Electronic Colloquium on Computational Complexity, Report No. 85 (2023)

Average-Case PAC-Learning from Nisan's Natural Proofs

Ari Karchmer*
Boston University
arika@bu.edu

June 4, 2023

Abstract

Carmosino et al. (2016) demonstrated that natural proofs of circuit lower bounds imply algorithms for learning circuits with membership queries over the uniform distribution. Indeed, they exercised this implication to obtain a quasi-polynomial time learning algorithm for $\mathsf{AC}^0[p]$ circuits, for any prime p, by leveraging the existing natural proofs from Razborov (1987) and Smolensky (1987). This achievement raises a logical question: can *existing* natural proofs be adapted into learning algorithms that utilize *random examples* and learn over *unknown, arbitrary example distributions*?

In this work, we show that natural circuit lower bounds proven by specific communication complexity arguments (e.g., Nisan (1994)) witness a "yes" answer to this question, under the one limitation of *average-case* learning. Our primary technical contribution demonstrates a connection between the complexity of learning a concept class in the average-case, and the *randomized communication complexity* of an *evaluation game* associated with the class. We apply this finding to derive polynomial time average-case PAC-learning algorithms that use only random examples from arbitrary and unknown distributions, for any concept class that may be evaluated by (for instance) a majority vote of linear threshold functions.

Additionally, our work contributes to a better understanding of the optimal parameters in XOR lemmas for communication complexity. We address a question posed by Viola and Wigderson (2007) by demonstrating that certain enhancements of parameters in their XOR lemmas are false, assuming the existence of one-way functions.

*Most of this work was completed while the author was visiting the Simons Institute for the theory of computing.

ISSN 1433-8092

1 Introduction

The theory of computation reveals profound connections between cryptography, computational learning, and complexity. For instance, Razborov and Rudich [RR97] introduced *natural proofs* of circuit lower bounds. Informally, a natural proof of a lower bound for a circuit class Λ encodes an efficient algorithm that can be used to distinguish between the truth tables of *easy* boolean functions (those with small Λ-circuit complexity), and *random* boolean functions. Razborov and Rudich observed such algorithms as features of many circuit lower bound proofs, and then used this observation to explain the lack of progress on one of the fundamental problems in complexity theory: whether NP is contained in $\mathsf{P/poly}$. Specifically, Razborov and Rudich showed that the widely-believed existence of cryptographic pseudorandom functions (PRFs) excludes natural proofs as a well-founded approach to NP vs. $\mathsf{P/poly}$.

Carmosino et al. [CIKK16] strengthened the result of [RR97] by demonstrating that hypothesized natural proofs of circuit lower bounds for Λ imply algorithms for *learning* Λ-circuits with *membership queries* over the *uniform distribution* (provided that Λ is a sufficiently strong circuit class). Yet, arguably the most significant achievement of [CIKK16] was the transformation of the *existing* natural proofs of lower bounds for $\mathsf{AC}^0[p]$ circuits, for any prime p [Raz87, Smo87], into an *unconditional* learning algorithm for $\mathsf{AC}^0[p]$ circuits.[1] Going forward, we refer to this algorithm as the CIKK algorithm.

Since the discovery of the CIKK algorithm, whether the learning from natural proofs paradigm can be extended to Valiant's *original* PAC-learning model [Val84] has remained an open problem (recently highlighted for instance in [GK23]). In Valiant's original model, learning algorithms are forced to utilize *random examples*, and learn over *unknown example distributions*.

> **Question 1.** Which *existing* natural circuit lower bounds, if any, can be transformed into efficient learning algorithms in Valiant's original PAC model?

Aside theoretical interest in complexity and learning theory, Question 1 is motivated by the prospect of implicitly extending the nonexistence of PRFs in low circuit classes (derived from [Raz87, Smo87, RR97, CIKK16]) to the nonexistence of *weak* PRFs.[2] Low complexity weak PRFs are used in a variety of important cryptographic applications such as symmetric-key encryption and message authentication (see e.g. [BCG+21] for more commentary).

In this work, we address Question 1 by showing that natural circuit lower bounds proved by certain *communication complexity* arguments (e.g. Nisan's lower bounds for depth-2 circuits of majority gates [Nis93]) imply PAC-learning algorithms, under the sole constraint that PAC-learning is *on average*, with respect to a *target distribution* over concepts. Key to obtaining our result is the discovery of a relationship between the computational complexity of average-case PAC-learning a concept class, and the *randomized communication complexity* of an *evaluation game* associated with the concept class.

As a second application of this relationship, we obtain insights to a seemingly unrelated question studied by Viola and Wigderson [VW07]:

> **Question 2.** What are the best possible parameters within XOR lemmas for 2-party communication complexity?

Towards answering Question 2, we prove the impossibility of various tighter-than-before XOR lemmas, under cryptographic assumptions. The tighter the XOR lemma, the weaker the cryptographic assumption needs to be; our assumptions range from security of the famous $\mathsf{XOR\text{-}MAJ}$ weak PRF candidate of Blum et al. [BFKL93], to refutation of standard one-way functions.

[1] All circuit classes are standard, and defined in Section 2.4.

[2] A weak PRF is a PRF that is only required to be secure if the adversary can inspect uniformly random points, as opposed to to chosen points (see Section 2.3 for formal definitions).

1.1 Average-Case PAC-Learning and Evaluation Games

The main technical contribution of this work is the relationship between the computational complexity of average-case learning a concept class, and the randomized communication complexity of the evaluation game associated with the class. We now describe this evaluation game.

For any boolean concept class $\mathfrak{C}$, we fix a corresponding *evaluation function* $\Phi : \{0,1\}^* \times \{0,1\}^* \to \{-1,1\}$. The first input to the evaluation function is a *representation* π_f of a *concept* f, and the second is an *input* to the concept x. The evaluation function is defined so that $\Phi(\pi_f, x) = f(x)$ for every f and x.

The 2-party communication game $\mathfrak{G}$ associated with a concept class and corresponding evaluation function is thus as follows. Party one is given the representation π_f of f, and party two is given the input x. The two parties communicate until they are ready to output a value b, and win the game if $b = \Phi(\pi_f, x) = f(x)$. We say that the boolean concept class $\mathfrak{C}$ is *evaluated* by a 2-party communication protocol with cost c if the two parties can communicate at most c bits before winning $\mathfrak{G}$. The definition of $\mathfrak{G}$ can easily be extended to γ-biased randomized communication protocols, where the two parties need only to win $\mathfrak{G}$ with probability $1/2 + \gamma$ over the choice of a shared random string.[3]

Denote by $\mathrm{EX}(f, \rho)$ an *example oracle* that returns labelled examples $\langle x, f(x)\rangle$ for $x \sim \rho$, where ρ is an *example distribution*. Let μ be an efficiently samplable *target distribution* over string representations of concepts $f \in \mathfrak{C}$. Informally, we prove that for any μ, there exists a learning algorithm that, for *any* (not necessarily efficiently samplable) example distribution ρ, PAC-learns a large probability mass of concepts $f \in \mathfrak{C}$ using access to $\mathrm{EX}(f, \rho)$. The computational complexity of the learning algorithm is exponential in the cost of $\mathfrak{G}$ associated with $\mathfrak{C}$. More formally:

Theorem 1.1 (Average-case PAC-learning — efficient evaluation games)**.** *Suppose $\mathfrak{C}$ is evaluated by a 2-party randomized communication protocol with cost c and bias γ, and fix an efficiently samplable target distribution μ. Then, there exists a learning algorithm A such that, for any $\varepsilon, \delta, \eta > 0$,*

$$\Pr_{\pi_f \sim \mu}\left[\Pr_{A}\left[\forall \rho : \Pr_{x \sim \rho}\left[h(x) \neq f(x) : h \leftarrow A^{\mathrm{EX}(f,\rho)}(\varepsilon, \delta, \eta)\right] \leq \varepsilon\right] \geq 1 - \delta\right] \geq 1 - \eta \tag{1}$$

where A runs in time polynomial in $|\pi_f|, \varepsilon^{-1}, \delta^{-1}, \eta^{-1}, \gamma^{-1}$ and 2^c.

We give an overview of the proof of this theorem in Section 1.3. We emphasize that the learning algorithm implicit in this theorem is totally robust to example distributions ρ, which need not be known to the algorithm or be efficiently samplable. This contrasts with other recent formalizations of average-case learning, such as heuristic PAC-learning (heurPAC-learning) [Nan21], where the order of quantifiers is different. In heurPAC-learning, it is only required that, for each ρ, a large (possibly different) probability mass of the concept class is learned. Therefore our notion of average-case learning is stronger. In Section 1.6, we further discuss the relation to heurPAC-learning, as well as the average-case learning model of [BFKL93].

1.2 Using Theorem 1.1 to Learn from Nisan's Circuit Lower Bounds

Consider the following template for proving a circuit lower bound. Identify a function $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, which requires *high* 2-party communication complexity (in Yao's standard model, for example). Then, identify a circuit class $\mathcal{C}$ such that for every $g \in \mathcal{C}$, g is computable by a *low-cost* 2-party communication protocols. Finally, conclude that f requires large $\mathcal{C}$-circuits (see Section 2 for essential definitions of 2-party communication complexity, and [KN96] for further reference). To provide an example, let $f(x, y) = \mathsf{IP2}(x, y) = \sum_{i=1}^{n} x_i y_i \mod 2$ be the inner product mod 2 function. It is known that $\mathsf{IP2}$ requires $\Omega(n)$ bits to be transmitted in any randomized communication complexity

[3] Similar evaluation games were considered in for instance [KNR99, FX14].

protocol; therefore, as shown by [Nis93], since $\mathsf{MAJ} \circ \mathsf{THR}$ circuits are computed by bounded-error randomized communication complexity protocols with cost $O(\log n)$, $\mathsf{IP2}$ must require $\mathsf{MAJ} \circ \mathsf{THR}$ circuits of exponential size. This lower bound remains one of the strongest known — as of now it is still not ruled out that $\mathsf{NEXP} \subseteq \mathsf{THR} \circ \mathsf{THR}$. In fact, proving NEXP is not contained in $\mathsf{THR} \circ \mathsf{THR}$ is considered a "major frontier" in complexity theory [Che18].

This circuit lower bound method is natural (in the sense of [RR97]), as noted briefly in [Raz00]. This fact was also leveraged in [Vio15], where it was used to obtain impossibility of pseudorandom functions in AC^0 with a few MAJ-gates. Our next theorem uses Theorem 1.1 to transform the exponential circuit lower bounds of [Nis93] into polynomial time average-case PAC-learning algorithms, rather than just pseudorandom function distinguishers:

Theorem 1.2 (Average-case PAC-learning — from Nisan's lower bounds)**.** *Any concept class with a corresponding evaluation function contained in either of the following function classes is average-case PAC-learnable (in the sense of Theorem 1.1) in polynomial time.*

- $\mathsf{MAJ} \circ \mathsf{THR}$.
- *Constant depth decision trees with linear threshold queries at the nodes.*

These learning algorithms follow from Theorem 1.1 since, as indicated in Nisan's lower bounds, every function in each class has a randomized communication protocol of cost $O(\log n)$ and large bias. We can also obtain quasi-polynomial time learning algorithms for concept classes with more complex evaluation functions, like a majority vote of polylogarithmic-depth decision trees with threshold nodes, by a similar argument.

Considering circuit classes by the complexity of their evaluation function is necessarily weaker than the learning-theoretic standard of considering the complexity of concepts directly; any concept class $\mathfrak{C}$ evaluated by the scheme $\Phi \in \mathcal{C}$ means that $\mathfrak{C} \subseteq \mathcal{C}$.[4] Measuring complexity of concepts by complexity of their evaluation scheme often arises when studying average-case learning (see e.g. [BFKL93] and more recently [Nan21]). In general, it is an open problem to characterize which concepts can be evaluated by circuit classes below NC. However, we know that $\mathsf{MAJ} \circ \mathsf{THR}$ is capable of adding n n-bit integers [SR94], and computing any polynomial size DNF or CNF [She09].

1.3 Proof Idea of Theorem 1.1

We will now overview the ideas behind the proof of Theorem 1.1. The most important tool we use is the 2-party norm of a function, $R_2(f)$, which is defined to be the expected product of a function computed on a list of correlated inputs.

Definition 1.1 (2-party norm)**.** *For $f : (\{0,1\}^n)^2 \to \{-1,1\}$, the 2-party norm of f is defined as*

$$R_2(f) := \mathop{\mathbb{E}}_{x_1^0, x_2^0, x_1^1, x_2^1 \sim \{0,1\}^n} \left[\prod_{\varepsilon_1, \varepsilon_2 \in \{0,1\}} f(x_1^{\varepsilon_1}, x_2^{\varepsilon_2}) \right] \tag{2}$$

The 2-party norm is a special case of the k-party norm (sometimes called the cube-measure), which was introduced by [BNS92] for obtaining lower bounds in k-party Number-on-Forehead communication complexity.

The crucial property about $R_2(f)$ is that, up to parameters, it upper bounds the correlation of f with functions computable by 2-party communication protocols. Implicit in all three of [CT93, Raz00, VW07] (who showed a related theorem in the more general k-party case), is the following bound:

[4] For classes admitting self-evaluation (e.g. NC, P), these coincide, but not necessarily for lower circuit classes.

Theorem 1.3 (*The* correlation bound — [CT93, Raz00, VW07])**.** *For every function* $f : (\{0,1\}^n)^2 \to \{-1,1\}$,

$$\mathrm{Cor}(f, \Pi[2,c]) \leq 2^c \cdot R_2(f)^{1/4} \tag{3}$$

Equation (3) indicates that $2^{-4c} \cdot \gamma^4 \leq R_2(f)$, where γ is the bias of a *randomized* communication protocol for f with cost c (e.g., $\gamma = 1/4$ if the randomized protocol succeeds with probability $3/4$).

The construction of the learning algorithm of Theorem 1.1 uses the lower bound on $R_2(f)$ to distinguish structure from randomness. In other words, hypothetically consider functions $f : (\{0,1\}^n)^2 \to \{-1,1\}$ such that the quantity $2^{-4c} \cdot \gamma^4$ is relatively *large* (greater than $1/\mathrm{poly}(n)$, say). Such functions can be distinguished from uniformly random functions, by taking a random sample from the distribution over the value inside the expectation in (3). This follows from the fact that $R_2(\psi)$ for a uniformly random function $\psi : (\{0,1\}^n)^2 \to \{-1,1\}$ is bounded from above by a negligible function of n.

Using this idea, we have the following proof outline. First, we can try to prove a "distinguisher-to-predictor" lemma, in the style of [Yao82], in order to obtain a weak randomized predictor for f (a weak predictor requires accuracy of a prediction for an unseen example to be only slightly more accurate than a coin toss). Second, we could apply standard averaging arguments to construct a weak PAC-learning algorithm. Finally, we could apply celebrated boosting results from learning theory [Sch90, DW+00] to produce a full-blown PAC-learning algorithm.

However, this proof outline remains incomplete. First, the 2-party norm of the function is the expectation of a product of *correlated* inputs, so we have not given any way of using independent random examples. Second, we have said nothing of how to handle arbitrary example distributions (the inputs to f on the right hand side of (3) should be uniformly random). We handle both of these problems simultaneously, roughly by thinking of f as the *evaluation* of *representations* of concepts, not the concept itself.

First, let us describe how f should be viewed. There are two inputs to f, x_1 and x_2. Without loss of generality, identify x_1 as a random string for sampling the target distribution μ over $\mathfrak{C}$, and identify x_2 as a random string for sampling the example distribution ρ, with $|x_1| = |x_2| = m$. Let $z = \rho(x_2)$ be an input to the concept g represented by $\pi_g = \mu(x_1)$. Next, fix an *evaluation function* Φ, which is the map that takes as input the concept representation π_g, plus the input z, and outputs $\Phi(\pi_g, z) = g(z) = y$. As a function of x_1, x_2, we can thus write the process of generating a labelled example as $\langle \rho(x_2), \Phi(\mu(x_1), \rho(x_2)) \rangle = \langle z, g(z) \rangle = \langle z, y \rangle$. We let $f(x_1, x_2) = \Phi(\mu(x_1), \rho(x_2))$. This allows us to write:

$$R_2(f) = \mathop{\mathbb{E}}_{x_1^0, x_2^0, x_1^1, x_2^1} \left[v(x_1^0, x_2^0, x_1^1, x_2^1) \right] \text{ for } v(x_1^0, x_2^0, x_1^1, x_2^1) := \prod_{\varepsilon_1, \varepsilon_2 \in \{0,1\}} \Phi(\mu(x_1^{\varepsilon_1}), \rho(x_2^{\varepsilon_2}))$$

Now, we describe how we construct a weak randomized predictor which only uses random examples from an arbitrary ρ. At the core, we will use the example oracle to sample a single instance of $v(x_1^0, x_2^0, x_1^1, x_2^1)$, over uniformly random $x_1^0, x_2^0, x_1^1, x_2^1 \in \{0,1\}^m$. To see the significance of this, observe that by definition $v(x_1^0, x_2^0, x_1^1, x_2^1)$ has expected value $R_2(f)$. Hence, the process of sampling this value distinguishes examples labelled by uniformly random functions from examples labelled by concepts sampled according to μ — as long as μ is supported on $\mathfrak{C}$ that is evaluated by an efficient 2-party communication protocol. This claim is justified because whenever it is possible to win the evaluation game $\mathfrak{G}$ associated with $\mathfrak{C}$ with high bias and low communication, Theorem 1.3 implies that $R_2(f)$ is large. In other words, $R_2(f)$ is guaranteed to be large whenever it is possible to efficiently (probabilistically) communicate the evaluation function Φ (because this implies winning $\mathfrak{G}$ with good bias). At this point, we use a simple hybrid argument to construct a randomized prediction algorithm for examples sampled according to ρ.

It remains to verify that the randomized prediction algorithm can actually sample $v(x_1^0, x_2^0, x_1^1, x_2^1)$, using only access to $\mathrm{EX}(g, \rho)$, where g is the concept sampled according to the target distribution

μ. To see this, observe that the distribution over $v(x_1^0, x_2^0, x_1^1, x_2^1)$ is identical to the distribution over $g(z_1)g(z_2)h(z_1)h(z_2)$, for $\langle z_1, g(z_1)\rangle, \langle z_2, g(z_2)\rangle \sim \mathrm{EX}(g, \rho)$, and $h \sim \mu$. The value $h(z_1)h(z_2)$ can be computed because $h(z_1)$ and $h(z_2)$ can be *queried*, since h is sampled *locally* by the algorithm. Therefore, we only need $\mathrm{EX}(g, \rho)$.

We also need to verify that ρ need not be efficiently samplable. To argue this this, we observe that communicating parties participating in $\mathfrak{G}$ have unbounded computational power. This means that, even if ρ is an arbitrary distribution, there is no effect on the communication cost of $\mathfrak{G}$. Indeed, the process of sampling ρ can be viewed as a local pre-processing step in the protocol for party two. Therefore, $R_2(f)$ does not decrease when ρ is arbitrary.

1.4 Comparison to CIKK Algorithm

Let us compare the techniques behind the learning algorithm of Theorem 1.1 to the CIKK algorithm. The CIKK algorithm is based on the observation that the output of the Nisan-Wigderson [NW94] pseudorandom generator on seed z, $x = \mathrm{NW}^f(z)$, can be viewed as the truth table of a function with circuit complexity related to that of the "hard function" f. Recall that the i^{th} bit of x is defined to be the value $f(z|_{S_i})$, where S_i is the i^{th} "Nisan-Wigderson design" and $z|_{S_i}$ is a projection of the seed onto the bits indexed by the set S_i. Therefore, if $f \in \mathsf{AC}^0[p]$ then x is the truth table of a function $g_z \in \mathsf{AC}^0[p]$. This follows because for any seed z, the Nisan-Wigderson designs that specify the inputs to f can be computed in $\mathsf{AC}^0[p]$. That is, the projection of z to $z|_{S_i}$ can be computed by an $\mathsf{AC}^0[p]$ circuit, given i as input.

The CIKK algorithm proceeds by operationalizing a hybrid argument to obtain a next-bit predictor for some random bit in x, assuming a distinguisher for x.[5] For a distinguisher, the CIKK algorithm uses the [Raz87, Smo87] natural proofs against $\mathsf{AC}^0[p]$, and they convert the next-bit predictor into a full-blown weak PAC-learning algorithm by *puncturing* the seed z at indices contained in S_i, where i is the random hybrid index. By using a pre-processing stage, the learning algorithm computes all possible $z|_{S_j}$ for all j, consistent with the punctured seed z. The algorithm makes oracle queries for all these points, and memorizes the result inside a lookup table. This way, the learning algorithm can print a hypothesis circuit that essentially computes the next-bit predictor on its input — a uniformly random challenge point w. Importantly, the pre-processing step prepares the hypothesis circuit for any seed z "completed" with w. This is *necessary* because the relevant queries *depend* on the challenge point w. Though the queries specified by the CIKK algorithm are randomized, they are not uniformly distributed.

The method described so far is enough to at least obtain a weak PAC-learning algorithm for $\mathsf{AC}^0[p]$, with membership queries over the uniform distribution. Carmosino et al. further show that it is possible to obtain a learning algorithm with just inverse polynomial error with respect to the uniform distribution by composing the weak learner with a suitable hardness amplification procedure.

We can now see how our algorithm differs at a few important technical checkpoints, and how this leads to some gains and some concessions on the qualities of the learning algorithm.

- **We do not require membership queries.** In our algorithm, membership queries are circumvented by instead making membership queries to the evaluation function, rather than the actual concept. The queries to the evaluation function are *simulated* using random examples of the concept obtained via the example oracle, and sample access to the target distribution μ. One important reason this is possible is because of the way the 2-party norm is defined: it is the expected product of the four possible evaluations of two pairs of *independently* sampled random concepts and example inputs. This is in contrast to the CIKK algorithm, which requires membership queries because of the complex correlations in the construction of the queries arising from the Nisan-Wigderson designs. It is unclear how queries arising from Nisan-Wigderson designs

[5] Essentially the same technique occurs in the proof of the main theorem of [NW94].

could be simulated using only random examples of a concept, even if we consider average-case learning and evaluation functions.

- **We obtain strong learning for arbitrary example distributions.** Our algorithm learns over arbitrary example distributions. We believe this is the apparent power of one of our main technical innovations: viewing the complexity of learning through the lens of the *communication complexity* of the evaluation game $\mathfrak{G}$ of the concept class. When considering complexity this way, evidently we get sampling of ρ "for free," as it is a computation that can be processed "locally" by one (computationally unbounded) communicating party participating in $\mathfrak{G}$. On the other hand, the CIKK algorithm uses the Razborov-Smolensky distinguishers for $\mathsf{AC}^0[p]$-circuit complexity. This seems inherently incapable of handling arbitrary example distributions, since it would blow-up the circuit complexity of the string which is given to the Razborov-Smolensky distinguisher. As a related note, our learning algorithm employs learning-theoretic boosting algorithms, unlike CIKK (which uses hardness amplification over the uniform distribution). This is an added benefit of obtaining weak learning with respect to arbitrary example distributions.

- **We only obtain learning for super-efficiently evaluated concepts.** Interestingly, our techniques unconditionally cannot imply efficient learning algorithms for concept classes with evaluation function classes a bit higher than $\mathsf{MAJ} \circ \mathsf{THR}$, like $\mathsf{THR} \circ \mathsf{THR}$. This is because $\mathsf{THR} \circ \mathsf{THR}$ is known to contain functions requiring $\omega(\mathrm{polylog}(n))$ randomized communication cost (see e.g. [BVdW07]). Hence it is *not* learnable in quasi-polynomial time using our method. This is major difference with the CIKK algorithm, which uses natural proofs in the most general sense, which at the moment cannot be ruled out unconditionally. In a related way, our technique also unconditionally cannot apply to efficient learning algorithms for $\mathsf{AC}^0[p]$, as the CIKK algorithm does, because of known $n^{\Omega(1)}$ randomized communication complexity lower bounds for this class (see e.g. [BH12] against AC^0 or [VW07] against $\mathsf{AC}^0[p]$).

- **We only obtain average-case learning.** Because our algorithm needs to sample independent concept representations according to μ, our techniques are inherently average-case. On the other hand, the CIKK algorithm handles worst-case learning.

1.5 Conditional Answers to a Question of Viola and Wigderson

It is simple to see that the algorithm of Theorem 1.1 relies on *the* correlation bound (Theorem 1.3) of [CT93, Raz00, VW07] to correctly distinguish structure from randomness. Therefore, it stands to reason that — if we can *improve* this bound — then we can obtain better learning algorithms. Specifically, consider the following "improved" bound as an example

$$\mathrm{Cor}(f, \Pi[2, c]) \leq c^{100} \cdot R_2(f)^{1/4} \tag{4}$$

Is (4) true? What implications would follow from (4)? Surprisingly, very little is known about this and related questions.

The question of [VW07]. Let the m-wise direct XOR of a function $f : (\{0,1\}^n)^2 \to \{-1,1\}$ be $f^{\times m} : ((\{0,1\}^n)^2)^m \to \{-1,1\}$ for $f^{\times m}(x_1, \cdots x_m) = \prod_{i=1}^m f(x_i)$. Viola and Wigderson [VW07] showed an XOR lemma for communication protocols, stated as follows for the 2-party case (they showed a general version for k-party Number-on-Forehead communication).

Theorem 1.4 (XOR lemma — [VW07])**.** *Let $f : (\{0,1\}^n)^2 \to \{-1,1\}$ be a function of correlation at most ε with any 2-party communication protocol with cost 2. Then the correlation of $f^{\times m}$ with any 2-party communication protocol with cost c is bounded from above by $2^c \cdot \varepsilon^{m/4}$.*

Regarding this theorem, the question posed by [VW07] was whether the exponential dependency on the parameter c is the *best possible*, or if it can even be removed completely (see section 3.2.1 of [VW07] for more commentary). The dependency on c is clearly attributed to communication complexity proof techniques of [VW07] as well as many prior works such as [BNS92].

The question of [VW07] is tightly related to the question of improving *the* correlation bound. In fact, improving the 2^c factor in the correlation bound directly suffices to improve the 2^c factor in the XOR lemma (implicit in [VW07]). The other direction is not as straightforward, but is "morally" true for our purposes.

Viola and Wigderson were unable to answer their question, except for showing a counterexample to the "ideal" relationship in the 2-party case. Namely, a counterexample for going from correlation ε to correlation εm, for 2 parties communicating $c = 2$ bits.

The final contributions of this paper are to show several consequences of certain answers to the [VW07] question. We prove the following informally stated facts about the dependency on c (see Section 4 for the formal statements):

Theorem 1.5 (Not much tighter XOR lemmas — informally stated)**.** *With respect to either Theorem 1.3 or Theorem 1.4:*

- *Improving the 2^c factor to $2^{c^{1/2}}$ implies the* XOR-MAJ *weak PRF candidate of [BFKL93] can be predicted in polynomial time.*
- *Improving the 2^c factor to $2^{c^{o(1)}}$ implies subexponentially secure one-way functions do not exist.*
- *Improving the 2^c factor to $\mathrm{poly}(c)$ implies one-way functions do not exist.*

The XOR-MAJ weak pseudo-random function refers to the well known distribution designed by [BFKL93]. This is an example of a long-standing hard $O(\log n)$-juntas learning problem. The problem is defined formally in Section 4.

To give some intuition, a sketch of the proof of the third bullet of Theorem 1.5 follows.

Proof idea. Consider a polynomial time computable pseudorandom function $\mathrm{F}(s; x)$, which takes as input a *seed* $s \in \{0,1\}^{m(n)}$ and an input $x \in \{0,1\}^n$, and outputs a value in $\{-1, 1\}$. In this case, F can be thought of as a fixed polynomial size circuit which *evaluates* a function specified by s, $\mathrm{F}(s; \cdot)$, on input x. Now, assume that for any function $f : (\{0,1\}^n)^2 \to \{-1, 1\}$, there exists a polynomial p such that $\mathrm{Cor}(f, \Pi[2, c]) \leq p(c) \cdot R_2(f)^{1/4}$ (i.e., Theorem 1.3 is improved as specified by the third bullet). Under this assumption, there exists an oracle algorithm that distinguishes F from a uniformly random function roughly as follows (by standard equivalences in cryptography [GGM86, HILL99], this suffices to invert any one-way function).

- Sample uniformly random $s' \in \{0,1\}^{m(n)}, x, x' \in \{0,1\}^n$
- Query oracle at x, x', and compute $\mathrm{F}(s'; x), \mathrm{F}(s'; x')$
- Print $v = \mathrm{F}(s; x) \cdot \mathrm{F}(s; x') \cdot \mathrm{F}(s'; x) \cdot \mathrm{F}(s'; x')$

We then claim that the expected value printed by this procedure is roughly 0 when given oracle access to a random function, while bounded below by $1/q(n)$ for some polynomial q. This follows from the fact that $\mathrm{Cor}(\mathrm{F}(s; \cdot), \Pi[2, m(n)]) = 1$, and therefore $R_2(\mathrm{F}) \geq 1/(p(m(n)))^4$. Finally, the observation that $\mathbb{E}[v] = R_2(\mathrm{F})$ concludes the sketch.

1.6 Further Related Work

We discuss further related work, including other learning algorithms, average-case learning models, related techniques, and known relationships between communication complexity and learning.

Learning functions of halfspaces: positive and negative results. Learning intersections of halfspaces (i.e., linear threshold functions) was considered by [KOS04]. Using Fourier-analytic techniques, [KOS04] showed a polynomial time learning algorithm for any function of a constant number of halfspaces with respect to the uniform distribution over examples. Additionally, [KOS04] gave a quasi-polynomial time algorithm for learning any Boolean function of a *polylogarithmic* number of bounded-weights linear threshold functions, under *any* distribution over examples. Our unconditional learning results (Theorem 1.2) are at the moment similar but incomparable; we get polynomial time *average-case* learning of concepts *evaluated* by functions of linear threshold functions over *any distribution* (while [KOS04] gets worst-case learning directly of concepts, but only the uniform distribution). However, we also get learning for concepts evaluated by functions of *adaptively* chosen linear threshold functions (i.e., decision trees with THR nodes). Finally, comparing with the quasi-polynomial time algorithm of [KOS04], we get average-case learning without any restriction on the weights of the linear threshold functions (with the same caveats as before).

Our average-case learning algorithm for the $\mathsf{MAJ} \circ \mathsf{THR}$ *evaluation class* complements hardness of worst-case PAC-learning results for the same *concept class*. For example, [FGKP06] proved that $\mathsf{MAJ} \circ \mathsf{THR}$ is hard to PAC-learn, based on the Ajtai-Dwork cryptosystem [AD97] (and the error-free version by [GGH97]). The proof goes by showing that [GGH97] cryptosystem has the property that, *for every key*, the decryption algorithm can be implemented by a $\mathsf{MAJ} \circ \mathsf{THR}$-circuit. This is weaker than showing that the decryption algorithm can be expressed *as a* $\mathsf{MAJ} \circ \mathsf{THR}$*-computable function* of the key and ciphertext. The latter would correspond to our notion of complexity by evaluation class. This dichotomy illustrates key differences in considering complexity of concepts by evaluation vs. directly.

More average-case learning. Nanashima [Nan21] introduced a theory of heuristic PAC-learning (heurPAC), in part by proposing a unified average-case learning model. A crucial difference between heurPAC-learning and the average-case PAC-learning in this work is in the order of quantifiers that govern the relationship between the learning algorithm, the example distribution, and the target distribution. In this work, our version of average-case learning essentially demonstrates that a large mass of the concept class is learnable over *any* example distribution. On the other hand, heurPAC-learning only demonstrates that for each example distribution, a (possibly different) mass of the concept class is learnable. This distinction is of crucial importance for correctly applying classical boosting algorithms in our work, which do not fit in a black-box way with heurPAC-learning. In the heurPAC model, [Nan21] obtained a polynomial time algorithm for learning $O(\log n)$-juntas. This algorithm is incomparable to our algorithms. For one, $O(\log n)$-juntas are not known to be evaluated by any of the evaluation classes that we obtain polynomial time learning for. However, our learning model is more strict because of the quantifier differences described above. Lastly, we handle arbitrary example distributions and polynomial time samplable target distributions (while the algorithm for $O(\log n)$-juntas in [Nan21] only handles the uniform distribution for each).

In comparison to the seminal work of [BFKL93], the average-case learning in this work also differs on the order of quantifiers. In the definition of the average-case prediction considered by [BFKL93], both the target distribution μ and the example distribution ρ are fixed. This means that there can be a different prediction algorithm, for each pair of μ and ρ. The average-case learning in this work constructs a single learning algorithm *for every* ρ, after fixing μ.

Related techniques. Constructing efficient average-case PAC-learning algorithms for $\mathsf{P/poly}$ using only random examples from hypothesized natural lower bounds for boolean circuits of size $2^{\Omega(n)}$ has been understood previously (e.g., implicitly in [NY15, Nan21]). However, these average-case learning algorithms are in the weaker heurPAC-learning model, where the example distribution is fixed (and needs to be efficiently samplable). Also, crucial techniques such as the Goldreich-Goldwasser-Micali PRF construction [GGM86] that are available in the setting of learning $\mathsf{P/poly}$ do not work for

learning lower circuit classes. In particular, *unconditional* average-case learning from *existing* natural proofs had not been shown possible.

Other learning/communication relationships. Many other relationships between learning theory and communication complexity have been studied. Some notable examples include [KNR99, LS09, FX14, KLMY19] (also see the references therein). All of these works study relationships between communication complexity and notions of learning complexity, such as sample complexity [KNR99, KLMY19], differentially private sample complexity [FX14], margin complexity [LS09], VC dimension [KNR99, FX14] and Littlestone dimension [FX14]. These works are all incomparable to ours, as they do not directly study relationships between communication complexity and the *computational* complexity of learning.

2 Preliminaries

2.1 Average-Case PAC-Learning Model

Representation and Evaluation of Concept Classes. Let $\mathfrak{C} = \{\mathfrak{C}_n\}_{n\in\mathbb{N}}$ be a boolean concept class (i.e., a set of boolean functions). We designate a representation scheme for $\mathfrak{C}$.[6] A representation scheme for $\mathfrak{C}$ is a sequence of pairs $\{(\Pi_n, \mathcal{E}_n)\}_{n\in\mathbb{N}}$, where $\Pi_n \subseteq \{0,1\}^{s(n)}$ and $\mathcal{E}_n : \Pi_n \to \mathfrak{C}_n$ is an onto mapping from bitstrings to functions.

We define efficient evaluation of concepts by their representation schemes as follows:

Definition 2.1 (Evaluated representation scheme)**.** *We say that a representation scheme $(\Pi, \mathcal{E})$ can be evaluated by a uniform circuit family $\Phi = \{\Phi_n\}_{n\in\mathbb{N}}$ if every Φ_n, which takes as input $\pi_f \in \Pi_n \subseteq \{0,1\}^{s(n)}, x \in \{0,1\}^n$ ($s(n)+n$ bits total), outputs $\Phi_n(\pi_f, x) = \Phi_n(\pi_f)(x) = f(x)$.*

The object Φ is call the evaluation function. To capture it succinctly, we define a notion of representable concept classes.

Definition 2.2 (Representable concept class)**.** *We say that a concept class is $s(n)$-representable by Φ if the concept class has a representation scheme that can be evaluated by a Φ with $s(n)+n$ input bits total.*

Average-Case PAC-Learning. To ease notation, we define the set $I^n := \{0,1\}^n$. Let U_n denote the uniform distribution over I^n. For some boolean function $f : I^n \to \{0,1\}$, and a distribution ensemble $\rho = \{\rho_n\}_{n\in\mathbb{N}}$, with each ρ_n over I^n, we denote by $\mathrm{Ex}(f,\rho)$ an *example oracle* that on invocation returns a labelled example pair $\langle x, f(x)\rangle$ where $x \sim \rho_n$.

Fix a distribution ensemble $\mu = \{\mu_n\}_{n\in\mathbb{N}}$ over a $s(n)$-representable concept class $\mathfrak{C}$.

Definition 2.3 (Average-case PAC-learning)**.** *We say that $\mathfrak{C}$ is average-case PAC-learnable with respect to μ if there exists an algorithm A such that, for any $n \in \mathbb{N}, \varepsilon, \delta, \eta > 0$,*

$$\Pr_{\pi_f\sim\mu_n}\left[\Pr_{A}\left[\forall \rho : \Pr_{x\sim\rho_n}\left[h(x) \neq f(x) : h \leftarrow A^{\mathrm{Ex}(f,\rho)}(n,\varepsilon,\delta,\eta)\right] \leq \varepsilon\right] \geq 1-\delta\right] \geq 1-\eta \tag{5}$$

When A runs in time $\mathrm{poly}(n, s(n), \varepsilon^{-1}, \delta^{-1}, \eta^{-1})$, A is considered an efficient average-case PAC-learning algorithm.

[6] This is standard, and our formalisms resemble strongly that of (for example) [BFKL93], [Nan21].

2.2 2-Party Communication Complexity and Norms

In the following, we discuss boolean functions $f : I^n \to \{-1, 1\}$. Here we identify -1 with False and 1 with True.

Communication Models, Norms, and Bounds. The 2-party communication model is the following. There are 2 parties, each having unbounded computational power, who try to collectively compute a function. The input to the function is separated into 2 segments, and the i^{th} party sees the i^{th} segment. The parties can send each other direct messages.

Each party may transmit messages according to a fixed protocol. The protocol determines, for every sequence of bits transmitted up to that point (the transcript), whether the protocol is finished (as a function of the transcript), or if (and which) party writes next (as a function of the transcript) and what that party transmits (as a function of the transcript and the input of that party). Finally, the last bit transmitted is the output of the protocol, which is a value in $\{-1, 1\}$. The complexity measure of the protocol is the total number of bits transmitted by the parties.

Definition 2.4 ($\Pi[2, c]$ class)**.** *$\Pi[2, c]$ is defined to be the class of functions $f : (I^n)^2 \to \{-1, 1\}$ that can be computed by a 2-party communication protocol with complexity c.*

Another communication model is *randomized* communications.

Definition 2.5 (Randomized $\Pi[2, c]$)**.** *The randomized 2-party communication model allows the protocol to depend on random bits. Therefore, we allow the protocol to err in its output. The probability of error of a randomized protocol is ε if for every input to the function f, the protocol errs in outputs with probability at most ε. We denote by $\mathrm{r}\Pi[2, c, \gamma]$ the class of 2-party randomized protocols that transmit at most c bits and err with probability at most $1/2 - \gamma$.*

For the sake of simplicity, this paper uses only the public coin version of randomized communication complexity. Namely the parties all share a string of random bits.

A model more relaxed than randomized communication is *distributional* communication.

Definition 2.6 (Distributional $\Pi[2, c]$)**.** *The distributional 2-party communication model allows the protocol to err on certain inputs. Fix a distribution ρ over $(I^n)^2$. A function $f : (I^n)^2 \to \{-1, 1\}$ is in $\mathrm{d}\Pi[2, c, \rho, \gamma]$ if there exists a communication protocol $p \in \Pi[2, c]$ such that*

$$\mathop{\mathbb{E}}_{(x_1,x_2)\sim\rho} [p(x_1, x_2) \cdot f(x_1, x_2)] \geq 2\gamma$$

Distributional communication complexity can be thought of as correlation.

Definition 2.7 (Boolean function correlation)**.** *Define $\mathrm{Cor}(f, \Lambda) := \max_{h\in\Lambda} |\mathbb{E}[f(x) \cdot h(x)]|$, where x is sampled uniformly at random from the domain.*

When we want to measure correlation between two function classes, we have it defined as follows:

Definition 2.8 (Boolean function correlation)**.** *Define $\mathrm{Cor}(\mathfrak{C}, \Lambda) := \min_{f\in\mathfrak{C}} \max_{h\in\Lambda} |\mathbb{E}[f(x) \cdot h(x)]|$, where x is sampled uniformly at random from the domain.*

When ρ is the uniform distribution, $f \in \mathrm{d}\Pi[2, c, \rho, \gamma]$ is equivalent to $\mathrm{Cor}(f, \Pi[2, c]) \geq 2\gamma$.

A simple fact is that for any distribution ρ, $f \in \mathrm{r}\Pi[2, c, \gamma]$ implies that $f \in \mathrm{d}\Pi[2, c, \rho, \gamma]$. Therefore, $f \in \mathrm{r}\Pi[2, c, \gamma]$ implies that $\mathrm{Cor}(f, \Pi[2, c]) \geq 2\gamma$.

Definition 2.9 (2-party norm)**.** *For $f : (I^n)^2 \to \{-1, 1\}$, the 2-party norm of f is defined as*

$$R_2(f) := \mathop{\mathbb{E}}_{x_1^0, x_2^0, x_1^1, x_2^1 \sim I^n} \left[\prod_{\varepsilon_1, \varepsilon_2 \in \{0,1\}} f(x_1^{\varepsilon_1}, x_2^{\varepsilon_2}) \right] \tag{6}$$

The 2-party norm is a special case of the k-party norm (sometimes called the cube-measure), which was introduced by [BNS92] for obtaining lower bounds in k-party Number-on-Forehead communication complexity.

The crucial property about $R_2(f)$ is that, up to parameters, it upper bounds the correlation of f with functions computable by 2-party communication protocols. Implicit in all three of [CT93, Raz00, VW07] (who showed a related theorem in the more general k-party case), is the following bound:

Theorem 2.1 (*The* correlation bound — [CT93, Raz00, VW07])**.** *For every function $f : (I^n)^2 \to \{-1, 1\}$,*

$$\mathrm{Cor}(f, \Pi[2, c]) \leq 2^c \cdot R_2(f)^{1/4} \tag{7}$$

An immediate corollary of this bound is:

Theorem 2.2. *For every function $f : (I^n)^2 \to \{-1, 1\}$, such that $f \in \mathrm{r}\Pi[2, c, \gamma]$,*

$$\gamma \leq 2^c \cdot R_2(f)^{1/4} \tag{8}$$

2.3 Cryptography

We define weak and strong pseudorandom functions.

Definition 2.10 (Weak and Strong PRFs)**.** *Let λ be a security parameter, and $n = n(\lambda), \kappa = \kappa(\lambda)$ for polynomially bounded functions n, κ. Consider a pair of algorithms $\mathsf{F} : I^\kappa \times I^n \to I^1, \mathsf{KeyGen} : \{1\}^\lambda \to I^\kappa$. KeyGen is a polynomial time sampling algorithm that given input parameter λ in unary and access to random coins $z \in I^{\mathrm{poly}(\lambda)}$ outputs a key $k \in I^\kappa$. F is a polynomial time algorithm that given a key k and input $x \in I^n$, outputs a value $\mathsf{F}(k, x) = v \in I^1$. For $t = t(\lambda), \varepsilon = \varepsilon(\lambda)$, we say that $(\mathsf{F}, \mathsf{KeyGen})$ is a (t, ε)-weak PRF if, for every size t oracle circuit C,*

$$\left| \Pr_{k \sim \mathsf{KeyGen}(1^\lambda)} \left[C^{\mathrm{Ex}(\mathsf{F}(k,\cdot), U_n)} = 1 \right] - \Pr_r \left[C^{\mathrm{Ex}(r, U_n)} = 1 \right] \right| \leq \varepsilon(\lambda)$$

where $r : I^n \to I^1$ is a uniformly random function.

Additionally, we say that $(\mathsf{F}, \mathsf{KeyGen})$ is a (t, ε)-PRF if, for every size t oracle circuit C,

$$\left| \Pr_{k \sim \mathsf{KeyGen}(1^\lambda)} \left[C^{\mathsf{F}(k,\cdot)} = 1 \right] - \Pr_r \left[C^r = 1 \right] \right| \leq \varepsilon(\lambda)$$

where $r : I^n \to I^1$ is a uniformly random function.

When $t, \varepsilon = 2^{\log^c(n)}$ for some constant $c > 1$, we say that $(\mathsf{F}, \mathsf{KeyGen})$ has *quasipolynomial* security. When $t, \varepsilon = 2^{\lambda^\delta}$ for some constant $\delta \in (0, 1)$, we say that $(\mathsf{F}, \mathsf{KeyGen})$ has *subexponential* security.

To measure the complexity of weak and strong PRFs, we say that $(\mathsf{F}, \mathsf{KeyGen})$ is *evaluated* by a uniform circuit class Λ, if $\mathsf{F}(k, x) \in \Lambda$. This notion of complexity differs somewhat from the standard case of fixed key complexity (where for every $k, \mathsf{F}(k, \cdot) \in \Lambda$), even though they coincide for evaluation classes that admit self-evaluation such as P or NC. However, we remark that in this definition we still allow polynomial time pre-processing inside KeyGen.

2.4 Circuit Classes and Other Computational Classes

We will consider various circuit classes with different bases (all appearing previously in the literature). AC^0 is the class of constant depth, polynomial size, unbounded fan-in $\mathsf{AND}/\mathsf{OR}/\mathsf{NOT}$ circuits. $\mathsf{AC}^0[p]$ is the class of constant depth, polynomial size, unbounded fan-in $\mathsf{AND}/\mathsf{OR}/\mathsf{NOT}/\mathsf{MOD}p$ circuits, where $p \in \mathbb{N}$ is a prime number. TC^0 is the class of constant-depth, polynomial size, unbounded fan-in circuits of THR gates, where a THR gate is a linear threshold function $t(x_1, \cdots x_m) := \sum_{i=1}^{m} w_i x_i \geq^? \theta$, which outputs 1 if and only if the sum of the inputs weighted by $w_1, \cdots w_m$ exceeds a threshold θ. When the weights are fixed to be 1 and $\theta = m/2$, we call it a MAJ gate. An XOR gate takes the sum modulo 2 of its inputs.

Many circuit classes considered are of the form $\mathcal{C} \circ \mathcal{C}'$ for circuit classes $\mathcal{C}, \mathcal{C}'$. The composed class $\mathcal{C} \circ \mathcal{C}'$ denotes the class of $\mathcal{C}$-circuits with inputs as the outputs of functions from $\mathcal{C}'$. For example, the class $\mathsf{THR} \circ \mathsf{THR}$ (a.k.a. depth-2 TC^0). Alternatively, $\mathsf{MAJ} \circ \mathsf{THR}$ is the class of circuits consisting of a MAJ gate composed with a bottom layer of THR gates. The class $\mathsf{AC}^0 \circ \mathsf{MOD2}$ denotes the circuit class AC^0 with a single layer of XOR gates at the bottom.

A decision tree is a binary tree where each node contains a query of the form what is x_i? and each branch is labelled by 0/1, corresponding to the answer of the query. A decision tree computes a function by performing adaptive queries in this manner until reaching a *leaf*, which has no outgoing branches. The decision tree has depth d if the longest path from root to leaf is length d. A decision tree with $\mathcal{G}$-nodes is a decision tree with nodes labelled by queries $g \in \mathcal{G}$. For example, the decision tree may have THR-nodes.

3 Average-Case PAC-Learning from Efficient Evaluation Games

In this section, we will prove Theorem 1.1: we will show average-case PAC-learning algorithms for concept classes that have simple associated *evaluation games*.

3.1 Evaluation Games

Definition 3.1 (2-party randomized evaluation game)**.** *With respect to concept class $\mathfrak{C}$ that is $s(n)$-representable by Φ, the 2-party evaluation game $\mathfrak{G}[\mathfrak{C}, \Phi, n]$ is the following:*

- *Party 1 gets as input a representation $\pi_f \in \{0,1\}^{s(n)}$ for some concept $f \in \mathfrak{C}_n$.*
- *Party 2 gets as input a string $x \in \{0,1\}^n$.*
- *The parties may access a shared random string to aid in communication.*
- *The object of the game is for the parties to output the value $\Phi_n(\pi_f, x) = f(x)$, using as few bits of communication as possible.*

We say that $\mathfrak{G}[\mathfrak{C}, \Phi, n] \in \mathrm{r}\Pi[2, c, \gamma]$ if the parties can communicate at most c bits (for any choice of shared random string), and win the game with probability $1/2 + \gamma$ (over the choice of shared random strings).

Other definitions. We direct the reader to Section 2 for the necessary definitions of communication complexity, and the average-case PAC-learning model.

3.2 Weak Learning

Towards Theorem 1.1, we will start by first obtaining a weak learning algorithm, which only requires prediction accuracy marginally better than a coin toss. We are going to construct the weak learning dependent on some *generic* lower bound on the R_2 norm of a function. Unconditional weak learning

will follow then from known bounds (stated in Section 2.2). We start with the generic theorem as it will be technically useful in the later stages of this paper.

Notation. In the following, we discuss boolean functions $f : I^n \to \{-1, 1\}$, and denote by U_n the uniform distribution over I^n. Let $\mathfrak{C} = \{\mathfrak{C}_n\}_{n\in\mathbb{N}}$ be a boolean concept class that is $s(n)$-representable by the evaluation function $\Phi = \{\Phi_n\}_{n\in\mathbb{N}}$. For shorthand, we will write $c := c(n), \gamma := \gamma(n)$, to denote number of bits of communication and protocol bias in $\mathrm{r}\Pi[2, c, \gamma]$, which are dependent on n, the input length of concepts in the class. Also, in the rest of the paper we will streamline notation by eliding the subscripts on distributions coming from ensembles indexed by $n \in \mathbb{N}$.

Theorem 3.1. *Suppose $\mathfrak{G}[\mathfrak{C}, \Phi, n] \in \mathrm{r}\Pi[2, c, \gamma]$, and fix a time $t(n)$ samplable distribution μ over concepts representations of $f \in \mathfrak{C}$. Further assume that there is a function $\beta : \mathbb{N} \times \mathbb{R} \to \mathbb{R}$ such that $R_2(f) \geq \beta(c, \gamma)$. Then, there exists an algorithm A such that, for any $n \in \mathbb{N}, \delta, \eta > 0$,*

$$\Pr_{\pi_f\sim\mu}\left[\Pr_{A}\left[\forall\rho : \Pr_{x\sim\rho}\left[h(x) \neq f(x) : h \leftarrow A^{\mathrm{Ex}(f,\rho)}(n, \delta, \eta)\right] \leq \frac{1}{2} - \Omega(\beta(c, \gamma))\right] \geq 1 - \delta\right] \geq 1 - \eta$$

A runs in time $\mathrm{poly}(n, t(n), s(n), \beta(c, \gamma)^{-1}, \delta^{-1})$.

Proof. To construct A, we will follow three steps:

1. Construct a weak randomized predictor L.
2. Argue that many good non-uniform but deterministic predictors exist, by fixing coins and samples for L.
3. Construct a deterministic predictor by sampling and then testing enough non-uniform predictors.

Steps 2 and 3 follow from standard techniques (i.e., "constructive averaging").

Claim 3.1 (Weak randomized predictor)**.** *Under the conditions of Theorem 3.3, there exists a randomized algorithm L, running in time* $\mathrm{poly}(n, t(n), s(n), \beta(c, \gamma)^{-1}, \delta^{-1}, \eta^{-1})$*, such that for any $n \in \mathbb{N}, \delta, \eta > 0$, the following equation is satisfied:*

$$\Pr_{\pi_f\sim\mu}\left[\Pr_{L}\left[\forall\rho : \Pr_{z\sim\rho}\left[L^{\mathrm{Ex}(f,\rho)}(z, n, \delta, \eta) \neq f(z)\right] \leq \frac{1}{2} - \Omega(\beta(c, \gamma))\right] \geq 1 - \delta\right] \geq 1 - \eta \tag{9}$$

Proof of Claim 3.1. See the randomized predictor L in Figure 1. Recall that Φ_n is the evaluation algorithm for the concept class $\mathfrak{C}$. To reduce notation, we do not write the subscript of the Φ_n.

Consider the distribution $\mathcal{M}$ over 2×2 matrices

$$C = \begin{bmatrix} \Phi(\pi_f, z) & \Phi(\pi_f, w) \\ \Phi(\pi_g, z) & \Phi(\pi_g, w) \end{bmatrix}$$

where $\pi_f, \pi_g \sim \mu$ and $z, w \sim \rho$. We now claim that, under the conditions of Theorem 3.1, this distribution is efficiently *distinguishable* from the distribution $\mathcal{R}$ over random 2×2 matrices,

$$R = \begin{bmatrix} r_{00} & r_{01} \\ r_{10} & r_{11} \end{bmatrix}$$

To see this, observe that the distribution over C is identical to the following distribution over 2×2 matrices

$$D = \begin{bmatrix} \Phi(\mu(x), \rho(y)) & \Phi(\mu(x), \rho(y')) \\ \Phi(\mu(x'), \rho(y)) & \Phi(\mu(x'), \rho(y')) \end{bmatrix}$$

Algorithm 1 $L^{\mathrm{EX}(f,\rho)}$

Input: $z \in I^n, n \in \mathbb{N}, \delta, \eta > 0$
Pick uniformly random values $b_1, b_2 \in \{0,1\}$.
Pick uniformly random values $r_{00}, r_{01}, r_{10}, r_{11} \in \{-1,1\}$
Sample $\pi_g \sim \mu$.
Sample $(w,y) \sim \mathrm{EX}(f,\rho)$.
if $b_1 = b_2 = 0$ **then**
    $v \leftarrow \prod_{i,j\in\{0,1\}} r_{ij}$
if $b_1 = 0, b_2 = 1$ **then**
    $v \leftarrow y \cdot r_{00} \cdot \prod_{i,j\in\{0,1\}} r_{ij}$
if $b_1 = 1, b_2 = 0$ **then**
    $v \leftarrow \Phi(\pi_g, w) \cdot \Phi(\pi_g, z) \cdot \prod_{j\in\{0,1\}} r_{1j}$
if $b_1 = 1, b_2 = 1$ **then**
    $v \leftarrow y \cdot \Phi(\pi_g, w) \cdot \Phi(\pi_g, z) \cdot r_{11}$
$b \leftarrow r_{b_1 b_2}$
Output $b \cdot v$

Figure 1: Randomized predictor L. Observe that L needs to sample a concept representation π_g from μ. This is, in a nutshell, the reason that we obtain average-case learning, and μ must also be fixed (i.e., known to L).

for x, x', y, y' being uniformly random strings. We can assume that without loss of generality that x, x', y, y' are all the same length, by defaulting to the maximum necessary length for sampling μ, ρ (and padding the shorter strings with useless random bits). Therefore, identifying $\xi(x,y) := \Phi(\mu(x), \rho(y))$, we can now see that

$$R_2(\xi) = \mathop{\mathbb{E}}_{\substack{\pi_f,\pi_g \\ z,w}} \left[\prod_{i,j\in\{0,1\}} C_{ij} \right]$$

It now readily follows that when $\mathfrak{G}[\mathfrak{C}, \Phi, n] \in \mathrm{r}\Pi[2, c, \gamma]$ (which is true by assumption), then

$$R_2(\xi) = \mathop{\mathbb{E}}_{\substack{\pi_f,\pi_g \\ z,w}} \left[\prod_{i,j\in\{0,1\}} C_{ij} \right] \geq \beta(c,\gamma) \tag{10}$$

This is justified by the fact that party one can locally compute the mapping $\mu(x)$, while party two can compute $\rho(y)$, and then they can play a winning communication game with c bits of communication and bias γ to complete a c-bit randomized protocol for ξ with bias γ.

On the other hand,

$$\mathop{\mathbb{E}}_{R} \left[\prod_{i,j\in\{0,1\}} R_{ij} \right] = 0$$

Now that we have established this, we may proceed by a hybrid argument. Define the neighboring hybrid distributions H_1, H_2, H_3, H_4, H_5 over 2×2 matrices, as in Figure 2.

It then follows that for random hybrid neighbors H_i, H_{i+1} ($i \in [4]$),

$$\mathop{\mathbb{E}}_{i\sim[4]} \left[\mathop{\mathbb{E}}_{H'\sim H_{i+1}} \left[\prod_{k,j\in\{0,1\}} H'_{kj} = 1 \right] - \mathop{\mathbb{E}}_{H\sim H_i} \left[\prod_{k,j\in\{0,1\}} H_{kj} = 1 \right] \right] \geq \beta(c,\gamma)/4 \tag{11}$$

$$
\begin{aligned}
H_1 &= \mathcal{R} \\
H_2 &= \begin{cases} C_{k\ell} \text{ when } k = 0, \ell = 0 \\ R_{k\ell} \text{ otherwise} \end{cases} \\
H_3 &= \begin{cases} C_{k\ell} \text{ when } k = 0, \ell \leq 1 \\ R_{k\ell} \text{ otherwise} \end{cases} \\
H_4 &= \begin{cases} C_{k\ell} \text{ when } k = 0 \text{ or } \ell \leq 0 \\ R_{k\ell} \text{ otherwise} \end{cases} \\
H_5 &= \mathcal{M}
\end{aligned}
$$

Figure 2: Hybrid sequence.

To ease notation, let $D(H) = \prod_{k,j \in \{0,1\}} H_{kj}$, and let V_i denote the event that $D(H_i) = 1$. Intuitively, the function D stands for "distinguisher," and can be thought of as such.

We continue by observing that, by definition, the value stored as v in L (Algorithm 1) is $D(H_i)$ for a random $i \in [4]$. Hence, the output of L, which is written as $D(H_i) \cdot b$, is interpreted as a prediction, where $b = r_{b_1 b_2}$ is the "guess bit." Note that, the string $b_1 b_2$ is the binary representation of i.

Now, conditioning on correctness of this guess bit, we have that for all ρ, and probabilities taken over $z \sim \rho, f \sim \mu$ and the randomness of L:

$$
\begin{aligned}
\Pr\left[L^{\mathrm{Ex}(f,\rho)}(z,n,\delta,\eta) = f(z)\right] &= \Pr\left[L^{\mathrm{Ex}(f,\rho)}(z,n,\delta,\eta) = f(z) \mid b = f(z)\right] \cdot \Pr[b = f(z)] \\
&\quad + \Pr\left[L^{\mathrm{Ex}(f,\rho)}(z,n,\delta,\eta) = f(z) \mid b \neq f(z)\right] \cdot \Pr[b \neq f(z)] \\
&= \frac{1}{2}\Big(\Pr[b \cdot D(H_i) = f(z) \mid b = f(z)] \\
&\quad + \Pr[b \cdot D(H_i) = f(z) \mid b \neq f(z)]\Big)
\end{aligned}
$$

Indeed, when V_i is unsatisfied, this means that the output of L is b. The case analysis follows:

$$
\begin{aligned}
\Pr[L^{\mathrm{Ex}(f,\rho)}(z,n,\delta,\eta) = f(z)] &= \frac{1}{2}\Big(\Pr[V_i \mid b = f(z)] + \Pr[\neg V_i \mid b \neq f(z)]\Big) \\
&= \frac{1}{2} + \frac{1}{2}\Big(\Pr[V_i \mid b = f(z)] - \Pr[V_i \mid b \neq f(z)]\Big)
\end{aligned}
$$

By conditioning, we know that:

$$
\Pr[V_i] = \frac{1}{2}\Pr[V_i \mid b = f(z)] + \frac{1}{2}\Pr[V_i \mid b \neq f(z)]
$$

rearranging the terms, we get:

$$
\frac{1}{2}\Pr[V_i \mid b \neq f(z)] = \Pr[V_i] - \frac{1}{2}\Pr[V_i \mid b = f(z)]
$$

We thus conclude:

$$
\Pr[L^{\mathrm{Ex}(f,\rho)}(z,n,\delta,\eta) = f(z)] = \frac{1}{2} + \underbrace{\Pr[V_i \mid b = f(z)]}_{(\alpha)} - \underbrace{\Pr[V_i]}_{(\beta)}
$$

Algorithm 2 $A^{\mathrm{Ex}(f,\rho)}$

Input: $n \in \mathbb{N}, \delta \in (0,1]$
Sample m (sufficiently many) candidate circuits, using oracle access to $\mathrm{Ex}(f,\rho)$ as needed.
Sample t (sufficiently many) additional random examples from $\mathrm{Ex}(f,\rho)$.
for each sampled circuit C_i, **do**
    Compute using random examples: $\alpha_i \leftarrow \mathrm{Cor}(f, C_i)$
output the circuit with largest α value.

Figure 3: Algorithm for sampling and testing candidate predictors.

The term (α) corresponds to the case that L computes the 2-party norm on a sample from H_{i+i} (i.e., the product of the entries of a matrix sampled from H_{i+i}), while term (β) is the case that L computes the 2-party norm on a sample from H_i (the product of the entries of a matrix sampled from H_i). Thus, by equation (11),

$$\begin{aligned}\Pr[L^{\mathrm{Ex}(f,\rho)}(z,n,\delta,\eta) = f(z)] &= \frac{1}{2} + (1 - \Pr[D(H_{i+1}) = 1] - (1 - \Pr[D(H_i) = 1]) \\ &\geq \frac{1}{2} + \beta(c,\gamma)/8\end{aligned}$$

□

Having established Claim 3.1, we now convert the randomized algorithm $L^{\mathrm{Ex}(f,\rho)}$ into a non-uniform learning algorithm by averaging. Let $L^{\mathrm{Ex}(f,\rho)}_{\mathrm{inp},i}(z;r)$ denote the Algorithm 1 where the random hybrid choice is fixed to be i, and the input parameters $\mathrm{inp} = (n,\delta,\eta)$ are hard-wired in, and the random bits r for computing other randomized aspects of the algorithm is treated as input. This allows us to consider the algorithm as a deterministic mapping of random bits and examples from $\mathrm{Ex}(f,\rho)$ to a circuit that weakly agrees with f. By a standard averaging argument, we obtain:

Claim 3.2 (Averaging, see lemma A.11 of [AB09])**.**

$$\Pr_r\left[\Pr_{z\sim\rho}\left[L^{\mathrm{Ex}(f,\rho)}_{\mathrm{inp},i}(z;r) = f(z) \mid r\right] > \frac{1}{2} + \frac{\beta(c,\gamma)}{32}\right] > \beta(c,\gamma)/4$$

Taking hybrid index i and r uniformly at random, we obtain "good" choices with good probability. Therefore such a circuit is efficiently found by randomized trial-and-error; we sample many candidate predictors in parallel and then compare each to the concept by checking random examples. By a standard application of Chernoff bounds, sufficiently many examples will be enough to check that a circuit with good enough accuracy is indeed good enough, with high probability.

Claim 3.3 (Without proof)**.** *With probability $1-\delta$, A (Algorithm 2 in Figure 3) outputs a "good" circuit that correctly classifies $1/2 + \Omega(\beta(c,\gamma))$ fraction of points, where t, m are quantities that are polynomially bounded as a function of $\beta(c,\gamma)$ and $\log(\delta^{-1})$.*

From the above claims it now follows, from a Markov argument, that:

$$\Pr_{\pi_f\sim\mu}\left[\Pr_A\left[\forall\rho : \Pr_{z\sim\rho}\left[h(z) \neq f(z) : h \leftarrow A^{\mathrm{Ex}(f,\rho)}(n,\delta)\right] \leq \frac{1}{2} - \Omega(\beta(c,\gamma))\right] \geq 1 - \delta/\eta\right] \geq 1 - \eta$$

This concludes the proof of Theorem 3.1.

□

Remark. Using the 2-party norm as we do is a *universal* distinguisher; namely it distinguishes any $f \in \mathrm{rII}[2, c, \gamma]$ from a random function (using the lower bound of $\beta(c, \gamma)$). Therefore it holds that for *arbitrary* choice of distribution ρ, we obtain the desired guarantee. Indeed, the choice of ρ can be adversarial with respect to $f \sim \mu$, and it never needs to be known by A.

3.2.1 Replacing the Generic Bound

We can now replace the generic bound for $R_2(f)$ with the known bound, $R_2(f) \geq (2^{-c}\gamma)^4$ (see Theorem 2.2). Thus, we obtain the following more concrete implication.

Theorem 3.2. *Suppose $\mathfrak{G}[\mathfrak{C}, \Phi, n] \in \mathrm{rII}[2, c, \gamma]$, and fix a time $t(n)$ samplable distribution μ over concepts representations of $f \in \mathfrak{C}$. Then, there exists an algorithm A such that, for any $n \in \mathbb{N}, \delta, \eta > 0$,*

$$\Pr_{\pi_f \sim \mu}\left[\Pr_{A}\left[\forall \rho : \Pr_{x \sim \rho}\left[h(x) \neq f(x) : h \leftarrow A^{\mathrm{Ex}(f,\rho)}(n, \delta)\right] \leq \frac{1}{2} - \Omega((\gamma 2^{-c})^4)\right] \geq 1 - \delta\right] \geq 1 - \eta$$

A runs in time $\mathrm{poly}(n, t(n), s(n), \gamma^{-1}2^c, \delta^{-1})$.

3.3 Strong Learning

Celebrated results of computational learning theory, indicate that efficient weak and strong PAC-learning are equivalent [Sch90] (in the "filtering" setting, this is shown by e.g. [DW+00]). Thus, Theorem 3.1 is enough prove Theorem 1.1. Recall that $\mathfrak{C} = \{\mathfrak{C}_n\}_{n \in \mathbb{N}}$ is a boolean concept class that is $s(n)$-representable by the evaluation function $\Phi = \{\Phi_n\}_{n \in \mathbb{N}}$.

Theorem 3.3 (Theorem 1.1, restated)**.** *Suppose $\mathfrak{G}[\mathfrak{C}, \Phi, n] \in \mathrm{rII}[2, c, \gamma]$, and fix a time $t(n)$ samplable distribution μ over concepts representations of $f \in \mathfrak{C}$. Then, there exists an algorithm A such that, for any $n \in \mathbb{N}, \varepsilon, \delta > 0$,*

$$\Pr_{\pi_f \sim \mu}\left[\Pr_{A}\left[\forall \rho : \Pr_{x \sim \rho}\left[h(x) \neq f(x) : h \leftarrow A^{\mathrm{Ex}(f,\rho)}(n, \varepsilon, \delta, \eta)\right] \leq \varepsilon\right] \geq 1 - \delta\right] \geq 1 - \eta \tag{12}$$

A runs in time $\mathrm{poly}(n, t(n), s(n), \gamma^{-1}2^c, \varepsilon^{-1}, \delta^{-1}, \eta^{-1})$.

Proof. We combine the equivalence of weak and strong PAC-learning [Sch90, DW+00] with Theorem 3.1 to conclude the desired expression. Note that, importantly, our average-case weak learner works for all ρ after taking the probability over $f \sim \mu$ and the randomness of A. If the quantifiers were in another order, then we could not guarantee boosting since there would be no guarantee that the same set of functions $f \in \mathfrak{C}$ could be learned. □

3.4 Unconditional Learning

In this section, we will apply Theorem 3.3 to concrete settings, to obtain (quasi-)polynomial time average-case PAC-learning algorithms. Theorem 3.4 restates Theorem 1.2 and adds item *(3)*.

Theorem 3.4 (Average-case PAC-learning — Theorem 1.2 restated)**.** *Let $\mathfrak{C}$ be a concept class that is* $\mathrm{poly}(n)$*-representable by Φ. Then, if Φ is contained in either of the following evaluation classes, then $\mathfrak{C}$ is efficiently average-case PAC-learnable with respect to any fixed polynomial time samplable μ:*

(1) $\mathsf{MAJ} \circ \mathsf{THR}$.

(2) $\mathsf{DT}[O(1), \mathsf{THR}]$. *That is, constant depth decision trees with* THR*-nodes.*

If Φ is contained in the following evaluation class, then $\mathfrak{C}$ is average-case PAC-learnable in quasipolynomial time with respect to any fixed quasipolynomial time samplable μ:

(3) $\mathsf{MAJ} \circ \mathsf{DT}[\text{polylog}(n), \mathsf{THR}]$. *That is, a quasi-polynomial fan-in majority vote of polylogarithmic depth decision trees with* THR*-nodes.*

To prove Theorem 3.4, we use a theorem credited to Nisan from [Nis93].

Theorem 3.5 ([Nis93])**.** $\mathsf{MAJ} \circ \mathsf{THR} \subseteq \mathrm{r}\Pi[2, O(\log(n)), 1/\text{poly}(n)]$. $\mathsf{DT}[d, \mathsf{THR}] \subseteq \mathrm{r}\Pi[2, d \cdot O(\log(n)), 1/6]$.

Sketch. The statement follows from combining a few arguments in [Nis93].

First, Nisan proves that a single THR gate is contained in $\mathrm{r}\Pi[2, O(\log(n)), 1/2 - 1/\text{poly}(n)]$ (Thm 1a.). Then, he shows that any depth $d = O(1)$ decision tree with node queries computed by $f \in \mathrm{r}\Pi[2, O(\log(n)), 1/2 - 1/\text{poly}(n)]$ is contained in $\mathrm{r}\Pi[2, O(\log(n)), 1/6]$ (Lemma 4). Therefore $\mathsf{DT}[O(1), \mathsf{THR}] \in \mathrm{r}\Pi[2, O(\log(n)), 1/6]$.

Second, Nisan shows that the majority vote of any $f \in \mathrm{r}\Pi[2, O(\log(n)), 1/2 - 1/\text{poly}(n)]$ is contained in $\mathrm{r}\Pi[2, O(\log(n)), 1/\text{poly}(n)]$ (Lemma 5).

Therefore we get that $\mathsf{MAJ} \circ \mathsf{THR} \in \mathrm{r}\Pi[2, O(\log(n)), 1/\text{poly}(n)]$. □

Proof of Theorem 3.4. We will establish that each evaluation class has a good randomized communication protocol, as this implies that $\mathfrak{G}[\mathfrak{C}, \Phi, n]$ does as well, whenever Φ is in the evaluation class. This is enough to conclude by Theorem 3.3.

- *(1)* and *(2)* follow from Theorem 3.5 and then invoking Theorem 3.3.
- For *(3)* we have that by Theorem 3.5, $\mathsf{DT}[\text{polylog}(n), \mathsf{THR}] \in \mathrm{r}\Pi[2, \text{polylog}(n), 1/6]$. Then, by standard error reduction of randomized protocols, we get

  $$\mathsf{DT}[\text{polylog}(n), \mathsf{THR}] \in \mathrm{r}\Pi[2, \text{polylog}(n), 1/2 - 1/\text{poly}(n)]$$

  This means that by Theorem 3.5, $\mathsf{MAJ} \circ \mathsf{DT}[\text{polylog}(n), \mathsf{THR}] \in \mathrm{r}\Pi[2, \text{polylog}(n), 1/\text{poly}(n)]$. Theorem 3.3 then gives *(3)*.

□

4 Implications of Tighter Correlation Bounds and XOR Lemmas

In this section, we will step back from considering what learning algorithms we can *unconditionally* obtain, and consider what learning algorithms we can *conditionally* obtain.

The learning algorithm in the previous section depends crucially on the lower bound for the 2-party norm. We constructed the learning algorithms based on the generic bound, and instantiated concrete algorithms based on Theorem 2.2 (Section 3). For the reader's convenience, we restate the concrete bounds:

$$(2^{-c} \cdot \mathrm{Cor}(f, \Pi[2, c]))^4 \leq R_2(f) \qquad \text{(Theorem 2.1)}$$

and analogously (for $f \in \mathrm{r}\Pi[2, c, \gamma]$)

$$(2^{-c} \cdot \gamma)^4 \leq R_2(f) \qquad \text{(Theorem 2.2)}$$

Indeed, it should follow that if we can *shrink* the 2^c factor inside (Theorem 2.1) in particular (since (Theorem 2.2) follows), we could obtain efficient learning algorithms for even more complex functions. The rest of this section will explore this possibility — what are the implications of reducing the dependency on c in (Theorem 2.1)?

4.1 On Small Improvements on the 2^c Factor

We will first show that improving the correlation bound even slightly will imply break-through learning algorithms for the notorious learning $O(\log n)$-juntas problem (c.f. [Blu03]).

One well-known example of a learning $O(\log n)$-juntas problem is the XOR-MAJ (quasi-polynomial security) weak PRF candidate of [BFKL93] (see Section 2.3 for a definition of weak PRFs). [BFKL93] claimed 30 years ago that "any method that could even weakly predict [in polynomial time] such functions over a uniform distribution would require profoundly new ideas." At the moment, there is still essentially no reason to believe that such methods will soon be developed.

Definition 4.1 ([BFKL93] XOR-MAJ weak PRF candidate)**.** *The* XOR-MAJ *weak PRF candidate is the pair of algorithms* $(\mathsf{XM}, \mathsf{XMKeyGen})$ *defined as follows.*

- $\mathsf{XMKeyGen}(1^n)$ *outputs the key* $k = (A, B) \subseteq [n]$*, consisting of uniformly random disjoint sets* $A, B \in [n]$ *of size* $\log n$ *each.* $k = (A, B)$ *can be considered a bitstring of length* $2\log^2(n)$*.*
- $\mathsf{XM}(k, x)$ *takes as input the key and a string* $x \in I^n$*.* $\mathsf{XM}(k, x)$ *is defined by*

$$\mathsf{XM}(k, x) = \mathsf{XOR}(\mathsf{XOR}(x|_A), \mathsf{MAJ}(x|_B))$$

  Here, $x|_S$ *is the projection of a string* $x \in I^n$ *to the coordinates indicated by a set* $S \subseteq [n]$*.*

Let $\mathfrak{C}_{\mathsf{XM}}$ denote the set $\{\mathsf{XM}(k, x) : \forall k \text{ in the keyspace}\}$. We can now see that, viewing $\mathfrak{C}_{\mathsf{XM}}$ as a concept class, it is $2\log^2(n)$-representable by XM. The $2\log^2(n)$ bits correspond to the indices of the relevant variables of the instance. Furthermore, XMKeyGen effectively defines a target distribution over $\mathfrak{C}_{\mathsf{XM}}$.

Theorem 4.1. *Suppose that, for every* f*, it is the case that* $(2^{-c^{1/2}} \cdot \mathrm{Cor}(f, \Pi[2, c]))^4 \leq R_2(f)$*. Then* $\mathfrak{C}_{\mathsf{XM}}$ *is efficiently average-case PAC-learnable with respect to* XMKeyGen*.*

Proof. If for every $f, (2^{-c^{1/2}} \cdot \mathrm{Cor}(f, \Pi[2, c]))^4 \leq R_2(f)$, then when $f \in \mathrm{r}\Pi[2, c, \gamma]$, it follows that $(2^{-c^{1/2}} \cdot \gamma)^4 \leq R_2(f)$. This is true because $f \in \mathrm{r}\Pi[2, c, \gamma]$ means f has a distributional protocol with respect to the uniform distribution with cost of c bits and bias γ, so $\mathrm{Cor}(f, \Pi[2, c]) \geq \gamma$.

Now, observe that $\mathfrak{G}[\mathfrak{C}_{\mathsf{XM}}, \mathsf{XM}, n] \in \mathrm{r}\Pi[2, O(\log^2 n), 1/2]$. Here, XM is the evaluation function. This is because the party holding the representation can just send the $2\log^2 n$ bits to the party holding the input, who then evaluates on the input. In fact, this protocol is deterministic; it has $\gamma = 1/2$ (no error). Finally, let $\beta(c, \gamma) = (2^{-c^{1/2}} \cdot \gamma)^4 = O(2^{-4c^{1/2}})$. The proof is completed by applying Theorem 3.3.

□

4.1.1 Improving Viola and Wigderson's XOR Lemma

Viola and Wigderson [VW07] used Theorem 2.1 to show an XOR lemma for communication protocols (for the special case of 2-party communication [Sha01] did similarly).

Define the m-wise direct XOR of $f : (I^n)^2 \to \{-1, 1\}$ as $f^{\times m} : ((I^n)^2)^m \to \{-1, 1\}$ for $f^{\times m}(x_1^1, x_1^2, \cdots, x_m, x_m^2) = \prod_{i=1}^m f(x_i^1, x_i^2)$.

Theorem 4.2 ([VW07] XOR lemma; $k = 2$ case)**.** *Let* $f : (I^n)^2 \to \{-1, 1\}$ *be a function such that* $\mathrm{Cor}(f, \Pi[2, 2]) \leq \varepsilon$*. Then* $\mathrm{Cor}(f^{\times m}, \Pi[2, c]) \leq 2^c \cdot \varepsilon^{m/4}$*.*

A question from [VW07]. Regarding this theorem, the main question posed by [VW07] was whether the parameters of the XOR lemma are the best possible. In particular, they asked if the 2^c factor can be eliminated. They were unable to answer the question, except for showing a counterexample

to the "ideal" relationship. Namely, going from correlation ε to correlation εm, for $k = 2$ parties communicating $c = 2$ bits.

Also in the $k = 2$ case, we will now prove that a small improvement on the 2^c factor in Theorem 4.2 implies XOR-MAJ is predictable. In other words, we show that based on the security of the XOR-MAJ weak PRF, there exists a counterexample to the below improvement of Theorem 4.2.

(i) For all $m \in \mathbb{N}$ and functions f satisfying $\mathrm{Cor}(f, \Pi[2,2]) \leq \varepsilon$, we have that $\mathrm{Cor}(f^{\times m}, \Pi[2,c]) \leq 2^{c^{1/2}} \cdot \varepsilon^{m/4}$.

Theorem 4.3. *Suppose that the* XOR-MAJ *weak PRF candidate has quasi-polynomial security. Then for every $m \in \mathbb{N}$, there exists f such that* $\mathrm{Cor}(f, \Pi[2,2]) \leq \varepsilon$, *while* $\mathrm{Cor}(f^{\times m}, \Pi[2,c]) > 2^{c^{1/2}} \cdot R_2(f)^4$.

Proof of Theorem 4.3. We prove the contrapositive. Suppose there exists m such that for all f,

$$\mathrm{Cor}(f, \Pi[2,2]) \leq \varepsilon \implies \mathrm{Cor}(f^{\times m}, \Pi[2,c]) \leq 2^{c^{1/2}} \cdot \varepsilon^{m/4}$$

Theorem 2.1 implies that for any f we may set $\varepsilon = 4R_2(f)^{1/4}$. Therefore, it follows that

$$\mathrm{Cor}(f^{\times m}, \Pi[2,c]) \leq 2^{c^{1/2}} \cdot 4R_2(f^{\times m})^{1/16} \tag{13}$$

(13) uses the fact that R_k is multiplicative over the direct XOR (see [VW07] fact 3.7). This implies that

$$(2^{-c^{1/2}-2} \cdot \mathrm{Cor}(f^{\times m}, \Pi[2,c]))^{16} \leq R_2(f^{\times m}) \tag{14}$$

What we have in (14) is essentially an improvement of *the* correlation bound for any function of the form $f^{\times m}$.

Putting this on hold, let $\mathfrak{C}_{\mathsf{XM}}^{\times m} = \{f^{\times m} : f \in \mathfrak{C}_{\mathsf{XM}}\}$. It is the case that $\mathfrak{C}_{\mathsf{XM}}^{\times m}$ is $m \log^2(n)$-representable by $\mathsf{XM}^{\times m}$. To see this, observe that to represent $f^{\times m} \in \mathfrak{C}_{\mathsf{XM}}^{\times m}$, we need m instances of a key $k \sim \mathsf{XMKeyGen}$.

By the same argument as in Theorem 4.1, we can thus conclude that $\mathfrak{G}[\mathfrak{C}_{\mathsf{XM}}^{\times m}, \mathsf{XM}^{\times m}, n] \in \mathrm{rI\!I}[2, m \log^2 n, 1/2]$. Therefore, using the improved bound in (14), we can again apply Theorem 3.3 to get that $\mathfrak{C}_{\mathsf{XM}}^{\times m}$ is average-case PAC-learnable with respect to μ, where μ is defined as m independent samples of the XMKeyGen distribution. Finally, by observing that one can easily reduce distinguishing the XOR-MAJ weak PRF to learning $\mathfrak{C}_{\mathsf{XM}}^{\times m}$ with respect to μ, this learning algorithm violates the pseudo-randomness of the XOR-MAJ weak PRF. □

4.2 Conditional Impossibility of Dramatic Improvements for 2^c Factor

In this section, we will prove that more dramatic improvement of the 2^c term in *the* correlation bound and/or the [VW07] XOR lemma is impossible under standard cryptographic assumptions.

4.2.1 Improving the Correlation Bound

Theorem 4.4. *Consider the following bounds, for every f,*

$$(2^{-c^{o(1)}} \cdot \mathrm{Cor}(f, \Pi[k,c]))^4 \leq R_2(f) \tag{i}$$

$$(1/\mathrm{poly}(c) \cdot \mathrm{Cor}(f, \Pi[k,c]))^4 \leq R_2(f) \tag{ii}$$

(i) *implies that subexponentially secure one-way functions do not exist, and* (ii) *implies one-way functions do not exist.*

Proof. Our argument is basically analogous to proof of Theorem 4.1. By Theorem 3.1, we can replace the generic bound $\beta(c,\gamma)$ by (i) ((ii) analogously), and then obtain a learning algorithm that weakly predicts any polynomial size circuit, in sub-subexponential time (polynomial, for (ii)). Therefore, by invoking generic equivalences of one-way functions and pseudorandom functions [GGM86, HILL99], this refutes the existence of subexponentially secure one-way functions and normal one-way functions, respectively. □

4.2.2 Improving the XOR Lemma

We can continue the study of the question of [VW07]; we use one-way functions to obtain counterexamples to dramatic improvements of the XOR lemma.

Theorem 4.5 (XOR lemma counterexamples from one-way functions)**.**

(1) Suppose that a subexponentially secure one-way function exists. Then for every $m \in \mathbb{N}$, there exists a function f, and constant $\tau > 0$ such that $\mathrm{Cor}(f, \Pi[2,2]) \leq \varepsilon$, while $\mathrm{Cor}(f^{\times m}, \Pi[2,c]) > 2^{c^\tau} \cdot R_2(f)^{1/4}$.

(2) Suppose that one-way functions exist. Then for every $m \in \mathbb{N}$, there exists a function f such that for every polynomial $p : \mathbb{N} \to \mathbb{N}$, $\mathrm{Cor}(f, \Pi[2,2]) \leq \varepsilon$, while $\mathrm{Cor}(f^{\times m}, \Pi[2,c]) > p(c) \cdot R_2(f)^{1/4}$.

Proof. We prove the statement for *(2)*, as *(1)* is exactly analogous. We prove *(2)* similarly to Theorem 4.3 by considering the contrapositive.

Suppose that there exists $m \in \mathbb{N}$, such that for all functions f, there exists a polynomial q, such that $\mathrm{Cor}(f, \Pi[2,2]) \leq \varepsilon$ implies that $\mathrm{Cor}(f^{\times m}, \Pi[2,c]) \leq q(c) \cdot \varepsilon^{m/4}$. Then, Theorem 2.1 implies that for any f we may set $\varepsilon = 4R_2(f)^{1/4}$. Therefore, it follows that there exists m such that

$$\mathrm{Cor}(f^{\times m}, \Pi[2,c]) \leq q(c) \cdot 4R_2(f^{\times m})^{1/16} \tag{15}$$

This implies that there is q' such that

$$1/q'(c) \cdot \mathrm{Cor}(f^{\times m}, \Pi[2,c])^{16} \leq R_2(f^{\times m}) \tag{16}$$

(15) uses the fact that R_k is multiplicative over the direct XOR (see [VW07] fact 3.7). Thus, what we have in (16), is essentially an improvement of *the* correlation bound for any function of the form $f^{\times m}$.

Fix any concept class and evaluation scheme $\mathfrak{C}, \Phi$, and let $\mathfrak{C}^{\times m}, \Phi^{\times m}$ be the naturally corresponding m-wise XOR concept class and evaluation function. Now consider that even when $c = mn$, if $\mathfrak{G}[\mathfrak{C}^{\times m}, \Phi^{\times m}, n] \in \mathrm{r}\Pi[2,c,\gamma]$, then $\gamma^{16}/q'(c) \leq R_2(f^{\times m})$. But for any concept class $\mathfrak{C}$, $\mathfrak{G}[\mathfrak{C}^{\times m}, \Phi^{\times m}, n] \in \mathrm{r}\Pi[2,c,1/2]$ trivially. This means there exists polynomial q'' such that $1/q''(c) \leq R_2(f^{\times m})$.

Together with the observation that refuting existence of pseudorandom functions of the form $f^{\times m}$ with suffices to refute one-way functions, we can apply Theorem 4.4 to conclude the proof. □

Acknowledgements

I thank Mark Bun, Ran Canetti, Russell Impagliazzo, and Emanuele Viola for thoughtful conversations about this research. I also thank Mauricio Karchmer for advice on presentational aspects of this paper. Finally, I give special thanks to Marco Carmosino for helpful comments on a draft of this paper, as well as many discussions pertaining to this research.

References

[AB09] Sanjeev Arora and Boaz Barak. *Computational complexity: a modern approach*. Cambridge University Press, 2009.

[AD97] Miklós Ajtai and Cynthia Dwork. A public-key cryptosystem with worst-case/average-case equivalence. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing*, pages 284–293, 1997.

[BCG+21] Elette Boyle, Geoffroy Couteau, Niv Gilboa, Yuval Ishai, Lisa Kohl, and Peter Scholl. Low-complexity weak pseudorandom functions in $\mathsf{ac}^0[\mathsf{mod2}]$. In *Annual International Cryptology Conference*, pages 487–516. Springer, 2021.

[BFKL93] Avrim Blum, Merrick Furst, Michael Kearns, and Richard J Lipton. Cryptographic primitives based on hard learning problems. In *Annual International Cryptology Conference*, pages 278–291. Springer, 1993.

[BH12] Paul Beame and Trinh Huynh. Multiparty communication complexity and threshold circuit size of ac^0. *SIAM Journal on Computing*, 41(3):484–518, 2012.

[Blu03] Avrim Blum. Open problems-learning a function of r relevant variables. *Lecture Notes in Computer Science*, 2777:731–733, 2003.

[BNS92] László Babai, Noam Nisan, and Márió Szegedy. Multiparty protocols, pseudorandom generators for logspace, and time-space trade-offs. *Journal of Computer and System Sciences*, 45(2):204–232, 1992.

[BVdW07] Harry Buhrman, Nikolay Vereshchagin, and Ronald de Wolf. On computation and communication with small bias. In *Twenty-Second Annual IEEE Conference on Computational Complexity (CCC'07)*, pages 24–32. IEEE, 2007.

[Che18] Lijie Chen. Toward super-polynomial size lower bounds for depth-two threshold circuits. *arXiv preprint arXiv:1805.10698*, 2018.

[CIKK16] Marco L Carmosino, Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. Learning algorithms from natural proofs. In *31st Conference on Computational Complexity (CCC 2016)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

[CT93] Fan RK Chung and Prasad Tetali. Communication complexity and quasi randomness. *SIAM Journal on Discrete Mathematics*, 6(1):110–123, 1993.

[DW+00] Carlos Domingo, Osamu Watanabe, et al. Madaboost: A modification of adaboost. In *COLT*, pages 180–189, 2000.

[FGKP06] Vitaly Feldman, Parikshit Gopalan, Subhash Khot, and Ashok Kumar Ponnuswami. New results for learning noisy parities and halfspaces. In *2006 47th Annual IEEE Symposium on Foundations of Computer Science (FOCS'06)*, pages 563–574. IEEE, 2006.

[FX14] Vitaly Feldman and David Xiao. Sample complexity bounds on differentially private learning via communication complexity. In *Conference on Learning Theory*, pages 1000–1019. PMLR, 2014.

[GGH97] Oded Goldreich, Shafi Goldwasser, and Shai Halevi. Eliminating decryption errors in the ajtai-dwork cryptosystem. In *Crypto*, volume 97, pages 105–111, 1997.

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM (JACM)*, 33(4):792–807, 1986.

[GK23] Halley Goldberg and Valentine Kabanets. Improved learning from kolmogorov complexity. *ECCC Report*, 2023.

[HILL99] Johan Håstad, Russell Impagliazzo, Leonid A Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999.

[KLMY19] Daniel Kane, Roi Livni, Shay Moran, and Amir Yehudayoff. On communication complexity of classification problems. In *Conference on Learning Theory*, pages 1903–1943. PMLR, 2019.

[KN96] Eyal Kushilevitz and Noam Nisan. Communication complexity, 1996.

[KNR99] Ilan Kremer, Noam Nisan, and Dana Ron. On randomized one-round communication complexity. *Computational Complexity*, 8:21–49, 1999.

[KOS04] Adam R Klivans, Ryan O'Donnell, and Rocco A Servedio. Learning intersections and thresholds of halfspaces. *Journal of Computer and System Sciences*, 68(4):808–840, 2004.

[LS09] Nati Linial and Adi Shraibman. Learning complexity vs communication complexity. *Combinatorics, Probability and Computing*, 18(1-2):227–245, 2009.

[Nan21] Mikito Nanashima. A theory of heuristic learnability. In *Conference on Learning Theory*, pages 3483–3525. PMLR, 2021.

[Nis93] Noam Nisan. The communication complexity of threshold gates. *Combinatorics, Paul Erdos is Eighty*, 1:301–315, 1993.

[NW94] Noam Nisan and Avi Wigderson. Hardness vs randomness. *Journal of computer and System Sciences*, 49(2):149–167, 1994.

[NY15] Moni Naor and Eylon Yogev. Bloom filters in adversarial environments. In *Advances in Cryptology–CRYPTO 2015: 35th Annual Cryptology Conference, Santa Barbara, CA, USA, August 16-20, 2015, Proceedings, Part II*, pages 565–584. Springer, 2015.

[Raz87] Alexander A Razborov. Lower bounds on the size of bounded depth circuits over a complete basis with logical addition. *Mathematical Notes of the Academy of Sciences of the USSR*, 41(4):333–338, 1987.

[Raz00] Ran Raz. The bns-chung criterion for multi-party communication complexity. *Computational Complexity*, 9(2):113–122, 2000.

[RR97] Alexander A Razborov and Steven Rudich. Natural proofs. *Journal of Computer and System Sciences*, 55(1):24–35, 1997.

[Sch90] Robert E. Schapire. The strength of weak learnability. *Mach. Learn.*, 5:197–227, 1990.

[Sha01] Ronen Shaltiel. Towards proving strong direct product theorems. In *Proceedings 16th Annual IEEE Conference on Computational Complexity*, pages 107–117. IEEE, 2001.

[She09] Alexander A Sherstov. Separating ac0 from depth-2 majority circuits. *SIAM Journal on Computing*, 38(6):2113, 2009.

[Smo87] Roman Smolensky. Algebraic methods in the theory of lower bounds for boolean circuit complexity. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pages 77–82, 1987.

[SR94] Kai-Yeung Siu and Vwani P Roychowdhury. On optimal depth threshold circuits for multiplication and related problems. *SIAM Journal on discrete Mathematics*, 7(2):284–292, 1994.

[Val84] Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[Vio15] Emanuele Viola. The communication complexity of addition. *Combinatorica*, 35:703–747, 2015.

[VW07] Emanuele Viola and Avi Wigderson. Norms, xor lemmas, and lower bounds for gf (2) polynomials and multiparty protocols. In *Twenty-Second Annual IEEE Conference on Computational Complexity (CCC'07)*, pages 141–154. IEEE, 2007.

[Yao82] Andrew C Yao. Theory and application of trapdoor functions. In *23rd Annual Symposium on Foundations of Computer Science (SFCS 1982)*, pages 80–91. IEEE, 1982.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il