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

Benny Applebaum[*] Boaz Barak[†] David Xiao[‡]

Abstract

We consider the question of whether **P** $\neq$ **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* **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.

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 [40] 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

[*] 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` .

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 [16, 24, 37]. 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 [36].[1] Such hardness results hold for several concept classes and for other variants and extensions of the PAC learning model (cf. [36, 7, 23, 4, 3, 12, 13, 22, 21]). 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, [21] 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 [25], Pessiland collapses to Minicrypt). The first consequence is considered to be implausible, while the latter would be a major breakthrough in cryptography.

[1] A different restriction on the power of the learner is studied in the Statistical Query model [28]. 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.

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 [35], 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 [9, 14, 35, 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 [16, 24, 35]. 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 [26, 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 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 $\mathsf{QuadraticResidue}, \mathsf{GraphIsomorphism}$ and $\mathsf{DiscreteLog}$ [20, 18, 17]. 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 [37] observed that LIH is implied by one-way functions by combining [16, 24], while hardness of learning specific concepts under specific cryptographic assumptions was shown in several other works including [29, 31].

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 [26, 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 $\mathbf{NP}$-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 $\mathbf{NP}$-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* $\mathbf{NP}$*-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 [32] (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 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 *[35] 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.*

[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 $\mathbf{NP}$-complete problem does not collapse the Polynomial Hierarchy by itself— in fact, assuming that OWFs exists, SAT has such a zero-knowledge protocol [33]. We collapse $\mathbf{PH}$ by relying on the special structure of reductions to circuit learning.

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 [25] 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 $L \in \mathbf{CoAM}$. When L is **NP**-complete this collapses the Polynomial-Hierarchy.

Extension to the Agnostic Setting. In the *agnostic* learning model of [30] 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 [21] 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 $z \in \mathsf{SAT}$. 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

[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.

$z \to (X_z, Y_z)$ 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]

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 [41]) 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 [25] 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 [27] and its equivalence to standard one-way functions. We also employ cryptographic reductions from [41, 16, 24]. Our use of "prediction via inversion" is similar to the notion of Universal Extrapolation of [26]. As we already mentioned we also use results from [35] 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 [34] and statistical zero-knowledge proofs [19].

[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.

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. The observation that auxiliary input one-way functions imply the LIH assumption appears in Section 5.

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 $x \overset{R}{\leftarrow} X$ 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 $\frac{1}{2}\sum_z |\Pr[X=z]-\Pr[Y=z]|$. We use some basic facts on the statistical distance mentioned in Appendix A.1.

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* [40] 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* [30] 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.

2.1 Auxiliary input primitives

Ostrovsky and Wigderson [35] 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. 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.) We move to a formal definition of auxiliary input primitives.

Definition 2.1. (Auxiliary-input functions) *An* auxiliary-input function *is a family $\mathcal{F} = \{f_z : \{0,1\}^{\ell(|z|)} \to \{0,1\}^{m(|z|)}\}_{z\in\{0,1\}^*}$, where $\ell(\cdot)$ and $m(\cdot)$ are polynomials; we will often write simply ℓ, m with the dependence on $|z|$ understood. We call $\mathcal{F}$ polynomial-time computable if there is a deterministic algorithm F running in time* $\mathrm{poly}(|z|)$ *such that for all $z \in \{0,1\}^*$ and $x \in \{0,1\}^{\ell(|z|)}$, we have $F(z,x) = f_z(x)$.*

Definition 2.2. (Auxiliary-input one-way function) *A polynomial-time computable auxiliary-input function $f_z : \{0,1\}^\ell \to \{0,1\}^m$ is an* auxiliary-input one-way function *(AIOWF) if for every probabilistic polynomial-time algorithm A, there exists an infinite set of strings $Z \subseteq \{0,1\}^*$, such that for every $z \in Z$ we have*

$$\Pr_{x \overset{R}{\leftarrow} U_{\ell(|z|)}} [A(z, f_z(x)) \in f_z^{-1}(f_z(x))] \leq \mathrm{neg}(|z|).$$

where $\text{neg}(|z|)$ *is a function that is smaller than* n^c *for all* c.

AIOWF's are natural generalizations of one-way functions, which is the special case where the auxiliary input z is given in unary instead of binary. Ostrovsky and Wigderson [35] proved the following:

Theorem 2.3 ([35])**.** *If* $\mathbf{ZK} \neq \mathbf{BPP}$ *then there exist AIOWF.*

It will be useful to consider the following two relaxed variants of AIOWF.

Definition 2.4. *Let* $f_z : \{0,1\}^{\ell(|z|)} \to \{0,1\}^{m(|z|)}$ *be polynomial-time computable auxiliary-input function family. Then,*

- **Auxiliary-input weak one-way function.** *The function family* f_z *is* weakly one-way *if there exists a polynomial* $p(\cdot)$*, such that for every probabilistic polynomial-time algorithm,* A*, there exists an infinite set of strings* $Z \subseteq \{0,1\}^*$*, such that for every* $z \in Z$ *we have* $\Pr[A(z, f_z(U_{\ell(|z|)})) \notin f_z^{-1}(f_z(U_{\ell(|z|)}))] > \frac{1}{p(|z|)}$ *(where the probability is taken over* $U_{\ell(|z|)}$ *and the internal coin tosses of* A*).*

- **Auxiliary-input distributional one-way function.** *We say that* A distributionally inverts f_z with distance ε *if the statistical distance between the distributions* $(A(z, f_z(U_\ell)), f_z(U_\ell))$ *and* $(U_\ell, f_z(U_\ell))$ *is at most* ε*. We say the function* f *is* distributionally one-way *if there exists a polynomial* $p(\cdot)$ *such that for every probabilistic polynomial-time algorithm* A*, there exists an infinite set of strings* $Z \subseteq \{0,1\}^*$*, such that for every* $z \in Z$*,* A *is unable to distributionally invert* f_z *with distance* $\frac{1}{p(|z|)}$.

It is well known that one can transform weak-OWFs or even distributional one-way functions to standard OWFs [41, 27]. These reductions carry over to the setting of auxiliary-input primitives (as most other cryptographic reductions); given an adversary A for the auxiliary-input weak-OWF (resp. auxiliary-input distributional OWF), the reductions of [41] construct an adversary A' for the underlying AIOWF. But by the definition of AIOWF there must exist a set Z of auxiliary inputs that is hard for A', and so Z constitutes also a hard set of auxiliary inputs for A.

Proposition 2.5 ([41, 27])**.** *(∃ auxiliary-input distributional-OWFs) ⇔ (∃ auxiliary-input OWFs) ⇔ (∃ auxiliary-input weak-OWFs). More precisely:*

1. *For any polynomials* p, q*, there is an efficient black-box reduction* (C, R) *that, for any auxiliary-input function family* f_z*, constructs another auxiliary-input function family* C^{f_z}*, and where* R *takes any black-box* A *that inverts* C^{f_z} *with success probability* $1/p(|z|)$ *and constructs* R^{A,f_z} *that inverts* f_z *with probability* $1 - 1/q(|z|)$*. Furthermore, the reduction is fixed-auxiliary-input,* i.e. *given auxiliary input* z_0*, the inverting reduction* R *only calls* A *with auxiliary input* z_0*.*

2. *For any polynomials* p, q*, there is an efficient black-box reduction* (C, R) *that, for any auxiliary-input function family* f_z*, constructs another auxiliary-input function family* C^{f_z}*, and where* R *takes any black-box* A *that inverts* C^{f_z} *with probability* $1/p(|z|)$ *and constructs* R^{A,f_z} *that distributionally inverts* f_z *with distance* $1/q(|z|)$*. Furthermore, the reduction is fixed-auxiliary-input.*

We will also work with pseudorandom functions [16], which are an efficiently-computable collection of functions which cannot be distinguished from a truly random function. As before, it is known [24, 16] how to construct PRF's from OWF's, and these reductions carry over to the auxiliary-input setting. Formally, we define:

Definition 2.6. Auxiliary-input pseudorandom functions. *An* AIPRF ensemble *is an efficiently-computable family $\mathcal{F} = \{f_z : \{0,1\}^{s(|z|)} \times \{0,1\}^{\ell(|z|)} \to \{0,1\}^{m(|z|)}\}_{z \in \{0,1\}^*}$, where $s(\cdot), \ell(\cdot)$ and $m(\cdot)$ are polynomials, that satisfies the following. For any efficient oracle algorithm A, there exists an infinite set $Z \subseteq \{0,1\}^*$ such that $\forall z \in Z$,*

$$|\Pr_{k \in \{0,1\}^s}[A^{f_z(k,\cdot)}(z) = 1] - \Pr_{\phi}[A^{\phi_{|z|}(\cdot)}(z) = 1]| \leq \mathrm{neg}(|z|)$$

where $\phi_{|z|}(\cdot) : \{0,1\}^{\ell(|z|)} \to \{0,1\}^{m(|z|)}$ is a truly random function.

Proposition 2.7 ([24, 16])**.** *($\exists$ auxiliary-input-OWFs) $\Leftrightarrow$ ($\exists$ auxiliary-input PRFs). More precisely, for any polynomials p, q, there is an efficient black-box reduction (C, R) that, for any auxiliary-input function family f_z, constructs another auxiliary-input function family C^{f_z}, and where R takes any black-box A that distinguishes C^{f_z} from a random function with advantage $1/p(|z|)$ and constructs R^{A,f_z} that inverts f_z with probability $1 - 1/q(n)$.*

3. Hardness of Learning via Karp Reduction Implies ZK Protocols

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* 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,* $\Pr[f(X) = Y] \leq \beta(n)$.

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

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 statistical sense (i.e., with respect to computationally unbounded parties). Specifically, we will show that SGL reduces to Entropy-Difference which is a complete problem for **SZKP**[19].

Definition 3.2 ([19]). *An input to* Entropy-Difference *consists of a pair of distributions* (W, Z) *represented by circuits.*

- *In YES inputs,* $H(Z) - H(W) \geq 1$.
- *In NO inputs,* $H(W) - H(Z) \geq 1$.

Where $H(\cdot)$ *is Shannon's entropy.*

Theorem 3.3. *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. We give a Karp reduction from $\mathsf{SGL}_{\alpha,\beta}$ to Entropy-Difference. This suffices as Entropy-Difference is in **SZKP** [19], and **SZKP** is closed under Karp reductions.

We begin by restricting our attention to the case where $\beta(n) \geq 0.55$ for every n. We will later show how to remove this condition. Let $C = (X, Y)$ be an instance of $\mathsf{SGL}_{\alpha,\beta}$, and let n be the size of the examples output by X. We map the joint distribution (X, Y) to the distributions $W = (X, Y)$ and $Z = (X, Y')$ where Y' is a Bernoulli random variable of success probability $(\alpha(n) + \beta(n))/2$. Clearly, the mapping is efficient. We claim that if (X, Y) is a YES instance, then $H(Z) - H(W) \geq 1/q(n)$ and if (X, Y) is a NO instance, then $H(W) - H(Z) \geq 1/q(n)$ for some polynomial $q(\cdot)$. This difference can amplified to 1 by taking $q(c)$ independent copies of W and Z.

First we write $H(W) - H(Z)$ as

$$H(X) + H(Y|X) - (H(X) + H(Y'|X)) = H(Y|X) - H_2(\alpha(c)/2 + \beta(c)/2),$$

where $[Y|X]$ is the conditional distribution of Y and $H_2(\cdot)$ denotes the binary entropy function which maps a real $0 < p < 1$ to $-p\log(p) - (1-p)\log(1-p)$. Let $\delta = \max_{b\in\{0,1\}} \Pr[[Y|X] = b]$. Note that $\delta \geq \alpha(c)$ in the YES case, and $\delta \leq \beta(c)$ in the NO case. Also, by definition, δ is at least 1/2. Therefore, since H_2 is decreasing in the interval $(1/2, 1)$, the entropy of $[Y|X]$ is at most $H_2(\alpha(c))$ in the YES case, and at least $H_2(\beta(c))$ in the NO case. Finally, we argue that when $\alpha(c) - \beta(c)$ is noticeable and $\beta(c) \geq 0.55$, the quantities $H_2(\alpha(c)/2 + \beta(c)/2) - H_2(\alpha(c))$ and $H_2(\beta(c)) - H_2(\alpha(c)/2 + \beta(c)/2)$ are also noticeable. This can be verified by examining the Taylor expansion of the binary entropy function.

It is left to justify the restriction to $\beta(n) \geq 0.55$. This is done by reducing $\mathsf{SGL}_{\alpha,\beta}$ to $\mathsf{SGL}_{\alpha',\beta'}$, where $\alpha' = 0.9\alpha + 0.1$ and $\beta' = 0.9\beta + 0.1 \geq 0.55$. This suffice as $\alpha' - \beta' = 0.9(\alpha - \beta)$ which is still noticeable. Let b be a Bernoulli random variable which equals to 0 with probability 1/10. We map an instance (X, Y) of $\mathsf{SGL}_{\alpha,\beta}$ to the distribution (X', Y') where $X' = (X, b)$ and Y' equals to 0 when $b = 0$, and equals to Y otherwise. The correctness of the reduction follows from the following simple claim:

Claim 3.4. *There exists a function* f *computable by a circuit of size* $t(c)$ *for which* $\Pr[f(X) = Y] \geq \varepsilon(c)$ *if and only if there exists a function* f' *computable by a circuit of size* $t(c) + \Theta(1)$ *for which* $\Pr[f'(X') = Y'] \geq 0.9\varepsilon(c) + 0.1$.

To prove the if direction, define f to be $f(x) = f'(x, 1)$. For the only-if direction, transform f to f' by letting $f'(X, b) = b \cdot f(X)$. This completes the proof of the theorem. □

Theorem 3.3 yields the following corollaries:

Corollary 3.5. *1. If a promise problem* Π *reduces to* SGL *via a Karp reduction, then* Π *has a statistical zero-knowledge proof. More generally, if* $\mathsf{SGL} \notin \mathbf{BPP}$ *then* $\mathbf{SZKP} \nsubseteq \mathbf{BPP}$.

2. There is no Karp reduction (or even non-adaptive Turing reduction) from SAT *to* SGL *unless the Polynomial Hierarchy (*$\mathbf{PH}$*) collapses.*

Proof. The first item follows from Theorem 3.3, and the fact that $\mathbf{SZKP}$ is closed under Karp-reductions. To prove the second item note that: (1) $\mathbf{SZKP} \subseteq \mathbf{CoAM}$ [1]; (2) If $\mathbf{NP} \subseteq \mathbf{CoAM}$, then the Polynomial Hierarchy collapses [9]; and (3) $\mathbf{CoAM}$ is closed under non-adaptive Turing reductions [10]. [8] □

The second item of the corollary shows that it is unlikely to base LIH on $\mathbf{NP}$-hardness via a Karp reduction to SGL. The first item shows that the intractability of $\mathbf{SZKP}$ is a necessary condition for proving LIH via a Karp reduction to SGL. As mentioned before some converse implication is also true, namely, LIH can be based on the intractability of $\mathbf{SZKP}$ (but not necessarily via Karp reduction).

Remark 3.6. *The first part of Corollary 3.5 holds even under* $\mathbf{NC}^1$ *truth-table reductions which strictly generalize standard Karp reductions. (See Appendix B for a formal definition.) This extension follows immediately from the fact that* $\mathbf{SZKP}$ *is closed under such reductions [38].*

3.2. Generalization to the 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 3.7. (Gap-Learning Problem – the computational version) *The input to the promise problem* $\mathsf{CGL}_{\alpha,\beta}$ *is a circuit* C *of* $\mathrm{poly}(n)$ *size 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 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 3.8. *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* $\mathbf{SZKA}$ *(the class of languages with statistically hiding but computationally sound zero knowledge proofs).*

The proof relies on the "SZK/OWF" characterization of $\mathbf{SZKA}$ recently shown by Ong and Vadhan [34].

[8] Note that we are dealing with classes of promise problems which are unlikely to be closed under arbitrary Turing reduction [10, 15].

Theorem 3.9 ([34, Thm. 1.4, Proposition 2.4, FN 3])**.** *A promise Problem $\Pi = (\Pi_{\text{Yes}}, \Pi_{\text{No}}) \in \mathbf{MA}$ has a statistical zero-knowledge argument system if and only if Π satisfies the following condition: there exists a set of instances $I \subseteq \Pi_{\text{No}}$ such that:*

- *The promise problem $(\Pi_{\text{Yes}}, \Pi_{\text{No}} \setminus I)$ is in* **SZKP***.*
- *There exists an efficiently computable function family $g_z : \{0,1\}^\ell \to \{0,1\}^m$ and a polynomial $q(\cdot)$ such that every non-uniform probabilistic polynomial-time adversary A fails to distributionally invert g_z with distance $1/q(|z|)$ on all sufficiently large $z \in I$.*

We will need the following lemma whose proof is implicit in Section 4.2.

Lemma 3.10. *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 non-efficient) 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 distance $\varepsilon(|z|)^3/(2|z|)$.*

Then, there exists an efficiently-computable hypothesis h for which

$$\Pr[h(X_z(U_{m(|z|)})) \neq Y((U_{m(|z|)}))] \leq \delta(|z|) + \varepsilon(|z|), \text{ for all } z \in I.$$

Proof of 3.8. We show that CGL satisfy the characterization of Theorem 3.9. It is not hard to verify that the problem is in **MA**. (Merlin sends Arthur a circuit f for which $\Pr[f(X) = Y] \geq \alpha$ and Arthur estimates $\Pr[f(X) = Y]$ up to an additive error of $(\alpha - \beta)/2$ by applying a Chernoff bound). We let the set I contain all the NO-instances (X, Y) for which there exists a function $f : \{0,1\}^n \to \{0,1\}$ that predicts Y with probability $(\alpha+\beta)/2$, that is $\Pr[f(X) = Y] > (\alpha+\beta)/2$. (Since (X, Y) is a NO-instance f is not efficiently computable.) By Theorem 3.3, the problem $\mathsf{CGL}_{\alpha,\beta} \setminus I = \mathsf{SGL}_{\alpha,\beta'=(\alpha+\beta)/2}$ is in **SZKP**. Let $z : \{0,1\}^{m(|z|)} \to \{0,1\}^{n(|z|)} \times \{0,1\}$ be an instance of $\mathsf{CGL}_{\alpha,\beta}$. That is z is a circuit that samples the joint distribution $(X, Y) = z(U_{m(|z|)})$. We define the function g_z from $\{0,1\}^{m(|z|)}$ to $\{0,1\}^{n(|z|)}$ as $g_z(r) = X(r)$, where $X(r)$ is the circuit that samples the distribution X (i.e., $X(r)$ outputs the first $n(|z|)$ bits of $z(r)$). The function g_z is computable in polynomial-time (since circuit evaluation is in **P**). We argue that g_z is distributionally hard to invert with success probability better than $\varepsilon(|z|)^3/(2|z|)$ for $\varepsilon(|z|) = (\alpha(|z|) - \beta(|z|))/4$ when $z \in I$.

Indeed, assume towards a contradiction that there exists a nonuniform probabilistic algorithm A that distributionally inverts $g_z(U_m)$ with distance $\varepsilon(|z|)^3/24|z|$ for infinitely many $z \in I$. Recall that for every $z = (X, Y) \in I$ there exists a function f for which $\Pr[f(X) = Y] \geq (\alpha(|z|) + \beta(|z|))/2$. Then, by Lemma 3.10, there exists a polynomial-size circuit h such that $\Pr[h(X) = Y] \geq (\alpha(|z|) + \beta(|z|))/2 - \varepsilon(|z|) = (\alpha(|z|) + 3\beta(|z|))/4 > \beta$ for infinitely many $z = (X, Y) \in I$, in contradiction to z being a NO-instances. □

Theorem 3.8 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 3.11. *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 4.8.

Remark 3.12. *The first part of Corollary 3.11 can be generalized to hold under monotone-*$\mathbf{NC}^1$ *truth-table reductions (defined in Appendix B) as one can prove that* **SZKA** *is closed under such reductions. (This follows by combining the results of [39, Cor. 7.12], and [34, Thm. 1.2].)*

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*[9] (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.*

[9] 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.

We first prove the theorem in the simpler case of a reduction which makes a single query to a PAC-learning oracle (Section 4.1), then examine the case of a single-query reduction in the agnostic setting (Section 4.2), and finally extend the proof to polynomially-many non-adaptive reductions and reductions of bounded adaptivity (Section 4.3). We end this section by drawing some corollaries from Theorem 4.2 and its proof (Section 4.4).

4.1. Single-query reduction in the PAC model

To give some idea of the proof we start with the much simpler case of a single-query deterministic reduction. Such a one-query reduction takes any instance z of the language L, and computes from it a query C_z. Here $C_z : \{0,1\}^{m(|z|)} \to \{0,1\}^{n(|z|)} \times \{0,1\}$ is a circuit that samples labeled examples from the distribution $(X, f_z(X))$, for some target function $f_z \in \mathcal{C}$.The reduction then gets a hypothesis h from the learner and applies some algorithm $M(z,h)$ to decide whether or not $z \in L$. We have that guarantee that if the hypothesis h is ε-good with respect to X (where ε is some inverse-polynomial accuracy parameter) then $M(z,h) = 1$ iff $z \in L$.

The proof follows the outline of Section 1.3. We will show the contrapositive: we describe a polynomial-time decision procedure for L under the assumption that auxiliary-input OWFs do not exist. This can be done by constructing an ε-good hypothesis h_z for the learning problem described by C_z, and then invoking $M(z,h)$ to decide z. However, it is not clear how to use the non-existence of auxiliary-input OWFs to learn the target function f_z. Instead, we will "learn" the *distribution* (X,Y) sampled by C_z. That is, we construct an algorithm h_z that given an example $x \overset{R}{\leftarrow} C_z(U_m(|z|))$ finds a label y such that $(x,y) \in \mathrm{Im}(C_z)$. Indeed, this is tantamount to inverting C_z which indeed can be done with high probability assuming that C_z is not even a weak auxiliary-input one-way function (which is the case under our assumption, see Section 2.1).

We move on to a formal description of h_z. Let $X_z(r), Y_z(r)$ denote the first and second elements of $C_z(r)$ respectively. Recall that, by Proposition 2.5, the non-existence of auxiliary-input OWFs implies that X_z cannot be even auxiliary-input weak-OWF. Hence, since $1/\varepsilon(\cdot)$ is bounded by a polynomial, there exists an efficient probabilistic inverting algorithm A such that for all $z \in \{0,1\}^*$ we have,

$$\Pr_{x \overset{R}{\leftarrow} X_z(U_{m(|z|)})} [A(z,x) \notin X_z^{-1}(x)] \leq \varepsilon(|z|)/12, \tag{1}$$

where the probability is also taken over the coin tosses of A. Let $t(|z|)$ be the randomness complexity of $A(z,\cdot)$. We define the randomized hypothesis h_z as follows:

1. Input: $x \in \{0,1\}^{n(|z|)}$. Random input: $s \in \{0,1\}^{t(|z|)}$.
2. Invoke the algorithm A on (z,x) with randomness s, and let r denote the output of A.
3. Output the label $y = Y_z(r)$.

Clearly, the resulting hypothesis is efficiently computable. First, we argue that the randomized hypothesis h_z is $(\varepsilon/12)$-good with respect to the target distribution X_z, and later we will de-randomize it. Assume, without loss of generality, that the query (X_z, Y_z) is honest (otherwise, h_z is ε-good by definition). Fix some $x \in \mathrm{Im}(X_z)$. Let r be a preimage of x under X_z. Then,

$$(x,y) = (X_z(r), Y_z(r)) = C_z(r) = (x, f_z(x)).$$

Therefore, whenever A finds an inverse of x under X_z, the hypothesis h_z labels x correctly. Hence, by Eq. 1, we have

$$\Pr_{x \overset{R}{\leftarrow} X_z(U_{m(|z|)}), s \overset{R}{\leftarrow} U_{t(|z|)}} [h_z(x;s) \neq f_z(x)] \leq \Pr_{x \overset{R}{\leftarrow} X_z(U_{m(|z|)}), s \overset{R}{\leftarrow} U_{t(|z|)}} [A(x;s) \notin X_z^{-1}(x)] \leq \varepsilon/12.$$

To obtain a deterministic hypothesis, we randomly choose a string $s \in \{0,1\}^t$ and fix the random coins of h_z to s. By a standard Markov argument (see Lemma A.4), except with probability 1/12, the resulting deterministic hypothesis $h_{z,s}(\cdot) \overset{\mathsf{def}}{=} h_z(\cdot; s)$ is ε-good. By a union bound, the resulting randomized procedure $M(z, h_{z,s}; \omega, s)$ decides z with error probability at most $1/3 + 1/12 = 5/12$ over the choice of the randomness (ω, s), which can amplified to $1/3$ (or to $2^{-|z|}$) using standard techniques. Hence, we get a BPP procedure for the language L, this completes the proof of the case of a single deterministically generated query.

Randomly generated queries. Now, suppose that the query is not deterministic that is, $C_{z,\omega} = (X_{z,\omega}, Y_{z,\omega}) = (X_{z,\omega}, f_{z,\omega}(X_{z,\omega}))$ is generated according to the input z and the randomness $\omega \in \{0,1\}^{\rho(|z|)}$. In this case, we will consider the (efficiently computable) function g_z which maps ω and r to the pair $(\omega, X_{z,\omega}(r))$. We will assume that this function is not an auxiliary-input weak-OWF. Hence, we have an efficient inverter A which, for every z, inverts $g_z(U_{\rho(|z|)}, U_{m(|z|)})$ with error probability $\varepsilon(|z|)/12$. We construct a randomized hypothesis $h_{z,\omega}(x; s)$ exactly as we did in the previous case, that is we label x by computing applying $Y_{z,\omega}$ to the string r that $A(z, (\omega, x); s)$ outputs (where s is chosen randomly and ω is fixed to the global randomness of the reduction). Note that, again, whenever A finds an inverse of (x, ω) under g_z, the hypothesis labels x correctly. Indeed, fix some z, ω and $x \in \mathrm{Im}(X_{z,\omega})$. Suppose that A found a preimage r of (x, ω) under g_z, and let y be the label outputted by $h_{z,\omega}$. Then,

$$(\omega, x, y) = (g_z(\omega, r), Y_{z,\omega}(r)) = (\omega, X_{z,\omega}(r), Y_{z,\omega}(r)) = (\omega, x, f_{z,\omega}(x)).$$

Therefore, for every z, we can bound $\Pr_{x,s,\omega}[h_{z,\omega}(x; s) \neq f_{z,\omega}(x)]$ by $\varepsilon(|z|)/12$, where the probability is taken over random s, ω and $x \overset{R}{\leftarrow} X_z(U_{m(|z|)})$. Hence, by Lemma A.4, for every z, with probability at least $1 - 11/12$ over the random choice of ω and s, the hypothesis $h_{z,\omega,s}(\cdot) \overset{\mathsf{def}}{=} h_{z,\omega}(\cdot; s)$ is ε-good with respect the query $C_{z,\omega}$. Hence, by a union bound, the BPP algorithm $M(z, h_{z,\omega,s}; \omega, s)$ decides z with error probability at most $1/3 + 1/12 = 5/12$ over the choice of the randomness (ω, s), and the lemma follows.

4.2. Single-query reduction 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 randomized reduction which makes a single query. Again, such a reduction is described by a pair of PPTs (T, M) and an accuracy parameter $\varepsilon(\cdot)$ where $T(z; \omega)$ outputs a query $C_{z,\omega} : \{0,1\}^{m(|z|)} \to \{0,1\}^{n(|z|)} \times \{0,1\}$ to the learner, and $M(z, h; \omega)$ attempts to decide whether $z \in L$. As before we prove the contrapositive: if there do not exist AIOWF then $L \in \mathbf{BPP}$. Let $C_{z,\omega} = (X_{z,\omega}, Y_{z,\omega})$ and let $f_{z,\omega}$ be a possibly (non-efficient) function which maximizes the agreement with the given query over all functions. That is, $f_{z,\omega}$ maximizes $\Pr[f_{z,\omega}(X_{z,\omega}) = Y_{z,\omega}]$. We will describe an efficient randomized procedure $B(z; \omega, s)$, that, with probability 11/12, outputs an hypothesis $h_{z,\omega}$ whose error probability is at most ε larger than the error probability of the optimal classifier $f_{z,\omega}$.

Such an hypothesis satisfies, in particular, the condition $\Pr[h_{z,\omega}(X_{z,\omega}) \neq Y_{z,\omega}] \leq \min_{\phi \in \mathcal{C}} \Pr[f(X_{z,\omega}) \neq Y_{z,\omega}] + \varepsilon(|z|)$ for any concept class $\mathcal{C}$. Therefore, given an access to such an hypothesis, the output of $M(z, h_{z,\omega}; \omega)$ is guaranteed to be correct with probability 2/3. As a result, we obtain a BPP algorithm for L which errs with probability $\leq 1/3 + 1/12 < 5/12$.

By Proposition 2.5, we may assume that g_z is not auxiliary-input distributional-OWF. Thus, for every inverse polynomial $\delta(\cdot)$ there exists an efficient probabilistic inverting algorithm A such that for all $z \in \{0,1\}^*$, we have

$$\Delta((z, (r, \omega), g_z(r, \omega)), (A(z, g_z(r, \omega); s), g_z(r, \omega))) \leq \delta(|z|), \tag{2}$$

where r, ω and s are uniformly chosen strings of appropriate lengths. Fix $\delta(|z|) = \varepsilon(|z|)^3/(24|z|)$ and let A be the corresponding adversary. Let $t(|z|)$ be the randomness complexity of $A(z, \cdot)$. We define the randomized hypothesis $h_{z,\omega}$ as follows:

1. Parameter: $q(|z|) = |z|/\varepsilon(|z|)^2$.
2. Input: $x \in \{0,1\}^{n(|z|)}$. Randomness: $\vec{s} = (s_1, \ldots, s_{q(|z|)})$ where $s_i \in \{0,1\}^{t(|z|)}$.
3. For $i = 1, \ldots, q(|z|)$:
   (a) Invoke the algorithm A on (z, ω, x) with randomness s_i, and let r_i denote the second output of A.
   (b) Let $y_i = Y_{z,\omega}(r_i)$.
4. Output the label $y = \text{Majority}_i(y_i)$.

Since A is efficient, the resulting hypothesis is efficiently computable. We prove that h is a good randomized hypothesis:

Lemma 4.3. *For every z, we have* $\Pr_{\omega,\vec{s}}[h_{z,\omega}(X_{z,\omega}) \neq Y_{z,\omega}] \leq \Pr_{\omega}[f_{z,\omega}(X_{z,\omega}) \neq Y_{z,\omega}] + \varepsilon(|z|)/12$.

Given Lemma 4.3, we can complete the proof of the theorem by letting $B(z; \omega, \vec{s})$ be the procedure which outputs the hypothesis $h_{z,\omega}(\cdot; \vec{s})$. By Lemma A.4, with probability 11/12 over $\vec{s}$, the resulting hypothesis is ε-good with respect to $f_{z,\omega}$ and $X_{z,\omega}$.

It is left to prove Lemma 4.3. Consider an (imaginary) "ideal" inverter $\hat{A}$ which satisfies Eq. 2 with $\delta(|z|) = 0$. Let $\hat{h}_{z,\omega}$ be the hypothesis which results from $h_{z,\omega}$ when A is replaced by $\hat{A}$. The proof of the claim follows by showing that (1) the performance of the real hypothesis $h_{z,\omega}$ is close to the performance of the ideal hypothesis $\hat{h}_{z,\omega}$; and (2) the ideal hypothesis $\hat{h}_{z,\omega}$ performs "almost" like the optimal classifier $f_{z,\omega}$. Formally, we prove the following two claims:

Claim 4.4. *For every z, the statistical distance between the random variable $h_{z,\omega}(X_{z,\omega}; \vec{s})$ and the random variable $\hat{h}_{z,\omega}(X_{z,\omega}; \vec{s})$ is at most $q(|z|) \cdot \delta = \varepsilon(|z|)/24$, where z, ω and $\vec{s}$ are uniformly chosen strings of appropriate lengths.*

Proof. By Eq. 2, for every z, the statistical distance between the random variable

$$(A(z, (\omega, X_{z,\omega}(r)); s), (\omega, X_{z,\omega}(r))), \tag{3}$$

and the random variable

$$(\hat{A}(z, (\omega, X_{z,\omega}(r)); s), (\omega, X_{z,\omega}(r))) \tag{4}$$

is bounded by $\delta(|z|)$. Note that real hypothesis $h_{z,\omega}(X_{z,\omega})$ simply applies some deterministic function Q to $q(|z|)$ copies of Eq. 3 where in each copy fresh randomness s is being used. The same holds with respect to the "perfect" hypothesis (where Q is applied to $q(|z|)$ copies of Eq. 4). Hence, by Facts A.1 and A.3, the claim follows. □

Claim 4.5. *For every z, we have $\Pr_{\omega,\bar{s}}[\hat{h}_{z,\omega}(X_{z,\omega}) \neq Y_{z,\omega}] \leq \Pr_{\omega}[f_{z,\omega}(X_{z,\omega}) \neq Y_{z,\omega}] + \varepsilon(|z|)/24$.*

Proof. First note that for every z and ω, the perfect inverter $\hat{A}(z,(\omega, X_{z,\omega}(r)))$ outputs ω and r' which is a random preimage of $X_{z,\omega}(r)$. Hence, we may fix z and ω and omit all the dependencies in them (e.g., we write $X = X_{z,\omega}, Y = Y_{z,\omega}, f = f_{z,\omega}, \varepsilon = \varepsilon(|z|)$ and so on). For every $x \in \mathrm{Im}(X)$ we let

$$\alpha(x) \stackrel{\mathsf{def}}{=} \max_{b\in\{0,1\}} \Pr_r[Y(r) = b|X(r) = x].$$

Observe that the function $f(x)$ always outputs the label $\arg\max_{b\in\{0,1\}} \Pr_r[Y(r) = b|X(r) = x]$. Fix x and let Y_x be the distribution of Y conditioned on $X = x$. Let χ_i be Bernoulli random variable which describes the i-th iteration of $\hat{h}(x)$. Specifically, $\chi_i = 1$ if and only if the algorithm $\hat{A}$ returns the label $f(x)$ in the i-th call, and $\chi_i = 0$ otherwise. Since $\hat{A}$ is perfect, we have $\mathbb{E}[\chi_i] = \alpha(x)$, also the hypothesis $\hat{h}(x)$ disagree with f only when $\sum_i \chi_i < q/2$. We consider two cases, the first for $\alpha(x) \geq 1/2 + \varepsilon(|z|)/25$ and the second for $\alpha(x) \leq 1/2 + \varepsilon(|z|)/25$.

Case 1 ($f(x)$ is easy to predict). Since $\alpha(x) \geq 1/2 + \varepsilon(|z|)/25$ we can use Chernoff bound to argue that $\hat{h}(x)$ almost always agrees with f. That is,

$$\Pr[\hat{h}(x) \neq f(x)] = \Pr\left[\frac{1}{q}\sum_{i=1}^{q(|z|)} \chi_i \leq 1/2\right] \leq \Pr\left[\alpha(x) - \frac{1}{q}\sum_{i=1}^{q(|z|)} \chi_i \geq \varepsilon(|z|)/25\right] \leq 2^{-\Omega(|z|)},$$

where the probability is taken over the randomness of $\hat{A}$. Therefore,

$$\Pr[\hat{h}(x) \neq Y_x] - \Pr[f(x) \neq Y_x] \leq 2^{-\Omega(|z|)}. \tag{5}$$

Case 2 ($f(x)$ is hard to predict). In this case, $\hat{h}$ disagrees with f with noticeable probability. However, the error probabilities of both, $\hat{h}$ and f, are close to 1/2, and therefore $\hat{h}$ performs almost as well as f. Formally, since $1/2 \leq \alpha(x) \leq 1/2 + \varepsilon(|z|)/25$, we have

$$\Pr[\hat{h}(x) \neq Y_x] - \Pr[f(x) \neq Y_x] \leq 1/2 - (1 - \alpha(x)) = \alpha(x) - 1/2 \leq \varepsilon(|z|)/25. \tag{6}$$

The claim follows from Eq. 5 and 6. □

Note that our argument actually proves Lemma 3.10.

4.3. Proof of Theorem 4.2

We continue with the proof of the general case.

Handling polynomially-many non-adaptive queries. We begin by considering the case of a non-adaptive randomized reductions that makes polynomially-many randomized queries (i.e., sends k distributions $(X_1, Y_1), \ldots, (X_k, Y_k)$ to the learner) based on the input z and randomness ω. Define the (efficiently computable) function g_z which maps the randomness ω, 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). We show that a sufficiently strong inverter for g succeeds in inverting $C_{z,\omega,i}$ for all i's and most ω's. Hence, it supplies a sequence of good hypotheses, which we use to build a decision procedure for the language.

Formally, assume that g_z is not an auxiliary-input distributional OWF, there exists an inverter A which distributionally inverts it on random input with deviation error of $\varepsilon(|z|)^3/(24|z| \cdot k(z))$. Hence, for every z and every fixed index $i \in [k(|z|)]$, the inverter A distributionally inverts $(\omega, i, X_i(r))$ with deviation error smaller than $\varepsilon(|z|)^3/(24|z|)$. Therefore, we can define the i-th (randomized) hypothesis $h_{z,\omega,i}(x; s)$ which invokes the algorithm A on (z, x) for $q = q(|z|) = |z|/\varepsilon(|z|)^2$ times (each time with independent randomness) and outputs the majority of $Y_z(r_i)$, where r_i denotes the i-th output of A. The proof proceeds with the same argument as in the previous section.

The adaptive case. Suppose that the reduction has $t = O(1)$ levels of adaptivity, that is we have efficient oracle machines $T_1, \ldots, T_t, M$ oracle machines such that at the i-th step the machine T_i generates polynomially many non-adaptive queries $((X_1, Y_1), \ldots, (X_{k(|z|)}, Y_{k(|z|)}))$ based on the input z, the global randomness ω, and an oracle access to the hypotheses given so far in response to the previous queries. At the final step $M(z; \omega)$ decides z based on an oracle access to all the hypotheses given by the learner. We will prove that such a reduction puts the language L in **BPP** (assuming that AIOWFs do not exist). The proof goes by induction on number of adaptive levels. For $t = 1$ we get the non-adaptive case. Suppose we have t levels of adaptivity. Then, the proof of the non-adaptive case shows how to answer the queries of the first step efficiently. More accurately, assuming that AIOWFs do not exist, there exists a probabilistic polynomial time procedure $B(z; \omega, s)$ which outputs a sequence of hypotheses $\vec{h} = (h_1, \ldots, h_{k(|z|)})$ which are ε-good for the queries of $T_1(z, \omega)$ with probability $1 - 1/(12t)$ (in fact, we can replace $12t$ with any arbitrary polynomial $p(|z|)$). Hence, we can replace the first two stages T_1, T_2 of the reduction by a single step in which we invoke $B(z; \omega, s)$ to generate the hypotheses $\vec{h}$, and then hand them to $T_2(z, \vec{h}; \omega)$. The claim now follows by applying the induction hypothesis. (The complexity of the procedure grows by a polynomial factor in each step, and so can only be repeated a constant number of times.)

Remark 4.6. *In fact, in the non-adaptive case, we proved the following stronger claim. Suppose that the language L reduces to Circuit-Learning via a non-adaptive Turing reduction. Then, we have a reduction from L to the task of inverting an auxiliary-input weak-OWF g which is "fixed" over the auxiliary input. Namely, there exists a PPT R and a polynomial $p(\cdot)$, such that for any inverter A for f and for any z, if A inverts f_z with success probability $1 - 1/p(|z|)$, then the machine R takes as an input the code of A and a string z and decides whether $z \in L$ with probability $2/3$. Furthermore, if L reduces to Circuit-Learning via a fully non-adaptive Turing reduction then R uses A in a black-box way and non-adaptively.*

4.4. Main Results

We say that a language L is hard on average if there exists an efficiently-samplable distribution ensemble $\{Z_n\}$, such that for every efficient algorithm A, and any polynomial $p(\cdot)$, we have $\Pr_z[A(z) = L(z)] \leq 1/2 + 1/p(|z|)$.

Lemma 4.7. *Suppose that the language L reduces to Circuit-Learning via a Turing reduction of bounded adaptivity. Then, if L is hard on average then (standard) one-way functions exist.*

Proof sketch. The non-adaptive case follows immediately from Remark 4.6. (The one-way function F samples an element z from the distribution Z and a random string x from the domain of g_z and outputs the pair $(z, g_z(x))$.) The proof of the $O(1)$-adaptive case is by induction similarly to the proof of Theorem 4.2 (the adaptive case). □

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

Corollary 4.8. *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 directly from Lemma 4.7. To prove the second part, we note that our proof showed that a 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 (see Remark 4.6). For such simple reductions, we can adapt the results of Akavia et al. [2] to put L in **CoAM** (see Appendix C).

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 the "moreover" part of Theorem 1 follows from Corollary 4.8.

5 Hardness of learning from AIOWF

In the opposite direction, it is not hard to see that the existence of AIOWF implies the hardness of learning small circuits, since from AIOWF we can construct auxiliary-input pseudorandom function (AIPRF) that are computable by small circuits, and by definition AIPRF are hard to learn.

Theorem 5.1. *If there exist AIPRF's then it is hard to learn small circuits (*i.e. *circuits that can compute the AIPRF's).*

Proof. Let f_z be a AIPRF function family. We claim that small circuits (just large enough to compute f_z) are hard to learn. For suppose not and there were such an A such that for all but finitely many z, there exist $p(n)$ and $p'(n)$ such that

$$\Pr_{A,k}[A^{(U_n, f_z(k,U_n))}(z) \text{ outputs } h \text{ that is } \tfrac{1+1/p(n)}{2} \text{ close to } f_z(k,\cdot)] > 1/p'(n)$$

Then we could build a distinguisher for f_z using A: the distinguisher A' with access to an oracle $\mathcal{O}$ would simply simulate A by simulating the example oracle that samples U at random and returns an example $(U, \mathcal{O}(U))$; after simulating A it gets back a hypothesis h. A' then performs the following sanity check to see whether h is indeed close to $\mathcal{O}$: sample $O(n(p(n))^2)$ times from U and checking whether h agrees with $\mathcal{O}$ on say $\geq \frac{1}{2} + \frac{1}{4p(n)}$ fraction of these samples. With probability $> 1 - 2^{-n}$, this test distinguishes between h with agreement $\geq \frac{1+1/p(n)}{2}$ from h with agreement $\leq 1/2$. If h passes the sanity check, sample one last example $x \overset{R}{\leftarrow} U$ output 1 if $h(x) = \mathcal{O}(x)$ and 0 otherwise, if it fails the sanity check then output a random bit.

To analyze this reduction, notice that if A is given an oracle $\mathcal{O} = f_z(k, \cdot)$ then with probability $> 1/p'(n)$ it obtains h with agreement at least $\frac{1+1/p(n)}{2}$ with $\mathcal{O}$. If A' obtains a good h then with probability $> 1 - 2^{-n}$ the sanity check passes and A' outputs 1 with probability $> \frac{1+1/p(n)}{2}$. On the other hand, if it obtains a bad h that has agreement less than $\frac{1+1/p(n)}{2}$ then A' outputs 1 with probability at least $1/2$: if the agreement of h and $\mathcal{O}$ is less than $1/2$ then our sanity check will almost always detect it in which case we will output a random bit; on the other hand if its agreement is $> 1/2$ then on the final example x clearly A will output 1 with probability whatever the agreement is. So overall its probability of outputting 1 is at least $\frac{1}{2} + \frac{1}{2p'(n)p(n)}$ (up to some negligible terms from the Chernoff bounds).

If A is given an oracle $\mathcal{O} = \phi$, then its probability of outputting 1 is $\leq 1/2 + 2^{-n}$. This gives a non-negligible advantage $\frac{1}{2p'(n)p(n)}$ (up to negligible terms) and proves the claim.

□

Since Ostrovsky and Wigderson [35] show that $\mathbf{ZK} \neq \mathbf{BPP}$ implies the existence of AIOWF, we have the following:

Corollary 5.2. *If* $\mathbf{ZK} \neq \mathbf{BPP}$*, then it is hard to learn small circuits.*

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, 1991.

[2] A. Akavia, O. Goldreich, S. Goldwasser, and D. Moshkovitz. On basing one-way functions on np-hardness. In *STOC '06: Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 701–710, New York, NY, USA, 2006. ACM.

[3] M. Alekhnovich, M. Braverman, V. Feldman, A. R. Klivans, and T. Pitassi. Learnability and automatizability. In *FOCS '04: Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science*, pages 621–630, Washington, DC, USA, 2004. IEEE Computer Society.

[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: Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 253–262, New York, NY, USA, 1994. ACM.

[6] A. Blum, M. L. Furst, M. J. Kearns, and R. J. Lipton. Cryptographic primitives based on hard learning problems. In *CRYPTO '93: Proceedings of the 13th Annual International Cryptology Conference on Advances in Cryptology*, pages 278–291, London, UK, 1993. Springer-Verlag.

[7] A. L. Blum and R. L. Rivest. Training a 3-node neural network is np-complete. *Neural Netw.*, 5(1):117–127, 1992.

[8] A. Bogdanov and L. Trevisan. On worst-case to average-case reductions for np problems. *SIAM J. Comput.*, 36(4):1119–1159, 2006.

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

[10] 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.

[11] J. Feigenbaum and L. Fortnow. Random-self-reducibility of complete sets. *SIAM J. Comput.*, 22(5):994–1005, 1993.

[12] V. Feldman. Hardness of approximate two-level logic minimization and pac learning with membership queries. In *STOC '06: Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 363–372, New York, NY, USA, 2006. ACM.

[13] V. Feldman. Optimal hardness results for maximizing agreements with monomials. *ccc*, 0:226–236, 2006.

[14] L. Fortnow. The complexity of perfect zero-knowledge. In *STOC '87: Proceedings of the nineteenth annual ACM conference on Theory of computing*, pages 204–209, New York, NY, USA, 1987. ACM.

[15] O. Goldreich. On promise problems (a survey in memory of shimon even [1935-2004]). Technical report, February 2005.

[16] O. Goldreich, S. Goldwasser, and S. Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, 1986. Preliminary version in FOCS' 84.

[17] O. Goldreich and E. Kushilevitz. A perfect zero-knowledge proof for a problem equivalent to discrete logarithm. In *CRYPTO '88: Proceedings on Advances in cryptology*, pages 57–70, New York, NY, USA, 1990. Springer-Verlag New York, Inc.

[18] O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. *Journal of the ACM*, 38(3):691–729, July 1991. Preliminary version in FOCS' 86.

[19] O. Goldreich and S. Vadhan. Comparing entropies in statistical zero knowledge with applications to the structure of szk. In *COCO '99: Proceedings of the Fourteenth Annual IEEE Conference on Computational Complexity*, page 54, Washington, DC, USA, 1999. IEEE Computer Society.

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

[21] P. Gopalan, S. Khot, and R. Saket. Hardness of reconstructing multivariate polynomials over finite fields. In *FOCS '07: Proceedings of the 48th Annual IEEE Symposium on Foundations of Computer Science*, pages 349–359, Washington, DC, USA, 2007. IEEE Computer Society.

[22] V. Guruswami and P. Raghavendra. Hardness of learning halfspaces with noise. In *FOCS '06: Proceedings of the 47th Annual IEEE Symposium on Foundations of Computer Science*, pages 543–552, Washington, DC, USA, 2006. IEEE Computer Society.

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

[24] 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. Preliminary versions appeared in STOC' 89 and STOC' 90.

[25] R. Impagliazzo. A personal view of average-case complexity. In *SCT '95: Proceedings of the 10th Annual Structure in Complexity Theory Conference (SCT'95)*, page 134, Washington, DC, USA, 1995. IEEE Computer Society.

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

[27] R. Impagliazzo and M. Luby. One-way functions are essential for complexity based cryptography (extended abstract). In *Proc. 30th FOCS*, pages 230–235. IEEE, 1989.

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

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

[30] M. J. Kearns, R. E. Schapire, and L. M. Sellie. Toward efficient agnostic learning. In *COLT '92: Proceedings of the fifth annual workshop on Computational learning theory*, pages 341–352, New York, NY, USA, 1992. ACM.

[31] M. Kharitonov. Cryptographic hardness of distribution-specific learning. In *STOC '93: Proceedings of the twenty-fifth annual ACM symposium on Theory of computing*, pages 372–381, New York, NY, USA, 1993. ACM.

[32] R. Ladner, N. Lynch, and A. Selman. Comparison of polynomial-time reducibilities. In *STOC '74: Proceedings of the sixth annual ACM symposium on Theory of computing*, pages 110–121, New York, NY, USA, 1974. ACM.

[33] M.-H. Nguyen, S.-J. Ong, and S. Vadhan. Statistical zero-knowledge arguments for np from any one-way function. In *FOCS 2006*, pages 3–14, 2006.

[34] S. J. Ong and S. P. Vadhan. Zero knowledge and soundness are symmetric. In *EUROCRYPT*, pages 187–209, 2007.

[35] R. Ostrovsky and A. Wigderson. One-way functions are essential for non-trivial zeroknowledge, 1993.

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

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

[38] A. Sahai and S. P. Vadhan. A complete promise problem for statistical zero-knowledge. In *38th Annual Symposium on Foundations of Computer Science*, pages 448–457, Miami Beach, Florida, 20–22 Oct. 1997. IEEE.

[39] S. P. Vadhan. An unconditional study of computational zero knowledge. *focs*, 00:176–185, 2004.

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

[41] A. C. Yao. Theory and applications of trapdoor functions. In *Proc. 23rd FOCS*, pages 80–91. IEEE, 1982.

A Some useful facts

A.1 Statistical distance

Fact A.1. *For all distributions X, Y and every (possibly randomized) function f we have $\Delta(f(X), f(Y)) \leq \Delta(X, Y)$.*

Let $A \otimes B$ denotes the product distribution of A, B, i.e., the joint distribution of independent samples from A and B.

Fact A.2. *For all distributions X, X', Y, Y' we have $\Delta((X \otimes X'), (Y \otimes Y')) \leq \Delta(X, Y) + \Delta(X', Y')$.*

Fact A.3. *Let X be a distribution, and let f and g be randomized algorithms for which the statistical distance between the random variable $(X, f(X))$ and the random variable $(X, g(X))$ is at most ε. Then, for every integer t, the statistical distance of $(X, \otimes^t f(X))$ and $(X, \otimes^t g(X))$ is at most $t \cdot \varepsilon$.*

Proof. By the definition of statistical distance we can write,

$$\begin{aligned}
\Delta((X, \otimes^t f(X)), (X, \otimes^t g(X))) &= \mathbb{E}_{x \overset{R}{\leftarrow} X}[\Delta(\otimes^t f(x), \otimes^t g(x))] \\
&\leq \mathbb{E}_{x \overset{R}{\leftarrow} X}[t \cdot \Delta(f(x), g(x))] \\
&= t \cdot \mathbb{E}_{x \overset{R}{\leftarrow} X}[\Delta(f(x), g(x))] \\
&= t \cdot \Delta((X, f(X)), (X, g(X))) \leq t \cdot \varepsilon,
\end{aligned}$$

where the first inequality is due to Fact A.2, and the second equality follows from the linearity of the expectation. □

A.2 Derandomizing good hypotheses

The following fact shows that if h is a good randomized hypothesis, then the deterministic hypothesis $h(\cdot; s)$ which results from h by randomly fixing its randomness to s is, with high probability, also good.

Fact A.4. *Let X be a target distribution. Let f be a target function and h be a hypothesis, both possibly randomized. Suppose that $\Pr[f(X) \neq h(X)] \leq \varepsilon$, where the probability is taken over X and the coin tosses of f and h. Then, for any $k > 0$, we have*

$$\Pr_s[\Pr[f(X) \neq h_s(X)] \geq k \cdot \varepsilon] \leq 1/k,$$

where h_s is the deterministic hypothesis obtained by fixing the randomness of h to s, and the internal probability is taken over X and the coin tosses of f.

Proof. Let $\chi_{x,s,r}$ be a Bernoulli random variable which equals to 1 if and only if $h(x; s) \neq f(x; r)$, where x is distributed according to X and s, r are the random coin tosses of h and f. Then, $\mathbb{E}_{x,s,r}[\chi_{x,s,r}] = \mathbb{E}_s[\mathbb{E}_{x,r}[\chi_{x,s,r}]] \leq \varepsilon$. Therefore, by Markov's inequality, we have, $\Pr_s[\mathbb{E}_{x,r}[\chi_{x,s,r}] \geq k \cdot \varepsilon] \leq 1/k$. The claim follows by noting that $\mathbb{E}_{x,r}[\chi_{x,s,r}] = \Pr_{r, x \overset{R}{\leftarrow} X}[f(x; r) \neq h(x; s)]$ for any fixed s. □

B Truth-Table Reductions

Consider variables taking value in $\{0, 1, \varnothing\}$. We extend standard boolean algebra so that $\neg\varnothing = \varnothing$, $\varnothing \wedge 1 = \varnothing \wedge \varnothing = \varnothing$, $\varnothing \wedge 0 = 0$, $\varnothing \vee 1 = 1$, $\varnothing \vee 0 = \varnothing \vee \varnothing = \varnothing$. For a promise problem Π, let

$$\chi_\Pi(x) = \begin{cases} 1 & x \in \Pi_Y \\ 0 & x \in \Pi_N \\ \varnothing & \text{else} \end{cases}$$

Definition B.1. *A promise problem Π reduces to a promise problem Γ via a (polynomial-time)* truth-table reduction *if there exists an efficient (possibly randomized) algorithm R taking an instance x of Π and some random bits and outputting a polynomial-size circuit C and instances $y_1, \ldots, y_k$ of Γ such that over the probability of the reduction R:*

$$x \in \Pi \Leftrightarrow \Pr[C(\chi_\Gamma(y_1), \ldots, \chi_\Gamma(y_k)) = 1] \geq 2/3$$
$$x \notin \Pi \Leftrightarrow \Pr[C(\chi_\Gamma(y_1), \ldots, \chi_\Gamma(y_k)) = 1] \leq 1/3$$

If R always outputs an $\mathbf{NC}^1$ circuit (resp. monotone $\mathbf{NC}^1$ circuit) then the reduction is called $\mathbf{NC}^1$ (rep. monotone $\mathbf{NC}^1$) truth-table reduction.

C Collapsing PH via [2]

Here we sketch the proof of the second part of Corollary 4.8. The idea is that applying Theorem 4.2 to a fully non-adaptive reduction from deciding L to learning $\mathcal{C}$ actually gives a reduction from deciding L to inverting AIOWF g_z with additional structure that can then be exploited to get more consequences. In particular, the reduction uses the inverter non-adaptively in a black-box way, and it is "fixed" over the auxiliary input (see Remark 4.6). We can therefore rely on the following theorem which is implicit in the work of Akavia et al. [2].

Theorem C.1. *Suppose that there is a non-adaptive fixed-auxiliary-input black-box reduction from inverting L to inverting AIOWF. Then $L \subseteq$ **CoAM**.*

Proof sketch. We apply the main idea of [2], which is to construct an **AM** protocol for the complementary language $\bar{L}$ by forcing the prover in the protocol to act as if it were an honest oracle that inverts an AIOWF.

The challenge is to force the prover never to cheat, *e.g.* by falsely claiming that some of R' queries are not invertible, or by adaptively choosing the inverses in a way that will affect R''s execution. This is done in [2], using techniques from [11, 8], by using hashing and hiding protocols.

We can in fact apply [2]'s proof verbatim to our setting except their use of a OWF with our AIOWF (with auxiliary input z_0) and replacing calls to their OWF-inverting oracle to our AIOWF-inverting oracle (with auxiliary input z_0). Recall roughly what [2] does:

1. Estimate the fraction of queries made by the reduction that are heavy.
2. Estimate the average preimage size of the reduction's queries conditioned on the query not being heavy using a hiding protocol.

3. Execute the reduction $p(n)$ many times to get queries $\vec{y}_1, \ldots, \vec{y}_p$ (each $\vec{y}_i$ is a list of the queries that the reduction would make). Get proofs from the prover about which queries are heavy; if the number of heavy queries is far from the estimate of the fraction of queries from the first step, reject.

4. Ask the prover to prove lower bounds on the preimage sizes of all the queries that are light, and check that the average is close to the estimate obtained in the second step. Reject if the average preimage size of the $\vec{y}_i$ is far from the estimate of the previous step.

5. Choose $j \in [q]$ at random and run the reduction using the queries $\vec{y}_j$.

To apply this to our setting, observe that the only difference between our setting and that of [2] is the [2] assume that OWF is uniformly computable where as in our case we need the additional auxiliary input z_0 to compute the function. But the uniform computability is only used in the following ways:

1. To sample input-output pairs $(x, f(x))$.

2. To apply the Goldwasser-Sipser lower bound and Aiello-Hastad upper bound protocols to approximate the size of $f^{-1}(y)$.

3. To verify that the inverting oracle returned a x that is a correct preimage, *i.e.* to verify that $f(x) = y$.

Replacing the OWF by an AIOWF clearly does not affect any of these uses because both the prover and verifier have access to the auxiliary input z_0, and so they can both use $f_{z_0}(\cdot)$ where the [2] uses $f(\cdot)$. □

We should note that removing the restriction that R' only query the inverting orace on auxiliary input z_0 would prevent us from applying [2] to our setting. To see a simple contrived example why [2] fails in this case, consider a reduction R' that gets input z_0 and queries its inverting oracle on $(y_1, y_1), \ldots, (y_k, y_k)$ where y_i are chosen at random; *i.e.* the auxiliary input and the input are identical. Then it is not clear how to hide queries to obtain the necessary statistics because the prover will always be able to recognize queries from the reduction, since the auxiliary input and standard input are equal.