in Theorems 4.2 and 4.4.

5.1 Verification with labeled samples

In this section we prove a lower bound on the number of labeled samples needed for verification in the two-prover setting when the verifier only has access to f via labeled samples (instead of queries).

Theorem 5.1 (Refereed learning with labeled samples)**.** *Fix a representation of functions. Let $c > 0$ be a sufficiently small absolute constant, and fix range $\mathcal{Y} = \{0,1\}$. For all $b \in \{0,1\}$ let $\mathcal{O}_b(f, h_0, h_1, \mathcal{D})$ provide query access to f, h_0, h_1 and $Q_{\mathcal{D}}$; and let $\mathcal{O}_{\mathcal{V}}(f, h_0, h_1, \mathcal{D})$ provide labeled samples $(x, f(x))$ where $x \sim \mathcal{D}$, and query access to h_0, h_1, and $Q_{\mathcal{D}}$. For all $d \in \mathbb{N}$, $\alpha \geq 1$ and $\eta \in (0,1)$, there exists a class of boolean functions $\mathcal{H}$ and distributions $\mathbb{D}$ such that every $(\alpha, \eta, 1/3)$-refereed learning protocol for $\mathcal{H}$ and $\mathbb{D}$ with respect to ℓ_{zo} and oracles $\mathcal{O}_0, \mathcal{O}_1$ and $\mathcal{O}_{\mathcal{V}}$ requires verifier sample-complexity $\frac{c}{\eta}$. Moreover, the lower bound holds even if the representation of h_0, h_1, and $Q_{\mathcal{D}}$ is given as input to all parties.*

[5] There is a caveat that lower bounds on runtime depend on the representation and may incur a factor that depends on the time complexity of evaluating h_0 and h_1. We deal with this issue explicitly in Section 5.3. On the other hand, white-box query and sample complexity lower bounds apply directly to the black-box setting.

The intuition behind this proof is straightforward. Fix hypotheses $\{h_0, h_1\}$, sample $b \sim \{0,1\}$ and let $f \leftarrow h_b$. Now, consider a malicious prover that executes the honest protocol, except it "pretends" that $f = h_{1-b}$. As long as the verifier does not obtain any sample x with $f(x) \neq h_{1-b}(x)$, it cannot refute the malicious prover's claim that h_{1-b} has zero loss, and thus cannot determine if it should accept h_0 or h_1.

Proof. We prove the lower bound when all parties receive as input a representation of h_0, h_1 and $Q_{\mathcal{D}}$. Set $\mathcal{H} = \{h_0, h_1\}$, where $h_0(x) = 0$ for all $x \in \{0,1\}^d$, and $h_1(x) = 0$ for all $x \neq 0^d$ and $h_1(0^d) = 1$. Set $\mathbb{D} = \{\mathcal{D}\}$, where $\mathcal{D}$ is the distribution that places probability mass η on the point 0^d and is uniform otherwise. Suppose $[\mathcal{P}_0, \mathcal{P}_1, \mathcal{V}]$ is an $(\alpha, \eta, \frac{1}{3})$-refereed learning protocol for $\mathcal{H}$ and $\mathbb{D}$ with respect ℓ_{zo} and oracles $\mathcal{O}_0, \mathcal{O}_1$, and $\mathcal{O}_{\mathcal{V}}$. Let $\mathcal{D}_f$ denote the distribution over $(x, f(x))$ where $x \sim \mathcal{D}$. By Definition 2.3 and the definition of $\mathcal{O}$ and $\mathcal{O}_{\mathcal{V}}$, for all $b \in \{0,1\}$, $f = h_b$, and $\mathcal{P}^*_{1-b}$ we have $\left[\mathcal{P}^f_b(h_0, h_1, Q_{\mathcal{D}}), \mathcal{P}^*_{1-b}, \mathcal{V}^{\mathcal{D}_f}(h_0, h_1, Q_{\mathcal{D}})\right] = b$ with probability at least $\frac{2}{3}$. Since h_0, h_1, and $Q_{\mathcal{D}}$ are fixed, we will not write them explicitly—that is, we will let $\mathcal{P}^f_b$ denote $\mathcal{P}^f_b(h_0, h_1, Q_{\mathcal{D}})$ and let $\mathcal{V}^{\mathcal{D}_f}$ denote $\mathcal{V}^{\mathcal{D}_f}(h_0, h_1, Q_{\mathcal{D}})$.

For each $b \in \{0,1\}$ let the malicious prover $\mathcal{P}^*_{1-b}$ execute the honest prover protocol $\mathcal{P}^{h_{1-b}}$—that is, the honest prover protocol run as if the true function f is h_{1-b}. At a high level, we will argue that if b is sampled uniformly at random, then the verifier cannot distinguish between $b = 0$ and $b = 1$ until it samples the point $(0^d, f(0^d))$ from $\mathcal{D}_f$, and thus cannot correctly output b.

Now, for each $b \in \{0,1\}$ and $f \leftarrow h_b$, define the *view of the verifier* $\mathsf{view}\big(\mathcal{V}^{\mathcal{D}_f}\big)_b$ as the distribution over the m samples $(x, f(x))$ drawn from $\mathcal{D}_f$, and the transcripts T_b and T_{1-b} between $\mathcal{V}$ and $\mathcal{P}^f_b$, and between $\mathcal{V}$ and $\mathcal{P}^*_{1-b}$. Let E be the event that one of the m samples drawn by the verifier is $(0^d, f(0^d))$. Since $\mathcal{P}^*_{1-b}$ executes $\mathcal{P}^{h_{1-b}}_{1-b}$ and the honest prover executes $\mathcal{P}^f_b = \mathcal{P}^{h_b}_b$, the distribution of (T_0, T_1) is independent of b (the verifier always interacts with $\mathcal{P}^{h_0}_0$ and $\mathcal{P}^{h_1}_1$), and thus $\mathsf{view}\big(\mathcal{V}^{\mathcal{D}_f} \mid \overline{E}\big)_0 = \mathsf{view}\big(\mathcal{V}^{\mathcal{D}_f} \mid \overline{E}\big)_1$. Next, we utilize the following fact from [RS06].

Fact 5.2 (Claim 4 [RS06])**.** *Let E be an event that happens with probability at least $1 - \delta$ under the distribution $\mathcal{D}$. Then $d_{\mathrm{TV}}(\mathcal{D}|_E, \mathcal{D}) \leq \delta'$, where $\delta' = \frac{\delta}{1-\delta}$.*

Applying the triangle inequality twice yields

$$\begin{aligned} d_{\mathrm{TV}}\big(\mathsf{view}\big(\mathcal{V}^{\mathcal{D}_f}\big)_0, \mathsf{view}\big(\mathcal{V}^{\mathcal{D}_f}\big)_1\big) &\leq d_{\mathrm{TV}}\big(\mathsf{view}\big(\mathcal{V}^{\mathcal{D}_f}\big)_0, \mathsf{view}\big(\mathcal{V}^{\mathcal{D}_f} \mid \overline{E}\big)_0\big) \\ &+ d_{\mathrm{TV}}\big(\mathsf{view}\big(\mathcal{V}^{\mathcal{D}_f} \mid \overline{E}\big)_0, \mathsf{view}\big(\mathcal{V}^{\mathcal{D}_f} \mid \overline{E}\big)_1\big) \\ &+ d_{\mathrm{TV}}\big(\mathsf{view}\big(\mathcal{V}^{\mathcal{D}_f} \mid \overline{E}\big)_1, \mathsf{view}\big(\mathcal{V}^{\mathcal{D}_f}\big)_1\big). \end{aligned}$$

Since $\Pr_{X \sim \mathcal{D}_d}\left[X = 0^d\right] = \eta$, the event E occurs with probability at most $m \cdot \eta$. Combined with Fact 5.2 and the fact that $\mathsf{view}\big(\mathcal{V}^{\mathcal{D}_f} \mid \overline{E}\big)_0 = \mathsf{view}\big(\mathcal{V}^{\mathcal{D}_f} \mid \overline{E}\big)_1$, we see that $d_{\mathrm{TV}}\big(\mathsf{view}\big(\mathcal{V}^{\mathcal{D}_f}\big)_0, \mathsf{view}\big(\mathcal{V}^{\mathcal{D}_f}\big)_1\big) < \frac{1}{3}$ whenever $m \leq \frac{c}{\eta}$ for a sufficiently small absolute constant $c > 0$. The rest of the proof follows

from standard arguments. Observe that

$$\begin{aligned}
\Pr_{\substack{b\sim\{0,1\}\\ f\leftarrow h_b}}\left[\left[\mathcal{P}_b^f, \mathcal{P}_{1-b}^*, \mathcal{V}^{\mathcal{D}_f}\right] = b\right] &= \frac{1}{2}\left(\Pr_{\substack{f\leftarrow h_0\\ s\sim\mathsf{view}\left(\mathcal{V}^{\mathcal{D}_f}\right)_0}}[\mathcal{V}(s)=0] + \Pr_{\substack{f\leftarrow h_1\\ s\sim\mathsf{view}\left(\mathcal{V}^{\mathcal{D}_f}\right)_1}}[\mathcal{V}(s)=1]\right)\\
&= \frac{1}{2} + \frac{1}{2}\left(\Pr_{\substack{f\leftarrow h_0\\ s\sim\mathsf{view}\left(\mathcal{V}^{\mathcal{D}_f}\right)_0}}[\mathcal{V}(s)=0] - \Pr_{\substack{f\leftarrow h_1\\ s\sim\mathsf{view}\left(\mathcal{V}^{\mathcal{D}_f}\right)_1}}[\mathcal{V}(s)=0]\right)\\
&\leq \frac{1}{2} + \frac{1}{2}\cdot d_{\mathrm{TV}}\left(\mathsf{view}\left(\mathcal{V}^{\mathcal{D}_f}\right)_0, \mathsf{view}\left(\mathcal{V}^{\mathcal{D}_f}\right)_1\right) < \frac{2}{3}.
\end{aligned}$$

Since this contradicts the definition of an $(\alpha, \eta, \frac{1}{3})$-refereed learning protocol, we must have verifier sample complexity $m \geq \frac{c}{\eta}$. □

5.2 Verification without query access to the PMF

In this section, we prove that non-trivial refereed learning requires query access to the probability mass function $Q_{\mathcal{D}}$ of the underlying distribution $\mathcal{D}$. Specifically, we show that a verifier with query access to f, but only sample access to $\mathcal{D}$, will require many samples from $\mathcal{D}$. (Note, however, that our refereed query delegation protocol in Section 3.2 means that the verifier only needs a single query to $Q_{\mathcal{D}}$ to be efficient.) This is in contrast to the lower bound of Theorem 5.1, where the distribution is known to be uniform, but the verifier is only given labeled samples $(x, f(x))$ and cannot query f.

Theorem 5.3 (Refereed learning without query access to $Q_{\mathcal{D}}$)**.** *Fix a representation of functions. Let $c > 0$ be a sufficiently small constant, and fix range $\mathcal{Y} = \{0,1\}$. For all $b \in \{0,1\}$ let $\mathcal{O}(f, h_0, h_1, \mathcal{D})$ provide query access to h_0, h_1, f, and $Q_{\mathcal{D}}$; and let $\mathcal{O}_{\mathcal{V}}(f, h_0, h_1, \mathcal{D})$ provide query access to h_0, h_1, and f, and samples $x \sim \mathcal{D}$. For all $d \in \mathbb{N}$, $\alpha \geq 1$, and $\eta \in (0,1)$, there exists a class of boolean functions $\mathcal{H}$ and distributions $\mathbb{D}$ such that every $(\alpha, \eta, 1/3)$-refereed learning protocol for $\mathcal{H}$ and $\mathbb{D}$ with respect to ℓ_{zo} and oracles $\mathcal{O}_0, \mathcal{O}_1$, and $\mathcal{O}_{\mathcal{V}}$ requires verifier sample complexity $\frac{c}{\eta}$. Moreover, the lower bound holds even if the representation of h_0, h_1, and f is given as input to all parties.*

Proof. We prove the lower bound when all parties receive as input the representation of h_0, h_1, and f. The proof is similar to the proof of Theorem 5.1, except instead of choosing the function f to be either h_0 or h_1, we randomly select a distribution $\mathcal{D}_0$ or $\mathcal{D}_1$ such that $\mathcal{L}_{\mathcal{D}_b}(h_b, f) = 0 < \alpha \cdot \mathcal{L}_{\mathcal{D}_b}(h_{1-b}, f)$ for all $b \in \{0,1\}$. Consider the family of functions $\mathcal{H} = \{h_0, h_1, f\}$ where $f(x) = 0$ for all $x \in \{0,1\}^d$, $h_0(x) = 0$ for all $x \neq 0^d$ and $h_0(0^d) = 1$, and $h_1(x) = 0$ for all $x \neq 1^d$ and $h_1(1^d) = 1$. Next we define the family of distributions. For each $b \in \{0,1\}$, let $\mathcal{D}_b$ place probability mass η on the point $(1-b)^d$ and be uniform over $\{0,1\}^d \setminus \{b^d, (1-b)^d\}$. Let $\mathbb{D} = \{\mathcal{D}_0, \mathcal{D}_1\}$. For simplicity, let Q_b denote the PMF $Q_{\mathcal{D}_b}$ for all $b \in \{0,1\}$. Notice that $\mathcal{L}_{\mathcal{D}_b}(h_b, f) = 0$ since f and h_b agree everywhere except $x = b^d$, whereas $\mathcal{L}_{\mathcal{D}_b}(h_{1-b}, f) = \eta$ since h_{1-b} and f disagree on $x = (1-b)^d$ which is in the support of $\mathcal{D}_b$. Thus, if $[\mathcal{P}_0, \mathcal{P}_1, \mathcal{V}]$ is an $(\alpha, \eta, \frac{1}{3})$-refereed learning protocol for $\mathcal{H}$ and $\mathbb{D}$ with respect to ℓ_{zo} and oracles $\mathcal{O}_0$, $\mathcal{O}_1$, and $\mathcal{O}_{\mathcal{V}}$, then for all $b \in \{0,1\}$ and all $\mathcal{P}_{1-b}^*$ we have $\left[\mathcal{P}_b^{Q_b}(f, h_0, h_1), \mathcal{P}_{1-b}^*, \mathcal{V}^{\mathcal{D}_b}(f, h_0, h_1)\right] = b$ with probability at least $\frac{2}{3}$. Since h_0, h_1, and f are fixed we will omit them in the rest of the proof—that is, we let $\mathcal{P}_b^{Q_b}$ denote $\mathcal{P}^{Q_b}(f, h_0, h_1)$ and let $\mathcal{V}^{\mathcal{D}_b}$ denote $\mathcal{V}^{\mathcal{D}_b}(f, h_0, h_1)$.

Now, for all $b \in \{0,1\}$ let $\mathcal{P}^*_{1-b}$ execute the honest prover protocol $\mathcal{P}^{Q_{1-b}}_{1-b}$—that is, execute the protocol as if the distribution were $\mathcal{D}_{1-b}$. Define the *view of the verifier* $\mathsf{view}\big(\mathcal{V}^{\mathcal{D}_b}\big)_b$ as the distribution over query answers from f, samples $x_1, \ldots, x_m \sim \mathcal{D}_b$, and transcripts T_b and T_{1-b} from the interaction with $\mathcal{P}^{Q_b}_b$ and $\mathcal{P}^*_{1-b}$. Note that since $\mathcal{P}^*_{1-b}$ executes the honest prover protocol $\mathcal{P}^{Q_{1-b}}_{1-b}$, the verifier always interacts with $\mathcal{P}^{Q_0}_0$ and $\mathcal{P}^{Q_1}_1$, and thus the transcripts T_0 and T_1 are independent of b. Similarly, since the function f is fixed in advance, the query answers are independent of b as well.

To complete the proof, we argue that $\mathcal{V}$ cannot distinguish whether $b = 0$ or $b = 1$ until it draws many samples from $\mathcal{D}_b$. Let E be the event that one of the m samples drawn by $\mathcal{V}^{\mathcal{D}_b}$ is either 0^d or 1^d. Since the transcripts and query answers are independent of b, we have that $\mathsf{view}\big(\mathcal{V}^{\mathcal{D}_b} \mid \overline{E}\big)_0 = \mathsf{view}\big(\mathcal{V}^{\mathcal{D}_b} \mid \overline{E}\big)_1$. Since $\mathcal{D}_b$ places probability mass η on $(1-b)^d$ and probability mass 0 on b^d, we have that $\Pr[E] \leq m \cdot \eta$. Applying Fact 5.2, and the same argument as in the proof of Theorem 5.1, we obtain $d_{\mathrm{TV}}\big(\mathsf{view}\big(\mathcal{V}^{\mathcal{D}_0}\big)_0, \mathsf{view}\big(\mathcal{V}^{\mathcal{D}_1}\big)_1\big) < \frac{1}{3}$, whenever $m \leq \frac{c}{\eta}$ and thus $\Pr_{b\sim\{0,1\}}\left[\left[\mathcal{P}^{Q_b}_b, \mathcal{P}^*_{1-b}, \mathcal{V}^{\mathcal{D}_b}\right] = b\right] < \frac{2}{3}$. Since this contradicts the definition of an $(\alpha, \eta, \frac{1}{3})$-refereed learning protocol, $\mathcal{V}$ must draw $m \geq \frac{c}{\eta}$ samples from $\mathcal{D}_b$. □

5.3 Prover time-complexity lower bound

In this section we show that any white-box refereed learning protocol can be used as a subroutine to decide if a 3-CNF formula is satisfiable. Formally, let SAT denote the set of satisfiable 3-CNF formulas with d variables and m clauses for all $d, m \in \mathbb{N}$, and suppose every algorithm that decides SAT with probability at least $\frac{2}{3}$ has runtime at least $T_{\mathrm{SAT}}(d, m)$. In Theorem 5.4 we show that every white-box refereed learning protocol must have either prover or verifier runtime $\Omega(T_{\mathrm{SAT}}(d, m-1)/m)$, which justifies the running time of our protocols under standard complexity assumptions.

Throughout the section we let $\mathcal{H}_{d,m} = \big\{\phi : \{0,1\}^d \to \{0,1\}\big\}$ where ϕ is a 3-CNF formula with d variables and m clauses.[6] Let U_d be the uniform distribution on $\{0,1\}^d$. Additionally, define oracles $\mathcal{O}_0 = \mathcal{O}_1 = \mathcal{O}_{\mathcal{V}} = \mathcal{O}$ by letting $\mathcal{O}(f, \phi_0, \phi_1, \mathcal{D})$ provide ϕ_0 and ϕ_1, and query access to f.

Theorem 5.4 (Prover time-complexity lower bound)**.** *Fix $\alpha \in \mathbb{N}$. Suppose there exists a protocol $[\mathcal{P}_0, \mathcal{P}_1, \mathcal{V}]$ that for all inputs $d, m \in \mathbb{N}$ is an $(\alpha, 0, \frac{1}{3})$-refereed learning protocol for $\mathcal{H}_{d,m}$ and $\{U_d\}$ with respect to ℓ_{zo} and oracles $\mathcal{O}_0$, $\mathcal{O}_1$, and $\mathcal{O}_{\mathcal{V}}$ (defined above). Then $[\mathcal{P}_0, \mathcal{P}_1, \mathcal{V}]$ has either prover or verifier runtime $\Omega(T_{SAT}(d, m-1)/m)$.*

Proof. The main step in the proof is Claim 5.5, which states that an $(\alpha, 0, \frac{1}{3})$-refereed learning protocol for $\mathcal{H}$ and $\{U\}$ can be used to decide 3-SAT.

Claim 5.5 (Reduction from 3-SAT)**.** *Let $[\mathcal{P}_0, \mathcal{P}_1, \mathcal{V}]$ be as in Theorem 5.4. If the prover and verifier runtime is at most $T(d, m)$, then there exists an algorithm $\mathcal{A}$ that decides 3-SAT with probability at least 2/3 in time $O(m \cdot T(d, m+1))$.*

Proof. Below, we construct an algorithm $\mathcal{A}$, which uses a refereed learning protocol as a subroutine to decide 3-SAT. Let $a > 0$ be a sufficiently large constant.

[6] Technically $\mathcal{H}_{d,m}$ is a multiset, i.e., we include distinct representations of the same function as distinct elements.

Algorithm $\mathcal{A}$

Input: 3-CNF formula ϕ
Output: accept/reject

1. Let $\phi_0(x) = \phi(x) \wedge (x_1)$ and $\phi_1(x) = \phi(x) \wedge (x_1 \oplus 1)$.
2. Repeat the following for each $j \in [a]$:
   (a) Sample $b_j \sim \{0,1\}$ uniformly and simulate $[\mathcal{P}_0, \mathcal{P}_1, \mathcal{V}]$ by providing $\mathcal{P}_0, \mathcal{P}_1$, and $\mathcal{V}$ with query access to $f = \phi_{b_j}$ and input ϕ_0 and ϕ_1.
   (b) Let ρ_j be the bit output by the verifier in the j^{th} simulation.
3. If $|\{j : \rho_j = b_j\}| \geq \frac{7 \cdot a}{12}$ then output **accept**; otherwise output **reject**.

Algorithm 1: reduction from SAT

First, we argue that $\mathcal{A}$ accepts with probability at least $\frac{2}{3}$ when ϕ is satisfiable. If ϕ is satisfiable, then at least one of ϕ_0 or ϕ_1 is satisfiable. Suppose ϕ_0 is satisfiable. Then $\phi_0(x) \neq \phi_1(x)$ whenever $\phi_0(x) = 1$. Moreover, since $f \in \{\phi_0, \phi_1\}$ either ϕ_0 or ϕ_1 has zero loss (but not both), and hence the verifier $\mathcal{V}$ must output $\rho_j = b_j$ with probability at least $\frac{2}{3}$. It follows by a Chernoff bound that for a sufficiently large absolute constant a, the verifier outputs $\rho_j = b_j$ on least $\frac{7 \cdot a}{12}$ of the simulations with probability at least $\frac{2}{3}$, and therefore $\mathcal{A}$ will output accept with probability at least $\frac{2}{3}$.

Next, suppose ϕ is not satisfiable—that is, $\phi(x) = 0$ for all $x \in \{0,1\}^d$. Then for all $b \in \{0,1\}$ formula ϕ_b is also unsatisfiable, and hence $\phi_b(x) = 0$ for all $x \in \{0,1\}^d$, and therefore $f(x) = 0$ for all $x \in \{0,1\}^d$. Since b_j is sampled uniformly at random, the probability that $\mathcal{V}$ outputs $\rho_j = b_j$ is exactly $\frac{1}{2}$. It follows that for sufficiently large absolute constant a, the verifier outputs $\rho_j = b_j$ on least $\frac{7 \cdot a}{12}$ of the simulations with probability at most $\frac{1}{3}$, and therefore $\mathcal{A}$ will output reject with probability at least $\frac{2}{3}$.

Thus, if the protocol has verifier and prover time complexity $T(d, m)$, then, since answering each query made by the protocol requires evaluating $f = \phi_b$, a 3-CNF formula with m clauses, algorithm $\mathcal{A}$ runs in time at most $O(m \cdot T(d, m+1))$ and decides 3-SAT with probability at least $\frac{2}{3}$. □

The proof of Theorem 5.4 follows since by definition of T_{SAT} and Claim 5.5 we must have $T(d, m) = \Omega(T_{\text{SAT}}(d, m-1)/m)$. □

6 Applications and extensions of our protocols

In this section we turn to applications and extensions of our refereed learning protocols. In Section 6.1 we show a natural setting (namely, where the hypothesis functions h_0 and h_1 are juntas) in which both the prover and the verifier can be implemented efficiently. In this regime, the verifier's runtime is an arbitrary poly(d) factor smaller than the time required to solve this problem without the provers, showing that a refereed learning protocol can save the verifier significant computational resources even when the provers are computationally bounded. In Section 6.2 we show how to extend our protocols, which deal with λ-precise loss functions and distributions, to handle loss functions and distributions specified to arbitrary precision.

6.1 Efficient refereed learning for juntas

We now show how our protocol from Theorem 4.2 can be implemented efficiently for a natural class of hypotheses; moreover, we show that the verifier in this protocol takes time which is smaller by an arbitrary polynomial factor than the time needed for this family of hypotheses without access to the provers.

For each $d, j \in \mathbb{N}$ let $\mathcal{H}_{d,j} = \big\{h : \{0,1\}^d \to \{0,1\} \mid h \text{ is a } j\text{-junta}\big\}$, and let U_d denote the uniform distribution over $\{0,1\}^d$. Recall that, for $d, j \in \mathbb{N}$, a function $h : \{0,1\}^d \to \{0,1\}$ is a j-junta if there exists a set $J \subseteq [d]$ with $|J| \leq j$ and a function $g_h : \{0,1\}^J \to \{0,1\}$ such that $h(x) = g_h(x_J)$ for all $x \in \{0,1\}^d$, where $x_J = x_{J_1} x_{J_2} \cdots x_{J_j}$—that is, the value of $h(x)$ is uniquely determined by the setting of x at the indices in J.

Now assume that, in addition to the usual query access to h_0 and h_1, the provers and verifier obtain the junta indices J_0 and J_1 as input. We show:

Proposition 6.1. *Let $\mathcal{O}_0 = \mathcal{O}_1$ provide query access to h_0 and h_1, and let $\mathcal{O}_{\mathcal{V}}(f, h_0, h_1, \mathcal{D})$ provide query access to f, h_0, and h_1. There exists a protocol $[\mathcal{P}_0, \mathcal{P}_1, \mathcal{V}]$ that, for all inputs $d, j \in \mathbb{N}$ and $\varepsilon, \beta > 0$, is a $(1+\varepsilon, 0, \beta)$-refereed learning protocol for $\mathcal{H}_{d,j}$ and $\{U_d\}$ with respect to ℓ_{zo} and $\mathcal{O}_0, \mathcal{O}_1, \mathcal{O}_{\mathcal{V}}$. Moreover, for all $h_0, h_1 \in \mathcal{H}_{d,j}$, if the junta bits J_0 and J_1 are given as input to all parties then the protocol has the following guarantees:*

- *The verifier runs in time $(1+\frac{1}{\varepsilon^2}) \log \frac{1}{\beta} \operatorname{poly} d$ and makes $O\big((1+\frac{1}{\varepsilon^2}) \log \frac{1}{\beta}\big)$ queries to f.*
- *The provers run in time $(1+\frac{1}{\varepsilon^2}) \log \frac{1}{\beta} \cdot 2^{2j} \operatorname{poly} d$.*
- *The communication complexity of the protocol is $(1+\frac{1}{\varepsilon^2}) \log \frac{1}{\beta} \operatorname{poly} d$.*

Proof. At a high level, we show that the provers can efficiently compute the set $S = \{x \mid h_0(x) \neq h_1(x)\}$. Moreover, since the distribution is uniform, the verifier can use certifiable sum (Lemma 3.2) and certifiable index (Claim 3.5) to efficiently sample a uniform element of S. To argue that the provers can also execute these protocols efficiently we leverage the junta structure of h_0 and h_1.

Let $J_0, J_1 \subseteq [d]$ be the set of junta bits for h_0 and h_1, respectively. In what follows, let $c > 0$ be a sufficiently large, absolute constant.

$$\Big[\mathcal{P}_0^{h_0,h_1}, \mathcal{P}_1^{h_0,h_1}, \mathcal{V}^{f,h_0,h_1}\Big](d, j, \varepsilon, \beta, J_0, J_1)$$

1. $\mathcal{P}_0, \mathcal{P}_1$: Let $J \leftarrow J_0 \cup J_1$. Query h_0 and h_1 on all settings of the bits in J. Let $S \leftarrow \big\{x \in \{0,1\}^d \mid h_0(x) \neq h_1(x)\big\}$, with a lexicographic ordering.[a]
2. $\mathcal{V}$: Execute certifiable sum (Lemma 3.2) with $t(x) = \mathbb{1}[h_0(x) \neq h_1(x)]$ to obtain $|S|$.
3. $\mathcal{V}$: Set $m \leftarrow c\big(1+\frac{1}{\varepsilon^2}\big) \log \frac{1}{\beta}$. Sample $x_1, \ldots, x_m \sim [|S|]$ and execute certifiable index (Claim 3.5) to obtain $x_1, \ldots, x_m \leftarrow S_{i_1}, \ldots, S_{i_m}$.
4. $\mathcal{V}$: Query f on $x_1, \ldots, x_m$ and output $\rho = \arg\min_{s \in \{0,1\}} |\{i \in [m] : h_s(x_i) \neq f(x_i)\}|$.

[a] The provers need not explicitly compute S.

Protocol 7: refereed learning for juntas

We first argue that the provers in Protocol 7 can be implemented efficiently. Since h_0 and h_1 are j-juntas with junta bits J_0 and J_1, the provers can compute $|S|$ and the i^{th} element of S (ordered

lexicographically) in time $O(2^{2j})$ by querying h_0 and h_1 on all 2^{2j} settings of the bits in J_0 and J_1. It follows that executing certifiable sum in the second step and certifiable index in the third step takes time at most $2^{2j} \operatorname{poly} d$. Thus, the provers run in time $(1 + \frac{1}{\varepsilon^2}) \cdot 2^{2j} \operatorname{poly} d$.

The soundness of the verifier follows by the same argument as in the proof of Theorem 4.2. The runtime and query complexity of the verifier, and the communication complexity of the protocol follows by Lemma 3.2 and Claim 3.5. □

Comparison to the case of a proverless learner. Below, we argue that any learner that does not use the help of the provers must run in time $\Omega(2^j)$. Hence, for $j = c \log d$, the runtime of a proverless learner is an arbitrary $\operatorname{poly} d$ factor worse than the runtime of the learner with provers. Moreover, for this setting of j the provers' runtime is still $\operatorname{poly} d$. To see this lower bound on the runtime of the proverless learner, consider the following simple argument: Let $J = [j]$ be the junta bits and let h_0 be some fixed j-junta—that is, $h_0(x) = g_{h_0}(x_J)$ for some function $g_{h_0} : \{0,1\}^J \to \{0,1\}$. Now, choose a random j-junta h_1 as follows: sample $z \sim \{0,1\}^J$, let $g_{h_1}(z) = 1 - g_{h_0}(z)$ and $g_{h_0} = g_{h_1}$ otherwise, and let $h_1(x) = g_{h_1}(x_J)$. Notice that h_0 and h_1 are each j-juntas[7] with junta bits J, and agree everywhere except on points x with $x_J = z$. Now, choose $b \sim \{0,1\}$ uniformly, set $f \leftarrow h_b$, and consider an algorithm $\mathcal{A}$ that, given input J and query access to h_0, h_1, and f, outputs b with probability at least $\frac{2}{3}$. Notice that before $\mathcal{A}$ queries one of the functions on a point x such that $h_0(x) \neq h_1(x)$, the query answers are independent of b. Since $h_0(x) \neq h_1(x)$ if and only if $x_J = z$, and since z is chosen uniformly at random, $\mathcal{A}$ will require $\Omega(2^j)$ queries to output b with probability at least $\frac{2}{3}$.

Only the provers need to know J_0 and J_1. Finally, we note that Proposition 6.1 can be readily extended to the setting where only provers are provided J_0 and J_1 as input, via the following simple protocol: for each $b \in \{0,1\}$ and $i \in J_b$, the provers send i as well as points x and $x' = x^{\oplus i}$ (x with the i^{th} bit flipped) such that $h_b(x) \neq h_b(x')$. The verifier can then query h_b and, if $h_b(x) \neq h_b(x')$, add i to the set $\widehat{J}_b$. Since h_b is a junta with indices J_b, only indices $i \in J_b$ will be added to $\widehat{J}_b$ by the verifier. Moreover, since at least one prover is honest and receives J_0 and J_1 as input, every $i \in J_b$ will be added to $\widehat{J}_b$ by the verifier. Thus, the verifier will obtain $\widehat{J}_b = J_b$ for each $b \in \{0,1\}$. Since finding such x and x' for each $i \in J_b$ can be done by querying h_b once for each of the 2^j settings of the bits in J_b, the provers run in time $O(2^j)$. Similarly, since checking each candidate i and pair x and x' received from the provers takes 2 queries the verifier runs in time $O(j)$.

6.2 When $\mathcal{D}$ and ℓ are arbitrarily precise

We now explain how our protocols can be implemented for distributions $\mathcal{D}$ and metrics ℓ that are not λ-precise (Definition 2.4); however, in order to bound communication complexity, we still require ℓ to be *M-bounded*—that is, $\ell(y, y') \leq M$ for all $y, y' \in \mathcal{Y}$. Specifically, we analyze the cost of rounding $Q_{\mathcal{D}}$ and ℓ so that they are λ-precise. Combined with the protocols in Section 4, this allows us to design protocols with additive error η for arbitrarily precise distributions and loss functions that only incur communication cost $\log \frac{1}{\eta} \operatorname{poly} d$ (there is no cost in query complexity).

[7] There is an annoying but fixable issue that can arise here where h_1 may be a $(j-1)$-junta. To remedy this, we can simply choose a g_{h_0} that cannot be made into a $(j-1)$-junta by flipping $g_{h_0}(x)$ on a single $x \in \{0,1\}^j$. For example, setting g_{h_0} at random for each $x \in \{0,1\}^J$ ensures this is satisfied with high probability.

Recall that by Claim 3.7, for all distributions $\mathcal{D}$ over $\{0,1\}^d$, the distribution $\mathcal{D}_\lambda$ (defined by $\mathcal{D}_\lambda(x) = \frac{\lfloor \mathcal{D}(x) \rfloor_\lambda}{\sum_{x \in \{0,1\}^d} \lfloor \mathcal{D}(x) \rfloor_\lambda}$, where $\lfloor y \rfloor_\lambda$ denotes $2^{-\lambda} \cdot \lfloor 2^\lambda \cdot y \rfloor$ for all $y \in \mathbb{R}$—that is, $\lfloor y \rfloor_\lambda$ denotes the nearest multiple of $2^{-\lambda}$ that is at most y) is λ-precise and satisfies $d_{\mathrm{TV}}(\mathcal{D}, \mathcal{D}_\lambda) \le 2^{d+1-\lambda}$. Now, for metric ℓ, let $\ell_\lambda(y, y')$ denote $\lfloor \ell(y, y') \rfloor_\lambda$. Then, by definition of $\lfloor \cdot \rfloor_\lambda$ we have $|\ell_\lambda(y, y') - \ell(y, y')| \le 2^{-\lambda}$. At a high level, the goal of this section is to explain how one can execute the protocols of Theorems 4.2 and 4.4 using $\mathcal{D}_\lambda$ and ℓ_λ, given an M-bounded metric ℓ and query access to $Q_\mathcal{D}$, at the cost of a small additive error term η and factor of $\log \frac{1}{\eta}$ in communication complexity and runtime.

In Proposition 6.2 we assume that for any metric ℓ, computing $\lfloor \ell(y, y') \rfloor_\lambda$ takes unit time.

Proposition 6.2 (Protocol for arbitrary $\mathcal{D}$ and M-bounded ℓ)**.** *For each $b \in \{0,1\}$ let $\mathcal{O}_b(f, h_0, h_1, \mathcal{D})$ provide query access to h_0, h_1, and $Q_\mathcal{D}$, and let $\mathcal{O}_\mathcal{V}$ provide query access to h_0, h_1, f and $Q_\mathcal{D}$. Fix integer $M \in \mathbb{N}$, range $\mathcal{Y}$, and M-bounded metric ℓ on $\mathcal{Y}$.*

For all $d \in \mathbb{N}$, $\alpha \ge 1$, $\beta > 0$, and $\eta \in (0,1)$, let $\lambda(d, \alpha, \beta, \eta) = d + \log \frac{\alpha M}{\eta}$ and let $[\mathcal{P}'_0, \mathcal{P}'_1, \mathcal{V}']_{d,\alpha,\beta,\lambda}$ be an $(\alpha, 0, \beta)$-refereed learning protocol for $\mathfrak{F}$ and $\mathbb{D}_\lambda$ with respect to ℓ_λ and $\mathcal{O}_0, \mathcal{O}_1$, and $\mathcal{O}_\mathcal{V}$. Suppose $[\mathcal{P}'_0, \mathcal{P}'_1, \mathcal{V}']_{d,\alpha,\beta,\lambda}$ has communication complexity and verifier runtime $T(d, \alpha, \beta, \lambda)$ and makes $q(d, \alpha, \beta, \lambda)$ queries to f. There exists a protocol $[\mathcal{P}_0, \mathcal{P}_1, \mathcal{V}]$ that, for all $d \in \mathbb{N}$, $\alpha \ge 1, \beta > 0$ and $\eta \in (0,1)$, is an (α, η, β)-refereed learning protocol for $\mathfrak{F}$ and $\mathfrak{D}$ with respect to ℓ and $\mathcal{O}_0, \mathcal{O}_1, \mathcal{O}_\mathcal{V}$. Moreover, $[\mathcal{P}_0, \mathcal{P}_1, \mathcal{V}]$ has verifier runtime and communication complexity $T(d, \alpha, \beta, \lambda) + \lambda \operatorname{poly} d$, and the verifier makes $q(d, \alpha, \beta, \lambda)$ queries to f.

Combining Proposition 6.2 with Theorems 4.2 and 4.4 yields a $(1+\varepsilon, \eta, \beta)$-refereed learning protocol for the zero-one loss, and a $(3+\varepsilon, \eta, \beta)$-refereed learning protocol for any M-bounded metric loss. In comparison to the protocols of Theorems 4.2 and 4.4, the new protocols work for any distribution, make the same number of queries to f, and only incur a cost of $d + \log \frac{\alpha M}{\eta}$ in the verifier runtime and communication complexity. In the remainder of the section we prove Proposition 6.2.

Proof. Let $[\mathcal{P}_0, \mathcal{P}_1, \mathcal{V}]$ be the following protocol:

$$\left[\mathcal{P}_0^{h_0,h_1,Q_\mathcal{D}}, \mathcal{P}_1^{h_0,h_1,Q_\mathcal{D}}, \mathcal{V}^{f,h_0,h_1,Q_\mathcal{D}}\right](d, \alpha, \eta, \beta)$$

1. $\mathcal{V}$: Set $\lambda \leftarrow 6d + \log \frac{\alpha M}{\eta}$. Execute certifiable sum (Lemma 3.2) with $t(x) = \lfloor D \rfloor_\lambda$ and $\lambda \leftarrow \lambda$ to obtain $T_\lambda = \sum_{x \in \{0,1\}^d} \lfloor \mathcal{D}(x) \rfloor_\lambda$.
2. $\mathcal{V}, \mathcal{P}_0, \mathcal{P}_1$: Using ℓ, T_λ, and query access to $Q_\mathcal{D}$, simulate $[\mathcal{P}'_0, \mathcal{P}'_1, \mathcal{V}']_{d,\alpha,\beta,\lambda}$ by providing access to ℓ_λ and query access to $Q_{\mathcal{D}|_\lambda}$.
3. Output the result of the simulation.

Protocol 8: Protocol for arbitrary precision $\mathcal{D}$ and ℓ

By Lemma 3.2, verifier $\mathcal{V}$ correctly obtains T_λ, and hence can provide the required query access to $Q_{\mathcal{D}_\lambda}$. Moreover, the overhead in runtime an communication cost is simply $\lambda \operatorname{poly} d$.

It remains to show that if $\mathcal{L}_\mathcal{D}(h_{1-b}, f \mid \ell) > \alpha \mathcal{L}_\mathcal{D}(h_b, f \mid \ell) + \eta$ for some $b \in \{0,1\}$, then $[\mathcal{P}_0, \mathcal{P}_1, \mathcal{V}]$ outputs b with probability at least $1 - \beta$. We will argue that the loss functions $\mathcal{L}_\mathcal{D}(f, h \mid \ell)$ and $\mathcal{L}_{\mathcal{D}_\lambda}(f, h \mid \ell_\lambda)$ are close. Let $a(x) = \ell(f(x), h(x)) - \ell_\lambda(f(x), h(x))$ and let $b(x) = \mathcal{D}(x) - \mathcal{D}_\lambda(x)$ for all $x \in \{0,1\}^d$. Then by the definition of $\lfloor \cdot \rfloor_\lambda$ and Claim 3.7, we have $|a(x)| \le 2^{-\lambda}$ and $|b(x)| \le 2^{d+1-\lambda}$,

and thus,

$$\begin{aligned}
|\mathcal{L}_{\mathcal{D}}(f,h \mid \ell) - \mathcal{L}_{\mathcal{D}_\lambda}(f,h \mid \ell_\lambda)| &= \left| \sum_{x\in\{0,1\}^d} \ell(f(x),h(x))\mathcal{D}(x) - \ell_\lambda(f(x),h(x))\mathcal{D}_\lambda(x) \right| \\
&= \left| \sum_{x\in\{0,1\}^d} \ell(f(x),h(x))\mathcal{D}(x) - (\ell(f(x),h(x)) - a(x))(\mathcal{D}(x) - b(x)) \right| \\
&= \left| \sum_{x\in\{0,1\}^d} a(x)\mathcal{D}(x) + \ell(f(x),h(x))b(x) - a(x)b(x) \right| \\
&\leq |a(x)| + 2^d M|b(x)| + 2^d|a(x)b(x)| \\
&\leq 2^{-\lambda} + 2^{2d+1-\lambda}M + 2^{2d+1-2\lambda} \leq 2^{5d-\lambda}M.
\end{aligned}$$

Let $r_b = \mathcal{L}_{\mathcal{D}}(f,h_b \mid \ell) - \mathcal{L}_{\mathcal{D}_\lambda}(f,h_b \mid \ell_\lambda)$ for each $b \in \{0,1\}$. If $\mathcal{L}_{\mathcal{D}}(h_{1-b},f \mid \ell) > \alpha\mathcal{L}_{\mathcal{D}}(h_b,f \mid \ell) + \eta$, then $\mathcal{L}_{\mathcal{D}_\lambda}(h_{1-b},f \mid \ell_\lambda) > \alpha\mathcal{L}_{\mathcal{D}_\lambda}(h_b,f \mid \ell) + \eta - |r_{1-b}| - \alpha|r_b|$. By the above reasoning we have $|r_{1-b}| + \alpha|r_b| \leq \alpha 2^{6d-\lambda}M \leq \eta$, and hence $\eta - |r_{1-b}| - \alpha|r_b| \geq 0$. It follows that $\mathcal{L}_{\mathcal{D}_\lambda}(h_{1-b},f \mid \ell_\lambda) \geq \alpha\mathcal{L}_{\mathcal{D}_\lambda}(h_b,f \mid \ell)$, and therefore $[\mathcal{P}'_0,\mathcal{P}'_1,\mathcal{V}']_{d,\alpha,\beta,\lambda}$ will output b with probability at least $1-\beta$. □

7 Protocols with additive and mixed error

In this section, we describe two protocols for the additive error setting, i.e., the setting where $\eta > 0$. We restrict our focus to the setting with the zero-one loss function ℓ_{zo}.

First, we briefly consider additive error in the single prover setting. That is, given query access to h_0, h_1, and f, a verifier $\mathcal{V}$, with the help of a prover $\mathcal{P}$, would like to decide which of h_0 and h_1 has better loss on f. Prior work of [GRSY21] gives a protocol for empirical risk minimization in the single prover setting that can be easily adapted to compare h_0 and h_1. At a high level, the protocol works as follows:[8] (1) The verifier draws $\Theta\left(\frac{1}{\eta^2}\right)$ unlabeled samples from $\mathcal{D}$ and $\Theta\left(\frac{1}{\eta}\right)$ labeled samples $(x,f(x))$ where $x \sim \mathcal{D}$, and sends the samples (not including labels) to the prover. (2) The prover labels the samples using f and sends them back to the verifier. (3) The verifier checks that the prover's labels agree with its own set of labeled samples. If the labels disagree then the verifier rejects. Otherwise, the verifier uses the labels, along with query access to h_0 and h_1, to determine which of h_0 and h_1 achieves smaller loss on f.

While the aforementioned protocol can be efficient for constant η, when η is too small the requirements that the verifier draw $\frac{1}{\eta}$ labeled samples and that the prover make $\frac{1}{\eta^2}$ queries to f may be prohibitive. In Propositions 7.1 and 7.2, we show that in the two prover setting where the verifier has query access to f, one can achieve a considerably better dependence on η. First, in Proposition 7.1, we show that in the additive error setting ($\alpha = 1, \eta > 0$), the verifier can replace the $\frac{1}{\eta}$ labeled samples with a single query to f. The protocol of Proposition 7.1 improves the efficiency of the verifier, but it still has provers that make $\frac{1}{\eta^2}$ queries to f. In contrast, in Proposition 7.2 we show that in the mixed additive/multiplicative error setting ($\alpha = 1+\varepsilon, \eta > 0$), the provers need only make $\frac{1}{\varepsilon^2\eta} + \frac{1}{\eta}$ queries to f, and the verifier still only makes a single query to f. For constant ε, this significantly improves the query complexity of the provers.

[8] We describe a modified version of their protocol tailored to our setting. For a complete description of the protocol see the proof of Claim 5.2 (Simple Query Delegation) in [GRSY21].

7.1 Additive error ($\alpha = 1, \eta > 0$)

Below, we consider an *additive-error guarantee* with $\alpha = 1$ and $\eta > 0$. We show a refereed learning protocol for this setting when the provers and verifier both have query access to f. In this protocol, both the prover and verifier are efficient.

At a high level, the protocol works as follows. The verifier draws $\frac{1}{\eta^2}$ *unlabeled* samples from $\mathcal{D}$ and executes refereed query delegation (Lemma 3.8) to obtain their labels. The verifier outputs the hypothesis with smaller loss on the labeled sample.

Proposition 7.1 (Additive error)**.** *For each $b \in \{0,1\}$ let $\mathcal{O}_b(f, h_0, h_1, \mathcal{D})$ provide query access to f, and let $\mathcal{O}_{\mathcal{V}}(f, h_0, h_1, \mathcal{D})$ provide sample access to $\mathcal{D}$ and query access to f, h_0 and h_1. Fix range $\mathcal{Y}$. There exists a protocol $[\mathcal{P}_0, \mathcal{P}_1, \mathcal{V}]$ that, for all inputs $d \in \mathbb{N}$, and $\eta, \beta \in (0,1)$, is a $(1, \eta, \beta)$-refereed learning protocol for $\mathfrak{F}$ and $\mathfrak{D}$ with respect to ℓ_{zo} and oracles $\mathcal{O}_0, \mathcal{O}_1$, and $\mathcal{O}_{\mathcal{V}}$. The protocol has the following guarantees:*

- *The verifier draws $O\big(\frac{1}{\eta^2}\log\frac{1}{\beta}\big)$ samples from $\mathcal{D}$, has runtime $O\big(\frac{1}{\eta^2}\log\frac{1}{\beta}\big)$, and makes 1 query to f.*
- *The provers make $O\big(\frac{1}{\eta^2}\log\frac{1}{\beta}\big)$ queries to f and have runtime $O\big(\frac{1}{\eta^2}\log\frac{1}{\beta}\big)$.*
- *The protocol has communication complexity $O\big((d + \log|\mathcal{Y}|)\cdot\frac{1}{\eta^2}\log\frac{1}{\beta}\big)$.*

Proof of Proposition 7.1. We use the following protocol. Let $c > 0$ be a sufficiently large absolute constant.

$$\Big[\mathcal{P}_0^f, \mathcal{P}_1^f, \mathcal{V}^{f,h_0,h_1,\mathcal{D}}\Big](d, \eta, \beta)$$

1. $\mathcal{V}$: Let $m = \frac{c\log 1/\beta}{\eta^2}$. Draw m samples $x_1, \ldots, x_m \sim \mathcal{D}$ and send them to $\mathcal{P}_0$ and $\mathcal{P}_1$.
2. $\mathcal{V}$: Execute refereed query delegation (Lemma 3.8) to simulate the protocol where the verifier queries f on $(x_1, \ldots, x_m)$ and obtains $\{(x_i, f(x_i))\}_{i\in[m]}$, using 1 query to f.
3. $\mathcal{V}$: Return $\rho \leftarrow \arg\min_{b\in\{0,1\}} |\{i \in [m] \mid h_b(x_i) \neq f(x_i)\}|$.

Protocol 9: refereed learning with additive error

For each $b \in \{0,1\}$ let $p_b = \Pr_{x\sim\mathcal{D}}[h_b(x) \neq f(x)]$, and assume without loss of generality that $p_{1-s} \geq p_s + \eta$ for some $s \in \{0,1\}$. We will show that $\rho = s$ with probability at least $1 - \beta$.

Let $\widehat{p}_b = \frac{1}{m}|\{i \in [m] : h_b(x_i) \neq f(x_i)\}|$. By Hoeffding's inequality, our choice of m, and the fact that $\mathbb{E}[\widehat{p}_b] = p_b$ we have, $\Pr\left[|\widehat{p}_b - p_b| \geq \frac{\eta}{4}\right] \leq 2\exp(-m\eta^2/8) < \beta$. Thus, with probability at least $1 - \beta$, we have $|\widehat{p}_b - p_b| \leq \eta/4$ for each $b \in \{0,1\}$. Since we assumed $p_{1-s} \geq p_s + \eta$, this implies that $\widehat{p}_s < \widehat{p}_{1-s}$, and that $\mathcal{V}$ outputs $\rho = s$ with probability at least $1 - \beta$. The sample, communication, query, and time complexity guarantees follow from Lemma 3.8 and by inspection of Protocol 9. □

7.2 Mixed additive and multiplicative error ($\alpha > 1, \eta > 0$)

Below, we consider a *mixed-error guarantee* with both $\alpha > 1$ and $\eta > 0$. When the prover has query access to f, we construct a refereed learning protocol with efficient provers and an efficient verifier that only needs sample access to $\mathcal{D}$ and a single query to f. In contrast to the setting where $\alpha = 1$

where the prover makes $\frac{1}{\eta^2}$ queries to f (see Proposition 7.1), the prover in Proposition 7.2 works for $\alpha = 1+\varepsilon$ and need only make $1+\frac{1}{\varepsilon^2}$ queries to f.

Proposition 7.2 (Mixed error)**.** *For each $b \in \{0,1\}$ let $\mathcal{O}_b(f, h_0, h_1, \mathcal{D})$ provide query access to f, and let $\mathcal{O}_{\mathcal{V}}(f, h_0, h_1, \mathcal{D})$ provide sample access to $\mathcal{D}$ and query access to h_0, h_1, and f. Fix range $\mathcal{Y}$. There exists a protocol $[\mathcal{P}_0, \mathcal{P}_1, \mathcal{V}]$ that, for all $d \in \mathbb{N}$, $\varepsilon > 0$ and $\eta, \beta \in (0,1)$, is a $(1+\varepsilon, \eta, \beta)$-refereed learning protocol for $\mathfrak{F}$ and $\mathfrak{D}$ with respect to ℓ_{zo} and oracles $\mathcal{O}_0, \mathcal{O}_1$, and $\mathcal{O}_{\mathcal{V}}$. The protocol has the following guarantees:*

- *The verifier draws $O\big(\big(1+\frac{1}{\varepsilon^2}\big)\cdot\frac{\log 1/\beta}{\eta}\big)$ samples from $\mathcal{D}$, has runtime $O\big(\big(1+\frac{1}{\varepsilon^2}\big)\cdot\frac{\log 1/\beta}{\eta}\big)$, and makes 1 query to f.*
- *The provers make $O\big(\big(1+\frac{1}{\varepsilon^2}\big)\log\frac{1}{\beta}\big)$ queries to f and have runtime $O\big(\big(1+\frac{1}{\varepsilon^2}\big)\log\frac{1}{\beta}\big)$.*
- *The protocol has communication complexity $O\Big(\big(d+\log|\mathcal{Y}|\big)\cdot\big(1+\frac{1}{\varepsilon^2}\big)\log\frac{1}{\beta}\Big)$.*

Proof of Proposition 7.2. Let $S = \{h_0(x) \neq h_1(x)\}$. At a high level, the proof proceeds by arguing that the verifier can efficiently generate $\Theta\big(1+\frac{1}{\varepsilon^2}\big)$ unlabeled samples from S. The verifier can then execute refereed query delegation (Lemma 3.8) to obtain labeled samples. By the same argument as in the proof of Theorem 4.2, the verifier can determine which of h_0 or h_1 has better loss on f (up to a multiplicative constant) without making any additional queries.

Let $c > 0$ be a sufficiently large absolute constant and consider the following protocol:

$$\Big[\mathcal{P}_0^f, \mathcal{P}_1^f, \mathcal{V}^{f,h_0,h_1,\mathcal{D}}\Big](d,\varepsilon,\eta,\beta)$$

1. $\mathcal{V}$: Set $m = c\cdot\Big(\frac{2(2+\varepsilon)}{\varepsilon}\Big)^2\log\frac{1}{\beta}$ and $t \leftarrow \frac{2m}{\eta^2}$. Draw t samples $x_1,\dots,x_t \sim \mathcal{D}$, and let $x_1,\dots,x_m$ denote the first m samples in $S = \{x \mid h_0(x) \neq h_1(x)\}$.[a] If fewer than m samples are in S then output $\rho \sim \{0,1\}$.
2. $\mathcal{V}$: Execute refereed query delegation (Lemma 3.8) to simulate the protocol where the verifier queries f on $(x_1,\dots,x_m)$ and obtains $\{(x_i, f(x_i))\}_{i\in[m]}$, using 1 query to f.
3. $\mathcal{V}$: Return $\rho \leftarrow \arg\min_{b\in\{0,1\}}|\{i\in[m] : h_b(x_i) \neq f(x_i)\}|$.

[a] The verifier need not construct S to determine membership, and can instead query h_0 and h_1 for each x_i in the sample.

Protocol 10: refereed learning with mixed error

Let $\mathcal{L}_b = \Pr_{x\sim\mathcal{D}}[h_b(x) \neq f(x)]$ and assume that $\mathcal{L}_1 > (1+\varepsilon)\mathcal{L}_0 + \eta$ (a symmetric argument suffices for the case when $\mathcal{L}_0 > (1+\varepsilon)\mathcal{L}_1 + \eta$). We show that the verifier outputs $\rho = 0$ with probability at least $1-\beta$. By the triangle inequality and the fact that $\mathcal{L}_0 \geq 0$ we have

$$\Pr_{x\sim\mathcal{D}}[h_1(x) \neq h_0(x)] \geq \mathcal{L}_1 - \mathcal{L}_0 > \eta. \tag{5}$$

We first argue that after t samples, the verifier obtains $x_1,\dots,x_m \sim \mathcal{D}|_S$ with probability at least 0.9. Let $K = \sum_{i\in[t]}\mathbb{1}[x_i \in S]$. By (5) we have $\mathbb{E}[K] = \eta\cdot t$, and thus by Hoeffding's inequality,

$\Pr[|K - \eta t| \geq \eta t/2] \leq 2\exp\left(-\eta^2 t/2\right) \leq \frac{\beta}{2}$, and hence, $K > \eta \cdot t/2 \geq m$ with probability at least $1 - \frac{\beta}{2}$.

Next, we argue that if the verifier obtains $x_1, \ldots, x_m \sim \mathcal{D}|_S$, then $\rho = 0$ with probability at least $1 - \frac{\beta}{2}$. By Lemma 3.8, the verifier correctly obtains $\{(x_i, f(x_i)\}_{i\in[m]}$. By Claim 4.3, since $\mathcal{L}_1 > (1+\varepsilon)\mathcal{L}_0$ we have $\Pr_{x\sim\mathcal{D}|_S}[h_1 \neq f(x)] > \frac{1}{2} + \frac{\varepsilon}{2(2+\varepsilon)}$. As in the proof of Theorem 4.2, if $\widehat{p} = \frac{1}{m}\sum_{i\in[m]} \mathbb{1}[h_1(x_i) \neq f(x_i)]$, then by Hoeffding's inequality with $\delta = \frac{\varepsilon}{2(2+\varepsilon)}$ we have $\Pr[|\widehat{p} - \mathbb{E}[\widehat{p}] \geq \delta] \leq 2\exp\left(-2m\delta^2\right)$, which, by our setting of m, is at most $\frac{\beta}{2}$. Hence, $\widehat{p} > \frac{1}{2}$ with probability at least $1 - \frac{\beta}{2}$, and thus the verifier will output $\rho = 0$. Combining the above two arguments we see that $\rho = 0$ with probability at least $1 - \beta$. The runtime, query complexity, and communication complexity follow from Lemma 3.8 and by inspection of Protocol 10. □

Acknowledgments

We thank Mark Bun for simplifying the proof of Claim 4.6.

References

[AAT+25] Arasu Arun, Adam St. Arnaud, Alexey Titov, Brian Wilcox, Viktor Kolobaric, Marc Brinkmann, Oguzhan Ersoy, Ben Fielding, and Joseph Bonneau. Verde: Verification via refereed delegation for machine learning programs. *CoRR*, abs/2502.19405, 2025. URL: https://doi.org/10.48550/arXiv.2502.19405, arXiv:2502.19405, doi:10.48550/ARXIV.2502.19405. (Cited on p. 10.)

[Bir87] Lucien Birgé. On the Risk of Histograms for Estimating Decreasing Densities. *The Annals of Statistics*, 15(3):1013 – 1022, 1987. doi:10.1214/aos/1176350489. (Cited on pp. 8 and 13.)

[Can20] Clément L. Canonne. *A Survey on Distribution Testing: Your Data is Big. But is it Blue?* Number 9 in Graduate Surveys. Theory of Computing Library, 2020. URL: http://www.theoryofcomputing.org/library.html, doi:10.4086/toc.gs.2020.009. (Cited on p. 13.)

[CK21] Ran Canetti and Ari Karchmer. Covert learning: How to learn with an untrusted intermediary. In Kobbi Nissim and Brent Waters, editors, *Theory of Cryptography - 19th International Conference, TCC 2021, Raleigh, NC, USA, November 8-11, 2021, Proceedings, Part III*, volume 13044 of *Lecture Notes in Computer Science*, pages 1–31. Springer, 2021. doi:10.1007/978-3-030-90456-2_1. (Cited on p. 3.)

[CRR11] Ran Canetti, Ben Riva, and Guy N. Rothblum. Practical delegation of computation using multiple servers. In Yan Chen, George Danezis, and Vitaly Shmatikov, editors, *Proceedings of the 18th ACM Conference on Computer and Communications Security, CCS 2011, Chicago, Illinois, USA, October 17-21, 2011*, pages 445–454. ACM, 2011. doi:10.1145/2046707.2046759. (Cited on pp. 3 and 9.)

[CRR13] Ran Canetti, Ben Riva, and Guy N. Rothblum. Refereed delegation of computation. *Inf. Comput.*, 226:16–36, 2013. URL: https://www.cs.tau.ac.il/~canetti/CRR11.pdf, doi:10.1016/J.IC.2013.03.003. (Cited on pp. 3 and 9.)

[EKR04] Funda Ergün, Ravi Kumar, and Ronitt Rubinfeld. Fast approximate probabilistically checkable proofs. *Inf. Comput.*, 189(2):135–159, 2004. URL: https://doi.org/10.1016/j.ic.2003.09.005, doi:10.1016/J.IC.2003.09.005. (Cited on pp. 3 and 10.)

[FK97] Uriel Feige and Joe Kilian. Making games short (extended abstract). In Frank Thomson Leighton and Peter W. Shor, editors, *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing, El Paso, Texas, USA, May 4-6, 1997*, pages 506–516. ACM, 1997. doi:10.1145/258533.258644. (Cited on pp. 3 and 9.)

[FST88] Uriel Feige, Adi Shamir, and Moshe Tennenholtz. The noisy oracle problem. In Shafi Goldwasser, editor, *Advances in Cryptology - CRYPTO '88, 8th Annual International Cryptology Conference, Santa Barbara, California, USA, August 21-25, 1988, Proceedings*, volume 403 of *Lecture Notes in Computer Science*, pages 284–296. Springer, 1988. doi:10.1007/0-387-34799-2_22. (Cited on pp. 3 and 9.)

[GCW+24] Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V. Chawla, Olaf Wiest, and Xiangliang Zhang. Large language model based multi-agents: A survey of progress and challenges, 2024. URL: https://arxiv.org/abs/2402.01680, arXiv:2402.01680. (Cited on p. 10.)

[GKR15] Shafi Goldwasser, Yael Tauman Kalai, and Guy N. Rothblum. Delegating computation: Interactive proofs for muggles. *J. ACM*, 62(4):27:1–27:64, 2015. doi:10.1145/2699436. (Cited on p. 3.)

[GRSY21] Shafi Goldwasser, Guy N. Rothblum, Jonathan Shafer, and Amir Yehudayoff. Interactive Proofs for Verifying Machine Learning. In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference (ITCS 2021)*, volume 185 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 41:1–41:19, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ITCS.2021.41, doi:10.4230/LIPIcs.ITCS.2021.41. (Cited on pp. 3, 5, 10, and 32.)

[HR22] Tal Herman and Guy N. Rothblum. Verifying the unseen: interactive proofs for label-invariant distribution properties. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 1208–1219. ACM, 2022. doi:10.1145/3519935.3519987. (Cited on pp. 3 and 10.)

[HR24a] Tal Herman and Guy Rothblum. How to verify any (reasonable) distribution property: Computationally sound argument systems for distributions, 2024. URL: https://arxiv.org/abs/2409.06594, arXiv:2409.06594. (Cited on pp. 3 and 10.)

[HR24b] Tal Herman and Guy N. Rothblum. Interactive proofs for general distribution properties. *Electron. Colloquium Comput. Complex.*, TR24-094, 2024. URL: https://eccc.weizmann.ac.il/report/2024/094, arXiv:TR24-094. (Cited on pp. 3 and 10.)

[ICA18] Geoffrey Irving, Paul Christiano, and Dario Amodei. Ai safety via debate, 2018. URL: https://arxiv.org/abs/1805.00899, arXiv:1805.00899. (Cited on p. 10.)

[Kil92] Joe Kilian. A note on efficient zero-knowledge proofs and arguments (extended abstract). In S. Rao Kosaraju, Mike Fellows, Avi Wigderson, and John A. Ellis, editors, *Proceedings of the 24th Annual ACM Symposium on Theory of Computing, May 4-6, 1992, Victoria, British Columbia, Canada*, pages 723–732. ACM, 1992. doi:10.1145/129712.129782. (Cited on p. 3.)

[KR14] Gillat Kol and Ran Raz. Competing-provers protocols for circuit evaluation. *Theory Comput.*, 10:107–131, 2014. URL: https://doi.org/10.4086/toc.2014.v010a005, doi:10.4086/TOC.2014.V010A005. (Cited on pp. 3 and 9.)

[Mic94] Silvio Micali. CS proofs (extended abstracts). In *35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, November 20-22, 1994*, pages 436–453. IEEE Computer Society, 1994. doi:10.1109/SFCS.1994.365746. (Cited on p. 3.)

[RS06] Sofya Raskhodnikova and Adam D. Smith. A note on adaptivity in testing properties of bounded degree graphs. *Electron. Colloquium Computational Complexity*, 13(089), 2006. URL: http://eccc.hpi-web.de/eccc-reports/2006/TR06-089/index.html. (Cited on p. 25.)

[RVW13] Guy N. Rothblum, Salil P. Vadhan, and Avi Wigderson. Interactive proofs of proximity: delegating computation in sublinear time. In Dan Boneh, Tim Roughgarden, and Joan Feigenbaum, editors, *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 793–802. ACM, 2013. doi:10.1145/2488608.2488709. (Cited on pp. 3 and 10.)

[WB15] Michael Walfish and Andrew J. Blumberg. Verifying computations without reexecuting them. *Commun. ACM*, 58(2):74–84, 2015. doi:10.1145/2641562. (Cited on p. 3.)