Electronic Colloquium on Computational Complexity, Report No. 23 (2017)

The Power of Natural Properties as Oracles

Russell Impagliazzo [*] Valentine Kabanets [†] Ilya Volkovich [‡]

Abstract

We study the power of randomized complexity classes that are given oracle access to a natural property of Razborov and Rudich (*JCSS*, 1997) or its special case, the Minimal Circuit Size Problem (MCSP). We obtain new circuit lower bounds, as well as some hardness results for the relativized version of MCSP under randomized reductions. For example, we show that

$$\mathsf{ZPEXP}^{\mathsf{MCSP}} \not\subseteq \mathsf{P/poly},$$

and that

$$\oplus\mathsf{P} \subseteq \mathsf{ZPP}^{\mathsf{MCSP}^{\oplus\mathsf{P}}}.$$

Our results build on the recent work of Carmosino, Impagliazzo, Kabanets, and Kolokolova (*CCC*, 2016) connecting natural properties and learning algorithms.

Keywords: natural properties, Minimal Circuit Size Problem (MCSP), circuit lower bounds, hardness of MCSP, learning algorithms

1 Introduction

Historically, the problem of minimizing a circuit representing a given Boolean function (MCSP) was one of the first where the prohibitive computational cost of searching through a huge space of candidate solutions was noted [Kar85, Tra84]. This issue would later be formalized in the theory of NP-completeness. However, the complexity of circuit minimization itself remains largely mysterious. It is an NP problem, but neither known to be NP-complete nor in any sub-class of NP thought proper. This mystery remains despite a large body of work devoted to this problem [KC00, ABK+06, AHM+08, AD14, AHK15, HP15, MW15, HW16].

We do know that MCSP is *not* NP-hard (even P-hard) under very restrictive reductions [MW15], and that the NP-hardness of MCSP under other kinds of restricted reductions would imply new circuit lower bounds [KC00, MW15, AHK15]. We also know that NP-hardness of MCSP cannot be shown with certain "black-box" reductions [HW16]. On the other hand, the only hardness results known for MCSP is that it is SZK-hard under randomized (BPP) reductions [AD14], and NC^1-hard under truth table reductions computable by non-uniform TC^0 circuits [OS16].

There are two interpretations of the negative results about completeness. Are these results about MCSP and its relationship to other problems, or about the weakness of certain types of

[*]Department of Computer Science, University of California San Diego, La Jolla, CA. Email russell@cs.ucsd.edu
[†]School of Computing Science, Simon Fraser University, Burnaby, BC, Canada. Email kabanets@cs.sfu.ca
[‡]Department of EECS, CSE Division, University of Michigan, Ann Arbor, MI. Email: ilyavol@umich.edu.

ISSN 1433-8092

reductions? If we can't prove completeness of MCSP, can we find other evidence that MCSP is indeed a hard problem, or at least that it will be difficult to design an efficient algorithm for it?

Motivated by these questions, we consider the strength of MCSP under much stronger classes of reductions: probabilistic Turing reductions. Since previous non-completeness results also held for relativized versions of this class, we also consider MCSP^B, the relativized version where the circuits have gates for an oracle B. (It is known MCSP^B is PSPACE-complete under randomized zero-error (ZPP) reductions, for any PSPACE-complete language B [ABK+06]. For more restrictive reductions and various oracles B, a number of impossibility results are given in [AHK15].) Since an efficient algorithm for MCSP would be an almost ideal "natural property" for circuit lower bounds, in the sense of [RR97], we were also motivated by some unexpected algorithmic uses of natural properties [CIKK16, OS16]. Since natural properties are useful in proving circuit lower bounds, we would also like to tie easiness of MCSP to the construction of explicit hard functions in the sense of circuit complexity.

1.1 Our results

While, for simplicity, we state these results below for the special case of reductions to MCSP, for most of our results, MCSP could be replaced with any other natural property (having largeness and usefulness, but with oracle access replacing constructivity). Roughly, our results are of two kinds:

- circuit lower bounds for randomized complexity classes with MCSP oracle, and
- hardness results for relativized versions of MCSP under randomized reductions.

The former give evidence that designing an efficient algorithm for MCSP, or even putting it in a very small complexity class, will be non-trivial, as it would yield new circuit lower bounds. The latter results give evidence that the non-completeness results for MCSP do not hold for the most powerful reductions, and so give hope that we will be able to show hardness results for MCSP.

To prove our circuit lower bounds, we first show a number of conditional collapses (*à la* Karp-Lipton [KL80]) saying that if a certain complexity class has low circuit complexity, then it can be computed by efficient randomized algorithms with the MCSP oracle. Below, the notation $\mathcal{C}$-MCSP means the version of MCSP for circuits of type $\mathcal{C}$, for some circuit class $\mathcal{C}$. Also, $\mathcal{C}$-$\mathsf{SIZE}[s]$ denotes the class of Boolean functions computable by size s circuits of type $\mathcal{C}$.

Conditional collapses. The results of [LFKN92], [BFL91], [IKW02] and [BH92] (building upon [KL80]) imply collapse theorems for the classes $\mathsf{P}^{\#\mathsf{P}}$, PSPACE and EXP, NEXP, $\mathsf{EXP}^{\mathsf{NP}}$, respectively. More specifically, they show that if any of the above classes has polynomial size Boolean circuits, then the corresponding class collapses to MA. Using the observation that $\mathsf{MA} \subseteq \mathsf{NP}^{\mathsf{MCSP}}$ (see e.g. [ABK+06]) and an argument similar to Lemma 2.14, one could extend these theorems to get a collapse down to $\mathsf{NP}^{\mathcal{C}\text{-}\mathsf{MCSP}}$, given that the underlying class has polynomial size circuits from $\mathcal{C}$. We strengthen the above collapse further to $\mathsf{ZPP}^{\mathcal{C}\text{-}\mathsf{MCSP}}$, under the same conditions.

Theorem 1. *Let $\mathcal{C}$ be any circuit class and let $\Gamma \in \{\oplus\mathsf{P}, \mathsf{P}^{\#\mathsf{P}}, \mathsf{PSPACE}, \mathsf{EXP}, \mathsf{NEXP}, \mathsf{EXP}^{\mathsf{NP}}\}$. If $\Gamma \subseteq \mathcal{C}$-$\mathsf{SIZE}[\mathsf{poly}]$, then $\Gamma \subseteq \mathsf{ZPP}^{\mathcal{C}\text{-}\mathsf{MCSP}}$.*

Remark 1.1. *The theorem above can be interpreted as follows: A proof that* MCSP *is* not NP*-hard (or even $\#\mathsf{P}$-hard) under* ZPP*-reductions would imply that* $\mathsf{P}^{\#\mathsf{P}} \not\subseteq \mathsf{P}/\mathsf{poly}$.

Circuit lower bounds for MCSP-oracle complexity classes. Given the collapse theorems above, we get fixed-polynomial and super-polynomial lower bounds for randomized polynomial and exponential times, respectfully. The extra bit of advice in the case of randomized polynomial time comes to accommodate the need to keep the promise of bounded away probabilities of acceptance and rejection; the same problem arises in [Bar02, FS04, MP07, San09, Vol14]. Alternatively, we can consider the corresponding class of promise problems (i.e., prZPP).

Theorem 2. *For every circuit class $\mathcal{C}$, we have the following:*

1. $\mathsf{ZPP}^{\mathcal{C}\text{-}\mathsf{MCSP}}/1 \not\subseteq \mathcal{C}\text{-}\mathsf{SIZE}[n^k]$ *and* $\mathsf{prZPP}^{\mathcal{C}\text{-}\mathsf{MCSP}} \not\subseteq \mathcal{C}\text{-}\mathsf{SIZE}[n^k]$, *for all $k \in \mathbb{N}$.*
2. $\mathsf{ZPEXP}^{\mathcal{C}\text{-}\mathsf{MCSP}} \not\subseteq \mathcal{C}\text{-}\mathsf{SIZE}[\mathsf{poly}]$.

The above result still holds if we relax the $\mathcal{C}$-MCSP oracle to a natural property strongly useful against $\mathcal{C}$ (see Theorem 9 for more details). Combining this result with Lemma 2.17, we obtain that PAC learning algorithms imply fixed-polynomial lower bounds against $\mathsf{BPP}/1$ and super polynomial lower bounds againt BPEXP. These bounds match the results of [Vol14] and [FK09, KKO13], respectively (see Corollary 2.18 for more details). In this sense, our *unconditional* lower bounds generalize the *conditional* lower bounds of [Vol14] and [FK09, KKO13]. Indeed, our result is obtained by extending the techniques of [FK09, KKO13, Vol14].

The following theorem should be contrasted with a result from [IKW02] saying that the existence of a P-natural property (even *without* the largeness condition) that is useful against P/poly would imply that $\mathsf{NEXP} \not\subseteq \mathsf{P}/\mathsf{poly}$. *With* the largeness condition, the circuit lower bound can be shown to hold for the potentially smaller uniform complexity class ZPEXP.

Theorem 3. *Let $\mathcal{C}$ be any circuit class such that there is a ZPP-natural property strongly useful against $\mathcal{C}$. Then* $\mathsf{ZPEXP} \not\subseteq \mathcal{C}\text{-}\mathsf{SIZE}[\mathsf{poly}]$.

Remark 1.2. *The conclusion of Theorem 3 still holds if we assume a natural property with only weakly-exponential usefulness, $2^{n^{\Omega(1)}}$.*

Corollary 1.3. *If there is a ZPP-natural property that is weakly-exponentially useful against ACC^0 circuits, then* $\mathsf{ZPEXP} \not\subseteq \mathsf{ACC}^0[\mathsf{poly}]$. [1]

In [HW16], Hirahara and Watanabe defined the notion of oracle-independent randomized reductions and initiated a study of the set of languages that are reducible in randomized polynomial time to MCSP^B for every B. As a part of their study, they showed that $\bigcap_B \mathsf{BPP}^{\mathsf{MCSP}^B[1]} \subseteq \mathsf{AM} \cap \mathsf{coAM}$; this implies that NP-hardness of MCSP cannot be established via oracle-independent reductions unless the polynomial hierarchy collapses. We show circuit lower bounds for the class $\bigcap_B \mathsf{BPP}^{\mathsf{MCSP}^B}$.

Theorem 4. *We have that* $\bigcap\limits_B \mathsf{BPP}^{\mathsf{MCSP}^B}/1 \not\subseteq \mathcal{C}\text{-}\mathsf{SIZE}[n^k]$ *and* $\bigcap\limits_B \mathsf{prBPP}^{\mathsf{MCSP}^B} \not\subseteq \mathsf{SIZE}(n^k)$, *for all $k \in \mathbb{N}$, and that* $\bigcap\limits_B \mathsf{BPEXP}^{\mathsf{MCSP}^B} \not\subseteq \mathsf{P}/\mathsf{poly}$.

The smallest uniform complexity classes for which unconditional lower bounds as above are known are $\mathsf{MA}/1$ and prMA [San09], and MAEXP [BFT98], respectively. The relationship between

[1] The result that P-natural properties against sub-exponential size circuits yield ZPEXP lower bounds was also obtained in independent work by Igor Oliveira and Rahul Santhanam [OS16].

$\mathsf{ZPP}^{\mathcal{C}\text{-}\mathsf{MCSP}}$ and MA is not entirely known. Indeed, much research [KC00, ABK+06, AHM+08, AHK15, HP15, MW15, HW16] has been invested in the question whether $\mathcal{C}$-MCSP is NP-hard or even if $\mathsf{NP} \subseteq \mathsf{ZPP}^{\mathcal{C}\text{-}\mathsf{MCSP}}$, for various circuit classes $\mathcal{C}$. We remark that for every $\mathcal{C}$: $\mathsf{NP} \subseteq \mathsf{ZPP}^{\mathcal{C}\text{-}\mathsf{MCSP}} \iff \mathsf{ZPP}^{\mathsf{NP}} \subseteq \mathsf{ZPP}^{\mathcal{C}\text{-}\mathsf{MCSP}}$. This, in turn, implies that $\mathsf{NP} \subseteq \mathsf{ZPP}^{\mathcal{C}\text{-}\mathsf{MCSP}} \iff \mathsf{MA} \subseteq \mathsf{ZPP}^{\mathcal{C}\text{-}\mathsf{MCSP}}$ since $\mathsf{MA} \subseteq \mathsf{ZPP}^{\mathsf{NP}}$ [GZ11]. Indeed, in the cases when $\mathsf{NP} \subseteq \mathsf{ZPP}^{\mathcal{C}\text{-}\mathsf{MCSP}}$, the theorems in this section provide alternative proofs for previously known results. On the other hand, if $\mathsf{ZPP}^{\mathsf{MCSP}} \subseteq \mathsf{NP}$, then $\mathsf{NP} = \mathsf{MA}$ (see e.g. [ABK+06]).

Hardness of relativized versions of MCSP. It is shown by [ABK+06] that every language in PSPACE is reducible to $\mathsf{MCSP}^{\mathsf{PSPACE}}$ via ZPP-reductions. We use different techniques to re-prove this result, as well as obtain a few new results along the same lines.

Theorem 5. *1.* $\mathsf{PSPACE} \subseteq \mathsf{ZPP}^{\mathsf{MCSP}^{\mathsf{PSPACE}}}$ *[ABK+06]*

2. $\oplus\mathsf{P} \subseteq \mathsf{ZPP}^{\mathsf{MCSP}^{\oplus\mathsf{P}}}$

3. $\mathsf{P}^{\#\mathsf{P}} \subseteq \mathsf{BPP}^{\mathsf{MCSP}^{\#\mathsf{P}}}$

4. $\mathsf{PP} \subseteq \mathsf{BPP}^{\mathsf{MCSP}^{\mathsf{PP}}}$. *Moreover, for* $k \geq 2$*:* $C_k\mathsf{P} \subseteq C_{k-1}\mathsf{P}^{\mathsf{MCSP}^{\mathsf{PP}}}$.

Next we show that every language B can be approximated by polynomial size circuits with MCSP^B gates. For self-correctable languages, this implies that they can be computed *exactly* by such circuits.

Theorem 6. *For any language B, $n \in \mathbb{N}$ and $\delta > 0$, there exists a MCSP^B-oracle circuit C of size $\mathsf{poly}(n, 1/\delta)$ that is $1-\delta$ close to $B|_n$. If, in addition, B is self-correctable then B has polynomial size MCSP^B-oracle circuits.*

Mimicking self-correctable languages, we extend the ability of MCSP^B-oracle circuits to compute the language exactly (rather than just approximate it) for a large family of languages.

Theorem 7. *Let B be a language such that PSPACE^B has polynomial size B-oracle circuits. Then B has polynomial-size MCSP^B-oracle circuits.*

In [ABK+06], the same outcome was achieved under a stronger assumption that $\mathsf{PSPACE}^B \subseteq \mathsf{P}^B$. We note our result is not a mere syntactical improvement, as there are numerous languages B for which $\mathsf{PSPACE}^B \subseteq \mathsf{P}^\mathsf{B}/\mathsf{poly}$ yet $\mathsf{PSPACE}^B \neq \mathsf{P}^B$; see Appendix B for more details. While we suspect that the consequent of the theorem holds unconditionally, we note that the precondition statement of the theorem cannot be improved further since Lemma 2.11 implies that, for every language B, the class PSPACE^B does not have fixed-polynomial size B-oracle circuits.

1.2 Our techniques

We rely on the result of [CIKK16] showing that natural properties useful against a (sufficiently powerful) circuit class $\mathcal{C}$ yield learning algorithms (under the uniform distribution, with membership queries) for the same circuit class. We note that this result relativizes in the following sense: if we have a natural property useful against circuits with L oracle gates (say, MCSP^L), for some language L, then we can approximately learn L, with the hypotheses being circuits with MCSP^L oracle gates.

If, in addition, this language L is both downward and random self-reducible, then we can learn L exactly, with the same type of MCSP^L oracle circuits, using the ideas of [IW98].

This allows us to prove, for example, that $\mathsf{P}^{\#\mathsf{P}} \subseteq \mathsf{BPP}^{\mathsf{MCSP}^{\#\mathsf{P}}}$, as $\#\mathsf{P}$ has a complete problem (the permanent) that is well-known to be both downward and random self-reducible. We show that $\oplus\mathsf{P}$ also has such a complete problem (building upon [TV07]), getting the inclusion $\oplus\mathsf{P} \subseteq \mathsf{BPP}^{\mathsf{MCSP}^{\oplus\mathsf{P}}}$. To get the stronger result that $\oplus\mathsf{P} \subseteq \mathsf{ZPP}^{\mathsf{MCSP}^{\oplus\mathsf{P}}}$, we use Toda's Theorem [Tod91] and hardness-randomness tradeoffs of [IW97] to get rid of the two-sided error of our BPP reduction (similarly to the work of [KC00]).

Our circuit lower bounds are proved using similar ideas. For example, to prove $\mathsf{ZPEXP}^{\mathsf{MCSP}} \not\subseteq \mathsf{P}/\mathsf{poly}$, we argue as follows. If $\mathsf{PSPACE} \not\subseteq \mathsf{P}/\mathsf{poly}$, we are done (as $\mathsf{PSPACE} \subseteq \mathsf{EXP}$). Assuming $\mathsf{PSPACE} \subseteq \mathsf{P}/\mathsf{poly}$, we get that $\mathsf{PSPACE} \subseteq \mathsf{ZPP}^{\mathsf{MCSP}}$, using the fact that PSPACE contains a complete problem that is both downward and random self-reducible [TV07], and that $\mathsf{MCSP}^{\mathsf{PSPACE}} \subseteq \mathsf{PSPACE} \subseteq \mathsf{P}/\mathsf{poly}$. The circuit lower bound then follows by a translation argument, as we get that $\mathsf{EXPSPACE} \subseteq \mathsf{ZPEXP}^{\mathsf{MCSP}}$ and $\mathsf{EXPSPACE}$ is known to contain languages of maximal circuit complexity (by a simple diagonalization argument).

Remainder of the paper. We give basic definitions and notation in Section 2. In Section 3, we prove our main results (Theorems 1 - 4) which show new collapse results as well as new circuit lower bounds for uniform complexity classes with oracle access to (relativized) MCSP. In fact, we prove somewhat stronger results (Theorems 8 and 9) which apply to the more general type of oracles: strongly useful natural properties. Next, in Section 3.3, we prove our results about reductions to the problem MCSP^B, for various languages B. Specifically, we give such reductions for several complexity classes (Theorem 5), and also show that every language B can be approximated by "small" Boolean circuits containing MCSP^B oracle gates (Theorem 6). Finally, we show that under certain conditions, a language B can be computed exactly by "small" Boolean circuits containing MCSP^B oracle gates (rather than just approximated) (Theorem 7). We conclude with some open questions in Section 4. Some of the proofs (e.g., our proof that $\oplus\mathsf{P}$ has a complete problem that is both downward and random self-reducible) are given in the appendix.

2 Preliminaries

2.1 Basics

For Boolean functions $f, g : \{0,1\}^n \to \{0,1\}$, we define the *relative distance* $\Delta(f,g)$ to be the fraction of inputs $x \in \{0,1\}^n$ where $f(x) \neq g(x)$. For $\varepsilon \geq 0$, we say that f is ε*-close* to g if $\Delta(f,g) \leq \varepsilon$, otherwise we say that f is ε*-far* from g.

Let $L \subseteq \{0,1\}^*$ be a language. We denote by $L|_n$ the set of the strings of length n in L. We will associate a language L with a corresponding Boolean function in the natural way: $L(x) = 1 \iff x \in L$. We say that L has circuits of size $a(n)$ and denote it by $L \in \mathsf{SIZE}(a(n))$ if for every $n \in \mathbb{N}$ the function $L|_n$ can be computed by a Boolean circuit of size $\mathcal{O}(a(n))$.

A *circuit class* $\mathcal{C}$ is a subset of all Boolean circuits (e.g. circuits with AND,OR and NOT gates, $\mathsf{AC}^0, \mathsf{ACC}^0, \mathsf{TC}^0, \mathsf{NC}^2$ etc.). We assume that the representation in $\mathcal{C}$ is chosen in way that a size s circuit can be described using $\mathsf{poly}(s)$ bits. In addition, given a circuit $C \in \mathcal{C}$ of size s the circuit $C|_{x_i=b}$ is also in $\mathcal{C}$ and of size at most s, when $C|_{x_i=b}$ is the circuit resulting from C by fixing the variable x_i to the bit $b \in \{0,1\}$. We aalso ssume that a circuit $C \in \mathcal{C}$ can be evaluated in polynomial time given its description.

We denote by $\mathcal{C}\text{-}\mathsf{SIZE}[a(n)]$ the set of languages having circuits of size $\mathcal{O}(a(n))$ from the class $\mathcal{C}$. The *circuit complexity* $\mathsf{s}_L^{\mathcal{C}}(n)$ of L with respect to the circuit class $\mathcal{C}$ at length n is the smallest integer t such that there is a circuit of size t from $\mathcal{C}$ that computes $L|_n$. We similarly define $\mathsf{s}_L(n)$ to be the circuit complexity of L with respect to general Boolean circuits, and $\mathsf{s}_L^B(n)$ to be the circuit complexity of L with respect to B-oracle circuits.

We say that a class $\mathcal{C}$ *implements* P if P has polynomial size circuits from $\mathcal{C}$ (i.e. $\mathsf{P} \subseteq \mathcal{C}\text{-}\mathsf{SIZE}[\mathsf{poly}]$). The following Lemma translates a uniform inclusion to a non-uniform.

Lemma 2.1. *Let $\mathcal{C}$ be a circuit class that implements P. Then for any circuit class $\mathcal{C}'$ there exists $k \in \mathbb{N}$ such that for every language L: $\mathsf{s}_L^{\mathcal{C}}(n) \le (\mathsf{s}_L^{\mathcal{C}'}(n) \cdot n)^k$.*

Proof. Consider the $\mathcal{C}'$-circuit value problem: $\mathcal{C}'\text{-val} \triangleq \{(C', x) \mid C' \in \mathcal{C}', C'(x) = 1\}$. By definition, $\mathcal{C}'\text{-val} \in \mathsf{P}$. By the assumption, there exists $k \in \mathbb{N}$ such that for every $n, m \in \mathbb{N}$ there exists a circuit $C_{m,n}(y, x) \in \mathcal{C}$ of size $(m \cdot n)^k$ such that for every circuit $C' \in \mathcal{C}$, described using m bits, we have that: $\hat{C}_{m,n}(x) \triangleq C_{m,n}(\langle C' \rangle, x) \equiv C'(x)$. Here $\langle C' \rangle$ denote the description of C' in bits. The claim follows by recalling that a circuit of size s from $\mathcal{C}'$ can be described using $\mathsf{poly}(s)$ bits. □

In particular, the lemma implies that if $\mathcal{C}$ implements P then $\mathcal{C}\text{-}\mathsf{SIZE}[\mathsf{poly}] = \mathsf{P/poly}$. We will use this relation implicitly going forward. Similarly, we have the following easy observation.

Observation 2.2. *Let A, B be two languages. Suppose that $A \in \mathsf{SIZE}^B(n^k)$ for some $k \in \mathbb{N}$. Then for every language L: $\mathsf{s}_L^B(n) \le \mathsf{s}_L^A(n)^{k+1}$.*

A promise problem is a relaxation of a language, defined as follows.

Definition 2.3 (Promise Problems)**.** *$\Pi = (\Pi_{YES}, \Pi_{NO})$ is a* promise problem *if $\Pi_{YES} \cap \Pi_{NO} = \emptyset$. We say that a language L is* consistent *with Π iff $x \in \Pi_{YES} \implies x \in L$ and $x \in \Pi_{NO} \implies x \notin L$. The containment of L outside of $\Pi_{YES} \dot\cup \Pi_{NO}$ can be arbitrary. We say that a set of languages Γ is consistent with a set of promise problems Λ iff for every $\Pi \in \Lambda$ there is $L \in \Gamma$ that is consistent with Π.*

Definition 2.4 (Lower Bounds for Promise Problems)**.** *Let $\mathcal{C}$ be a circuit class and $f(n)$ be a function. Then $\Pi \notin \mathcal{C}\text{-}\mathsf{SIZE}(f(n)) \iff \forall L$ consistent with Π: $L \notin \mathcal{C}\text{-}\mathsf{SIZE}(f(n))$.*

In other words, in order to translate a lower bound for a set of promise problems Λ into a lower bound for a set of languages, one must find a set of languages Γ that is consistent with Λ.

We refer the reader to [AB09] for the definitions of standard complexity classes such as P, ZPP, RP, BPP, NP, MA, PSPACE, etc. We say that a language $L \in \mathsf{BPP}/1$ if L can be decided by a BPP machine with an auxiliary *advice* bit b_n for each input of length n; note that given the complement advice bit $\bar{b}_n$, the machine is not guaranteed to be a BPP machine (i.e., may not have bounded away acceptance and rejection probabilities on all inputs of length n). We define $\mathsf{ZPP}/1$ in a similar fashion.

We define a family of natural problems complete for prBPP relative to any oracle.

Definition 2.5 (Circuit Approximation)**.** *For a language B, define $\mathrm{CA}^B \triangleq (\mathrm{CA}^B_{YES}, \mathrm{CA}^B_{NO})$:*

$$\mathrm{CA}^B_{YES} = \{C \text{ is a } B\text{-oracle circuit} \mid \Delta(C, \bar{0}) \ge 3/4\},$$
$$\mathrm{CA}^B_{NO} = \{C \text{ is a } B\text{-oracle circuit} \mid \Delta(C, \bar{0}) \le 1/4\}.$$

CA^B *is* prBPP^B*-complete for any* B*.*

To prove lower bounds against randomized classes with one bit of advice, we shall rely on the following definitions (and their extensions) from [San09, Vol14].

Definition 2.6 (Padded Languages)**.** *Let* L *be a language and* $\mathcal{C}$ *be a circuit class. For* $k \in \mathbb{N}$ *we define the padded version of* L*, denoted* $L'_{k,\mathcal{C}}$*, to consist of the strings* $1^m x$ *satisfying the following: (1)* m *is power of* 2*; (2)* $r \triangleq |x| \leq m$*; (3)* $x \in L$*; and (4)* $\mathsf{s}_L^{\mathcal{C}}(r) \leq m^{2k}$*.*

The main property of the padded languages is that, for every L, sufficiently small circuits for $L'_{k,\mathcal{C}}$ can be used to construct small circuits for L.

Lemma 2.7 ([San09, Vol14])**.** *Let* $k \in \mathbb{N}$*. Suppose* $L'_{k,\mathcal{C}} \in \mathcal{C}\text{-}\mathsf{SIZE}[n^k]$*. Then* $\mathsf{s}_L^{\mathcal{C}}(n) = \mathcal{O}(n^{2k})$*.*

The next lemma is implicit in [Vol14]. We provide the proof for completeness.

Lemma 2.8. *Let* $\mathcal{R}$ *be a strongly useful property against* $\mathcal{C}$ *and let* L *be a downward self-reducible and self-correctable language. Then for all* $k \in \mathbb{N}$*:* $L'_{k,\mathcal{C}} \in \mathsf{BPP}^{\mathcal{R}}/1$*.*

Proof. Let $y = 1^m x$ be an input for $L'_{k,\mathcal{C}}$. Conditions 1 and 2 can be checked easily. As y has a unique interpretation, we use the advice bit to determine whether $\mathsf{s}_L^{\mathcal{C}}(|x|) \leq m^{2k}$. If the advice bit is 0 (i.e "no") we reject. Otherwise, we apply Lemma 3.4 with $t = m^{2k}$ to decide if $x \in L$. □

We also need the following result that shows that a lower bound on $\mathsf{ZPP}/1$ carries over to prZPP.

Lemma 2.9 ([San09])**.** *For every circuit class* $\mathcal{C}$ *and a function* $u(n)$*, if* $\mathsf{ZPP}/1 \not\subseteq \mathcal{C}\text{-}\mathsf{SIZE}[u(n)]$*, then* $\mathsf{prZPP} \not\subseteq \mathcal{C}\text{-}\mathsf{SIZE}[u(n)]$*.*

Finally, we need the following collapse results and a simple circuit lower bounds against PSPACE.

Lemma 2.10 ([BFL91, IKW02, BH92])**.** *If* $\Gamma \in \left\{\mathsf{EXP}, \mathsf{NEXP}, \mathsf{EXP}^{\mathsf{NP}}\right\}$ *is in* P/poly*, then* $\Gamma = \mathsf{MA}$*.*

Lemma 2.11 (Folklore)**.** *For any circuit class* $\mathcal{C}$ *and a language* B*, let* $\mathcal{C}^B$ *be the class of circuits from* $\mathcal{C}$ *with* B*-oracle gates. Then, for any* $k \in \mathbb{N}$*,* $\mathsf{PSPACE}^B \not\subseteq \mathcal{C}^B\text{-}\mathsf{SIZE}[n^k]$*. More generally, for every function* $s(n) = \mathcal{O}(2^n)$*,* $\mathsf{DSPACE}^B(\mathsf{poly}(s(n))) \not\subseteq \mathcal{C}^B\text{-}\mathsf{SIZE}[s(n)]$*.*

2.2 Derandomization from hardness

We recall the celebrated hardness-randomness tradeoff.

Lemma 2.12 ([NW94, BFNW93, IW97, Uma03, KM02])**.** *There is a polynomial-time computable oracle predicate* $M^B(x, y)$ *and a constant* $\ell \in \mathbb{N}$ *such that the following holds for every language* B *and* $s \in \mathbb{N}$*. If* $tt \in \{0,1\}^{2^m}$ *is a string that represents the truth table of an* m*-variate Boolean function* f *which requires* B*-oracle circuits of size* s^ℓ*, then, for all* s*-size* B*-oracle circuits* C*,* $M^B(C, tt)$ *is consistent with* CA^B*.*

The non-relativized version of this result was used in [KC00] to show that $\mathsf{BPP} \subseteq \mathsf{ZPP}^{\mathsf{MCSP}}$. We use the relativized version to show that under certain assumptions $\mathsf{BPP}^A = \mathsf{ZPP}^A$.

Lemma 2.13. *Let* A, B *be any languages such that:*

1. $A \in \mathsf{P}^{\mathsf{B}}/\mathsf{poly}$.

2. $\mathsf{MCSP}^B \in \mathsf{ZPP}^A$.

Then $\mathsf{BPP}^A = \mathsf{ZPP}^A$.

Proof. By definition, $\mathsf{ZPP}^A \subseteq \mathsf{BPP}^A$. For the second direction, let $L \in \mathsf{BPP}^A$. Then for each n there exists an A-oracle circuit $C(w, r)$ of size $\mathsf{poly}(n)$ such that $x \in L \iff C(x, \cdot) \in \mathrm{CA}^A$. We now describe a machine that decides L:

- For $m = \mathcal{O}(\log n)$ pick a truth table $tt \in \{0,1\}^{2^m}$ at random
- If tt has B-circuits of size less than $2^{m/4}$ return "?" (using an oracle to MCSP^B).
- Otherwise, run $M^A(C(x,\cdot), tt)$ and answer the same (using M^A from Lemma 2.12)

By counting arguments, a random function requires exponential size circuits w.h.p. Therefore, the algorithm will output "?" extremly rarely. By Observation 2.2 tt requires A-oracle circuits of size $2^{\Omega(m)} = n^{\Omega(1)}$. Consequently, the correctness of the algorithm follows from Lemma 2.12. As described, the algorithm can be implemented in $\mathsf{ZPP}^{A,\mathsf{MCSP}^B}$. By the preconditions,

$$\mathsf{ZPP}^{A,\mathsf{MCSP}^B} \subseteq \mathsf{ZPP}^{A,\mathsf{ZPP}^A} = \mathsf{ZPP}^A$$

due to the self-lowness of ZPP. □

Lemma 2.14. *Let $\mathcal{C}$ be a circuit class that implements* P. *Then* $\mathsf{BPP} \subseteq \mathsf{ZPP}^{\mathcal{C}\text{-}\mathsf{MCSP}}$. *If, in addition,* $\mathcal{C}$-$\mathsf{MCSP} \in \mathcal{C}$-$\mathsf{SIZE}[\mathsf{poly}]$*, then* $\mathsf{BPP}^{\mathcal{C}\text{-}\mathsf{MCSP}} = \mathsf{ZPP}^{\mathcal{C}\text{-}\mathsf{MCSP}}$.

Proof. Similar to the proof of Lemma 2.13, by using Lemma 2.1 instead of Observation 2.2. □

2.3 Natural properties, PAC learning and MCSP

We first define natural properties.

Definition 2.15 (Natural Property [RR97]). *Let Γ and Λ be complexity classes. We say that a property $\mathcal{R}$ is* Γ-natural with density δ_n *and* useful *against Λ if the following holds:*

1. ***Constructivity:*** *Given a binary string $tt \in \{0,1\}^{2^m}$, $tt \in \mathcal{R}$ can be decided in Γ.*

2. ***Largeness:*** *For all n, $\mathcal{R}$ contains at least a δ_n fraction of all 2^n binary strings, representing n-variate Boolean functions.*

3. ***Usefulness:*** *For every sequence of Boolean functions $f_1, f_2, \ldots,$ where f_n is a function on n variables, such that $\{tt \mid tt$ is a truth table of some $f_n\} \subseteq \mathcal{R}$ and sufficiently large n, we have $f_n \notin \Lambda$.*

We say that $\mathcal{R}$ is strongly useful *against a circuit class $\mathcal{C}$ if there exists $a \in \mathbb{N}$ such that $\mathcal{R}$ is useful against $\mathcal{C}$-$\mathsf{SIZE}[2^{an}]$ and has density $\delta_n \geq 2^{-an}$.*

Considering $\mathcal{R}$ as an oracle allows us to "ignore" its complexity. In addition, if $\mathcal{R}$ is strongly useful against $\mathcal{C}$, then, as observed in [CIKK16, Lemma 2.7], there exists another property $\mathcal{R}' \in \mathsf{P}^{\mathcal{R}}$ that is strongly useful against $\mathcal{C}$ with density $\delta_n \geq 1/2$. Therefore, when considering a strongly useful property as an oracle we can assume w.l.o.g that it has density $\delta_n \geq 1/2$.

Observe that MCSP yields a strongly useful natural property. Often, the only requirement from an MCSP oracle is to "serve" as a strongly useful natural property. Consequently, the oracle can be relaxed. The following can be shown along the lines of previous claims.

Lemma 2.16. *Let $\mathcal{C}$ be a circuit class that implements* P *and let $\mathcal{R}$ be a natural property strongly useful against $\mathcal{C}$. Then $\mathsf{BPP} \subseteq \mathsf{ZPP}^{\mathcal{R}}$. If, in addition, $\mathcal{R} \in \mathcal{C}\text{-}\mathsf{SIZE}[\mathsf{poly}]$ then $\mathsf{BPP}^{\mathcal{R}} = \mathsf{ZPP}^{\mathcal{R}}$.*

Recall Valiant's PAC learning model [Val84]. We have a (computationally bounded) learner that is given a set of samples of the form $(\bar{x}, f(\bar{x}))$ from some fixed function $f \in \mathcal{C}$, where $\bar{x}$ is chosen according to some unknown distribution D. Given $\varepsilon > 0$ and $\delta > 0$, the learner's goal is to output, with probability $1 - \varepsilon$ a hypothesis $\hat{f}$ such that $\hat{f}$ is a $1 - \delta$ close to f under D. We say that a function class $\mathcal{C}$ is PAC *learnable* if there exists a learner which given any $f \in \mathcal{C}$, $\varepsilon > 0$ and $\delta > 0$ in time polynomial in $n, 1/\varepsilon, 1/\delta, |f|$ outputs a hypothesis as required. In a more general model, the learner is allowed membership queries (as in the exact learning model). In this case, we say that $\mathcal{C}$ is PAC *learnable* with *membership queries*.

In [CIKK16] it was shown that natural properties yield efficient learning algorithms. Specifically, a BPP-natural property that is strongly useful against a circuit class $\mathcal{C}$ implies that $\mathcal{C}$ is PAC learnable under the uniform distribution, with membership queries (see Section 3.2 for more details). Here we show that the contrary holds as well: if $\mathcal{C}$ is PAC learnable under the uniform distribution, with membership queries then there is a BPP-natural property that is strongly useful against $\mathcal{C}$.

Lemma 2.17. *Let $\mathcal{C}$ be a circuit class. If $\mathcal{C}$ is* PAC *learnable under the uniform distribution, with membership queries, then there exists a* BPP*-natural property that is strongly useful against $\mathcal{C}$.*

The proof goes along the lines of Theorem 3 from [Vol14], where it is shown how to turn an efficient randomized exact learner $\mathcal{A}$ for a circuit class $\mathcal{C}$ into a P/poly-natural property strongly useful against $\mathcal{C}$. Combined with Theorem 2, we obtain a somewhat different proof for the conditional lowers bounds of [Vol14] and [FK09, KKO13].

Corollary 2.18. *For every circuit class $\mathcal{C}$, if $\mathcal{C}$ is* PAC *learnable under the uniform distribution, with membership queries then:*

1. $\mathsf{BPP}/1 \not\subseteq \mathcal{C}\text{-}\mathsf{SIZE}[n^k]$ *and* $\mathsf{prBPP} \not\subseteq \mathcal{C}\text{-}\mathsf{SIZE}[n^k]$, *for all* $k \in \mathbb{N}$ *[Vol14] .*
2. $\mathsf{BPEXP} \not\subseteq \mathcal{C}\text{-}\mathsf{SIZE}[\mathsf{poly}]$ *[FK09, KKO13].*

2.4 Downward self-reducible and self-correctable languages

Definition 2.19. *We say that a language L is* downward self-reducible *if there is a deterministic polynomial-time algorithm* COMPUTE *such that for all $n \geq 1$:*

$$\mathsf{COMPUTE}^{L|_{n-1}} = L|_n.$$

In other words, COMPUTE *efficiently computes L on inputs of size n given oracle access to a procedure that computes L on inputs of size $n - 1$.*

We say that a language L is self-correctable[2] *if there is a probabilistic polynomial-time algorithm* $\mathsf{CORRECT}$ *such that, for any $n \in \mathbb{N}$ and a Boolean function $f : \{0,1\}^n \to \{0,1\}$ it holds that if $\Delta(f, L|_n) \leq 1/n$ then for all $\bar{x} \in \{0,1\}^n$:* $\Pr[\mathsf{CORRECT}^f(\bar{x}) \neq L|_n(\bar{x})] \leq 1/\mathsf{poly}(n)$.

Several complexity classes have complete problems that are both downward self-reducible and self-correctable.[3]

Lemma 2.20 ([Val79, BF90, LFKN92, IW98])**.** *There exists a downward self-reducible and self-correctable $\#\mathsf{P}$-complete language L_{perm}.*

Lemma 2.21 ([TV07])**.** *There is a downward self-reducible and self-correctable* PSPACE*-complete language L_{PSPACE}.*

Using similar ideas as in [TV07], we also show the following; see Appendix A for the proof.

Lemma 2.22. *There is a downward self-reducible and self-correctable $\oplus\mathsf{P}$-complete language $L_{\oplus\mathsf{P}}$.*

To handle a larger family of languages, we generalize the notion of self-correctability.

Definition 2.23. *A language L is* $(\varepsilon(n), A)$-correctable *if there are a polynomial $r(n)$ and a randomized polynomial-time algorithm* $\mathsf{CORRECT}$ *such that, for all $n \in \mathbb{N}$ and $f \colon \{0,1\}^{r(n)} \to \{0,1\}$, if $\Delta\left(f, A|_{r(n)}\right) \leq \varepsilon(n)$, then, for all $\bar{x} \in \{0,1\}^n$,* $\Pr\left[\mathsf{CORRECT}^f(\bar{x}) \neq L|_n(\bar{x})\right] \leq 1/\mathsf{poly}(n)$.

In other words, there is a randomized polynomial-time algorithm that can decide $L|_n$ given an oracle to a function that approximates a $A|_{r(n)}$. Self-correctability is special case when $\varepsilon = 1/n$, $A = L$ and $r(n) = n$. The following is immediate using Adleman's result [Adl78]:

Lemma 2.24. *Let L be a $(\varepsilon(n), A)$-correctable language with $r(n)$, and let B be a language. Given $n \in \mathbb{N}$, suppose C is an $r(n)$-variate B-oracle circuit of size s such that $\Delta(C, A|_{r(n)}) \leq \varepsilon(n)$. Then there exists an n-variate B-oracle circuit C' of size $\mathsf{poly}(r(n), s)$ such that $C' \equiv L|_n$. Moreover, C' can be produced from C in randomized polynomial time.*

Klivans and van Melkebeek [KM02] show that any language L is $(\varepsilon(n), A)$-correctable for A computable in PSPACE with an oracle to L (by encoding the truth table of L with a list-decodable code of [STV01]).

Theorem 2.25. *For any language L and $\varepsilon(n)$ there exist a language $A \in \mathsf{DSPACE}^L(n + 1/\varepsilon(n))$ such that L is $\left(\frac{1}{2} - \varepsilon(n), A\right)$-correctable with $r(n) = \mathsf{poly}(n, 1/\varepsilon(n))$.*

2.5 Learning downward self-reducible and self-correctable languages

Lemma 2.26. *Let B be a language and let $\mathcal{C}$ be a circuit class that is* PAC *learnable using membership and B queries with hypotheses being B-oracle circuits. Suppose L is a downward self-reducible and self-correctable language. Then there is a randomized algorithm making oracle queries to B, that, given x and t, computes $L(x)$ with probability at least $1 - 1/\mathsf{poly}(|x|)$ in time $\mathsf{poly}(|x|, t)$, provided that $t \geq \mathsf{s}_L^{\mathcal{C}}(|x|)$.*

[2]More generally, such languages are referred to as "random self-reducible" languages.

[3]It is not hard to see that every downward self-reducible language is computable in PSPACE. On the other hand, the results of [FF93] suggest that there cannot be self-correctable languages which are complete for any level of the polynomial hierarchy, unless the hierarchy collapses.

Proof. Let $\mathcal{A}$ be a PAC learner for $\mathcal{C}$ and let $n = |x|$. First, we describe an algorithm that produces B-oracle circuits for $L|_1$, $L|_2$, $\ldots$, $L|_n$ w.h.p. We then use the circuit for $L|_n$ to decide x.

- Begin with a lookup table $\tilde{C}_1 = C_1$ for $L|_1$.
- For $i \geq 2$, invoke $\mathcal{A}$ with $\varepsilon = 1/i^3$ and $\delta = 1/i$ to learn a circuit $\tilde{C}_i$ of size t for $L|_i$.
- Answer the queries to B using the provided oracle.
- Given a query to $L|_i$, invoke $\mathsf{COMPUTE}$ with $C_{i-1}(x)$ as an oracle.
- Set $C_i \triangleq \mathsf{CORRECT}^{\tilde{C}_i}$ (convert the algorithm into a circuit using Lemma 2.24).

We claim that w.h.p it holds for all $1 \leq i \leq n$ that C_i is a B-oracle circuit of size $\mathsf{poly}(i, t)$ computing $L|_i$. The proof is by induction on i. Basis $i = 1$ is clear. Now assume that hypothesis holds for $i - 1$. Observe that since $C_{i-1}(x)$ is B-oracle circuit, it can be evaluated in polynomial time given and an oracle to B. Hence, by downward self-reducibility of L invoking $\mathsf{COMPUTE}$ with $C_{i-1}(x)$ can be used to obtain oracle access to $L|_i$. As $t \geq \mathsf{s}_L^{\mathcal{C}}(i)$, $\mathcal{A}$ will output a circuit $\tilde{C}_i$ of size $\mathsf{poly}(i, t)$. which is $1/i$ close to $L|_i$. Finally, using Lemma 2.24 the algorithm will produce a circuit C_i of size $\mathsf{poly}(i, t)$ that computes $L|_i$.

The above analysis is correct assuming that no errors have occurred. Note that the total number of steps is $\mathsf{poly}(i)$ while each steps has at most $1/\mathsf{poly}(i)$ probability error. As the latter polynomial can be made arbitrary small, we obtain that w.h.p. for all i, $C_i \equiv L|_i$.

Finally, all the listed procedures are in time $\mathsf{poly}(n, t)$, given oracle access to B. $\square$

3 The proofs

Our proofs will use the following.

Lemma 3.1 (Extension of Theorem 5.1 from [CIKK16])**.** *Let $\mathcal{C}$ be any circuit class that implements* P*. Let $\mathcal{R}$ be a natural property with density at least $1/5$, that is useful against $\mathcal{C}$-$\mathsf{SIZE}[u(n)]$, for some size function $u(n) \colon \mathbb{N} \to \mathbb{N}$. Then there is a randomized algorithm that makes oracle queries to $\mathcal{R}$ such that, given $s \in \mathbb{N}$, oracle access to a function $f \in \mathcal{C}$ of size s on n variables, and $\delta > 0$, it produces in time $\mathsf{poly}(n, 1/\delta, 2^{u^{-1}(\mathsf{poly}(n,1/\delta,s))})$ an $\mathcal{R}$-oracle circuit C where $\Delta(C, f) \leq \delta$.*

Corollary 3.2. *Let $\mathcal{C}$ be any circuit class that implements* P *and let $\mathcal{R}$ be a strongly useful property against $\mathcal{C}$. Then $\mathcal{C}$ is* PAC *learnable under the uniform distribution, using membership and $\mathcal{R}$ queries with hypotheses being $\mathcal{R}$-oracle circuits.*

Theorem 3.3. *Let $\mathcal{C}$ be any circuit class that implements* P*. Then $\mathcal{C}$ is* PAC *learnable under the uniform distribution, using membership and $\mathcal{C}$-MCSP queries with hypotheses being $\mathcal{C}$-MCSP-oracle circuits.*

By combining Lemma 2.26 and Corollary 3.2, we get the following.

Lemma 3.4. *Let $\mathcal{C}$ be any circuit class that implements* P *and let $\mathcal{R}$ be a strongly useful property against $\mathcal{C}$. Furthermore, let L be a downward self-reducible and self-correctable language. Then there is a randomized algorithm that makes oracle queries to $\mathcal{R}$, that given x and t computes $L(x)$ with probability at least $1 - 1/\mathsf{poly}(|x|)$ in time $\mathsf{poly}(|x|, t)$ provided that $t \geq \mathsf{s}_L^{\mathcal{C}}(|x|)$.*

3.1 Conditional collapses: Proof of Theorem 1

Theorem 1 by follows as a corollary from the next, somewhat stronger, theorem.

Theorem 8. *Let $\mathcal{C}$ be any circuit class and let $\mathcal{R}$ be a strongly useful property against $\mathcal{C}$. Furthermore, let*

$$\Gamma \in \left\{\oplus\mathsf{P}, \mathsf{P}^{\#\mathsf{P}}, \mathsf{PSPACE}, \mathsf{EXP}, \mathsf{NEXP}, \mathsf{EXP}^{\mathsf{NP}}\right\}.$$

Then, if $\Gamma \subseteq \mathcal{C}\text{-}\mathsf{SIZE}[\mathsf{poly}]$, then $\Gamma \subseteq \mathsf{BPP}^{\mathcal{R}}$. If, in addition, $\mathcal{R}$ is PH-natural then $\Gamma \subseteq \mathsf{ZPP}^{\mathcal{R}}$.

Proof. Clearly, $\mathcal{C}$ implements P. First, consider the case of Γ such that $\mathsf{PSPACE} \subseteq \Gamma$. For L_{PSPACE} from Lemma 2.21, we have that $\mathsf{s}^{\mathcal{C}}_{L_{\mathsf{PSPACE}}}(n) = \mathcal{O}(n^k)$ for some $k \in \mathbb{N}$. By Lemma 3.4, given x, we can compute $L_{\mathsf{PSPACE}}(x)$ in randomized polynomial time given oracle to $\mathcal{R}$. Consequently, $\mathsf{PSPACE} \subseteq \mathsf{BPP}^{\mathcal{R}}$. By Lemma 2.10, we get $\Gamma = \mathsf{MA}$. Hence, we have

$$\Gamma \subseteq \mathsf{MA} \subseteq \mathsf{PSPACE} \subseteq \mathsf{BPP}^{\mathcal{R}}.$$

If, in addition, $\mathcal{R} \in \mathsf{PH}$ then $\mathcal{R} \in \Gamma \subseteq \mathcal{C}\text{-}\mathsf{SIZE}[\mathsf{poly}]$. By Lemma 2.16, $\mathsf{BPP}^{\mathcal{R}} \subseteq \mathsf{ZPP}^{\mathcal{R}}$.

For $\Gamma = \oplus\mathsf{P}$, we argue as before, using Lemma 2.22 instead of Lemma 2.21, to obtain that $\oplus\mathsf{P} \subseteq \mathsf{BPP}^{\mathcal{R}}$. If, in addition, $\mathcal{R} \in \mathsf{PH}$, then, by Toda's Theorem [Tod91], $\mathsf{PH} \subseteq \mathsf{BPP}^{\oplus\mathsf{P}}$, and hence $\mathcal{R} \in \mathsf{BPP}^{\oplus\mathsf{P}} \subseteq \mathcal{C}\text{-}\mathsf{SIZE}[\mathsf{poly}]$. The rest of the argument follows as above.

For $\Gamma = \mathsf{P}^{\#\mathsf{P}}$, we argue as before using Lemma 2.20 instead of 2.21, with the additional observation that in this case the permanent has small circuits from $\mathcal{C}$. The only difference is that in this case the function has multiple inputs. For more details we refer the reader to [IW98]. ☐

3.2 Circuit lower bounds for MCSP-oracle classes: Proofs of Theorems 2–4

Theorems 2 and 3 follow from the next theorem.

Theorem 9. *For every circuit class $\mathcal{C}$ and property $\mathcal{R}$ that is strongly useful against $\mathcal{C}$, we have*

1. $\mathsf{ZPP}^{\mathcal{R}}/1 \not\subseteq \mathcal{C}\text{-}\mathsf{SIZE}[n^k]$ *and* $\mathsf{prZPP}^{\mathcal{R}} \not\subseteq \mathcal{C}\text{-}\mathsf{SIZE}[n^k]$ *for all $k \in \mathbb{N}$, and*
2. $\mathsf{ZPEXP}^{\mathcal{R}} \not\subseteq \mathcal{C}\text{-}\mathsf{SIZE}[\mathsf{poly}]$.

Proof. Assume w.l.o.g that $\mathcal{C}$ implements P and $\mathcal{R} \in \mathcal{C}\text{-}\mathsf{SIZE}[\mathsf{poly}]$ (otherwise there is nothing to prove). Consider $L = L_{\mathsf{PSPACE}}$ from Lemma 2.21. As $L \in \mathsf{PSPACE} \subseteq \mathsf{EXP}$, by a translation argument there exists $d \geq 1$ such that $L \in \mathsf{SIZE}(2^{n^d})$. Therefore, $\mathsf{s}^{\mathcal{C}}_{L}(n)$ is well-defined and in particular $\mathsf{s}^{\mathcal{C}}_{L}(n) = \mathcal{O}(2^{n^d})$.

We first prove part (1) of the theorem. We focus on the class $\mathsf{ZPP}^{\mathcal{R}}/1$; the claim about prZPP will follow by Lemma 2.9. We consider two cases:

Case 1: $\mathsf{PSPACE} \subseteq \mathcal{C}\text{-}\mathsf{SIZE}[\mathsf{poly}]$. By Theorem 1 and Lemma 2.16, $\mathsf{PSPACE} \subseteq \mathsf{BPP}^{\mathcal{R}} \subseteq \mathsf{ZPP}^{\mathcal{R}}$. Hence, by Lemma 2.11, for all $k \in \mathbb{N} : \mathsf{ZPP}^{\mathcal{R}} \not\subseteq \mathcal{C}\text{-}\mathsf{SIZE}[n^k]$.

Case 2: $\mathsf{PSPACE} \not\subseteq \mathcal{C}\text{-}\mathsf{SIZE}[\mathsf{poly}]$. As L is PSPACE-complete, we have that $L \notin \mathcal{C}\text{-}\mathsf{SIZE}[\mathsf{poly}]$. Assume towards contradiction that $\mathsf{BPP}^{\mathcal{R}}/1 \subseteq \mathcal{C}\text{-}\mathsf{SIZE}[n^k]$, for some $k \in \mathbb{N}$. By Lemma 2.8, $L'_{k,\mathcal{C}} \in \mathcal{C}\text{-}\mathsf{SIZE}[n^k]$. And thus, by Lemma 2.7, $\mathsf{s}^{\mathcal{C}}_{L}(n) = \mathcal{O}(n^{2k})$. This contradicts the assumption that $L \notin \mathcal{C}\text{-}\mathsf{SIZE}[\mathsf{poly}]$. As in Lemma 2.16, we obtain that, for all $k \in \mathbb{N}$, $\mathsf{ZPP}^{\mathcal{R}}/1 \not\subseteq \mathcal{C}\text{-}\mathsf{SIZE}[n^k]$.

Part (2) of the theorem is also shown by considering two cases:

Case 1: $\mathsf{PSPACE} \subseteq \mathcal{C}\text{-}\mathsf{SIZE}[\mathsf{poly}]$. As above, $\mathsf{PSPACE} \subseteq \mathsf{ZPP}^{\mathcal{R}}$. By a translation argument, $\mathsf{EXPSPACE} \subseteq \mathsf{ZPEXP}^{\mathcal{R}}$. By Lemma 2.11, $\mathsf{ZPEXP}^{\mathcal{R}} \not\subseteq \mathcal{C}\text{-}\mathsf{SIZE}[\mathsf{poly}]$.

Case 2: $\mathsf{PSPACE} \not\subseteq \mathcal{C}\text{-}\mathsf{SIZE}[\mathsf{poly}]$. Since $\mathsf{PSPACE} \subseteq \mathsf{EXP} \subseteq \mathsf{ZPEXP}^{\mathcal{R}}$, the theorem follows. ☐

We now prove Theorem 4, which we re-state below.

Theorem 10 (Theorem 4 re-stated)**.** *We have* $\bigcap_B \mathsf{BPP}^{\mathsf{MCSP}^B}/1 \not\subseteq \mathcal{C}\text{-}\mathsf{SIZE}[n^k]$ *and* $\bigcap_B \mathsf{prBPP}^{\mathsf{MCSP}^B} \not\subseteq \mathsf{SIZE}(n^k)$ *for all* $k \in \mathbb{N}$*, and* $\bigcap_B \mathsf{BPEXP}^{\mathsf{MCSP}^B} \not\subseteq \mathsf{P/poly}$*.*

Proof. As in the proof of Theorem 9, we consider two cases:

Case 1: $\mathsf{PSPACE} \subseteq \mathsf{P/poly}$. Let L_{PSPACE} be the language from Lemma 2.21. Observe that for every language B, we have $L_{\mathsf{PSPACE}} \in \mathsf{P^B/poly}$. By Lemma 3.6 (3), $\mathsf{PSPACE} \subseteq \mathsf{BPP}^{\mathsf{MCSP}^B}$ for all B. By Lemma 2.11, the required circuit lower bound follows.

Case 2: $\mathsf{PSPACE} \not\subseteq \mathsf{P/poly}$. For each $k \in \mathbb{N}$ there exists $L'_k \notin \mathsf{SIZE}(n^k)$ such that $L'_k \in \mathsf{BPP}^{\mathsf{MCSP}^B}/1$ for all B. □

3.3 Hardness of relativized versions of MCSP: Proofs of Theorems 5–7

First we observe that, for every oracle B, there is a $\mathsf{P}^{\mathsf{MCSP}^B}$-natural property for B-oracle circuits. Combined with Lemma 3.1, this yields the following theorem along the lines of Theorem 3.3.

Theorem 3.5. *For every oracle B the class of B-oracle circuits is* PAC *learnable under the uniform distribution, using membership and* MCSP^B *queries with hypotheses being* MCSP^B*-oracle circuits.*

Lemma 3.6. *Let A, B be two oracles (languages) such that $A \in \mathsf{P^B/poly}$. Then:*

1. *For every $n \in \mathbb{N}$ and $\delta > 0$, there exists a* MCSP^B*-oracle circuit C of size* $\mathsf{poly}(n, 1/\delta)$ *such that* $\Delta(C, A|_n) \leq \delta$*.*
2. *If, in addition, A is self-correctable then* $A \in \mathsf{P^{MCSP^B}/poly}$*.*
3. *If, in addition to the above, A is downward self-reducible, then* $A \in \mathsf{BPP}^{\mathsf{MCSP}^B}$*.*

Proof.
1. By the assumption, for every $n \in \mathbb{N}$ the function $A|_n(x)$ has a B-oracle circuit of size $\mathsf{poly}(n)$. Therefore, by Theorem 3.5, given oracle access to $A|_n$, the learning algorithm produces an MCSP^B-oracle circuit C of size $\mathsf{poly}(n, 1/\delta)$ such that $\Delta(C, B|_n) \leq \delta$.
2. Follows from Lemma 2.24.
3. Follows by combining Theorem 3.5 with Lemma 2.26.

□

Remark 3.7. *In the lemma above, although the learning algorithm actually needs an oracle access to $A|_n$ to produce C, in parts 1 and 2 we are only interested in mere existence. In part 3, on the other hand, we actually benefit from the learning algorithm.*

We are now ready to give the proofs of Theorems 5–7. For convenience, we re-state them below.

Theorem 11 (Theorem 5 re-stated)**.**
1. $\mathsf{PSPACE} \subseteq \mathsf{ZPP}^{\mathsf{MCSP}^{\mathsf{PSPACE}}}$ [ABK+06]
2. $\oplus\mathsf{P} \subseteq \mathsf{ZPP}^{\mathsf{MCSP}^{\oplus\mathsf{P}}}$
3. $\mathsf{P}^{\#\mathsf{P}} \subseteq \mathsf{BPP}^{\mathsf{MCSP}^{\#\mathsf{P}}}$

4. $\mathsf{PP} \subseteq \mathsf{BPP}^{\mathsf{MCSP}^{\mathsf{PP}}}$. *Moreover, for $k \geq 2$: $C_k\mathsf{P} \subseteq C_{k-1}\mathsf{P}^{\mathsf{MCSP}^{\mathsf{PP}}}$.*

Proof. 1. Consider any PSPACE-complete language B. Let L_{PSPACE} be the language from Lemma 2.21. By Lemma 3.6 (3), $L_{\mathsf{PSPACE}} \in \mathsf{BPP}^{\mathsf{MCSP}^B}$, and hence $\mathsf{PSPACE} \subseteq \mathsf{BPP}^{\mathsf{MCSP}^B}$. Observe that

$$\mathsf{MCSP}^B \in \mathsf{NP}^B \subseteq \mathsf{PSPACE}^B = \mathsf{PSPACE},$$

and so $\mathsf{MCSP}^B \in \mathsf{P}^{\mathsf{B}}/\mathsf{poly}$. By Lemma 2.13, we conclude that $\mathsf{BPP}^{\mathsf{MCSP}^B} = \mathsf{ZPP}^{\mathsf{MCSP}^B}$.

2. Arguing as above, using Lemma 2.22 instead of Lemma 2.21, we obtain $\oplus\mathsf{P} \subseteq \mathsf{BPP}^{\mathsf{MCSP}^{\oplus\mathsf{P}}}$. By Toda's Theorem [Tod91], $\mathsf{NP}^{\oplus\mathsf{P}} \subseteq \mathsf{BPP}^{\oplus\mathsf{P}}$. Therefore, we get

$$\mathsf{MCSP}^{\oplus\mathsf{P}} \in \mathsf{NP}^{\oplus\mathsf{P}} \subseteq \mathsf{BPP}^{\oplus\mathsf{P}} \subseteq \mathsf{P}^{\oplus\mathsf{P}}/\mathsf{poly}.$$

The rest of the argument is the same as above.

3. Similar to the first proof, using Lemma 2.20 instead of Lemma 2.21.

4. Since $\mathsf{P}^{\#\mathsf{P}} = \mathsf{P}^{\mathsf{PP}}$ [Tod91], we get that $\mathsf{PP} \subseteq \mathsf{P}^{\#\mathsf{P}} \subseteq \mathsf{BPP}^{\mathsf{MCSP}^{\mathsf{PP}}}$.

5. $\mathsf{PP} \subseteq \mathsf{BPP}^{\mathsf{MCSP}^{\mathsf{PP}}}$. The second part of the claim follow by induction since $C_k\mathsf{P} = \mathsf{PP}^{C_{k-1}\mathsf{P}}$. $\square$

Theorem 12 (Theorem 6 re-stated)**.** *For any language B, $n \in \mathbb{N}$ and $\delta > 0$, there exists a MCSP^B-oracle circuit C of size $\mathsf{poly}(n, 1/\delta)$ that is $1-\delta$ close to $B|_n$. If, in addition, B is self-correctable then B has polynomial size MCSP^B-oracle circuits.*

Proof. The proof follows from Lemma 3.6 for $A = B$, since $B \in \mathsf{P}^{\mathsf{B}}/\mathsf{poly}$. $\square$

Theorem 13 (Theorem 7 re-stated)**.** *Let B be a language such that PSPACE^B has polynomial size B-oracle circuits. Then B has polynomial-size MCSP^B-oracle circuits.*

Proof. Let $A \in \mathsf{PSPACE}^B$ be a language such that B is $(1/\mathsf{poly}(n), A)$-correctable with $r(n) = \mathsf{poly}(n)$, as guaranteed by Theorem 2.25. By assumption, $A \in \mathsf{PSPACE}^B \subseteq \mathsf{P}^{\mathsf{B}}/\mathsf{poly}$. By Lemma 3.6 (1), for every $n \in \mathbb{N}$, there exists an MCSP^B-oracle circuit C_n of size $\mathsf{poly}(n)$ such that $\Delta(C_n, A|_{r(n)}) \leq 1/\mathsf{poly}(n)$. Lemma 2.24 completes the proof. $\square$

4 Open questions

Some of our hardness results for the relativized MCSP (Theorem 5) are for ZPP reductions, while others for BPP reductions. Is it possible to replace the BPP reductions with ZPP reductions? We have shown it for PSPACE and $\oplus\mathsf{P}$, but not for $\#\mathsf{P}$.

Our circuit lower bound $\mathsf{ZPEXP}^{\mathsf{MCSP}} \not\subseteq \mathsf{P}/\mathsf{poly}$ shows that MCSP is powerful enough to get a hard function (and that the full power of SAT is not needed). It would be interesting to see more examples of complexity results proved with the SAT oracle that remain true when SAT is replaced with MCSP. For example, is it true that if $\mathsf{SAT} \in \mathsf{P}/\mathsf{poly}$, then SAT circuits can be found by a $\mathsf{ZPP}^{\mathsf{MCSP}}$ algorithm (strengthening the $\mathsf{ZPP}^{\mathsf{NP}}$ result by [BCG+96, KW98])? Probably a simpler question along these lines is: Does $\mathsf{SAT} \in \mathsf{P}/\mathsf{poly}$ imply that $\mathsf{NP} \subseteq \mathsf{ZPP}^{\mathsf{MCSP}}$?

We proved that, under some assumptions, every language L is computable by a polynomial-size circuit with MCSP^L oracle gates (Theorem 7). Is it true without any assumptions?

The main open question is, of course, to determine the complexity of MCSP. The results in this paper may be interpreted as giving some hope that hardness of MCSP is possible to prove under randomized Turing reductions, as we see a growing list of non-trivial computational tasks that can be solved with the help of the MCSP oracle.

Acknowledgments. We thank Eric Allender and Scott Aaronson for answering our questions and many useful conversations.

References

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

[ABK+06] E. Allender, H. Buhrman, M.l Koucký, D. van Melkebeek, and D. Ronneburger. Power from random strings. *SIAM J. Comput.*, 35(6):1467–1493, 2006.

[AD14] Eric Allender and Bireswar Das. Zero knowledge and circuit minimization. In Erzsébet Csuhaj-Varjú, Martin Dietzfelbinger, and Zoltán Ésik, editors, *Mathematical Foundations of Computer Science 2014 - 39th International Symposium, MFCS 2014, Budapest, Hungary, August 25-29, 2014. Proceedings, Part II*, volume 8635 of *Lecture Notes in Computer Science*, pages 25–32. Springer, 2014.

[Adl78] L. M. Adleman. Two theorems on random polynomial time. In *Proceedings of the 19th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 75–83, 1978.

[AHK15] E. Allender, D. Holden, and V. Kabanets. The minimum oracle circuit size problem. In *32nd International Symposium on Theoretical Aspects of Computer Science, STACS 2015, March 4-7, 2015, Garching, Germany*, pages 21–33, 2015.

[AHM+08] E. Allender, L. Hellerstein, P. McCabe, T. Pitassi, and M. E. Saks. Minimizing disjunctive normal form formulas and AC^0 circuits given a truth table. *SIAM J. Comput.*, 38(1):63–84, 2008.

[Bar02] B. Barak. A probabilistic-time hierarchy theorem for ”slightly non-uniform” algorithms. In *RANDOM*, pages 194–208, 2002.

[BCG+96] Nader H. Bshouty, Richard Cleve, Ricard Gavaldà, Sampath Kannan, and Christino Tamon. Oracles and queries that are sufficient for exact learning. *J. Comput. Syst. Sci.*, 52(3):421–433, 1996.

[BF90] D. Beaver and J. Feigenbaum. Hiding instances in multioracle queries. In *STACS*, pages 37–48, 1990.

[BFL91] L. Babai, L. Fortnow, and C. Lund. Non-deterministic exponential time has two-prover interactive protocols. *Computational Complexity*, 1:3–40, 1991.

[BFNW93] L. Babai, L. Fortnow, N. Nisan, and A. Wigderson. BPP has subexponential time simulations unless EXPTIME has publishable proofs. *Computational Complexity*, 3:307–318, 1993.

[BFT98] H. Buhrman, L. Fortnow, and T. Thierauf. Nonrelativizing separations. In *Proceedings of the 13th Annual IEEE Conference on Computational Complexity (CCC)*, pages 8–12, 1998.

[BH92] H. Buhrman and S. Homer. Superpolynomial circuits, almost sparse oracles and the exponential hierarchy. In *Foundations of Software Technology and Theoretical Computer Science, 12th Conference, New Delhi, India, December 18-20, 1992, Proceedings*, pages 116–127, 1992.

[CIKK16] M. L. Carmosino, R. Impagliazzo, V. Kabanets, and A. Kolokolova. Learning algorithms from natural proofs. In *31st Conference on Computational Complexity, CCC*, pages 1–24, 2016.

[FF93] J. Feigenbaum and L. Fortnow. Random-self-reducibility of complete sets. *SIAM J. on Computing*, 22(5):994–1005, 1993.

[FK09] L. Fortnow and A. R. Klivans. Efficient learning algorithms yield circuit lower bounds. *J. Comput. Syst. Sci.*, 75(1):27–36, 2009.

[FS04] L. Fortnow and R. Santhanam. Hierarchy theorems for probabilistic polynomial time. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 316–324, 2004.

[GZ11] O. Goldreich and D. Zuckerman. Another proof that BPP $\subseteq$ PH (and more). *Studies in Complexity and Cryptography*, pages 40–53, 2011.

[Hel86] H. Heller. On relativized exponential and probabilistic complexity classes. *Information and Control*, 71(3):231–243, 1986.

[HP15] J. M. Hitchcock and A. Pavan. On the NP-completeness of the minimum circuit size problem. In *35th IARCS Annual Conference on Foundation of Software Technology and Theoretical Computer Science, FSTTCS*, pages 236–245, 2015.

[HW16] Sh. Hirahara and O. Watanabe. Limits of minimum circuit size problem as oracle. In *31st Conference on Computational Complexity, CCC*, pages 18:1–18:20, 2016.

[IKW02] R. Impagliazzo, V. Kabanets, and A. Wigderson. In search of an easy witness: Exponential time vs. probabilistic polynomial time. *J. of Computer and System Sciences*, 65(4):672–694, 2002.

[IW97] R. Impagliazzo and A. Wigderson. P=BPP unless E has subexponential circuits: Derandomizing the XOR lemma. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing (STOC)*, pages 220–229, 1997.

[IW98] R. Impagliazzo and A. Wigderson. Randomness vs. time: De-randomization under a uniform assumption. In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 734–743, 1998.

[Kar85] Richard M. Karp. Turing award lecture. In Bill Healy and Judith D. Schlesinger, editors, *Proceedings of the 1985 ACM annual conference on The range of computing: mid-80's perspective: mid-80's perspective, Denver, Colorado, USA, October 14-16, 1985*, page 193. ACM, 1985.

[KC00] V. Kabanets and J.-Y. Cai. Circuit minimization problem. In *Proceedings of the 32nd Annual ACM Symposium on Theory of Computing (STOC)*, pages 73–79, 2000.

[KKO13] A. Klivans, P. Kothari, and I. Oliveira. Constructing hard functions from learning algorithms. In *Proceedings of the 28th Annual IEEE Conference on Computational Complexity (CCC)*, pages 86–97, 2013.

[KL80] R. M. Karp and R. J. Lipton. Some connections between nonuniform and uniform complexity classes. In *Proceedings of the 12th Annual ACM Symposium on Theory of Computing, April 28-30, 1980, Los Angeles, California, USA*, pages 302–309, 1980.

[KM02] A. Klivans and D. van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. *SIAM J. Comput.*, 31(5):1501–1526, 2002.

[KW98] Johannes Köbler and Osamu Watanabe. New collapse consequences of NP having small circuits. *SIAM J. Comput.*, 28(1):311–324, 1998.

[LFKN92] C. Lund, L. Fortnow, H. Karloff, and N. Nisan. Algebraic methods for interactive proof systems. *JACM*, 39(4):859–868, 1992.

[MP07] D. van Melkebeek and K. Pervyshev. A generic time hierarchy with one bit of advice. *Computational Complexity*, 16(2):139–179, 2007.

[MW15] C. D. Murray and R. R. Williams. On the (non) NP-hardness of computing circuit complexity. In *30th Conference on Computational Complexity, CCC*, pages 365–380, 2015.

[NW94] N. Nisan and A. Wigderson. Hardness vs. randomness. *J. Comput. Syst. Sci.*, 49(2):149–167, 1994.

[OS16] I. C. Oliveira and R. Santhanam. Conspiracies between learning algorithms, circuit lower bounds and pseudorandomness. *CoRR*, abs/1611.01190, 2016.

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

[San09] R. Santhanam. Circuit lower bounds for Merlin–Arthur classes. *SIAM J. Comput.*, 39(3):1038–1061, 2009.

[STV01] M. Sudan, L. Trevisan, and S. P. Vadhan. Pseudorandom generators without the XOR lemma. *J. Comput. Syst. Sci.*, 62(2):236–266, 2001.

[Tod91] S. Toda. PP is as hard as the polynomial time hierarchy. *SIAM J. on Computing*, 20(5):865–877, 1991.

[Tra84] Boris A. Trakhtenbrot. A survey of russian approaches to perebor (brute-force searches) algorithms. *IEEE Annals of the History of Computing*, 6(4):384–400, 1984.

[TV07] L. Trevisan and S. P. Vadhan. Pseudorandomness and average-case complexity via uniform reductions. *Computational Complexity*, 16(4):331–364, 2007.

[Uma03] C. Umans. Pseudo-random generators for all hardnesses. *J. of Computer and System Sciences*, 67(2):419–440, 2003.

[Val79] L. G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189–201, 1979.

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

[Vol14] I. Volkovich. On learning, lower bounds and (un)keeping promises. In *Proceedings of the 41st ICALP*, pages 1027–1038, 2014.

A Self-correctable and downward-reducible $\oplus\mathsf{P}$-complete problem

In this section we prove Lemma 2.22

Lemma A.1. *There is a downward self-reducible and self-correctable $\oplus\mathsf{P}$-complete language $L_{\oplus\mathsf{P}}$.*

Proof sketch. The proof is very similar to the one in [TV07] for the case of PSPACE. We define a formula $\Phi_n(\bar{x}, \bar{y})$ that is universal for n-variate 3-cnf formulas on the variables $\bar{x} = (x_1, \ldots, x_n)$, where $\bar{y} = (y_1, \ldots, y_{8n^3})$ describes a particular 3-cnf formula ϕ by specifying, for each possible clause on 3 variables, whether this clause is present in ϕ. For example, if $c_1, \ldots, c_m$, for $m = 8n^3$, is a sequence of all possible 3-clauses on n variables $x_1, \ldots, x_n$, we can define Φ as follows:

$$\Phi(\bar{x}, \bar{y}) = \wedge_{i=1}^{m}((y_i \wedge c_i) \vee \neg y_i).$$

We now "arithmetize" the formula Φ, getting a polynomial that agrees with Φ over all Boolean inputs. We will work over the finite field $\mathbb{F}_{2^k}$ of characteristic 2, for $k = 5 \log n$. Arithmetizing all clauses c_i's (by replacing each c_i with a degree 3 multilinear polynomial c'_i, in the same 3 variables, that agrees with c_i over Boolean all assignments), we get the following arithmetization Φ' of Φ:

$$\Phi'(\bar{x}, \bar{y}) = \prod_{i=1}^{m}(y_i \cdot c'_i + 1 + y_i).$$

For each $0 \leq i \leq n$, define a polynomial

$$f_{n,i}(x_1, \ldots, x_i, \bar{y}) = \sum_{x_{i+1} \in \{0,1\}} \cdots \sum_{x_n \in \{0,1\}} \Phi'(\bar{x}, \bar{y}),$$

where the summation is over our field $\mathbb{F}_{2^k}$ of characteristic 2. Note that $f_{n,0}(\bar{y})$, for a Boolean $\bar{y}$, is exactly $\oplus\mathsf{SAT}$ on the 3-cnf instance described by $\bar{y}$. So $f_{n,0}$ is $\oplus\mathsf{P}$-hard to compute.

We have that $f_{n,i}$ can be expressed in terms of $f_{n,i+1}$, for $i < n$, by the formula:

$$f_{n,i}(x_1, \ldots, x_i, \bar{y}) = f_{n,i+1}(x_1, \ldots, x_i, 0) + f_{n,i+1}(x_1, \ldots, x_i, 1).$$

So $f_{n,i}$ can be computed in polynomial time with oracle access to $f_{n,i+1}$. It is also clear that $f_{n,n}$ can be evaluated in polynomial time (directly).

Next, in the same way as in [TV07], we define a Boolean function family $F = \{F_t\}_{t\geq 1}$ so that each $f_{n,i}$ is "embedded" into some $F_{h(n,i)}$, for some function $h\colon \mathbb{N}\times\mathbb{N}\to\mathbb{N}$. Namely, h can be chosen so that

- $h(n,i) > h(n,i+1)$ (and so we have downward-reducibility for $f_{n,i}$'s), and
- the length $h(n,i)$ is large enough to accommodate both an input to $f_{n,i}$ and an index $j\in[k]$.

We then define

$$F_{h(n,i)}(x_1,\dots,x_i,\bar{y},j) = f_{n,i}(x_1,\dots,x_i,\bar{y})_j,$$

i.e., the jth bit of the value of $f_{n,i}$ in the field $\mathbb{F}_{2^k}$, whose elements are viewed as k-bit vectors.

The downward-reducibility of F follows from the properties of $f_{n,i}$ (and the way we arranged the lengths $h(n,i)$). The self-correctability of F follows from the fact each $f_{n,i}$ is a $O(n^3)$-degree polynomial over the field of size $2^k\geq n^5$ (see [TV07] for more details). The $\oplus\mathsf{P}$-hardness of F follows from $\oplus\mathsf{P}$-hardness of $f_{n,0}$'s.

It remains to show that $F\in\oplus\mathsf{P}$. Note that every bit j of the value of $f_{n,n}(\bar{y})$, for every input $\bar{y}$, is computable in P, and hence also in $\oplus\mathsf{P}$. For any $0\leq i<n$, the jth bit of $f_{n,i}(x_1,\dots,x_i,\bar{y})$ can be computed in $\oplus\mathsf{P}$ using the following nondeterministic algorithm:

> "Nondeterministically guess Boolean values $b_{i+1},\dots,b_n$. Compute the value
>
> $$v = \Phi'(x_1,\dots,x_i,b_{i+1},\dots,b_n,\bar{y}).$$
>
> Accept if the jth bit of the computed field element v is 1, and reject otherwise."

The parity of the number of accepting paths of the algorithm above is exactly the sum modulo 2 of the bits v_j, over all Boolean assignments to $x_{i+1},\dots,x_n$. The latter is exactly the jth bit of $f_{n,i}$ because addition in the field $\mathbb{F}_{2^k}$ is the bit-wise XOR of the corresponding k-bit vectors. □

B Oracles B where $\mathsf{PSPACE}^B\subseteq\mathsf{P}^{\mathsf{B}}/\mathsf{poly}$ but $\mathsf{PSPACE}^B\neq\mathsf{P}^B$

Lemma B.1. *Let B be a language such that $\mathsf{EXP}^B\subseteq\mathsf{P}^{\mathsf{B}}/\mathsf{poly}$. Then $\mathsf{PSPACE}^B\neq\mathsf{P}^B$.*

Proof. Assume the contrary. By Meyer's Theorem [KL80]: $\mathsf{EXP}^B\subseteq\Sigma_2^B\subseteq\mathsf{PSPACE}^B$. By the assumption, $\mathsf{EXP}^B\subseteq\mathsf{PSPACE}^B\subseteq\mathsf{P}^B$ which contradicts Time Hierarchy Theorem. □

There are numerous examples of languages satisfying the preconditions of the Lemma; see, e.g., [Hel86, BFT98].

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il