On Basing Lower-Bounds for Learning on Worst-Case Assumptions

Benny Applebaum* Boaz Barak† David Xiao‡

Abstract

We consider the question of whether $\mathbf{P} \neq \mathbf{NP}$ *implies that there exists some concept class that is efficiently representable but is still hard to learn in the PAC model of Valiant (CACM '84), where the learner is allowed to output any efficient hypothesis approximating the concept, including an "improper" hypothesis that is not itself in the concept class. We show that unless the Polynomial Hierarchy collapses, such a statement cannot be proven via a large class of reductions including Karp reductions, truth-table reductions, and a restricted form of non-adaptive Turing reductions. Also, a proof that uses a Turing reduction of constant levels of adaptivity would imply an important consequence in cryptography as it yields a transformation from any average-case hard problem in* $\mathbf{NP}$ *to a one-way function. Our results hold even in the stronger model of agnostic learning.*

These results are obtained by showing that lower bounds for improper learning are intimately related to the complexity of zero-knowledge arguments and to the existence of weak cryptographic primitives. In particular, we prove that if a language L reduces to the task of improper learning of circuits, then, depending on the type of the reduction in use, either (1) L has a statistical zero-knowledge argument system, or (2) the worst-case hardness of L implies the existence of a weak variant of one-way functions defined by Ostrovsky-Wigderson (ISTCS '93). Interestingly, we observe that the converse implication also holds. Namely, if (1) or (2) hold then the intractability of L implies that improper learning is hard.

*Department of Computer Science, Princeton University, benny.applebaum@gmail.com. Supported by NSF grant CCF-0426583.

†Department of Computer Science, Princeton University, boaz@cs.princeton.edu. Supported by NSF grants CNS-0627526 and CCF-0426582, US-Israel BSF grant 2004288 and Packard and Sloan fellowships.

‡Department of Computer Science, Princeton University, dxiao@cs.princeton.edu .

1. Introduction

Computational learning theory captures the intuitive notion of *learning from examples* in a computational framework. In particular, Valiant's PAC (Probably Approximately Correct) learning model [35] considers the following setting: a learner attempts to approximate an unknown target function $f : \{0,1\}^n \to \{0,1\}$ taken from a predefined class of functions $\mathcal{C}$ (*e.g.* the class of small DNFs). He gets access to an oracle that outputs labeled examples (x, y) where x is drawn from some unknown distribution X over the domain of f and $y = f(x)$. At the end, the learner outputs an hypothesis h which is supposed to approximate the target function f, in the sense that, say, $\Pr_X[h(x) \neq f(x)] < \varepsilon$ for some small $\varepsilon > 0$. The class $\mathcal{C}$ is efficiently *PAC-learnable* if there is a polynomial-time learner that succeeds in this task with high probability for every $f \in \mathcal{C}$ and every distribution X on the inputs (see Section 2 for a formal definition). For most applications it is not important how the output hypothesis h is represented as long as h is efficiently computable and it predicts the value of the target function f correctly on most inputs. Indeed the general definition of PAC learning allows h to be represented as an arbitrary polynomial-size circuit even if the target function is chosen from a more restricted class. A *proper* learner is a learning algorithm that only outputs hypothesis in the class $\mathcal{C}$; thus the task of general PAC learning is sometimes known as "improper" learning.

Computational learning theory has provided many strong algorithmic tools and showed that non-trivial concept classes are efficiently learnable. But despite these successes it seems that some (even simple) concept classes are hard to learn. It is considered all the more unlikely that *every* efficiently computable function can be learned efficiently. We refer to this belief as the "Learning is Hard" (LIH) assumption:

Assumption 1. ***(Learning is Hard (LIH))*** *The concept class of polynomial-size Boolean circuits cannot be learned efficiently in the PAC model.*

The LIH assumption is easily shown to be stronger

than the conjecture that $\mathbf{P} \neq \mathbf{NP}$, but it is still widely believed to be true. In fact, LIH is implied by cryptographic assumptions, namely the existence of one-way functions [13, 21, 34]. But such cryptographic assumptions seem qualitatively stronger than the assumption that $\mathbf{P} \neq \mathbf{NP}$. The main question we are concerned with in this paper is whether one can prove that if $\mathbf{P} \neq \mathbf{NP}$ then the LIH assumption is true.

Basing LIH on NP-hardness. Some previous works suggest that there is some hope to derive hardness of learning from $\mathbf{P} \neq \mathbf{NP}$. *Proper learning* (where the hypothesis h has to be in the class $\mathcal{C}$) is known to be **NP**-hard in general [33].[1] Such hardness results hold for several concept classes and for other variants and extensions of the PAC learning model (cf. [33, 7, 20, 4, 3, 11, 10, 19, 18]). One may hope to use these results as a starting point for proving **NP**-hardness in the general (*i.e.* improper) setting. Indeed, although some of the aforementioned lower bounds seem useless for this purpose (as they apply to concept classes which are known to be improperly learnable), others might still be relevant in our context. In particular, [3] show that it is **NP**-hard to learn the intersection of two halfspaces even in a semi-proper setting where the learner is allowed to use an intersection of any (constant) number of halfspaces. Similarly, [18] show that learning parity in the agnostic model, where the data is noisy, is **NP**-hard even if the learner is allowed to use a low degree polynomial. These concept classes are not known to be efficiently learnable. Also, both works rely on highly non-trivial PCP machinery. This may give some hope that similar techniques will eventually prove that (improper) learning is **NP**-hard.

1.1. Our Results

As indicated above, it is not known whether $\mathbf{NP} \neq \mathbf{P}$ implies the LIH assumption. We show that a wide range of known techniques are unlikely to prove this statement.[2] Specifically, our main result shows that if learning circuits is proved to be **NP**-hard via a large family of reductions (including Karp reductions, truth-table reductions, or Turing reductions of bounded adaptivity) then, depending on the type of the reduction, either the Polynomial-Hierarchy (**PH**) collapses or any average-case hard problem in **NP** can be converted into a one-way function (in terms of [22], Pessiland collapses to Minicrypt). The first consequence is considered to be implausible, while the latter would be a major breakthrough in cryptography.

These results are obtained by showing that lower bounds for improper learning are intimately related to the complexity of zero-knowledge and to the existence of weak cryptographic primitives. In particular, we prove that if deciding a language L reduces to the task of learning circuits, then, depending on the type of the reduction in use, either (1) L has a statistical zero-knowledge argument system, or (2) the worst-case hardness of L implies the existence of auxiliary-input one-way functions [32], which are a weak variant of one-way functions. This holds even in the stronger model of agnostic learning. While the aforementioned implications are too weak to be useful for cryptographic applications, we can still show that when L is **NP**-complete they lead to unlikely consequences such as the collapse of **PH** or Pessiland=Minicrypt. This is proved by relying on the works of [8, 12, 32, 2].

Interestingly, we observe that the converse implication is also true. Namely, if (1) or (2) hold then the intractability of L implies that improper learning is hard in a relatively strong sense (*i.e.* even when the examples are drawn from the uniform distribution and the learner is allowed to query the target function on any given point). This is proved by a simple combination of [13, 21, 32]. Overall, we get some form of "necessary and sufficient" condition for proving LIH via reductions. We use it to conclude that proving a weak version of LIH (via standard techniques) is not easier than proving a strong version of LIH.

Constructing one-way functions from LIH. A different approach would be to show that LIH is *sufficient* for cryptography. That is, the LIH assumption is equivalent to the assumption that one-way functions exist. From a first look, the probabilistic aspect of the PAC model may give some hope that a hard to learn problem can be used for cryptographic applications. The works of [23, 6] show that this is indeed the case when an *average-case* version of the PAC model is considered. But there appear to be significant obstacles to extending this result to the case of standard PAC learning. In particular, LIH only guarantees that every learner fails to learn *some* function family over *some* distribution ensemble – a single hard-to-learn function and distribution might not exist. A more serious problem is that because the PAC model ignores the complexity of the target distribution, LIH might hold only

[1] A different restriction on the power of the learner is studied in the Statistical Query model [25]. In this model the learner has a limited access to the examples, and hardness of learning can be proven unconditionally without relying on computational assumptions [5].

[2] Clearly we cannot hope to unconditionally rule out the implication "$\mathbf{P} \neq \mathbf{NP} \Rightarrow$ LIH", as it trivially holds under the assumption that one-way functions exist.

with respect to distributions which are not efficiently samplable; such distributions seem useless for cryptography. More concretely, we observe that LIH can be based on the non-triviality of zero-knowledge proofs (*i.e.* $\mathbf{ZK} \not\subseteq \mathbf{BPP}$), and consequently, on the worst-case hardness of QuadraticResidue, GraphIsomorphism and DiscreteLog [17, 15, 14]. Hence, proving that LIH suffices for the existence of one-way functions, would show that one-way functions can be based on $\mathbf{ZK} \not\subseteq \mathbf{BPP}$. Again, such a result would have a major impact on cryptography.

Related work. As mentioned above, several works gave **NP**-hardness results for *proper* and *semi-proper* learning, but known hardness results for general (improper) learning are only based on cryptographic assumptions. In particular, Pitt and Warmuth [34] observed that LIH is implied by one-way functions by combining [13, 21], while hardness of learning specific concepts under specific cryptographic assumptions was shown in several other works including [27, 28].

The question of whether worst-case assumptions such as $\mathbf{P} \neq \mathbf{NP}$ are sufficient for learning lower-bounds was first raised by Akavia, Goldwasser, Malkin and Wan (personal communication, Winter 2006), who showed results of similar flavor to this work— namely that under widely believed complexity assumptions, certain types of black-box reductions will not be able to show statements of the form "$\mathbf{P} \neq \mathbf{NP}$ implies hardness of learning". However, the notion of hardness of learning they studied was either hardness of *average-case* learning (that as mentioned above is known to imply the existence of one-way functions [23, 6]), or hardness of worst-case learning of *specific concept classes* (in particular not including those concept classes that are known to be **NP**-hard to learn *properly*). In contrast, the LIH Assumption we study talks about worst-case learning of any efficiently representable concept class.

1.2. Proving LIH via Reductions

We proceed with a detailed account of our main result. We consider the existence of reductions that prove that $\mathbf{NP} \neq \mathbf{P}$ implies LIH by solving an **NP**-hard language L such as SAT using the power of a PAC learning algorithm for the concept class of Boolean circuits. More formally, a *circuit learner* takes as an input an accuracy parameter ε, and oracle access to a joint distribution (X, Y) where X is the target distribution over $\{0,1\}^n$ and $Y = f(X)$. The learner outputs an hypothesis h, represented by a circuit, which ε-approximates f with respect to X (*i.e.* $\Pr[h(X) \neq Y] \leq \varepsilon(n)$). The complexity of the learner is polynomial in $1/\varepsilon$ and in the circuit size of f.[3] We consider several possible ways (reductions) in which one can use a circuit learner to decide a language L.

Karp reductions. Perhaps the most natural way to reduce a language L to a circuit learner is to give a Karp reduction mapping an instance z of L into a circuit sampling a distribution (X, Y) over $\{0,1\}^n \times \{0,1\}$ such that Y is equal to $f(X)$ for some f computable by a polynomial-sized Boolean circuit if and only if $x \in L$ (see Section 3 for a formal definition). In fact, to the best of our knowledge, all the previous **NP**-hardness results for learning (i.e., in the proper and semi-proper cases) were proved via such reductions. Our first result rules out such a reduction:

Theorem 1. *For every language L, if L reduces to circuit learning via a Karp reduction then L has a statistical zero knowledge argument system. Moreover, if L is* **NP***-complete and such a reduction exists then the polynomial hierarchy collapses to the second level.*[4]

The second part of the theorem generalizes to the case of (randomized) truth-table reductions [29] (which are equivalent to non-adaptive Turing reductions).

Turing reductions. Karp reductions use the learning algorithm in a very limited way, namely, as a distinguisher between learnable instances and non-learnable instances. This motivates the study of reductions that exploit the hypothesis generated by the learner in a stronger way. Formally, we can think of a circuit learner as an algorithm which solves the *Circuit Learning search problem*, and consider Turing reductions from L to this problem. Such reductions interact with the learner by supplying it with distributions of labeled examples, and obtaining from the learner hypotheses predicting the labels under the distributions (if such predictors exist). We allow the reduction to use the hypotheses returned by the learner arbitrarily. That is, the reduction can apply any (efficient) computation to the circuits that describe the hypotheses

[3] The real definition of PAC learning allows the learner to err with some probability according to a given confidence parameter. For simplicity we do not allow such an error, which only makes our results stronger. Also, by a simple padding argument we may assume in our context without loss of generality that f has a circuit of size, say, n^2.

[4] The "moreover" part does not follow immediately since the existence of a statistical zero-knowledge argument system for an **NP**-complete problem does not collapse the Polynomial Hierarchy by itself— in fact, assuming that OWFs exists, SAT has such a zero-knowledge protocol [30]. We collapse **PH** by relying on the special structure of reductions to circuit learning.

in order to solve the underlying language L. Unfortunately, we are not able to rule out fully adaptive Turing reductions, and so will only consider reductions of *bounded adaptivity* where the interaction between the reduction and the learner proceeds in a constant number of adaptive rounds. (See Section 4 for the formal definition.) Our main result for such reductions is the following:

Theorem 2. *If L reduces to circuit learning via a Turing reduction of bounded adaptivity, then there is an* auxiliary input one-way function *[32] based on the hardness of L. Moreover, if such a reduction exists and L is hard on the average then there exists a (standard) one-way function.*

Note that Theorem 2 means that a reduction of an **NP**-complete problem to circuit learning shows that if there is an hard-on-average problem in **NP** then one-way functions exist. In Impagliazzo's terms [22] this means that such a reduction would collapse the worlds "Pessiland" and "Minicrypt". We also show that if L reduces to Circuit-Learning via a special family of reductions (*i.e.* Turing reductions in which the queries are generated non-adaptively and the hypotheses are used in a non-adaptive and black-box way) then $L \in$ **CoAM**. When L is **NP**-complete this collapses the Polynomial-Hierarchy.

Extension to the Agnostic Setting. In the *agnostic* learning model of [26] the learner still gets a joint distribution (X, Y) but the distribution of the labels Y is arbitrary and does not necessarily fit to any target function f in the concept class. The learner is guaranteed to output an hypothesis h whose error with respect to (X, Y) (*i.e.* $\Pr[h(X) \neq Y]$) is at most ε larger than the error of the best function f in the concept class $\mathcal{C}$. Learning in the agnostic model seems much harder, and there are examples of concept classes (*e.g.* parity) that are PAC learnable but not known to be learnable in the agnostic model. Thus one may hope that it would be easier to prove **NP**-hardness results in this model. (Indeed as mentioned above [18] prove a semi-proper **NP**-hardness result for agnostic learning of parity.) Alas, all our results extend (with some work) to the agnostic model as well, thus ruling out proving such hardness results via a large class of reductions.

1.3. Our Techniques

To illustrate some of our techniques we consider the simple case of a deterministic Turing reduction that decides SAT by making a single query to the Circuit-Learner. Such a reduction is described by a pair of probabilistic polynomial-time algorithms (T, M) and an accuracy parameter ε. On input a 3CNF formula represented by a string z, the algorithm $T(z)$ outputs a query to the learner (X_z, Y_z), while the algorithm M uses z and (the code of) an hypothesis h, returned by the learner, to decide whether $z \in$ SAT. We say the query (X_z, Y_z) is "honest" if there exists some f_z in the concept class of polynomial-sized Boolean circuits such that $\Pr[Y_z = f(X_z)] = 1$. If the query is honest then the reduction expects the hypothesis h to be ε-good (*i.e.* $\Pr[Y_z \neq h(X_z)] < \varepsilon$).[5]

We would like to show that this reduction leads to some implausible consequence such as collapsing the polynomial hierarchy or constructing a one-way function based on the hardness of SAT. Let's focus on the latter case. We assume that such a reduction exists but one-way functions do not exist, and we'll use that to show a polynomial-time algorithm for SAT. (Thus if such a reduction exists then $\mathbf{P} \neq \mathbf{NP} \Rightarrow \exists$ OWF.)

A problematic approach. To use the reduction to solve SAT, it suffices to show that given an honest query (X_z, Y_z) we can find an ε-good hypothesis h for (X_z, Y_z). At first glance this seems easy under our assumption that one-way functions do not exist. After all, if the query is honest then there exists some efficiently computable function f_z such that $Y_z = f_z(X_z)$, and f_z has no "cryptographic strength". In particular this means that the collection of functions $\{f_z\}$ is not pseudorandom and can be predicted by a polynomial-time adversary. Indeed, this approach was used in [6] where an average-case version of LIH is considered. However, in our setting this solution suffers from two problems. First, to obtain a worst-case algorithm for SAT we should be able to predict f_z for *every* string z and not just a random z. The second and main problem is that the mapping $(x, z) \mapsto f_z(x)$ might not be efficiently computable, and therefore the collection $\{f_z\}$ can be pseudorandom without contradicting our assumption. Indeed, the only efficiency guarantee we have is that f_z has a small circuit for every *fixed* z. One may try to argue that the query generator T can be used to compute this mapping but T only computes the mapping $z \to (X_z, Y_z)$ and does not provide a circuit for f_z or even a guarantee that such a small circuit exists. In fact, if T could compute f_z, it could feed it directly to M and solve SAT without using the learner.[6]

[5] We do not assume in our proofs that the reduction queries are always honest. However, the learner is not required to return any meaningful answer for non-honest queries.

[6] Both problems were bypassed in [6] by assuming an average-case version of LIH. This assumption guarantees the existence of efficiently samplable distributions, F over target functions and X over examples, which are universally hard for all learners.

Solving the main problem. Assume for now that the query is honest (i.e., $Y_z = f_z(X_z)$). Instead of learning the target function f_z we will "learn" the *distribution* (X_z, Y_z). That is, given x and z we will try to find an element y in the support of the marginal distribution $Y_z|X_z = x$. Note that if the query is honest then indeed $y = f_z(x)$. To find y we will use our ability to invert the circuit C_z that samples the joint distribution (X_z, Y_z). Specifically, we "break" the circuit C_z into two parts: $C_z^{(1)}, C_z^{(2)}$ such that $(C_z^{(1)}(r), C_z^{(2)}(r)) \equiv (X_z, Y_z)$ for a randomly chosen r. In order to classify an example x, our hypothesis h uses an inverter for $C_z^{(1)}$ to find an r such that $C_z^{(1)}(r) = x$ and then computes the value of $y = C_z^{(2)}(r)$. Clearly, whenever the inversion succeeds the hypothesis h classifies x correctly. Assuming that $C_z^{(1)}$ is not even a weak one-way function (as otherwise one-way functions exist [36]) we can invert $C^{(1)}(r)$ with probability $1 - \varepsilon$ (for arbitrary polynomially small $\varepsilon > 0$) when r is chosen randomly. Since our hypothesis is tested exactly over this distribution (*i.e.* , over $X_z \equiv C_z^{(1)}(r)$), by using an ε-inverter we can get an ε-good hypothesis h. To deal with the case of dishonest query we estimate the accuracy of our hypothesis (by testing it on (X_z, Y_z)) and output $\perp$ if it is not ε-good.

It is important to note that although this approach leads to an ε-good hypothesis, it does not mean that we PAC-learned f_z. Indeed, the complexity of our hypothesis depends on the complexity of the *target distribution* (as well as on the complexity of the inverter), while a true PAC-learner outputs an hypothesis whose complexity is *independent* of the target distribution. However, since the learner is assumed to be improper (*i.e.* it can output an hypothesis whose complexity is polynomially-related to the complexity of f_z), the reduction will "miss" this difference and will act properly as long as the hypothesis h_z is ε-close to f_z.

Obtaining strong consequences. Our approach still suffers from the first problem— the non-existence of one-way functions will only imply that we can invert $C_z^{(1)}$ on a random z rather than for *every* z. But in particular this implies that if one-way functions do not exist then SAT can be solved on the average! This collapses the worlds "Minicrypt" and "Pessiland" of Impagliazzo [22] and would be a major breakthrough in complexity. Moreover in some special (yet interesting cases) we can use properties of the reduction and apply the results of [2] to this setting and show that $\mathsf{SAT} \in \mathbf{CoAM}$, which collapses the Polynomial Hierarchy to the second level.

Additional tools. The extension to the agnostic-case and to the case of Turing reductions of bounded adaptivity requires additional tools and ideas. One important ingredient is the notion of distributionally one-way functions [24] and its equivalence to standard one-way functions. We also employ cryptographic reductions from [36, 13, 21]. Our use of "prediction via inversion" is similar to the notion of Universal Extrapolation of [23]. As we already mentioned we also use results from [32] and [2]. For our results on *Karp reductions* (which were not discussed on this subsection) we crucially rely on characterization theorems for statistical zero-knowledge arguments [31] and statistical zero-knowledge proofs [16].

1.4. Organization

Some preliminary definitions are in Section 2. Our results for Karp reductions are in Section 3. The results for Turing reductions are in Section 4. Due to space considerations, many of the proofs are only sketched. The full proofs, as well as the observation that auxiliary input one-way functions imply the LIH assumption, are deferred to the full version of this paper.

2. Preliminaries

Notation. We use U_n to denote a random variable uniformly distributed over $\{0,1\}^n$. If X is a probability distribution, or a random variable, we write $x \stackrel{R}{\leftarrow} X$ to indicate that x is a sample taken from X. The *statistical distance* between discrete probability distributions X and Y, denoted $\Delta(X,Y)$, is defined as the maximum, over all functions A, of the *distinguishing advantage* $|\Pr[A(X)=1] - \Pr[A(Y)=1]|$. Equivalently, the statistical distance between X and Y may be defined as $\frac{1}{2}\sum_z |\Pr[X=z] - \Pr[Y=z]|$.

Learning models. A *learning algorithm* gets access to an *example oracle* which samples from a distribution (X,Y) over $\{0,1\}^n \times \{0,1\}$ and tries to output an *ε-good hypothesis*, which is a function h such that $\Pr[h(X) \neq Y] < \varepsilon$. In *PAC learning* [35] the example oracle is guaranteed to satisfy $Y = f(X)$ where f is a function from some *concept class* $\mathcal{C}$. Throughout this paper we fix $\mathcal{C}$ to be the concept class of Boolean circuits of size n^c for some absolute constant c (e.g. $c = 2$). In our context this is the most general choice and the one that makes our results the strongest. The PAC learner is efficient if it runs in time $\mathrm{poly}(n, 1/\varepsilon, 1/\delta)$ where δ upper bounds the probability that the learner fails to output an ε-good hypothesis. In *agnostic learning* [26] there is no guarantee on the example oracle,

and the learner simply needs to output an hypothesis h such that $\Pr[h(X) \neq Y] < \min_{f \in \mathcal{C}}[f(X) = Y] + \varepsilon$. Since it's harder to learn in the agnostic model, allowing this model makes our results stronger.

Auxiliary-input primitives. Ostrovsky and Wigderson [32] defined the notion of *auxiliary input cryptographic primitives* which is a significantly weakened variant of standard cryptographic primitives. An auxiliary input (AI) function is an efficiently computable function $f(\cdot)$ that in addition to its input $x \in \{0,1\}^*$ gets an additional input $z \in \{0,1\}^{|x|}$ (we typically denote the function $x, z \mapsto f(x,z)$ by $f_z(\cdot)$). The security condition of an AI primitive is relaxed to requiring that for every potential efficient adversary A, there exists an infinite set $Z_A \subseteq \{0,1\}^*$ (depending on A) such that A fails to "break" the function whenever the auxiliary input comes from Z_A. (The fact that Z depends on A is the reason why auxiliary input primitives are generally not sufficient for most cryptographic applications.) In particular, f is auxiliary input one-way function (AIOWF) if for every $z \in Z_A$ the success probability $\Pr_x[A(z, f_z(x)) \in f_z^{-1}(f_z(x))]$ is negligible in $|z|$. The function is AI weak-OWF if there exists a polynomial $p(\cdot)$ such that for every inverter A and all $z \in Z_A$ we have $\Pr_x[A(z, f_z(x)) \notin f_z^{-1}(f_z(x))] > 1/p(|z|)$. We say that f is AI distributional-OWF if there exists a polynomial $p(\cdot)$ such that for every distributional inverter A, and all $z \in Z_A$ the random variables $(x, f_z(x))$ and $(A(z, f_z(x)), f_z(x))$ where $x \stackrel{R}{\leftarrow} U_n$, are at least $1/p(n)$ far in statistical distance. [32] showed that if **ZK** $\neq$ **BPP** then there exist AIOWF. Most cryptographic reductions carries to the auxiliary input setting. In particular, one can transform AI distributional-OWF to AI weak-OWF, and AI weak-OWF to AIOWF[36, 24].

3. Karp Reductions to Learning

Perhaps the most natural route to prove that **NP** $\neq$ **P** implies LIH is via *Karp* reductions. Indeed, all the **NP**-hardness results for learning we are aware of (including the new PCP-based results) are of this type. In this section we define formally Karp reductions and show that such reductions cannot show **NP**-hardness of learning unless the polynomial hierarchy collapses. Moreover, we show that if a language L (which is not necessarily **NP**-hard) Karp reduces to the task of improper learning circuits, then, L has a statistical zero-knowledge argument system. Hence, the intractability of **SZKA** is a necessary condition for proving LIH via Karp reductions. We start in Section 3.1 by considering a simplified notion of Karp reduction in which the NO case is mapped to a distribution that cannot be predicted in an information theoretic sense (i.e., even using a computationally unbounded hypothesis). Then, in Section 3.2 we extend our results to the case in which the NO condition only holds for computationally bounded hypothesis. Our main approach is to relate the existence of such reductions to the existence of zero knowledge proofs or arguments for a certain promise problems that we call SGL (for statistical gap-learning) and CGL (for computational gap-learning).

In this section (as is throughout the paper) we also consider the case of *agnostic* learning. In agnostic learning the learner must work even given examples that are not perfectly predictable using a concept. This makes the task of reductions to the learner easier, and hence makes our results (that rule out such reductions) stronger.

3.1. Information-Theoretic Setting

We define a decision version of the problem of agnostic PAC learning circuits.

Definition 3.1. (Gap-Learning Problem – the information theoretic version) *Let α, β be some functions mapping $\mathbb{N}$ to $[0,1]$ such that $1/2 \leq \beta(n) < \alpha(n) \leq 1$ for every $n \in \mathbb{N}$ and $p(\cdot)$ be some polynomial. The input to the promise problem $\mathsf{SGL}_{\alpha,\beta}$ is a circuit C of $\mathrm{poly}(n)$ size[7] which samples a joint distribution (X,Y) over $\{0,1\}^n \times \{0,1\}$:*

- *YES instance: there exists a function $f \in \mathcal{C}$ such that $\Pr[f(X) = Y] \geq \alpha(n)$.*
- *NO instance: for every (even inefficient) function f, $\Pr[f(X) = Y] \leq \beta(n)$.*

The parameters. The special case of $\alpha = 1$ corresponds to the PAC-learning setting, while smaller α corresponds to agnostic learning. In order to be useful in our context, $\alpha(n) - \beta(n)$ must be noticeable (i.e., larger than $1/p(n)$ for some polynomial $p(\cdot)$). In this setting of parameters, an agnostic learner can be used to decide $\mathsf{SGL}_{\alpha,\beta}$, while a PAC learner can be used to decide $\mathsf{SGL}_{1,\beta}$. Our results hold for any choice of parameters that satisfy these conditions.

Our main result in this section shows that this problem is in **SZKP** (also known as **SZK**)— the class of languages that has a zero knowledge proof system where both soundness and zero knowledge hold in a

[7] We can think of $|C| = n^2$ without loss of generality.

statistical sense (i.e., with respect to computationally unbounded parties).

Theorem 3.2. *For every α and β which satisfy $1/2 \leq \beta(n) < \alpha(n) \leq 1$ and $\alpha(n) - \beta(n) > 1/p(n)$ for some polynomial $p(\cdot)$, the problem $\mathsf{SGL}_{\alpha,\beta}$ is in* **SZKP**.

Proof sketch. We show this by reducing $\mathsf{SGL}_{\alpha,\beta}$ to the *Entropy Difference* problem which is **SZKP**-complete [16]. An instance of the entropy difference problem are a pair of circuits W, Z on n bit inputs, which we identify with their output distributions on random input. The YES instances satisfy $H(W) \geq H(Z) + 1/100$, where H is the Shannon entropy, and the NO instances satisfy $H(Z) \geq H(W) + 1/100$. Now set Z to be a circuit outputting the distribution (X, Y), and set W to be a circuit outputting the distribution (X, Y') where Y' is an independent random variable that is equal to 1 with probability $(\alpha + \beta)/2$, and to 0 otherwise. It's not hard to see that if $\beta \leq 0.6$ and $\alpha \geq 0.7$ then this reduction maps YES instances (resp. NO instances) of $\mathsf{SGL}_{\alpha,\beta}$ to YES instances (resp. NO instances) of Entropy Difference. The proof for general α, β is obtained by some simple manipulations. We omit the details. □

Theorem 3.2 yields the following corollaries:

Corollary 3.3. *(1) If a promise problem Π reduces to SGL via a Karp reduction, then Π has a statistical zero-knowledge proof. More generally, if $\mathsf{SGL} \notin$* **BPP** *then* **SZKP** $\nsubseteq$ **BPP**. *(2) There is no Karp reduction (or even non-adaptive Turing reduction) from SAT to SGL unless the Polynomial Hierarchy (***PH***) collapses.*

Proof. The first item follows from Theorem 3.2, and the fact that **SZKP** is closed under Karp-reductions. To prove the second item note that: (1) **SZKP** $\subseteq$ **CoAM** [1]; (2) If **NP** $\subseteq$ **CoAM**, then the Polynomial Hierarchy collapses [8]; and (3) **CoAM** is closed under non-adaptive Turing reductions [9]. □

3.2. Computational Setting

We now consider a computational version of SGL denoted as CGL in which the non-learnability in the NO case holds with respect to *efficient* hypotheses. This generalizes the information theoretic version as any YES (resp. NO) instance of CGL is also a YES (resp. NO) instance of CGL.

Definition 3.4. (Gap-Learning Problem – the computational version) *The promise problem $\mathsf{CGL}_{\alpha,\beta}$ is defined similarly to $\mathsf{SGL}_{\alpha,\beta}$, except that NO instances satisfy the following: for every function f computable by a circuit of size at most $n^{\log n}$ we have $\Pr[f(X) = Y] \leq \beta(n)$.*

The choices $n^{\log n}$ is arbitrary and any function $s(n) = n^{\omega(1)}$ will do. Again, we assume that the parameters α, β satisfy $1/2 \leq \beta(n) < \alpha(n) \leq 1$ and $\alpha(n) - \beta(n)$ is noticeable. Our main theorem on this problem is the following:

Theorem 3.5. *For every α and β which satisfy $1/2 \leq \beta(n) < \alpha(n) \leq 1$ and $\alpha(n) - \beta(n) > 1/p(n)$ for some polynomial $p(\cdot)$, the problem $\mathsf{CGL}_{\alpha,\beta}$ is in* **SZKA** *(the class of languages with statistically hiding but computationally sound zero knowledge proofs).*

The proof relies on the "SZK/OWF" characterization of **SZKA** recently shown by Ong and Vadhan [31], which says that a promise problem $\Pi = (\Pi_{\text{Yes}}, \Pi_{\text{No}}) \in$ **MA** is in **SZKA** iff there exists a set $I \subseteq \Pi_{\text{No}}$ such that $(\Pi_{\text{Yes}}, \Pi_{\text{No}} \setminus I) \in$ **SZKP** and one can construct an auxiliary-input function g_z which is one-way for instances $z \in I$.

We will also need the following lemma. The proof uses the ideas described in Section 4.2, and is omitted from this version.

Lemma 3.6. *Let $X_z : \{0,1\}^{m(|z|)} \to \{0,1\}^{n(|z|)}$ and $Y_z : \{0,1\}^{m(|z|)} \to \{0,1\}$ be auxiliary-input functions which are polynomial-time computable. Let $I \subset \{0,1\}^*$ be a set, A be an efficient inverting algorithm, and $\delta, \varepsilon : \mathbb{N} \to [0,1]$ be functions. Suppose that for each $z \in I$:*

1. *There exists (possibly inefficient) function f such that $\Pr[f(X_z(U_{m(|z|)})) \neq Y(U_{m(|z|)})] \leq \delta(|z|)$.*
2. *A distributionally inverts $X_z(U_m)$ with statistical-distance $\varepsilon(|z|)^3/(2|z|)$.*

Then, there exists an efficiently-computable hypothesis h for which $\Pr[h(X_z(U_{m(|z|)})) \neq Y((U_{m(|z|)}))] \leq \delta(|z|) + \varepsilon(|z|)$, for all $z \in I$.

Proof sketch of Thm 3.5. We show that $\mathsf{CGL}_{\alpha,\beta}$ satisfies the "SZK/OWF" characterization. It's not hard to show that $\mathsf{CGL}_{\alpha,\beta}$ is in **MA**. We define I to be the set of NO-instances (X, Y) for which there exists some function f such that $\Pr[f(X) = Y] \geq (\alpha + \beta)/2$. (Of course since (X, Y) is a NO instance, f is not efficiently computable.) Theorem 3.2 implies that $\mathsf{CGL}_{\alpha,\beta} \setminus I$ is in **SZKP**. We argue that if (X, Y) is a NO instance in I, then the circuit g_X sampling X is a distributional one-way function. Indeed, by Lemma 3.6, a nonuniform probabilistic algorithm A that distributionally inverts g_X with sufficiently (polynomially) small statistical-distance yields a polynomial-size hypothesis h for which $\Pr[h(X) = Y] > \beta$, in contradiction to (X, Y) being a NO-instances. □

Theorem 3.5 together with the fact that **SZKA** is closed under Karp-reductions implies the the following

corollary (which is a restatement of the first part of Theorem 1):

Corollary 3.7. *If a promise problem* Π *reduces to* CGL *via a Karp reduction, then* Π *has a statistical zero-knowledge argument. More generally, if* $\mathsf{CGL} \notin \mathbf{BPP}$ *then* $\mathbf{SZKA} \not\subseteq \mathbf{BPP}$.

As mentioned earlier, we can also show that there is no Karp reduction (or even non-adaptive Turing reduction) from SAT to CGL unless the Polynomial Hierarchy collapses. This will be proved in the next section as a special case of Corollary 4.3.

4. Turing Reductions to Learning

We now consider a much more general class of reductions than Karp reductions, namely Turing reductions with bounded adaptivity. Such reductions use the learner not just as a distinguisher between learnable and non-learnable sets of examples, but may also use the actual hypotheses supplied by the learner. We show that if L has a bounded-adaptivity Turing reduction to circuit learning then the worst-case hardness of L can be used to construct AI one-way function. Furthermore, if L is hard on the average, we get a (standard) one-way function. These results hold even if the learner is guaranteed to learn in the agnostic setting.

We start by formally defining non-adaptive and bounded-adaptive Turing reductions to circuit learning. In the following we let $t \in \mathbb{N}$ be a constant and ε be a noticeable function (*i.e.* bounded by some inverse polynomial).

Definition 4.1. (Turing reductions to learning of bounded-adaptivity) *A query*[8] (X, Y) *to the learner is a joint distribution over* $\{0,1\}^n \times \{0,1\}$ *which is represented, as a circuit* C *which takes* m *random bits and samples* (X, Y), i.e. $C(U_m) \equiv (X, Y)$. *A* t*-adaptive Turing reduction from deciding* L *to* ε*-PAC-learning* $\mathcal{C}$ *(resp.* ε*-agnostically learning* $\mathcal{C}$*) is a tuple of probabilistic polynomial-time algorithms* $(T_1, \ldots, T_t, M)$. *Let* $q(\cdot)$ *denote the query-complexity of each round of the reduction. The reduction attempts to decide whether an input* z *is in* L *in the following way:*

- T_1 *takes input* $z \in \{0,1\}^n$ *and fresh random bits* ω *and outputs* $q(n)$ *queries for the learner,* i.e. *distributions* $(X_1, Y_1), \ldots, (X_{q(n)}, Y_{q(n)})$ *where each joint distribution* (X_i, Y_i) *is sampled a circuit* C_i.
- *For each* $j \geq 2$, *the machine* T_j *takes input* z, ω *and additionally gets all* $(j-1)q(n)$ *hypotheses (represented as circuits) answering queries from previous rounds, and outputs* $q(n)$ *new queries for the learner.*
- M *takes as input* z, ω, *as well as the* $t \cdot q(n)$ *hypotheses (represented as circuits) answering all previous queries as additional input, and outputs a decision bit* b.

Guarantee: *The reduction guarantees that if all hypotheses returned by the learner are* ε*-good in the PAC model (resp. in the agnostic model) with respect to the corresponding queries of* $T_1, \ldots, T_t$, *then* M *decides* z *correctly with probability* $2/3$ *over the choice of* ω. *The reduction is called* non-adaptive *if* $t = 1$, *and* fully non-adaptive *if in addition,* M *uses the hypotheses as black-boxes and in a non-adaptive way.*

Our main result of this section is the following:

Theorem 4.2. *Suppose that the language* L *reduces to Circuit-Learning in the agnostic model via a Turing reduction of bounded adaptivity. Then,* $L \notin \mathbf{BPP}$ *implies the existence of AI-one-way functions.*

We first prove the theorem in the simpler case of the PAC model (Section 4.1), and then sketch the proof for the agnostic case (Section 4.2). Some of the details are omitted and will be given at the full version.

4.1. Proof of Thm 4.2 in the PAC model

Consider the case that the reduction is non-adaptive. Hence the first stage of the reduction takes any instance z of the language L, randomness ω, and computes from these polynomially many circuits $C_{z,\omega,1}, \ldots, C_{z,\omega,k}$ where $C_{z,\omega,i}$ samples a distribution (X_i, Y_i) on inputs and labels that is given to the learner. (For simplicity assume that all queries are honest, and hence $Y_i = f(X_i)$ for some $f = f_{z,\omega,i}$ in the concept class $\mathcal{C}$.) The learner responds with a sequence of hypothesis $h_1, \ldots, h_k$, which are given as input to the second stage of the reduction. We have the guarantee that if these hypothesis are ε-good then the second stage will correctly decide if $z \in L$.

We will prove the contrapositive: if there do not exist AIOWF then $L \in \mathbf{BPP}$. In the case that the reduction has only one query and is deterministic (i.e., $k = 1$ and ω is empty) the proof was outlined in Section 1.3. The general case is similar: define the (efficiently computable) function g_z which maps the randomness ω,

[8] Other more general notions of queries are also possible, for example the reduction could output a set of labelled examples that are not generated as independent samples from a sampling circuit. However we believe that our definition is the most natural and useful notion, as the definition of learning assumes that the examples seen are generated by independent identical samples from a target distribution and labelling.

an index $i \in [k]$ and r, to the tuple ω, i and $C_{z,\omega,i}(r)$ (which is the circuit that samples X_i). By [36], the non-existence of auxilliary-input OWF means that g_z cannot be even an auxilliary-input *weak* OWF, and hence we have a polynomial-time inverter algorithm for g that will succeed in inverting $C_{z,\omega,i}$ for all i's and most ω's. By the reasoning outlined in Section 1.3, this inverter supplies a sequence of good hypotheses, which we use to build a decision procedure for the language.

Bounded adaptivity. In the case that the reduction has $t > 1$ rounds of adaptivity $T_1, \ldots, T_t$ we use induction, reducing the number of rounds by one by "assimilating" the hypothesis into the reduction. Specifically, as we showed in the non-adaptive case, assuming that AIOWFs do not exist, there exists a probabilistic polynomial time procedure B which outputs a sequence of hypotheses that answer the queries of T_1. Hence, we can reduce one round by replacing the first two stages T_1, T_2 of the reduction with a single step in which we invoke B to generate hypotheses for queries asked by T_1 and then hand them to T_2. (Each inductive step causes a polynomial blow up in the complexity and hence we cannot continue this procedure for more than a constant number of times.)

4.2. Proof of Thm 4.2 in the Agnostic model

Generalizing to the agnostic setting introduces some more technical difficulties. We do not know how well functions in $\mathcal{C}$ classify a given query (X, Y). So, instead of competing with these functions, we will try to compete with the best (information-theoretic) classifier f. Given an example x, the optimal classifier outputs the "majority label" b which maximizes $\Pr[Y = b|X = x]$. Our hypothesis will try to estimate this majority bit by sampling many random elements from the marginal $[Y|X = x]$ and taking the majority. Although this hypothesis might not always approximate f well (*e.g.* when the majority label has probability slightly larger than 1/2), we show that its error is not much larger than the error of f. To implement this approach, we rely on the ability to invert the sampling circuit even in a *distributional* sense.

Formally, consider the case of a deterministic reduction which makes a single query $C_z : \{0,1\}^{m(|z|)} \to \{0,1\}^{n(|z|)} \times \{0,1\}$ to the learner. Let $C_z = (X_z, Y_z)$ and let f_z be the (possibly non-efficient) optimal classifier which labels x by $\arg\max_{b\in\{0,1\}} \Pr[Y_z = b|X_z = x]$. By [24], the non-existence of auxiliary-input OWFs implies that X_z cannot be even auxiliary-input distributional OWF. Hence, there exists an efficient probabilistic distributional inverter A such that for all $z \in \{0,1\}^*$ the distributions $(z, r, g_z(r))$ and $(z, A(z, g_z(r)), g_z(r))$, where r is randomly chosen, are $\delta = \varepsilon(|z|)^3/(10|z|)$ close. We define the (randomized) hypothesis h_z as follows: Given x invoke the algorithm A on (z, x) for $q = q(|z|) = |z|/\varepsilon(|z|)^2$ times (each time with independent randomness) and let r_i denote the i-th output of A. Output the majority of $Y_z(r_i)$. We prove that for every fixed z the error of h_z (with respect to (X_z, Y_z)) is at most ε larger than the error of f_z.

Let $\hat{A}$ be an "ideal" inverter which distributionally inverts X_z with no deviation error, and let $\hat{h}_z$ be the hypothesis which results from h_z when A is replaced by $\hat{A}$. It is not hard to show that the performance of the real hypothesis h_z is $q \cdot \delta = \varepsilon(|z|)/10$ close to the performance of the ideal hypothesis $\hat{h}_z$. Hence, it suffices to prove that the ideal hypothesis $\hat{h}_z$ performs "almost" like the optimal classifier f_z. Fix z and let $\alpha(x)$ denote $\max_{b\in\{0,1\}} \Pr_r[Y_z(r) = b|X_z(r) = x]$. We distinguish between two cases: if the marginal $[Y_z|X_z = x]$ is biased (say, $\alpha(x) \geq 1/2 + \varepsilon(|z|)/10$) then, we can use Chernoff bound to argue that $\hat{h}_z(x)$ almost always agrees with $f_z(x)$. In the other case, when $[Y_z|X_z = x]$ is balanced (*i.e.* $1/2 \leq \alpha(x) \leq 1/2 + \varepsilon(|z|)/10$), the error probabilities of both, $\hat{h}$ and f, are close to 1/2, and therefore $\hat{h}$ performs almost as well as f. The proof extends to the case of randomized reductions with bounded adaptivity by using the same ideas described in the previous section.

4.3. Main Results

The following corollary of Theorem 4.2 completes the proof of Theorem 2 mentioned in the introduction:

Corollary 4.3. *Suppose that an* **NP***-complete language L reduces to Circuit-Learning in the agnostic model via a Turing reduction R of bounded adaptivity. Then,*

1. *If there exist a hard-on average language in* **NP** *then there exist one-way functions,* i.e. *Pessiland=Minicrypt.*
2. *If the reduction R is fully non-adaptive then the Polynomial Hierarchy collapse to the second level.*

The first part of the corollary follows from (the proof of) Theorem 4.2 and from the fact that if some language in **NP** is hard-on average then any **NP**-hard language is also hard on average.[9] To prove the second part, we note that our proof showed that a

[9]Indeed, if L is hard on average over the distribution Z and f is a Karp reduction from L to an **NP**-hard language $\hat{L}$ then $\hat{L}$ is hard on averae with respect to the distribution $f(Z)$.

fully non-adaptive reduction from L to learning gives an auxiliary-input one-way function whose security is based on the hardness of L via a "simple" (*i.e.* fixed-auxiliary-input, black-box non-adaptive) reduction R.[10] For such simple reductions, we can adapt the results of Akavia et al. [2] to put L in **CoAM** (details are deferred to the full version).

Note that any non-adaptive Turing reduction to CGL is a special case of a fully non-adaptive Turing reduction to Agnostic-Circuit-Learning. Hence Corollary 3.7 follows from Corollary 4.3.

Acknowledgements. We thank Tal Malkin and Parikshit Gopalan for fruitful discussions.

References

[1] W. Aiello and J. Hastad. Statistical zero-knowledge languages can be recognized in two rounds. *JCSS*, 42:327–345, 1991.

[2] A. Akavia, O. Goldreich, S. Goldwasser, and D. Moshkovitz. On basing one-way functions on NP-hardness. In *STOC '06*, pages 701–710, 2006.

[3] M. Alekhnovich, M. Braverman, V. Feldman, A. R. Klivans, and T. Pitassi. Learnability and automatizability. In *FOCS '04*, pages 621–630, 2004.

[4] S. Ben-David, N. Eiron, and P. M. Long. On the difficulty of approximately maximizing agreements. *J. Comput. Syst. Sci.*, 66(3):496–514, 2003.

[5] A. Blum, M. Furst, J. Jackson, M. Kearns, Y. Mansour, and S. Rudich. Weakly learning DNF and characterizing statistical query learning using fourier analysis. In *STOC '94*, pages 253–262, 1994.

[6] A. Blum, M. L. Furst, M. J. Kearns, and R. J. Lipton. Cryptographic primitives based on hard learning problems. In *CRYPTO '93*, pages 278–291, 1993.

[7] A. Blum and R. Rivest. Training a 3-node neural network is NP-complete. In *COLT '88*, pages 9–18.

[8] R. B. Boppana, J. Hastad, and S. Zachos. Does co-NP have short interactive proofs? *Inf. Process. Lett.*, 25(2):127–132, 1987.

[9] S. Even, A. L. Selman, and Y. Yacobi. The complexity of promise problems with applications to public-key cryptography. *Inf. Control*, 61(2):159–173, 1984.

[10] V. Feldman. Optimal hardness results for maximizing agreements with monomials. *CCC '06*, pages 226–236.

[11] V. Feldman. Hardness of approximate two-level logic minimization and PAC learning with membership queries. In *STOC '06*, pages 363–372, 2006.

[12] L. Fortnow. The complexity of perfect zero-knowledge. In *STOC '87*, pages 204–209, 1987.

[13] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. In *FOCS' 84*.

[14] O. Goldreich and E. Kushilevitz. A perfect zero-knowledge proof for a problem equivalent to discrete logarithm. In *CRYPTO '88*, pages 57–70, 1990.

[15] O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. In *FOCS'86*.

[16] O. Goldreich and S. Vadhan. Comparing entropies in statistical zero knowledge with applications to the structure of SZK. In *COCO '99*, pages 54–73, 1999.

[17] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems. In *STOC '85*, pages 291–304, 1985.

[18] P. Gopalan, S. Khot, and R. Saket. Hardness of reconstructing multivariate polynomials over finite fields. In *FOCS '07*, pages 349–359, 2007.

[19] V. Guruswami and P. Raghavendra. Hardness of learning halfspaces with noise. In *FOCS '06*, pages 543–552.

[20] T. Hancock, T. Jiang, M. Li, and J. Tromp. Lower bounds on learning decision lists and trees. *Inf. Comput.*, 126(2):114–122, 1996.

[21] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM J. Comput.*, 28(4):1364–1396, 1999.

[22] R. Impagliazzo. A personal view of average-case complexity. In *SCT '95*, page 134, 1995.

[23] R. Impagliazzo and L. A. Levin. No better ways to generate hard NP instances than picking uniformly at random. In *FOCS '90*, pages 812–821, 1990.

[24] R. Impagliazzo and M. Luby. One-way functions are essential for complexity based cryptography. In *FOCS '89*, pages 230–235, 1989.

[25] M. Kearns. Efficient noise-tolerant learning from statistical queries. *J. ACM*, 45(6):983–1006, 1998.

[26] M. Kearns, R. Schapire, and L. Sellie. Toward efficient agnostic learning. In *COLT '92*, pages 341–352, 1992.

[27] M. Kearns and L. Valiant. Cryptographic limitations on learning boolean formulae and finite automata. *J. ACM*, 41(1):67–95, 1994.

[28] M. Kharitonov. Cryptographic hardness of distribution-specific learning. In *STOC '93*.

[29] R. Ladner, N. Lynch, and A. Selman. Comparison of polynomial-time reducibilities. In *STOC '74*.

[30] M. H. Nguyen, S. J. Ong, and S. Vadhan. Statistical zero-knowledge arguments for NP from any one-way function. In *FOCS '06*.

[31] S. J. Ong and S. Vadhan. Zero knowledge and soundness are symmetric. In *EUROCRYPT '07*.

[32] R. Ostrovsky and A. Wigderson. One-way functions are essential for non-trivial zero-knowledge. In *ISTCS '93*, pages 3–17.

[33] L. Pitt and L. Valiant. Computational limitations on learning from examples. *J. ACM*, 35(4):965–984, 1988.

[34] L. Pitt and M. K. Warmuth. Prediction-preserving reducibility. *J. Comput. Syst. Sci.*, 41(3):430–467, 1990.

[35] L. G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, 1984.

[36] A. C. Yao. Theory and applications of trapdoor functions. In *FOCS '82*, pages 80–91, 1982.

[10] Namely, for any inverter A and for any z, if A inverts f_z with sufficiently large probability then, whp, $R^A(z)$ outputs $L(z)$.