On basing $\mathbf{ZK} \neq \mathbf{BPP}$ on the hardness of PAC learning

David Xiao
Computer Science Department
Princeton University
Princeton, NJ 08540
dxiao@cs.princeton.edu

May 27, 2009

Abstract

Learning is a central task in computer science, and there are various formalisms for capturing the notion. One important model studied in computational learning theory is the PAC model of Valiant (CACM 1984). On the other hand, in cryptography the notion of "learning nothing" is often modelled by the simulation paradigm: in an interactive protocol, a party learns nothing if it can produce a transcript of the protocol by itself that is indistinguishable from what it gets by interacting with other parties. The most famous example of this paradigm is zero knowledge proofs, introduced by Goldwasser, Micali, and Rackoff (SICOMP 1989).

Applebaum *et al.* (FOCS 2008) observed that a theorem of Ostrovsky and Wigderson (ISTCS 1993) combined with the transformation of one-way functions to pseudo-random functions (Håstad *et al.* SICOMP 1999, Goldreich *et al.* J. ACM 1986) implies that if there exist non-trivial languages with zero-knowledge arguments, then no efficient algorithm can PAC learn polynomial-size circuits. They also prove a weak reverse implication, that if a certain non-standard learning task is hard, then zero knowledge is non-trivial. This motivates the question we explore here: can one prove that hardness of PAC learning is *equivalent* to non-triviality of zero-knowledge? We show that this statement cannot be proven via the following techniques:

1. Relativizing techniques: there exists an oracle relative to which learning polynomial-size circuits is hard and yet the class of languages with zero knowledge arguments is trivial.
2. Semi-black-box techniques: if there is a black-box construction of a zero-knowledge argument for an NP-complete language (possibly with a non-black-box security reduction) based on hardness of PAC learning, then NP has statistical zero knowledge proofs, namely NP is contained in SZK.

Under the standard conjecture that NP is not contained in SZK, our results imply that most standard techniques do not suffice to prove the equivalence between the non-triviality of zero knowledge and the hardness of PAC learning. Our results hold even when considering non-uniform hardness of PAC learning with membership queries. In addition, our technique relies on a new kind of separating oracle that may be of independent interest.

1 Introduction

PAC learning was one of the earliest models studied in computational learning theory [Val84], and understanding what efficient algorithms can learn in the PAC model remains an important goal.

A learning algorithm is said to PAC learn a *concept class* F (*e.g.* linear functions over $\mathbb{F}_2^n$, half-spaces, DNF's) if given access to many labelled examples (x, y) drawn from a distribution $(X, f(X))$ where X is an arbitrary input distribution and $f \in F$, the learner outputs with high probability a *hypothesis* h (expressed as a circuit) such that $\Pr_X[f(X) \neq h(X)]$ is small. Unfortunately, there are a variety of seemingly elementary classes of functions for which we still know only sub-exponential or quasi-polynomial learning algorithms (*e.g.* DNF [KS01, LMN93]). In fact, it has been shown that various concept classes are hard to learn based on cryptographic assumptions [GGM86, PW90] or even based on **NP**-hardness if we restrict the form of the hypothesis h the learner outputs (*e.g.* k-DNF [PV88]; it seems unlikely that we can prove hardness of learning based on **NP**-hardness using standard techniques if h is unrestricted, see [ABX08]). Throughout this paper, we say that PAC learning is hard if size n^2 circuits are hard to learn. By a standard padding argument, the n^2 bound can be replaced by any n^c for any constant $c > 1$ without affecting our results. We consider the problem of learning polynomial-size circuits because circuits are a universal model of computation.

In cryptography, a different notion of "learning" was developed in the study of *zero knowledge proof systems* [GMR85]. In this context, the goal was to construct proof systems where an unbounded prover P interacts with an efficient verifier V in order to prove a statement such that the verifier "learns nothing" except that the statement is true. In this setting, we say that V learns nothing if it is able to simulate its interaction with the prover by itself, so anything the verifier could compute after interacting with the prover, it could compute without interaction.

Although these notions superficially seem unrelated besides intuitively capturing some notion of "learning", Applebaum, Barak, and the author [ABX08] observed that a theorem of Ostrovsky and Wigderson (Theorem 2.1) which states that $\mathbf{ZK} \neq \mathbf{BPP}$ implies the existence of "auxiliary-input one-way functions" (defined in Section 2), combined with the standard transformation of one-way functions to pseudorandom functions [HILL89, GGM86] together show that if there are non-trivial zero knowledge protocols (*i.e.* $\mathbf{ZK} \neq \mathbf{BPP}$) then PAC learning is hard.

Already [ABX08] showed a partial reverse implication, working with the promise problem $\mathsf{Learnability}$, defined as follows. Consider circuits $C : \{0,1\}^m \to \{0,1\}^{n+1}$, and let X denote the distribution on the first n bits of $C(U_m)$ where U_m is uniform on $\{0,1\}^m$, and let Y denote the distribution of the last bit of $C(U_m)$. C is a YES instance of $\mathsf{Learnability}$ if there exists a function f computable by a circuit of size n^2 such that the distribution $(X, Y) = (X, f(X))$. That is, a YES instance is "learnable" because the problem of PAC learning (X, Y) has at least one solution. On the other hand, C is a NO instance if the distribution (X, Y) is such that for all functions g computable by circuits of size $n^{\log \log n}$, $\Pr[Y = g(X)] \leq 3/4$ (the choice of $n^{\log \log n}$ and $3/4$ can be replaced by any superpolynomial function and number bounded away from 1). That is, a NO instance is "unlearnable" because no good hypothesis for labelling (X, Y) exists.

[ABX08] show that $\mathsf{Learnability} \in \mathbf{ZK}$, and so if $\mathsf{Learnability} \notin \mathbf{BPP}$ then $\mathbf{ZK} \neq \mathbf{BPP}$. One might hope it is also possible to show that hardness of standard PAC learning implies $\mathbf{ZK} \neq \mathbf{BPP}$. To do so, one would have to generalize the techniques above to handle the search problem of *finding* the hidden labelling function rather than simply deciding whether it exists, as well as deal with the fact that in standard PAC learning one does not have access to the circuit generating labelled examples. In this paper we show that many standard proof techniques do not suffice to prove that the hardness of PAC learning implies $\mathbf{ZK} \neq \mathbf{BPP}$.

1.1 Our results

Throughout this paper we say learning is hard if every non-uniform algorithm (equivalently family of circuits) fails to learn the concept class of functions computable by circuits of size n^2 under the uniform input distribution[1] on all but finitely many input lengths, given access to an example oracle *and* a membership oracle (see Section 2 for formal definitions). This notion is extremely strong (in particular it implies the standard notions of hardness), and we consider such a notion in order to obtain stronger results.[2] Likewise, there are various notions of zero knowledge (see *e.g.* [OV07]), but in order to obtain stronger results, we consider the broad notion of zero knowledge where the zero-knowledge property is only required against an honest-but-curious verifier and efficient distinguisher, and the soundness property is required only against efficient cheating provers. Following [OV07], we let **HV-CZKA** denote the class of languages with such protocols. In particular, by ruling out even proofs that use hardness of learning to show that this broad notion of zero knowledge is non-trivial, we also rule out proofs for more restricted notions of zero knowledge (*e.g.* with soundness against unbounded cheating provers or small statistical simulator deviation). In this paper, **ZK** always refers to **HV-CZKA** (defined formally in Section 2).

Relativizing proofs: Our first theorem shows that relativizing techniques cannot prove that if learning is hard, then $\mathbf{ZK} \neq \mathbf{BPP}$.

Theorem 1.1. *There exists an oracle $\mathcal{O}$ for which PAC learning is hard, but $\mathbf{ZK}^{\mathcal{O}} = \mathbf{BPP}^{\mathcal{O}}$.*

In fact, we prove the stronger statement that relative to $\mathcal{O}$, learning is hard but there exist no auxiliary-input one-way functions (AIOWF), which then implies $\mathbf{ZK}^{\mathcal{O}} = \mathbf{BPP}^{\mathcal{O}}$ by the theorem of Ostrovsky and Wigderson (stated in Theorem 2.1). We define AIOWF formally in Section 2, but in essence AIOWF do not exist if and only if there is an inverter $I(f, y)$ such that for every efficient function f, given as a circuit, $\Pr_{I,x}[I(f, y) \in f^{-1}(y) \mid y = f(x)]$ is non-negligible where the probability is over uniform x and the internal coin tosses of I. Notice the contrast with the usual notion of one-way functions, where the function f is fixed ahead of time and not given as input.

Unfortunately, in this setting ruling out relativizing proofs is not very convincing because we have non-relativizing proofs that base $\mathbf{ZK} \neq \mathbf{BPP}$ on various complexity assumptions. In particular the celebrated result of Goldreich, Micali, and Wigderson [GMW86], which proves that **NP** has a zero knowledge protocol based on the existence of one-way functions, does not relativize because they work directly with the explicit **NP**-complete problem Three Coloring (3-COL).

Black-box proofs: [GMW86] does not relativize, but it is black-box: they require only black-box access to a one-way function to construct a zero-knowledge protocol for 3-COL. Our next result rules out *black-box proofs* that zero knowledge is non-trivial based on the hardness of learning. A fully-black-box proof uses black-box access to a collection of functions F that are hard to learn (specified by an oracle taking two inputs, one an index specifying some $f \in F$ and another the input to f) in order to construct a zero-knowledge protocol, and also provides an analysis that uses an adversary for breaking zero knowledge or breaking computational soundness as a black-box to build a learning algorithm. In a semi-black-box proof, the analysis may use the *code* of the

[1] Actually any input distribution with super-polynomial min-entropy also works, but we only consider the uniform distribution for simplicity.

[2] Indeed, working with hardness of learning for uniform algorithms and only requiring that infinitely many input lengths be hard would make our results less convincing, since many existing proofs basing $\mathbf{ZK} \neq \mathbf{BPP}$ on various hardness assumptions rely on non-uniform hardness assumptions where all but finitely many input lengths are hard *e.g.* [GMW86]

adversary breaking zero knowledge as well. See the taxonomy of Reingold, Trevisan, and Vadhan [RTV04] for more details about classifying black-box proofs.

Unlike Theorem 1.1, our second theorem does not unconditionally rule out black-box proofs because there *are* zero knowledge protocols whose security is unconditional (*e.g.* for Graph Isomorphism, Quadratic Residuosity). It is conceivable that even 3-COL has such a protocol (*i.e.* $\mathbf{NP} \subseteq \mathbf{SZK}$, defined as in Section 2), in which case its security proof would use no complexity assumptions and hence would be trivially black-box. This is considered unlikely, and we prove that this is the only possibility:

Theorem 1.2. *If there exists a semi-black-box proof that constructs a* **ZK** *protocol for a language L assuming PAC learning is hard, then in fact $L \in \mathbf{SZK}$.*

Under the standard conjecture that $\mathbf{NP} \not\subseteq \mathbf{SZK}$, Theorem 1.2 says that such proofs for an **NP**-complete language L cannot exist.

1.2 Our techniques

Proving Theorem 1.1: the intuitive difference between PAC learning and inverting AIOWF we exploit is that in PAC learning, the learner knows nothing about how the labelled examples (X, Y) are produced, whereas with AIOWF, the inverting algorithm *does* know a description of the function f it is trying to invert.

Our oracle will be defined using a distribution over functions $\mathcal{R}^{(n)} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, which defines the collection of functions $\mathcal{R}_z = \mathcal{R}^{(n)}(z, \cdot)$. For each $z \in \{0,1\}^n$, with probability $2^{-n/2}$ the distribution sets z to be a "hard instance", *i.e.* it sets $\mathcal{R}_z$ to be a uniformly random function, and with probability $1 - 2^{-n/2}$ it sets $\mathcal{R}_z$ to be the all zero function $\mathcal{R}_z \equiv 0$.

We show (in Lemma 3.3) that almost surely over the choice of $\mathcal{R}$, the concept class $F = \{\mathcal{R}_z\}_{z \in \{0,1\}^*}$ is hard to learn for non-uniform algorithms with $\mathcal{R}$ gates. The intuition is that there are roughly $2^{n/2}$ hard instances z on inputs of length n, and they are chosen at random, so no polynomial-size circuit can find all of them, and no circuit can learn hard instances it cannot find because hard instances are random functions. Notice that we must choose *many* hard instances because a circuit's non-uniform advice can be specified *after* the oracle is chosen, and so the advice may reveal where some of the hard instances are hidden; by choosing $2^{n/2}$ hard instances, no polynomial amount of advice can specify all of the hard instances, and so for any polynomial-size circuit some hard instances remain random-looking.

The second condition is to check that AIOWF do not exist. One idea to assure this is to define another oracle $\mathcal{I}$ that inverts circuits with $\mathcal{R}$ gates. It is straight-forward to show that no non-uniform circuit family can learn F even given access to $\mathcal{I}$, but since $\mathcal{I}$ can invert all circuits with $\mathcal{R}$ gates, AIOWF do not exist. This type of proof technique is common in the cryptographic literature (see *e.g.* [HHRS07, HR04]) and rules out fully black-box reductions building AIOWF from hardness of learning. However, it does not rule out relativizing reductions, which allow the circuit *computing* the AIOWF to also use $\mathcal{I}$ gates: it is not at all obvious how or even if $\mathcal{I}$ can invert circuits that contain $\mathcal{I}$ gates.[3] This distinction is not merely cosmetic: in particular, the Ostrovsky-Wigderson theorem (Theorem 2.1) is *not* fully black-box but it is relativizing (see Appendix B for a discussion of this distinction). Therefore, in order to invoke it we must rule out relativizing reductions and

[3] Simon [Sim98] proposes a technique to overcome this problem that may be applicable in our setting. However, we present our result with our $\mathbf{PSPACE}_*^{\mathcal{R}}$ oracle, which we believe may be of independent interest.

not just fully black-box reductions. Doing so requires a more general oracle, which we describe now.

Definition 1.3. A language L is in $\mathbf{PSPACE}_*^{\mathcal{R}}$ if there exists a pair (M_1, M_2) where M_1 is a polynomial-time Turing machine and M_2 is a polynomial-space oracle Turing machine such that $x \in L$ if and only if $M_1(x)$ outputs $z_1, \ldots, z_m \in \{0,1\}^*$ and $M_2(x)$ using only oracle gates $\mathcal{R}_{z_1}, \ldots, \mathcal{R}_{z_m}$ outputs 1.

There is a natural complete language $\mathsf{QBF}_*^{\mathcal{R}}$ for this class, described in Section 2. Our separating oracle $\mathcal{O}$ decides $\mathsf{QBF}_*^{\mathcal{R}}$ where $\mathcal{R}$ is chosen from the same distribution as above. Learning is still hard relative to $\mathcal{O}$: even with access to $\mathcal{O}$, the learner can only "see" $\mathcal{R}_z$ for polynomially many z because M_2 in Definition 1.3 can only call $\mathcal{R}_{z_1}, \ldots, \mathcal{R}_{z_m}$ and in particular cannot enumerate over all exponentially many $\mathcal{R}_z$. Thus $\mathcal{O}$ does not help the learner find additional hard instances, and so the hard instances $\mathcal{R}_z$ that remain hidden also remain random-looking and therefore, since nothing can learn a random function, hard to learn.

On the other hand, we can use $\mathcal{O}$ to build an inverter that inverts any AIOWF. Given any f as a circuit with $\mathcal{O}$ gates, we show that it is possible to use $\mathcal{O}$ to find "heavy queries", *i.e.* z such that the computation of $f(x)$ queries $\mathcal{R}_z$ with probability $\geq 1/\mathrm{poly}(n)$ over the choice of random x. Notice this means there can be at most $\mathrm{poly}(n)$ many heavy z. We show that if f only ever queried $\mathcal{O}$ on either easy or heavy z, then one can efficiently invert f using oracle queries only for the $\mathrm{poly}(n)$ heavy instances. Of course f may actually query $\mathcal{O}$ on "bad z" that are hard and yet not heavy, but we show that on a *typical* $y = f(x)$ where x is chosen at random, the computation of $f(x)$ is unlikely to call $\mathcal{O}$ on any bad z. Therefore, the inverter that finds the heavy queries and then inverts pretending that f only calls $\mathcal{O}$ on good z succeeds with noticeable probability over random $y = f(x)$. Finally, applying the Ostrovsky-Wigderson theorem (Theorem 2.1) implies Theorem 1.1.

Proving Theorem 1.2: here we describe the intuition behind Theorem 1.2 for fully-black-box reductions. We use the same $\mathcal{R}$ as above and let $\mathcal{O}$ decide $\mathsf{QBF}_*^{\mathcal{R}}$.

The family $F = \{\mathcal{R}_z\}_{z \in \{0,1\}^*}$ is hard to learn for circuits for the same reason as before, so the hypothetical black-box reduction implies that $L \in \mathbf{ZK}^{\mathcal{O}}$. On the other hand, we also have as before that AIOWF do not exist relative to $\mathcal{O}$. Ong and Vadhan [OV07] showed (Theorem 2.3) that if $L \in \mathbf{ZK}$, then either there exists an AIOWF, or L reduces to "Statistical Difference" (SD) which is a complete problem for the class $\mathbf{SZK}$, and in fact their result relativizes. Since $L \in \mathbf{ZK}^{\mathcal{O}}$ and AIOWF do not exist relative to $\mathcal{O}$, we deduce from Theorem 2.3 that L reduces to $\mathsf{SD}^{\mathcal{O}}$ (where circuits can contain $\mathcal{O}$ gates).

Furthermore, because the construction is black-box, the simulator only uses oracle access to F, which is implementable using only access to $\mathcal{R}$, so the proof of Theorem 2.3 says this means L reduces to $\mathsf{SD}^{\mathcal{R}}$. Finally, we deduce that $L \in \mathbf{SZK}$: the zero knowledge property of $\mathbf{SZK}$ is statistical, so intuitively the computational hardness of learning $F = \{\mathcal{R}_z\}$ cannot help; furthermore, since $L \in \mathbf{SZK}^{\mathcal{R}}$ for random $\mathcal{R}$, the oracle $\mathcal{R}$ does not contain information about L itself. To use this intuition formally, we replace $\mathcal{R}$ gates in instances of $\mathsf{SD}^{\mathcal{R}}$ with a fake oracle that is distributed identically to $\mathcal{R}$ on inputs of length $O(\log n)$, and always responds 0 on longer inputs. Notice this can be done efficiently and so that the size of the resulting SD instance is only polynomially larger than the starting $\mathsf{SD}^{\mathcal{R}}$ instance. We show that this can be done in a way such that the resulting instance of SD is a YES (resp. NO) instance if and only if the starting $x \in L$ (resp. $x \notin L$), and therefore this gives a good randomized reduction to SD and puts $L \in \mathbf{SZK}$.

2 Preliminaries

For any distribution X, let $x \leftarrow_{\mathrm{R}} X$ denote a random variable sampled according to X. If S is a finite set, $x \leftarrow_{\mathrm{R}} S$ denotes a random variable sampled uniformly from S. U_n denotes the uniform distribution on $\{0,1\}^n$. For any function f and distribution X, we let $f(X)$ denote the distribution of outputs $f(x)$ given an input $x \leftarrow_{\mathrm{R}} X$. The statistical difference $\Delta(X,Y)$ of two distributions X, Y over a common universe U is defined as $\Delta(X,Y) = \frac{1}{2}\sum_{u \in U} |\Pr[X = u] - \Pr[Y = u]|$. We say that X, Y are computationally indistinguishable for non-uniform adversaries if for every family of polynomial-size circuits $\{C_n\}$, $|\Pr[C_n(X) = 1] - \Pr[C_n(Y) = 1]| \leq n^{-\omega(1)}$.

Let QBF denote the language of satisfiable quantified boolean formulas. It is well-known that QBF is **PSPACE**-complete (see *e.g.* [AB09]). For every oracle $\mathcal{R} = \{\mathcal{R}^{(n)}\}_{n \geq 1}$ where $\mathcal{R}^{(n)} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, let $\mathsf{QBF}^{\mathcal{R}}_*$ be the language of satisfiable QBF where the final propositional formula is allowed $\mathcal{R}_z = \mathcal{R}^{(n)}(z, \cdot)$ gates, but only for *fixed* z (for example, "$\exists z, \mathcal{R}_z(x)$" is not a valid formula for $\mathsf{QBF}^{\mathcal{R}}_*$). It follows immediately from the proof that QBF is complete for **PSPACE** that $\mathsf{QBF}^{\mathcal{R}}_*$ is complete for $\mathbf{PSPACE}^{\mathcal{R}}_*$ (defined previously in Definition 1.3).

PAC Learning: we say that a family of circuits $C = \{C_n\}$ learns a family of functions F (called a *concept class*) using membership queries with advantage ε if for every $f \in F$, $f : \{0,1\}^n \to \{0,1\}$, and every distribution X over $\{0,1\}^n$, given a set S of many labelled examples each drawn independently from the joint distribution $(X, f(X))$ and access to an oracle computing f, C_n produces a hypothesis circuit h that with probability $1 - 2^{-n}$ satisfies $\Pr[h(X) = f(X)] > \frac{1+\varepsilon}{2}$. We say PAC learning a concept class F is hard if no family of polynomial-size circuits can learn functions in F *on infinitely many input lengths* using membership queries with advantage $\varepsilon = 1/\mathrm{poly}(n)$. In this paper, we say PAC learning is hard if PAC learning functions computable by circuits of size n^2 on the uniform input distribution is hard. Learning relative to an oracle $\mathcal{O}$ means the concept classes, learning algorithms, and hypothesis are allowed $\mathcal{O}$ gates.

Auxiliary-input one-way functions: we say that AIOWF against uniform (resp. non-uniform) inverters exist if for every uniform (resp. non-uniform) inverter I, there exists an infinite collection W of functions where for every $f \in W$, $f : \{0,1\}^n \to \{0,1\}^m$, f is computable by a circuit of size s, and it holds that

$$\Pr_{x \leftarrow_{\mathrm{R}} U_n}[I(f, y) \in f^{-1}(y) \mid y = f(x)] < s^{-\omega(1)}$$

where $f^{-1}(y) = \{x \mid f(x) = y\}$. Note that f is given as input to I as a circuit, and the collection W may depend on I. The definition relativizes by allowing I and the circuits in W oracle gates.

Zero-knowledge: zero knowledge come in many varieties, depending on requirements such as round complexity, public vs. private coin, and composability criteria. In this work, we ignore these issues and work with a very broad definition of zero knowledge, called honest-verifier computational zero knowledge arguments **HV-CZKA** in the work of [OV07], which we denote simply by **ZK**. We let $\langle P, V\rangle(x)$ denote the transcript of an interactive protocol between a prover P and a verifier V on common input x. We say that $L \in \mathbf{ZK}$ if there exists an efficient (randomized) verifier strategy such that the following hold:

- Completeness: $\forall x \in L$, there is a prover strategy such that V accepts the transcript $\langle P, V\rangle(x)$ with probability $1 - 2^{-n}$.
- Soundness: $\forall x \notin L$, for any efficient prover strategy P^*, V accepts the transcript $\langle P^*, V\rangle(x)$ with probability at most 2^{-n}.

- Zero knowledge: there exists an efficient simulator S such that $\forall x \in L$, the distribution $\langle P, V\rangle(x)$ is computationally indistinguishable from $S(x)$.

Furthermore we say that L has a *honest-verifier statistical zero knowledge proof* (*i.e.* **HV-SZKP** in the terminology of [OV07], which we abbreviate as **SZK**) if the soundness condition holds with respect to all (possibly inefficient) prover strategies and the zero knowledge condition guarantees not only computational indistinguishability but also statistical indistinguishability, *i.e.* $\Delta(\langle P, V\rangle(x), S(x)) \leq n^{-\omega(1)}$. It is known that $\mathbf{SZK} \subseteq \mathbf{AM} \cap \mathbf{coAM}$ [For87, AH91]. We review some facts about **ZK** and **SZK**.

Theorem 2.1 ([Ost91, OW93]). $\mathbf{ZK} \neq \mathbf{BPP}$ *implies AIOWF against uniform inverters exist.*

Theorem 2.2 ([Vad04]). *The following promise problem (Statistical Difference, $\mathsf{SD}_{\alpha,\beta}$) is* **SZK**-*complete for any choice $0 < \beta < \alpha < 1$ satisfying $\beta < \alpha^2$. An input is a pair of circuits X_0, X_1 taking n-bit inputs, where we identify each circuit X_i with the distribution it samples, $X_i(U_n)$. A YES instance satisfies $\Delta(X_0, X_1) \geq \alpha$, while a NO instance satisfies $\Delta(X_0, X_1) \leq \beta$. We write simply* SD *when the particular choice of α, β is unimportant.*

Theorem 2.3 ([Vad04, OV07]). *If $L \in$* **ZK**, *then either there exists an efficient reduction* Red *from L to* SD, *or there exist AIOWF's against non-uniform inverters.*[4]

Since we will study black-box constructions of zero-knowledge protocols, we will work with relativized versions of **ZK**. We say $L \in \mathbf{ZK}^{\mathcal{O}}$ if it satisfies the definition of **ZK** as defined above except the prover, verifier, simulator, and distinguisher are all allowed access to the oracle $\mathcal{O}$. Also, $\mathsf{SD}^{\mathcal{O}}$ is like SD except circuits are allowed $\mathcal{O}$ gates. Examining the proofs of the above Theorem 2.1, Theorem 2.2, Theorem 2.3, we observe that they all relativize.[5]

3 Relativizing techniques

Our main result for relativizing techniques is to separate hardness of learning and AIOWF. Recall the oracle:

Definition 3.1. $\mathcal{O}$ is drawn from the following the distribution. First, for each n select a function $\mathcal{R}^{(n)} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ by letting each $z \in \{0,1\}^n$ be a *hard instance* with probability $2^{-n/2}$, where we set $\mathcal{R}_z = \mathcal{R}^{(n)}(z, \cdot)$ to be a random function, and letting z be an *easy instance* with probability $1 - 2^{-n/2}$, where $\mathcal{R}_z \equiv 0$. Let $\mathcal{O}$ decide $\mathsf{QBF}^{\mathcal{R}}_*$, which is $\mathbf{PSPACE}^{\mathcal{R}}_*$-complete.

Theorem 3.2. *With probability 1 over the choice of oracle $\mathcal{O}$ as in Definition 3.1, the concept class $F = \{\mathcal{R}_z\}_{z\in\{0,1\}^*}$ is hard to learn, but no AIOWF against uniform inverters exists.*

Proof. The theorem immediately follows from the following two lemmas, proved in the following subsections.

Lemma 3.3 (Learning is hard relative to $\mathcal{O}$). *With probability 1 over the choice of $\mathcal{O}$, F is hard to learn even when the learner has access to $\mathcal{O}$.*

[4] [OV07] actually proved there exist *instance-dependent* one-way functions, a stronger notion than AIOWF.

[5] Readers familiar with [OV07] will note that they also prove a converse to Theorem 2.3 via non-relativizing techniques. This does not affect us since we only use the direction stated in Theorem 2.3, which is relativizing.

Lemma 3.4 (AIOWF do not exist relative to $\mathcal{O}$)**.** *There is an efficient oracle algorithm I that, with probability 1 over choice of $\mathcal{O}$ as in Definition 3.1, given any function $f : \{0,1\}^n \to \{0,1\}^m$ described as a circuit of size s with $\mathcal{O}$ gates, satisfies:*

$$\Pr_{x \leftarrow_R \{0,1\}^n}[I^{\mathcal{O}}(f^{\mathcal{O}}, y) \in (f^{\mathcal{O}})^{-1}(y) \mid f^{\mathcal{O}}(x) = y] > 1/2$$

■

Combining the Ostrovsky-Wigerson theorem (Theorem 2.1), which relativizes, with Theorem 3.2 we obtain our main theorem about relativizing proofs Theorem 1.1.[6]

3.1 Learning is hard relative to $\mathcal{O}$

Proof of Lemma 3.3. To prove this lemma, we show that any oracle circuit $C^{\mathcal{O}}$ has probability $2^{-2^{\Omega(n)}}$ of learning $\mathcal{R}_z$ simultaneously on all z of length n. This proof follows from a case analysis: for a hard instance z, we first show that it is unlikely the hypothesis produced, $h^{\mathcal{O}}$, agrees with $\mathcal{R}_z$ without querying z (because if $h^{\mathcal{O}}$ cannot query $\mathcal{O}$ on $\mathcal{R}_z$ then $\mathcal{R}_z$ looks like a random function), and we then show that it is extremely unlikely that $C^{\mathcal{O}}$ can produce a hypothesis $h^{\mathcal{O}}$ that queries z with noticeable probability since the function $\mathcal{R}_z$ is random and therefore the labelled examples that $C^{\mathcal{O}}$ sees contain no information about z.

Fix n and any circuit C of size s. We prove the claim:

Claim 3.5. *For $\varepsilon = 2^{-\log^2 n}$.*

$$\Pr_{\mathcal{O}}\left[\bigwedge_{z \in \{0,1\}^n} C^{\mathcal{O}} \textit{ learns } \mathcal{R}_z \textit{ with advantage } \varepsilon\right] \leq 2^{-2^{\Omega(n)}}$$

This claim implies the lemma, since taking a union bound over all $2^{O(s\log(s))}$ circuits of size s for any $s = \mathrm{poly}(n)$ shows that the probability of there existing any circuit learning all the $\mathcal{R}_z$ is still $2^{-2^{\Omega(n)}}$. By the Borel-Cantelli lemma, this means that with probability 1, no family of circuits learns F on infinitely many input lengths.

Let $p = p(n) = \mathrm{poly}(n)$ be the number of labelled examples that C observes, and for $x_1, \ldots, x_p \leftarrow_{\mathrm{R}} U_n$, let $S_z = \{(x_1, \mathcal{R}_z(x_1)), \ldots, (x_p, \mathcal{R}_z(x_p))\}$. Let $C^{\mathcal{O}}(S_z)$ denote the hypothesis that $C^{\mathcal{O}}$ outputs given labelled examples S_z. We say that $C^{\mathcal{O}}(S_z)(x)$ queries $\mathcal{R}_z$ if either in constructing the hypothesis $C^{\mathcal{O}}(S_z)$ or in evaluating the hypothesis $C^{\mathcal{O}}(S_z)$ on x the oracle $\mathcal{O}$ is queried with a formula φ that contains a $\mathcal{R}_z$ gate. We will show that the probability $C^{\mathcal{O}}(S_z)$ approximates $\mathcal{R}_z$ is small if $C^{\mathcal{O}}(S_z)(x)$ rarely queries $\mathcal{R}_z$ because $\mathcal{R}_z$ is random. Then we will show that $C^{\mathcal{O}}(S_z)(x)$ rarely queries z because the labelled examples S_z and the membership oracle contain essentially no information about z itself.

Define

- A_z^{ε} as the event over the choice of $\mathcal{O}$ that $C^{\mathcal{O}}$ learns $\mathcal{R}_z$ with advantage ε
- $B_z^{\varepsilon^4}$ as the event over the choice of $\mathcal{O}$ that $\Pr_{S_z,x}[C^{\mathcal{O}}(S_z)(x) \text{ queries } \mathcal{R}_z)] > \varepsilon^4$

[6] Actually, this argument already rules out a more general class of proofs, namely so-called $\forall\exists$ semi-black-box reductions. We omit the details in the proceedings version.

We develop the LHS of Claim 3.5

$$\Pr_{\mathcal{O}}\left[\bigwedge_{z\in\{0,1\}^n} A_z^\varepsilon\right] \leq \Pr_{\mathcal{O}}\left[\bigwedge_{z \text{ hard}} A_z^\varepsilon\right] \tag{3.1}$$

$$\leq \Pr_{\mathcal{O}}\left[\bigwedge_{z \text{ hard}} (A_z^\varepsilon \vee B_z^{\varepsilon^4})\right] \tag{3.2}$$

$$\leq \Pr_{\mathcal{O}}\left[\exists z \text{ hard},\ A_z^\varepsilon \wedge \overline{B_z^{\varepsilon^4}}\right] + \Pr_{\mathcal{O}}\left[\bigwedge_{z \text{ hard}} B_z^{\varepsilon^4}\right] \tag{3.3}$$

This formalizes our above intuition, since the first term is the probability that, for some hard $z \in \{0,1\}^n$, $C^{\mathcal{O}}$ learns $\mathcal{R}(z,x)$ but rarely queries z, and the second term is the probability that $C^{\mathcal{O}}(S_z)(x)$ queries $\mathcal{R}_z$ with noticeable probability for every the hard z.

Bounding the first term of Inequality 3.3. Fix a hard z (of which there are at most 2^n). We want to bound the quantity

$$\Pr_{\mathcal{O}}[A_z^\varepsilon \wedge \overline{B_z^{\varepsilon^4}}] = \mathbb{E}_{\mathcal{R}'} \Pr_{\mathcal{O}}[A_z^\varepsilon \wedge \overline{B_z^{\varepsilon^4}} \mid \mathcal{R}'] \tag{3.4}$$

Here, $\mathcal{R}'$ is a fixing of the entire oracle $\mathcal{R}$ *except* for the function $\mathcal{R}_z$, which remains random, and $\mathcal{O}|\mathcal{R}'$ is the oracle constructed as in Definition 3.1 except with $\mathcal{R}'$ replacing $\mathcal{R}$. We will show $\Pr[A_z^\varepsilon \wedge \overline{B_z^{\varepsilon^4}} \mid \mathcal{R}'] \leq 2^{-2^{\Omega(n)}}$ for any fixing $\mathcal{R}'$.

If $C^{\mathcal{O}|\mathcal{R}'}(S_z)$ never queried $\mathcal{R}_z$, then the number of functions that $C^{\mathcal{O}|\mathcal{R}'}$ could possibly learn is bounded by the number of possible inputs (*i.e.* labelled examples) plus membership oracle responses for each input, which is at most $2^{p(n)(n+2)}$, which is a negligible fraction of the 2^{2^n} possible functions $\mathcal{R}_z$ could be.

Now consider $C^{\mathcal{O}|\mathcal{R}'}(S_z)$ that can query $\mathcal{R}_z$. Fix any $\mathcal{R}_z$ such that $A_z^\varepsilon \wedge \overline{B_z^{\varepsilon^4}}$ occurs. Since A_z^ε holds therefore

$$\Pr_{S_z}\left[\Pr_x[C^{\mathcal{O}|\mathcal{R}'}(S_z)(x) = \mathcal{R}_z(x)] > \tfrac{1+\varepsilon}{2}\right] > 1 - 2^{-n}$$

Since $\overline{B_z^{\varepsilon^4}}$ holds, by Markov it holds that

$$\Pr_{S_z}\left[\Pr_x[C^{\mathcal{O}|\mathcal{R}'}(S_z)(x) \text{ queries } \mathcal{R}_z] \geq 4\varepsilon^3\right] < \varepsilon/4$$

This implies that there must exist some fixed S' such that both events $\Pr_x[C^{\mathcal{O}|\mathcal{R}'}(S')(x) = \mathcal{R}_z(x)] > \frac{1+\varepsilon}{2}$ and $\Pr_x[C^{\mathcal{O}|\mathcal{R}'}(S')(x) \text{ queries } \mathcal{R}_z] < 4\varepsilon^3$ occur. Thus, the string S', the answers to all the membership queries on input S' (which consists of $p(n)$ bits and are independent of x), and an explicit labelling of all $4\varepsilon^3 2^n$ points x where the $C^{\mathcal{O}|\mathcal{R}'}(S')(x)$ queries $\mathcal{R}_z$ gives us a description of $\mathcal{R}_z$ that is accurate up to relative distance $\frac{1-\varepsilon}{2}$; call this the noisy description of $\mathcal{R}_z$. A Chernoff bound implies that the number of vectors of length 2^n of relative weight less than $\frac{1-\varepsilon}{2}$ is at most $2^{2^n - \Omega(\varepsilon^2 2^n)}$. Therefore, every function $\mathcal{R}_z$ that $C^{\mathcal{O}|\mathcal{R}'}$ is able to learn can be specified by first giving the noisy description of $\mathcal{R}_z$ and then giving the low-weight vector that equals the difference between the noisy description and the true function. This means that $C^{\mathcal{O}|\mathcal{R}'}$ can learn at most $2^{p(n)(n+2)+4\varepsilon^3(n+1)2^n+2^n-\Omega(\varepsilon^2 2^n)}$ different functions out of 2^{2^n} functions in total, and therefore

$$\begin{aligned}\Pr_{\mathcal{O}}[A_z^\varepsilon \wedge \overline{B_z^{\varepsilon^4}} \mid \mathcal{R}'] &\leq 2^{p(n)(n+2)+4\varepsilon^3(n+1)2^n+2^n-\Omega(\varepsilon^2 2^n)-2^n} \\ &= 2^{-2^{\Omega(n)}}\end{aligned}$$

where in the last line we use that $\varepsilon = 2^{-\log^2 n}$. Combined with Inequality 3.4 and a union bound over all hard instances z, this bounds the first term of Inequality 3.3.

Bounding the second term of Inequality 3.3. We will show that if the learner C can query $\mathcal{R}_z$ with noticeable probability given a random S, it can be used to "invert" $\mathcal{R}$ in the following sense: view $\mathcal{R}$ as a function $\{0,1\}^n \to \{0,1\}^{2^n}$ where each input z is mapped to the truth table of $\mathcal{R}_z$. We say that a (computationally unbounded) procedure $A^{\mathcal{R}}$ inverts $\mathcal{R}$ using q queries if for every non-zero y in the image of $\mathcal{R}$, we have $A^{\mathcal{R}}(y) = \mathcal{R}^{-1}(y)$ ($\mathcal{R}$ is almost surely injective, so we assume it to be without loss of generality) and A makes at most q queries to $\mathcal{R}$.

To apply this to our setting, we will show that if $C^{\mathcal{O}}(S_z)$ is able to query $\mathcal{R}_z$ with probability $\geq \varepsilon^4$ over S_z, then it can be used to build an inverter for $\mathcal{R}$ making only $O(p(n)n/\varepsilon^4)$ queries. Then we show that with high probability this is impossible.

First let us show that the event $\bigwedge_{z \text{ hard}} B_z^{\varepsilon^4}$ implies one can invert $\mathcal{R}$ using $O(p(n)n/\varepsilon^4)$ queries. We first describe a randomized procedure A' for inverting $\mathcal{R}$. A' is defined using the learning circuit C as follows: on every non-zero input $y \in \{0,1\}^{2^n}$ which is the truth table of some function, emulate C $O(n/\varepsilon^4)$ times using independent randomness, answering C's queries to the example oracle and membership oracle using y as the truth table of the hidden labelling function. To answer queries φ that C makes to $\mathcal{O}$, let Z be the set of z such that $\mathcal{R}_z$ appears in φ. For each $z \in Z$ of length n, A' will query $\mathcal{R}$ to get the truth table $\mathcal{R}_z$. Furthermore, A' checks whether $\mathcal{R}_z = y$, and if so it halts and outputs z. For every $z' \in Z$ where $|z'| = n' \neq n$, A' sets $\mathcal{R}_{z'}$ using independent coin tosses to be $0^{2^{n'}}$ with probability $1 - 2^{-n'/2}$ and to be a random function $\{0,1\}^{n'} \to \{0,1\}$ with probability $2^{-n'/2}$. Then A' decides the $\mathsf{QBF}_*^{\mathcal{R}}$ formula φ using these truth tables (A' can do this since it is unbounded). All these independent runs together query the oracle at most $O(p(n)n/\varepsilon^4)$ times. Because $B_z^{\varepsilon^4}$ holds for every z, *i.e.* for each z, when trying to learn $\mathcal{R}_z$ the circuit C queries $\mathcal{R}_z$ with probability at least ε^4, this means with probability $1-(1-\varepsilon^4)^{O(n/\varepsilon^4)} \geq 1 - 2^{-2n}$ at least one of the emulations will query $z = \mathcal{R}^{-1}(y)$, and so A' will find z. Now take a union bound over all possible non-zero inputs $y = \mathcal{R}_z$ of which there are at most 2^n, still with probability $1 - 2^{-n}$ the random bits used are simultaneously good for all y.

This means for any $\mathcal{R}$ where $\bigwedge_{z \text{ hard}} B_z^{\varepsilon^4}$ holds, A' inverts $\mathcal{R}$ with probability $1 - 2^{-n}$. This implies

$$\begin{aligned}\mathbb{E}_{\mathcal{R}|\bigwedge_{z \text{ hard}} B_z^{\varepsilon^4}} \; &\Pr[A' \text{ inverts } \mathcal{R} \text{ using } O(p(n)n/\varepsilon^4) \text{ queries}] \\ &\geq 1 - 2^{-n}\end{aligned}$$

By averaging, this means there *exists* a fixing of the random coins of A' (call A' with these fixed coins A) such that for a $1-2^{-n}$ fraction of the $\mathcal{R}$ where $\bigwedge_{z \text{ hard}} B_z^{\varepsilon^4}$ holds, A inverts $\mathcal{R}$. It therefore follows that

$$\begin{aligned}&\Pr_{\mathcal{O}}\left[\bigwedge_{z \text{ hard}} B_z^{\varepsilon^4}\right] \cdot (1 - 2^{-n}) \\ &\quad \leq \Pr_{\mathcal{R}}\left[A \text{ inverts } \mathcal{R} \text{ using } O(p(n)n/\varepsilon^4) \text{ queries}\right]\end{aligned}$$

The following lemma concludes the proof of the bound on the second term of Inequality 3.3.

Lemma 3.6. *For any* $A^{\mathcal{R}}$,
$\Pr_{\mathcal{R}}[A^{\mathcal{R}}$ *inverts* $\mathcal{R}$ *using* $O(p(n)n/\varepsilon^4)$ *queries*$] \leq 2^{-2^{\Omega(n)}}$

Proof. The proof is a straightforward generalization of Gennaro and Trevisan's proof [GT00] that a random permutation is hard to invert for circuits, generalized to the case where the function is not a permutation but is still injective. The idea is that given A, any function that A can invert can be "succinctly described", and therefore there cannot be too many of them.

Fix any oracle procedure $A^{\mathcal{R}}$ making at most $O(p(n)n/\varepsilon^4)$ to $\mathcal{R}$. Let $N = |\{x \mid \mathcal{R}(x) \leftarrow_{\mathrm{R}} U_{2^n}\}|$ denote the number of hard outputs of $\mathcal{R}$; by Chernoff the probability that $N \notin [2^{n/2-1}, 2^{n/2+1}]$ is bounded by $2^{-\Omega(2^{n/2})}$, so in the following we condition on this event not happening:

$$\Pr_{\mathcal{R}}[A \text{ inverts } \mathcal{R}] \leq 2^{-\Omega(2^{n/2})} \quad + \quad \mathbb{E}_{N \in [2^{n/2-1}, 2^{n/2+1}]} \left[\Pr_{\mathcal{R}}[A \text{ inverts } \mathcal{R} \mid N \text{ hard outputs}] \right]$$

We will further throw out the oracles $\mathcal{R}$ that are not injective (this occurs with probability at most $\leq \binom{N}{2} 2^{-2^n}$). We call $\mathcal{R}$ where neither of these conditions hold "good". Therefore our bound is now:

$$\Pr_{\mathcal{R}}[A \text{ inverts } \mathcal{R}] \leq 2^{-2^{\Omega(n)}} \quad + \quad \mathbb{E}_{N \in [2^{n/2-1}, 2^{n/2+1}]} \left[\Pr_{\mathcal{R} \text{ good}}[A \text{ inverts } \mathcal{R} \mid N \text{ hard outputs}] \right]$$

Notice that with this conditioning, $\mathcal{R}$ is uniform in the set of good $\mathcal{R}$.

To bound the probability on the RHS, we show that A is only capable of inverting very few functions. Here, we follow the argument of [GT00] proving that one-way permutations are hard against circuits.

We give a procedure for describing all possible injective functions $\mathcal{R}$ with N hard outputs as follows: we will keep track of a set $Y \subseteq \{0,1\}^{2n}$ of "easily describable outputs" y for which we will be able to compute the preimage $x = \mathcal{R}^{-1}(y)$ with very little information using A. For the "hard-to-describe outputs" outside Y we will just explicitly record the function. We show that this is sufficient for reconstructing any $\mathcal{R}$ that A is able to invert. We then prove that the number of functions describable this way is small compared to all possible functions, which gives us the desired bound.

For a fixed $\mathcal{R}$, define Y constructively as follows. Initialize $Y = \varnothing$ and the set $T \subseteq \{0,1\}^{2n}$ to be the image of the hard instances of $\mathcal{R}$, namely $t \in T$ iff $t = \mathcal{R}(z)$ for some hard instance z. Since we are conditioning on good $\mathcal{R}$, we have that initially $|T| = N$.

Repeatedly perform the following until T is empty: remove the lexicographically first element $t \in T$ and add it to Y. Execute $A^{\mathcal{R}}(t)$ and record the queries $x_1, \ldots, x_m$ (in the order that A makes them) that A makes to $\mathcal{R}$, where $m = O(p(n)n/\varepsilon^4)$. If none of the x_i satisfy $\mathcal{R}(x_i) = t$, then remove all of the $x_1, \ldots, x_m$ from T. If some x_i satisfies $\mathcal{R}(x_i) = y$, then remove $x_1, \ldots, x_{i-1}$ from T. Repeat by removing the next lexicographically first element of T, adding it to Y, etc.

Clearly we have that $|Y| \geq N/m$. We claim that given the set of hard instances $Z = \mathcal{R}^{-1}(T) \subseteq \{0,1\}^n$ (which is of size N), the set Y, $X = \mathcal{R}^{-1}(Y) \subseteq Z$ (the preimage of Y), and the explicit values of $\mathcal{R}$ on all inputs $x \in Z \setminus X$, we can completely reconstruct $\mathcal{R}$ as follows. For each $x \notin Z$, $\mathcal{R}(x) = 0^{2n}$. For each $x \in Z \setminus X$, output the explicitly recorded value. It only remains to match the elements of Y with their correct preimage in X. For each $y \in Y$ in lexicographic order, run $A^{\mathcal{R}}(y)$. The queries $A^{\mathcal{R}}(y)$ makes to $\mathcal{R}$ will all either be for $x \notin X$ in which case we know the answer explicitly, for $x \in X$ such that $\mathcal{R}(x)$ is lexicographically smaller than y and so we already computed the answer previously, or for some $x \in X$ we have not seen in a previous computation, which by construction must mean $x = \mathcal{R}^{-1}(y)$. Either way, we obtain the value $\mathcal{R}^{-1}(y)$.

The number of functions describable in this way is exactly

$$\binom{2^n}{N}\binom{N}{|Y|}\binom{2^{2^n}}{|Y|}\cdot\frac{(2^{2^n}-|Y|)!}{(2^{2^n}-N)!}$$

where the first factor is the number of ways of choosing N hard instances, the second is the choice of X, the third is the choice of Y, and the final is the number of ways of explicitly defining the function on $Z \setminus X$ assuming the function is injective. Therefore, the probability over $\mathcal{R}$ that A inverts $\mathcal{R}$ is exactly the above quantity divided by the total number of good $\mathcal{R}$, namely $\binom{2^n}{N}\frac{(2^{2^n})!}{(2^{2^n}-N)!}$. So we can calculate that:

$$\Pr_{\mathcal{R}\text{ injective}}[A \text{ inverts } \mathcal{R} \text{ everywhere} \mid N \text{ hard instances}] \leq \frac{\binom{2^n}{N}\binom{N}{|Y|}\binom{2^{2^n}}{|Y|}\cdot\frac{(2^{2^n}-|Y|)!}{(2^{2^n}-N)!}}{\binom{2^n}{N}\frac{(2^{2^n})!}{(2^{2^n}-N)!}} \tag{3.5}$$

$$= \frac{\binom{N}{|Y|}}{|Y|!} \tag{3.6}$$

$$\leq \left(\frac{N3e}{|Y|^2}\right)^{|Y|} \tag{3.7}$$

which is $2^{-2^{\Omega(n)}}$ for $N \leq 2^{n/2+1}$ and $|Y| > N/m = 2^{(1-o(1))n/2}$. ■

This concludes the proof of Lemma 3.3. ■

3.2 AIOWF do not exist relative to $\mathcal{O}$

Proof of Lemma 3.4. We would like to use the simple fact that for any $\mathcal{O}$, being able to compute $\mathbf{PSPACE}^{\mathcal{O}}$-complete problems implies being able to invert $\mathbf{PSPACE}^{\mathcal{O}}$ computations (*i.e.* polynomial-space computations with $\mathcal{O}$ gates, see for example Proposition A.2). This does not work for $\mathbf{PSPACE}_*^{\mathcal{R}}$ because we restricted the polynomial-space machine's access to $\mathcal{R}$: recall that a $\mathbf{PSPACE}_*^{\mathcal{R}}$ machine is not allowed to enumerate over $\mathcal{R}_z$ over all z.

To overcome this, the inverter I works as follows: it finds all the z such that $f^{\mathcal{O}}(x)$ queries $\mathcal{R}_z$ with noticeable probability over choice of random input x; call this set H the "heavy" queries. We show that by finding H, I knows most of the *hard instances* z such that $f^{\mathcal{O}}$ queries $\mathcal{R}_z$. We will show that this allows I to decide $\mathbf{PSPACE}^{\mathcal{R}'}$-complete problems, where $\mathcal{R}'_z = \mathcal{R}_z$ for all $z \in H$ and $\mathcal{R}'_{z'} \equiv 0$ for all $z' \notin H$. Note that, in contrast to $\mathbf{PSPACE}_*^{\mathcal{R}}$, a $\mathbf{PSPACE}^{\mathcal{R}'}$ *is allowed to enumerate over* $\mathcal{R}'_z$. Let $\mathcal{O}'$ decide a $\mathbf{PSPACE}^{\mathcal{R}'}$-complete language. We show that with knowledge of H and access to $\mathcal{O}$ the inverter I can efficiently compute $\mathcal{O}'$, and it follows (for example from Proposition A.2) that this allows I to invert any $\mathbf{PSPACE}^{\mathcal{R}'}$ computation.

Since $\mathbf{PSPACE}^{\mathcal{R}'}$ is very similar to $\mathbf{PSPACE}_*^{\mathcal{R}}$, we could try to use $\mathcal{O}'$ to invert $f^{\mathcal{O}}$, but the computation of $f^{\mathcal{O}}(x)$ may query hard instances outside H, and so $f^{\mathcal{O}}(x) \neq f^{\mathcal{O}'}(x)$ for some x. However, we argue that, by the definition of heavy and because hard instances are scattered at random, the probability over a *random* x that the computation $f^{\mathcal{O}}(x)$ queries a hard instance outside H cannot be too high. Therefore, the distributions $(x, f^{\mathcal{O}}(x))$ and $(x, f^{\mathcal{O}'}(x))$ for $x \leftarrow_{\mathrm{R}} U_n$ are statistically close. If I simulates $\mathcal{O}'$ and uses this to invert y by finding some x such that $y = f^{\mathcal{O}'}(x)$, then with high probability over random y the x also satisfies $f^{\mathcal{O}}(x) = y$

We proceed with the formal argument. We describe and analyze an algorithm I that with probability 2^{-s} over the choice of oracle, inverts all f computable by a circuit of size s. This proves the lemma, since by the Borel-Cantelli lemma this means $I^{\mathcal{O}}$ inverts all except finitely many circuits with probability 1 over $\mathcal{O}$.

Let f be any function computable by a circuit C (with $\mathcal{O}$ gates) of size s, where f takes inputs of length n. Let $g_1, \ldots, g_s$ be the oracle gates of C in topologically sorted order.

Set the heaviness threshold to be $\alpha = 100s^8$. In sorted order, I finds all z such that $C^{\mathcal{O}}(U_n)$ queries $\mathcal{O}$ with a formula containing a $\mathcal{R}_z$ gate with probability larger than $1/\alpha$ using the following procedure.

First, I initializes the set $Z_0 = \{z \mid |z| \leq 8 \log s\}$. Then, to construct Z_i, the set of heavy queries up till the i'th query, using Z_{i-1}, I does the following. Let the circuit Q'_i be the sub-circuit of C that computes queries for g_i. We transform Q'_i into a related circuit Q_i by replacing each oracle gate g_j, $j < i$ that appears in Q'_i (these are the only oracle gates that g_i depends on since we work in sorted order) with the following: on input φ, replace each $\mathcal{R}_z$ gate inside φ where $z \notin Z_j$ by a constant 0 gate, and then call $\mathcal{O}$ with this modified formula. This transformation forces all the hard instances that g_j queries to be in Z_j.

Note that $Q_i(x) = \varphi$ is exactly saying that $C(x)$ queries φ at g_i, conditioned on each previous oracle gate g_j querying only heavy instances ($z \in Z_j$) or easy instances ($\mathcal{R}_z \equiv 0$). Since Q_i only makes oracle queries containing $\mathcal{R}_z$ gates for $z \in Z_{i-1}$, this means Q_i is computable using only a $\mathbf{PSPACE}^{\mathcal{R}'_{i-1}}$ oracle, where $\mathcal{R}'_{i-1}(z,x) = \mathcal{R}(z,x)$ for $z \in Z_{i-1}$ and is zero otherwise. Since I knows Z_{i-1}, it can simulate a $\mathbf{PSPACE}^{\mathcal{R}'_{i-1}}$ oracle: any L decidable by a $\mathbf{PSPACE}^{\mathcal{R}'_{i-1}}$ machine M can be decided by a $\mathbf{PSPACE}_*^{\mathcal{R}}$ pair (M_1, M_2) where M_1 outputs Z_{i-1} and M_2 emulates M. A **PSPACE** oracle is able to compute the probabilities in the output distribution of any **PSPACE** computations, and this relativizes (Proposition A.3). Namely, since Q_i is computable in $\mathbf{PSPACE}^{\mathcal{R}'_{i-1}}$ and I can simulate a $\mathbf{PSPACE}^{\mathcal{R}'_{i-1}}$ oracle, I can run the algorithm of Proposition A.3 on input $(Q_i, D, 1^\alpha)$, where D is the efficient circuit taking a formula φ and $z \in \{0,1\}^*$ and outputting 1 if φ contains a $\mathcal{R}_z$ gate, and outputs 0 otherwise. This outputs $\{z \mid \Pr[Q_i(x) = \varphi \wedge D(\varphi, z) = 1] > 1/\alpha\}$, which we add to Z_{i-1} to obtain Z_i.

Proposition A.3 guarantees Z_s is the collection of all z such that there exists some Q_i querying z with probability $> 1/\alpha$ over the choice of random input x.

We now show that with high probability over $\mathcal{O}$, if I knows Z_s then it knows most of the hard instances that $f^{\mathcal{O}}$ might have queried, and so it can invert $f^{\mathcal{O}}$ almost everywhere. Formally, let $B(x)$ be the bad event that $f^{\mathcal{O}}(x)$ queries some hard z outside Z_s. We claim:

$$\Pr_{\mathcal{R}}\left[\Pr_{x \leftarrow_{\mathrm{R}} U_n}[B(x)] > \tfrac{1}{s}\right] \leq 2^{-s^2} \tag{3.8}$$

First we use this inequality to prove the lemma: by a union bound over all f computable by size s circuits, of which there are at most $2^{O(s \log s)}$, this means that for a $1 - 2^{-s^2 - O(s \log s)} \geq 1 - 2^{-s}$ fraction of the $\mathcal{R}$ that with probability $1 - 1/s$ over x, $f^{\mathcal{O}}(x)$ never queries hard $z \notin Z_s$. That is, for such good x and $\mathcal{R}$, $f^{\mathcal{O}}(x) = f^{\mathcal{O}'}(x)$, where $\mathcal{O}'$ decides a $\mathbf{PSPACE}^{\mathcal{R}'_s}$-complete problem, and $\mathcal{R}'_s$ is defined as before. This implies $\Delta\left((x, f^{\mathcal{O}}(x)), (x, f^{\mathcal{O}'}(x))\right) \leq 1/s$. Furthermore, a **PSPACE** oracle can invert **PSPACE** computations and this relativizes (Proposition A.2). I knows Z_s so it can use Z_s and $\mathcal{O}$ to simulate $\mathcal{O}'$, so it can apply the algorithm of Proposition A.2 to compute uniformly random preimages of $f^{\mathcal{O}'}$ with failure probability 2^{-m}, giving us

$$\Delta\left((x, f^{\mathcal{O}'}(x)), \quad (I^{\mathcal{O}}(y), y \mid y = f^{\mathcal{O}'}(x))\right)$$
$$\leq 2^{-m}$$

Putting these together by the triangle inequality, we have

$$\Delta((x, f^{\mathcal{O}}(x)), (I^{\mathcal{O}}(y), y \mid y = f^{\mathcal{O}}(x))) \leq 2/s + 2^{-m}$$

which proves the lemma modulo Inequality 3.8. In fact, we prove something much better: $I^{\mathcal{O}}$ actually gives an almost uniformly random preimage of y.

It remains to prove Inequality 3.8. Define inductively $B_i(x)$ as the event that $f^{\mathcal{O}}(x)$ queries a hard $z \notin Z_i$ in the i'th query but all prior queries j are either easy or in Z_j. Since $Z_i \subseteq Z_{i+1}$, we have that $B(x) \subseteq \bigcup_{i=1}^{s} B_i(x)$. By averaging:

$$\begin{aligned}
\Pr_{\mathcal{R}}\left[\Pr_x[B(x)] > \tfrac{1}{s}\right] &\leq \Pr_{\mathcal{R}}\left[\Pr_x\left[\bigcup_{i=1}^{s} B_i(x)\right] > \tfrac{1}{s}\right] \\
&\leq \Pr_{\mathcal{R}}\left[\exists i,\ \Pr_x[B_i(x)] > \tfrac{1}{s^2}\right] \\
&\leq \sum_{i=1}^{s} \Pr_{\mathcal{R}}\left[\Pr_x[B_i(x)] > \tfrac{1}{s^2}\right]
\end{aligned}$$

We claim that for each i, $\Pr_{\mathcal{R}}[\Pr_x[B_i(x)] > 1/s^2] \leq 2^{-2s^2}$, which we prove using a case analysis. Showing this concludes the proof of the lemma since $s2^{-2s^2} \leq 2^{-s^2}$.

The case analysis roughly goes as follows: either the probability that $f^{\mathcal{O}}$ makes a light i'th query (*i.e.* a query not in Z_i) is small, in which case the probability it makes a light and hard query is also small, or the probability that $f^{\mathcal{O}}$ makes a light i'th query is large, in which case the conditional probability of *each individual light query* is not too large, and in this case we can show that it is unlikely over the choice of oracle that many light queries are hard.

Formally, let $\mathsf{Light}_i(x)$ be the event that $f^{\mathcal{O}}$'s i'th query is light, *i.e.* it is not in Z_i conditioned on all queries $j < i$ being either in Z_j or easy. (The only difference between Light_i and B_i is that in B_i we also demand the i'th query be hard.) We have that

$$\begin{aligned}
&\Pr_{\mathcal{R}}[\Pr_x[B_i(x)] > 1/s^2] \\
&\quad = \Pr_{\mathcal{R}}\left[\left\{\Pr_x[B_i(x)] > 1/s^2\right\} \wedge \left\{\Pr_x[\mathsf{Light}_i(x)] \geq 1/s^2\right\}\right] \\
&\quad\quad + \Pr_{\mathcal{R}}\left[\left\{\Pr_x[B_i(x)] > 1/s^2\right\} \wedge \left\{\Pr_x[\mathsf{Light}_i(x)] < 1/s^2\right\}\right]
\end{aligned}$$

Clearly the second term is 0 because $B_i(x) \subseteq \mathsf{Light}_i(x)$.

To bound the first term, we inductively fix $\mathcal{R}$ up until the i'th query as follows: let $\mathcal{R}_0$ be a fixing of all $\mathcal{R}_z$ with $z \in Z_0$. Let Z_i be the set of heavy i'th queries conditioned on $Z_{i-1}, \mathcal{R}_{i-1}$ and the event that $f^{\mathcal{O}}(x)$'s first $i-1$ queries are either easy or in Z_{i-1}, and let $\mathcal{R}_i$ be a fixing of all $\mathcal{R}_z$

for $z \in Z_i$ conditioned on $\mathcal{R}_{i-1}$. Thus, we can write:

$$\begin{aligned}
&\Pr_{\mathcal{R}}\left[\left\{\Pr_x[B_i(x)] > 1/s^2\right\} \wedge \left\{\Pr_x[\mathsf{Light}_i(x)] \geq 1/s^2\right\}\right] \\
&= \mathbb{E}_{\mathcal{R}_{i-1}} \Pr_{\mathcal{R}}\left[\left\{\Pr_x[B_i(x)] > 1/s^2\right\}\right. \\
&\qquad \left. \wedge \left\{\Pr_x[\mathsf{Light}_i(x)] \geq 1/s^2\right\} \mid \mathcal{R}_{i-1}\right] \\
&\leq \mathbb{E}_{\mathcal{R}_{i-1}} \Pr_{\mathcal{R}}\left[\left\{\Pr_x[B_i(x) \mid \mathsf{Light}_i(x)] > 1/s^2\right\}\right. \\
&\qquad \left. \mid \left\{\Pr_x[\mathsf{Light}_i(x)] \geq 1/s^2\right\} \wedge \mathcal{R}_{i-1}\right]
\end{aligned}$$

where in the last line we used the fact that $B_i(x) \subseteq \mathsf{Light}_i(x)$. For each such fixing of $\mathcal{R}_{i-1}$, since the probability that the i'th query is light is at least $1/s^2$, the probability that a specific light z is asked as the i'th query conditioned on $\mathsf{Light}_i(x)$ is at most $1/s^2 \cdot 1/\alpha = 1/(100s^6)$. Each i'th query is hard independently with probability at most $1/s^4$ over the choice of oracle (because Z_0 contains all queries of length up to $8 \log s$, the oracle is random only on longer inputs). If each light query were asked with probability *exactly* $1/(100s^6)$ then we could apply a Chernoff bound, which says that the probability that more than $1/s^2$ of the light queries are hard given that each light query is hard with probability $1/s^4$ is at most $2^{-100s^6/(4s^4))} \leq 2^{-2s^2}$. By a simple generalization of the Chernoff bound stated in Lemma A.4, we can show that the same bound holds even though we are guaranteed that each light query is asked with probability *at most* $1/(100s^6)$, so this concludes the proof of the lemma. ■

4 Black-box techniques

We first prove the result for fully-black-box reductions, then explain how to extend the proof to semi-black-box reductions.

Theorem 1.2 (Restated)**.** *If there exists a semi-black-box proof that constructs a* **ZK** *protocol for a language L assuming PAC learning is hard, then in fact $L \in$* **SZK***.*

Proof of fully-black-box case. A fully-black-box proof is relativizing, so both the construction and analysis must hold relative to any oracle. We will use the same oracle from Definition 3.1.

Recall that Lemma 3.3 says with probability 1 over the choice of $\mathcal{R}$, $F = \{\mathcal{R}_z\}_{z \in \{0,1\}^*}$ is hard to learn. By our hypothesis, this implies $L \in \mathbf{ZK}^{\mathcal{O}}$, and furthermore in the zero-knowledge protocol for L, the prover, verifier, and simulator all use access only to the hard concept class F, which can be implemented using just $\mathcal{R}$ (and not $\mathcal{O}$) gates.

Next, we claim that the protocol is in fact *statistically* zero knowledge. Applying the relativized version of the SZK/AIOWF characterization (Theorem 2.3), we know that if $L \in \mathbf{ZK}^{\mathcal{O}}$ then (a) there is an efficient reduction Red reducing L to $\mathsf{SD}^{\mathcal{O}}$, or (b) there exists AIOWF against non-uniform inverters relative to $\mathcal{O}$. Case (b) never occurs because Lemma 3.4 tells us that AIOWF do not exist relative to $\mathcal{O}$, so we must be in case (a).

In fact, the proof of Theorem 2.3 actually proves not only that Red reduces L to $\mathsf{SD}^{\mathcal{O}}$ but the circuits that Red produce are defined simply in terms of the (code of the) simulator of the original $\mathbf{ZK}^{\mathcal{O}}$ protocol. Because the simulator of the original protocol needed access only to $\mathcal{R}$, we can conclude that with probability 1 over the choice of $\mathcal{R}$, Red reduces every $x \in L$ to a YES instance of $\mathsf{SD}^{\mathcal{R}}$ and every $x \notin L$ to a NO instance of $\mathsf{SD}^{\mathcal{R}}$. (This observation is crucial: proving L reduces to $\mathsf{SD}^{\mathcal{O}}$ is meaningless, since $\mathcal{O}$ can decide **PSPACE**-complete languages. In fact, this is why one cannot use Theorem 2.1 to conclude that the non-existence of AIOWF implies $L \in \mathbf{BPP}$; applying it would only show $L \in \mathbf{BPP}^{\mathcal{O}}$, which is meaningless for the same reason.)

We can now deduce that with high probability over $\mathcal{R}$, the reduction Red is good for all long enough instances. Let us say that "Red succeeds on L_n" if for all x of length n, $\mathsf{Red}(x)$ maps each $x \in L$ to a YES instance of $\mathsf{SD}^{\mathcal{R}}$ and each $x \notin L$ reduction to a NO instance of $\mathsf{SD}^{\mathcal{R}}$ (*i.e.* they satisfy the promise of $\mathsf{SD}^{\mathcal{R}}$).

Claim 4.1. *If Red reduces L to $\mathsf{SD}^{\mathcal{R}}$ with probability 1 over $\mathcal{R}$, then $\Pr_{\mathcal{R}}[\mathsf{Red}$ succeeds on $L_n] \to 1$ as $n \to \infty$.*

To prove the claim, let A_n be the event that Red succeeds on $L_{\geq n}$, *i.e.* Red succeeds on all inputs of length at least n (rather than exactly n). Notice that it suffices to show $\Pr[A_n] \to 1$ as $n \to \infty$. We know by hypothesis that $1 = \Pr_{\mathcal{R}}[\mathsf{Red}$ reduces L to $\mathsf{SD}^{\mathcal{R}}] \leq \Pr_{\mathcal{R}}[\bigcup_{i=1}^{\infty} A_i]$. Since $A_n \subseteq A_{n+1}$, we have that:

$$\Pr\left[\bigcup_{i=1}^{\infty} A_i\right] = \sum_{i=1}^{\infty} \Pr[A_i \wedge \overline{A_{i-1}}]$$

But since $\Pr[A_n] = \sum_{i=1}^{n} \Pr[A_i \wedge \overline{A_{i-1}}]$, the claim follows.

Lemma 4.2 (Removing the oracle). *If for sufficiently large n, $\Pr_{\mathcal{R}}[\mathsf{Red}$ succeeds on $L_n] > 99/100$, then $L \in \mathbf{SZK}$.*

Claim 4.1 means that the hypothesis of this lemma is satisfied, and so the lemma implies the theorem. ∎

4.1 Removing the oracle

Proof of Lemma 4.2. Red efficiently maps each input x to an instance $(X_0^{\mathcal{R}}, X_1^{\mathcal{R}})$ of Statistical Difference with $\mathcal{R}$ gates, say with $\alpha = 99/100$ and $\beta = 1/100$. Namely, Red maps L to $\mathsf{SD}^{\mathcal{R}}_{99/100,1/100}$. By padding, we can assume without loss of generality that the input and output length of $X_i^{\mathcal{R}}$ is n, and $|X_i| \leq p(n) = \mathrm{poly}(n)$ for $i = 0, 1$.

By hypothesis, with probability $99/100$ over the choice of $\mathcal{R}$, for every x of length n, $x \in L$ reduces to $(X_0^{\mathcal{R}}, X_1^{\mathcal{R}})$ such that $\Delta(X_0^{\mathcal{R}}, X_1^{\mathcal{R}}) > 99/100$ while $x \notin L$ reduces to $(X_0^{\mathcal{R}}, X_1^{\mathcal{R}})$ such that $\Delta(X_0^{\mathcal{R}}, X_1^{\mathcal{R}}) < 1/100$.

Claim 4.3. *There is an efficient deterministic reduction Red' such that for all $x \in L$, $\mathsf{Red}'(x) = (X'_0, X'_1)$ satisfies $\Delta(X'_0, X'_1) > 24/25$ and for all $x \notin L$, $\mathsf{Red}'(x) = (X'_0, X'_1)$ satisfies $\Delta(X'_0, X'_1) < 1/25$.*

Since $(24/25)^2 > 1/25$, this is still in **SZK** and so the claim shows that Red' puts $L \in \mathbf{SZK}$.

To prove the claim, let Red' work by first running Red to produce $(X_0^{\mathcal{R}}, X_1^{\mathcal{R}})$, and then transforming those circuits the following way. Let Q be a circuit that takes some random bits and generates a "fake" oracle $\mathcal{R}_Q$ whose distribution on inputs of up to length $2\log 10^8 p(n)$ is identical to the real distribution $\mathcal{R}$, and for inputs longer than $2\log 10^8 p$ always returns 0. It is clear $\mathcal{R}_Q$ can be described and evaluated in polynomial time, and there is a circuit Q of size $\mathrm{poly}(p)$ that constructs $\mathcal{R}_Q$ using $m = \mathrm{poly}(p)$ random bits.

$\mathsf{Red}'(x) = (X'_0, X'_1)$ where X'_0 is the circuit that takes $m + n$ random bits and uses m bits (call these m bits ω) for Q to generate a fake random oracle $\mathcal{R}_Q$, and uses n bits to sample a random $x \leftarrow_{\mathrm{R}} X_0^{\mathcal{R}_Q}$ and then outputs (ω, x). X'_1 is the circuit that takes $m + n$ random bits just as above except it outputs (ω, x) where $x \leftarrow_{\mathrm{R}} X_1^{\mathcal{R}_Q}$.

We prove that Red' satisfies the claim. Let X be either X_0 or X_1 (the same argument applies to both). For $r \in \{0,1\}^n$, let $B(r)$ be the bad event over the choice of $\mathcal{R}$ that $X^{\mathcal{R}}(r)$ queries a hard instance z of length $> 2 \log 10^8 p$, and $B_i(r)$ be the event that the i'th oracle query of $X^{\mathcal{R}}(r)$ (for some arbitrary ordering of the queries) is a hard instance z of length $> 2 \log 10^8 p$. It holds that:

$$\Pr_{\mathcal{R}, r \leftarrow_{\mathrm{R}} U_n}[B(r)] = \mathbb{E}_r \Pr_{\mathcal{R}}[B(r)] \leq \mathbb{E}_r \sum_{i=1}^{p} \Pr_{\mathcal{R}}[B_i(r)] \leq 1/10^8$$

since over the randomness of $\mathcal{R}$, the probability that any query of length $> 2 \log 10^8 p$ is hard is at most $1/(10^8 p)$.

Now by Markov, we have that

$$\Pr_{\mathcal{R}}[\Pr_{r \leftarrow_{\mathrm{R}} U_n}[B(r)] > 1/10^4] < 1/10^4$$

Notice that for good $\mathcal{R}$ where $B(r)$ occurs with probability $\leq 1/10^4$, we have that $\Delta(X^{\mathcal{R}}, X^{\mathcal{R}_Q}) \leq 1/10^4$. Therefore, with probability $> 99/100 - 2/10^4$ we get a good fixing of ω used by Q to generate $\mathcal{R}_Q$, where by good we mean that

$$\begin{aligned} x \in L &\Rightarrow \Delta(X_0^{\mathcal{R}_Q}, X_1^{\mathcal{R}_Q}) > 99/100 - 2/10^4 \\ x \notin L &\Rightarrow \Delta(X_0^{\mathcal{R}_Q}, X_1^{\mathcal{R}_Q}) < 1/100 + 2/10^4 \end{aligned}$$

Therefore, the claim follows by averaging over all ω and using the fact that a $\frac{99}{100} - \frac{2}{10^4}$ fraction of the ω are good, so that

$$\begin{aligned} x \in L &\Rightarrow \Delta(X'_0, X'_1) > (\tfrac{99}{100} - \tfrac{2}{10^4})(\tfrac{99}{100} - \tfrac{2}{10^4}) > \tfrac{24}{25} \\ x \notin L &\Rightarrow \Delta(X'_0, X'_1) < \tfrac{1}{100} + \tfrac{2}{10^4} + \tfrac{1}{100} + \tfrac{2}{10^4} < \tfrac{1}{25} \end{aligned}$$

■

4.2 Semi-black-box reductions

The proof above fails to rule out semi-black-box reductions because we use Lemma 3.4, which says any efficiently computable function can be inverted by an adversary with access to $\mathcal{O}$. In contrast, in a semi-black-box reduction the adversary is allowed access *only to the hard concept class*, which in the above proof is $F = \{\mathcal{R}_z\}$. To rule out semi-black-box reductions we will "embed" $\mathbf{PSPACE}_*^{\mathcal{R}}$ inside F itself (an idea of Simon [Sim98], see also [RTV04]), but this must be done carefully. We have to balance two requirements: first, there must still be a way to call $\mathbf{PSPACE}_*^{\mathcal{R}}$ in order to invert all AIOWF. On the other hand, the verifier in the zero knowledge protocol *must not* be able to call the $\mathbf{PSPACE}_*^{\mathcal{R}}$ oracle, or else it could decide **PSPACE** on its own and all of **PSPACE** would trivially have a zero knowledge protocol in this relativized world. The key to achieve these two conflicting goals simultaneously is that Theorem 2.3 allows the inverter for the AIOWF to be *non-uniform*, while the verifier in the protocol is uniform.

Definition 4.4. Let $\mathcal{R}^{(n)} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ be chosen as follows: for each $z \in \{0,1\}^n$, with probability $2^{-n/2}$ let $\mathcal{R}_z$ be a function drawn from the distribution R described below (hard instances) and with probability $1 - 2^{-n/2}$ let $\mathcal{R}_z \equiv 0$ (easy instances).

The distribution R over functions $\{0,1\}^n \to \{0,1\}$ is defined as follows: on input $x \in \{0,1\}^n$, if the first $n - \sqrt{n}$ bits of x are not identically zero then output a random bit. If the first $n - \sqrt{n}$ bits of x are all 0 then let φ be the remaining $\sqrt{n}$ bits of x and interpret φ as a $\mathsf{QBF}_*^{\mathcal{R}}$ formula, and output whether φ is satisfiable.

First we check that $\mathcal{R}$ is well-defined. Namely, what if one queries $\mathcal{R}_z(0^{n-\sqrt{n}}\varphi)$ where z is a hard instance and φ is a $\mathsf{QBF}_*^{\mathcal{R}}$ formula that calls $\mathcal{R}_z$? This cannot happen: because $|z| = n$ and $|\varphi| = \sqrt{n}$, there can be no self-reference, *i.e.* φ can never have $\mathcal{R}_z$ gates because it cannot even describe z. Since φ does not call $\mathcal{R}_z$ then the oracle is well-defined as all the oracle calls made in all possible φ of length $\sqrt{n}$ are independent of $\mathcal{R}_z$'s responses.

Proof of Theorem 1.2, semi-black-box case. We use the oracle $\mathcal{R}$ of Definition 4.4 to prove the theorem.

Claim 4.5. *With probability* 1 *over* $\mathcal{R}$*, learning* F *is hard for circuits relative to* $\mathcal{R}$*.*

As before, it suffices to show that for any efficient circuit C we have $\Pr_{\mathcal{R}}\left[C^{\mathcal{R}} \text{ learns } F \text{ on length } n\right] \leq 2^{-2^{\Omega(n)}}$. We use the same notation as the proof of Lemma 3.3. Let A_z^{ε} denote the event that $C^{\mathcal{R}}$ learns $\mathcal{R}_z$ with advantage ε, and let $B_z^{\varepsilon^4}$ be the event that $\Pr_{S_z,x}[C^{\mathcal{R}}(S_z)(x) \text{ queries } \mathcal{R}_z] > \varepsilon^4$. Here, "$C^{\mathcal{R}}(S_z)(x)$ queries $\mathcal{R}_z$" means either during the construction of the hypothesis $C^{\mathcal{R}}(S_z)$ or while evaluating the hypothesis on x, the oracle is queried on $\mathcal{R}_z$ or on $\mathcal{R}_{z'}(0^{n'-\sqrt{n'}}\varphi)$ where $|z'| = n' > |z|^2$ and φ is a $\mathsf{QBF}_*^{\mathcal{R}}$ formula containing a $\mathcal{R}_z$ gate.

We have as before that

$$\Pr_{\mathcal{R}}[C^{\mathcal{R}} \text{ learns } F \text{ on length } n] \tag{4.1}$$

$$\leq \Pr_{\mathcal{R}}\left[\exists z \text{ hard of length } n,\ A_z^{\varepsilon} \wedge \overline{B_z^{\varepsilon^4}}\right] \tag{4.2}$$

$$+ \Pr_{\mathcal{R}}\left[\bigwedge_{z \text{hard}} B_z^{\varepsilon^4}\right] \tag{4.3}$$

To bound the term in Inequality 4.2, fix a hard instance z and let $\mathcal{R}'$ denote a fixing of the entire oracle $\mathcal{R}$ except for $\mathcal{R}_z$ and all $\mathcal{R}_{z'}(0^{n'-\sqrt{n'}}\varphi)$ where $|z'| = n' > |z|^2$ and φ contains a $\mathcal{R}_z$ gate. With such a fixing, $C^{\mathcal{R}'}$ can be viewed as a deterministic procedure for learning $\mathcal{R}_z$, which is a random function, except on inputs of the form $x = 0^{n-\sqrt{n}}\varphi$. But the probability that $C^{\mathcal{R}'}$ will be asked to label such a x is $2^{-n+\sqrt{n}}$, which means such x contribute a negligible to $C^{\mathcal{R}'}$'s advantage. Therefore, we can apply the proof bounding the first term of Inequality 3.3 to bound the first term by $2^{-2^{\Omega(n)}}$.

To bound the term in Inequality 4.3, we can show as in the proof of Lemma 3.3 that any $C^{\mathcal{R}}$ of size $p(n)$ that queries z with greater than ε^4 can be transformed into a procedure A that inverts $\mathcal{R}$ (in the sense of Lemma 3.6) making $O(p(n)n/\varepsilon^4)$ queries. We omit this transformation, which is identical to the previous one, except to point out that a non-zero y of length 2^n in the image of $\mathcal{R}$ contains all the truth tables of $\mathcal{R}_{z'}$ for $|z'| \leq \sqrt{n}$. If $C^{\mathcal{R}}$ queries anything depending on such $\mathcal{R}_{z'}$, A can answer consistently with y without making additional queries to $\mathcal{R}$. It suffices to show that with overwhelming probability, no procedure can invert $\mathcal{R}$ with $O(p(n)n/\varepsilon^4)$ queries, *i.e.* the analogous result of Lemma 3.6, which we omit since it is almost identical to the proof of Lemma 3.6.

Claim 4.5 and the hypothetical semi-black-box reduction implies that $L \in \mathbf{ZK}^{\mathcal{R}}$. Second, we show that AIOWF do not exist relative to $\mathcal{R}$.

Claim 4.6. *With probability* 1 *over* $\mathcal{R}$, *there exist no AIOWF against non-uniform adversaries relative to* $\mathcal{R}$.

We exhibit a family of polynomial-size oracle circuits $\{I_s\}$ that inverts every function $f : \{0,1\}^n \to \{0,1\}^m$ computable by an oracle circuit of size s. Namely, for every such f

$$\Pr_{x \leftarrow_{\mathrm{R}} U_n}[I_s^{\mathcal{R}}(f, y) \in (f^{\mathcal{R}})^{-1}(y) \mid y = f^{\mathcal{R}}(x)] > 1/2$$

In fact, the circuits I_s do exactly the same thing as the uniform inverter I in the proof of Lemma 3.4 except that in order to get access to a $\mathbf{PSPACE}_*^{\mathcal{R}}$ oracle, we hardwire into I_s a hard instance z' of length $n' = O(s^2)$. Using z', C gets access to $\mathcal{R}_{z'}$, and it can use $\mathcal{R}_{z'}(0^{n'-\sqrt{n'}}\varphi)$ to decide $\mathsf{QBF}_*^{\mathcal{R}}$ instances φ of size up to s. This is sufficient for I_s to implement the strategy of I from the proof of Lemma 3.4 to invert every f computable by an s size circuit, and the claim follows.

Using Theorem 2.3 and the fact that AIOWF against non-uniform inverters do not exist, we deduce that there is an efficient reduction Red such that with probability 1 over $\mathcal{R}$, Red reduces L to $\mathsf{SD}^{\mathcal{R}}$. As before, we claim that if Red reduces L to $\mathsf{SD}^{\mathcal{R}}$ with probability 1 over $\mathcal{R}$, then $\Pr_{\mathcal{R}}[\mathsf{Red}$ succeeds on $L_n] \to 1$ as $n \to \infty$. The proof is identical to the proof of Claim 4.1.

Since for large enough n, Red succeeds on L_n with probability 99/100 over the choice of $\mathcal{R}$, and we can then hardwire $\mathcal{R}$ to place $L \in \mathbf{SZK}$:

Claim 4.7. *If for all large enough* n $\Pr_{\mathcal{R}}[\mathsf{Red}$ *succeeds on* $L_n] \geq 99/100$, *then* $L \in \mathbf{SZK}$.

The claim is proven by reducing L to SD using the same argument as in the proof of Lemma 4.2. The only difference is that in order for the circuit Q to sample the fake oracle $\mathcal{R}_Q$ identically distributed to $\mathcal{R}$ on all inputs up to length $2 \log 10^8 p = O(\log n)$ and 0 on longer inputs, Q must be able to decide $\mathsf{QBF}_*^{\mathcal{R}}$ formulas of size up to $\sqrt{2 \log 10^8 p} = O(\sqrt{\log n})$. Q can do this in polynomial size by brute force, because $\mathsf{QBF}_*^{\mathcal{R}}$ can be decided in time $2^{O(n^2)}$ (using standard results about QBF, *e.g.* Proposition A.1) and the inputs here are of size $O(\sqrt{\log n})$.

■

5 Open Questions

Recently, Aaronson and Wigderson [AW08] proposed another barrier to reductions, algebrization. It is natural to ask whether one can rule out algebrizing techniques for showing that hardness of PAC learning is equivalent to $\mathbf{ZK} \neq \mathbf{BPP}$. More generally, it would be interesting to understand better the role of algebrizing techniques in cryptography and learning theory.

Our results are silent about whether reductions where the construction is black-box but the security analysis only holds for adversaries making no oracle calls (so-called mildly black-box reductions, see [RTV04]) can base $\mathbf{ZK} \neq \mathbf{BPP}$ on hardness of learning. For example, the zero knowledge argument of Barak [Bar01] is bases $\mathbf{NP} \subseteq \mathbf{ZK}$ (with nice additional properties) on standard assumptions, and is only mildly-black-box because the security analysis uses the PCP theorem. It would be interesting to understand whether such techniques are useful in our setting.

Our proof of Theorem 1.2 also does *not* rule out relativizing constructions of zero-knowledge protocols for **NP**-complete languages from hardness of learning. This is because we use the fact that in semi-black-box proofs, there is a single procedure that uses black-box access to $\mathcal{R}$ and produces a zero-knowledge protocol, and this implies we have a single reduction Red reducing L to $\mathsf{SD}^{\mathcal{R}}$. A relativizing proof could conceivably imply a radically different Red for each $\mathcal{R}$, and so there may not be a single Red reducing L to $\mathsf{SD}^{\mathcal{R}}$. It is an interesting open question whether one can rule out relativizing reductions in this setting as well.

Acknowledgements

The author thanks Avi Wigderson for several stimulating conversations and suggestions. The author thanks the anonymous CCC reviewers for many helpful comments. The author also thanks Boaz Barak and Salil Vadhan for interesting discussions. The author was supported by NSF grants CNS-0627526, CCF-0426582 and CCF-0832797, and a grant from the Packard foundation.

References

[AW08] S. Aaronson and A. Wigderson. Algebrization: a new barrier in complexity theory. In *Proc. STOC '08*, pages 731–740, New York, NY, USA, 2008. ACM.

[AH91] W. Aiello and J. Hastad. Statistical Zero-Knowledge Languages can be Recognized in Two Rounds. *JCSS*, 42:327–345, 1991.

[ABX08] B. Applebaum, B. Barak, and D. Xiao. On Basing Lower-Bounds for Learning on Worst-Case Assumptions. In *Proc. FOCS '08*, pages 211–220, 2008.

[AB09] S. Arora and B. Barak. *Complexity Theory: A Modern Approach*. Cambridge University Press, 2009.

[Bar01] B. Barak. How to go beyond the black-box simulation barrier. In *Proc. 42nd FOCS*, pages 106–115. IEEE, 2001.

[For87] L. Fortnow. The complexity of perfect zero-knowledge. In *STOC '87*, pages 204–209, 1987.

[GT00] R. Gennaro and L. Trevisan. Lower bounds on the efficiency of generic cryptographic constructions. In *Proc. 41st FOCS*, pages 305–313. IEEE, 2000.

[GGM86] 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.

[GMW86] 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.

[GMR85] S. Goldwasser, S. Micali, and C. Rackoff. The knowledge complexity of interactive proof-systems. In *Proc. 17th STOC*, pages 291–304. ACM, 1985.

[HHRS07] I. Haitner, J. J. Hoch, O. Reingold, and G. Segev. Finding collisions in interactive protocols - A tight lower bound on the round complexity of statistically-hiding commitments. In *Proc. FOCS '07*, pages 669–679, 2007.

[HILL89] J. Håstad, R. Impagliazzo, L. A. Levin, and M. Luby. A pseudorandom generator from any one-way function. *SIAM J. of Com.*, 28(4):1364–1396, 1999. Preliminary versions appeared in STOC' 89 and STOC' 90.

[HR04] C. Hsiao and L. Reyzin. Finding Collisions on a Public Road, or Do Secure Hash Functions Need Secret Coins. In *Proc. CRYPTO '04*, pages 92–105. Springer, 2004.

[KS01] A. R. Klivans and R. A. Servedio. Learning DNF in Time $2^{\tilde{O}(n^{1/3})}$. In *Proc. STOC '01*, pages 258–265, 2001.

[LMN93] N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, Fourier transform, and learnability. *J. ACM*, 40(3):607–620, 1993.

[OV07] S. J. Ong and S. P. Vadhan. Zero Knowledge and Soundness Are Symmetric. In *EUROCRYPT '07*, pages 187–209, 2007.

[Ost91] R. Ostrovsky. One-way functions, hard on average problems, and statistical zeroknowledge proofs. In *In Proc. 6th Annual Structure in Complexity Theory Conf.*, pages 133–138, 1991.

[OW93] R. Ostrovsky and A. Wigderson. One-Way Functions are essential for Non-Trivial Zero-Knowledge. In *ISTCS '93*, pages 3–17, 1993.

[PV88] L. Pitt and L. G. Valiant. Computational limitations on learning from examples. *J. ACM*, 35(4):965–984, 1988.

[PW90] L. Pitt and M. K. Warmuth. Prediction-preserving reducibility. *J. Comput. Syst. Sci.*, 41(3):430–467, 1990.

[RTV04] O. Reingold, L. Trevisan, and S. Vadhan. Notions of Reducibility Between Cryptographic Primitives. In *Proc. 1st TCC*, pages 1–20, 2004.

[Sim98] D. R. Simon. Finding collisions on a one-way street: Can secure hash functions be based on general assumptions? In *Proc. EUROCRYPT '98*, volume 1403, pages 334–345, 1998.

[Vad04] S. P. Vadhan. An Unconditional Study of Computational Zero Knowledge. *FOCS '04*, pages 176–185, 2004.

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

A Technical lemmas

Proposition A.1. $\mathsf{QBF}_*^{\mathcal{R}}$ *is* $\mathbf{PSPACE}_*^{\mathcal{R}}$*-complete.*

Proof. $\mathsf{QBF}_*^{\mathcal{R}} \in \mathbf{PSPACE}_*^{\mathcal{R}}$: this immediate because the proof that $\mathsf{QBF} \in \mathbf{PSPACE}$ relativizes. On input φ, M_1 takes φ and outputs all the z such that φ contains a $\mathcal{R}_z$ gate to obtain $z_1, \ldots, z_m$. M_2 then simply decides φ using access to the $\mathcal{R}_{z_i}$ gates. This runs in space $O(n^2)$ and therefore time $2^{O(n^2)}$, see *e.g.* the analysis in [AB09].

All $L \in \mathbf{PSPACE}_*^{\mathcal{R}}$ reduce to $\mathsf{QBF}_*^{\mathcal{R}}$: recall the proof that QBF is complete for $\mathbf{PSPACE}$ (see *e.g.* [AB09]). For a $\mathbf{PSPACE}$ machine M with space bound $p(n)$ and an input x, we look at the configuration graph of M on input x. A state of the configuration graph is describable by a string of size $O(p(n))$. Furthermore, there is a $O(p(n))$ size formula $\phi_{M,x}$ that describes edges in the configuration graph: namely, given $S, S' \in \{0,1\}^{p(n)}$, $\phi_{M,x}(S, S') = 1$ iff S' follows from one step of the computation of M starting with configuration S. The QBF formula is constructed recursively by contracting paths in the configuration graph: we initialize $\psi_1 = \phi$ and define

$$\psi_i(S, S') = \exists S'', \forall T_1, T_2, (T_1 = S \wedge T_2 = S'') \vee (T_1 = S'' \wedge T_2 = S') \Rightarrow \psi_{i-1}(T_1, T_2)$$

and the final output formula is $\psi_{p(n)}(S_0, S_a)$ where S_0 is the initial configuration and S_a is an accepting final configuration. One can check that $|\psi_{p(n)}(S_0, S_a)| = O(p(n)^2)$.

To generalize this reduction to $\mathbf{PSPACE}_*^{\mathcal{R}}$, on input x our reduction first uses M_1 to obtain $z_1, \ldots, z_m$. Now, it produces the formula $\phi_{M,x}$, which contains only (say) NAND gates and gates of the form $\mathcal{R}_{z_i}$. Then, run the same reduction as in the $\mathbf{PSPACE}$ case, which gives us the final formula $\psi_{p(n)}(S_0, S_a)$ which contains only $\mathcal{R}_z$ gates with explicit z (*i.e.* those obtained from M_1). ■

For any $\mathbf{PSPACE}^{\mathcal{O}}$ relation R, a $\mathbf{PSPACE}^{\mathcal{O}}$ oracle can count the number of satisfying pairs $\{(x, y) \mid R(x, y) = 1\}$ by enumerating over all pairs and checking the relation. We use this to show the following two facts.

Proposition A.2. *There is an efficient oracle algorithm A that, for every $\mathcal{O}$, $A^{\mathbf{PSPACE}^{\mathcal{O}}}$ takes input a circuit $C : \{0,1\}^\ell \to \{0,1\}^m$ with oracle gates and a string $y \in \{0,1\}^m$, and outputs a uniform element of the set $\{x \mid C^{\mathbf{PSPACE}^{\mathcal{O}}}(x) = y\}$ with probability at least $1 - 2^{-|y|}$, and outputs a special failure symbol $\perp$ with the remaining probability.*

Proof. The computation of C on inputs of length ℓ can be expressed as a polynomial-size $\mathsf{QBF}^{\mathcal{O}}$ formula (QBF with $\mathcal{O}$ gates), and so we can use a $\mathbf{PSPACE}^{\mathcal{O}}$ oracle to compute $s = |(C^{\mathbf{PSPACE}^{\mathcal{O}}})^{-1}(y)|$. Now pick a random number $i \leftarrow_{\mathrm{R}} [s]$ and use the $\mathbf{PSPACE}^{\mathcal{O}}$ oracle to output the i'th lexicographically ordered string in $f^{-1}(y)$. There is some probability of failure because sampling a number in $[s]$ may have a probability of failure if s is not a power of 2, but this can be made to be smaller than $2^{-|y|}$ by repeating the procedure. ■

Proposition A.3. *There is an efficient oracle algorithm A that, for every $\mathcal{O}$, $A^{\mathbf{PSPACE}^{\mathcal{O}}}$ takes input two oracle circuits $C : \{0,1\}^\ell \to \{0,1\}^m$ and circuit $D : \{0,1\}^m \times \{0,1\}^n \to \{0,1\}$ computing a predicate, and a unary string 1^p and outputs a set*

$$S = \left\{ y \;\middle|\; \Pr_{x \leftarrow_R U_\ell}\left[D^{\mathbf{PSPACE}^{\mathcal{O}}}(C^{\mathbf{PSPACE}^{\mathcal{O}}}(x), y) = 1\right] \geq 1/p \right\}$$

Proof. Since $\mathbf{PSPACE}^{\mathcal{O}}$ is capable of counting $\mathbf{PSPACE}^{\mathcal{O}}$ relations, A simply iterates over all $x \in \{0,1\}^m, y \in \{0,1\}^n$ and outputs all y such that the number of x such that $D^{\mathbf{PSPACE}^{\mathcal{O}}}(C^{\mathbf{PSPACE}^{\mathcal{O}}}(x), y) = 1$ is larger than $2^n/p$. There can be at most p such y, so the procedure runs in polynomial space. ■

The standard Chernoff shows that the empirical average of many samples drawn from a distribution deviates from the mean of the distribution with exponentially small probability. We use the fact that this also holds for weighted empirical averages, as long as the weights are relatively smooth.

Lemma A.4 (Generalized Chernoff bound)**.** *Let D be a distribution over a finite universe U such that $\max_{u \in U} \Pr[D = u] \leq 1/k$. Let F be a distribution on functions $f : U \to \{0, 1\}$. Let $\mu = \mathbb{E}_{D,F}[F(D)]$ and let $\mu_u = \mathbb{E}_F[F(u)]$. Then*

$$\Pr_F \left[\mathbb{E}_D[F(D)] > \mu + \gamma\right] < e^{-\gamma^2 k/2}$$

Proof. One can plug in the fact that $\max_{u \in U} \Pr[D = u] \leq 1/k$ in a straight-foward way into the proof of the standard Chernoff bound (see for example the appendix of [AB09]), which considers the case that D is uniform over a universe of size k. That is, we derive that for any positive constant t:

$$\begin{aligned}
\Pr_F \left[\mathbb{E}_D[F(D)] > \mu + \gamma\right] &= \Pr_F \left[e^{t(k\mathbb{E}_D[F(D)] - k\mu)} > e^{tk\gamma}\right] \\
&\leq e^{-tk\gamma} \mathbb{E}_F \left[e^{t(k\mathbb{E}_D[F(D)] - k\mu)}\right] \\
&\leq e^{-tk\gamma} \mathbb{E}_F \left[e^{t(k\mathbb{E}_D[F(D) - \mu_D])}\right] \\
&\leq e^{-tk\gamma} \mathbb{E}_F \left[e^{t(\sum_{u \in \mathrm{supp}(D)} F(u) - \mu_u)}\right] \qquad \textit{(using } \Pr[D = u] \leq 1/k\textit{)} \\
&= e^{-tk\gamma} \prod_{u \in \mathrm{supp}(D)} \mathbb{E}_F \left[e^{t(F(u) - \mu_u)}\right] \\
&\leq e^{-tk\gamma + t^2 k} \qquad \textit{(using } |\mathrm{supp}(D)| \geq k \textit{ plus Taylor expansion)} \\
&= e^{-\gamma^2 k/2}
\end{aligned}$$

where the last line follows from setting $t = \gamma/2$. ∎

B Ostrovsky-Wigderson Theorem

The Ostrovsky-Wigderson theorem (Theorem 2.1) is relativizing but not fully black-box. We sketch the proof in order to point out the precise argument that is non-black-box: supposing that there exist no AIOWF, we show that **ZK** = **BPP**. Fix any $L \in$ **ZK** with simulator S. It suffices to show that the "simulation-based prover" is efficiently computable: the simulation-based prover is defined by the conditional distribution of the simulator. Given a prefix of messages $m_1, \ldots, m_i$ (say m_i is a verifier message), the simulated prover samples a message m_{i+1} according to the distribution $S(x, U_r)$ conditioned on the first i messages being $m_1, \ldots, m_i$. If one could efficiently compute the simulated prover distribution (or approximate it) then this would give an algorithm for L: run the honest verifier and interact it with the simulated prover. By the zero-knowledge property the verifier will accept $x \in L$, and by soundness the verifier will reject $x \notin L$.

We show how to approximate the simulated prover assuming AIOWF do not exist. Let $S_i(x, \cdot)$ be the function that outputs the first i messages of the simulator. Suppose that the i'th message is sent by the receiver, then one way to sample the simulated prover's $i+1$'th message in response to a partial transcript $\tau_i = (m_1, \ldots, m_i)$ is to first invert invert $S_i(x, \cdot)$ on τ_i to obtain random coins r such that $S_i(x, r) = \tau_i$, and then compute $S_{i+1}(x, r)$ and output the $i+1$'th message. Assuming that AIOWF do not exist, this inversion procedure is efficient. In addition, the fact that the inversion procedure is *efficient* is critical because we only have the guarantee that the output of the simulator is *computationally* indistinguishable from the honest transcript. A priori, it is conceivable that the honest transcript and the simulator transcript have disjoint support but remain computationally indistinguishable, in which case inverting an honest transcript as if it were output by the simulator is information-theoretically impossible. But the assumption that the protocol is zero knowledge combined with the assumption that AIOWF do not exist means that this is not the case, since otherwise the inverter for the AIOWF would give an efficient distinguisher for the simulator. Thus the proof is not black-box since the proof uses the fact that the inverter is efficient in a critical way. Note however that it is relativizing: if all the algorithms are given access to an oracle and the hardness is assumed to be against algorithms with access to the oracle, the same reasoning goes through.