Electronic Colloquium on Computational Complexity, Report No. 6 (2009)

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

David Xiao

January 19, 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, Barak, and Xiao (FOCS 2008) established a connection between these two different notions of learning by observing that if there exist non-trivial languages with zero-knowledge proofs (*i.e.* $\mathbf{ZK} \neq \mathbf{BPP}$), then no polynomial-time algorithm can PAC learn polynomial-size circuits. In this paper, we consider the reverse implication: is it true that if learning is hard then zero-knowledge proofs exist for non-trivial languages? We rule out two classes of techniques for proving this statement:

1. Relativizing techniques: there exists an oracle $\mathcal{O}$ relative to which learning polynomial-size circuits is hard and yet $\mathbf{ZK}^{\mathcal{O}} = \mathbf{BPP}^{\mathcal{O}}$.
2. Black-box techniques: if there is a (semi-)black-box proof that uses the hardness of PAC learning polynomial-size circuits to construct a zero knowledge proof for some language L, then in fact $L \in \mathbf{AM} \cap \mathbf{coAM}$.

Together, these results rule out all known techniques for proving that hardness of learning implies $\mathbf{ZK} \neq \mathbf{BPP}$, including partially non-black-box techniques such as those of Barak (FOCS 2001). In addition, our technique relies on a new kind of separating oracle that may be of independent interest.

1 Introduction

Computational learning theory began with the study of PAC learning [Val84], and understanding what efficient algorithms can learn in the PAC model remains an important goal. In the PAC model, the learning algorithm is said to learn all functions in a *concept class* F (*e.g.* linear functions over $\mathbb{F}_2^n$, half-spaces, DNF's) if given access to many labelled examples $(X, f(X))$ drawn from an arbitrary input distribution X and where $f \in F$, the learner outputs with high probability a *hypothesis* h 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 only know 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 average-case hardness assumptions [GGM86, PW90] or even based on **NP**-hardness if we restrict the form of the hypothesis h the

ISSN 1433-8092

learner outputs (*e.g.* [PV88]; it seems unlikely that we can prove hardness of learning under **NP**-hardness if h is unrestricted, see [ABX08]).

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 a prover P interacts with an efficient verifier V in order to prove a statement π such that the verifier "learns nothing" except that π is true. In this setting, we say that V learns nothing if it is able to simulate its interaction with the prover by itself, in other words anything the verifier could compute after interacting with the prover, it could compute by itself anyway.

Although these notions superficially seem unrelated besides intuitively capturing some notion of "learning", Applebaum, Barak, and the author [ABX08] observed that they are intimately connected. Specifically, we observed that if there are non-trivial zero knowledge protocols (*i.e.* $\mathbf{ZK} \neq \mathbf{BPP}$) then PAC learning is hard.

This raises the natural question: does hardness of (non-uniform) learning imply $\mathbf{ZK} \neq \mathbf{BPP}$? Such a result would mean that the two disparate notions of learning are in fact the same (at least from a complexity-theoretic point of view), and so since both bodies of literature are large and well-developed, many results from one side might have yet-to-be-discovered consequences on the other side.

Already [ABX08] showed a partial result of this form, working with the promise problem 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 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" since Y can be computed from X by a size n^2 circuit. 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 $n^{\log\log n}$ (any super-polynomial size will do), $\Pr[Y = g(X)] \leq 1 - 1/\mathrm{poly}(n)$. That is, a no instance is "unlearnable" because no small circuit can compute Y from X. [ABX08] show the following proposition:

Proposition 1.1 ([ABX08])**.** Learnability $\in$ **ZK**.

Thus, if Learnability $\notin$ **BPP** then $\mathbf{ZK} \neq \mathbf{BPP}$. One might hope to generalize this result to show that (non-uniform) hardness of learning implies $\mathbf{ZK} \neq \mathbf{BPP}$. In this paper we show that this is not the case, at least if we restrict our attention to standard proof techniques. As we will explain later, the key difference between Learnability and standard PAC learning is that in Learnability, we are given the circuit C sampling (X,Y), whereas in PAC learning we are only given a set of labelled examples and cannot otherwise describe the distribution (X,Y).

1.1 Our results

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 non-uniform 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 negligible 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 against circuits, then $\mathbf{ZK} \neq \mathbf{BPP}$.

Theorem 1.2. *There exists an oracle $\mathcal{O}$ relative to which learning is hard against circuits but where $\mathbf{ZK}^{\mathcal{O}} = \mathbf{BPP}^{\mathcal{O}}$.*

In fact, we prove the stronger statement that relative to $\mathcal{O}$, learning is hard against circuits 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 [OW93] (stated in Corollary 2.4).[1] An AIOWF is an efficiently computable function $f : \{0,1\}^n \times \{0,1\}^{\ell(n)} \to \{0,1\}^{m(n)}$ such that for every efficient algorithm A, there exists an infinite sequence of w such that for $y \overset{R}{\leftarrow} f(w, U_\ell)$, the probability $A(w, y)$ outputs a preimage of y is negligible.

Unfortunately, in this setting ruling out relativizing proofs is not very convincing because we have various non-relativizing proofs that $\mathbf{ZK} \neq \mathbf{BPP}$. In particular the proofs by Goldreich, Micali, and Wigderson [GMW91] and Nguyen, Ong, and Vadhan [NOV06] that **NP** has various forms of zero knowledge protocols based on the existence of one-way functions do not relativize because they work directly with the **NP**-complete problem Three Coloring (3-COL).

Semi-black-box proofs: [GMW91, NOV06] do not relativize in the traditional sense, but are black-box: they require only black-box access to a one-way function to construct a protocol for 3-COL, which they then prove is zero-knowledge by assuming the one-way function is hard to invert. Our second result rules out *semi-black-box proofs* that zero knowledge is non-trivial based on the hardness of learning. A semi-black-box proof uses the hard concept class as a black-box to construct a zero-knowledge protocol, but the analysis, which takes an adversary for breaking zero knowledge and converts it into a learning algorithm, may be non-black-box (although the adversary breaking zero knowledge is still allowed black-box access to the hard concept class). In contrast, a proof is *fully-black-box* if the analysis is also black-box, *i.e.* the adversary breaking zero knowledge is only accessed as a black box. See the taxonomy of Reingold, Trevisan, and Vadhan [RTV04] for more details about classifying black-box proofs. We note that by ruling out semi-black-box proofs, our second theorem rules out even constructions such as those of Barak [Bar01], which have a non-black-box analysis but a black-box construction.

Unlike Theorem 1.2, our second theorem does not unconditionally rule out semi-black-box proofs because there *are* zero knowledge protocols whose security is unconditional (*e.g.* for Graph Isomorphism, Quadratic Residuosity). That is, it is conceivable one can build a statistical zero knowledge proof (*i.e.* a **SZK** proof, defined in Section 2) for **NP** without complexity assumptions, which is then trivially semi-black-box. We prove roughly that this is the only thing that can happen:

Theorem 1.3. *If there exists a semi-black-box proof that constructs a* **ZK** *protocol for a language L based on non-uniform hardness of learning, then in fact $L \in \mathbf{AM} \cap \mathbf{coAM}$.*

Notice the theorem is not quite what our intuition suggests: our conclusion is $L \in \mathbf{AM} \cap \mathbf{coAM}$ rather than $L \in \mathbf{SZK}$. We elaborate on this discrepancy in Remark 4.4. Nevertheless, if L is **NP**-complete then $L \in \mathbf{AM} \cap \mathbf{coAM}$ implies that the polynomial hierarchy collapses to $\mathbf{\Sigma_2}$

[1] In fact, our proof also rules out so-called $\forall\exists$ semi-black-box proofs, which are even more general than relativizing proofs. See Appendix B for details.

[BHZ87, For87], which we interpret to mean that such a semi-black-box proof for an **NP**-complete languages is unlikely to exist.

1.2 Our techniques

Relativizing proofs: the intuitive difference between PAC learning and inverting AIOWF we exploit is that in PAC learning, the learner knows nothing about the function labelling the examples other than it belongs to some class F. This is intrinsic to the model, since if the learner knew the description of the function it could simply use it to label new examples. On the other hand, in AIOWF, the inverting algorithm knows exactly a *description* of the function $f(w, \cdot)$ it is trying to invert.

Our oracle will be defined using a distribution over functions $\mathcal{R}^{(n)} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, where for z of length n, we let $\mathcal{R}_z$ denote the function $\mathcal{R}^{(n)}(z, \cdot)$. 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 $\mathcal{R}_z \equiv 0$.

It can be shown (*e.g.* 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 polynomial-size circuits (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 it cannot learn the hard instances it cannot find because they look like random functions.

As a first step, let us see how to rule out fully-black-box reductions that use non-uniform hardness of learning to construct AIOWF. Consider the oracle $\mathcal{O} = (\mathcal{R}, \mathcal{I})$, where $\mathcal{R}$ is defined as above and $\mathcal{I}$ takes input $(C^{\mathcal{R}}, y)$ where y is a string and $C^{\mathcal{R}}$ is a circuit with $\mathcal{R}$ gates with output length $|y|$, and $\mathcal{I}$ outputs an arbitrary element of $(C^{\mathcal{R}})^{-1}(y)$. It can be shown that $F = \{\mathcal{R}_z\}_{z\in\{0,1\}^*}$ cannot be learned by any efficient circuit even if given access to $\mathcal{O}$: $\mathcal{I}$ does the learning circuit C no good since it cannot help C to find more hard instances than C can find by itself. Any fully-black-box proof produces from F an AIOWF, and the AIOWF is computable using only calls to $\mathcal{R}$ (since F only has $\mathcal{R}$ gates and no $\mathcal{I}$ gates). But any such AIOWF can be inverted using $\mathcal{I}$ because we know the description of the AIOWF, which we can then use $\mathcal{I}$ to invert. Since the analysis is also black-box we get a learner for F that only makes $\text{poly}(n)$ queries to $\mathcal{O}$, a contradiction.

This type of proof technique is common in the cryptographic literature (see *e.g.* [HHRS07, HR04]) but it does not rule out relativizing reductions: it does not rule out an AIOWF where the *function itself* calls $\mathcal{I}$, because $\mathcal{I}$ may not be able to invert circuits that call $\mathcal{I}$. Ruling out relativizing techniques requires a more general oracle, which we describe below.[2] To the best of our knowledge, this is the first time such an oracle has been proposed, and it may be of use for other black-box separations as well.

Definition 1.4. Let $\mathbf{PSPACE}^{\mathcal{R}}_*$ be the class of languages decidable by the following kind of machine: on input x first run a polynomial-time machine $M_1(x)$ to obtain outputs $z_1, \ldots, z_m \in \{0,1\}^*$, then run a **PSPACE** machine $M_2(x)$ with access to oracle gates $\mathcal{R}_{z_1}, \ldots, \mathcal{R}_{z_m}$ and output the result of M_2.

[2] Simon [Sim98] solves a similar problem: in our setting, his technique would allow $\mathcal{I}$ to also invert circuits containing $\mathcal{I}$ queries (but only up to logarthmic recursions), and then we would need to prove that still this does not help the learner. While such a technique may succeed, we present our result with the oracle above, which we believe may be of independent interest.

There is a natural complete language $\mathsf{QBF}_*^{\mathcal{R}}$ for this class, described in Section 2.

Our oracle $\mathcal{O}$ is a $\mathbf{PSPACE}_*^{\mathcal{R}}$ oracle where $\mathcal{R}$ is chosen from the same distribution as above. We will argue that relative to this oracle, the same concept class F is still hard to learn for circuits, again because no learner can find all the hard instances. On the other hand, we will show that all functions f in $\mathbf{PSPACE}_*^{\mathcal{R}}$ can also be inverted in $\mathbf{PSPACE}_*^{\mathcal{R}}$ given the description of f, so no f can be an AIOWF because the inverter knows the description of f.

Semi-black-box proofs: we describe the intuition behind Theorem 1.3 for fully-black-box reductions, where the analysis is also black-box. We use the same distribution of functions $\mathcal{R}$ and also set $\mathcal{O} = \mathbf{PSPACE}_*^{\mathcal{R}}$. (To rule out semi-black-box reductions we "embed" $\mathbf{PSPACE}_*^{\mathcal{R}}$ inside $\mathcal{R}$. See Section 4 for details.)

The family $F = \{\mathcal{R}_z\}_{z\in\{0,1\}^*}$ is hard to learn for circuits for the same reason as before, so we have by the black-box reduction 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 that if $L \in \mathbf{ZK}$, then either there exists an AIOWF, or L reduces to "Statistical Difference" (SD) which is complete for the class $\mathbf{SZK}$ (see Theorem 2.3), 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 [OV07] that L reduces to $\mathsf{SD}^{\mathcal{O}}$ (where circuits can contain $\mathcal{O}$ gates).

Furthermore, since the proof is black-box, the construction only uses oracle access to F, which is implementable using only access to $\mathcal{R}$, and we can observe that [OV07] 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 almost all $\mathcal{R}$, the oracle $\mathcal{R}$ does not contain information about L itself. To use this intuition formally, we simply replace the $\mathcal{R}$ gates in instances of $\mathsf{SD}^{\mathcal{R}}$ by hard-wired random bits, which gives an instance of plain SD. Since all but a "small" fraction of $\mathcal{R}$ are good, this gives a good randomized reduction to SD. However, the "small" fraction of bad $\mathcal{R}$ is not negligible, which is why we cannot prove $L \in \mathbf{SZK}$, but instead prove just $L \in \mathbf{AM} \cap \mathbf{coAM}$.

2 Preliminaries

For any distribution X, we let $x \xleftarrow{R} X$ denote a random variable sampled according to X. If S is a finite set, $x \xleftarrow{R} S$ 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 $x \xleftarrow{R} X$. 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 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 standard that QBF is $\mathbf{PSPACE}$-complete (see *e.g.* [AB]). 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 $\mathcal{R}^{(n)}(z,\cdot)$ gates in addition to NAND gates, but only for *fixed* auxiliary inputs z (*e.g.* $\exists z, \mathcal{R}^{(n)}(z,x)$ is not a valid formula for $\mathsf{QBF}_*^{\mathcal{R}}$). It follows immediately from the proof that QBF is complete for $\mathbf{PSPACE}$ that $\mathsf{QBF}_*^{\mathcal{R}}$ is complete for $\mathbf{PSPACE}_*^{\mathcal{R}}$ (defined previously in Definition 1.4). We include a proof of this in Appendix A for the sake of completeness.

PAC Learning: we say that a family of circuits $C = \{C_n\}$ learns a family of functions F (called a *concept class*) 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 an unlabelled $x' \stackrel{R}{\leftarrow} X$, $C_n(S, x') = f(x')$ with probability at least $\frac{1+\varepsilon}{2}$.[3] We say learning is non-uniformly hard or hard against circuits if no family of poly-size circuits can learn functions computable by circuits of size n^2 with advantage $\varepsilon = 1/\text{poly}(n)$.[4] Learning relative to an oracle $\mathcal{O}$ is defined in the obvious way: both the concept classes and learning circuits are allowed queries to $\mathcal{O}$.

Auxiliary-input one-way functions: we say that a family of functions $f^{(n)} : \{0,1\}^n \times \{0,1\}^{\ell(n)} \to \{0,1\}^{m(n)}$ is a family of *auxiliary-input one-way functions* (abbreviated AIOWF) against uniform (resp. non-uniform) adversaries if it is efficiently computable and for every polynomial-time inverting algorithm A (resp. non-uniform algorithm), there exists an infinite set $W \subseteq \{0,1\}^*$ such that for every $w \in W$ of length n

$$\Pr_{x \stackrel{R}{\leftarrow} U_{\ell(n)}} [A(w, y) \in f_w^{-1}(y) \mid y = f_w(x)] < n^{-\omega(1)}$$

where we use the short-hand $f_w(x) = f^{(n)}(w, x)$ and $f_w^{-1}(y) = \{x \mid f_w(x) = y\}$. Note that in the above the hard instances W might depend on the inverter A. We say that $f^{(n)}$ has a set of *universally hard instances* $W \subseteq \{0,1\}^*$ if the *same* W is hard against all A. The definitions relativize in the obvious way: both the algorithm computing the function as well as the adversaries are allowed access to the oracle.

Zero-knowledge: zero knowledge proofs 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 computation zero knowledge arguments **HV-CZKA** in 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 is a prover strategy and efficient (randomized) verifier strategy such that the following hold:

- Completeness: $\forall x \in L$, 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}.[5]
- Zero knowledge: there exists an efficient simulator S such that $\forall x \in L$ and any auxiliary input a of length $\text{poly}(|x|)$, the distribution $\langle P, V(a) \rangle(x)$ is computationally indistinguishable from $S(x, a)$.

[3] The success condition is syntactically different from Valiant's original definition [Val84], but can be proven to be equivalent [HKLW88]. We use it here to simplify the presentation of our results. Also, we define learning non-uniformly, which makes the statements of our results stronger than if we used a uniform definition.

[4] We consider n^2-size circuits rather than $\text{poly}(n)$-size circuits since *wlog* one follows from the other by standard padding arguments.

[5] There is an odd asymmetry here: we allow the honest prover to be unbounded but demand only soundness against efficient cheating provers. This can be removed using [OV07], who show that if $L \in \mathbf{NP}$ then any **ZK** argument for L can be transformed into one with many additional properties including an efficient prover. However, this is irrelevant for us: we prove that *even allowing the honest prover to be unbounded*, still such an argument system is unlikely to exist.

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.* $\Delta(\langle P, V(a)\rangle(x), S(x,a)) \leq n^{-\omega(1)}$. We now review some facts about **ZK** and **SZK**.

Theorem 2.1 ([Ost91])**.** *If* $\mathbf{SZK} \neq \mathbf{BPP}$ *then there exist AIOWF against uniform algorithms.*

Theorem 2.2 ([SV97])**.** *For any* $\alpha, \beta \in [0,1]$ *satisfying* $\alpha^2 > \beta$*, the following promise problem (Statistical Difference,* SD*) is complete for* **SZK***. We identify a circuit* $X : \{0,1\}^m \to \{0,1\}^n$ *with the output distribution* $X(U_m)$*. A yes instance is a pair of circuits* (X, Y) *such that the* $\Delta(X, Y) > \alpha$*, and a no instance is a pair of circuits* (X, Y) *such that* $\Delta(X, Y) < \beta$*.*

Theorem 2.3 (SZKP/AIOWF characterization, [OV07])**.** *If* $L \in \mathbf{ZK}$*, then there exists an efficient reduction* Red *such that exactly one of the following holds:*

1. Red *reduces* L *to* SD
2. *There exists an infinite subset* $W \subseteq \{0,1\}^*$ *and an efficiently computable* $f^{(n)} : \{0,1\}^n \times \{0,1\}^{\ell(n)} \to \{0,1\}^{m(n)}$ *such that* $f^{(n)}$ *is an AIOWF against non-uniform circuits with universally hard instances* W*.*

Corollary 2.4 ([OV07, OW93])**.** *If* $\mathbf{ZK} \neq \mathbf{BPP}$ *then there exists AIOWF against uniform algorithms.*

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 theorems, we observe that they all relativize, that is:

Proposition 2.5. *For any oracle* $\mathcal{O}$*, Theorem 2.1, Theorem 2.2, Theorem 2.3, Corollary 2.4 all hold relative to* $\mathcal{O}$*.*

3 Relativizing techniques

Our main result for relativizing techniques is to separate hardness of learning and AIOWF. First we recall the oracle.

Definition 3.1. We define the distribution over oracles $\mathcal{O}$ as follows. First, 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 for circuits, but no AIOWF exists.*

Proof. The theorem immediately follows from the following two lemmas, proven in Appendix B. We sketch their proofs here.

Lemma 3.3. *With probability 1 over the choice of $\mathcal{R}$, the concept class $F = \{\mathcal{R}_z\}_{z\in\{0,1\}^*}$ is hard to learn by any efficient oracle machine with access to $\mathcal{O}$.*

To prove this lemma, we show that any oracle circuit $C^{\mathcal{O}}$ has probability $2^{-2^{\Omega(n)}}$ of learning all $\mathcal{R}_z$ simultaneously on z of length n. This proof follows from a case analysis: first we show that it is unlikely $C^{\mathcal{O}}$ learns $\mathcal{R}_z$ without querying the oracle at $\mathcal{R}_z$ (since otherwise $\mathcal{R}_z$ looks like a random function), and then we show that the probability $C^{\mathcal{O}}$ queries $\mathcal{R}_z$ given just oracle access to an example oracle is negligible since the function $\mathcal{R}_z$ is random and therefore contains no information about z.

Lemma 3.4. *With probability 1 over choice of $\mathcal{O}$ as in Definition 3.1, it holds that for any efficient oracle algorithm f computing a function $f^{\mathcal{O}} : \{0,1\}^n \times \{0,1\}^{\ell(n)} \to \{0,1\}^{m(n)}$, $f^{\mathcal{O}}$ can be inverted by an efficient oracle algorithm A on every auxiliary input w. Namely, for every $w \in \{0,1\}^n$*

$$\Pr_{x\xleftarrow{R}\{0,1\}^{\ell(n)}}[A^{\mathcal{O}}(w, y) \in (f_w^{\mathcal{O}})^{-1}(y) \mid f_w^{\mathcal{O}}(x) = y] > 1/2$$

Roughly A works as follows: it finds all the z such that $f_w^{\mathcal{O}}(x)$ queries $\mathcal{R}_z$ with noticeable probability over choice of random input x. We show that by finding all such "heavy" queries, A knows most of the *hard instances* z such that $f_w^{\mathcal{O}}$ queries $\mathcal{R}_z$. It is well-known that a $\mathbf{PSPACE}^{\mathcal{R}}$ oracle can invert all $\mathbf{PSPACE}^{\mathcal{R}}$ computations (see *e.g.* Proposition A.2), so since $f_w^{\mathcal{O}}$ is computable using a $\mathbf{PSPACE}^{\mathcal{R}}$ oracle and since A knows all the hard instances that $f_w^{\mathcal{O}}$ queries, it can "effectively"simulate $\mathbf{PSPACE}^{\mathcal{R}}$ for the purposes of inverting $f_w^{\mathcal{O}}$. ■

Notice that combining Corollary 2.4 (which relativizes) with Theorem 3.2 we obtain our main theorem about relativizing proofs Theorem 1.2. Actually, this argument already rules out a more general class of proofs, namely so-called $\forall\exists$ semi-black-box reductions. We define these reduction in Appendix B and explain why our result rules them out.

4 GMW-style techniques

We first prove the result for fully-black-box reductions to demonstrate the main ideas. The proofs of the lemmas can be found in Appendix C.

Theorem 4.1. *If there exists a fully-black-box proof that a language $L \in \mathbf{ZK}$ based on the non-uniform hardness of PAC learning, then in fact $L \in \mathbf{AM} \cap \mathbf{coAM}$.*

Proof. If there were such a fully-black-box proof, then 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 for circuits relative to $\mathcal{O}$. Since zero knowledge protocol is fully-black-box by hypothesis, this means that we have a zero-knowledge protocol for L where the prover, verifier, and simulator all use only the hard concept class F, which can be implemented using just $\mathcal{R}$ (and not $\mathcal{O}$).

We claim that in fact, not only is the protocol computationally zero knowledge, it is statistically zero knowledge. Formally, applying the relativized version of the SZK/AIOWF characterization (Theorem 2.3 and Proposition 2.5) 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 an AIOWF $f^{\mathcal{O}}$ against non-uniform

circuits with a set of universally hard instances $W \subseteq \{0,1\}^*$. 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 are define simply in terms of the (code of the) simulator of the original $\mathbf{ZK}^{\mathcal{O}}$ protocol. But recall that the simulator of the original protocol needed access only to $\mathcal{R}$. Therefore, we actually 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 $\mathsf{SD}^{\mathcal{R}}$.

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 $L_{\geq n}$" if for all x of length at least 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 $\mathsf{SD}^{\mathcal{R}}$.

Lemma 4.2. *If* Red *reduces* L *to* $\mathsf{SD}^{\mathcal{R}}$ *with probability* 1 *over* $\mathcal{R}$*, then* $\Pr_{\mathcal{R}}[\mathsf{Red}$ *succeeds on* $L_{\geq n}]$ *approaches* 1 *as* $n \to \infty$*.*

This claim is elementary but we prove it for completeness in Appendix C.

The next lemma finishes the proof.

Lemma 4.3. *If there exists a constant* n_0 *such that* Red *succeeds on* $L_{\geq n_0}$ *with probability* $\geq 99/100$ *over the choice of* $\mathcal{R}$*, then in fact* $L \in \mathbf{AM} \cap \mathbf{coAM}$*.*

Red produces a pair of circuits $(X^{\mathcal{R}}, Y^{\mathcal{R}})$ with $\mathcal{R}$ gates, and we will remove the $\mathcal{R}$ gates without changing their output distributions much by hard-wiring explicit values for $\mathcal{R}$ into the circuits: we show that we only need to hard-wire a description of the oracle $\mathcal{R}$ up to inputs of length $O(\log n)$, and for longer inputs we simply set the oracle gates to output 0. Intuitively, this works because the long queries the circuit makes are unlikely to be hard instances anyway, since hard instances become very sparse. The resulting instance of SD is still good with probability 98/100 over the hard-wiring we chose (we lose a little because of the long hard instances we might simulate as 0), which means that for every x, with probability 98/100 the reduction produces a yes instance of SD if $x \in L$ and a no instance of SD if $x \notin L$. Since SD can be decided in $\mathbf{SZK}$, it and its complement can also be decided in $\mathbf{AM}$ ([For87, AH91]), which the reduction shows $L \in \mathbf{AM} \cap \mathbf{coAM}$. ■

Remark 4.4. Notice because hardwiring the oracle $\mathcal{R}$ into the circuits in the above proof incurs non-negligible error, we do *not* show that $L \in \mathbf{SZK}$: the reduction may produce an instance of SD that does not satisfy the promise of SD, and interacting with the prover on such a bad instance may reveal information. In particular, the prover might reveal to us that this instance fails the promise of SD, something the verifier cannot discover on its own.[6]

The proof of Theorem 4.1 fails to rule out semi-black-box reductions. In the above proof, we use Lemma 3.4, which in turn describes how to invert all AIOWF by using queries to $\mathcal{O}$. In contrast, in a semi-black-box reduction the adversary is allowed to 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" $\mathcal{O} = \mathbf{PSPACE}^{\mathcal{R}}_*$ inside the hard concept class itself (an idea of Simon [Sim98], see

[6] If in addition $\Pr[\mathsf{Red}$ succeeds on $L_{\geq n}] = 1 - n^{-\omega(1)}$ then we would be able to conclude that $L \in \mathbf{SZK}$ because the probability of the reduction producing a bad instance of SD is negligible. Indeed, we *can* prove such a statement for fully-black-box proofs because we can relate the simulator error to the advantage parameter in the hardness of learning. However we omit this argument here because it does not generalize to semi-black-box proofs, and in any case the weaker conclusion $L \in \mathbf{AM} \cap \mathbf{coAM}$ is still strong enough to show that such proofs are unlikely to exist.

also [RTV04]), but this must be done carefully. We have to balance two requirements: first, the inverter for the AIOWF must be able to decide $\mathsf{QBF}_*^{\mathcal{R}}$ so that we can still invert all AIOWF. On the other hand, the verifier in the zero knowledge protocol *must not* be able to decide $\mathsf{QBF}_*^{\mathcal{R}}$, 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 the inverter for the AIOWF is allowed to be *non-uniform*, while the verifier in the construction of the protocol is uniform. Finally we must ensure that the embedding procedure is well-defined, *i.e.* its definition is not circular.

Definition 4.5. Let $\mathcal{R} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ be chosen as follows: for each $z \in \{0,1\}^n$, with probability $1/2^{n/2}$ let $\mathcal{R}_z$ be a function drawn from the distribution R described below (call such z hard instances) and with probability $1 - 1/2^{n/2}$ let $\mathcal{R}_z \equiv 0$ be the constant 0 function.

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 $n/2$ bits of x are not identically zero then output a random bit. If the first $n/2$ bits of x are all 0 then let φ be the second $n/2$ 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 $\mathcal{R}_z(0^{n/2}\varphi)$ where z is a hard instance and φ is a $\mathsf{QBF}_*^{\mathcal{R}}$ that calls $\mathcal{R}_z$? We argue that this cannot happen: because $|z| = n$ and $|\varphi| = n/2$, there can be no self-reference, *i.e.* φ can never have $\mathcal{R}_z$ gates because it cannot even describe z. If φ does not call $\mathcal{R}_z$ then the oracle is well-defined since all the oracle calls made in all possible φ of length $n/2$ are independent of $\mathcal{R}_z$'s responses.

Theorem 1.3 (Restated)**.** *If there exists a semi-black-box proof that constructs a* **ZK** *protocol for a language L based on non-uniform hardness of learning, then in fact $L \in \mathbf{AM} \cap \mathbf{coAM}$.*

Proof sketch. We show that the theorem holds with probability 1 with the oracle $\mathcal{R}$ of Definition 4.5. The full proof may be found in Appendix C. First, by an argument almost identical to Lemma 3.3 we claim that $F = \{\mathcal{R}_z\}_{z \in \{0,1\}^*}$ is hard to learn for circuits relative to $\mathcal{R}$, which by the semi-black-box construction gives us that $L \in \mathbf{ZK}^{\mathcal{R}}$. Then, we claim that there exist no AIOWF relative to $\mathcal{R}$: for any function computable in time $p(n)$, we define a non-uniform inverter C with a hard instance z' of length $\mathrm{poly}(p(n))$ as advice, which it can use to query $\mathcal{R}_{z'}(0^{n'/2}\varphi)$ effectively giving it a $\mathbf{PSPACE}_*^{\mathcal{R}}$ for queries of length $\mathrm{poly}(p(n))$. C then uses this $\mathbf{PSPACE}_*^{\mathcal{R}}$ oracle and the same strategy as the inverter in Lemma 3.4 to invert f. As before, combining this with Theorem 2.3 implies that there is an efficient reduction Red that, with probability 1 over $\mathcal{R}$, reduces L to $\mathsf{SD}^{\mathcal{R}}$. Then, as before $\mathcal{R}$ succeeds on $L_{\geq n}$ with high probability as $n \to \infty$, which means that by hardwiring outputs of $\mathcal{R}$ on instances of length $O(\log n)$ directly into the $\mathsf{SD}^{\mathcal{R}}$ we obtain randomized reduction from L to SD without oracle gates. The only detail is that in order to hardwire outputs of $\mathcal{R}$ on inputs of length $O(\log n)$, we must also be able to decide $\mathsf{QBF}_*^{\mathcal{R}}$ instances of length $O(\log n)$, which we can do by brute force because the instances are so short. ∎

As a final remark, let us explain why the proof of Theorem 1.3 does *not* rule out relativizing reductions. In semi-black-box proofs, there is a single procedure that uses black-box access to $\mathcal{R}$ and produces a zero-knowledge protocol. We use this fact because 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 Lemma 4.3 would no longer hold. It is an interesting open question whether one can rule out relativizing reductions in this setting as well.

References

[AB] Sanjeev Arora and Boaz Barak. Complexity theory: A modern approach. http://www.cs.princeton.edu/theory/index.php/Compbook/Draft. Draft.

[ABX08] Benny Applebaum, Boaz Barak, and David Xiao. On basing lower-bounds for learning on worst-case assumptions. In *Proc. FOCS '08*, pages 211–220, 2008.

[AH91] W. Aiello and J. Hastad. Statistical zero-knowledge languages can be recognized in two rounds. *JCSS*, 42:327–345, 1991.

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

[BHZ87] R. B. Boppana, J. Hastad, and S. Zachos. Does co-NP have short interactive proofs? *Inf. Process. Lett.*, 25(2):127–132, 1987.

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

[GGM86] Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *Journal of the ACM*, 33(4):792–807, 1986. Preliminary version in FOCS' 84.

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

[GMW91] Oded Goldreich, Silvio Micali, and Avi 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.

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

[HHRS07] Iftach Haitner, Jonathan J. Hoch, Omer Reingold, and Gil 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.

[HKLW88] David Haussler, Michael Kearns, Nick Littlestone, and Manfred K. Warmuth. Equivalence of models for polynomial learnability. In *Proc. COLT '88*, pages 42–55, San Francisco, CA, USA, 1988. Morgan Kaufmann Publishers Inc.

[HR04] Chunyuan Hsiao and Leonid 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] Adam R. Klivans and Rocco A. Servedio. Learning dnf in time $2^{\tilde{o}(n^{1/3})}$. In *Proc. STOC '01*, pages 258–265, 2001.

[LMN93] Nathan Linial, Yishay Mansour, and Noam Nisan. Constant depth circuits, fourier transform, and learnability. *J. ACM*, 40(3):607–620, 1993.

[NOV06] Minh-Huyen Nguyen, Shien-Jin Ong, and Salil Vadhan. Statistical zero-knowledge arguments for NP from any one-way function. In *FOCS '06*, pages 3–14, 2006.

[Ost91] Rafail 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.

[OV07] Shien Jin Ong and Salil P. Vadhan. Zero knowledge and soundness are symmetric. In *EUROCRYPT '07*, pages 187–209, 2007.

[OW93] R. Ostrovsky and A. Wigderson. One-way functions are essential for non-trivial zero-knowledge. In *ISTCS '93*, pages 3–17, 1993.

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

[PW90] Leonard Pitt and Manfred 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 *1st Theory of Cryptography Conference*, pages 1–20, 2004.

[Sim98] Daniel 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.

[SV97] Amit Sahai and Salil P. Vadhan. A complete promise problem for statistical zero-knowledge. In *Proc. FOCS '97*, pages 448–457, 1997.

[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.

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.* [AB]). 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 **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).

■

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

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 $\mathsf{QBF}^{\mathcal{O}}$, 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 $i \xleftarrow{R} [s]$ 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. ■

Second, $\mathbf{PSPACE}^{\mathcal{O}}$ is able to find "heavy" outputs of a $\mathbf{PSPACE}^{\mathcal{O}}$ computation, say over the uniform distribution of inputs (in the following, think of D as being the equality predicate; we will state it more generally because of how we use it in our proofs):

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*

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

Proof. Since $\mathbf{PSPACE}^{\mathcal{O}}$ is capable of counting $\mathbf{PSPACE}^{\mathcal{O}}$ relations, A simply iterates over all $y \in \{0,1\}^n$ 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 $\max_{u \in U} \Pr[D = u] \leq 1/k$ (equivalently, it has min-entropy $H_\infty(D) \geq \log k$). 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. By the guarantee on min-entropy, we know that for each $u \in U$, $\Pr[D = u] \leq 1/k$ and the support is large, $|\mathrm{supp}(D)| \geq k$. We derive that for any positive constant t:

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

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

B Proofs of lemmas of Section 3

First, let us recall the oracle $\mathcal{O}$.

Definition 3.1 (Restated)**.** We define the distribution over oracles $\mathcal{O}$ as follows. First, 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.

B.1 Proof that non-uniform hardness of learning holds

Lemma 3.3 (Restated)**.** *With probability 1 over the choice of $\mathcal{R}$, the concept class $F = \{\mathcal{R}_z\}_{z\in\{0,1\}^*}$ is hard to learn by any efficient oracle machine with access to $\mathcal{O}$.*

Proof. Fix n and any circuit C of size $p(n) = \mathrm{poly}(n)$. We calculate the probability that C learns $\mathcal{R}_z$ *for every* $z \in \{0,1\}^n$.

Let $S_z = \{(x_1, \mathcal{R}(z,x_1)), \ldots, (x_{p(n)-1}, \mathcal{R}(z,x_{p(n)}))\}$ where the $x_i \xleftarrow{R} U_n$, and let $x \xleftarrow{R} U_n$ be the example C must label.

Claim B.1. *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 $2^{O(p(n)\log(p(n)))}$ circuits of size $p(n)$ for any $p(n) = \mathrm{poly}(n)$ shows that the probability of there existing any circuit learning all the $\mathcal{R}_z$ is still $2^{-2^{\Omega(n)}}$.

We say that C queries $\mathcal{R}_z$ if it asks $\mathcal{O}$ a formula φ that contains a $\mathcal{R}_z$ gate. We will show that the probability C learns $\mathcal{R}_z$ without querying $\mathcal{R}_z$ is small because the function $\mathcal{R}_z$ is random, and then we will show that the probability of C querying z given only $p(n)$ queries to the oracle is small because the output of $\mathcal{R}_z$ contains 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 $\Pr[C^{\mathcal{O}}(S_z)(x) \text{ queries } \mathcal{R}_z)] > \varepsilon^4$

We develop the LHS of Claim B.1

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

$$\le \Pr_{\mathcal{O}}\left[\bigwedge_{z \text{ hard}} (A_z^\varepsilon \vee B_z^{\varepsilon^4})\right] \tag{B.2}$$

$$\le \Pr_{\mathcal{O}}\left[\exists z \text{ hard},\ A_z^\varepsilon \wedge \overline{B_z^{\varepsilon^4}}\right] + \Pr_{\mathcal{O}}\left[\bigwedge_{z \text{ hard}} B_z^{\varepsilon^4}\right] \tag{B.3}$$

This formalizes our above intuition, since the first term is the probability that, for some $z \in \{0,1\}^n$, $C^{\mathcal{O}}$ learns $\mathcal{R}(z,x)$ without querying it, and the second term is the probability that $C^{\mathcal{O}}(S_z, x)$ queries $\mathcal{R}_z$ on all the hard z.

Bounding the first term of Inequality B.3. Fix a hard z (of which there are at most 2^n). We want to bound the quantity

$$\Pr_{\mathcal{O}}\left[\left\{\Pr_{S,x}[C^{\mathcal{O}}(S_z,x) = \mathcal{R}(z,x)] > \tfrac{1+\varepsilon}{2}\right\} \wedge \left\{\Pr_{S,x}[C^{\mathcal{O}}(S_z,x) \text{ queries } \mathcal{R}_z] \le \varepsilon^4\right\}\right]$$
$$= \mathbb{E}_{\mathcal{R}'} \Pr_{\mathcal{O}}\left[\left\{\Pr_{S,x}[C^{\mathcal{O}|\mathcal{R}'}(S_z,x) = \mathcal{R}(z,x)] > \tfrac{1+\varepsilon}{2}\right\} \wedge \left\{\Pr_{S,x}[C^{\mathcal{O}|\mathcal{R}'}(S_z,x) \text{ queries } \mathcal{R}_z] \le \varepsilon^4\right\}\right]$$

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 before except with $\mathcal{R}'$ replacing the fully random $\mathcal{R}$. To bound this last probability, we will count the number of functions that $C^{\mathcal{O}}$ can possibly learn without querying $\mathcal{R}(z,x)$ too often and then show that it is a small fraction of all functions that $\mathcal{R}(z,x)$ can be.

Let us model this problem abstractly, since we will use a similar idea in the proof of the semi-black-box case. What we want to show is that a random function is hard to learn if we do not query the random function with high probability. Think of $\mathcal{R}_z$ as a random function $\mathcal{F}$, and think of $C^{\mathcal{O}|\mathcal{R}'}$ as a computationally-unbounded procedure A with access to a $\mathcal{F}$ oracle. The above is simply the probability that A learns $\mathcal{F}$ without querying $\mathcal{F}$ too often, which cannot happen with high probability over $\mathcal{F}$. We state this formally:

Lemma B.2. *Let $\mathcal{F} : \{0,1\}^n \to \{0,1\}$ be a random function, and let $A^{\mathcal{F}}$ be a computationally unbounded procedure with oracle access to $\mathcal{F}$. Let $S = ((x_1, \mathcal{F}(x_1)), \ldots, (x_p, \mathcal{F}(x_p)))$ be a set of labelled examples where $p = p(n) = \text{poly}(n)$, the $x_i \xleftarrow{R} U_n$, and let $x \xleftarrow{R} U_n$ be an unlabelled*

example. Let $\varepsilon = 2^{\log^2 n}$. *Then:*

$$\Pr_{\mathcal{F}}\left[\left\{\Pr_{S,x}[A^{\mathcal{F}}(S,x) = \mathcal{F}(x)] > \tfrac{1+\varepsilon}{2}\right\} \wedge \left\{\Pr_{S,x}[A^{\mathcal{F}}(S,x) \text{ queries } \mathcal{F}] < \varepsilon^4\right\}\right] < 2^{-2^{\Omega(n)}}$$

This lemma and a union bound over all hard instances z bounds the first term of Inequality B.3.

Proof of Lemma B.2. If A never queried $\mathcal{F}$, then the number of functions that $C^{\mathcal{O}}$ can possibly learn is bounded by the number of possible inputs (*i.e.* labelled examples), which is at most $2^{p(n)(n+1)}$, which is a negligible fraction of the 2^{2^n} possible functions $\mathcal{F}$ could be.

Now consider A that can query $\mathcal{F}$ but only with probability at most ε^4 over random S, x. This means by Markov that

$$\Pr_{S}\left[\Pr_{x}[A^{\mathcal{F}}(S,x) \text{ queries } \mathcal{F}(x)] \geq 4\varepsilon^3\right] < \varepsilon/4$$

Meanwhile, we also have by averaging that

$$\Pr_{S}[\Pr_{x}[A^{\mathcal{F}}(S,x) = \mathcal{F}(x)] > \tfrac{1+\varepsilon/2}{2}] > \varepsilon/4$$

This means that there must exist some fixed S' such that both events $\Pr_x[A^{\mathcal{F}}(S',x) = \mathcal{F}(x)] > \frac{1+\varepsilon/2}{2}$ and $\Pr_x[A^{\mathcal{F}}(S',x) \text{ queries } \mathcal{F}] < 4\varepsilon^3$ occur. Thus, the string S' plus an explicit labelling of all $4\varepsilon^3 2^n$ points x where the circuit queries $\mathcal{F}$ gives us a description of $\mathcal{F}$ that is accurate up to relative distance $\frac{1-\varepsilon/2}{2}$; call this the noisy description of $\mathcal{F}$. It is easy to see by Chernoff that the number of vectors of length 2^n of relative weight at most $\frac{1-\varepsilon/2}{2}$ is at most $2^{2^n - \Omega(\varepsilon^2 2^n)}$. Therefore, every function $\mathcal{F}$ that A is able to learn can be specified by first giving the noisy description of $\mathcal{F}$ and then giving the low-weight vector that equals the difference between the noisy description and the true function. This means that A can learn at most $2^{p(n)(n+1)+4\varepsilon^3(n+1)2^n+2^n-\Omega(\varepsilon^2 2^n)}$ different functions, and therefore

$$\begin{aligned}
&\Pr_{\mathcal{F}}\left[\left\{\Pr_{S,x}[A^{\mathcal{F}}(S,x) = \mathcal{F}(x)] > \tfrac{1+\varepsilon}{2}\right\} \wedge \left\{\Pr_{S,x}[A^{\mathcal{F}}(S,x) \text{ queries } \mathcal{F}] \leq \varepsilon^2\right\}\right] \\
&\quad \leq 2^{p(n)(n+1)+2\varepsilon^3(n+1)2^n+2^n-\Omega(\varepsilon^2 2^n)-2^n} \\
&\quad = 2^{-2^{\Omega(n)}}
\end{aligned}$$

where in the last line we use that $\varepsilon = 2^{-\log^2 n}$. ∎

Bounding the second term of Inequality B.3. We will show that if the learner C can query $\mathcal{R}_z$ with noticeable probability given a random S, x, it solves the following inversion problem, and then we show that no algorithm can solve the inversion problem with only polynomial number of queries to the oracle.

Definition B.3. The D-inversion problem for a family of distributions $D = \{D_n\}$ over strings $\{0,1\}^{2^n}$ is defined as follows. Let $\mathcal{F}$ denote the distribution of over functions $\{0,1\}^n \to \{0,1\}^{2^n}$ where, for each $z \in \{0,1\}^n$, with probability $2^{-n/2}$ we set $\mathcal{F}(z)$ to be a string sampled from D_n, and with probability $1 - 2^{-n/2}$ we set $\mathcal{F}(z) = 0^{2^n}$. We say that a (computationally unbounded) oracle procedure $I^{\mathcal{F}}$ solves the R-inversion problem with q queries if for every $y \in \{0,1\}^{2^n}$ such that $\mathcal{F}^{-1}(y) \neq \varnothing$, we have $I^{\mathcal{F}}(y) \in \mathcal{F}^{-1}(y)$ and I makes at most q queries to $\mathcal{F}$.

To apply this to our setting, we will show that if $C^{\mathcal{O}}(S_z)(x)$ is able to query $\mathcal{R}_z$ with probability $\geq \varepsilon^4$ over S_z, x, then in fact there is a deterministic procedure that solves the U_{2^n}-inversion problem making only $O(p(n)n/\varepsilon^4)$ queries. Then we show this is impossible with high probability.

First let us show that the event $\bigwedge_{z \text{ hard}} B_z^{\varepsilon^4}$ implies one can solve the U_{2^n}-inversion problem using $O(p(n)n/\varepsilon^4)$ queries. Notice that the function $\mathcal{R}(z) = \mathcal{R}_z$, where we interpret the function $\mathcal{R}_z$ as its truth table, is exactly the distribution of $\mathcal{F}$ in the U_{2^n}-inversion problem, and if C queries $\mathcal{R}_z$ then in particular it finds z.

First we describe a randomized procedure I' for solving the U_{2^n}-inversion problem, then we show that there exists a way to fix the random coin tosses to obtain a deterministic I. I' is defined using the learning circuit C as follows: simulate C using independent randomness $O(n/\varepsilon^4)$ times; for each query 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, I' will query $\mathcal{F}$ to get $\mathcal{F}(z)$ which it uses as the truth table of $\mathcal{R}_z$. For every $z' \in Z$ where $|z'| = n' \neq n$, I' sets $\mathcal{R}_{z'}$ using independent coin tosses to be $0^{2^{n'}}$ with probability $1 - 2^{-n'/2}$ and to be a random string $U_{2^{n'}}$ with probability $2^{-n'/2}$. Then I' decides the QBF formula φ using these truth tables (I' 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 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 simulations will query $\mathcal{R}_z$, and so I' will query $\mathcal{F}(z)$; I will simply check each query z whether $\mathcal{F}(z)$ matches its input, and if so it will halt and output z. Now take a union bound over all possible inputs $y = \mathcal{F}(z)$ of which there are at most 2^n, still with probability $1 - 2^{-n}$ the random bits used are simultaneously good for all y; fix any such a good choice of the random bits and let that be our inverting deterministic inverting procedure I.

It therefore follows that

$$\Pr_{\mathcal{O}}\left[\bigwedge_{z \text{ hard}} B_z^{\varepsilon^4}\right] \leq \Pr_{\mathcal{R}}\left[I \text{ inverts } \mathcal{R} \text{ and makes at most } O(p(n)n/\varepsilon^4) \text{ queries}\right]$$

Since $\mathcal{R}$ is distributed as $\mathcal{F}$ in the U_{2^n}-inversion problem, the following lemma concludes the bound on the second term of Inequality B.3.

Lemma B.4. *For any $I^{\mathcal{F}}$, the probability it solves U_{2^n}-inversion problem with $O(p(n)n/\varepsilon^4)$ queries over the choice of $\mathcal{F}$ is at most $2^{-2^{\Omega(n)}}$.*

Proof. Fix any oracle procedure $I^{\mathcal{F}}$ making at most $O(p(n)n/\varepsilon^4)$ to $\mathcal{F}$. Let $N = |\{x \mid \mathcal{F}(x) \overset{R}{\leftarrow} U_{2^n}\}|$ denote the number of hard outputs of $\mathcal{F}$; 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:

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

We will further throw out the oracles $\mathcal{F}$ that are not injective (this occurs with probability at most $\leq \binom{N}{2} 2^{-2^n}$) and $\mathcal{F}$ where one of the hard instances $\mathcal{F}(x) = U_{2^n}$ samples the all zero string (this occurs with probability at most $N2^{-2^n}$). We call $\mathcal{F}$ where neither of these conditions hold "good". Therefore our bound is now:

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

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

To bound the probability on the RHS, we show that I 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 $\mathcal{F}$ with N hard outputs as follows: we will keep track of a set $Y \subseteq \{0,1\}^{2^n}$ of "easily describable outputs" y for which we will be able to compute the preimage $x = \mathcal{F}^{-1}(y)$ with very little information using I. For the "hard-to-describe outputs" outside Y we will just explicitly record the function. We show that this is sufficient for reconstructing any $\mathcal{F}$ that I 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 $\mathcal{F}$, define Y constructively as follows. Initialize $Y = \varnothing$ and the set $T \subseteq \{0,1\}^{2^n}$ to be the image of the hard instances of $\mathcal{F}$, namely $t \in T$ iff $t = \mathcal{F}(z) \neq 0$ for some hard instance z. Since we are conditioning on good $\mathcal{F}$, 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 $I^{\mathcal{F}}(t)$ and record the queries $x_1, \ldots, x_m$ (in the order that I makes them) that I makes to $\mathcal{F}$, where $m = O(p(n)n/\varepsilon^4)$. If none of the x_i satisfy $\mathcal{F}(x_i) = t$, then remove all of the $x_1, \ldots, x_m$ from T. If some x_i satisfies $\mathcal{F}(x_i) = y$, 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 $Z = \mathcal{F}^{-1}(T) \subseteq \{0,1\}^n$ (which is of size N), the set Y, the preimage of Y which we call $X = \mathcal{F}^{-1}(Y) \subseteq Z$, and the explicit values of $\mathcal{F}$ on all inputs $x \in Z \setminus X$, we can completely reconstruct $\mathcal{F}$ as follows. For each $x \notin Z$, $\mathcal{F}(x) = 0^{2^n}$. 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 $I^{\mathcal{F}}(y)$. The queries $I^{\mathcal{F}}(y)$ makes to $\mathcal{F}$ will all either be for $x \notin X$ in which case we know the answer explicitly, for $x \in X$ such that $\mathcal{F}(x)$ 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 $x = \mathcal{F}^{-1}(y)$. Either way, we obtain the value $\mathcal{F}^{-1}(y)$.

The number of functions describable in this way is exactly

$$\binom{2^n}{N}\binom{N}{|Y|}\binom{2^{2^n}-1}{|Y|} \cdot \frac{(2^{2^n}-1-|Y|)!}{(2^{2^n}-N)!}$$

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 and never maps to 0^{2^n}. Therefore, the probability over $\mathcal{F}$ that I inverts $\mathcal{F}$ is exactly the above quantity divided by the total number

of good $\mathcal{F}$, namely $\binom{2^n}{N}\frac{(2^{2^n-1})!}{(2^{2^n-1}-N)!}$. So we can calculate that:

$$\Pr_{\mathcal{F}\text{ injective}}[I \text{ inverts } \mathcal{F} \text{ everywhere} \mid N \text{ hard instances}] \leq \frac{\binom{2^n}{N}\binom{N}{|Y|}\binom{2^{2^n-1}}{|Y|}\cdot\frac{(2^{2^n-1}-|Y|)!}{(2^{2^n-1}-N)!}}{\binom{2^n}{N}\frac{(2^{2^n-1})!}{(2^{2^n-1}-N)!}} \tag{B.4}$$

$$= \frac{\binom{N}{|Y|}}{|Y|!} \tag{B.5}$$

$$\leq \left(\frac{N3e}{|Y|^2}\right)^{|Y|} \tag{B.6}$$

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

To conclude the proof of Lemma 3.3, notice that since both terms of Inequality B.3 are bounded by $2^{-2^{\Omega(n)}}$, so is their sum. ■

B.2 AIOWF do not exist

Lemma 3.4 (Restated)**.** *With probability* 1 *over choice of* $\mathcal{O}$ *as in Definition 3.1, it holds that for any efficient oracle algorithm* f *computing a function* $f^{\mathcal{O}} : \{0,1\}^n \times \{0,1\}^{\ell(n)} \to \{0,1\}^{m(n)}$*,* $f^{\mathcal{O}}$ *can be inverted by an efficient oracle algorithm* A *on every auxiliary input* w*. Namely, for every* $w \in \{0,1\}^n$

$$\Pr_{x \xleftarrow{R} \{0,1\}^{\ell(n)}}[A^{\mathcal{O}}(w, y) \in (f_w^{\mathcal{O}})^{-1}(y) \mid f_w^{\mathcal{O}}(x) = y] > 1/2$$

Proof. Recall that a query to $\mathcal{O}$ is a $\mathsf{QBF}_*^{\mathcal{R}}$ formula φ, which is a QBF formula with $\mathcal{R}_z$ gates for explicit strings z, and $\mathsf{QBF}_*^{\mathcal{R}}$ complete for $\mathbf{PSPACE}_*^{\mathcal{R}}$. We say that z is a heavy query if f queries $\mathcal{O}$ with a formula φ containing a $\mathcal{R}_z$ gate with noticeable probability over $x \xleftarrow{R} U_\ell$. We show that by finding all the heavy z, A is also able to find most of the hard instances that f finds, and this allows A to invert f. The key is that unlike in the case of learning, A knows the description of f_w.

We describe and analyze an algorithm A such that for every efficiently computable oracle function $f^{\mathcal{O}} : \{0,1\}^n \times \{0,1\}^{\ell(n)} \to \{0,1\}^{m(n)}$, for every large enough n, with probability $1 - 2^{-n}$ over the choice of $\mathcal{O}$, $A^{\mathcal{O}}$ takes as input a description of f, $w \in \{0,1\}^n$, and $y \in \{0,1\}^m$, and guarantees that for all $w \in \{0,1\}^n$,

$$\Pr_{x \in U_\ell}[A^{\mathcal{O}}(y) \in (f_w^{\mathcal{O}})^{-1}(y) \mid y = f_w^{\mathcal{O}}(x)] > 1/2 \tag{B.7}$$

This inequality proves the lemma, since by the Borel-Cantelli lemma this means $A^{\mathcal{O}}$ inverts $f^{\mathcal{O}}$ on all but a finite number of input lengths n with probability 1 over $\mathcal{O}$, and since there are only a countable number of f this means $A^{\mathcal{O}}$ can invert all efficient f with probability 1 over $\mathcal{O}$.

We turn to proving Inequality B.7. Since f is efficiently computable with oracle queries, let C be the circuit computing f_w (with $\mathcal{O}$ gates) on inputs of length $\ell(n)$, and suppose $|C| \leq p = p(n)$. Let $g_1, \ldots, g_p$ be the oracle gates of C in topologically sorted order. Let D be the circuit taking inputs φ, z and outputting 1 if φ contains a $\mathcal{R}_z$ gate, and outputs 0 otherwise.

Set the heaviness threshold to be $\alpha = 100np^6$. In sorted order, A finds all z such that $C^{\mathcal{O}}(U_\ell)$ queries $\mathcal{O}$ with a formula containing a $\mathcal{R}_z$ gate with probability larger than $1/\alpha$ using the following procedure.

First, A initializes the set $Z_0 = \{z \mid |z| \leq 8 \log p\}$. Then, to construct Z_i, the set of heavy queries up till the i'th query, using Z_{i-1}, A 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 other oracle gates 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 pass this modified formula to $\mathcal{O}$. This transformation forces all the hard instances that φ 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 query g_j containing only heavy instances ($z \in Z_j$) or easy instances ($\mathcal{R}_z \equiv 0$), and $D(\varphi, z) = 1$ means exactly that φ contains a $\mathcal{R}_z$ gate. Since Q_i only makes oracle queries containing $\mathcal{R}_z$ gates for $z \in Z_{i-1}$, and since A knows Z_{i-1}, it can simulate 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. This means it can invoke Proposition A.3 with input $(Q_i, D, 1^\alpha)$ to get the set $\{z \mid \Pr[Q_i(x) = \varphi \wedge D(\varphi, z) = 1] > 1/\alpha\}$, which we add to Z_{i-1} to obtain Z_i.

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

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

$$\Pr_{\mathcal{R}}\left[\Pr_{x \xleftarrow{R} U_\ell}[B^w(x)] > \tfrac{1}{p(n)}\right] \leq 2^{-2n} \tag{B.8}$$

First we use this inequality to prove the lemma: by a union bound over all w, this means that for a $1 - 2^{-n}$ fraction of the $\mathcal{R}$ that with probability $1 - p$ over x, $f_w^{\mathcal{O}}(x)$ never queries hard $z \notin Z_p$. But in this case we can replace $\mathcal{O}$ by an oracle that only has access to the hard instances in Z_p. Namely, let $\mathcal{R}'$ be the oracle where $\mathcal{R}'(z, x) = \mathcal{R}(z, x)$ for all $z \in Z_p$ and is 0 elsewhere, and we have that $\Delta\left((x, f_w^{\mathcal{O}}(x)), (x, f_w^{\mathbf{PSPACE}^{\mathcal{R}'}}(x))\right) \leq 1/p$. Furthermore, since A knows Z_p it can use Z_p and $\mathcal{O}$ to simulate $\mathbf{PSPACE}^{\mathcal{R}'}$, so it can use Proposition A.2 to compute uniformly random preimages of $f_w^{\mathbf{PSPACE}^{\mathcal{R}'}}$ with failure probability 2^{-m}, giving us

$$\Delta\left((x, f_w^{\mathbf{PSPACE}^{\mathcal{R}'}}(x)), \quad (A^{\mathcal{O}}(y), y \mid y = f_w^{\mathbf{PSPACE}^{\mathcal{R}'}}(x))\right) \leq 2^{-m}$$

Putting the two together by the triangle inequality, we have

$$\Delta((x, f_w^{\mathcal{O}}(x)), (A^{\mathcal{O}}(y), y \mid y = f_w^{\mathcal{O}}(x))) \leq 1/p + 2^{-m}$$

which proves the lemma modulo Inequality B.8. In fact, we prove something much better: $A^{\mathcal{O}}$ actually gives an almost uniformly random preimage of y.

It remains to prove Inequality B.8. We fix w and remove it from the notation, letting $B(x) = B^w(x)$. 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 $B(x) \subseteq \bigcup_{i=1}^{p} B_i(x)$.

By averaging:

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

We claim that for each i, $\Pr_{\mathcal{R}}[\Pr_x[B_i(x)] > 1/p^2] \le 2^{-3n}$, which we prove using a case analysis. Showing this concludes the proof of the lemma since $p(n)2^{-3n} \le 2^{-2n}$.

The case analysis roughly goes as follows: either the probability that Q_i makes a light 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 Q_i makes a light 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 $\mathsf{NotInZ}_i(x)$ be the event that $f^{\mathcal{O}}$'s i'th query is not in Z_i conditioned on all queries $j < i$ being either in Z_j or easy. (The only difference between NotInZ_i and B_i is that in B_i we also demand the query be hard.) We have that

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

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

To bound the first term, notice that we can inductively fix $\mathcal{R}$ up until the i'th query as follows: let $\mathcal{R}_0$ be a fixing of all the values of the oracle with auxiliary input in Z_0. Let $\mathcal{R}_1$ be a fixing of all the responses of the oracle for queries in Z_1 conditioned on $\mathcal{R}_0$. Inductively, let $\mathcal{R}_i$ be a fixing of all the responses of the oracle for queries in Z_i conditioned on $\mathcal{R}_{i-1}$ and the event that $f^{\mathcal{O}}(x)$'s first $i-1$ queries are either easy or in Z_{i-1}. Thus, we have:

$$\begin{aligned}&\Pr_{\mathcal{R}}\left[\left\{\Pr_x[B_i(x)] > 1/p^2\right\} \wedge \left\{\Pr_x[\mathsf{NotInZ}_i(x)] \ge 1/p^2\right\}\right] \\ &= \mathbb{E}_{\mathcal{R}_{i-1}} \Pr_{\mathcal{R}}\left[\left\{\Pr_x[B_i(x)] > 1/p^2\right\} \wedge \left\{\Pr_x[\mathsf{NotInZ}_i(x)] \ge 1/p^2\right\} \mid \mathcal{R}_{i-1}\right] \\ &\le \mathbb{E}_{\mathcal{R}_{i-1}} \Pr_{\mathcal{R}}\left[\left\{\Pr_x[B_i(x) \mid \mathsf{NotInZ}_i(x)] > 1/p^2\right\} \mid \left\{\Pr_x[\mathsf{NotInZ}_i(x)] \ge 1/p^2\right\} \wedge \mathcal{R}_{i-1}\right]\end{aligned}$$

where in the last line we used the fact that $B_i(x)$ implies $\mathsf{NotInZ}_i(x)$. Now for each such fixing of $\mathcal{R}_{i-1}$, notice that because each individual light i'th query z has probability at most $1/(100np^6)$ and the probability that the i'th query is light is at least $1/p^2$, the probability that the i'th query is z conditioned on $\mathsf{NotInZ}_i(x)$ is at most $1/(100np^4)$. Each i'th query is hard independently with probability at most $1/p^4$ over the choice of oracle (because Z_0 contains all queries of length up to $8\log p$, the oracle is random only on longer inputs), so by a generalized Chernoff bound (Lemma A.4) (the universe is $\{0,1\}^n$, D is the distribution of the i'th query conditioned on $\mathsf{NotInZ}_i(x)$, and F is the choice of hard instances), the probability that a larger than $1/p^2$ fraction of the queries not in Z are hard is 2^{-3n}.

■

We now show that in fact we can rule out relativizing reductions but also $\forall\exists$ semi-black-box reductions. Such a reduction guarantees that for every concept class F that is hard to learn with non-negligible advantage, there exists an oracle algorithm f^F (the algorithm itself that can depend on F) such that for every efficient inverting algorithm A^F, there exists an efficient family of oracle circuits C_n^F learning learning F. Theorem 1.2 already rules out such proofs because since $F = \{\mathcal{R}_z\}$ is hard to learn, in particular so is $\mathsf{QBF}_*^{\mathcal{R}}$. Since $\mathcal{O}$ is a $\mathsf{QBF}_*^{\mathcal{R}}$ oracle, this means we have a concept class that is hard to learn but relative to which AIOWF do not exist.

Theorem B.5. *There exists no $\forall\exists$ semi-black-box reduction from non-uniform hardness of learning to AIOWF.*

C Proofs of lemmas of Section 4

Lemma 4.2 (Restated)**.** *If* Red *reduces L to $\mathsf{SD}^{\mathcal{R}}$ with probability 1 over $\mathcal{R}$, then $\Pr_{\mathcal{R}}[\mathsf{Red}$ succeeds on $L_{\geq n}]$ approaches 1 as $n \to \infty$.*

Proof. Let A_n be the event that Red succeeds on $L_{\geq n}$. We know by hypothesis that $1 = \Pr_{\mathcal{R}}[\mathsf{Red}$ reduces L to $L\mathsf{SD}^{\mathcal{R}}] \leq \Pr_{\mathcal{R}}[\bigcup_{i=1}^{\infty} A_i]$. Since $A_n \subseteq A_{n+1}$, we have that:

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

therefore it follows that $\Pr[A_n] \to \Pr[\bigcup_{i=1}^{\infty} A_i] = 1$ as $n \to \infty$. ■

Lemma 4.3 (Restated)**.** *If there exists a constant n_0 such that* Red *succeeds on $L_{\geq n_0}$ with probability $\geq 99/100$ over the choice of $\mathcal{R}$, then in fact $L \in \mathbf{AM} \cap \mathbf{coAM}$.*

Proof. We have by hypothesis that Red that efficiently maps each input x to a pair of oracle circuits $(X^{\mathcal{R}}, Y^{\mathcal{R}})$ such that, with probability 99/100 over the choice of $\mathcal{R}$, for every $x \in L, |x| \geq n_0$ reduces to circuits such that $\Delta(X^{\mathcal{R}}, Y^{\mathcal{R}}) > 99/100$ and each $x \notin L, |x| \geq n_0$ reduces to circuits such that $\Delta(X^{\mathcal{R}}, Y^{\mathcal{R}}) < 1/100$ (we identify the circuit $X^{\mathcal{R}}$ with the output distribution of $X^{\mathcal{R}}$ on uniform input).

We describe the following efficient randomized reduction Red' that produces circuits (without oracle gates) (X', Y') and claim that for every x, with probability 98/100, $x \in L$ reduces to circuits satisfying $\Delta(X', Y') > 98/100$ and $x \notin L$ reduces to circuits satisfying $\Delta(X', Y') < 2/100$. Red' runs Red to produce $(X^{\mathcal{R}}, Y^{\mathcal{R}})$, and then flips its own random coins to generate a "fake" $\mathcal{R}'$ for inputs of up to length $6 \log p$ where $p \geq \max\{|X^{\mathcal{R}}|, |Y^{\mathcal{R}}|\}$. This can clearly be done in polynomial time. Red' then hardwires this fake $\mathcal{R}'$ into $X^{\mathcal{R}}, Y^{\mathcal{R}}$ to obtain X', Y'.

We prove that Red' satisfies the claim. Let $B(r)$ be the bad event that $X^{\mathcal{R}}(r)$ queries a hard instance z of length $> 6 \log p$, and $B_i(r)$ be the event that the i'th oracle query of $X^{\mathcal{R}}(r)$ queries a hard instance z of length $> 6 \log p$. We see that:

$$\Pr_{\mathcal{R}, r \xleftarrow{R} U_\ell}[B(r)] \quad = \quad \mathbb{E}_r \Pr_{\mathcal{R}}[B(r)] \quad \leq \quad \mathbb{E}_r \sum_{i=1}^{p} \Pr_{\mathcal{R}}[B_i(r)] \quad \leq \quad 1/p^2$$

since over the randomness of $\mathcal{R}$, the probability that any query of length $> 6 \log p$ is hard is at most $1/p^3$.

Now by Markov, we have that $\Pr_{\mathcal{R}}[\Pr_{r \xleftarrow{R} U_\ell}[B(r)] > 1/p] < 1/p$. Let $\mathcal{R}'$ be identical to $\mathcal{R}$ for all inputs of length $\leq 6 \log p$ and zero on longer inputs. Notice that for good $\mathcal{R}$ where $B(r)$ occurs with probability $\leq 1/p$, we have that $\Delta(X^{\mathcal{R}}, X^{\mathcal{R}'}) \leq 1/p$. But constructing the distribution $X^{\mathcal{R}'}$ for a random $\mathcal{R}'$, is exactly the same as the distribution of X' constructed by Red'!

Therefore, for each x with probability $99/100 - 1/p \geq 98/100$ we get a pair of circuits (X', Y') satisfying $\Delta(X', Y') \geq 98/100$ for $x \in L$ and $\Delta(X', Y') \leq 2/100$ for $x \notin L$. This is an instance of SD (with slightly different parameters, but which is still in **SZK**).

Finally, SD and $\overline{SD}$ can both be decided by **AM** protocols [AH91, For87], so this puts $L \in \mathbf{AM} \cap \mathbf{coAM}$. ■

Using the above techniques we can also prove Theorem 1.3.

Theorem 1.3 (Restated)**.** *If there exists a semi-black-box proof that constructs a* **ZK** *protocol for a language L based on non-uniform hardness of learning, then in fact $L \in \mathbf{AM} \cap \mathbf{coAM}$.*

Proof. Recall the oracle we use: $\mathcal{R} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ is chosen such that for each $z \in \{0,1\}^n$, with probability $1/2^{n/2}$, $\mathcal{R}_z$ is a function drawn from the distribution R described below (call such z hard instances) and with probability $1 - 1/2^{n/2}$ we set $\mathcal{R}_z \equiv 0$. The overall oracle is just $\mathcal{R}$ (rather than an oracle for $\mathsf{QBF}_*^{\mathcal{R}}$, as in Theorem 4.1).

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 $n/2$ bits of x are not identically zero then output a random bit. If the first $n/2$ bits of x are all 0 then let φ be the second $n/2$ bits of x and interpret φ as a $\mathsf{QBF}_*^{\mathcal{R}}$ formula, and output whether φ is satisfiable. We already argued that such $\mathcal{R}$ is well-defined.

First we show that $F = \{\mathcal{R}_z\}_{z \in \{0,1\}^*}$ is hard to learn with advantage $\varepsilon = 2^{-\log^2 n}$ even for non-uniform circuits.

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

Therefore the semi-black-box proof gives us that $L \in \mathbf{ZK}^{\mathcal{R}}$.

Second, we show that AIOWF do not exist relative to $\mathcal{R}$.

Claim C.2. *With probability* 1 *over $\mathcal{R}$, there exist no AIOWF against uniform adversaries relative to $\mathcal{R}$.*

Using Theorem 2.3 and the fact that AIOWF do not exist, we deduce that there is an efficient reduction Red such that with probability 1 over $\mathcal{R}$, Red reduces L to $\mathsf{SD}^{\mathcal{R}}$.

By Lemma 4.2 there exists n_0 such that Red succeeds on $L_{\geq n_0}$ with probability $99/100$ over the choice of $\mathcal{R}$, and then we can use the reduction to SD to place $L \in \mathbf{AM} \cap \mathbf{coAM}$:

Claim C.3. *If for some constant n_0 the reduction Red succeeds on $L_{\geq n_0}$ with probability $99/100$ over the choice of $\mathcal{R}$, then $L \in \mathbf{AM} \cap \mathbf{coAM}$.*

■

Proof of Claim C.1. 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 $\Pr_{S,x}[C^{\mathcal{R}}(S,x) \text{ queries } \mathcal{R}_z] > \varepsilon^4$. Here, "$C^{\mathcal{R}}(S,x)$ queries $\mathcal{R}_z$" means $C^{\mathcal{R}}$ makes a query $\mathcal{R}(z,x)$

for some x, or $C^{\mathcal{R}}$ queries $\mathcal{R}(z', 0^{n'/2}\varphi)$ where $|z'| = n' > 2|z|$ and φ is a $\mathsf{QBF}_*^{\mathcal{R}}$ formula containing a $\mathcal{R}_z$ gate.

We have that

$$\Pr_{\mathcal{F}}[C^{\mathcal{R}} \text{ learns } F \text{ on length } n] \leq \Pr_{\mathcal{F}}\left[\exists z \text{ hard of length } n,\ A_z^{\varepsilon} \wedge \overline{B_z^{\varepsilon^4}}\right] + \Pr_{\mathcal{F}}\left[\bigwedge_{z \text{ hard}} B_z^{\varepsilon^4}\right] \quad \text{(C.1)}$$

To bound the first term, 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 $\mathcal{R}(z', 0^{n'/2}\varphi)$ where $|z'| = n' > 2|z|$ 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 $x = 0^{n/2}\varphi$. But the probability that $C^{\mathcal{R}'}$ will be challenged with such a x is $2^{-n/2}$, which means such x contribute a negligible to $C^{\mathcal{R}'}$'s advantage. Therefore, we we can still apply Lemma B.2 to bound the first term by $2^{-2^{\Omega(n)}}$.

To bound the second term, 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 I that solves the R-inversion problem making $O(p(n)n/\varepsilon^4)$ queries (defined in Definition B.3; notice here also that R is the distribution defined in Definition 4.5, not U_{2^n}). We omit this transformation, which is identical to the previous one, except to point out that I can sample $R^{(n')}$ for all lengths $n' < n$ by itself because it is unbounded and so can decide $\mathsf{QBF}_*^{\mathcal{R}}$ instances with $\mathcal{R}_{z'}$ gates where $|z'| < n$ without querying $\mathcal{F}$. Therefore, it suffices to show that the R-inversion procedure cannot be solved with $O(p(n)n/\varepsilon^4)$ queries. But again this follows almost identically to the proof of Lemma B.4. The two differences are first that $R^{(n)}$ depends on $R^{(n')}$ for $n' < n$, which can be safely ignored by conditioning on any setting of $\mathcal{F}$ on inputs of length $< n$, and second that $R^{(n)}$ is uniform on the set of functions $\{0,1\}^n \to \{0,1\}$ where inputs of the form $0^{n/2}\varphi$ are decided according to $\mathsf{QBF}_*^{\mathcal{R}}$ (there are $2^{2^n - 2^{n/2}} = 2^{\Omega(2^n)}$ such functions) rather than uniform over the set of all functions $\{0,1\}^n \to \{0,1\}$. The reader is left to check that the proof of Lemma B.4 still holds if U_{2^n} is replaced by any distribution that is uniform on a set of size $2^{\Omega(2^n)}$. ■

Proof of Claim C.2. For every efficiently computable $f^{\mathcal{R}} : \{0,1\}^n \times \{0,1\}^{\ell} \to \{0,1\}^m$, we give an family of circuits $C^{\mathcal{R}}$ that for every $w \in \{0,1\}^n$ guarantees

$$\Pr_{x \xleftarrow{R} U_{\ell}}[C^{\mathcal{R}}(y) \in (f_w^{\mathcal{R}})^{-1}(y) \mid y = f_w^{\mathcal{R}}(x)] > 1/2$$

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

Proof of Claim C.3. We reduce L to SD using the same argument as in the proof of Lemma 4.3. The only point to check is that in order to describe $\mathcal{R}$ on all inputs up to length $6 \log p = O(\log n)$, we need to be able to decide $\mathsf{QBF}_*^{\mathcal{R}}$ formulas of size up to $3 \log p = O(\log n)$. But we can do this in polynomial time by brute force because the formulas are so short, and so the same argument follows. ■

ECCC ISSN 1433-8092
http://eccc.hpi-web.de/