Electronic Colloquium on Computational Complexity, Report No. 58 (2020)

Interactive Proofs for Verifying Machine Learning

Shafi Goldwasser[1], Guy N. Rothblum[2], Jonathan Shafer[1], and Amir Yehudayoff[3]

[1]UC Berkeley
[2]Weizmann Institute of Science
[3]Technion–IIT

April 24, 2020

Abstract

We consider the following question: using a source of labeled data and interaction with an untrusted prover, what is the complexity of verifying that a given hypothesis is "approximately correct"? We study interactive proof systems for *PAC verification*, where a verifier that interacts with a prover is required to accept good hypotheses, and reject bad hypotheses. Both the verifier and the prover are efficient and have access to data samples from an unknown distribution. We are interested in cases where the verifier can use significantly less data than is required for (agnostic) PAC learning, or use a substantially cheaper data source (e.g., using only random samples for verification, even though learning requires membership queries). We believe that today, when data and data-driven algorithms are quickly gaining prominence, the question of verifying purported outcomes of data analyses is very well-motivated.

We show three main results. First, we prove that for a specific hypothesis class, verification is significantly cheaper than learning in terms of the number of random samples required, even if the verifier engages with the prover only in a single-round (NP-like) protocol. Moreover, for this class we prove that single-round verification is also significantly cheaper than testing closeness to the class. Second, for the broad class of Fourier-sparse boolean functions, we show a multi-round (IP-like) verification protocol, where the prover uses membership queries, and the verifier is able to assess the result while only using random samples. Third, we show that verification is not always more efficient. Namely, we show a class of functions where verification requires as many samples as learning does, up to a logarithmic factor.

[1] Email: {shafi.goldwasser,shaferjo}@berkeley.edu.
[2] Email: rothblum@alum.mit.edu.
[3] Email: amir.yehudayoff@gmail.com.

ISSN 1433-8092

Contents

> *A simple idea underpins science: "trust, but verify". Results should always be subject to challenge from experiment. That simple but powerful idea has generated a vast body of knowledge. Since its birth in the 17th century, modern science has changed the world beyond recognition, and overwhelmingly for the better. But success can breed complacency. Modern scientists are doing too much trusting and not enough verifying – to the detriment of the whole of science, and of humanity.*
>
> The Economist, "How Science Goes Wrong" (2013)

1 Introduction

Data and data-driven algorithms are transforming science and society. State-of-the-art machine learning and statistical analysis algorithms use access to data at scales and granularities that would have been unimaginable even a few years ago: from medical records and genomic information to financial transactions and transportation networks. This revolution spans scientific studies, commercial applications and the operation of governments. It holds transformational promise, but also raises new concerns. If data analysis requires huge amounts of data and computational power, how can one verify the correctness and accuracy of the results? Might there be asymmetric cases, where performing the analysis is expensive, but verification is less costly?

There are many types of statistical analyses, and many ways to formalize the notion of verifying the outcome. In this work we focus on interactive proof systems (Goldwasser, Micali, & Rackoff, 1989) for verifying supervised learning, as defined by the PAC model of learning (L. G. Valiant, 1984). Our emphasis throughout is on access to the underlying data distribution as the critical resource: both quantitatively (how many samples are used for learning versus for verification), and qualitatively (what types of samples are used). We embark on tackling a series of new questions:

Suppose a learner (which we also call "prover") claims to have arrived at a good hypothesis with regard to an unknown data distribution by analyzing random samples from the distribution. Can one verify the quality of the hypothesis with respect to the unknown distribution by using significantly fewer samples than the number needed to independently repeat the analysis? The crucial difference between this question and questions that appear in the property testing and distribution testing literature is that we allow the prover and verifier to engage in an *interactive* communication protocol (see Section 1.1.1 for a comparison). We are interested in the case where both the verifier and an honest prover are efficient (i.e., use polynomial runtime and sample complexity), and furthermore, a dishonest prover with unbounded computational resources cannot fool the verifier:

Question 1 (**Runtime and sample complexity of learning vs. verifying**). *Are there machine learning tasks for which the runtime and sample complexity of learning a good hypothesis is significantly larger than the complexity of verifying a hypothesis provided by someone else?*

In the learning theory literature, various types of access to training data have been considered, such as random samples, membership queries, and statistical queries. It is interesting to consider whether it is possible to verify a hypothesis using a weaker type of access than is necessary for learning:

Question 2 (**Sample type of learning vs. verifying**). *Are there machine learning problems where membership queries are necessary for finding a good hypothesis, but verification is possible using random samples alone?*

The answers to these fundamental questions are motivated by real-world applications. If data analysis requires huge amounts of data and computational resources while verification is a simpler task, then a natural approach for individuals and weaker entities would be to delegate the data collection and analysis to more powerful entities. Going beyond supervised learning, this applies also to verifying the results of scientific studies without replicating the entire experiment. We elaborate on these (and more) motivating applications in Section 1.2 below.

1.1 PAC Verification: A Proposed Model

Our primary focus in this work is verifying the results of agnostic supervised machine learning algorithms that receive a labeled dataset, and aim to learn a classifier that predicts the labels of unseen examples. We consider an interactive proof system for verification of PAC learning, which we call *PAC Verification* (see Definition 1.17). Here, the entity running the learning algorithms (which we refer to as the "prover" or the "learner") proves the correctness of the results by engaging in an interactive communication protocol with a verifier. One special case is where the prover only sends a single message constituting an (NP-like) certificate of correctness. The honest prover should be able to convince the verifier to accept its proposed hypothesis with high probability. A dishonest prover (even an unbounded one) should not be able to convince the verifier to accept a hypothesis that is not sufficiently good (as defined below), except with small probability over the verifier's random coins and samples. The proof system is interesting if the amount of resources used for verification is significantly smaller than what is needed for performing the learning task. We are especially interested in *doubly-efficient* proof systems (Goldwasser, Kalai, & Rothblum, 2015), where the honest prover also runs in polynomial time.

More formally, let $\mathcal{X}$ be a set, and consider a distribution $\mathcal{D}$ over samples of the form (x,y) where $x \in \mathcal{X}$ and $y \in \{0,1\}$. Assume there is some *hypothesis class* $\mathcal{H}$, which is a set of functions $\mathcal{X} \to \{0,1\}$, and we are interested in finding a function $h \in \mathcal{H}$ that predicts the label y given a previously unseen x with high accuracy with respect to $\mathcal{D}$. To capture this we use a *loss function* denoted $L_{\mathcal{D}}$, which maps every hypothesis h to a real-valued *loss* that quantifies how poorly $h(x)$ predicts y: $L_{\mathcal{D}}(h) = \mathbb{P}_{(x,y)\in\mathcal{D}}[h(x) \neq y]$. Our goal is to design protocols that guarantee that with high probability, a proposed hypothesis $\tilde{h}$ is accepted if and only if it is ε-*good*, which means that

$$L_{\mathcal{D}}(\tilde{h}) \leq L_{\mathcal{D}}(\mathcal{H}) + \varepsilon, \tag{1}$$

where $L_{\mathcal{D}}(\mathcal{H}) = \inf_{h \in \mathcal{H}} L_{\mathcal{D}}(h)$.

Note that in the *realizable case* (or promise case), in which we assume that $L_{\mathcal{D}}(\mathcal{H}) = 0$, this task is trivial. A natural thing for the verifier to do under this assumption is to take a few samples from $\mathcal{D}$, and accept if and only if $\tilde{h}$ classifies at most, say, an $\frac{9}{10}\varepsilon$-fraction of them incorrectly. From Hoeffding's inequality, taking $O\left(\frac{1}{\varepsilon^2}\right)$ samples is sufficient to ensure that with constant probability the *empirical loss*[1] of $\tilde{h}$ is close to the true loss, e.g. with probability at least $\frac{9}{10}$, the difference between the empirical loss and the true loss will be at most $\frac{\varepsilon}{10}$. Therefore, if $L_{\mathcal{D}}(\tilde{h}) \leq \frac{8}{10}\varepsilon$ then $\tilde{h}$ will be accepted with probability $\frac{9}{10}$, and if $L_{\mathcal{D}}(\tilde{h}) > \varepsilon$ then $\tilde{h}$ will be rejected with probability $\frac{9}{10}$. In contrast, PAC learning a hypothesis that with probability at least $\frac{9}{10}$ has loss at most ε requires $\Omega\left(\frac{d}{\varepsilon}\right)$ samples, where the parameter d, which is the VC dimension of the class, can be arbitrarily large.[2] Hence, we obtain an unbounded sample complexity and time complexity separation between learning and verifying in the realizable case.

Therefore, the focus of this paper is the *agnostic case*, where no assumptions are made regarding $L_{\mathcal{D}}(\mathcal{H})$. Here, things become more interesting, and deciding whether $\tilde{h}$ is ε-good is non-trivial. Indeed, the verifier can efficiently estimate $L_{\mathcal{D}}(\tilde{h})$ using Hoeffding's inequality, but estimating the term $L_{\mathcal{D}}(\mathcal{H})$ on the right hand side of (1) is considerably more challenging. If $\tilde{h}$ has a loss of say 15%, it could be an amazingly-good hypothesis compared to the other members of $\mathcal{H}$, or it could be very poor. Distinguishing between these two cases may be difficult when $\mathcal{H}$ is a large and complicated class.

1.1.1 Related Models

We discuss two related models studied in prior work, and their relationship to the PAC verification model proposed in this work.

Property Testing. Goldreich, Goldwasser, and Ron (1998) initiated the study of a property testing problem that naturally accompanies proper PAC learning: Given access to samples from an unknown distribution $\mathcal{D}$, decide whether $L_{\mathcal{D}}(\mathcal{H}) = 0$ or $L_{\mathcal{D}}(\mathcal{H}) \geq \varepsilon$ for some fixed hypothesis class $\mathcal{H}$. Further developments and variations appeared in Kearns and Ron (2000) and Balcan, Blais, Blum, and Yang (2012). Blum and Hu (2018) consider *tolerant* closeness testing and a related task of distance approximation (see Parnas, Ron, & Rubinfeld, 2006), where the algorithm is required to approximate $L_{\mathcal{D}}(\mathcal{H})$ up to a small additive error. As discussed above, the main challenge faced by the verifier in PAC verification is approximating $L_{\mathcal{D}}(\mathcal{H})$. However, there is a crucial difference between testing and PAC verification: In addition to taking samples from $\mathcal{D}$, the verifier in PAC verification can also interact with a prover, and thus PAC verification can (potentially) be easier than testing. Indeed, this difference is exemplified by the *proper* testing question, where we only need to distinguish the zero-loss case from large loss. As discussed above, proper PAC verification is trivial. Proper testing, one the other hand, can be a challenging goal (and, indeed, has been the focus of a rich body of work). For the *tolerant* setting, we prove a separation between testing and

[1] I.e., the fraction of the samples that is misclassified.

[2] See preliminaries in Section 1.6.2 for more about VC dimension.

PAC verification: we show a hypothesis class for which the help of the prover allows the verifier to save a (roughly) quadratic factor over the number of samples that are required for closeness testing or distance approximation. See Section 3 for further details.

Proofs of Proximity for Distributions. Chiesa and Gur (2018) study interactive proof systems for distribution testing. For some fixed property Π, the verifier receives samples from an unknown distribution $\mathcal{D}$, and interacts with a prover to decide whether $\mathcal{D} \in \Pi$ or whether $\mathcal{D}$ is ε-far in total variation distance from any distribution in Π. While that work does not consider machine learning, the question of verifying a lower bound ℓ on the loss of a hypothesis class can be viewed as a special case of distribution testing, where $\Pi = \{\mathcal{D} : L_{\mathcal{D}}(\mathcal{H}) \geq \ell\}$. Beyond our focus on PAC verification, an important distinction between the works is that in Chiesa and Gur's model and results, the honest prover's access to the distribution is unlimited – the honest prover can have complete information about the distribution. In this paper, we focus on doubly-efficient proofs, where the verifier and the honest prover must both be efficient in the number of data samples they require. With real-world applications in mind, this focus seems quite natural.

We survey further related works in Section 1.5.

1.2 Applications

The P vs. NP problem asks whether finding a solution ourselves is harder than verifying a solution supplied by someone else. It is natural to ask a similar question in data science: Are there machine learning problems for which learning a good hypothesis is harder than verifying one proposed by someone else? We find this theoretical motivation compelling in and of itself. Nevertheless, we now proceed to elaborate on a few more practical aspects of this question.

1.2.1 Delegation of Learning

In a commercial context, consider a scenario in which a client is interested in developing a machine learning (ML) model, and decides to outsource that task to a company P that provides ML services. For example, P promises to train a deep neural net using a big server farm. Furthermore, P claims to possess a large amount of high quality data that is not available to the client, and promises to use that data for training.

How could the client ascertain that a model provided by P is actually a good model? The client could use a general-purpose cryptographic delegation-of-computation protocol, but that would be insufficient. Indeed, a general-purpose delegation protocol can only ensure that P executed the computation as promised, but it cannot provide any guarantees about the quality of the outcome, and in particular cannot ensure that the outcome is ε-good: If P used skewed or otherwise low-quality training data (whether maliciously or inadvertently), a general-purpose delegation protocol has no way of detecting that. Moreover, even if the the data and the execution of the computation were both flawless, this still provides no guarantees on the quality of the output, because an ML model might have poor performance despite being trained as prescribed.[3,4]

A different solution could be to have P provide a proof to establish that its output is indeed ε-good. In cases where the resource gap between learning and verifying is significant enough, the client could cost-effectively verify the proof, obtaining sound guarantees on the quality of the ML model it is purchasing from P.

1.2.2 Verification of Scientific Studies

It has been claimed that many or most published research findings are false (Ioannidis, 2005). Others refer to an ongoing *replication crisis* (Pashler & Wagenmakers, 2012; Fidler & Wilcox, 2018), where many scientific studies are hard or impossible to replicate or reproduce. Addressing this crisis is a scientific and societal priority. While resolving this crisis is well beyond the scope of any single work, we suggest a novel mitigation approach: rather than requiring full re-execution for verification of an analysis or experiment, there are natural examples where verification can be much less expensive in terms of access to data samples and in terms of computational resources.[5] One can envision a process by which scientific papers include a proof (interactive or non-interactive) of the results, that allows the community to verify the results using a small number of samples (these samples can be drawn as part of the review process, or perhaps they can come from existing repositories). We hope that the first steps taken in this work can lead to further development of verification techniques and protocols.

1.2.3 Towards a Formal Theory of AI Safety

Another motivation comes from the fledgling field of AI safety, concerned with ensuring that AI systems will not cause harm to their operators and to humanity in general (Bostrom, 2017). Consider this salient example, due to Russell (2019, ch. 5): An AI system is tasked with finding a cure for cancer. If the system is smart enough, it might decide to forcefully induce cancer tumors into millions of living people as part of its R&D efforts; furthermore, it could easily anticipate and stifle any human attempts at resistance. Thus, the system may accomplish its mission of identifying a cure for cancer, and still be a monstrous disaster.

As a solution, we suggest that by employing an interactive proof system, an AI development project could formally prove that a proposed AI design will be well-aligned with the desired human utility function *before* activating the AI, and *without* needing to formally specify what the desired human utility function is. Further discussion of this idea appears in Appendix A.

[3] E.g., a neural network might get stuck at a local minimum.

[4] Additionally, note that state-of-the-art delegation protocols are not efficient enough at present to make it practicable to delegate intensive ML computations. See the survey by Walfish and Blumberg (2015) for progress and challenges in developing such systems.

[5] E.g., the case of learning a Gaussian as mentioned in Section 1.5, as well as our result in Lemma 2.5.

1.3 Our Setting

In this paper we consider the following form of interaction between a verifier and a prover.

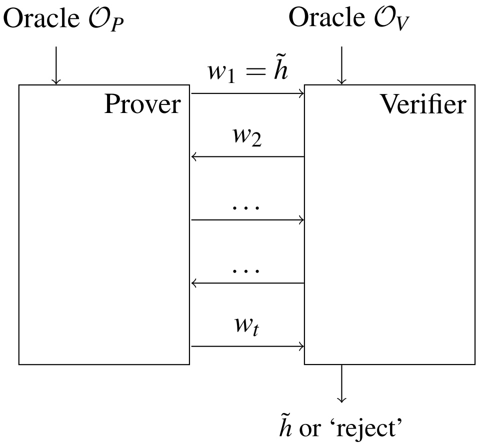

Figure 1: The verifier and prover each have access to an oracle, and they exchange messages with each other. Eventually, the verifier outputs a hypothesis, or rejects the interaction. One natural case is where the prover suggests a hypothesis $\tilde{h}$, and the verifier either accepts or rejects this suggestion.

Let $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ be a class of hypotheses, and let $\mathcal{D}$ be a distribution over $\mathcal{X} \times \{0,1\}$. The verifier and the prover each have access to an oracle, denoted $\mathcal{O}_V$ and $\mathcal{O}_P$ respectively. In the simplest case, both oracles provide i.i.d. samples from $\mathcal{D}$. That is, each time an oracle is accessed, it returns a sample from $\mathcal{D}$ taken independently of all previous samples and events. In addition, the verifier and prover each have access to a (private) random coin value, denoted ρ_V and ρ_P respectively, which are sampled from some known distributions over $\{0,1\}^*$ independently of each other and of all other events. During the interaction, the prover and verifier take turns sending each other messages $w_1, w_2, \ldots$, where $w_i \in \{0,1\}^*$ for all i. Finally, at some point during the exchange of messages, V halts and outputs either ‘reject’ or a hypothesis $h: \mathcal{X} \to \{0,1\}$. The goal of the verifier is to output an ε-*good* hypothesis, meaning that

$$L_{\mathcal{D}}(h) \leq L_{\mathcal{D}}(\mathcal{H}) + \varepsilon.$$

A natural special case of interest is when the prover's and verifier's oracles provide sample access to $\mathcal{D}$. The prover can learn a “good” hypothesis $\tilde{h}: \mathcal{X} \to \{0,1\}$ and send it to the verifier as its first message, as in Figure 1 above. The prover and verifier then exchange further messages, wherein the prover tries to convince the verifier that $\tilde{h}$ is ε-good, and the verifier tries to asses the veracity of that claim. If the verifier is convinced, it outputs $\tilde{h}$, otherwise it rejects.

We proceed with an informal definition of PAC verification (see full definitions in Section 1.7).

Before doing so, we first recall a relaxed variant of PAC learning, where we allow a multiplicative slack in the error guarantee. This is captured by an additional parameter $\alpha \geq 1$.

Definition (α-PAC Learnability – informal version of Definition 1.19). *A class of hypothesis $\mathcal{H}$ is* <u>*α-PAC learnable*</u> *if there exists an algorithm A such that for every distribution $\mathcal{D}$ and every $\varepsilon, \delta > 0$, with probability at least $1-\delta$, A outputs h that satisfies*

$$L_{\mathcal{D}}(h) \leq \alpha \cdot L_{\mathcal{D}}(\mathcal{H}) + \varepsilon. \tag{2}$$

PAC verification is the corresponding notion for interactive proof systems:

Definition (α-PAC Verifiability – informal version of Definition 1.17). *A class of hypothesis $\mathcal{H}$ is* <u>*α-PAC verifiable*</u> *if there exists a pair of algorithms (P, V) that satisfy the following conditions for every distribution $\mathcal{D}$ and every $\varepsilon, \delta > 0$:*

- ***Completeness.*** *After interacting with P, V outputs h such that with probability at least $1-\delta$, $h \neq \text{reject}$ and h satisfies (2).*
- ***Soundness.*** *After interacting with any (possibly unbounded) prover P', V outputs h such that with probability at least $1-\delta$, either $h = \text{reject}$ or h satisfies (2).*

Remark 1.1. *We insist on double efficiency; that is, that the sample complexity and running times of both V and P must be polynomial in $\frac{1}{\varepsilon}$, $\log\left(\frac{1}{\delta}\right)$, and perhaps also in some parameters that depend on $\mathcal{H}$, such as the VC dimension or Fourier sparsity of $\mathcal{H}$.*

1.4 Overview of Results

In this paper, we start charting the landscape of machine learning problems with respect to Questions 1 and 2 mentioned above. In Section 2 we provide evidence for an affirmative answer to Questions 2. We show an interactive proof system that efficiently verifies the class of Fourier-sparse boolean functions, where the prover uses an oracle that provides query access, and the verifier uses an oracle that only provides random samples. In this proof system, both the verifier and prover send and receive messages.

The class of Fourier-sparse functions is very broad, and includes decision trees, bounded-depth boolean circuits and many other important classes of functions. Moreover, the result is interesting because it supplements the widely-held learning parity with noise (LPN) assumption, which entails that PAC learning this class from random samples alone without the help of a prover is hard (see Blum, Kalai, & Wasserman, 2003; Yu & Steinberger, 2016).

Lemma (Informal version of Lemma 2.5). *Let $\mathcal{H}$ be the class of boolean functions $\{0,1\}^n \to \mathbb{R}$ that are t-sparse, as in Definition 1.15. Then $\mathcal{H}$ is 1-PAC verifiable with respect to the uniform distribution using a verifier that has access only to random samples of the form $(x, f(x))$, and a prover that has query access to f. The verifier in this protocol is not proper; the output is not necessarily t-sparse, but it is $\text{poly}(n,t)$-sparse. The number of samples used by the verifier, the number of queries made by the prover, and their running times are all bounded by*

$\operatorname{poly}\left(n, t, \log\left(\frac{1}{\delta}\right), \frac{1}{\varepsilon}\right)$.

Proof idea. The proof uses two standard tools, albeit in a less-standard way. The first standard tool is the Kushilevitz-Mansour algorithm (Kushilevitz & Mansour, 1993), which can PAC learn any t-sparse function using random samples, but only if the set of non-zero Fourier coefficients is *known*. The second standard tool is the Goldreich-Levin algorithm (Goldreich & Levin, 1989; Goldreich, 2007, Section 2.5.2.3), which can identify the set of non-zero Fourier coefficients, but requires *query access* in order to do so. The protocol combines the two tools in a manner that overcomes the limitations of each of them. First, the verifier executes the Goldreich-Levin algorithm, but whenever it needs to query the target function, it requests that the prover perform the query and send back the result. However, the verifier cannot trust the prover, and so the verifier engineers the queries in such a way that the answers to a certain random subset of the queries are known to the verifier based on its random sample access. This allows the verifier to detect dishonest provers. When the Goldreich-Levin algorithm terminates and outputs the set of non-zero coefficients, the verifier then feeds them as input to the Kushilevitz-Mansour algorithm to find an ε-good hypothesis using its random sample access. ■

In Section 3 we formally answer Question 1 affirmatively by showing that a certain simple class of functions (generalized thresholds) exhibits a quadratic gap in sample complexity between learning and verifying:

Lemma (Informal version of Lemma 3.8). *There exists a sequence of classes of functions*

$$\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3, \ldots \subseteq \{0,1\}^{\mathbb{R}}$$

such that for any fixed $\varepsilon, \delta \in (0, \frac{1}{2})$:

(i) The class $\mathcal{T}_d$ is 2-PAC verifiable, where both the verifier and prover have access to random samples, and the verifier requires only $\tilde{O}\left(\sqrt{d}\right)$ samples. Moreover, both the prover and verifier are efficient.

(ii) PAC learning the class $\mathcal{T}_d$ requires $\Omega(d)$ samples.

At this point, a perceptive reader would be justified in raising the following challenges. Perhaps 2-PAC verification requires less samples than 1-PAC learning simply because of the multiplicative slack factor of 2? Alternatively, perhaps the separation follows trivially from property testing results: maybe it is possible to achieve 2-PAC verification simply by having the verifier perform closeness testing using random samples, without needing the help of the prover except for finding the candidate hypothesis? The second part of the lemma dismisses both of these concerns.

Informal version of Lemma 3.8 – Continued. *Furthermore, for any fixed $\varepsilon, \delta \in (0, \frac{1}{2})$:*

(iii) 2-PAC learning the class $\mathcal{T}_d$ requires $\tilde{\Omega}(d)$ samples. This is true even if we assume that $L_{\mathcal{D}}(\mathcal{T}_d) > 0$, where $\mathcal{D}$ is the underlying distribution.[6]

(iv) Testing whether $L_{\mathcal{D}}(\mathcal{T}_d) \leq \alpha$ or $L_{\mathcal{D}}(\mathcal{T}_d) \geq \beta$ for any $0 < \alpha < \beta < \frac{1}{2}$ with success probability

[6] In the case where $L_{\mathcal{D}}(\mathcal{T}_d) = 0$, 2-PAC learning is the same as PAC learning, so the stronger lower bound in *(ii)* applies.

at least $1-\delta$ *when* $\mathcal{D}$ *is an unknown distribution (without the help of a prover) requires* $\tilde{\Omega}(d)$ *random samples from* $\mathcal{D}$.

Proof idea. *(ii)* follows from a standard application of Theorem 1.10, because $\mathsf{VC}(\mathcal{T}_d) = d$. *(iii)* follows by a reduction from *(iv)*. We prove *(iv)* by showing a further reduction from the problem of approximating the support size of a distribution, and applying a lower bound for that problem (see Theorem 3.19).

For *(i)*, recall from the introduction that the difficulty in designing a PAC verification proof system revolves around convincing the verifier that the term $L_{\mathcal{D}}(\mathcal{H})$ in Equation (1) is large. Therefore, we design our class $\mathcal{T}_d$ such that it admits a simple *certificate of loss*, which is a string that helps the verifier ascertain that $L_{\mathcal{D}}(\mathcal{H}) \geq \ell$ for some value ℓ.

To see how that works, first consider the simple class $\mathcal{T}$ of monotone increasing threshold functions $\mathbb{R} \to \{0,1\}$, as in Figure 2 on page 27 below. Observe that if there are two events $A = [0,a) \times \{1\}$ and $B = [b,1] \times \{0\}$ such that $a \leq b$ and $\mathcal{D}(A) = \mathcal{D}(B) = \ell$, then it must be the case that $L_{\mathcal{D}}(\mathcal{T}) \geq \ell$. This is true because $a \leq b$, and so if a monotone increasing threshold classifies any point in A correctly it must classify all point in B incorrectly. Furthermore, if the prover sends a description of A and B to the verifier, then the verifier can check, using a constant number of samples, that each of these events has weight approximately ℓ with high probability.

This type of certificate of loss can be generalized to the class $\mathcal{T}_d$, in which each function is a concatenation of d monotone increasing thresholds. A certificate of loss for $\mathcal{T}_d$ is simply a set of d certificates of loss $\{A_i, B_i\}_{i=1}^d$, one for each of the d thresholds. The question that arises at this point is how can the verifier verify d separate certificates while using only $\tilde{O}\left(\sqrt{d}\right)$ samples. This is performed using tools from distribution testing: the verifier checks whether the distribution of "errors" in the sets specified by the certificates is close to the prover's claims. I.e., whether the "weight" of 1-labels in each A_i and 0-labels in each B_i in the actual distribution, are close to the weights claimed by the prover. Using an identity tester for distributions this can be done using $O(\sqrt{d})$ samples (note that the identity tester need not be tolerant!). See Theorem D.1 for further details. ■

Finally, in Section 4 we show that verification is not always easier than learning:

Lemma (Informal version of Lemma 4.1). *There exists a sequence of classes* $\mathcal{H}_1, \mathcal{H}_2, \ldots$ *such that:*

- *It is possible to PAC learn the class* $\mathcal{H}_d$ *using* $\tilde{O}(d)$ *samples.*
- *For any interactive proof system that proper* 1*-PAC verifies* $\mathcal{H}_d$*, in which the verifier uses an oracle providing random samples, the verifier must use at least* $\Omega(d)$ *samples.*

Remark 1.2. *The lower bound on the sample complexity of the verifier holds regardless of what oracle is used by the prover.*

Proof idea. We specify a set $\mathcal{X}$ of cardinality $\Omega(d^2)$, and take $\mathcal{H}_d$ to be a randomly-chosen subset of all the balanced functions $\mathcal{X} \to \{0,1\}$ (i.e., functions f such that $|f^{-1}(0)| = |f^{-1}(1)|$).

The sample complexity of PAC learning $\mathcal{H}_d$ follows from its VC dimension being $\tilde{O}(d)$. For the lower bound, consider proper PAC verifying $\mathcal{H}_d$ in the special case where the distribution $\mathcal{D}$ satisfies $\mathbb{P}_{(x,y)\in\mathcal{D}}[y=1]=1$, but the marginal of $\mathcal{D}$ on $\mathcal{X}$ is unknown to the verifier. Because every hypothesis in the class assigns the incorrect label 0 to precisely half of the domain, a hypothesis achieves minimal loss if it assigns the 0 labels to a subset of size $\frac{|\mathcal{X}|}{2}$ that has minimal weight. Hence, the verifier must learn enough about the distribution to identify a specific subset of size $\frac{|\mathcal{X}|}{2}$ with weight close to minimal. We show that doing so requires $\Omega\left(\sqrt{|\mathcal{X}|}\right)=\Omega(d)$ samples. ■

1.5 Further Related Works

The growing role of data and predictive algorithms in a variety of fields has made the analysis of semi-unreliable data into a central research focus of the theoretical computer science (TCS) community. Recent research efforts that (broadly) fall into this theme include: (1) parameter estimation with greater emphasis on high dimensional data in the presence of partially unreliable data; (2) consideration of new corruption models such as list-decoding notions where some data is guaranteed to be properly sampled and the rest is subject to high error rate; (3) testing general properties of distributions beyond parameter estimation; and (4) analysis of machine learning algorithms with access to partially unreliable data. See Charikar, Steinhardt, and Valiant (2017); Diakonikolas et al. (2019, 2018); Ilyas, Jalal, Asteri, Daskalakis, and Dimakis (2017); Daskalakis, Gouleakis, Tzamos, and Zampetakis (2018). In contrast to all these directions, our focus is on interactive proof systems (or non-interactive certificates) by which an untrusted prover can convince a verifier that claimed results of a statistical analysis are correct, where the verifier is only allowed bounded access to the underlying data distribution.

A large body of work spanning the TCS and secure systems communities studies protocols for delegating computation to be performed by an untrusted prover (see e.g. Babai, Fortnow, Levin, & Szegedy, 1991; Micali, 1994; Goldwasser et al., 2015; Walfish & Blumberg, 2015). There are two significant differences between that line of work and the present paper. First, in these protocols the input is fixed and known to the prover and the verifier. The question is whether a computation was performed correctly on this (fixed and known) input. In contrast, in our setting there is no fixed and known input: the distribution $\mathcal{D}$ is unknown to the verifier, and can only be accessed by sampling. Second, we are interested in guaranteeing that a certain statistical conclusion is valid with respect to this unknown distribution, regardless of whether any specific algorithm was executed as promised. That is, if some known learning algorithm was executed by the prover and happened to produce a poor result (e.g. a neural network got stuck in a local minimum), this result should be rejected by the verifier despite being the outcome of a correct computation. One final contrast with the literature on delegating computations is that the focus there is on verifying general computations, and this generality often results in impractical protocols. One benefit of our focus on specific and structured machine learning problems is that this focus may result in tailored protocols (for important problems) with improved efficiency.

The setting we investigate bears some similarity to sublinear proof verification (see e.g. Ergün, Kumar, & Rubinfeld, 2004; Rothblum, Vadhan, & Wigderson, 2013), where the verifier cannot read the entire input. However, in that setting the verifier enjoys *query* access to its input, whereas in our setting the verifier only gets random samples (a much more limited form of access).

Another related result, in the area of parameter estimation, is due to Diakonikolas, Kane, and Stewart (2017, Appendix C). They proved a gap between the sample complexity of estimating and verifying the center of a Gaussian. The verifier is given a parameter $\tilde{\theta} \in \mathbb{R}^n$ and access to samples from an n-dimensional Gaussian distribution $\mathcal{N}(\theta, I)$. The verifier can distinguish between the case $\tilde{\theta} = \theta$ and the case $\|\tilde{\theta} - \theta\|_2 > \varepsilon$ using $O(\sqrt{n}/\varepsilon^2)$ samples. This contrasts with estimating θ up to an ε error from samples alone (without access to $\tilde{\theta}$), which requires $\Omega(n/\varepsilon^2)$ samples. They show that the result is sharp, and also can be generalized to a setting of tolerant testing.[7]

Finally, a paper by Axelrod, Garg, Sharan, and Valiant (2019) investigates a setting somewhat resembling ours. They consider the task of "amplifying" a set of samples taken from some unknown target distribution, that is, producing an additional synthetic dataset that appears as if it was drawn from the target distribution. The authors show that generating a dataset close in total variation distance to the target distribution can be done using less samples from the distribution than are necessary for learning the distribution up to the same total variation distance.

1.6 Preliminaries

1.6.1 Probability

Notation 1.3. *For any probability space $(\Omega, \mathcal{F})$, let $\Delta(\Omega, \mathcal{F})$ denote the set of all probability distributions over $(\Omega, \mathcal{F})$. We will often simply write $\Delta(\Omega)$ to denote this set when the σ-algebra $\mathcal{F}$ is understood.*

Definition 1.4. *Let $\mathcal{P}, \mathcal{Q} \in \Delta(\Omega, \mathcal{F})$. The total variation distance between $\mathcal{P}$ and $\mathcal{Q}$ is*

$$d_{\mathsf{TV}}(\mathcal{P}, \mathcal{Q}) = \sup_{X \in \mathcal{F}} \left|\mathcal{P}(X) - \mathcal{Q}(X)\right| = \frac{1}{2}\left\|\mathcal{P} - \mathcal{Q}\right\|_1.$$

1.6.2 PAC Learning

We use the *Probably Approximately Correct* (PAC) definition of learning, introduced by L. G. Valiant (1984). See Shalev-Shwartz and Ben-David (2014) for a textbook on learning theory.

Let $\mathcal{X}$ be a set, and let $\mathcal{H} \subseteq \mathbb{R}^{\mathcal{X}}$ be a class of functions, i.e. $\mathcal{H}$ is a subset of the functions $\mathcal{X} \to \mathbb{R}$. In this paper, we use the ℓ_2 loss function, which is popular in machine learning.

Definition 1.5. *Let $h \in \mathcal{H}$, and let $\mathcal{D} \in \Delta(\mathcal{X} \times \{0,1\})$. The loss of h with respect to $\mathcal{D}$ is $L_{\mathcal{D}}(h) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[(h(x) - y)^2\right]$. Furthermore, we denote $L_{\mathcal{D}}(\mathcal{H}) = \inf_{h \in \mathcal{H}} L_{\mathcal{D}}(h)$.*

[7] That is, distinguishing between the case $d \geq \varepsilon$ and $d \leq \varepsilon/2$ for $d = d_{\mathsf{TV}}\left(\mathcal{N}(\tilde{\theta}, I), \mathcal{N}(\theta, I)\right)$.

Remark 1.6. *In the special case of boolean labels, where $y \in \{0,1\}$ and $h : \mathcal{X} \to \{0,1\}$, the ℓ_2 loss function is the same as the* 0-1 *loss function:* $L_{\mathcal{D}}(h) = \mathbb{P}_{(x,y)\sim\mathcal{D}}[h(x) \neq y]$.

Definition 1.7. *We say that $\mathcal{H}$ is agnostically PAC learnable if there exist an algorithm A and a function $m\colon [0,1]^2 \to \mathbb{N}$ such that for any $\varepsilon, \delta > 0$ and any distribution $\mathcal{D} \in \Delta(\mathcal{X} \times \mathbb{R})$, if A receives as input a tuple of $m(\varepsilon, \delta)$ i.i.d. samples from $\mathcal{D}$, then A outputs a function $h \in \mathcal{H}$ satisfying*

$$\mathbb{P}[L_{\mathcal{D}}(h) \leq L_{\mathcal{D}}(\mathcal{H}) + \varepsilon] \geq 1 - \delta.$$

In words, this means that h is probably (with confidence $1-\delta$) approximately correct (has loss at most ε worse than optimal). The point-wise minimal such function m is called the sample complexity of $\mathcal{H}$.

The following definitions and result apply for the special case of boolean labels, where $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ and we only consider distributions $\mathcal{D} \in \Delta(\mathcal{X} \times \{0,1\})$.

Definition 1.8. *Let $h \in \mathcal{H}$ and $C \subseteq \mathcal{X}$. We denote by $h|_C$ the function $C \to \{0,1\}$ that agrees with h on C. The restriction of $\mathcal{H}$ to C is $\mathcal{H}|_C := \{h|_C : h \in \mathcal{H}\}$, and we say that $\mathcal{H}$ shatters C if $\mathcal{H}|_C = \{0,1\}^C$.*

Definition 1.9 (Vapnik & Chervonenkis, 1971)**.** *The VC dimension of $\mathcal{H}$ denoted $\mathsf{VC}(\mathcal{H})$ is the maximal size of a set $C \subseteq \mathcal{X}$ such that $\mathcal{H}$ shatters C. If $\mathcal{H}$ can shatter sets of arbitrary size, we say that the VC dimension is ∞.*

Theorem 1.10 (Blumer, Ehrenfeucht, Haussler, & Warmuth, 1989)**.** *Let $d = \mathsf{VC}(\mathcal{H})$. Then $\mathcal{H}$ is PAC learnable if and only if $d < \infty$, and furthermore, the sample complexity satisfies*

$$m(\varepsilon, \delta) = \Theta\left(\frac{d + \log\left(\frac{1}{\delta}\right)}{\varepsilon^2}\right).$$

1.6.3 Fourier Analysis of Boolean Functions

To formulate and prove Lemma 2.5 below, we need several basic notions from the Fourier analysis of boolean functions. For a comprehensive introduction, see O'Donnell (2014).

Consider the linear space $\mathcal{F}$ of all functions of the form $f\colon \{0,1\}^n \to \mathbb{R}$.

Fact 1.11. *The operator $\langle \cdot, \cdot \rangle : \mathcal{F}^2 \to \mathbb{R}$ given by $\langle f, g \rangle := \mathbb{E}_{x \in \{0,1\}^n}[f(x)g(x)]$ constitutes an inner product, where $x \in \{0,1\}^n$ denotes sampling from the uniform distribution.*

Notation 1.12. *For any set $S \subseteq [n]$, $\chi_S\colon \{0,1\}^n \to \{0,1\}$ denotes the function $\chi_S(x) := (-1)^{\sum_i x_i}$.*

Fact 1.13. *The set $\{\chi_S : S \subseteq [n]\}$ is an orthonormal basis of $\mathcal{F}$. In particular, any $f \in \mathcal{F}$ has a unique representation $f(x) = \sum_{S \subseteq [n]} \hat{f}(S)\chi_S(x)$, where $\hat{f}(S) = \langle f, \chi_S \rangle$.*

Fact 1.14 (Parseval's identity)**.** *Let $f \in \mathcal{F}$. Then $\langle f, f \rangle = \sum_{S \subseteq [n]} \hat{f}(S)^2$. In particular, if $f\colon \{0,1\}^n \to \{0,1\}$ then $\sum_{S \subseteq [n]} \hat{f}(S)^2 = \mathbb{E}_x[f(x)] \leq 1$.*

Definition 1.15. *Let $t \in \mathbb{N}$. A function $f\colon \{0,1\}^n \to \mathbb{R}$ is t-sparse if it has at most t non-zero*

Fourier coefficients, namely $|\{S \subseteq [n] : \hat{f}(S) \neq 0\}| \leq t$.

1.7 Definition of PAC Verification

In Section 1.3 we informally described the setting of this paper. Here, we complete that discussion by providing a formal definition of PAC verification, which is the main object of study in this paper.

Notation 1.16. *We write*

$$[V^{\mathcal{O}_V}(x_V), P^{\mathcal{O}_P}(x_P)]$$

for the random variable denoting the output of the verifier V *after interacting with a prover* P*, when* V *and* P *receive inputs* x_V *and* x_P *respectively, and have access to oracles* $\mathcal{O}_V$ *and* $\mathcal{O}_P$ *respectively. The inputs* x_V *and* x_P *can specify parameters of the interaction, such as the accuracy and confidence parameters* ε *and* δ*. This random variable takes values in* $\{0,1\}^{\mathcal{X}} \cup \{\text{reject}\}$*, namely, it is either a function* $\mathcal{X} \to \{0,1\}$ *or it is the value "*reject*". The random variable depends on the (possibly randomized) responses of the oracles, and on the random coins of* V *and* P*.*

For a distribution $\mathcal{D}$*, we write* $V^{\mathcal{D}}$ *(or* $P^{\mathcal{D}}$*) to denote use of an oracle that provides i.i.d. samples from the distributions* $\mathcal{D}$*. Likewise, for a function* f*, we write* V^f *(or* P^f*) to denote use of an oracle that provides query access to* f*. That is, in each access to the oracle,* V *(or* P*) sends some* $x \in \mathcal{X}$ *to the oracle, and receives the answer* $f(x)$*.*

We also write

$$[V(S_V, \rho_V), P(S_P, \rho_P)] \in \{0,1\}^{\mathcal{X}} \cup \{\text{reject}\}$$

to denote the deterministic output of the verifier V *after interacting with* P *in the case where* V *and* P *receive fixed random coin values* ρ_V *and* ρ_P *respectively, and receive fixed samples* S_V *and* S_P *from their oracles* $\mathcal{O}_V$ *and* $\mathcal{O}_P$ *respectively.*

We are interested in classes $\mathcal{H}$ for which an ε-good hypothesis can always be verified with high probability via this form of interaction between an efficient prover and verifier, as formalized in the following definition. Note that the following definitions include an additional multiplicative slack parameter $\alpha \geq 1$ in the error guarantee. This parameter does not exist in the standard definition of PAC learning; the standard definition corresponds to the case $\alpha = 1$.

Definition 1.17 **(α-PAC Verifiability).** *Let* $\mathcal{H} \subseteq \{0,1\}^{\mathcal{X}}$ *be a class of hypotheses, let* $\mathfrak{D} \subseteq \Delta(\mathcal{X} \times \{0,1\})$ *be some family of distributions, and let* $\alpha \geq 1$*. We say that* $\mathcal{H}$ *is* *α-PAC verifiable with respect to* $\mathfrak{D}$ *using oracles* $\mathcal{O}_V$ *and* $\mathcal{O}_P$ *if there exists a pair of algorithms* (V,P) *that satisfy the following conditions for every input* $\varepsilon, \delta > 0$*:*

- ***Completeness.*** *For any distribution* $\mathcal{D} \in \mathfrak{D}$*, the random variable* $h := [V^{\mathcal{O}_V}(\varepsilon,\delta), P^{\mathcal{O}_P}(\varepsilon,\delta)]$ *satisfies*

$$\mathbb{P}\left[h \neq \text{reject} \wedge \left(L_{\mathcal{D}}(h) \leq \alpha \cdot L_{\mathcal{D}}(\mathcal{H}) + \varepsilon\right)\right] \geq 1 - \delta.$$

- ***Soundness.*** *For any distribution* $\mathcal{D} \in \mathfrak{D}$ *and any (possibly unbounded) prover* P'*, the random* *variable* $h := [V^{\mathcal{O}_V}(\varepsilon,\delta), P'^{\mathcal{O}_P}(\varepsilon,\delta)]$ *satisfies*

$$\mathbb{P}\left[h \neq \text{reject} \wedge \Big(L_{\mathcal{D}}(h) > \alpha \cdot L_{\mathcal{D}}(\mathcal{H}) + \varepsilon\Big)\right] \leq \delta.$$

Definition 1.18 (Interactive Proof System for PAC Verification). *A pair of algorithms (V,P) satisfying soundness and completeness as above, is called an interactive proof system that α-PAC verifies $\mathcal{H}$ with respect to $\mathfrak{D}$ using oracles $\mathcal{O}_V$ and $\mathcal{O}_P$.*

Definition 1.19 (α-PAC Learnability). *Similarly, $\mathcal{H}$ is α-PAC learnable with respect to $\mathfrak{D}$ using oracle $\mathcal{O}$ if there exists an algorithm A that for every input $\varepsilon,\delta > 0$ and every $\mathcal{D} \in \mathfrak{D}$, outputs $h := A^{\mathcal{O}}(\varepsilon,\delta)$ such that $\mathbb{P}[L_{\mathcal{D}}(h) \leq \alpha \cdot L_{\mathcal{D}}(\mathcal{H}) + \varepsilon] \geq 1-\delta$.*

Remark 1.20. *Some comments about these definitions:*

- *The behavior of the oracles $\mathcal{O}_V$ and $\mathcal{O}_P$ may depend on the specific underlying distribution $\mathcal{D} \in \mathfrak{D}$, which is unknown to the prover and verifier. For example, they may provide samples from $\mathcal{D}$.*
- *We insist on double efficiency; that is, that the sample complexity and running times of both V and P must be polynomial in $\frac{1}{\varepsilon}$, $\log\left(\frac{1}{\delta}\right)$, and perhaps also in some parameters that depend on $\mathcal{H}$, such as the VC dimension or Fourier sparsity of $\mathcal{H}$.*
- *If for every $\varepsilon,\delta > 0$, and for any (possibly unbounded) prover P', the value $h := [V^{\mathcal{O}_V}(\varepsilon,\delta), P'^{\mathcal{O}_P}(\varepsilon,\delta)]$ satisfies $h \in \mathcal{H} \cup \{\text{reject}\}$ with probability 1 (i.e., V never outputs a function that is not in $\mathcal{H}$), then we say that $\mathcal{H}$ is proper α-PAC verifiable, and that the proof system proper α-PAC verifies $\mathcal{H}$.*

Remark 1.21. *An important type of learning (studied e.g. by Angluin, 1987 and Kushilevitz & Mansour, 1993) is* learning with membership queries with respect to the uniform distribution. *In this setting, the family $\mathfrak{D}$ consists of distributions $\mathcal{D}$ such that: (1) the marginal distribution of $\mathcal{D}$ over $\mathcal{X}$ is uniform; (2) $\mathcal{D}$ has a target function $f : \mathcal{X} \to \{1,-1\}$ satisfying $\mathbb{P}_{(x,y)\sim\mathcal{D}}[y = f(x)] = 1$.*[8] *In Section 2, we will consider protocols for this type of learning that have the form $[V^{\mathcal{D}}, P^f]$, such that the verifier has access to an oracle providing random samples from a distribution $\mathcal{D} \in \mathfrak{D}$, and the prover has access to an oracle providing query access to f, the target function of $\mathcal{D}$. This type of protocol models a real-world scenario where P has qualitatively more powerful access to training data than V.*

1.8 Organization of this Paper

In Section 1.7 we formally define interactive proofs for PAC verification. In Section 1.4 we provide an overview of our results and their respective proof ideas. Our first result appears in Section 3, where we answer Question 1 above affirmatively by showing that a certain simple

[8] Note that f is not necessarily a member of $\mathcal{H}$, so this is still an *agnostic* (rather than *realizable*) case.

class of functions (generalized thresholds) exhibits a quadratic gap in sample complexity between learning and verifying. The verifier for this class is an NP-like verifier, in the sense that it takes as input a succinct witness string that helps it reach a decision.

In Section 2 answer Question 2 affirmatively, showing that the broad and important class of Fourier-sparse boolean functions admits a doubly-efficient verification protocol in which the prover has query access, but the verifier only uses random samples. Note that according to the widely-held LPN assumption, learning this class is not possible without query access (see Section 1.6.3 for more about Fourier analysis, and Blum et al., 2003; Yu & Steinberger, 2016 for more about the LPN assumption).

Interestingly, however, verification is not always more efficient. In Section 4 we show a lower bound for a class of randomly-chosen functions, entailing that for this class, verification requires as many samples as learning does, up to a logarithmic factor.

2 Efficient Verification for the Class of Fourier-Sparse Functions

The class $\mathcal{T}_d$ of multi-thresholds shows that in some cases verification is strictly easier than learning and closeness testing. The verification protocol for $\mathcal{T}_d$ has a single round, where the prover simply sends a hypothesis and a proof that it is (approximately) optimal. In this section, we describe a multi-round protocol that demonstrates that interaction is helpful for verification.

The interactive protocol we present PAC verifies the class of *Fourier-sparse functions*. This is a broad class of functions, which includes decision trees, DNF formulas with small clauses, and AC^0 circuits.[9] Every function $f : \{0,1\}^n \to \mathbb{R}$ can be written as a linear combination $f = \sum_{T\subseteq[n]} \hat{f}(T)\chi_T$.[10] In Fourier-sparse functions, only a small number of coefficients are non-zero.

An important technicality is that throughout this section we focus solely on PAC verification with respect to families of distributions that have a uniform marginal over $\{0,1\}^n$, and have a target function $f : \{0,1\}^n \to \{1,-1\}$ such that $\mathbb{P}_{(x,y)\sim\mathcal{D}}[y = f(x)] = 1$. See further discussion in Remark 1.21 on page 14. One of the advantages of this setting is that in order to learn f, it is sufficient to approximate its heavy Fourier coefficients.

Notation 2.1. *Let $f : \{0,1\}^n \to \mathbb{R}$, and let $\tau \geq 0$. The set of τ-heavy coefficients of f is*

$$\hat{f}^{\geq\tau} = \{T \subseteq [n] : |\hat{f}(T)| \geq \tau\}.$$

Furthermore, approximating a single coefficient is easy given random samples from the uniform distribution. There are, however, an exponential number of coefficients, so approximating all of them is not feasible. This is where verification comes in. If the set of heavy coefficients is known, and if the function is Fourier-sparse, then one can efficiently learn the function by approximating that particular set of coefficients. The prover can provide the list of heavy coefficients, and then the

[9] See Mansour, 1994, Section 5.2.2, Theorems 5.15 and 5.16. (AC^0 is the set of functions computable by constant-depth boolean circuits with a polynomial number of AND, OR and NOT gates.)

[10] The real numbers $\hat{f}(T)$ are called *Fourier coefficients*, and the functions χ_T are called *characters*.

verifier can learn the function by approximating these coefficients.

The challenge that remains in designing such a verification protocol is to verify that the provided list of heavy coefficients is correct. If the list contains some characters that are not actually heavy, no harm is done.[11] However, if a dishonest prover omits some of the heavy coefficients from the list, how can the verifier detect this omission? The following result provides an answer to this question.

Lemma 2.2 (Interactive Goldreich-Levin). *There exists an interactive proof system (V, P^*) as follows. For every every $n \in \mathbb{N}$, $\delta > 0$, every $\tau \geq 2^{-\frac{n}{10}}$, every function $f : \{0,1\}^n \to \{0,1\}$, and every prover P, let*

$$L_P = [V(S, n, \tau, \delta, \rho_V), P^f(n, \tau, \delta, \rho_P)]$$

be a random variable denoting the output of V after interacting with the prover P, which has query access to f, where $S = \big((x_1, f(x_1)), \dots, (x_m, f(x_m))\big)$ is a random sample with $x_1, \dots, x_m$ taken independently and uniformly from $\{0,1\}^n$, and ρ_V, ρ_P are strings of private random coins. L_P takes values that are either a collection of subsets of $[n]$, or 'reject'*.*

The following properties hold:

- ***Completeness.*** $\mathbb{P}\left[L_{P^*} \neq \text{reject} \wedge \hat{f}^{\geq\tau} \subseteq L_{P^*}\right] \geq 1 - \delta$.
- ***Soundness.*** *For any (possibly unbounded) prover P,*
$$\mathbb{P}\left[L_P \neq \text{reject} \wedge \hat{f}^{\geq\tau} \nsubseteq L_P\right] \leq \delta.$$
- ***Double efficiency.*** *The verifier V uses at most $O\left(\frac{n}{\tau}\log\left(\frac{n}{\tau}\right)\log\left(\frac{1}{\delta}\right)\right)$ random samples from f and runs in time $\text{poly}\left(n, \frac{1}{\tau}, \log\left(\frac{1}{\delta}\right)\right)$. The runtime of the prover P^*, and the number of queries it makes to f, are at most $O\left(\frac{n^3}{\tau^5}\log\left(\frac{1}{\delta}\right)\right)$. Whenever $L_P \neq \text{reject}$, the cardinality of L_P is at most $O\left(\frac{n^2}{\tau^5}\log\left(\frac{1}{\delta}\right)\right)$.*

Remark 2.3. *In Definition 1.18 on page 14, we defined interactive proof systems specifically for PAC verification. The proof system in Lemma 2.2 is technically different, satisfying different completeness and soundness conditions. Additionally, in Definition 1.18 the verifier outputs a value that is either a function or* 'reject'*, while here the verifier outputs a value that is either a collection of subsets of $[n]$, or* 'reject'*.*

The verifier V operates by simulating the Goldreich-Levin (GL) algorithm for finding $\hat{f}^{\geq\tau}$. However, the GL algorithm requires query access to f, while V has access only to random samples. To overcome this limitation, V delegates the task of querying f to the prover P, who does have the necessary query access. Because P is not trusted, V engineers the set of queries it delegates to P in such a way that some random subset of them already appear in the sample S which V has received as input. This allows V to independently verify a random subset of the results sent by P, ensuring

[11] The verifier can approximate each coefficient in the list and discard of those that are not heavy. Alternatively, the verifier can include the additional coefficients in its approximation of the target function, because the approximation improves as the number of estimated coefficients grows (so long as the list is polynomial in n).

that a dishonest prover is discovered with high probability.

As a corollary of Lemma 2.2, we obtain the following lemma, which is an interactive version of the Kushilevitz-Mansour algorithm (Kushilevitz & Mansour, 1993; see also Linial, Mansour, & Nisan, 1993). It says that the class of t-sparse boolean functions is efficiently PAC verifiable with respect to the uniform distribution using an interactive proof system of the form $[V^{\mathcal{D}}, P^{f}]$, where the prover has query access and the verifier has random samples.

Notation 2.4. *Let $\mathcal{X}$ be a finite set. We write $\mathfrak{D}^{\text{func}}_{\mathcal{U}}(\mathcal{X})$ to denote the set of all distributions $\mathcal{D}$ over $\mathcal{X} \times \{1,-1\}$ that have the following two properties:*

- *The marginal distribution of $\mathcal{D}$ over $\mathcal{X}$ is uniform. Namely, $\sum_{y \in \{0,1\}} \mathcal{D}\Big((x,y)\Big) = \frac{1}{|\mathcal{X}|}$ for all $x \in \mathcal{X}$.*
- *$\mathcal{D}$ has a target function $f: \mathcal{X} \to \{1,-1\}$ satisfying $\mathbb{P}_{(x,y)\sim\mathcal{D}}[y = f(x)] = 1$.*

Lemma 2.5. *Let $\mathcal{X} = \{0,1\}^n$, and let $\mathcal{H}$ be the class of functions $\mathcal{X} \to \mathbb{R}$ that are t-sparse, as in Definition 1.15. The class $\mathcal{H}$ is 1-PAC verifiable for any $\varepsilon \geq 4t2^{-\frac{n}{10}}$ with respect to $\mathfrak{D}^{\text{func}}_{\mathcal{U}}(\mathcal{X})$ by a proof system in which the verifier has access to random samples from a distribution $\mathcal{D} \in \mathfrak{D}^{\text{func}}_{\mathcal{U}}(\mathcal{X})$, and the honest prover has oracle access to the target function $f: \mathcal{X} \to \{1,-1\}$ of $\mathcal{D}$. The running time of both parties is at most* $\text{poly}\left(n, t, \frac{1}{\varepsilon}, \log\left(\frac{1}{\delta}\right)\right)$. *The verifier in this protocol is not proper; the output is not necessarily t-sparse, but it is* $\text{poly}\left(n, t, \frac{1}{\varepsilon}, \log\left(\frac{1}{\delta}\right)\right)$*-sparse.*

2.1 The Interactive Goldreich-Levin Protocol

The verifier follows Protocol 1, which repeatedly applies Protocol 2 (IGL-ITERATION).

Protocol 1 Interactive Goldreich-Levin: $\text{IGL}(n, \tau, \delta)$

```
V performs the following:
  $r \leftarrow \left\lceil \left(\frac{4n}{\tau} + 1\right) \log\left(\frac{1}{\delta}\right) \right\rceil$
  for $i \in [r]$ do
      $L_i \leftarrow$ IGL-ITERATION$(n, \tau)$
      if $L_i =$ reject then
          output reject
  $L \leftarrow \bigcup_{i \in [r]} L_i$
  output $L$
```

Protocol 2 Interactive Goldreich-Levin Iteration: IGL-ITERATION(n,τ)

Assumption: V receives a sample $S = \Big((x_1, f(x_1)), \dots, (x_m, f(x_m))\Big)$ such that $m = \left\lceil \log\left(\frac{40n}{\tau^4}+1\right)\right\rceil$, for all $i \in [m]$, $x_i \in \{0,1\}^n$ is chosen independently and uniformly, and $f(x_i) \in \{0,1\}$.

1. V selects $i^* \in [n]$ uniformly at random, and then sends B to P, where
$$B = \{b_1, \dots, b_k\} \subseteq \{0,1\}^n$$
is a basis chosen uniformly at random from the set of bases of the subspace
$$H = \operatorname{span}(\{x_1 \oplus e_{i^*}, \dots, x_m \oplus e_{i^*}\}).$$
(For any j, e_j is a vector in which the j-th entry is 1 and all other entries are 0.)
2. P sends V the following set:
$$\{(x \oplus e_i, \tilde{f}(x \oplus e_i)) : \ i \in [n] \ \wedge \ x \in H\},$$
where for any z, $\tilde{f}(z)$ is purportedly the value of $f(z)$ obtained using P's query access to f.
3. V checks that for all $i \in [m]$, the evaluation $f(x_i)$ provided by V equals that which appeared in the sample S. If there are any discrepancies, V rejects and the interaction and terminates. Otherwise:
4. Let $\mathcal{K} = \{K : \ \varnothing \subsetneq K \subseteq [k]\}$. V Performs the following computation and outputs L:

```
$L \leftarrow \varnothing$
for $(y_1, \dots, y_k) \in \{0,1\}^k$ do
    for $K \in \mathcal{K}$ do
        $x^K \leftarrow \bigoplus_{i \in K} b_i$
        $y^K \leftarrow \bigoplus_{i \in K} y_i$
    for $i \in [n]$ do
        $a_i \leftarrow \text{majority}_{K \in \mathcal{K}} \left(\tilde{f}\left(x^K \oplus e_i\right) \oplus y^K\right)$
    add $\{i : \ a_i = 1\}$ and $\{i : \ a_i = 0\}$ to $L$
output $L$
```

We partition the proof of Lemma 2.2 into two claims. First, we show that if the prover is honest, then the output is correct.

Claim 2.6. *Consider an execution of* IGL-ITERATION(n,τ) *for* $\tau \geq 2^{-\frac{n}{10}}$. *For any prover P and any randomness ρ_P, if V did not reject, and the evaluations provided by P were mostly honest, in the sense that*
$$\forall i \in [n] : \ \mathbb{P}_{x \in H}\left[\tilde{f}(x \oplus e_i) \neq f(x \oplus e_i)\right] \leq \frac{\tau}{4},$$
then
$$\mathbb{P}\left[\hat{f}^{\geq \tau} \subseteq L\right] \geq \frac{1}{2},$$
where the probability is over the sample S and the randomness ρ_V.

Proof of Claim 2.6. Let E denote the event in which the samples $\{x_1,\dots,x_m\}$ are linearly independent. From Claim G.2, $\mathbb{P}[E] \geq \frac{3}{4}$. We will show that

$$\forall T \in \hat{f}^{\geq\tau} :\ \mathbb{P}[T \notin L \mid E] \leq \frac{\tau^2}{4}.$$

This is sufficient to prove the claim, because Parseval's identity entails that $|\hat{f}^{\geq\tau}| \leq \frac{1}{\tau^2}$, and so from the union bound and the law of total probability,

$$\begin{aligned}\mathbb{P}\left[\hat{f}^{\geq\tau} \nsubseteq L\right] &\leq \mathbb{P}\left[\hat{f}^{\geq\tau} \nsubseteq L \mid E\right] + \mathbb{P}[\neg E]\\ &\leq |\hat{f}^{\geq\tau}| \cdot \max_{T\in\hat{f}^{\geq\tau}} \mathbb{P}[T \notin L \mid E] + \mathbb{P}[\neg E]\\ &\leq \frac{1}{\tau^2}\cdot\frac{\tau^2}{4} + \frac{1}{4} = \frac{1}{2}.\end{aligned}$$

Fix some $T \in \hat{f}^{\geq\tau}$. Note that $T \in \hat{f}^{\geq\tau}$ entails that

$$\mathbb{P}_{x\in\{0,1\}^n}[f(x) = \ell(x)] \geq \frac{1}{2} + \frac{\tau}{2} \tag{3}$$

where $\ell(x)$ is either $\bigoplus_{i\in T} x_i$ or $1 \oplus (\bigoplus_{i\in T} x_i)$. Now, consider the iteration of the outer loop in Step 4 in which $y_j = \ell(b_j)$ for all $j \in [k]$. For any $i \in [n]$ and any $K \in \mathcal{K}$, let

$$G_{i,K} := \mathbb{1}\left(\tilde{f}\left(x^K \oplus e_i\right) = \ell(x^K \oplus e_i)\right),$$

and observe that if $G_{i,K} = 1$ then from linearity of ℓ,

$$\tilde{f}\left(x^K \oplus e_i\right) \oplus y^K = \ell(x^K \oplus e_i) \oplus \ell(x^K) = \ell(e_i) = \begin{cases} \mathbb{1}(i \in T) & \ell(x) = \bigoplus_{i\in T} x_i \\ 1 \oplus \mathbb{1}(i \in T) & \ell(x) = 1 \oplus (\bigoplus_{i\in T} x_i)\,. \end{cases}$$

Therefore, if

$$\forall i \in [n] :\ \frac{1}{|\mathcal{K}|}\sum_{K\in\mathcal{K}} G_{i,K} > \frac{1}{2}$$

then T will be added to L during the abovementioned iteration of the outer loop. Let

$$A_{i,K} := \mathbb{1}\left(f\left(x^K \oplus e_i\right) = \ell(x^K \oplus e_i)\right)$$

indicate cases where f agrees with ℓ, and let

$$D_{i,K} := \mathbb{1}\left(\tilde{f}\left(x^K \oplus e_i\right) \neq f(x^K \oplus e_i)\right)$$

indicates cases where P is dishonest about the value of f. Then for all $i \in [n]$,

$$\begin{aligned}
\frac{1}{|\mathcal{K}|}\sum_{K} G_{i,K} &\geq \frac{1}{|\mathcal{K}|}\left(\sum_{K} A_{i,K} - \sum_{K} D_{i,K}\right) \\
&\overset{(i)}{\geq} \frac{1}{|\mathcal{K}|}\sum_{K} A_{i,K} - \frac{\tau}{4} \\
&\overset{(ii)}{\geq} \frac{1}{|\mathcal{K}|}\sum_{K} A^*_{i,K} - \frac{\tau}{4},
\end{aligned}$$

where (i) follows from the assumption that P is dishonest about at most a $\frac{\tau}{4}$-fraction of the evaluations, and (ii) holds for

$$A^*_{i,K} = \begin{cases} A_{i,K} & x^K \oplus e_i \notin \{e_1, e_2, \dots, e_n\} \\ 0 & \text{otherwise.} \end{cases}$$

Therefore, we can bound the probability that $T \notin L$ based on how well f and ℓ agree:

$$\begin{aligned}
\mathbb{P}_{x_1,\dots,x_k}[T \notin L \mid E] &\leq \mathbb{P}\left[\exists i \in [n] :\ \frac{1}{|\mathcal{K}|}\sum_{K \in \mathcal{K}} A^*_{i,K} \leq \frac{1}{2} + \frac{\tau}{4} \;\middle|\; E\right] \\
&\leq \sum_{i=1}^{n} \mathbb{P}\left[\frac{1}{|\mathcal{K}|}\sum_{K} A^*_{i,K} \leq \frac{1}{2} + \frac{\tau}{4} \;\middle|\; E\right] && \text{(union bound)} \\
&\overset{(i)}{\leq} \sum_{i=1}^{n} \mathbb{P}\left[\left|\frac{1}{|\mathcal{K}|}\sum_{K} A^*_{i,K} - \mu\right| \geq \frac{\tau}{4} - \frac{n}{2^n} \;\middle|\; E\right] \\
&\leq \sum_{i=1}^{n} \mathbb{P}\left[\left|\frac{1}{|\mathcal{K}|}\sum_{K} A^*_{i,K} - \mu\right| \geq \frac{\tau}{5} \;\middle|\; E\right] && (\tau \geq 2^{-\frac{n}{10}}) \\
&\leq 25\sum_{i=1}^{n} \frac{\operatorname{Var}\left[\frac{1}{|\mathcal{K}|}\sum_{K} A^*_{i,K} \;\middle|\; E\right]}{\tau^2} && \text{(Chebyshev's inequality)} \\
&= 25\sum_{i=1}^{n} \frac{\operatorname{Var}\left[\sum_{K} A^*_{i,K} \mid E\right]}{|\mathcal{K}|^2\tau^2} \\
&\overset{(ii)}{\leq} 25\sum_{i=1}^{n} \frac{\sum_{K}\operatorname{Var}\left[A^*_{i,K} \mid E\right]}{|\mathcal{K}|^2\tau^2} \\
&\leq \frac{10n}{|\mathcal{K}|\tau^2} && \text{(variance of an indicator is } \leq \tfrac{1}{4}) \\
&= \frac{10n}{(2^k-1)\tau^2}.
\end{aligned}$$

Inequality (i) is justified because $\mu := \mathbb{E}\left[A^*_{i,K}\right] \geq \mathbb{E}\left[A_{i,K}\right] - \frac{n}{2^n} \geq \frac{1}{2} + \frac{\tau}{2} - \frac{n}{2^n}$, which follows from (3). For inequality (ii), we argue that given E, $\operatorname{Cov}(A^*_{i,K}, A^*_{i,K'}) \leq 0$ for any fixed $K, K' \in \mathcal{K}$, $K \neq K'$ and fixed $i \in [n]$. To see this, observe the following.

1. For any fixed sample $x_1,\dots,x_m \in \{0,1\}^n$, the pair $(x^K, x^{K'})$ is distributed uniformly over the set $\{(u,u') :\ u,u' \in H\setminus\{0\}\ \wedge\ u \neq u'\}$. This is true because the base $\{b_1,\dots,b_k\}$ is chosen uniformly from all bases of H, implying that $x^K = \bigoplus_{i\in K} b_i$ is a uniform point in $H\setminus\{0\}$. Furthermore, for any fixed value of x^K, $u_{\text{diff}} := x^K \oplus x^{K'} = \bigoplus_{i\in K\Delta K'} b_i$ is a uniform point in $H\setminus\{0,x^K\}$. Hence for any fixed value of x^K, the point $x^{K'} = x^K \oplus u_{\text{diff}}$ is uniform in $H\setminus\{0,x^K\}$.
2. If $x_1,\dots,x_m \in \{0,1\}^n$ are sampled independently and uniformly and we assume E occurs, then H is a random subspace of dimension m within $\{0,1\}^n$. Therefore, the pair $(x^K, x^{K'})$ is distributed uniformly over the set $\{(u,u') :\ u,u' \in \{0,1\}^n\setminus\{0\}\ \wedge\ u \neq u'\}$.
3. Hence, the pair $(x^K \oplus e_i,\ x^{K'} \oplus e_i)$ is distributed uniformly over the set
$$W = \{(u,u') :\ u,u' \in U\ \wedge\ u \neq u'\},$$
where $U = \{0,1\}^n \setminus \{e_i\}$.
4. Denote $A^* = \{x \in \{0,1\}^n :\ f(x) = \ell(x)\} \setminus \{e_1, e_2, \dots, e_n\}$. Then
$$\begin{aligned}\mathrm{Cov}(A^*_{i,K}, A^*_{i,K'}) &= \mathbb{E}\left[A^*_{i,K}A^*_{i,K'}\right] - \mathbb{E}\left[A^*_{i,K}\right]\mathbb{E}\left[A^*_{i,K'}\right] \\ &= \mathbb{P}_{(x,y)\in W}\left[x \in A^*\right]\left(\mathbb{P}_{(x,y)\in W}\left[y \in A^* \mid x \in A^*\right] - \mathbb{P}_{(x,y)\in W}\left[x \in A^*\right]\right) \\ &\leq \mathbb{P}_{(x,y)\in W}\left[y \in A^* \mid x \in A^*\right] - \mathbb{P}_{(x,y)\in W}\left[x \in A^*\right] \\ &= \frac{|A^*|-1}{|U|-1} - \frac{|A^*|}{|U|} < 0.\end{aligned}$$

Finally, note that when E occurs (the samples $\{x_1,\dots,x_m\}$ are linearly independent) then
$$k = m \geq \log\left(\frac{40n}{\tau^4} + 1\right),$$
and so
$$\mathbb{P}_{x_1,\dots,x_k}\left[T \notin L \mid E\right] \leq \frac{10n}{(2^k-1)\tau^2} \leq \frac{\tau^2}{4},$$
as desired. ■

Next, we show that if the prover is dishonest, it will be rejected.

Claim 2.7. *Consider an execution of* IGL-ITERATION(n,τ). *For any prover P and any randomness value ρ_P, if there exists $i \in [n]$ for which P was too dishonest in the sense that*
$$\mathbb{P}_{x\in H}\left[\tilde{f}(x\oplus e_i) \neq f(x\oplus e_i)\right] > \frac{\tau}{4},$$
then
$$\mathbb{P}\left[L = \text{reject}\right] \geq \frac{\tau}{4n},$$
where the probability is over the sample S and the randomness ρ_V.

Proof. Let E denote the event in which the index i^* selected by V is one for which P is too dishonest. We now focus on the case where this event occurred. Let $H^* = H \oplus e_{i^*}$, and let $X \subseteq H^*$ denote the subset of H^* that appeared in the sample S received by V. Observe that

$$1 - \frac{\tau}{4} > \mathbb{E}_{x \in H^*}\left[\mathbb{1}(\tilde{f}(x) = f(x)) \mid E\right] = \mathbb{E}_X\left[\mathbb{E}_{x \in X}\left[\mathbb{1}(\tilde{f}(x) = f(x))\right] \mid E\right] = \mathbb{E}_X\left[h_X \mid E\right],$$

where $h_X := \mathbb{E}_{x \in X}\left[\mathbb{1}(\tilde{f}(x) = f(x))\right]$, is the fraction of the sample on which P was honest. Notice that the only assumptions we have made about the distribution of X is that for every $x, x' \in H^*$, $\mathbb{P}[x \in X] = \mathbb{P}[x' \in X]$.

From Markov's inequality,

$$\mathbb{P}_X\left[h_X = 1 \mid E\right] \leq \mathbb{P}_X\left[h_X \geq 1 \mid E\right] \leq \mathbb{E}_X\left[h_X \mid E\right] < 1 - \frac{\tau}{4}.$$

This means that

$$\mathbb{P}\left[L = \text{reject} \mid E\right] = \mathbb{P}\left[\exists x \in X : \ \tilde{f}(x) \neq f(x) \mid E\right] \geq \frac{\tau}{4},$$

and we conclude that

$$\mathbb{P}\left[L = \text{reject}\right] \geq \mathbb{P}\left[L = \text{reject} \mid E\right]\mathbb{P}\left[E\right] \geq \frac{\tau}{4} \cdot \frac{1}{n}. \ \blacksquare$$

We now prove Lemma 2.2 using Claims 2.6 and 2.7.

Proof of Lemma 2.2. We show that the protocol $\mathrm{IGL}(n, \tau, \delta)$ satisfies the requirements of Lemma 2.2. For the completeness, consider the deterministic prover P^* that simply uses its query access to f in order to send the set

$$\{(x \oplus e_i, f(x \oplus e_i)) : \ i \in [n] \ \wedge \ x \in H\},$$

to V, and observe that P^* will never be rejected. Furthermore, for every $i \in [r]$, Claim 2.6 entails that $\mathbb{P}\left[\hat{f}^{\geq \tau} \not\subseteq L_i\right] \leq \frac{1}{2}$. Thus, $\hat{f}^{\geq \tau} \subseteq L_{P^*} \geq 1 - 2^{-r} \geq 1 - \delta$, as desired.

For the soundness, assume for contradiction that there exists some malicious prover $\tilde{P}$ such that

$$\mathbb{P}\left[L_{\tilde{P}} \neq \text{reject} \ \wedge \ \hat{f}^{\geq \tau} \not\subseteq L_{\tilde{P}}\right] > \delta.$$

The IGL protocol consists of r executions of IGL-ITERATION. We say that $\tilde{P}$ was *sufficiently honest* in a particular execution of IGL-ITERATION if in that execution,

$$\forall i \in [n] : \ \mathbb{P}_{x \in H}\left[\tilde{f}(x \oplus e_i) \neq f(x \oplus e_i)\right] \leq \frac{\tau}{4}.$$

Let D be an indicator denoting the event that throughout the r executions, $\tilde{P}$ was too dishonest, meaning that the number of executions in which $\tilde{P}$ was sufficiently honest is strictly less than $\log(\frac{1}{\delta})$.

Consider the following two case:

- The dishonest case ($D = 1$): There were at least $r' := r - \log(\frac{1}{\delta}) \geq \frac{4n}{\tau} \log\left(\frac{1}{\delta}\right)$ executions in which $\tilde{P}$ was not sufficiently honest. From Claim 2.7, the probability of rejection in each of these r' repetitions is at least $\frac{\tau}{4n}$. Hence, because the rounds are independent,

$$\mathbb{P}\left[L_{\tilde{P}} \neq \text{reject} \mid D = 1\right] \leq \left(1 - \frac{\tau}{4n}\right)^{r'} \leq \left(1 - \frac{\tau}{4n}\right)^{\frac{4n}{\tau}\log\left(\frac{1}{\delta}\right)} \leq e^{-\log\left(\frac{1}{\delta}\right)} \leq \delta.$$

- The honest case ($D=0$): Let $j_1,\dots,j_{r'} \in [r]$ be the rounds in which $\tilde{P}$ was sufficiently honest, with $r' \geq \log(\frac{1}{\delta})$. From Claim 2.6, with probability at least $1-\delta$, the result L_{j_t} for each $t \in [r']$ satisfies
$$\mathbb{P}\left[\hat{f}^{\geq\tau} \subseteq L_{j_t}\right] \geq \frac{1}{2}.$$
Hence, because the rounds are independent,
$$\mathbb{P}\left[\hat{f}^{\geq\tau} \nsubseteq L_{\tilde{P}} \mid D=0\right] \leq 2^{-r'} \leq \delta.$$

Putting the two cases together, we obtain the desired contradiction:

$$\begin{aligned}\mathbb{P}\left[L_{\tilde{P}} \neq \text{reject} \wedge \hat{f}^{\geq\tau} \nsubseteq L_{\tilde{P}}\right] &= \mathbb{P}\left[L_{\tilde{P}} \neq \text{reject} \wedge \hat{f}^{\geq\tau} \nsubseteq L_{\tilde{P}} \mid D=0\right]\mathbb{P}[D=0]+ \\ &\quad\ \mathbb{P}\left[L_{\tilde{P}} \neq \text{reject} \wedge \hat{f}^{\geq\tau} \nsubseteq L_{\tilde{P}} \mid D=1\right]\mathbb{P}[D=1] \\ &\leq \mathbb{P}\left[\hat{f}^{\geq\tau} \nsubseteq L_{\tilde{P}} \mid D=0\right]\mathbb{P}[D=0]+ \\ &\quad\ \mathbb{P}[L_{\tilde{P}} \neq \text{reject} \mid D=1]\,\mathbb{P}[D=1] \\ &\leq \delta.\end{aligned}$$

This completes the proof of the soundness property. For the efficiency, observe the following:

- V performs $r = \left\lceil (\frac{4n}{\tau}+1)\log\left(\frac{1}{\delta}\right)\right\rceil$ repetitions of the IGL-Iteration protocol, and each repetition requires $m = \left\lceil \log\left(\frac{40n}{\tau^4}+1\right)\right\rceil$ fresh samples. Thus, V requires a total of
$$r\cdot m = O\left(\frac{n}{\tau}\log\left(\frac{n}{\tau}\right)\log\left(\frac{1}{\delta}\right)\right).$$
random samples from f.
- P^* also performs r repetitions of the IGL-Iteration protocol, and makes at most $n2^m$ queries to f in each repetition. Thus, P^* uses at most
$$q = r\cdot n2^m = O\left(\frac{n^3}{\tau^5}\log\left(\frac{1}{\delta}\right)\right)$$
queries to f.
- P^* runs in time $O(q)$, and V runs in time polynomial in q.
- For the bound on the cardinality of L_P, observe that V performs r repetitions of IGL-Iteration, and in each repetition, the number of items added to the list in Step 4 is at most $2^k \leq 2^m$. Thus, the total list length is at most
$$r\cdot 2^m = O\left(\frac{n^2}{\tau^5}\log\left(\frac{1}{\delta}\right)\right).$$

This completes the proof. ■

Remark 2.8. *It is possible to run all repetitions of the IGL protocol in parallel such that only* 2 *messages are exchanged.*

2.2 Efficient Verification of Fourier-Sparse Functions

The verification protocol of Lemma 2.5 is described in Protocol 3. In the IGL protocol, we worked with functions $f : \{0,1\}^n \to \{0,1\}$. Now, we move to working with functions $f : \{0,1\}^n \to \{1,-1\}$. We translate data from $\{1,-1\}$ to $\{0,1\}$ as follows: $b \in \{1,-1\}$ is mapped to $\frac{1-b}{2} \in \{0,1\}$, and $b \in \{0,1\}$ is mapped to $(-1)^b \in \{1,-1\}$.

Protocol 3 PAC Verification of t-Sparse Functions: $\textsc{VerifyFourierSparse}(n,t,\varepsilon,\delta)$

```
V performs the following:
  $\tau \leftarrow \frac{\varepsilon}{4t}$
  $L \leftarrow \mathrm{IGL}(n, \tau, \frac{\delta}{2})$
  if $L =$ reject then
      output reject
  else
      $\lambda \leftarrow \sqrt{\frac{\varepsilon}{8|L|}}$
      for $T \in L$ do
          $\alpha_T \leftarrow \textsc{EstimateCoefficient}(T, \lambda, \frac{\delta}{2|L|})$
      $h \leftarrow \sum_{T \in L} \alpha_T \chi_T$
      output $h$
```

Remark 2.9. *The output of* $\textsc{VerifyFourierSparse}$ *is a function $h : \{0,1\}^n \to \mathbb{R}$, not necessarily a boolean function.*

Algorithm 4 Estimating a Fourier Coefficient: $\textsc{EstimateCoefficient}(T,\lambda,\delta)$

```
  $m \leftarrow \left\lceil \frac{2\ln(2/\delta)}{\lambda^2} \right\rceil$
  for $i \in [m]$ do
      sample $(x_i, y_i) \leftarrow \mathcal{D}$                ▷ Takes i.i.d. samples from $\mathcal{D}$.
  $\alpha_T \leftarrow \sum_{i=1}^{m} y_i \chi_T(x_i)$
  output $\alpha_T$
```

2.2.1 Proof

Lemma 2.5 follows from Lemma 2.2 via standard techniques (see exposition in Mansour, 1994). The proof is provided below for completeness. We start with the following claim.

Claim 2.10. *Let $\lambda, \delta > 0$, $T \subseteq [n]$, and let $\mathcal{D} \in \mathfrak{D}_{\mathcal{U}}^{\mathrm{func}}(\{0,1\}^n)$ with target function $f : \{0,1\}^n \to \{1,-1\}$. Then* $\textsc{EstimateCoefficient}(T,\lambda,\delta)$ *uses $m = \left\lceil \frac{2\ln(2/\delta)}{\lambda^2} \right\rceil$ random* *samples from $\mathcal{D}$ and outputs a number α_T such that*

$$\mathbb{P}\left[|\alpha_T - \hat{f}(T)| \geq \lambda\right] \leq \delta,$$

where the probability is over the samples.

Proof. Let $\Big((x_1, f(x_1), \ldots, (x_m, f(x_m))\Big)$ denote the sample. Recall that

$$\hat{f}(T) = \langle f, \chi_T \rangle := \mathbb{E}_{x \in \{0,1\}^n}\left[f(x)\chi_T(x)\right],$$

where $|f(x)\chi_T(x)| \leq 1$. Therefore, if we take

$$\alpha_T := \sum_{i=1}^{m} f(x_i)\chi_T(x_i)$$

then Hoeffding's inequality yields

$$\mathbb{P}\left[\left|\alpha_T - \hat{f}(T)\right| \geq \lambda\right] \leq 2\exp(-m\lambda^2/2) \leq \delta. \blacksquare$$

Proof of Lemma 2.5. Fix $\varepsilon, \delta > 0$ and a distribution $\mathcal{D} \in \mathfrak{D}_{\mathcal{U}}^{\text{func}}(\{0,1\}^n)$ with target function $f: \{0,1\}^n \to \{1,-1\}$. Consider an execution of VERIFYFOURIERSPARSE(n,t,ε,δ). We show completeness, soundness, double efficiency and sparsity.

- **Completeness.** Assume that the prover P was honest. Then from Lemma 2.2, with probability at least $1-\frac{\delta}{2}$, $L \neq$ reject and $\hat{f}^{\geq\tau} \subseteq L$. Additionally, from Claim 2.10, with probability at least $1-\frac{\delta}{2}$ it holds that
$$\forall T \in L: \left|\alpha_T - \hat{f}(T)\right| \leq \lambda.$$
Hence, from the union bound, with probability at least $1-\delta$ all the assumptions of Claim F.1 hold, in which case Claim F.1 guarantees that $L_{\mathcal{D}}(h) \leq L_{\mathcal{D}}(\mathcal{H}) + \varepsilon$, as desired.
- **Soundness.** Assume for contradiction that there exists some (possibly unbounded) prover P such that the verifier's output h satisfies
$$\mathbb{P}\Big[h \neq \text{reject} \wedge \Big(L_{\mathcal{D}}(h) > L_{\mathcal{D}}(\mathcal{H}) + \varepsilon\Big)\Big] > \delta. \tag{4}$$
From the soundness property of the IGL protocol (Lemma 2.2),
$$\mathbb{P}\left[h \neq \text{reject} \wedge \hat{f}^{\geq\tau} \nsubseteq L\right] \leq \frac{\delta}{2}. \tag{5}$$
Likewise, from Claim 2.10 and the union bound,
$$\mathbb{P}\left[h \neq \text{reject} \wedge \exists T \in L: \left|\alpha_T - \hat{f}(T)\right| > \lambda\right] \leq \frac{\delta}{2}. \tag{6}$$
From Equations (4), (5) and (6), we obtain that
$$\mathbb{P}\Big[h \neq \text{reject} \wedge \Big(L_{\mathcal{D}}(h) > L_{\mathcal{D}}(\mathcal{H}) + \varepsilon\Big) \wedge G\Big] > 0. \tag{7}$$
where G denotes the event in which $\hat{f}^{\geq\tau} \subseteq L \wedge \forall T \in L: \left|\alpha_T - \hat{f}(T)\right| \leq \lambda$. Claim F.1 asserts that
$$G \implies L_{\mathcal{D}}\Big(\sum_{T \in L} \alpha_T \chi_T(x)\Big) \leq L_{\mathcal{D}}(\mathcal{H}) + \varepsilon. \tag{8}$$

Note that if $h \neq$ ‘reject’ then $h = \sum_{T \in L} \alpha_T \chi_T(x)$. Hence, putting together Equations (7) and (8), we conclude that

$$\mathbb{P}\left[h \neq \text{reject} \wedge \left(L_{\mathcal{D}}(h) > L_{\mathcal{D}}(\mathcal{H}) + \varepsilon\right) \wedge \left(L_{\mathcal{D}}(h) \leq L_{\mathcal{D}}(\mathcal{H}) + \varepsilon\right)\right] > 0,$$

which is a contradiction.

- **Double efficiency.** From Lemma 2.2, V uses at most

$$O\left(\frac{n}{\tau}\log\left(\frac{n}{\tau}\right)\log\left(\frac{1}{\delta}\right)\right) = O\left(\frac{nt}{\varepsilon}\log\left(\frac{nt}{\varepsilon}\right)\log\left(\frac{1}{\delta}\right)\right)$$

samples for the IGL protocol, which produces a set L of coefficients such that

$$|L| = O\left(\frac{n^2}{\tau^5}\log\left(\frac{1}{\delta}\right)\right) = O\left(\frac{n^2 t^5}{\varepsilon^5}\log\left(\frac{1}{\delta}\right)\right).$$

Then, it uses

$$\left\lceil \frac{2\ln(2/\delta)}{\lambda^2} \right\rceil = O\left(\frac{\log(1/\delta)|L|}{\varepsilon}\right)$$

samples for estimating each of the coefficients. In total, V uses at most

$$O\left(\frac{nt}{\varepsilon}\log\left(\frac{nt}{\varepsilon}\right)\log\left(\frac{1}{\delta}\right)\right) + |L| \cdot O\left(\frac{2\log(1/\delta)|L|}{\varepsilon}\right) = \text{poly}\left(n, t, \frac{1}{\varepsilon}, \log\left(\frac{1}{\delta}\right)\right)$$

random samples.

Also from Lemma 2.2, when executing the IGL protocol, the honest prover makes at most

$$O\left(\frac{n^3}{\tau^5}\log\left(\frac{1}{\delta}\right)\right) = O\left(\frac{n^3 t^5}{\varepsilon^5}\log\left(\frac{1}{\delta}\right)\right) = \text{poly}\left(n, t, \frac{1}{\varepsilon}, \log\left(\frac{1}{\delta}\right)\right)$$

queries.

Clearly, both parties run in time polynomial in the number of their samples or queries.

- **Sparsity.** The output $h = \sum_{T \in L} \alpha_T \chi_T$ is $|L|$-sparse, where

$$|L| = O\left(\frac{n^2}{\tau^5}\log\left(\frac{1}{\delta}\right)\right) = O\left(\frac{n^2 t^5}{\varepsilon^5}\log\left(\frac{1}{\delta}\right)\right). \blacksquare$$

3 Separation Between Learning, Testing, and PAC Verification

In this section we demonstrate a gap in sample complexity between *learning* and *verification*. Conceptually, the result tells us that at least in some scenarios, delegating a learning task to an untrusted party is worthwhile, because verifying that their final result is correct is significantly cheaper than finding that result ourselves.

Recall from the discussion in Section 1.1 that when an untrusted prover provides a hypothesis $\tilde{h}$ which is allegedly ε-good, the straightforward approach for the verifier is to approximate each of the terms $L_{\mathcal{D}}(\tilde{h})$ and $L_{\mathcal{D}}(\mathcal{H})$, and then determine whether the inequality $L_{\mathcal{D}}(\tilde{h}) \leq L_{\mathcal{D}}(\mathcal{H}) + \varepsilon$ holds. From Hoeffding’s inequality, the term $L_{\mathcal{D}}(\tilde{h})$ can easily be approximated with constant confidence up to any $O(\varepsilon)$ additive error using only $O(\frac{1}{\varepsilon^2})$ samples. However, approximating the term $L_{\mathcal{D}}(\mathcal{H})$ is more challenging, because it involves the loss values of all the hypotheses in the class $\mathcal{H}$.

In this section we show an MA-like proof system wherein the prover sends a single message $(\tilde{h}, \tilde{C}, \tilde{\ell})$ such that allegedly $\tilde{h}$ is an ε-good hypothesis with loss at most $\tilde{\ell} > 0$, and $\tilde{C} \in \{0,1\}^*$ is a string called a *certificate of loss*. The verifier operate as follows:[12]

- Verify that $L_{\mathcal{D}}(\tilde{h}) \leq \tilde{\ell}$ with high probability. That is, estimate the loss of $\tilde{h}$ with respect to $\mathcal{D}$, and check that with high probability it is at most $\tilde{\ell}$.
- Use the certificate of loss $\tilde{C}$ to verify that with high probability, $L_{\mathcal{D}}(\mathcal{H}) \geq \tilde{\ell} - \varepsilon$. This step is called *verifying the certificate*.

That is, a certificate of loss is a string that helps the verifier ascertain that $\mathcal{H}$ has a large loss with respect to the unknown distribution $\mathcal{D}$. Whenever one defines algorithms for generating and verifying certificates of loss for a class $\mathcal{H}$, that also defines an associated single-message interactive proof system for PAC verifying $\mathcal{H}$.

3.1 Warm-Up: The Class of Thresholds

For clarity of exposition, we start with a warm-up that investigates the class $\mathcal{T}$ of threshold functions (see definition below). This class admits certificates that are easy to explain and visualize. We will show that the certificates of loss for $\mathcal{T}$ induce a proof system for PAC verifying $\mathcal{T}$ that is complete, sounds, and doubly efficient. However, verifying certificates for $\mathcal{T}$ requires as much resources as PAC learning $\mathcal{T}$ without the help of a prover, and so using this proof system to delegate learning of $\mathcal{T}$ is not worthwhile. Therefore, the next step (in Section 3.2 below) will show that $\mathcal{T}$ and its certification easily generalize to the class $\mathcal{T}_d$ of multi-thresholds. The gap between verifying and learning is demonstrated for $\mathcal{T}_d$.

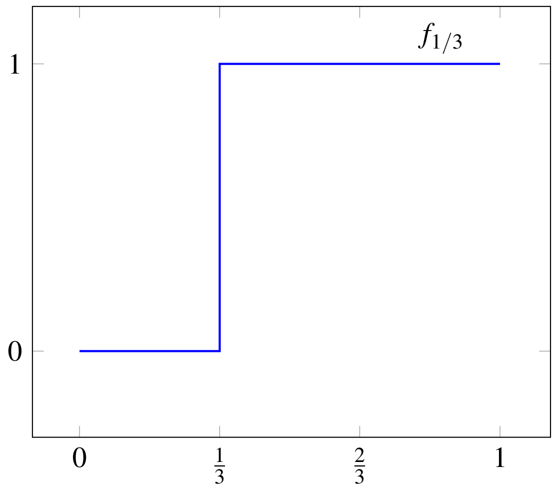

Figure 2: The function $f_{1/3} \in \mathcal{T}$.

[12] We provide a more detailed description of the verification procedure in Claim 3.14 below.

Definition 3.1. *The class $\mathcal{T}$ is the set of all monotone increasing boolean functions on $[0,1]$, as follows:*

$$\mathcal{T} = \{f_t : t \in [0,1]\},$$

where for any $t \in [0,1]$, the function $f_t : [0,1] \to \{0,1\}$ is given by

$$f_t(x) = \begin{cases} 0 & x < t \\ 1 & x \geq t. \end{cases}$$

Figure 2 illustrates an example of a function in $\mathcal{T}$.

Remark 3.2. *For convenience, we present the separation result with respect to thresholds defined over a continuous interval $\mathcal{X} \subseteq \mathbb{R}$. Furthermore, we assume that the marginal distribution on $\mathcal{X}$ is absolutely continuous with respect to the Lebesgue measure, and we also ignore issues relating to the representation of real numbers in computations and protocol messages. This provides for a smooth exposition of the ideas. In Appendix B, we show how the results can be discretized.*

3.1.1 Existence of Certificates of Loss for Thresholds

We want to design certificates such that for every distribution $\mathcal{D} \in \Delta([0,1] \times \{0,1\})$ the class $\mathcal{T}$ has large loss, $L_{\mathcal{D}}(\mathcal{T}) \geq \ell$, if and only if there exists a certificate for that fact.

The idea is straightforward. Consider two sets $A \subseteq [0,1] \times \{1\}$ and $B \subseteq [0,1] \times \{0\}$, such that all the points in A are located to the left of all the points in B, as in Figure 3. Because we only allow thresholds that are monotone increasing, a threshold that labels any point in A correctly

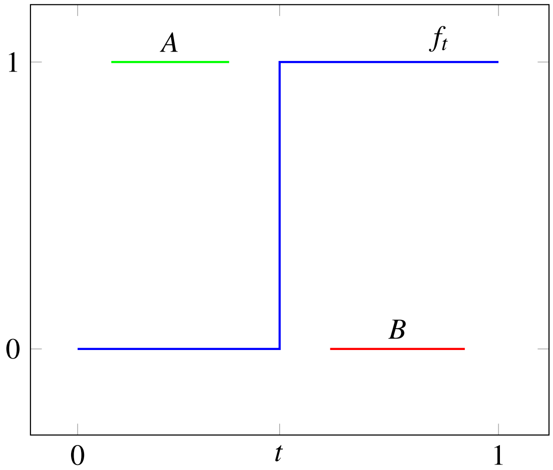

Figure 3: Structure of a simple certificate of loss for monotone increasing thresholds. The set A is labeled with 1, and B is labeled 0. The depicted threshold f_t happens to misclassify both A and B, but it is just one possible threshold.

must label all points of B incorrectly, and vice versa. Hence, any threshold must have loss at least $\min\{\mathcal{D}(A), \mathcal{D}(B)\}$. Formally:

Definition 3.3. *Let $\mathcal{D} \in \Delta([0,1] \times \{0,1\})$ be a distribution and $\ell, \eta \geq 0$. A certificate of loss at least ℓ for class $\mathcal{T}$ is a pair (a,b) where $0 < a \leq b < 1$.*

We say that the certificate is η-valid with respect to distribution $\mathcal{D}$ if the events

$$\begin{aligned} A &= [0,a) \times \{1\} \\ B &= [b,1] \times \{0\} \end{aligned} \tag{9}$$

satisfy

$$|\mathcal{D}(A) - \ell| + |\mathcal{D}(B) - \ell| \leq \eta. \tag{10}$$

The following claim shows the soundness of the certificate, i.e., that a valid certificate of loss does indeed entail that $L_{\mathcal{D}}(\mathcal{T})$ is large.

Claim 3.4. *Let $\mathcal{D} \in \Delta([0,1] \times \{0,1\})$ be a distribution and $\ell, \eta \geq 0$. If $\mathcal{D}$ has a certificate of loss at least ℓ which is η-valid with respect to $\mathcal{D}$, then $L_{\mathcal{D}}(\mathcal{T}) \geq \ell - \eta$.*

Proof. Assume $C = (a,b)$ is an η-valid certificate of loss at least ℓ for $\mathcal{T}$ with respect to $\mathcal{D}$. For any $t \in [0,1]$, we show that $L_{\mathcal{D}}(f_t) \geq \ell - \eta$.

Consider two cases:

- Case 1: $t < a$. Then for any $x \geq a$, $f_t(x) = 1$. In particular, taking B as in (9), we obtain that
$$\forall (x,y) \in B: \; f_t(x) \neq y.$$
Observe from Equation (10) that $\mathcal{D}(B) \geq \ell - \eta$. Therefore,
$$L_{\mathcal{D}}(f_t) = \mathbb{P}_{(x,y) \in \mathcal{D}}[f_t(x) \neq y] \geq \mathcal{D}(B) \geq \ell - \eta.$$
- Case 2: $t \geq a$. This case is symmetric to the previous one, replacing B with $A = [0,a) \times \{1\}$.

Next, we show completeness, meaning that whenever $L_{\mathcal{D}}(\mathcal{T})$ is large there exists a certificate to that effect. However, the certificate is not tight, conceding a factor of 2:

Claim 3.5. *Let $\mathcal{D} \in \Delta([0,1] \times \{0,1\})$ be a distribution and $\ell \geq 0$. If $L_{\mathcal{D}}(\mathcal{T}) = \ell$ then there exists a 0-valid certificate of loss at least $\frac{\ell}{2}$ with respect to $\mathcal{D}$.*

Proof of Claim 3.5. Let f_t be an optimal threshold for $\mathcal{D}$, that is, $L_{\mathcal{D}}(f_t) = \ell$.[13] Let

$$\tilde{A} = [0,t) \times \{1\}$$

$$\tilde{B} = [t,1] \times \{0\}$$

denote the two events in which f_t misclassifies a point.[14] It follows that

$$\ell = \mathcal{D}(\tilde{A}) + \mathcal{D}(\tilde{B}).$$

[13] Note that an optimal threshold $t \in [0,1]$ exists because $[0,1]$ is compact, and the mapping $t \mapsto L_{\mathcal{D}}(f_t)$ is continuous.

[14] Namely, $\tilde{A}$ is the event in which a point has label 1, but f_t assigns label 0 to it, and $\tilde{B}$ is the event in which a point has label 0, but f_t assigns label 1 to it.

If $\mathcal{D}(\tilde{A}) = \mathcal{D}(\tilde{B}) = \frac{\ell}{2}$, then (t,t) is the desired certificate. Otherwise, assume w.l.o.g. that

$$\mathcal{D}(\tilde{A}) > \frac{\ell}{2} > \mathcal{D}(\tilde{B}).$$

Because the marginal distribution of $\mathcal{D}$ on $[0,1]$ is absolutely continuous, there exists a point $a \in [0,t)$ that partitions the event $\tilde{A}$ to

$$A := [0,a) \times \{1\},$$

$$A' := [a,t) \times \{1\},$$

such that $\mathcal{D}(A) = \frac{\ell}{2}$. Considering the event $B' := [a,t) \times \{0\}$. The optimality of f_t implies that

$$\mathcal{D}(B') \geq \mathcal{D}(A')$$

because otherwise the threshold f_a would have loss strictly smaller than that of f_t.

Notice that

$$\mathcal{D}(B') \geq \mathcal{D}(A') = \mathcal{D}(\tilde{A}) - \mathcal{D}(A) = \left(\ell - \mathcal{D}(\tilde{B})\right) - \mathcal{D}(A) = \ell - \mathcal{D}(\tilde{B}) - \frac{\ell}{2} = \frac{\ell}{2} - \mathcal{D}(\tilde{B}).$$

Hence, again invoking absolute continuity of measure as above, there exists a point $b \in [a,t)$ such that

$$\mathcal{D}([b,t) \times \{0\}) = \frac{\ell}{2} - \mathcal{D}(\tilde{B}).$$

Therefore, taking

$$B := [b,1) \times \{0\}$$

yields

$$\mathcal{D}(B) = \mathcal{D}([b,t) \times \{0\}) + \mathcal{D}(\tilde{B}) = \frac{\ell}{2}.$$

So (a,b) is the desired certificate. ■

3.1.2 Efficient Generation and Verification of Certificates for Thresholds

The following two claims show that certificates of loss for $\mathcal{T}$ do not merely exist, but they can be generated and verified efficiently, making delegation feasible.

Claim 3.6 (Efficient Verification). *Let $\mathcal{D} \in \Delta([0,1] \times \{0,1\})$ be a distribution and $\ell, \delta, \eta \geq 0$. There exists an algorithm that, upon receiving input (a,b) such that $0 < a \leq b < 1$, takes $O\left(\frac{\log\left(\frac{1}{\delta}\right)}{\eta^2}\right)$ i.i.d. samples from $\mathcal{D}$ and satisfies the following:*

- *Completeness. If (a,b) is an η-valid certificate of loss at least ℓ with respect to $\mathcal{D}$, then the algorithm accepts with probability at least $1-\delta$.*
- *Soundness. If (a,b) is not a 2η-valid certificate of loss at least ℓ with respect to $\mathcal{D}$, then the algorithm rejects with probability at least $1-\delta$.*

Furthermore, the algorithm runs in time polynomial[15] *in the number of samples.*

Proof. Let A, B be as in Equation (9), and let $(x_1, y_1), \ldots, (x_m, y_m)$ be the samples the algorithm received. The algorithm calculates the empirical measures of A, B by

$$\hat{\ell}_A := \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}\big((x_i, y_i) \in A\big)$$

$$\hat{\ell}_B := \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}\big((x_i, y_i) \in B\big)$$

and accepts if and only if

$$|\hat{\ell}_A - \ell| + |\hat{\ell}_B - \ell| < \frac{3}{2}\eta.$$

The running time is clear, and correctness follows from Hoeffding's inequality,

$$\mathbb{P}\left[\left|\hat{\ell}_A - \mathcal{D}(A)\right| \geq \frac{\eta}{4}\right] \leq 2\exp\left(-2m\left(\frac{\eta}{4}\right)^2\right).$$

Requiring that this probability be strictly less than $\frac{\delta}{2}$ yields the bound

$$m > \frac{2\log\frac{16}{\delta}}{\eta^2}.$$

The same holds for $\hat{\ell}_B$. The union bound entails that with probability at least $1 - \delta$ both estimates are $\frac{\eta}{4}$-close to their expectations, in which case the algorithm decides correctly. ■

Claim 3.7 (Efficient Generation). *There exists an algorithm as follows. For any distribution $\mathcal{D} \in \Delta([0,1] \times \{0,1\})$ and any $\delta, \eta \in (0, \frac{1}{2})$, the algorithm outputs a certificate $(\hat{a}, \hat{b})$ for $\mathcal{T}$ that with probability at least $1 - \delta$ is an η-valid certificate of loss at least $\ell = L_{\mathcal{D}}(\mathcal{T})/2$ with respect to $\mathcal{D}$. The algorithm uses*

$$O\left(\frac{1}{\eta^2}\log\frac{1}{\eta} + \frac{1}{\eta^2}\log\frac{1}{\delta}\right)$$

i.i.d. samples from $\mathcal{D}$ and runs in time polynomial in the number of samples.

Proof. The proof is a standard application of uniform convergence, VC dimension and empirical risk minimization (ERM), as covered e.g. in Shalev-Shwartz and Ben-David (2014). For completeness, we provide a self-contained proof that depends only on Theorem C.3, which upper bounds the number of samples necessary to obtain an *ε-sample* for a set system of finite VC dimension (see definitions in Appendix C).

We start by stating the following consequence of Theorem C.3. Let $S = ((x_1, y_1), \ldots, (x_m, y_m))$ denote the samples that the algorithm receives, and let $\mathcal{I}$ denote the following set of intervals:

$$\mathcal{I} = \{[u, v) : u, v \in \mathbb{R}\} \cup \{[u, v] : u, v \in \mathbb{R}\}.$$

[15] Recall that we ignore the cost performing calculations with real numbers.

Observe that the set system $\mathcal{A} = (\mathbb{R} \times \{0,1\}, \mathcal{I} \times \{0,1\})$ has VC dimension 2. Hence, from Theorem C.3, with probability at least $1-\delta$, we have that S is an η'-sample for $\mathcal{A}$ with respect to $\mathcal{D}$, where $\eta' := \frac{\eta}{16}$.

The algorithm operates in two steps. In the first step, the algorithm estimates ℓ. For any $t \in \mathbb{R}$, denote by $L_S(f_t)$ the empirical loss of f_t, namely

$$L_S(f_t) := L_S^{\text{left}}(f_t) + L_S^{\text{right}}(f_t)$$

for

$$L_S^{\text{left}}(f_t) := \frac{|([0,t) \times \{1\}) \cap S|}{|S|}$$

and

$$L_S^{\text{right}}(f_t) := \frac{|([t,1] \times \{0\}) \cap S|}{|S|}.$$

(Cardinalities are computed with S viewed as a multiset.)

The algorithm uses the sample S to find the threshold $f_{\hat{t}} \in \mathcal{T}$ defined by

$$\hat{t} := \text{argmin}_{t \in X} L_S(f_t),$$

where $X = \{x_1, \ldots, x_m, 1\}$.

The algorithm estimates ℓ by taking

$$\hat{\ell} := L_S(f_{\hat{t}})/2 + 3\eta'.$$

We argue that $\hat{\ell}$ is a good estimate whenever S is an η'-sample: Let $f^* = \text{argmin}_{f \in \mathcal{T}} L_{\mathcal{D}}(f)$. If S is an η'-sample then

$$\begin{aligned} L_{\mathcal{D}}(f_{\hat{t}}) &\leq L_S(f_{\hat{t}}) + 2\eta' \\ &= \min_{t \in X} L_S(f_t) + 2\eta' \\ &= \min_{t \in \mathbb{R}} L_S(f_t) + 2\eta' \\ &\leq L_S(f^*) + 2\eta' \\ &\leq L_{\mathcal{D}}(f^*) + 4\eta'. \end{aligned}$$

Therefore,

$$\begin{aligned} |L_S(f_{\hat{t}}) - L_{\mathcal{D}}(f^*)| &\leq |L_S(f_{\hat{t}}) - L_{\mathcal{D}}(f_{\hat{t}})| + |L_{\mathcal{D}}(f_{\hat{t}}) - L_{\mathcal{D}}(f^*)| \\ &\leq 2\eta' + 4\eta' = 6\eta'. \end{aligned}$$

Thus, the estimate $\hat{\ell}$ satisfies

$$|\hat{\ell} - \ell| = \left| \frac{L_S(f_{\hat{t}})}{2} + 3\eta' - \frac{L_{\mathcal{D}}(f^*)}{2} \right| \leq 3\eta' + \frac{|L_S(f_{\hat{t}}) - L_{\mathcal{D}}(f^*)|}{2} \leq 6\eta'.$$

Furthermore,

$$\begin{aligned}\hat{\ell} &= \frac{L_S(f_{\hat{t}})}{2} + 3\eta' \\ &\geq \frac{L_{\mathcal{D}}(f^*)}{2} - \frac{|L_S(f_{\hat{t}}) - L_{\mathcal{D}}(f^*)|}{2} + 3\eta' \\ &\geq \frac{L_{\mathcal{D}}(f^*)}{2} = \ell.\end{aligned}$$

This completes the first step.

In the second step, the algorithm calculates

$$(\hat{a},\hat{b}) := \operatorname{argmin}_{a',b' \in X\colon\, a' \leq b'} \left|L_S^{\text{left}}(f_{a'}) - \hat{\ell}\right| + \left|L_S^{\text{right}}(f_{b'}) - \hat{\ell}\right|.$$

We claim that $(\hat{a},\hat{b})$ is an η-valid certificate of loss $\hat{\ell}$. From Claim 3.5 and the assumption that $\mathcal{D}$ is absolutely continuous, there exist (a,b) constituting a 0-valid certificate of loss exactly ℓ.

Denote

$$\begin{aligned}\hat{A} &= [0,\hat{a}) \times \{1\}, \quad A = [0,a) \times \{1\} \\ \hat{B} &= [\hat{b},1] \times \{0\}, \quad B = [b,1] \times \{0\}.\end{aligned}$$

Then

$$\begin{aligned}|\mathcal{D}(\hat{A}) - \hat{\ell}| + |\mathcal{D}(\hat{B}) - \hat{\ell}| &\leq |\mathcal{D}(\hat{A}) - L_S^{\text{left}}(f_{\hat{a}})| + |L_S^{\text{left}}(f_{\hat{a}}) - \hat{\ell}| \\ &\qquad + |\mathcal{D}(\hat{B}) - L_S^{\text{right}}(f_{\hat{b}})| + |L_S^{\text{right}}(f_{\hat{b}}) - \hat{\ell}| \\ &\leq |L_S^{\text{left}}(f_{\hat{a}}) - \hat{\ell}| + |L_S^{\text{right}}(f_{\hat{b}}) - \hat{\ell}| + 2\eta' \\ &= \min_{\hat{a},\hat{b} \in X\colon\, \hat{a} \leq \hat{b}} \left|L_S^{\text{left}}(f_{\hat{a}}) - \hat{\ell}\right| + \left|L_S^{\text{right}}(f_{\hat{b}}) - \hat{\ell}\right| + 2\eta' \\ &= \min_{\hat{a},\hat{b} \in \mathbb{R}\colon\, \hat{a} \leq \hat{b}} \left|L_S^{\text{left}}(f_{\hat{a}}) - \hat{\ell}\right| + \left|L_S^{\text{right}}(f_{\hat{b}}) - \hat{\ell}\right| + 2\eta' \\ &\leq \left|L_S^{\text{left}}(f_a) - \hat{\ell}\right| + \left|L_S^{\text{right}}(f_b) - \hat{\ell}\right| + 2\eta' \\ &\leq \left|L_S^{\text{left}}(f_a) - \ell\right| + \left|L_S^{\text{right}}(f_b) - \ell\right| + 14\eta' \\ &= \left|L_S^{\text{left}}(f_a) - \mathcal{D}(A)\right| + \left|L_S^{\text{left}}(f_b) - \mathcal{D}(B)\right| + 14\eta' \\ &\leq \eta' + \eta' + 14\eta' = \eta.\end{aligned}$$

We conclude that $(\hat{a},\hat{b})$ is an η-valid certificate of loss at least ℓ, provided that S is an η'-sample with respect to $\mathcal{D}$, which happens with probability at least $1-\delta$. Seeing as the algorithm runs in time polynomial in the number of samples, the proof is complete. ■

3.1.3 Warm-Up Summary

We explained how certificates of loss induce a proof system for PAC verification, and described a specific instance of this for the class $\mathcal{T}$ of threshold functions. We saw that the honest prover is able to generate a message $(\tilde{h}, \tilde{C}, \tilde{\ell})$ that is accepted by the verifier. If $\tilde{h}$ has loss greater than double the true loss, no certificate can convince the verifier to accept $\tilde{h}$. Both the verifier and the honest prover are efficient. The certificate is not tight; if the true loss is $\ell = L_{\mathcal{D}}(\mathcal{T})$, the certificate of loss only proves that the loss is at least $\frac{\ell}{2}$.

However, the example of the class $\mathcal{T}$ is lacking an essential ingredient. The sample complexity used by the verifier is the same as is necessary for learning without a prover, and so delegation is not beneficial. In the next section, we present a generalization of this class, where there is a substantial gap between the resources necessary for verification and those required for learning, making it worthwhile to delegate the learning task to an untrusted prover.

3.2 Efficient PAC Verification for the Class $\mathcal{T}_d$ of Multi-Thresholds

In the warm-up we saw certificates of loss that induce a proof system for PAC verification for the class of thresholds $\mathcal{T}$. We now extend this construction to a class $\mathcal{T}_d$ of multi-thresholds, construct a PAC verification proof system for $\mathcal{T}_d$ that obtains the following sample complexity separation between PAC verification on the one hand and PAC learning and tolerant testing or distance approximation on the other hand.

Lemma 3.8. *There exists a sequence of classes of functions*

$$\mathcal{T}_1, \mathcal{T}_2, \mathcal{T}_3, \ldots \subseteq \{0,1\}^{\mathbb{R}}$$

such that for any fixed $\varepsilon, \delta \in (0, \frac{1}{2})$ all of the following hold:

(i) $\mathcal{T}_d$ is 2-PAC verifiable, where the verifier uses

$$m_V = O\left(\frac{\sqrt{d}\log(d)\log\left(\frac{1}{\delta}\right)}{\varepsilon^6}\right)$$

random samples, the honest prover uses

$$m_P = O\left(\frac{d^3\log^2(d)}{\varepsilon^4}\log\left(\frac{d}{\varepsilon}\right) + \frac{d\sqrt{d}\log(d)}{\varepsilon^2}\log\left(\frac{1}{\delta}\right)\right)$$

random samples, and each of them runs in time polynomial in its number of samples.

(ii) Agnostic PAC learning $\mathcal{T}_d$ requires $\Omega\left(\frac{d+\log(\frac{1}{\delta})}{\varepsilon^2}\right)$ samples.

(iii) If $\varepsilon \leq \frac{1}{32}$ then 2-PAC learning the class $\mathcal{T}_d$ requires $\Omega\left(\frac{d}{\log(d)}\right)$ samples. This is true even if we assume that $L_{\mathcal{D}}(\mathcal{T}_d) > 0$, where $\mathcal{D}$ is the underlying distribution.

(iv) Testing whether $L_{\mathcal{D}}(\mathcal{T}_d) \leq \alpha$ or $L_{\mathcal{D}}(\mathcal{T}_d) \geq \beta$ for any $0 < \alpha < \beta < \frac{1}{2}$ with success probability at least $1 - \delta$ when $\mathcal{D}$ is an unknown distribution (without the help of a prover) requires

$\Omega\left(\frac{d}{\log(d)}\right)$ *random samples from $\mathcal{D}$.*

3.2.1 The Class $\mathcal{T}_d$

We start by defining the class of multi-thresholds.

Definition 3.9. *For any $d \in \mathbb{N}$, denote by $\mathcal{T}_d$ the class of functions*

$$\mathcal{T}_d = \{f_{t_1,\ldots,t_d} : t_1,\ldots,t_d \in \mathbb{R}\}$$

where for all $t_1,\ldots,t_d \in \mathbb{R}$ and $x \in [0,d]$, the function $f_{t_1,\ldots,t_d} : \mathbb{R} \to \{0,1\}$ is given by

$$f_{t_1,\ldots,t_d}(x) = \begin{cases} 0 & x < t_{\lceil x \rceil} \\ 1 & x \geq t_{\lceil x \rceil}, \end{cases}$$

and $f_{t_1,\ldots,t_d}$ vanishes on the complement of $[0,d]$.

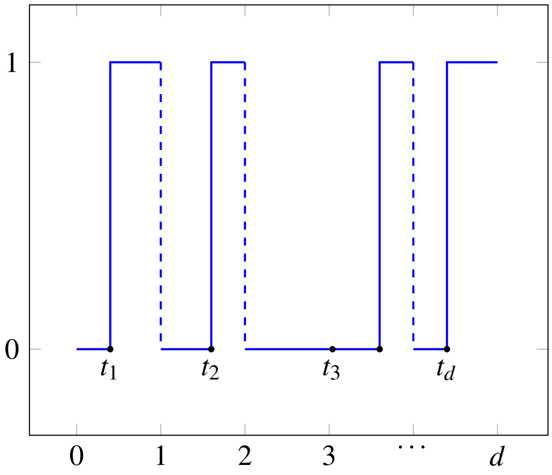

Figure 4: Example of a function in $\mathcal{T}_d$.

3.2.2 Existence of Certificates of Loss for $\mathcal{T}_d$

For each $i \in [d]$, the class $\mathcal{T}_d$ restricted to $[i-1,i]$ is a shifted copy of the class $\mathcal{T}$. Hence, exactly as we did for $\mathcal{T}$, we can construct a certificate of loss which proves that $\mathcal{T}_d$ must have loss ℓ_i within the interval $[i-1,i]$. Therefore, we define certificates for $\mathcal{T}_d$ as collections of d certificates of loss for $\mathcal{T}$.

Definition 3.10. *Let $\mathcal{D} \in \Delta(\mathbb{R} \times \{0,1\})$ be a distribution and $\ell, \eta \geq 0$. A* <u>*certificate of loss at least ℓ for the class $\mathcal{T}_d$*</u> *is a tuple*

$$(C_1, \ell_1, C_2, \ell_2 \ldots, C_d, \ell_d)$$

where for all $i \in [d]$:

- $C_i = (a_i, b_i)$,
- $i - 1 < a_i \leq b_i \leq i$,
- $\ell_i \geq 0$, *and*

$$\sum_{i=1}^{d} \ell_i = \ell.$$

The certificate is *η-valid with respect to $\mathcal{D}$* *if the events*

$$A_i = [i-1, a_i) \times \{1\}$$

$$B_i = [b_i, i] \times \{0\}$$

defined for all $i \in [d]$ satisfy

$$\sum_{i=1}^{d} |\mathcal{D}(A_i) - \ell_i| + |\mathcal{D}(B_i) - \ell_i| \leq \eta.$$

The following analogs of Claims 3.4 and 3.5 follow similarly.

Claim 3.11. *Let $\mathcal{D} \in \Delta(\mathbb{R} \times \{0,1\})$ be a distribution and $\ell, \eta \geq 0$. If $\mathcal{D}$ has a certificate of loss at least ℓ for $\mathcal{T}_d$ that is η-valid with respect to $\mathcal{D}$, then every function in $\mathcal{T}_d$ must have loss at least $\ell - \eta$ with respect to $\mathcal{D}$.*

Claim 3.12. *Let $\mathcal{D} \in \Delta(\mathbb{R} \times \{0,1\})$ be a distribution and $\ell \geq 0$. If $L_{\mathcal{D}}(\mathcal{T}_d) = \ell$ then there exists a* 0*-valid certificate of loss at least $\frac{\ell}{2}$ for $\mathcal{T}_d$ with respect to $\mathcal{D}$.*

3.2.3 Efficient Generation and Verification of Certificates for $\mathcal{T}_d$

The following is a straightforward analogue of Claim 3.7.

Claim 3.13 (Efficient Generation). *There exists an algorithm as follows. For any distribution $\mathcal{D} \in \Delta([0,1] \times \{0,1\})$ and any $\delta, \eta \in (0, \frac{1}{2})$, the algorithm outputs a certificate of loss for $\mathcal{T}_d$ that with probability at least $1 - \delta$ is an η-valid certificate of loss at least $\ell = L_{\mathcal{D}}(\mathcal{T}_d)/2$ with respect to $\mathcal{D}$. The algorithm uses*

$$O\left(\frac{d^2}{\eta^2} \log \frac{d}{\eta} + \frac{d^2}{\eta^2} \log \frac{1}{\delta}\right)$$

i.i.d. samples from $\mathcal{D}$ and runs in time polynomial in the number of samples.

Proof sketch. The proof follows the same lines as for Claim 3.7. Recall that in that proof, the algorithm takes a sample of size $O\left(\frac{1}{\eta^2} \log \frac{1}{\eta} + \frac{1}{\eta^2} \log \frac{1}{\delta}\right)$. Whenever the sample is an η'-sample with respect to the set system $\mathcal{A}$ defined in that proof, the algorithm is able to generate a certificate that is η-valid.

Here, the algorithm instead takes a sample that with probability at least $1 - \delta$ is an $\frac{\eta'}{d}$-sample with respect to $\mathcal{A}$. This leads to the sample size mentioned in the statement. The algorithm proceeds as in the previous case, using the sample to generate d certificates of loss, one for each interval of the form $[i-1,i]$ for $i\in[d]$. Whenever the sample is an $\frac{\eta'}{d}$-sample, each of these certificates will be $\frac{\eta}{d}$-valid. Combining these certificates together yields a certificate for $\mathcal{T}_d$ that is η-valid. ■

Agnostic PAC learning $\mathcal{T}_d$ requires

$$\Theta\left(\frac{d+\log(\frac{1}{\delta})}{\varepsilon^2}\right)$$

samples, because its VC dimension is d. Thus, the certificate generation procedure outlined above requires that the prover use a larger number of samples than what is necessary for learning. This may be worthwhile, because, as stated in the following claim, the verifier can verify the certificate using less samples than what is required for learning.

Claim 3.14 (**Efficient verification**). *Let $d\in\mathbb{N}$ and $\lambda\in(0,1)$. Let $C=(C_1,\ell_1,\dots,C_d,\ell_d)$ be a certificate of loss ℓ for $\mathcal{T}_d$, and let $\mathcal{D}$ be a distribution. There exists an algorithm that takes*

$$m=O\left(\log\left(\frac{1}{\delta}\right)\frac{\sqrt{d}}{\lambda^6}\log(d)\right)$$

samples from $\mathcal{D}$, and satisfies:

- ***Completeness.*** *Let*

$$\lambda' := \frac{\lambda^3}{300\sqrt{d}\log d}.$$

If C is λ'-valid with respect to $\mathcal{D}$, then the algorithm accepts with probability at least $1-\delta$.

- ***Soundness.*** *If C is not 2λ-valid with respect to $\mathcal{D}$, then the algorithm rejects with probability at least $1-\delta$.*

Proof. The proof uses ideas from distribution identity testing stated in Corollary D.2. For all $i\in[d]$, let

$$A_i=[i-1,a_i)\times\{1\}, \text{ and}$$

$$B_i=[b_i,i]\times\{0\}.$$

The algorithm is required to decide whether the validity v of the certificate is less than λ', i.e., whether

$$v:=\sum_{i=1}^{d}|\mathcal{D}(A_i)-\ell_i|+|\mathcal{D}(B_i)-\ell_i|\leq\lambda',$$

or whether $v>2\lambda$.

Form the partition $R:=\{A_1,B_1,\dots,A_d,B_d,E\}$ of $\mathbb{R}\times\{0,1\}$, where

$$E=(\mathbb{R}\times\{0,1\})\setminus\Big(\bigcup_{i\in[d]}A_i\cup B_i\Big).$$

Define two probability functions, $\mathcal{D}_R$ and $\mathcal{D}^*$, both over this finite set R of cardinality $2d+1$. Let $\mathcal{D}_R$ be the distribution induced on R by $\mathcal{D}$; namely, $\mathcal{D}_R(r)=\mathcal{D}(r)$ for each $r\in R$. Let $\mathcal{D}^*$ denote the distribution over R corresponding to the certificate C. Namely, $\mathcal{D}^*(A_i)=\mathcal{D}^*(B_i)=\ell_i$ for all

$i \in [d]$, and $\mathcal{D}^*(E) = 1 - 2\sum_{i=1}^{d} \ell_i = 1 - 2\ell$.

Consider the mapping M_R that sends each point to the member of R it belongs to:

$$M_R(x,y) = \begin{cases} A_i & (x,y) \in A_i, \\ B_i & (x,y) \in B_i, \\ E & \text{otherwise.} \end{cases}$$

Observe that if $S = \big((x_1,y_1),\ldots,(x_m,y_m)\big)$ is sampled i.i.d. from $\mathcal{D}$, then

$$M_R(S) := (M_R(x_1,y_1),\ldots,M_R(x_m,y_m))$$

is an i.i.d. sample from $\mathcal{D}_R$. Observe the following connection between $d_{\mathsf{TV}}(\mathcal{D}_R,\mathcal{D}^*)$ and the validity v of the certificate:

$$\begin{aligned} v &= \sum_{i=1}^{d} |\mathcal{D}(A_i) - \ell_i| + |\mathcal{D}(B_i) - \ell_i| \\ &= \sum_{i=1}^{d} |\mathcal{D}_R(A_i) - \mathcal{D}^*(A_i)| + |\mathcal{D}_R(B_i) - \mathcal{D}^*(B_i)| \\ &= 2d_{\mathsf{TV}}(\mathcal{D}_R,\mathcal{D}^*) - |\mathcal{D}_R(E) - \mathcal{D}^*(E)|. \end{aligned}$$

Furthermore,

$$|\mathcal{D}_R(E) - \mathcal{D}^*(E)| \leq d_{\mathsf{TV}}(\mathcal{D}_R,\mathcal{D}^*).$$

Thus,

$$d_{\mathsf{TV}}(\mathcal{D}_R,\mathcal{D}^*) \leq v \leq 2d_{\mathsf{TV}}(\mathcal{D}_R,\mathcal{D}^*).$$

The algorithm operates as follows. It executes the distribution identity test stated in Corollary D.2 with respect to distribution $\mathcal{D}^*$ and the sample $M_R(S)$. Because $\mathcal{D}^*$ is a distribution over a set of size $2d+1$, taking a sample $M_R(S)$ of size m as specified in the statement is sufficient to ensure that with probability at least $1-\delta$, the test distinguishes correctly between the case $d_{\mathsf{TV}}(\mathcal{D}_R,\mathcal{D}^*) \leq \lambda'$ and the case $d_{\mathsf{TV}}(\mathcal{D}_R,\mathcal{D}^*) \geq \lambda$. The algorithm accepts the certificate if and only if the test concludes that $d_{\mathsf{TV}}(\mathcal{D}_R,\mathcal{D}^*) \leq \lambda'$.

The desired properties hold:

- Completeness. If $v \leq \lambda'$, then $d_{\mathsf{TV}}(\mathcal{D}_R,\mathcal{D}^*) \leq v \leq \lambda'$, and so with probability at least $1-\delta$ the algorithm accepts.
- Soundness. If $v > 2\lambda$, then $\lambda < \frac{v}{2} \leq d_{\mathsf{TV}}(\mathcal{D}_R,\mathcal{D}^*)$, and so with probability at least $1-\delta$ the algorithm rejects.

This concludes the proof. ■

We now use the previous two claims to construct the efficient PAC verification protocol for ‘$\mathcal{T}_d$.

Claim 3.15. *$\mathcal{T}_d$ is 2-PAC verifiable with sample and runtime complexities as in part (i) of Lemma 3.8.*

Proof. The interactive proof system for 2-PAC verification operates as follows. Let

$\mathcal{D} \in \Delta(\mathbb{R} \times \{0,1\})$, and let $\ell = L_{\mathcal{D}}(\mathcal{T}_d)$.

1. The honest prover learns a function $\tilde{h} \in \mathcal{T}_d$ that has loss at most $\ell + \frac{\varepsilon}{6}$, with probability at least $1 - \frac{\delta}{4}$. This can be done with the required sample complexity, and the computation runs in time polynomial in the number of samples, because an ERM can be computed in polynomial time (as discussed in the proof of Claim 3.7).
2. From Claim 3.12, there exists a 0-valid certificate of loss at least $\frac{\ell}{2}$ for $\mathcal{T}_d$ with respect to $\mathcal{D}$, where $\ell = L_{\mathcal{D}}(\mathcal{T}_d)$. From Claim 3.13, the honest prover can generate a certificate $\tilde{C} = (C_1, \ell_1, \dots, C_d, \ell_d)$ of loss $\tilde{\ell} := \sum_i \ell_i \geq \frac{\ell}{2}$ that with probability at least $1 - \frac{\delta}{4}$ is η-valid, for
$$\eta = \frac{(\varepsilon/8)^2}{300\sqrt{d}\log(d)}.$$
The prover can do this using m_P samples as in the statement.
3. The honest prover sends $(\tilde{h}, \tilde{C}, \tilde{\ell})$ to the verifier V.
4. The verifier V uses $O\left(\log\left(\frac{1}{\delta}\right)/\varepsilon^2\right)$ samples to estimate the loss $L_{\mathcal{D}}(\tilde{h})$ up to an additive error of $\frac{\varepsilon}{6}$ with confidence at least $1 - \frac{\delta}{4}$, and rejects if the estimate is greater than $2\tilde{\ell} + \frac{\varepsilon}{3}$. This ensures that V accepts only if $L_{\mathcal{D}}(\tilde{h}) \leq 2\tilde{\ell} + \frac{\varepsilon}{2}$.
5. From Claim 3.14, the verifier can use m_V samples to verify $\tilde{C}$, such that if $\tilde{C}$ is η-valid then V accepts with probability at least $1 - \frac{\delta}{4}$, and if $\tilde{C}$ is not $\frac{\varepsilon}{4}$-valid, then V rejects with probability at least $1 - \frac{\delta}{4}$.

For the completeness, observe that when interacting with the honest prover, each of the operations in Steps 1, 2, 4 and 5 succeeds with probability at least $1 - \frac{\delta}{4}$, and so with probability at least $1 - \delta$ they all succeed and V accepts $\tilde{h}$, which has loss at most $\ell + \frac{\varepsilon}{6}$.

For soundness, let $H \in \mathcal{T}_d \cup \{\text{reject}\}$ denote the output of V, and let
$$B = \{h \in \mathcal{T}_d : \; L_{\mathcal{D}}(h) > 2\ell + \varepsilon\}.$$
Assume towards a contradiction there exists a prover P for which $\mathbb{P}[H \in B] > \delta$. Let W denote the message $(\tilde{h}, \tilde{C}, \tilde{\ell})$ sent by P. Because
$$\delta < \mathbb{P}[H \in B] = \sum_w \mathbb{P}[H \in B \mid W = w]\,\mathbb{P}[W = w],$$
there exists some $w_0 = (\tilde{h}_0, \tilde{C}_0, \tilde{\ell}_0)$ such that
$$\mathbb{P}[H \in B \mid W = w_0] > \delta. \tag{11}$$
When the verifier V does not reject, V outputs the hypothesis sent by P. Thus, $\tilde{h}_0 \in B$ and yet V accepts w_0 with probability $> \delta$. We show that this is impossible, based on the following two facts:

- If $L_{\mathcal{D}}(\tilde{h}_0) > 2\tilde{\ell} + \frac{\varepsilon}{2}$, then from Step 4, the verifier V accepts w_0 with probability at most $\frac{\delta}{4}$.
- If $\tilde{C}_0$ is not an $\frac{\varepsilon}{4}$-valid certificate of loss $\tilde{\ell}$, then from Step 5, the verifier V accepts w_0 with probability at most $\frac{\delta}{4}$.

This implies that $\tilde{h}_0 \in B$, that $L_{\mathcal{D}}(\tilde{h}_0) \leq 2\tilde{\ell} + \frac{\varepsilon}{2}$ and that $\tilde{C}_0$ is an $\frac{\varepsilon}{4}$-valid certificate of loss $\tilde{\ell}$.

Claim 3.11 yields the contradiction:

$$\ell = L_{\mathcal{D}}(\mathcal{T}_d) \geq \tilde{\ell} - \frac{\varepsilon}{4} \geq \frac{L_{\mathcal{D}}(\tilde{h}_0)}{2} - \frac{\varepsilon}{2} > \ell. \blacksquare$$

3.3 Lower Bounds for Closeness Testing and 2-PAC Learning of the Class $\mathcal{T}_d$

In this section we show near-linear lower bounds for testing closeness and 2-PAC learning of the class $\mathcal{T}_d$.

Definition 3.16. *Let $0 < \alpha < \beta < 1$ and $d \in \mathbb{N}$. The (α, β, d)-threshold closeness testing problem is the following promise problem. Given sample access to an unknown distribution $\mathcal{D} \in \Delta([0,d] \times \{0,1\})$, distinguish between the following two cases:*

(i) $L_{\mathcal{D}}(\mathcal{T}_d) \leq \alpha$.

(ii) $L_{\mathcal{D}}(\mathcal{T}_d) \geq \beta$.

Lemma 3.17. *Fix $0 < \alpha < \beta < \frac{1}{2}$. Any tester that uses sample access to an unknown distribution $\mathcal{D} \in \Delta([0,d] \times \{0,1\})$ and solves the (α, β, d)-threshold closeness testing problem correctly with probability at least $\frac{2}{3}$ for all $d \in \mathbb{N}$ must use at least $\Omega\left(\frac{d}{\log(d)}\right)$ samples from $\mathcal{D}$.*

The proof of this lemma relies on a lower bound for testing support size of a distribution.

Definition 3.18. *Let $0 < \alpha < \beta < 1$ and let $n \in \mathbb{N}$. The (α, β, n)-support size testing problem is the following promise problem. Let $\mathcal{D} \in \Delta([n])$ be an unknown distribution such that $\forall i \in \text{supp}(\mathcal{D}): \mathcal{D}(i) \geq \frac{1}{n}$. Given sample access to $\mathcal{D}$, distinguish between the following two cases:*

(i) $|\text{supp}(D)| \leq \alpha \cdot n$.

(ii) $|\text{supp}(D)| \geq \beta \cdot n$.

The following tight lower bound for this problem is due to G. Valiant and Valiant (2010a, 2010b). The formulation we use a is adapted from Canonne (2015).[16]

Theorem 3.19 (G. Valiant & Valiant, 2010a, 2010b; Canonne, 2015, Theorem 3.5.3)**.** *Let $0 < \alpha < \beta < 1$. Any tester that uses sample access to an unknown distribution $\mathcal{D} \in \Delta([n])$ and solves the (α, β, n)-support size testing problem correctly with probability at least $\frac{2}{3}$ for all $n \in \mathbb{N}$ must use at least $\Omega\left(\frac{n}{\log(n)}\right)$ samples from $\mathcal{D}$.*

Proof of Lemma 3.17. We show the following reduction from the support size testing problem to the threshold closeness problem: Assume T' is a tester that solves the (α, β, d)-threshold closeness testing problem correctly with probability at least $\frac{2}{3}$ for all $d \in \mathbb{N}$ using $m(d)$ samples. Then there

[16] See also the discussion following Theorem 3.1 in Ron and Tsur (2013), and Theorem 5.3 in G. Valiant (2012). Similar bounds that appear in P. Valiant (2011, Claim 3.10) and Raskhodnikova, Ron, Shpilka, and Smith (2009, Theorem 2.1 and Corollary 2.2) are slightly weaker, but would also suffice for separating between 2-PAC verification versus 2-PAC learning of $\mathcal{T}_d$, as in Claim 3.20.

exists a tester T that solves the $(2\alpha, 2\beta, d)$-support size testing problem correctly with probability at least $\frac{2}{3}$ for all $d \in \mathbb{N}$, and uses at most $m(d)$ samples.

For any distribution $\mathcal{D} \in \Delta([d])$, define a corresponding distribution $\mathcal{D}' \in \Delta([0,d] \times \{0,1\})$ as follows. For all $i \in [d]$, let $a_i = i - \frac{3}{4}$ and $b_i = i - \frac{1}{4}$. Then $\mathcal{D}'(a_i, 1) = \frac{\mathcal{D}(i)}{2}$ and $\mathcal{D}'(b_i, 0) = \frac{1}{2d}$ for all $i \in [d]$, and $\mathcal{D}'$ vanishes elsewhere.

Given sample access to $\mathcal{D}$, it is possible to simulate sample access to $\mathcal{D}'$: with probability $\frac{1}{2}$, sample $i \in \mathcal{D}$, and output $(a_i, 1)$; with probability $\frac{1}{2}$ select $i \in [d]$ uniformly at random, and output $(b_i, 0)$.

Because $\mathcal{T}_d$ consists of monotone increasing thresholds,

$$\begin{aligned} L_{\mathcal{D}'}(\mathcal{T}_d) &= \sum_{i=1}^{n} \min\{\mathcal{D}'(a_i,1), \mathcal{D}'(b_i,0)\} \\ &= \sum_{i=1}^{n} \min\left\{\frac{\mathcal{D}(i)}{2}, \frac{1}{2d}\right\} \\ &\overset{(*)}{=} \sum_{i \in [d]\setminus \mathrm{supp}(\mathcal{D})} 0 + \sum_{i \in \mathrm{supp}(\mathcal{D})} \frac{1}{2d} \\ &= \frac{|\mathrm{supp}(\mathcal{D})|}{2d}. \end{aligned}$$

Equality $(*)$ holds whenever $\mathcal{D}$ is an input for the support size testing problem, because we assume that $\mathcal{D}(i) \geq \frac{1}{d}$ for all $i \in \mathrm{supp}(\mathcal{D})$.

To solve the $(2\alpha, 2\beta, d)$-support size testing problem, T operates as follows. Given access to an unknown distribution $\mathcal{D} \in \Delta([d])$, it simulates an execution of T' with access to $\mathcal{D}'$ that solves the (α, β, d)-threshold closeness testing problem. If T' decides that $L_{\mathcal{D}'}(\mathcal{T}_d) \leq \alpha$, then T outputs that $|\mathrm{supp}(\mathcal{D})| \leq 2\alpha \cdot d$, and if T' decides that $L_{\mathcal{D}'}(\mathcal{T}_d) \geq \beta$ then T outputs that $|\mathrm{supp}(\mathcal{D})| \geq 2\beta \cdot d$. T decides correctly with probability at least $\frac{2}{3}$, because we assume that T' decides correctly with probability at least $\frac{2}{3}$, and

$$\begin{aligned} L_{\mathcal{D}'}(\mathcal{T}_d) \leq \alpha &\iff |\mathrm{supp}(\mathcal{D})| \leq 2\alpha \cdot d \\ L_{\mathcal{D}'}(\mathcal{T}_d) \geq \beta &\iff |\mathrm{supp}(\mathcal{D})| \geq 2\beta \cdot d. \end{aligned}$$

T requires at most as many samples as T' does, because simulating one sample from $\mathcal{D}'$ requires taking at most one sample from $\mathcal{D}$.

The claim follows from this reduction and from Theorem 3.19. ■

The previous claim also implies the following lower bound for 2-PAC learning of $\mathcal{T}_d$ without the help of a prover.

Claim 3.20. *2-PAC learning the class $\mathcal{T}_d$ with $\varepsilon \in (0, \frac{1}{32})$ requires at least $\Omega\left(\frac{d}{\log(d)}\right)$ random samples. This is true even if we assume that the unknown underlying distribution $\mathcal{D}$ satisfies $L_{\mathcal{D}}(\mathcal{T}_d) > 0$.*

Proof of Claim 3.20. Assume for contradiction that there exists an algorithm A that 2-PAC learns

$\mathcal{T}_d$ using only $o\left(\frac{d}{\log(d)}\right)$ samples from $\mathcal{D}$. We construct a tester T that solves the $(\frac{1}{8},\frac{3}{8},d)$-threshold closeness testing problem using only $o\left(\frac{d}{\log(d)}\right)$ samples.

Let $\mathcal{D}\in\Delta([0,d]\times\{0,1\})$ be the unknown distribution that T has access to. Fix positive $\varepsilon\leq\frac{1}{32}$, $\delta\leq\frac{1}{6}$. T operates as follows. It simulates A using samples from $\mathcal{D}$ to obtain $h\in\mathcal{T}_d$ such that with probability at least $1-\delta$,

$$L_{\mathcal{D}}(h)\leq 2\cdot L_{\mathcal{D}}(\mathcal{T}_d)+\varepsilon. \tag{12}$$

Next, it takes an additional $O(1)$ samples from $\mathcal{D}$ to obtain an estimate $\widehat{\ell}$ such that with probability at least $1-\delta$,

$$\left|\widehat{\ell}-L_{\mathcal{D}}(h)\right|\leq\varepsilon \tag{13}$$

If $\widehat{\ell}\leq\frac{5}{16}$, then T outputs $L_{\mathcal{D}}(\mathcal{T}_d)\leq\frac{1}{8}$. Otherwise, if $\widehat{\ell}>\frac{5}{16}$, then T outputs $L_{\mathcal{D}}(\mathcal{T}_d)\geq\frac{3}{8}$.

From the union bound, with probability at least $1-2\delta\geq\frac{2}{3}$, both (12) and (13) hold. Correctness follows by considering each case separately:

- **Case 1:** $L_{\mathcal{D}}(\mathcal{T}_d)\leq\frac{1}{8}$**.** Then
$$\widehat{\ell}\leq L_{\mathcal{D}}(h)+\varepsilon\leq 2L_{\mathcal{D}}(\mathcal{T}_d)+2\varepsilon\leq\frac{2}{8}+\frac{2}{32}=\frac{5}{16}.$$
- **Case 2:** $L_{\mathcal{D}}(\mathcal{T}_d)\geq\frac{3}{8}$**.** Then
$$\widehat{\ell}\geq L_{\mathcal{D}}(h)-\varepsilon\geq L_{\mathcal{D}}(\mathcal{T}_d)-\varepsilon\geq\frac{3}{8}-\frac{1}{32}=\frac{11}{32}>\frac{5}{16}.$$

Finally, T uses the same number of samples as A does, which is a contradiction to Lemma 3.17.

From an amplification argument, the claim holds for any $\delta\in(0,\frac{1}{2})$. To see that the claim is true even if we assume that $L_{\mathcal{D}}(\mathcal{T}_d)>0$, note that the distribution $\mathcal{D}'$ constructed in the proof of Lemma 3.17 always satisfies $L_{\mathcal{D}'}(\mathcal{T}_d)\geq\frac{1}{2d}$, and so we may assume that the hard distributions for T in the current proof have this property. ■

Finally, we have obtained the desired separation, showing that PAC verification can be more efficient than PAC learning and closeness testing.

Proof of Lemma 3.8.

(i) Follows from Claim 3.15.

(ii) Follows from Theorem 1.10, because $\mathsf{VC}(\mathcal{T}_d)\geq d$.

(iii) Follows from Claim 3.20.

(iv) Follows from Lemma 3.17. ■

4 Lower Bound of $\tilde{\Omega}(d)$

We saw in Section 3.2 that for every natural number d there exists a class of VC dimension d that has a verification protocol requiring only $O\left(\sqrt{d}\right)$ samples for the verifier – a considerable saving compared to the cost of learning, which is $\Omega(d)$. A natural question to ask is, "Does every class of VC dimension d admit a verification protocol with sample complexity $O\left(\sqrt{d}\right)$?" In other words, is it always worthwhile to delegate a learning task? In this section we provide a partial negative answer to this question, presenting for every natural number d an example of a class with VC dimension $O(d\log(d))$ where the sample complexity for verification is $\Omega(d)$. That is, for these classes the sample complexity of learning and of verification are equal up to a logarithmic factor. Formally:

Lemma 4.1. *For every $\varepsilon,\delta \in \left(0,\frac{1}{8}\right)$ there exist constants $c_0,c_1,c_2 > 0$ and a sequence of classes $\mathcal{H}_1,\mathcal{H}_2,\dots$ such that:*

- *For all $d \in \mathbb{N}$, the class $\mathcal{H}_d$ has VC dimension at most $c_0 \cdot d\log(d)$.*
- *The sample complexity of proper 1-PAC verifying $\mathcal{H}_d$ is $\Omega(d)$. That is, if $(V_1,P_1),(V_2,P_2),\dots$ is a sequence such that for all $d \in \mathbb{N}$, (V_d,P_d) is an interactive proof systems that 1-PAC verifies $\mathcal{H}_d$ using oracles that provide random samples such that the output is either* 'reject' *or in $\mathcal{H}_d$, then for all $d \geq c_1$, V_d uses at least $c_2 \cdot d$ random samples when executed on input (ε,δ).*

4.1 The Class $\mathcal{H}_d$

Notation 4.2. *For any $d \in \mathbb{N}$, we write $\mathcal{X}_d$ to denote some fixed set of cardinality $n_d = 2d^2$.*

Notation 4.3. *For any $d \in \mathbb{N}$, we write $\mathcal{F}_{d,\frac{1}{2}}$ to denote the set of balanced boolean functions over $\mathcal{X}_d$, namely,*

$$\mathcal{F}_{d,\frac{1}{2}} = \left\{ f \in \{0,1\}^{\mathcal{X}_d} : \, |f^{-1}(1)| = \frac{n_d}{2} = |f^{-1}(0)| \right\}.$$

Notation 4.4. *For any $f \in \mathcal{F}_{d,\frac{1}{2}}$, we write $\mathcal{D}_f$ to denote the distribution over tuples in $\mathcal{X}^t$ in which t elements are samples independently and uniformly at random from* supp(f). *Namely, for any $(x_1,\dots,x_t) \in \mathcal{X}^t$,*

$$\mathcal{D}_f\left((x_1,\dots,x_t)\right) = \begin{cases} \left(\frac{2}{n}\right)^t & x_1,\dots,x_t \in \mathrm{supp}(f) \\ 0 & \text{o.w.} \end{cases}$$

Furthermore, for any $F = \{f_1,\dots,f_k\} \subseteq \mathcal{F}_{d,\frac{1}{2}}$, we write $\mathcal{D}_F$ to denote the distribution over $\mathcal{X}^t$ given by

$$\mathcal{D}_F(x_1,\dots,x_t) := \frac{1}{k}\sum_{i=1}^{k} \mathcal{D}_{f_i}(x_1,\dots,x_t).$$

Lastly, $\mathcal{U}_{\mathcal{X}^t}$ denotes the uniform distribution over $\mathcal{X}^t$.

We now define the sequence of classes $\mathcal{H}_d$ for $d \in \mathbb{N}$.

Definition 4.5. *Fix $\delta \in (0,1)$. For any $d \in \mathbb{N}$, let $\mathcal{X}_d = [n_d]$ for $n_d = 2d^2$, and let $t_d = \left\lfloor c_2 \cdot d \right\rfloor$*

where

$$c_2 = \sqrt{\frac{\log(1-\delta/3)}{\log(1/2e)}}.$$

The class $\mathcal{H}_d$ is a subset of $\mathcal{F}_{d,\frac{1}{2}}$ of cardinality

$$k_d = \left(\frac{3n_d^{\sqrt{n_d}}}{\delta}\right)^3$$

which is defined as follows. For all values d in which this is possible, the subset $\mathcal{H}_d$ is chosen such that the following three properties hold:

H1. $d_{\mathsf{TV}}(\mathcal{D}_{\mathcal{H}_d}, \mathcal{U}_{\mathcal{X}^t}) \leq \delta$.

H2. *Every distinct $g_1, g_2 \in \mathcal{H}_d$ satisfy $|\text{supp}(g_1) \cap \text{supp}(g_2)| \leq \frac{3n_d}{8}$.*

H3. *All subsets $X \subseteq \mathcal{X}_d$ of size at most $\sqrt{n}$ satisfy*

$$|\{f \in \mathcal{H}_d : X \subseteq \text{supp}(f)\}| \geq \frac{1}{\delta}.$$

However, if for some value of d there exists no subset of cardinality k_d that satisfies these properties, then for that d the class $\mathcal{H}_d$ is simply fixed to be some arbitrary subset of cardinality k_d.

Remark 4.6. *It is not obvious that a set $\mathcal{H}_d$ as in the definition above exists. In Lemma 4.10 below, we prove the existence of $\mathcal{H}_d$ for all d large enough.*

Notation 4.7. *For the remainder of this section, we often neglect to write the subscript d wherever it is readily understood from the context.*

Note that the VC dimension of $\mathcal{H}_d$ is at most $\log(|\mathcal{H}_d|) = O(d \log(d))$, matching the requirement in the lemma.

4.2 Proof Idea

For any d large enough, we want to show that at least $t_d = \Omega(d)$ samples are necessary.

Consider PAC learning the class $\mathcal{H}_d$ in the special case where all $x \in \mathcal{X}$ are labeled 1, but the distribution over $\mathcal{X}_d$ is not known to the prover. Because every hypothesis in the class assigns incorrect labels of 0 to precisely half of the domain, a hypothesis achieves minimal loss if it assigns the 0 labels to a subset of size $\frac{n}{2}$ that has minimal weight with respect to the distribution over $\mathcal{X}_d$. Hence, to be successful the prover must learn enough about the distribution to identify a lightweight subset of size $\frac{n}{2}$ – but doing that requires $\Omega(\sqrt{n}) = \Omega(d)$ samples.

To formalize this idea we construct a stochastic process as follows. Let $P_{\mathcal{U}}$ denote a prover that causes V to accept with probability at least $1 - \delta$ when V receives samples from the uniform distribution over $\mathcal{X}$ (such a prover exists from the completeness property that V satisfies as a PAC learning verifier).

First, a set X_P of t_P samples is taken independently and uniformly from $\mathcal{X}$, where t_P is the number of samples required by $P_{\mathcal{U}}$. Next, two functions f_1 and f_2 are chosen uniformly from $\mathcal{H}_d$, and sets X_1 and X_2 each with t_d i.i.d. samples are taken from $\mathcal{D}_{f_1}$ and $\mathcal{D}_{f_2}$ respectively. A third set $X_{\mathcal{U}}$ is taken from $\mathcal{U}_{\mathcal{X}^t}$. The dependencies between these variables will be designed in such a way that with high probability $X_1 = X_2 = X_{\mathcal{U}}$. All samples are labeled with 1.

Finally, randomness values ρ_V and ρ_P are sampled for the prover and verifier, which are then executed to produce three hypotheses:

$$\begin{aligned} h_1 &:= [V(X_1, \rho_V), P_{\mathcal{U}}(X_P, \rho_P)], \\ h_2 &:= [V(X_2, \rho_V), P_{\mathcal{U}}(X_P, \rho_P)], \\ h_{\mathcal{U}} &:= [V(X_{\mathcal{U}}, \rho_V), P_{\mathcal{U}}(X_P, \rho_P)]. \end{aligned}$$

Observe that for $i = 1, 2$, because $X_i \sim \mathcal{D}_{f_i}$ and V is a PAC learner, with probability at least $1 - \delta$ either h_i is 'reject' or $L_{\mathcal{D}_{f_i}}(h_i) < \varepsilon$.

Observe further that when $X_1 = X_2 = X_{\mathcal{U}}$, the view of V (which consists of its samples, its randomness, and the transcript) is the same in all three executions, entailing that $h_1 = h_2 = h_{\mathcal{U}}$. Additionally, by the definition of $P_{\mathcal{U}}$, with probability at least $1 - \delta$ the output $h_{\mathcal{U}}$ is not 'reject', and so $h_1 = h_2$ are not 'reject'.

However, Property H2 ensures that f_1 and f_2 have a small intersection, causing any hypothesis that has a small loss with respect to $\mathcal{D}_{f_1}$ to have a large loss with respect to $\mathcal{D}_{f_2}$, and vice versa. This is a contradiction to the above observation that $L_{\mathcal{D}_{f_i}}(h_i) < \varepsilon$ for both $i = 1$ and $i = 2$.

Remark 4.8. *Because we are dealing exclusively with the case of learning the constant function that assigns the label* 1 *to all $x \in \mathcal{X}$, for the remainder of this section we will neglect to mention or denote the labels, which are always* 1.

4.3 Proof

We now translate the above proof idea into a formal proof of Lemma 4.1. The main step is to construct the following joint probability space.

Lemma 4.9. *For every $d \in \mathbb{N}$ large enough there exists a probability space with random variables*

$$(f_1, f_2, h_1, h_2, h_{\mathcal{U}}, X_1, X_2, X_{\mathcal{U}}, X_P, \rho_P, \rho_V)$$

such that $f_1, f_2, h_1, h_2, h_{\mathcal{U}} \in \mathcal{H}_d$ and $X_1, X_2, X_{\mathcal{U}} \in \mathcal{X}^t$ and the following properties hold:

P1. *X_P is a tuple of t_P samples taken independently and uniformly from $\mathcal{X}$, and is independent of all other variables.*

P2. *The marginal distribution of $X_{\mathcal{U}}$ is uniform over $\mathcal{X}^t$.*

P3. *For $i = 1, 2$, X_i is distributed according to $\mathcal{D}_{f_i}$. Namely, for any $g \in \mathcal{H}_d$ and any $x_1, \ldots, x_t \in \mathcal{X}$,*

$$\mathbb{P}[X_i = (x_1, \ldots, x_t) \mid f_i = g] = \mathcal{D}_g((x_1, \ldots, x_t)).$$

P4. *$X_1 = X_2$ with probability* 1.

P5. *$X_1 = X_{\mathcal{U}}$ with probability at least $1 - \delta$.*

P6. *ρ_V and ρ_P are randomness values for V and P with suitable marginal distributions and are independent of each other and of all other random variables.*

P7. $h_\alpha = [V(X_\alpha,\rho_V), P_{\mathcal{U}}(X_P,\rho_P)]$ *for* $\alpha \in \{1,2,\mathcal{U}\}$ *with probability* 1.

P8. $|\mathrm{supp}(f_1) \cap \mathrm{supp}(f_2)| \leq \frac{3n}{8}$ *with probability at least* $1-\delta$.

Before constructing the probability space, we show that the existence of such a space establishes the theorem:

Proof of Lemma 4.1. The requirement on the VC dimension holds because a class of cardinality k_d can have VC dimension at most $\log(k_d)$, and

$$\log(k_d) = \log\left(\left(\frac{3n_d^{\sqrt{n_d}}}{\delta}\right)^3\right) \leq 6d\log\left(\frac{6d^2}{\delta}\right) = O(d\log(d)).$$

For the lower bound on the sample complexity, fix d large enough such that $\mathcal{H}_d$ enjoys Properties H1, H2 and H3, and assume for contradiction that there exists a verifier that 1-PAC verifies $\mathcal{H} = \mathcal{H}_d$ with accuracy ε and confidence $1-\delta$ using at most $t = t_d$ samples. Because $X_i \sim \mathcal{D}_{f_i}$ (Property P3), the assumption that V is a PAC learner entails that

$$\forall i \in \{1,2\}: \mathbb{P}\left[h_i = \text{reject} \vee \left(h_i \neq \text{reject} \wedge L_{\mathcal{D}_{f_i}}(h_i) < \varepsilon\right)\right] \geq 1-\delta. \tag{14}$$

Because $X_{\mathcal{U}}$ is uniform over $\mathcal{X}^t$ and $h_{\mathcal{U}} := [V(X_{\mathcal{U}},\rho_V), P_{\mathcal{U}}(X_P,\rho_P)]$ (by P2 and P7), the definition of $P_{\mathcal{U}}$ entails that

$$\mathbb{P}[h_{\mathcal{U}} \neq \text{reject}] \geq 1-\delta. \tag{15}$$

Next, because $\mathbb{P}[X_1 = X_{\mathcal{U}}] \geq 1-\delta$, $\mathbb{P}[X_1 = X_2] = 1$ and $h_i := [V(X_i,\rho_V), P_{\mathcal{U}}(X_P,\rho_P)]$ for $i \in \{1,2,\mathcal{U}\}$ (by P5, P4 and P7), it follows that with probability at least $1-\delta$ the view of V when computing h_1 and h_2 is identical to its view when computing $h_{\mathcal{U}}$, and so

$$\mathbb{P}[h_1 = h_2 = h_{\mathcal{U}}] \geq 1-\delta. \tag{16}$$

Combining Equations (15) and (16) yields

$$\mathbb{P}[h_1 = h_2 \neq \text{reject}] \geq 1-2\delta.$$

Together with Equation (14), this entails that

$$\mathbb{P}\left[(h_1 = h_2 \neq \text{reject}) \wedge \left(L_{\mathcal{D}_{f_1}}(h_1) < \varepsilon\right) \wedge \left(L_{\mathcal{D}_{f_2}}(h_2) < \varepsilon\right)\right] \geq 1-4\delta. \tag{17}$$

However, low loss of h_i with respect to $\mathcal{D}_{f_i}$ entails that the supports of h_i and f_i have a large intersection. Indeed, for all $i \in \{1,2\}$,

$$\begin{aligned}\varepsilon \geq L_{\mathcal{D}_{f_i}}(h_i) :&= \mathbb{P}_{x\sim\mathcal{D}_{f_i}}[h_i(x) \neq f_i(x)] = \sum_{x\in\mathcal{X}} \mathcal{D}_{f_i}(x)\cdot \mathbb{1}_{h_i\neq f_i}(x) \\ &= \sum_{x\in\mathrm{supp}(f_i)} \frac{2}{n}\cdot \mathbb{1}_{h_i\neq f_i}(x) = |\mathrm{supp}(f_i)\setminus\mathrm{supp}(h_i)|\cdot\frac{2}{n}.\end{aligned}$$

Thus,

$$|\mathrm{supp}(f_i)\setminus\mathrm{supp}(h_i)| \leq \frac{\varepsilon n}{2},$$

and so,

$$|\mathrm{supp}(f_i) \cap \mathrm{supp}(h_i)| = \frac{n}{2} - |\mathrm{supp}(f_i) \setminus \mathrm{supp}(h_i)| \geq \frac{n}{2} - \frac{\varepsilon n}{2},$$

Furthermore, because $h_1 = h_2$ the identity $|A \cap B| = |A| + |B| - |A \cup B|$ shows that the supports of f_1 and f_2 also have a large intersection:

$$\begin{aligned}
|\mathrm{supp}(f_1) \cap \mathrm{supp}(f_2)| &\geq |\mathrm{supp}(f_1) \cap \mathrm{supp}(f_2) \cap \mathrm{supp}(h_1)| \\
&= \Big|\mathrm{supp}(f_1) \cap \mathrm{supp}(h_1)\Big| + \Big|\mathrm{supp}(f_2) \cap \mathrm{supp}(h_2)\Big| \\
&\qquad - \Big|(\mathrm{supp}(f_1) \cap \mathrm{supp}(h_1)) \bigcup (\mathrm{supp}(f_2) \cap \mathrm{supp}(h_2))\Big| \\
&\geq \Big|\mathrm{supp}(f_1) \cap \mathrm{supp}(h_1)\Big| + \Big|\mathrm{supp}(f_2) \cap \mathrm{supp}(h_2)\Big| - \Big|\mathrm{supp}(h_1)\Big| \\
&\geq 2\left(\frac{n}{2} - \frac{\varepsilon n}{2}\right) - \frac{n}{2} \\
&\geq \frac{n}{2} - \varepsilon n.
\end{aligned}$$

That is, Equation (17) entails that

$$\mathbb{P}\left[|\mathrm{supp}(f_1) \cap \mathrm{supp}(f_2)| \geq \frac{n}{2} - \varepsilon n\right] \geq 1 - 4\delta.$$

In contrast, Property P8 states that

$$\mathbb{P}\left[|\mathrm{supp}(f_1) \cap \mathrm{supp}(f_2)| \leq \frac{3n}{8}\right] \geq 1 - \delta.$$

This is a contradiction whenever $\varepsilon < \frac{1}{8}$ and $\delta < \frac{1}{5}$. ■

4.4 Construction of $\mathcal{H}_d$

To complete the proof, we construct the probability space of Lemma 4.9. The first step is to show that for large enough values of d, a suitable class $\mathcal{H}_d$ can be constructed simply by choosing a set of k functions uniformly at random from $\mathcal{F}_{1/2}$.

Lemma 4.10. *Fix $\delta \in (0,1)$. The following holds for any value $d \in \mathbb{N}$ that is large enough. Let F denote a set of k_d functions chosen uniformly and independently from $\mathcal{F}_{d,\frac{1}{2}}$. Then with probability at least $1 - 3\delta$, F satisfies Properties H1, H2 and H3.*

The lemma follows immediately from Claims 4.11, 4.20 and 4.22 below, so the remainder of this section is devoted to stating and proving those claims.

4.4.1 Property H1: $d_{\mathsf{TV}}(\mathcal{U}_{\mathcal{X}'}, \mathcal{D}_{\mathcal{H}_d}) \leq \delta$

In this subsection we prove that the distribution $\mathcal{D}_F$ defined by F is close to the uniform distribution on $\mathcal{X}$ in the following sense.

Claim 4.11. *Fix $\delta \in (0,1)$. The following holds for all values of n that are large enough. Let*

$F = \{f_1, \ldots, f_k\}$ denote a set of functions chosen uniformly and independently from $\mathcal{F}_{1/2}$. If

$$k \geq \left(\frac{3n^{\sqrt{n}}}{\delta}\right)^3$$

and $t_d = \left\lfloor c_2 \cdot d \right\rfloor$ for

$$c_2 = \sqrt{\frac{\log(1-\delta/3)}{\log(1/2e)}}$$

then

$$\mathbb{P}_F \left[d_{\mathsf{TV}}\left(\mathcal{U}_{\mathcal{X}'}, \mathcal{D}_F\right) \leq \delta\right] \geq 1 - \delta.$$

The proof is partitioned to the following claims.

Claim 4.12. *For any integer $0 \leq s \leq n$, any set $X \subseteq \mathcal{X}$ of size s and any $z \in [n]$,*

$$\mathbb{P}_{f \in \{0,1\}^{\mathcal{X}}} \left[X \subseteq \operatorname{supp}(f) \,\middle|\, |\operatorname{supp}(f)| = z\right] = \frac{\binom{z}{s}}{\binom{n}{s}}.$$

Proof. If $z < s$ then the probability is clearly 0. Otherwise,

$$\begin{aligned}
\mathbb{P}_{f \in \{0,1\}^{\mathcal{X}}} \left[X \subseteq \operatorname{supp}(f) \,\middle|\, |\operatorname{supp}(f)| = z\right] &= \frac{\left|\{g \in \{0,1\}^{\mathcal{X}} : \ |\operatorname{supp}(g)| = z \wedge X \subseteq \operatorname{supp}(g)\}\right|}{|\{g \in \{0,1\}^{\mathcal{X}} : \ |\operatorname{supp}(g)| = z\}|} \\
&= \frac{\binom{n-s}{z-s}}{\binom{n}{z}} \\
&= \frac{\binom{n-s}{z-s}}{\binom{n}{z}} \cdot \frac{\binom{n}{s}}{\binom{n}{s}} \\
&\overset{(*)}{=} \frac{\binom{n}{z}\binom{z}{s}}{\binom{n}{z}\binom{n}{s}} \\
&= \frac{\binom{z}{s}}{\binom{n}{s}},
\end{aligned}$$

where $(*)$ follows from the identity $\binom{n}{s}\binom{n-s}{z-s} = \binom{n}{z}\binom{z}{s}$, which holds because both expressions count the number of ways to choose a committee of size z with a sub-committee of size s from a set of n candidates. ■

Corollary 4.13. *For any set $X \subseteq \mathcal{X}$ of size s,*

$$\mathbb{P}_{f \in \mathcal{F}_{1/2}} \left[X \subseteq \operatorname{supp}(f)\right] = \frac{\binom{\frac{n}{2}}{s}}{\binom{n}{s}}.$$

Notation 4.14. *For any $f \in \mathcal{F}_{1/2}$, we write $\mathcal{D}_f^{\text{distinct}}$ to denote the uniform distribution over tuples of length t that contain t distinct elements from* $\operatorname{supp}(f)$. *That is, for any $(x_1,\dots,x_t) \in \mathcal{X}^t$,*

$$\mathcal{D}_f^{\text{distinct}}((x_1,\dots,x_t)) = \begin{cases} \frac{1}{\binom{\frac{n}{2}}{t}\cdot t!} & x_1,\dots,x_t \in \operatorname{supp}(f) \ \wedge\ |\{x_1,\dots,x_t\}| = t \\ 0 & \text{o.w.} \end{cases}$$

Furthermore, let $\mathcal{U}_{\mathcal{X}^t}^{\text{distinct}}$ denote the uniform distribution over the set of tuples of length t from $\mathcal{X}$ with distinct elements,

$$\left\{(x_1,\dots,x_t) \in \mathcal{X}^t :\ |\{x_1,\dots,x_t\}| = t\right\}.$$

That is,

$$\mathcal{U}_{\mathcal{X}^t}^{\text{distinct}}((x_1,\dots,x_t)) = \begin{cases} \frac{1}{\binom{n}{t}\cdot t!} & x_1,\dots,x_t \in \mathcal{X} \ \wedge\ |\{x_1,\dots,x_t\}| = t \\ 0 & \text{o.w.} \end{cases}$$

Claim 4.15. *For any ordered tuple $X \in \mathcal{X}^t$ with distinct elements,*

$$\mathbb{E}_{f\in\mathcal{F}_{1/2}}\left[\mathcal{D}_f^{\text{distinct}}(X)\right] = \frac{1}{\binom{n}{t}t!}.$$

Proof. Using Corollary 4.13,

$$\begin{aligned}\mathbb{E}_{f\in\mathcal{F}_{1/2}}\left[\mathcal{D}_f^{\text{distinct}}(X)\right] &= \mathbb{P}\left[X \subseteq \operatorname{supp}(f)\right]\cdot\frac{1}{\binom{\frac{n}{2}}{t}t!} + \mathbb{P}_{f\in\mathcal{F}_{1/2}}\left[X \nsubseteq \operatorname{supp}(f)\right]\cdot 0 \\ &= \frac{\binom{\frac{n}{2}}{t}}{\binom{n}{t}}\cdot\frac{1}{\binom{\frac{n}{2}}{t}t!} \\ &= \frac{1}{\binom{n}{t}t!}. \ \blacksquare\end{aligned}$$

Claim 4.16. *Consider k functions $f_1,\dots,f_k$ chosen independently and uniformly at random from $\mathcal{F}_{1/2}$. For any $\delta \in (0,1)$ and ordered tuple $X \in \mathcal{X}^t$ with distinct elements, if*

$$k \geq \left(\frac{n^{\sqrt{n}}}{\delta}\right)^3$$

then

$$\mathbb{P}_{f_1,\dots,f_k\in\mathcal{F}_{1/2}}\left[\left|\mathcal{U}_{\mathcal{X}^t}^{\text{distinct}}(X) - \frac{1}{k}\sum_{i=1}^{k}\mathcal{D}_{f_i}^{\text{distinct}}(X)\right| > \frac{\delta}{\binom{n}{t}t!}\right] \leq \frac{\delta}{\binom{n}{t}t!}.$$

Proof. Fix X. Observe that when $f_1,\dots,f_k$ are chosen independently and uniformly then $\left\{\mathcal{D}_{f_i}^{\text{distinct}}(X)\right\}_{i\in[k]}$ is a set of i.i.d. random variables each of which takes values in $[0,1]$. Furthermore, from Claim 4.15 the expectation of each of these random variables is $\mathcal{U}_{\mathcal{X}^t}^{\text{distinct}}(X)$.

Thus, from Hoeffding's inequality, the left hand side in the claim is at most

$$2\exp\left(-2k\left(\frac{\delta}{\binom{n}{t}t!}\right)^2\right),$$

and so taking

$$k \geq \frac{1}{2}\left(\frac{\binom{n}{t}t!}{\delta}\right)^2 \log\left(\frac{2\binom{n}{t}t!}{\delta}\right)$$

is sufficient to obtain the desired bound. A direct calculation shows that

$$\begin{aligned}
\frac{1}{2}\left(\frac{\binom{n}{t}t!}{\delta}\right)^2 \log\left(\frac{2\binom{n}{t}t!}{\delta}\right) &\leq \left(\frac{\binom{n}{t}t!}{\delta}\right)^3 \\
&\leq \left(\frac{\binom{n}{\sqrt{n}}\sqrt{n}!}{\delta}\right)^3 \\
&= \left(\frac{n(n-1)\cdots(n-\sqrt{n}+1)}{\delta}\right)^3 \\
&\leq \left(\frac{n^{\sqrt{n}}}{\delta}\right)^3,
\end{aligned}$$

as desired. ■

Notation 4.17. *For any $F = \{f_1, \ldots, f_k\} \subseteq \mathcal{F}_{1/2}$, we write $\mathcal{D}_F^{\text{distinct}}$ to denote the distribution over $\mathcal{X}^t$ given by*

$$\mathcal{D}_F^{\text{distinct}}(x_1, \ldots, x_t) := \frac{1}{k}\sum_{i=1}^{k} \mathcal{D}_{f_i}^{\text{distinct}}(x_1, \ldots, x_t).$$

Claim 4.18. *Let $F = \{f_1, \ldots, f_k\}$ denote a set of functions chosen uniformly and independently from $\mathcal{F}_{1/2}$. For any $\delta \in (0,1)$, if*

$$k \geq \left(\frac{3n^{\sqrt{n}}}{\delta}\right)^3$$

then

$$\mathbb{P}_F\left[d_{\mathsf{TV}}\left(\mathcal{U}_{\mathcal{X}^t}^{\text{distinct}}, \mathcal{D}_F^{\text{distinct}}\right) \leq \frac{\delta}{3}\right] \geq 1 - \frac{\delta}{3}.$$

Proof. From Claim 4.16, taking k as in the statement ensures that for any particular tuple $X \in \mathcal{X}^t$ with distinct elements,

$$\mathbb{P}_{f_1, \ldots, f_k \in \mathcal{F}_{1/2}}\left[\left|\mathcal{U}_{\mathcal{X}^t}^{\text{distinct}}(X) - \mathcal{D}_F^{\text{distinct}}(X)\right| > \frac{\delta}{3\binom{n}{t}t!}\right] \leq \frac{\delta}{3\binom{n}{t}t!}.$$

From the union bound, we conclude that with probability at least $1-\frac{\delta}{3}$, the inequality

$$\left|\mathcal{U}^{\text{distinct}}_{\mathcal{X}^t}(X)-\mathcal{D}^{\text{distinct}}_{F}(X)\right| \le \frac{\delta}{3\binom{n}{t}t!}$$

holds for all $\binom{n}{t}t!$ such tuples simultaneously. In this case,

$$d_{\mathsf{TV}}\left(\mathcal{U}^{\text{distinct}}_{\mathcal{X}^t},\mathcal{D}^{\text{distinct}}_{F}\right) = \frac{1}{2}\sum_{X\in\mathcal{X}^t}\left|\mathcal{U}^{\text{distinct}}_{\mathcal{X}^t}(X)-\mathcal{D}^{\text{distinct}}_{F}(X)\right| \le \frac{\delta}{6}. \blacksquare$$

Proof of Claim 4.11. From the triangle inequality

$$d_{\mathsf{TV}}\left(\mathcal{U}_{\mathcal{X}^t},\mathcal{D}_F\right) \le d_{\mathsf{TV}}\left(\mathcal{U}_{\mathcal{X}^t},\mathcal{U}^{\text{distinct}}_{\mathcal{X}^t}\right)+d_{\mathsf{TV}}\left(\mathcal{U}^{\text{distinct}}_{\mathcal{X}^t},\mathcal{D}^{\text{distinct}}_{F}\right)+d_{\mathsf{TV}}\left(\mathcal{D}^{\text{distinct}}_{F},\mathcal{D}_F\right).$$

Therefore, it suffices to show the following three inequalities:

(*i*) $d_{\mathsf{TV}}\left(\mathcal{U}_{\mathcal{X}^t},\mathcal{U}^{\text{distinct}}_{\mathcal{X}^t}\right) \le \frac{\delta}{3}$ for n large enough. Indeed,

$$\begin{aligned}
d_{\mathsf{TV}}\left(\mathcal{U}_{\mathcal{X}^t},\mathcal{U}^{\text{distinct}}_{\mathcal{X}^t}\right) &= \max_{A\subseteq\mathcal{X}^t}\left(\mathcal{U}_{\mathcal{X}^t}(A)-\mathcal{U}^{\text{distinct}}_{\mathcal{X}^t}(A)\right)\\
&= \sum_{(x_1,\dots,x_t)\in\mathcal{X}^t:\ |\{x_1,\dots,x_t\}|<t}\left(\frac{1}{n^t}-0\right)\\
&= \left(n^t-\binom{n}{t}t!\right)\frac{1}{n^t}\\
&= 1-\frac{n(n-1)\cdots(n-t+1)}{n^t}\\
&\le 1-\left(1-\frac{t}{n}\right)^t\\
&\le 1-\left(1-\frac{(c_2\sqrt{n})}{n}\right)^{c_2\sqrt{n}}\\
&= 1-\left(1-\frac{c_2}{\sqrt{n}}\right)^{\frac{\sqrt{n}}{c_2}\cdot c_2^2}\\
&\overset{(*)}{\le} 1-\left(\frac{1}{2e}\right)^{c_2^2}\\
&\overset{(**)}{\le} \frac{\delta}{3},
\end{aligned}$$

where $(*)$ holds for all n large enough because $\left(1-\frac{c_2}{\sqrt{n}}\right)^{\frac{\sqrt{n}}{c_2}} \xrightarrow{n\to\infty} \frac{1}{e}$ from below, and $(**)$ holds whenever

$$c_2 \le \sqrt{\frac{\log(1-\delta/3)}{\log(1/2e)}}.$$

(*ii*) $\mathbb{P}_F\left[d_{\mathsf{TV}}\left(\mathcal{U}^{\text{distinct}}_{\mathcal{X}^t},\mathcal{D}^{\text{distinct}}_{F}\right) > \frac{\delta}{3}\right] \le \delta$. This is true by Claim 4.18.

(*iii*) $d_{\mathsf{TV}}\left(\mathcal{D}_F^{\text{distinct}}, \mathcal{D}_F\right) \le \frac{\delta}{3}$ or n large enough. This follows from a calculation very similar to (*i*).

We conclude that for n large enough, with probability at least $1-\delta$ over the choice of F,

$$d_{\mathsf{TV}}\left(\mathcal{U}_{\mathcal{X}^t}, \mathcal{D}_F\right) \le \delta,$$

as desired. ■

4.4.2 Property H2: $\forall i \neq j: \ |\text{supp}(f_i) \cap \text{supp}(f_j)| \le \frac{3n}{8}$

In this section we show that random sets typically form a code.

Claim 4.19. $\mathbb{P}_{f_1, f_2 \in \mathcal{F}_{1/2}}\left[|\text{supp}(f_1) \cap \text{supp}(f_2)| > \frac{3n}{8}\right] \le \frac{\delta}{k^2}$.

Proof. Let $\text{supp}(f_2) = \{x_1, \dots, x_{n/2}\}$. We think of this experiment as if f_1 is chosen first, and then we count how many members of $\text{supp}(f_2)$ fall inside $\text{supp}(f_1)$. The expected number of hits is $\frac{n}{4}$, and they are independent, so we can use Hoeffding's bound to prove the claim.

$$\begin{aligned}
\mathbb{P}_{f_1, f_2 \in \mathcal{F}_{1/2}}\left[|\text{supp}(f_1) \cap \text{supp}(f_2)| > \frac{3n}{8}\right] &\le \mathbb{P}_{f_1, f_2 \in \mathcal{F}_{1/2}}\left[\sum_{i=1}^{n/2} \mathbb{1}(x_i \in \text{supp}(f_1)) > \frac{3n}{8}\right] \\
&= \mathbb{P}_{f_1, f_2 \in \mathcal{F}_{1/2}}\left[\frac{2}{n}\sum_{i=1}^{n/2} \mathbb{1}(x_i \in \text{supp}(f_1)) > \frac{3}{4}\right] \\
&\le \mathbb{P}_{f_1, f_2 \in \mathcal{F}_{1/2}}\left[\left|\frac{2}{n}\sum_{i=1}^{n/2} \mathbb{1}(x_i \in \text{supp}(f_1)) - \frac{1}{2}\right| > \frac{1}{4}\right] \\
&\le 2\exp\left(-2 \cdot \frac{n}{2} \cdot \left(\frac{1}{4}\right)^2\right) = 2^{\Theta(-n)}.
\end{aligned}$$

In contrast, considering δ to be a constant, it holds that

$$\frac{\delta}{k^2} = 2^{\Theta\left(-\log(n)\sqrt{n}\right)},$$

and so for n large enough we obtain $\mathbb{P}_{f_1, f_2 \in \mathcal{F}_{1/2}}\left[|\text{supp}(f_1) \cap \text{supp}(f_2)| > \frac{3n}{8}\right] \le \frac{\delta}{k^2}$, as desired. ■

Claim 4.20. $\mathbb{P}_{f_1, \dots, f_k \in \mathcal{F}_{1/2}}\left[\forall i \neq j \in [k]: \ |\text{supp}(f_i) \cap \text{supp}(f_j)| \le \frac{3n}{8}\right] \ge 1 - \delta$.

Proof.

$$\begin{aligned}
&\mathbb{P}_{f_1,\ldots,f_k\in\mathcal{F}_{1/2}}\left[\forall i \neq j \in [k]:\ |\mathrm{supp}(f_i)\cap\mathrm{supp}(f_j)| \leq \frac{3n}{8}\right] \\
&= 1-\mathbb{P}\left[\bigcup_{i\neq j}\left\{|\mathrm{supp}(f_i)\cap\mathrm{supp}(f_j)| > \frac{3n}{8}\right\}\right] \\
&\geq 1-\sum_{i\neq j}\mathbb{P}\left[|\mathrm{supp}(f_i)\cap\mathrm{supp}(f_j)| > \frac{3n}{8}\right] \\
&\geq 1-k^2\cdot\frac{\delta}{k^2} = 1-\delta.
\end{aligned}$$

where the last inequality follows from Claim 4.19. ■

4.4.3 Property H3: $|F_X| \geq \frac{1}{\delta}$

In this section we show that there are typically many sets that contain a given subset of size order $\sqrt{n}$.

Notation 4.21. *Let $F \subseteq \mathcal{F}_{1/2}$, and let $X \subseteq \mathcal{X}$. We write F_X to denote the set*

$$\{f \in F:\ X \subseteq \mathrm{supp}(f)\}.$$

Claim 4.22. *Fix $\delta \in (0,1)$. Let $F = \{f_1,\ldots,f_k\}$ denote a set of functions chosen uniformly and independently from $\mathcal{F}_{1/2}$. There exists N_0 such that for all $n \geq N_0$, if*

$$k \geq \left(\frac{n^{\sqrt{n}}}{\delta}\right)^3$$

then with probability at least $1-\delta$ over the choice of F, all subsets $X \subseteq \mathcal{X}$ of size at most $\sqrt{n}$ satisfy

$$|F_X| \geq \frac{1}{\delta}.$$

Proof of Claim 4.22. Let $X \subseteq \mathcal{X}$ such that $|X| = t$. From Corollary 4.13,

$$\begin{aligned}
\mathbb{P}_{f \in \mathcal{F}_{1/2}}[X \subseteq \operatorname{supp}(f)] &= \frac{\binom{\frac{n}{2}}{t}}{\binom{n}{t}} = \frac{\frac{n}{2}!}{(\frac{n}{2}-t)!t!} \cdot \frac{(n-t)!t!}{n!} \\
&= \frac{n-t}{n} \cdot \frac{n-t-1}{(n-1)} \cdots \frac{\frac{n}{2}-t+1}{\frac{n}{2}+1} \\
&= \frac{\frac{n}{2}}{n} \cdot \frac{\frac{n}{2}-1}{(n-1)} \cdots \frac{\frac{n}{2}-t+1}{n-t+1} \\
&\geq \left(\frac{\frac{n}{2}-t}{n}\right)^{t} \\
&\geq \left(\frac{\frac{n}{2}-\sqrt{n}}{n}\right)^{\sqrt{n}} \\
&= \left(\frac{1}{2}-\frac{1}{\sqrt{n}}\right)^{\sqrt{n}} \geq 4^{-\sqrt{n}}.
\end{aligned}$$

where the last inequality holds for $n \geq 16$. Observe that

$$\mu := \mathbb{E}_{f_1,\dots,f_k \in \mathcal{F}_{1/2}}[|F_X|] \geq k \cdot 4^{-\sqrt{n}} \geq 2^{\log(n)\sqrt{n}-2\sqrt{n}} \xrightarrow{n\to\infty} \infty,$$

and choose N_0 large enough such that for all $n \geq N_0$, $\mathbb{E}[|F_X|] \geq \frac{2}{\delta}$.

Now, for any $n \geq N_0$ and any set X of size t, Hoeffding's inequality entails

$$\begin{aligned}
\mathbb{P}_{f_1,\dots,f_k \in \mathcal{F}_{1/2}}\left[|F_X| \leq \frac{1}{\delta}\right] &\leq \mathbb{P}\left[\Big||F_X| - \mu\Big| \geq \frac{k \cdot 4^{-\sqrt{n}}}{2}\right] \\
&= \mathbb{P}\left[\Big|\frac{1}{k}\sum_{i=1}^{k} \mathbb{1}(X \subseteq \operatorname{supp}(f_i)) - \frac{\mu}{k}\Big| \geq \frac{4^{-\sqrt{n}}}{2}\right] \\
&\leq 2\exp\left(-2k\left(\frac{4^{-\sqrt{n}}}{2}\right)^{2}\right).
\end{aligned}$$

Hence, taking

$$k \geq \frac{1}{2} \cdot 4^{2\sqrt{n}+1} \cdot \log\left(\frac{2n^{\sqrt{n}}}{\delta}\right)$$

is sufficient to ensure that

$$\forall X \in \binom{\mathcal{X}}{t} : \mathbb{P}_{f_1,\dots,f_k \in \mathcal{F}_{1/2}}\left[|F_X| \leq \frac{1}{\delta}\right] \leq \frac{\delta}{n^{\sqrt{n}}}$$

Taking k as in the claim is therefore more than sufficient to this end. Seeing as there exist less than $n^{\sqrt{n}}$ such sets, the union bound yields that

$$\mathbb{P}_{f_1,\dots,f_k \in \mathcal{F}_{1/2}}\left[\forall X \subseteq \mathcal{X} \text{ s.t. } |X| \leq t : \ |F_X| \geq \frac{1}{\delta}\right] \geq 1 - \delta.$$

Note that for the case $|X| < t$ in the previous line, we have used the facts that X is contained in some set of size precisely t, and that $|F_X|$ is monotone decreasing with the cardinality of X. ■

4.5 Construction of the Joint Probability Space

Assume $\mathcal{H}_d$ is a class that satisfies Properties H1, H2, and H3. We show how to use these properties to construct a joint probability space that satisfies Properties P1–8, proving Lemma 4.9.

The construction is as follows:

1. X_P is sampled uniformly from $\mathcal{X}^{t_P}$.
2. A function f_1 is chosen uniformly from $\mathcal{H}_d$.
3. $X_1 = (x_1, \ldots, x_t)$ is sampled i.i.d. from D_{f_1}.
4. X_2 is set to be equal to X_1.
5. A function f_2 is chosen uniformly from $\{f \in \mathcal{H}_d :\ X_2 \subseteq \mathrm{supp}(f)\}$.
6. $X_{\mathcal{U}} = (x_1^{\mathcal{U}}, \ldots, x_t^{\mathcal{U}})$ is sampled such that its marginal distribution is uniform over $(\mathcal{X}_d)^t$, and also $\mathbb{P}[X_{\mathcal{U}} = X_1] \geq 1 - \delta$. This is possible due to Property H1 of the class $\mathcal{H}_d$.
7. ρ_V and ρ_P are sampled from the distributions of randomness used by V and $P_{\mathcal{U}}$ respectively, independently of each other and of everything else.
8. For $\alpha \in \{1, 2, \mathcal{U}\}$, compute $h_\alpha := [V(X_\alpha, \rho_V), P_{\mathcal{U}}(X_P, \rho_P)]$.

Note that Properties P1, P2, P4, P5, P6 and P7 are satisfied immediately by the construction, as is Property P3 for the case of $i = 1$. Property P8 is immediate from the construction together with H2 and H3. Hence, to prove the correctness of the construction, it suffices to prove that Property P3 holds also for the case $i = 2$, as in the following claim.

Claim 4.23. *The constriction in Section 4.5 satisfies that $X_2 \sim D_{f_2}$. More formally, for any $g \in \mathcal{H}_d$ and $x_1, \ldots, x_t \in \mathcal{X}$,*

$$\mathbb{P}[X_2 = (x_1, \ldots, x_t) \mid f_2 = g] = \mathcal{D}_g((x_1, \ldots, x_t)).$$

Proof. By construction, $X_1 \sim D_{f_1}$. Hence, it is sufficient to show that

$$(X_1, f_1) \overset{d}{=} (X_2, f_2),$$

where $\overset{d}{=}$ denotes equality in distribution. Indeed, conditioned on $X_1 = X_2 = x$, both f_1 and f_2 are chosen i.i.d. uniformly in

$$F_x := \{f \in \mathcal{H}_d :\ x \subseteq \mathrm{supp}(f)\}.$$

More formally, for any $g \in \mathcal{H}_d$ and $x \in \mathcal{X}^t$,

- If $x \subseteq \mathrm{supp}(g)$ then

$$\begin{aligned}\mathbb{P}[f_1 = g \mid X_1 = x] &= \frac{\mathbb{P}[X_1 = x \mid f_1 = g]\,\mathbb{P}[f_1 = g]}{\mathbb{P}[X_1 = x]} \\ &= \frac{\mathbb{P}[X_1 = x \mid f_1 = g]\,\mathbb{P}[f_1 = g]}{\sum_{g' \in F_x} \mathbb{P}[X_1 = x \mid f_1 = g']\,\mathbb{P}[f_1 = g']} \\ &= \frac{\mathbb{P}[X_1 = x \mid f_1 = g]}{\sum_{g' \in F_x} \mathbb{P}[X_1 = x \mid f_1 = g']} \\ &= \frac{1}{|F_x|} = \mathbb{P}[f_2 = g \mid X_2 = x]\,.\end{aligned}$$

- Otherwise, if $x \nsubseteq \mathrm{supp}(g)$ then

$$\mathbb{P}[f_1 = g \mid X_1 = x] = 0 = \mathbb{P}[f_2 = g \mid X_2 = x]\,.$$

That is, for any $g \in \mathcal{H}_d$ and $x \in \mathcal{X}^t$,

$$\begin{aligned}&\mathbb{P}[f_1 = g \ \wedge \ X_1 = x] = \mathbb{P}[f_1 = g \mid X_1 = x]\,\mathbb{P}[X_1 = x] = \\ &\qquad = \mathbb{P}[f_2 = g \mid X_2 = x]\,\mathbb{P}[X_2 = x] = \mathbb{P}[f_2 = g \ \wedge \ X_2 = x]\,. \blacksquare\end{aligned}$$

This proves Lemma 4.9, thereby concluding our proof of Lemma 4.1.

5 Directions for Future Work

This work initializes the study of verification in the context of machine learning. We have seen separations between the sample complexity of verification versus learning and testing, an algorithm that uses interaction to efficiently learn sparse boolean functions, and have seen that in some cases the sample complexities of verification and learning are the same.

Building a theory that can help guide verification procedures is a main objective for future research. A specific approach is to identify dimension-like quantities that describe the sample complexity of verification, similarly to role VC dimension plays in characterizing learnability. A different approach is to understand the trade-offs between the various resources in the system – the amount of time, space and samples used by the prover and the verifier, as well as the amount of interaction between the parties.

From a practical perspective, we described potential applications for delegation and safety of machine learning, and for verification of experimental data. It seems beneficial to build efficient verification protocols for machine learning problems that are commonly used in practice. This would have commercial and scientific applications.

There are also some technical improvements that we find interesting. For example, is there a simple way to improve the NP-like protocol for the multi-thresholds class $\mathcal{T}_d$ to achieve 1-PAC verification (instead of 2-PAC verification)?

Finally, one can also consider variations of the settings we investigated here. One case has $\mathcal{O}_V$

and $\mathcal{O}_P$ providing i.i.d. sample access to different distributions, $\mathcal{D}_V$ and $\mathcal{D}_P$ respectively, where $\mathcal{D}_P$ has better quality data in some sense. For instance, for some target function f it might be the case that

$$\mathbb{P}_{(x,y)\sim\mathcal{D}_V}[y = f(x)] < \mathbb{P}_{(x,y)\sim\mathcal{D}_P}[y = f(x)] .$$

Can a prover who has access to $\mathcal{D}_P$ efficiently provide an advantage to the verifier? Alternatively, it might be the case that $\mathcal{D}_P$ provides data with "higher resolution" than $\mathcal{D}_V$ (i.e., the σ-algebra of $\mathcal{D}_V$ is a sub-σ-algebra of that of $\mathcal{D}_P$). One can also consider verification of other types of learning, such as clustering, parameter estimation and reinforcement learning.

Acknowledgements

SG would like to thank Constantinos Daskalakis and Ankur Moitra for insightful correspondence and helpful references. GR would like to thank Oded Goldreich for insightful discussions and helpful references. JS would like to thank Orr Paradise, Ido Nachum, Avishay Tal, Alessandro Chiesa, and attendants of the UC Berkeley theory group's TGIF for insightful discussions.

SG's research was supported in part by DARPA under Contract No. HR001120C0015. GR's research was supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 819702). AY's research was supported by ISF grant No. 1162/15.

References

Alon, N., & Spencer, J. H. (2000). *The probabilistic method* (Second ed.). John Wiley & Sons.

Angluin, D. (1987). Learning regular sets from queries and counterexamples. *Inf. Comput.*, *75*(2), 87–106. Retrieved from https://doi.org/10.1016/0890-5401(87)90052-6 doi: 10.1016/0890-5401(87)90052-6

Axelrod, B., Garg, S., Sharan, V., & Valiant, G. (2019). Sample amplification: Increasing dataset size even when learning is impossible. *arXiv preprint arXiv:1904.12053*.

Babai, L., Fortnow, L., Levin, L. A., & Szegedy, M. (1991). Checking computations in polylogarithmic time. In *Proceedings of the 23rd annual ACM symposium on theory of computing, may 5-8, 1991, new orleans, louisiana, USA* (pp. 21–31). Retrieved from https://doi.org/10.1145/103418.103428 doi: 10.1145/103418.103428

Balcan, M., Blais, E., Blum, A., & Yang, L. (2012). Active property testing. In *53rd annual IEEE symposium on foundations of computer science, FOCS 2012, new brunswick, nj, usa, october 20-23, 2012* (pp. 21–30). IEEE Computer Society. Retrieved from https://doi.org/10.1109/FOCS.2012.64 doi: 10.1109/FOCS.2012.64

Batu, T., Fischer, E., Fortnow, L., Kumar, R., Rubinfeld, R., & White, P. (2001). Testing random variables for independence and identity. In *Proceedings 42nd ieee symposium on foundations of computer science* (pp. 442–451).

Blum, A., & Hu, L. (2018). Active tolerant testing. In S. Bubeck, V. Perchet, & P. Rigollet (Eds.), *Conference on learning theory, COLT 2018, stockholm, sweden, 6-9 july 2018* (Vol. 75, pp. 474–497). PMLR. Retrieved from http://proceedings.mlr.press/v75/blum18a.html

Blum, A., Kalai, A., & Wasserman, H. (2003). Noise-tolerant learning, the parity problem, and the statistical query model. *Journal of the ACM (JACM)*, *50*(4), 506–519.

Blumer, A., Ehrenfeucht, A., Haussler, D., & Warmuth, M. K. (1989). Learnability and the Vapnik-Chervonenkis dimension. *Journal of the ACM (JACM)*, *36*(4), 929–965.

Bostrom, N. (2017). *Superintelligence*. Dunod.

Canonne, C. L. (2015). A survey on distribution testing: Your data is big. but is it blue? *Electronic Colloquium on Computational Complexity (ECCC)*, *22*, 63. Retrieved from http://eccc.hpi-web.de/report/2015/063

Charikar, M., Steinhardt, J., & Valiant, G. (2017). Learning from untrusted data. In *Proceedings of the 49th annual acm sigact symposium on theory of computing* (pp. 47–60).

Chiesa, A., & Gur, T. (2018). Proofs of proximity for distribution testing. In *9th innovations in theoretical computer science conference, ITCS 2018, january 11-14, 2018, cambridge, ma, USA* (pp. 53:1–53:14). Retrieved from https://doi.org/10.4230/LIPIcs.ITCS.2018.53 doi: 10.4230/LIPIcs.ITCS.2018.53

Daskalakis, C., Gouleakis, T., Tzamos, C., & Zampetakis, M. (2018). Efficient statistics, in high dimensions, from truncated samples. In *2018 ieee 59th annual symposium on foundations of computer science (focs)* (pp. 639–649).

Diakonikolas, I., Kamath, G., Kane, D., Li, J., Moitra, A., & Stewart, A. (2019). Robust estimators in high-dimensions without the computational intractability. *SIAM Journal on Computing*, *48*(2), 742–864.

Diakonikolas, I., Kamath, G., Kane, D. M., Li, J., Moitra, A., & Stewart, A. (2018). Robustly learning a gaussian: Getting optimal error, efficiently. In *Proceedings of the twenty-ninth annual acm-siam symposium on discrete algorithms* (pp. 2683–2702).

Diakonikolas, I., Kane, D. M., & Stewart, A. (2017). Statistical query lower bounds for robust estimation of high-dimensional gaussians and gaussian mixtures. In *2017 ieee 58th annual symposium on foundations of computer science (focs)* (pp. 73–84).

Ergün, F., Kumar, R., & Rubinfeld, R. (2004). Fast approximate probabilistically checkable proofs. *Inf. Comput.*, *189*(2), 135–159. Retrieved from https://doi.org/10.1016/j.ic.2003.09.005 doi: 10.1016/j.ic.2003.09.005

Fidler, F., & Wilcox, J. (2018). Reproducibility of scientific results. In E. N. Zalta (Ed.), *The stanford encyclopedia of philosophy* (Winter 2018 ed.). Metaphysics Research Lab, Stanford University. https://plato.stanford.edu/archives/win2018/entries/scientific-reproducibility/.

Goldreich, O. (2007). *Foundations of cryptography: volume 1, basic tools*. Cambridge university press.

Goldreich, O., Goldwasser, S., & Ron, D. (1998). Property testing and its connection to learning and approximation. *J. ACM*, *45*(4), 653–750. Retrieved from https://doi.org/10.1145/

285055.285060 doi: 10.1145/285055.285060

Goldreich, O., & Levin, L. A. (1989). A hard-core predicate for all one-way functions. In *Proceedings of the twenty-first annual acm symposium on theory of computing* (pp. 25–32).

Goldwasser, S., Kalai, Y. T., & Rothblum, G. N. (2015). Delegating computation: Interactive proofs for muggles. *J. ACM*, *62*(4), 27:1–27:64. Retrieved from https://doi.org/10.1145/2699436 doi: 10.1145/2699436

Goldwasser, S., Micali, S., & Rackoff, C. (1989). The knowledge complexity of interactive proof systems. *SIAM Journal on computing*, *18*(1), 186–208.

Ilyas, A., Jalal, A., Asteri, E., Daskalakis, C., & Dimakis, A. G. (2017). The robust manifold defense: Adversarial training using generative models. *arXiv preprint arXiv:1712.09196*.

Ioannidis, J. P. (2005). Why most published research findings are false. *PLoS medicine*, *2*(8), e124.

Kearns, M. J., & Ron, D. (2000). Testing problems with sublearning sample complexity. *J. Comput. Syst. Sci.*, *61*(3), 428–456. Retrieved from https://doi.org/10.1006/jcss.1999.1656 doi: 10.1006/jcss.1999.1656

Kushilevitz, E., & Mansour, Y. (1993). Learning decision trees using the fourier spectrum. *SIAM Journal on Computing*, *22*(6), 1331–1348.

Linial, N., Mansour, Y., & Nisan, N. (1993). Constant depth circuits, fourier transform, and learnability. *Journal of the ACM (JACM)*, *40*(3), 607–620.

Mansour, Y. (1994). Learning boolean functions via the fourier transform. In *Theoretical advances in neural computation and learning* (pp. 391–424). Springer.

Micali, S. (1994). CS proofs (extended abstracts). In *35th annual symposium on foundations of computer science, santa fe, new mexico, usa, 20-22 november 1994* (pp. 436–453). Retrieved from https://doi.org/10.1109/SFCS.1994.365746 doi: 10.1109/SFCS.1994.365746

O'Donnell, R. (2014). *Analysis of boolean functions*. Cambridge University Press.

Parnas, M., Ron, D., & Rubinfeld, R. (2006). Tolerant property testing and distance approximation. *J. Comput. Syst. Sci.*, *72*(6), 1012–1042. Retrieved from https://doi.org/10.1016/j.jcss.2006.03.002 doi: 10.1016/j.jcss.2006.03.002

Pashler, H., & Wagenmakers, E.-J. (2012). Editors' introduction to the special section on replicability in psychological science: A crisis of confidence? *Perspectives on Psychological Science*, *7*(6), 528–530.

Pinker, S. (2018). *Enlightenment now: The case for reason, science, humanism, and progress*. Viking.

Raskhodnikova, S., Ron, D., Shpilka, A., & Smith, A. D. (2009). Strong lower bounds for approximating distribution support size and the distinct elements problem. *SIAM J. Comput.*, *39*(3), 813–842. Retrieved from https://doi.org/10.1137/070701649 doi: 10.1137/070701649

Ron, D., & Tsur, G. (2013). On approximating the number of relevant variables in a function. *TOCT*, *5*(2), 7:1–7:19. Retrieved from https://doi.org/10.1145/2493246.2493250 doi: 10.1145/2493246.2493250

Rothblum, G. N., Vadhan, S. P., & Wigderson, A. (2013). Interactive proofs of proximity: delegating computation in sublinear time. In *Symposium on theory of computing conference, stoc'13, palo alto, ca, usa, june 1-4, 2013* (pp. 793–802). Retrieved from https://doi.org/10.1145/2488608.2488709 doi: 10.1145/2488608.2488709

Russell, S. (2019). *Human compatible*. Viking.

Shalev-Shwartz, S., & Ben-David, S. (2014). *Understanding machine learning: From theory to algorithms*. Cambridge university press.

Soares, N. (2015). Aligning superintelligence with human interests: An annotated bibliography. *Intelligence*, *17*(4), 391–444.

Taylor, J., Yudkowsky, E., LaVictoire, P., & Critch, A. (2016). Alignment for advanced machine learning systems. *Machine Intelligence Research Institute*.

Valiant, G. (2012). *Algorithmic approaches to statistical questions* (Unpublished doctoral dissertation). UC Berkeley.

Valiant, G., & Valiant, P. (2010a). A CLT and tight lower bounds for estimating entropy. *Electronic Colloquium on Computational Complexity (ECCC)*, *17*, 179. Retrieved from http://eccc.hpi-web.de/report/2010/179

Valiant, G., & Valiant, P. (2010b). Estimating the unseen: A sublinear-sample canonical estimator of distributions. *Electronic Colloquium on Computational Complexity (ECCC)*, *17*, 180. Retrieved from http://eccc.hpi-web.de/report/2010/180

Valiant, L. G. (1984). A theory of the learnable. *Communications of the ACM*, *27*(11), 1134–1142.

Valiant, P. (2011). Testing symmetric properties of distributions. *SIAM J. Comput.*, *40*(6), 1927–1968. Retrieved from https://doi.org/10.1137/080734066 doi: 10.1137/080734066

Vapnik, V., & Chervonenkis, A. Y. (1971). On the uniform convergence of relative frequencies of events to their probabilities. *Measures of Complexity*, *16*(2), 11.

Walfish, M., & Blumberg, A. J. (2015). Verifying computations without reexecuting them. *Commun. ACM*, *58*(2), 74–84. Retrieved from https://doi.org/10.1145/2641562 doi: 10.1145/2641562

Yu, Y., & Steinberger, J. (2016). Pseudorandom functions in almost constant depth from low-noise lpn. In *Annual international conference on the theory and applications of cryptographic techniques* (pp. 154–183).

Appendices

A Towards a Formal Theory of AI Safety

Another motivation for this work comes from the fledgling field of AI safety, concerned with ensuring that AI systems will not cause harm to their operators and to humanity in general (Bostrom, 2017). More specifically, the *value alignment problem* (Taylor, Yudkowsky, LaVictoire, & Critch, 2016; Soares, 2015), asks:

Question (value alignment). *How can the designers of an AI system ensure that the utility function being maximized by the system is well aligned with the designers' utility function?*

Consider this salient example, due to Russell (2019, ch. 5): An AI system is tasked with finding a cure for cancer. If the system is smart enough, it might decide to forcefully induce cancer tumors into millions of living people as part of its R&D efforts; furthermore, it could easily anticipate and stifle any human attempts at resistance. Thus, the system may accomplish its mission of identifying a cure for cancer, and still be a monstrous disaster.[17]

The point is that formulating an exhaustive and foolproof description of human preferences is a difficult task, and a sufficiently intelligent AI is likely to find loopholes that were not anticipated by human designers. A more promising approach might be to have the system *learn* the preferences of humans. But this leads to the problem of ensuring that preferences that were learned by the system actually align well with human preferences – without first having a clear formulation of what the human preferences are!

As a solution, we suggest that by employing an interactive proof system, an AI development project could formally prove that a proposed AI design will be well-aligned with the desired human utility function *before* activating the AI, and *without* needing to formally specify precisely what the desired human utility function is. As an informal illustration, consider the following: Let $\mathcal{H}$ be a class of possible AI policies, and let $L\colon \mathcal{H} \to \mathbb{R}^+$ be a loss function representing how "bad" a policy is with respect to human preferences. The human designers do not have a formal specification of L, yet they can estimate its value at a small number of points, denoted by S. For instance they can distributing questionnaires to large number of people, asking them to rate how good or bad a specific hypothetical AI behavior would be.

The design process of the AI system proceeds as follows:

1. A machine learning component is fed large quantities of data and learns some hypothesis $\tilde{h} \in \mathcal{H}$ that it believes has low loss with respect to L.
2. A prover component P interacts with a separate verifier component V, which has access to S. They execute an interactive proof protocol for verifying the statement
$$\mathbb{P}_S\left[L(\tilde{h}) \leq \inf_{h\in\mathcal{H}} L(h) + \varepsilon\right] \geq 1 - \delta,$$
for some minuscule $\varepsilon, \delta > 0$. Note that V does not necessarily need to have a formal specification of L. Rather, it might suffice that V can evaluate L at a small number of random points by using the sample S.
3. If V accepts the proof, the AI policy $\tilde{h}$ is made operative. Otherwise, it is never activated.

Implemented correctly, such a system could provide mathematically-sound and independently-verifiable guarantees that the policy selected for activation is indeed likely to

[17] As noted by Pinker and others, the logic of this style of doomsday scenario appears somewhat self-contradictory: On the one hand it assumes that the AI is brilliant enough to anticipate and overcome human opposition, and at the same time assumes that the AI is so moronic as to make elementary blunders of understanding when interpreting the instructions of its human operators (see Pinker, 2018, p. 299). Still, an AI with a patchy understanding of humans could cause considerable harm if it is made operative without sufficient prior verification.

be aligned with human preferences.

B Thresholds Over Discrete Sets

In Section 3 we presented the class $\mathcal{T}$ of thresholds over the interval $[0,1] \subseteq \mathbb{R}$, and neglected issues pertaining to the the representation of real numbers. Here, we outline how similar results can be obtained for the class of threshold over a finite set $\mathcal{X} \subseteq [0,1]$. We write $\mathcal{T}^{\mathcal{X}} = \{f_t\}_{t\in\mathcal{X}} \subseteq \mathcal{T}$, and are interested in 2-PAC verification of $\mathcal{T}^{\mathcal{X}}$ with respect to any distribution $\mathcal{D} \in \Delta(\mathcal{X} \times \{0,1\})$.[18]

This boils down to the following. Recall that when constructing certificates of loss for $\mathcal{T}$, we used the following primitive in the proof of Claim 3.5:

Fact B.1. *Let $[\alpha,\beta] \subseteq \mathbb{R}$ be an interval, and let p be a distribution over $\mathbb{R}$ that is absolutely continuous with respect to the Lebesgue measure. If $p\Big([\alpha,\beta]\Big) > r \geq 0$, then there exists $\gamma \in [\alpha,\beta]$ such that $p\Big([\alpha,\gamma]\Big) = r$.*

The following alternative primitive, which has the additional property that $\gamma \in \mathcal{X}$, will be used instead when producing certificates for $\mathcal{T}^{\mathcal{X}}$ that have succinct representations.

Claim B.2. *Let $N \in \mathbb{N}$, let $[\alpha,\beta] \subseteq \mathbb{R}$ be an interval with $\alpha,\beta \in \mathcal{X}$, and let p be a probability mass function over $\mathcal{X}$. If $p\Big([\alpha,\beta]\Big) > r \geq 0$, then there exists a pair (γ,q) where $\gamma \in \mathcal{X} \cap [\alpha,\beta]$ and $q \in [N]$, such that:*

$$\left| p\Big([\alpha,\gamma)\Big) + \frac{q}{N} \cdot p(\gamma) - r \right| \leq \frac{1}{2N}.$$

Likewise, there exists (γ',q') such that

$$\left| p\Big((\gamma',\beta]\Big) + \frac{q'}{N} \cdot p(\gamma') - r \right| \leq \frac{1}{2N}.$$

Proof. Take

$$\gamma = \min\Big\{x \in \mathcal{X} :\ p\Big([\alpha,x]\Big) \geq r\Big\}$$

and

$$q = \mathrm{argmin}_{i\in[N]} \left| \frac{i}{N} - \frac{r - p\Big([\alpha,\gamma)\Big)}{p(\gamma)} \right|.$$

[18] That is, any probability space $(\Omega,\mathcal{D},\Sigma)$ with sample space $\Omega = \mathcal{X} \times \{0,1\}$, probability mass function $\mathcal{D}$, and σ-algebra $\Sigma = 2^{\Omega}$.

Observe that

$$\left|p\Big([\alpha,\gamma)\Big)+\frac{q}{N}\cdot p(\gamma)-r\right| \leq \left|p\Big([\alpha,\gamma)\Big)+\frac{r-p\Big([\alpha,\gamma)\Big)}{p(\gamma)}\cdot p(\gamma)-r\right|+p(\gamma)\left|\frac{r-p\Big([\alpha,\gamma)\Big)}{p(\gamma)}-\frac{q}{N}\right|$$

$$=\left|\frac{r-p\Big([\alpha,\gamma)\Big)}{p(\gamma)}-\frac{q}{N}\right| \leq \frac{1}{2N}.$$

The proof for (γ',q') is similar. ■

Recall that a 0-valid certificate of loss ℓ for $\mathcal{T}$ with respect to distribution $\mathcal{D}$ was a pair (a,b) such that $\mathcal{D}^1\Big([0,a)\Big)=\mathcal{D}^0\Big([b,1]\Big)=\ell$, where $\mathcal{D}^i(X):=\mathcal{D}(X\times\{i\})$. For the discete case, we use the following definition of a *certificate with finite resolution*.

Definition B.3. *Fix $N\in\mathbb{N}$, and let $\mathcal{X}\subseteq[0,1]$ be a finite set. Let $\mathcal{D}\in\Delta(\mathcal{X}\times\{0,1\})$ be a distribution and $\ell,\eta\geq 0$. A certificate of loss at least ℓ for class $\mathcal{T}^{\mathcal{X}}$ with resolution $\frac{1}{N}$ is a tuple*

$$(a,q_a,b,q_b)$$

where $0<a\leq b<1$ and $q_a,q_b\in[N]$, and if $a=b$ then $q_a+q_b\leq N$.

We say that the certificate is η-valid with respect to distribution $\mathcal{D}$ if

$$\left|\mathcal{D}^1\Big([0,a)\Big)+\frac{q_a}{N}\cdot p(a)-\ell\right|+\left|\mathcal{D}^0\Big((b,1]\Big)+\frac{q_b}{N}\cdot p(b)-\ell\right|\leq\eta.$$

Using Claim B.2, one can repeat the proof of Claim 3.5 to show the following.

Claim B.4. *Fix $N\in\mathbb{N}$, and let $\mathcal{X}\subseteq[0,1]$ be a finite set. Let $\mathcal{D}\in\Delta(\mathcal{X}\times\{0,1\})$ be a distribution and $\ell\geq 0$. If $L_{\mathcal{D}}(\mathcal{T}^{\mathcal{X}})=\ell$, then there exist (a,q_a,b,q_b) such that $a,b\in\mathcal{X}$ and $q_a,q_b\in[N]$, which constitute a certificate of loss $\frac{\ell}{2}$ for the class $\mathcal{T}^{\mathcal{X}}$ that is $\frac{1}{N}$-valid with respect to $\mathcal{D}$.*

In particular, one can obtain an η-valid certificate of finite precision by choosing the precision parameter N to satisfy $N\geq\frac{1}{\eta}$. Likewise, it is possible to repeat the rest of the analysis, and show that an η-valid certificate of loss ℓ entails that $L_{\mathcal{D}}(\mathcal{T}^{\mathcal{X}})\geq\ell-\eta$, and that certificates can be generated and verified efficiently. Finally, we can generalize these results to a multi-threshold class $\mathcal{T}_d^{\mathcal{X}}$, and obtain that $\mathcal{T}_d^{\mathcal{X}}$ is 2-PAC verifiable, and exhibits a quadratic gap in sample complexity between learning and verification, as in Lemma 3.8.

C Uniform Convergence for Set Systems

The following theorem is due to Vapnik and Chervonenkis (1971). See also the exposition by Alon and Spencer (2000, Theorem 13.4.4).

Definition C.1. *A set system is a tuple $(X,\mathcal{S})$, where X is any set, and $\mathcal{S}\subseteq 2^X$ is any collection of subsets of X. The members of X are called points.*

The VC dimension of a set system $(X,\mathcal{S})$ is the VC dimension of the set of indicator functions

$\{\mathbb{1}_S : S \in \mathcal{S}\}$ as defined in Definition 1.9.

Definition C.2. *Let $(X, \mathcal{S})$ be a set system, let $\mathcal{D}$ be a distribution over X, and let $\varepsilon \in (0,1)$. We say that a multiset $A \subseteq X$ is an $\underline{\varepsilon\text{-sample with respect to } \mathcal{D}}$ if*

$$\forall S \in \mathcal{S} : \quad \left| \frac{|A \cap S|}{|A|} - \mathcal{D}(S) \right| \leq \varepsilon.$$

Theorem C.3. *There exists a constant $c > 0$ such that for any set system $(X, \mathcal{S})$ of VC-dimension at most d and any $0 < \varepsilon, \delta < \frac{1}{2}$, a sequence of at least*

$$\frac{c}{\varepsilon^2} \left(d \log \frac{d}{\varepsilon} + \log \frac{1}{\delta} \right)$$

i.i.d. samples from $\mathcal{D}$ will be an ε-sample with respect to $\mathcal{D}$ with probability at least $1 - \delta$.

D Identity Testing for Distributions

The following theorem is due to Batu et al. (2001, Theorem 24). See exposition in Canonne (2015, Theorem 3.2.7).

Theorem D.1. *Let $\mathcal{D}^* = (d_1, \ldots, d_n)$ be a distribution over a finite set of size n, and let $\varepsilon \in (0,1)$. There exists an algorithm which, given the full specification of D^* and sample access to an unknown distribution D, takes*

$$O\left(\frac{\sqrt{n}}{\varepsilon^6} \log(n) \right)$$

samples from D, and satisfies:

- *Completeness. If*

$$d_{\mathsf{TV}}\left(D, D^*\right) \leq \frac{\varepsilon^3}{300\sqrt{n}\log n},$$

  then the algorithm accepts with probability at least $\frac{2}{3}$.
- *Soundness. If*

$$d_{\mathsf{TV}}\left(D, D^*\right) > \varepsilon,$$

  then the algorithm rejects with probability at least $\frac{2}{3}$.

A standard amplification argument yields the following:

Corollary D.2. *Taking*

$$O\left(\log\left(\frac{1}{\delta}\right) \frac{\sqrt{n}}{\varepsilon^6} \log(n) \right)$$

samples is sufficient to ensure completeness and soundness at least $1 - \delta$ (instead of $\frac{2}{3}$).

E Total Variation Distance

Claim E.1. *Let $\delta \in (0,1)$, $\mathcal{X} := [n]$. Consider a sequence $x_1, x_2 \dots, x_t$ of i.i.d. samples taken from $\mathcal{U}_{\mathcal{X}}$, and let G denote the event in which all the samples are distinct, that is $|\{x_1, \dots, x_t\}| = t$. Then taking*

$$n \geq \frac{\log(2e)}{\log\left(\frac{1}{1-\delta}\right)} \cdot t^2$$

entails that

$$\mathbb{P}[G] \geq 1 - \delta.$$

Claim E.2. *Let $\mathbb{P}, \mathbb{Q}$ be probability functions over a probability space $(\Omega, \mathcal{F})$. Then for all $\alpha \in [0,1]$,*

$$d_{\mathsf{TV}}\left((1-\alpha)\mathbb{P} + \alpha\mathbb{Q}, \mathbb{P}\right) \leq \alpha.$$

In particular, if X is a random variable and E is an event, then

$$d_{\mathsf{TV}}\left(X, X|E\right) \leq 1 - \mathbb{P}[E] = \mathbb{P}\left[\overline{E}\right].$$

Proof.

$$\begin{aligned} d_{\mathsf{TV}}\left((1-\alpha)\mathbb{P} + \alpha\mathbb{Q}, \mathbb{P}\right) &= \max_{A \in \mathcal{F}}\ (1-\alpha)\mathbb{P}(A) + \alpha\mathbb{Q}(A) - \mathbb{P}(A) \\ &= \max_{A \in \mathcal{F}}\ \alpha \cdot (\mathbb{Q}(A) - \mathbb{P}(A)) \leq \alpha. \end{aligned}$$

In particular, if $\mathbb{P}_X, \mathbb{P}_{X|E}$ denote the distributions of X and $X|E$ then

$$d_{\mathsf{TV}}\left(\mathbb{P}_X, \mathbb{P}_{X|E}\right) = d_{\mathsf{TV}}\left(\left(1 - \mathbb{P}\left[\overline{E}\right]\right) \cdot \mathbb{P}_{X|E} + \mathbb{P}\left[\overline{E}\right] \cdot \mathbb{P}_{X|\overline{E}}, \mathbb{P}_{X|E}\right) \leq \mathbb{P}\left[\overline{E}\right].$$

F Learning Fourier-Sparse Functions By Estimating Heavy Coefficients

Let $\mathcal{H}$ be the set of t-sparse functions $\{0,1\}^n \to \mathbb{R}$. In this appendix we prove that one can PAC learn $\mathcal{H}$ with respect to $\mathfrak{D}_{\mathcal{U}}^{\text{func}}(\{0,1\}^n)$ by estimating heavy Fourier coefficients.

Claim F.1. *Let $\varepsilon > 0$. Let $\mathcal{D} \in \mathfrak{D}_{\mathcal{U}}^{\text{func}}(\{0,1\}^n)$ have target function $f\colon \{0,1\}^n \to \{1,-1\}$. Consider the function*

$$h(x) = \sum_{T \in L} \alpha_T \chi_T(x),$$

where L is a set such that $\hat{f}^{\geq\tau} \subseteq L$ for $\tau = \frac{\varepsilon}{4t}$. If

$$\forall T \in L\colon\ |\alpha_T - \hat{f}(L)| \leq \sqrt{\frac{\varepsilon}{8|L|}},$$

then $L_{\mathcal{D}}(h) \leq L_{\mathcal{D}}(\mathcal{H}) + \varepsilon$.

Before proving this claim, we show that if a function f is close to being sparse, then it can be approximated by a sparse function g that includes only coefficients where f has high Fourier weight.

Claim F.2. *Let $t \in \mathbb{N}$, let $\beta, \ell \in (0,1)$, and let $\mathcal{D} \in \mathfrak{D}_{\mathcal{U}}^{\text{func}}(\{0,1\}^n)$ have target function $f:\ \{0,1\}^n \to \{1,-1\}$. Assume $L_{\mathcal{D}}(\mathcal{H}) \leq \ell$. Then exists $g \in \mathcal{H}$ such that*

$$L_{\mathcal{D}}(g) \leq (1+\beta)\cdot \ell,$$

and $\hat{g}^{>0} = \{T:\ |\hat{g}(T)| > 0\} \subseteq \hat{f}^{\geq \tau}$ with $\tau := \sqrt{\frac{\beta \cdot \ell}{t}}$.

Proof. Because $L_{\mathcal{D}}(\mathcal{H}) \leq \ell$, there exists a function $w \in \mathcal{H}$ such that $L_{\mathcal{D}}(w) \leq \ell$. Let $\hat{w}^{>0} = \{T:\ |\hat{w}(T)| > 0\}$. Consider the function

$$g(x) = \sum_{T \in \left(\hat{w}^{>0} \cap \hat{f}^{\geq \tau}\right)} \hat{f}(T)\chi_T(x).$$

Clearly, g is t-sparse (because w is t-sparse), and $\hat{g}^{>0} \subseteq \hat{f}^{\geq \tau}$. Furthermore, we have

$$\begin{aligned} L_{\mathcal{D}}(g) &= \mathbb{E}_{x \in \{0,1\}^n}\left[(f(x) - g(x))^2\right] \\ &= \sum_{T \subseteq [n]} \left(\hat{f}(T) - \hat{g}(T)\right)^2 && \text{(Parseval's identity)} \\ &= \sum_{T \notin \left(\hat{w}^{>0} \cap \hat{f}^{\geq \tau}\right)} \left(\hat{f}(T) - \hat{g}(T)\right)^2 \\ &= \sum_{T \notin \hat{w}^{>0}} \hat{f}^2(T) + \sum_{T \in \hat{w}^{>0} \setminus \hat{f}^{\geq \tau}} \hat{f}^2(T). \end{aligned}$$

We bound each sum separately.

$$\sum_{T \notin \hat{w}^{>0}} \hat{f}^2(T) = \sum_{T \notin \hat{w}^{>0}} \left(\hat{f}(T) - \hat{w}(T)\right)^2 \leq \sum_{T \subseteq [n]} \left(\hat{f}(T) - \hat{w}(T)\right)^2 = L_{\mathcal{D}}(w) \leq \ell,$$

and

$$\sum_{T \in \hat{w}^{>0} \setminus \hat{f}^{\geq \tau}} \hat{f}^2(T) \leq |\hat{w}^{>0}| \cdot \tau^2 \leq t \cdot \frac{\beta \ell}{t} = \beta \ell. \blacksquare$$

Proof of Claim F.1. Observe that

$$L_{\mathcal{D}}(h) = \mathbb{E}\left[(f(x) - h(x))^2\right] = \sum_{T \in L} \left(\hat{f}(T) - \hat{h}(T)\right)^2 + \sum_{T \notin L} \hat{f}^2(T),$$

and the first sum is bounded by

$$\sum_{T \in L} \left(\hat{f}(T) - \hat{h}(T)\right)^2 \leq |L| \cdot \frac{\varepsilon}{2|L|} = \frac{\varepsilon}{2}.$$

Therefore, to complete the proof it suffices to show that $\sum_{T\notin L}\hat{f}^2(T) \le L_{\mathcal{D}}(\mathcal{H}) + \frac{\varepsilon}{2}$. Invoking Claim F.2 with $\beta := \frac{\varepsilon}{2}$ and $\ell := \max\{L_{\mathcal{D}}(\mathcal{H}), \frac{\varepsilon}{4}\}$, there exists a t-sparse function $g :\ \{0,1\}^n \to \mathbb{R}$ such that

$$L_{\mathcal{D}}(g) \le (1+\beta)\ell \le L_{\mathcal{D}}(\mathcal{H}) + \frac{\varepsilon}{2},$$

and $\hat{g}^{>0} = \{T :\ |\hat{g}(T)| > 0\} \subseteq \hat{f}^{\ge \tau}$ with $\tau := \sqrt{\frac{\varepsilon \ell}{2t}} \ge \frac{\varepsilon}{4t}$. This entails that

$$\begin{aligned}\sum_{T\notin L} \hat{f}^2(T) &\le \sum_{T \in \hat{f}^{<\tau}} \hat{f}^2(T)\\ &\le \sum_{T\subseteq[n]} \left(\hat{f}(T) - \hat{g}(T)\right)^2\\ &= \mathbb{E}\left[(f(x)-g(x))^2\right] \le L_{\mathcal{D}}(\mathcal{H}) + \frac{\varepsilon}{2}. \blacksquare\end{aligned}$$

G Random Matrices Have Full Rank

Claim G.1. *Let $\tau > 0$, $n \in \mathbb{N}$. If $\tau \ge 2^{-\frac{n}{10}}$*

$$\frac{n}{\tau^4 2^n} \le \frac{1}{128 \log\left(\frac{n}{\tau^4}\right)}$$

for n large enough.

Proof.

$$\begin{aligned}\tau \ge 2^{-0.1n} \implies \tau^8 \ge 2^{-0.8n} \ge \frac{128 n \log(n)}{2^n} \implies \frac{2^n \tau^8}{n} \ge 128\log(n)\\ \implies \frac{2^n\tau^4}{n} \ge \frac{1}{\tau^4} 128\log(n) \ge 128 \log\left(\frac{n}{\tau^4}\right). \blacksquare\end{aligned}$$

Claim G.2. *Let $n, m \in \mathbb{N}$, $\tau \ge 2^{-\frac{n}{10}}$, $m \le \log\left(\frac{32n}{\tau^4}\right)$. Let $X = \{x_1, \dots, x_m\}$ be a set of m vectors chosen independently and uniformly from $(\mathbb{F}_2)^n$. Then with probability at least $\frac{3}{4}$, the set X is linearly independent for n large enough.*

Proof. Think of the vectors as being chosen one by one. The probability that the first vector is non-zero is

$$\frac{2^n - 1}{2^n},$$

because we can chose any vector except 0. The probability that vector x_{k+1} is linearly independent of the first k vectors is

$$\frac{2^n - 2^k}{2^n},$$

because we can choose any vector not in $\operatorname{span}(\{x_1, \dots, x_k\})$. Because the choices are made independently, the probability that all m vectors are linearly independent is

$$\frac{2^n-2^0}{2^n}\cdot\frac{2^n-2^1}{2^n}\cdots\frac{2^n-2^{m-1}}{2^n}\geq\left(\frac{2^n-2^m}{2^n}\right)^m$$

$$\geq\left(\frac{2^n-\frac{32n}{\tau^4}}{2^n}\right)^m=\left(1-\frac{\frac{32n}{\tau^4}}{2^n}\right)^m=\left(1-\frac{1}{4\log\left(\frac{n}{\tau^4}\right)}\right)^{\log\left(\frac{n}{\tau^4}\right)}\geq 1-\frac{1}{4},$$

where the last inequality is Bernoulli's inequality. ■