Electronic Colloquium on Computational Complexity, Revision 5 of Report No. 3 (2018)

Proving that $prBPP = prP$ is as hard as proving that "almost NP" is not contained in P/poly*

Roei Tell †

November 7, 2019

Abstract

What circuit lower bounds are *necessary* in order to prove that *promise-*$\mathcal{BPP}$ = *promise-*$\mathcal{P}$? We show that the recent breakthrough result of Murray and Williams (STOC 2018) can be used to show a dramatic strengthening of the previously-known answer to this question. Specifically, we show that if *promise-*$\mathcal{BPP}$ = *promise-*$\mathcal{P}$, then $NTIME[n^{f(n)}] \not\subseteq \mathcal{P}/\text{poly}$, for essentially any $f(n) = \omega(1)$.

We also prove a technical strengthening of this result. Specifically, we show that if *promise-*$\mathcal{BPP}$ = *promise-*$\mathcal{P}$, then for essentially any $s : \mathbb{N} \to \mathbb{N}$ it holds that $NTIME[s^{O(1)} \circ s^{O(1)}] \not\subseteq SIZE[s]$. Moreover, we show that size-s circuits fail to compute the "hard" function in any interval of length $\text{poly}(s(\text{poly}(n)))$. The proof of this result uses tools of Murray and Williams, but relies on a different proof strategy. (Their proof strategy yields three compositions of s instead of two, and does not yield the guarantee of failure in any small interval.)

Lastly, we present an alternative proof of the main result, which only relies on a generalization of the well-known lower bound of Santhanam (SICOMP, 2009).

*The current paper is a revised version of a technical report that appeared online under a slightly different title (ECCC, TR18-003). The current version includes additional results and a revised exposition.

†Department of Computer Science and Applied Mathematics, Weizmann Institute of Science, Rehovot, Israel. Email: roei.tell@weizmann.ac.il

ISSN 1433-8092

Contents

1 Introduction

The $\mathcal{BPP} = \mathcal{P}$ conjecture asserts that any decision problem that can be efficiently solved using randomness (while allowing for a small error) can also be efficiently solved deterministically. In other words, the conjecture asserts that randomness is not needed to efficiently solve decision problems. This conjecture is central to the complexity-theoretic study of the role of randomness in computation.

The $\mathcal{BPP} = \mathcal{P}$ conjecture is often interpreted as an *algorithmic* problem, namely the problem of explicitly constructing efficient deterministic algorithms that simulate randomized algorithms. In fact, the stronger conjecture that $pr\mathcal{BPP} = pr\mathcal{P}$ (i.e., that the "promise problem" versions of $\mathcal{BPP}$ and of $\mathcal{P}$ are equal) is *equivalent* to the conjectured existence of an algorithm for a single, specific problem (i.e., the *circuit acceptance probability problem*; see Proposition 6). In this paper our focus will indeed be on the stronger conjecture that $pr\mathcal{BPP} = pr\mathcal{P}$.

In addition to the equivalence of $pr\mathcal{BPP} = pr\mathcal{P}$ to the existence of an algorithm (for the *CAPP* problem), it has also been known for at least two decades that the conjecture is intimately related to *circuit lower bounds*; that is, to lower bounds for non-uniform models of computation. Specifically:

- On the one hand, *any proof of sufficiently strong circuit lower bounds would also prove that $pr\mathcal{BPP} = pr\mathcal{P}$*. Specifically, if there is a function in $\mathcal{E}$ that requires exponential-sized circuits, then $pr\mathcal{BPP} = pr\mathcal{P}$ (see [IW99], which relies on the classical hardness-randomness paradigm [Yao82; BM84; NW94]).
- On the other hand, *any proof that $pr\mathcal{P} = pr\mathcal{BPP}$ implies long-sought circuit lower bounds*. As a prominent example, if $pr\mathcal{BPP} = pr\mathcal{P}$ then there exists a function in $\mathcal{NEXP}$ that cannot be computed by any polynomial-sized circuit family [BFT98].[1] In fact, the latter circuit lower bound follows even from much weaker hypotheses, such as $\mathcal{MA} \neq \mathcal{NEXP}$ (see, e.g., [IKW02; Wil13]).

1.1 The main new result

Informally, following a very recent breakthrough by Murray and Williams [MW18], the main result in this paper considerably strengthens the known connection between the conjecture that $pr\mathcal{BPP} = pr\mathcal{P}$ and circuit lower bounds.

The starting point of the current work is the observation that an immediate corollary of a result from the recent work of Murray and Williams [MW18, Thm 1.2] is the following: *If $pr\mathcal{BPP} = pr\mathcal{P}$, then there exists a function in $NTIME[n^{\mathrm{poly}\log(n)}]$* (rather than $\mathcal{NEXP}$) *that cannot be computed by any polynomial-sized circuit family*. This is a dramatic (almost exponential) strengthening of previously-known results (i.e., of [BFT98; IKW02]), and we believe that it is a fundamental result that is worth spelling out and highlighting. Furthermore, this result can even be further strengthened. In particular,

[1] In [BFT98] it is shown, unconditionally, that $\mathcal{MAEXP} \not\subseteq \mathcal{P}/\mathrm{poly}$. Thus, under the hypothesis $pr\mathcal{BPP} \subseteq pr\mathcal{NP}$ we have that $\mathcal{MAEXP} = \mathcal{NEXP} \not\subseteq \mathcal{P}/\mathrm{poly}$ (see [IKW02, Rmk. 26]).

by using the proof approach of [MW18] while instantiating their technical tools with different parameters, we get the following:

Theorem 1 *(main theorem; informal). If $pr\mathcal{BPP} = pr\mathcal{P}$, then, for essentially any super-constant function $f(n) = \omega(1)$, there exists a set in $NTIME[n^{f(n)}] \setminus \mathcal{P}/\mathrm{poly}$.*

One might a-priori hope to strengthen the conclusion of Theorem 1 by improving the time bound in the non-deterministic class; that is, to prove that "if $pr\mathcal{BPP} = pr\mathcal{P}$, then $\mathcal{NP} \not\subseteq \mathcal{P}/\mathrm{poly}$ (and $\mathcal{P} \neq \mathcal{NP}$)". However, such a result cannot be proved without *unconditionally* proving that $\mathcal{P} \neq \mathcal{NP}$, since any proof of the conditional statement "$pr\mathcal{BPP} = pr\mathcal{P} \Longrightarrow \mathcal{P} \neq \mathcal{NP}$" would unconditionally imply that $\mathcal{P} \neq \mathcal{NP}$ (see Proposition 13). Therefore, the conclusion of Theorem 1 is optimal in this sense.

Theorem 1 is a special case of a more general "derandomization implies lower bounds" result that follows using the technical tools of Murray and Williams [MW18], and in particular their new "easy witness lemma" (see Section 2 for details on the latter). In this general result, the lower bound in the conclusion can be parameterized:

Theorem 2 *(a generalized version of Theorem 1; informal, see Corollary 16). There exists $\epsilon > 0$ such that for any time-computable $s : \mathbb{N} \to \mathbb{N}$ satisfying $n < s(n) < 2^{\epsilon \cdot n}$ it holds that*

$$pr\mathcal{BPP} = pr\mathcal{P} \Longrightarrow NTIME[s' \circ s' \circ s'] \not\subseteq SIZE[s] \,,$$

where $s' = \mathrm{poly}(s(O(n)))$.

Indeed, Theorem 1 follows as a special case of Theorem 2 by using $s(n) = n^{\omega(1)}$ (in which case $s' \circ s' \circ s' = n^{\omega(1)}$ and $SIZE[s] \supset \mathcal{P}/\mathrm{poly}$; see Corollary 18). The hypothesis of Theorem 2 can also be significantly relaxed, since its proof relies on Williams' [Wil13] celebrated proof strategy, which is well-known to support such relaxations. Recall that, loosely speaking, Williams' proof strategy shows that certain "non-trivial" circuit-analysis algorithms imply circuit lower bounds. In our case, Theorem 2 holds, for example, under the hypothesis that there exists a "non-trivial" algorithm for the Circuit Acceptance Probability Problem (CAPP) (i.e., an algorithm that approximates the acceptance probability of a circuit of size m with v variables in time $2^{.99 \cdot v} \cdot \mathrm{poly}(m)$); Theorem 2 also holds under the hypothesis that $pr\text{-}co\mathcal{RP} \subseteq pr\mathcal{NP}$. See the end of Section 4.1 for details of possible relaxations.

Theorem 2 may be compared to the following result of Kinne, van Melkebeek, and Shaltiel [KMS12, Thm. 9], which builds on the well-known result of Kabanets and Impagliazzo [KI04]: If $\mathcal{BPP} = \mathcal{P}$ (i.e., the "non-promise" version of the $\mathcal{BPP} = \mathcal{P}$ conjecture holds), then either the *permanent function* of $\{0,1\}$-matrices over $\mathbb{Z}$ does not have polynomial-sized arithmetic circuits, or for essentially any $s : \mathbb{N} \to \mathbb{N}$ it holds that $NTIME[s^{O(1)}] \not\subseteq SIZE[s]$. Indeed, in this result from [KMS12] the hypothesis is weaker than in Theorem 2, since it only refers to the "non-promise" conjecture $\mathcal{BPP} = \mathcal{P}$ (rather than to $pr\mathcal{BPP} = pr\mathcal{P}$); whereas the conclusion is not a circuit lower bound, but rather a disjunction of two circuit lower bounds.[2] Nevertheless, in the lower bound $NTIME[s^{O(1)}] \not\subseteq SIZE[s]$, the time-complexity of the "hard" function is just $s^{O(1)}$, rather than $s^{O(1)} \circ s^{O(1)} \circ s^{O(1)}$ as in Theorem 2.

[2] Similarly to Theorem 2, the result in [KMS12] also follows from the weaker hypothesis $co\mathcal{RP} \subseteq \mathcal{NP}$.

1.2 A technical strengthening of Theorem 2

Observe that the lower bound in the conclusion of Theorem 2 becomes trivial when s is half-exponential or larger (i.e., when $s(s(n)) \geq 2^n$), because there are three compositions of s in the time-bound for the "hard" function.[3] The following improvement removes this limitation: We prove that if $pr\mathcal{BPP} = pr\mathcal{P}$, then there exists a function in $NTIME[s^{O(1)} \circ s^{O(1)}]$ that cannot be computed by circuits of size s.

Moreover, we also improve the concluded lower bound by showing that size-s circuits fail to compute the "hard" function on a "dense" set of input lengths (the conclusion in Theorem 2 only guarantees failure on infinitely-many input lengths). Specifically, for $s_I(n) = \mathrm{poly}(s(\mathrm{poly}(n)))$, we conclude that size-$s$ circuits fail to compute the "hard" function on an input length in any interval of the form $[n, s_I(n)]$.

Similarly to Theorem 2, the foregoing (stronger) conclusions follow also from the hypothesis $pr\text{-}co\mathcal{RP} \subseteq pr\mathcal{NP}$ (which is weaker than $pr\mathcal{BPP} = pr\mathcal{P}$):

Theorem 3 *(strengthening the conclusion of Theorem 2; informal, see Theorem 24). There exists $\epsilon > 0$ such that for any time-computable $s : \mathbb{N} \to \mathbb{N}$ satisfying $n < s(n) < 2^{\epsilon \cdot n}$ it holds that*

$$pr\text{-}co\mathcal{RP} \subseteq pr\mathcal{NP} \Longrightarrow NTIME[s' \circ s'] \not\subseteq \mathtt{i.o.}_{[s_I]}\text{-}SIZE[s] ,$$

where $s' = \mathrm{poly}(s)$, and $\mathtt{i.o.}_{[s_I]}\text{-}SIZE[s]$ is the class of problems such that there exists a size-s circuit that, for infinitely-many intervals of length $s_I(n) = \mathrm{poly}(s(n^2))$, solves the problem on some input length in the interval.

The proof of Theorem 3 does not follow the proof approach of Murray and Williams, and in particular does not use their new "easy witness lemma". Nevertheless, the proof crucially relies on one of their technical results, namely their strengthening of Santhanam's circuit lower bound [San09]. The proof strategy that underlies Theorem 3 was suggested to us by Igor Oliveira after a preliminary version of this paper appeared online. See Section 2.2 for further details.

In Appendix A we present an alternative and relatively simple proof of a weaker form of Theorem 3, which does not include the guarantee of failure in every "small" interval. This proof does not use the results of Murray and Williams, but is based only on (a generalization of) the well-known circuit lower bound of Santhanam [San09].

1.3 The meaning of the results in this paper

What is the meaning of the statement "$pr\mathcal{BPP} = pr\mathcal{P} \Longrightarrow NTIME[n^{\omega(1)}] \not\subseteq \mathcal{P}/\mathrm{poly}$"? Note that the lower bound $NTIME[n^{\omega(1)}] \not\subseteq \mathcal{P}/\mathrm{poly}$ asserts that polynomial-sized cir-

---

Also, Jansen and Santhanam [JS12] showed how to remove the disjunction in the conclusion, at the cost of deducing a weaker lower bound: Loosely speaking, they showed that if $\mathcal{BPP} = \mathcal{P}$, then polynomial-sized arithmetic circuits cannot compute the class of polynomials such that each bit in the binary representation of the output of the polynomial can be computed in $\mathcal{NEXP} \cap co\mathcal{NEXP}$ (see [JS12] for precise details).

[3] Specifically, when $s(s(n)) \geq 2^n$ we have that $NTIME[s^{O(1)} \circ s^{O(1)} \circ s^{O(1)}] \supseteq NTIME[2^{\mathrm{poly}(s(n))}]$, whereas $DTIME[2^{\mathrm{poly}(s(n))}] \not\subseteq SIZE[s]$ holds unconditionally (by a diagonalization argument).

cuits cannot simulate *both "slightly" super-polynomial running time and non-determinism*.[4]

Thus, on the one hand, one may view the lower bound $NTIME[n^{\omega(1)}] \not\subseteq \mathcal{P}/\text{poly}$ as a weaker form of $\mathcal{NP} \not\subseteq \mathcal{P}/\text{poly}$. Hence, Theorem 1 can be interpreted as saying that proving that $pr\mathcal{BPP} = pr\mathcal{P}$ is as hard as proving a lower bound that is essentially a precursor of $\mathcal{NP} \subseteq \mathcal{P}/\text{poly}$. From this perspective the conclusion of the theorem is essentially optimal, since (as mentioned after the statement of Theorem 1), we cannot strengthen the conclusion of the theorem to $\mathcal{NP} \not\subseteq \mathcal{P}/\text{poly}$ without unconditionally proving that $\mathcal{P} \neq \mathcal{NP}$.

On the other hand, as pointed out by Ryan Williams, one can alternatively view the statement $NTIME[n^{\omega(1)}] \not\subseteq \mathcal{P}/\text{poly}$ as a weaker form of the statement $DTIME[n^{\omega(1)}] \not\subseteq \mathcal{P}/\text{poly}$. The latter statement asserts that polynomial-sized circuits cannot simulate algorithms with superpolynomial running time. From this perspective, Theorem 1 implies that proving that $pr\mathcal{BPP} = pr\mathcal{P}$ is as hard as proving a weak form of a "strengthened time-hierarchy theorem" (in which we compare uniform algorithms to non-uniform circuits).

Continuing the latter view, it seems instructive to compare the lower bounds implied by $pr\mathcal{BPP} = pr\mathcal{P}$ to the lower bounds that are known to *imply* $pr\mathcal{BPP} = pr\mathcal{P}$ (using the results of Impagliazzo and Wigderson [IW99]). Specifically, being slightly informal,[5] we have that:

$$\begin{aligned}
&\forall s(n) < 2^{\epsilon \cdot n},\ DTIME[\text{poly}(s)] \not\subseteq \texttt{i.o.}\text{-}SIZE[s] && (1)\\
&\Downarrow && \text{(by [IW99])}\\
&pr\mathcal{BPP} = pr\mathcal{P} &&\\
&\Downarrow && \text{(by Thm 3)}\\
&\forall s(n) < 2^{\epsilon \cdot n},\ NTIME[s' \circ s'] \not\subseteq \texttt{i.o.}_{[s_I]}\text{-}SIZE[s] && (2)
\end{aligned}$$

where $\epsilon > 0$, s', and s_I are defined as in Theorem 3. This comparative perspective suggests the following interpretation of the results in this paper:

> The lower bounds implied by $pr\mathcal{BPP} = pr\mathcal{P}$ are now significantly stronger (compared to the lower bounds that were previously known to be implied by this conjecture); but they are nevertheless still weaker than the lower bounds that are known to *imply* that $pr\mathcal{BPP} = pr\mathcal{P}$.

Is $pr\mathcal{BPP} = pr\mathcal{P}$ equivalent to a specific circuit lower bound? The circuit lower bounds implied in Eq. (2) hold not only when $pr\mathcal{BPP} = pr\mathcal{P}$, but also under the (intuitively) weaker hypothesis $pr\text{-}co\mathcal{RP} \subseteq pr\mathcal{NP}$. Therefore, one might suspect that the conclusion in Theorem 2 can be strengthened. Recall that the question of whether

[4] This lower bound can be viewed as a significant strengthening of the (unconditionally-known) lower bound $\Sigma_3[n^{\omega(1)}] \not\subseteq \mathcal{P}/\text{poly}$, which asserts that polynomial-sized circuits cannot simulate both super-polynomial running time and "several levels" of non-determinism/alterations. (The proof of the lower bound is a diagonalization argument a-la Kannan's theorem; see, e.g., [Juk12, Lem. 20.12].)

[5] The informality is by ignoring time-computability constraints on s.

specific derandomization results are *equivalent* to *specific* circuit lower bounds has been raised several times in the past (see, e.g., [IKW02] and [TV07, Sec. 1.1]). We thus propose the following natural conjecture (we have found no explicit prior mentions of this conjecture in the literature):

Conjecture 4 *($pr\mathcal{BPP} = pr\mathcal{P}$ is equivalent to the [IW99] lower bounds). The statement that $pr\mathcal{BPP} = pr\mathcal{P}$ is equivalent to the statement that for some $\epsilon > 0$ and every $s(n) < 2^{\epsilon \cdot n}$ it holds that $\mathcal{DTIME}[\mathrm{poly}(s)] \not\subseteq \mathtt{i.o.}\text{-}\mathcal{SIZE}[s]$.*

The most important gap between Theorem 3 and Conjecture 4 is that in Theorem 3, the lower bounds implied by $pr\mathcal{BPP} = pr\mathcal{P}$ are against *non-deterministic* classes. Note that even a modest first step towards proving Conjecture 4, namely proving that $pr\mathcal{BPP} = pr\mathcal{P} \implies \mathcal{EXP} \not\subseteq \mathcal{P}/\mathrm{poly}$, already implies that any polynomial-time derandomization of $pr\mathcal{BPP}$ requires pseudorandom generators (see [Bab+93]).

1.4 Organization

In Section 2 we present high-level overviews of the proofs of our main theorems. In Section 3 we present preliminary definitions. In Section 4 we prove Theorems 1 and 2, and in Section 5 we prove Theorem 3.

2 Overviews of the proofs

In Section 2.1 we present an overview of the proof of Theorem 2, and in Section 2.2 we present an overview of the proof of Theorem 3. Since the proof approaches for the two theorems are very different, one may read Section 2.2 without first reading Section 2.1.

2.1 Proof overview for Theorem 2

The proof of Theorem 2 follows the approach used by Murray and Williams [MW18], which is based on the celebrated proof strategy of Williams [Wil13]. The main new component in [MW18] is a new "easy witness lemma", which allows for flexible scaling of the parameters in the original proof strategy of Williams (see below; this new lemma improves the original easy witness lemma of [IKW02]). Murray and Williams stated consequences with two *specific parameter settings*. We extend their result by stating a *general* (parametrized) "derandomization implies lower bounds" result that uses this proof approach with the new easy witness lemma (see Theorem 15), and deduce Theorems 1 and 2 as special cases.

Let us now overview the proof of Theorem 2. The point of the overview is to describe how the (well-known) proof strategy of Williams can be instantiated with the new easy witness lemma for general parameters in order to deduce Theorem 2. The starting point for the proof is the Circuit Acceptance Probability Problem (or CAPP, in short): Given as input the description of a Boolean circuit C, the problem is to distinguish between the case that the acceptance probability of C is at least $2/3$ and the case that the acceptance probability of C is at most $1/3$. It is well-known that a deterministic polynomial-time algorithm for CAPP exists if and only if $pr\mathcal{BPP} = pr\mathcal{P}$ (see Proposition 6). The current argument relies on the much weaker hypothesis that CAPP for circuits of size m with v input variables can be solved in time $2^{.99\cdot v}\cdot \mathrm{poly}(m)$; for simplicity, let us assume that the CAPP algorithm runs in time $2^{.99\cdot v}\cdot m^2$.

Fix any time-computable function $n < s(n) < 2^{\epsilon\cdot n}$, where $\epsilon > 0$ is a universal constant. Denoting $t = s^{O(1)} \circ s^{O(1)} \circ s^{O(1)}$, our goal is to prove that $NTIME[t] \not\subset SIZE[s]$. (The definition of t in this high-level overview is slightly informal; see Definition 9 and Corollary 16 for precise details.) To do so, assume towards a contradiction that $NTIME[t] \subseteq SIZE[s]$, and let $t_0(n) = t(n)^{\delta}$, where $\delta > 0$ is sufficiently small. We will construct, for any $L \in NTIME[t_0]$, a non-deterministic machine that decides L in time $t_0^{1-\Omega(1)}$; this will contradict the non-deterministic time hierarchy [Coo72].

The new easy witness lemma asserts that if $NTIME[t] \subseteq SIZE[s]$ where $t = t_0^{1/\delta}$, then for every $L' \in NTIME[(t_0)^2]$, every $(t_0)^2$-time verifier V for L' and every $x \in L'$, there exists a circuit $P_x \in SIZE[t_0^{.001}]$ that encodes a witness π_x such that $V(x,\pi_x)$ accepts.[6] (Again, our parameters in the overview are informal; see Lemma 10 for a statement that uses precise parameters.) The point is that witnesses for the verifier V are a-priori of size $(t_0)^2$, but the lemma asserts that (under the hypothesis) every $x \in L'$ has a witness that can be concisely represented by a circuit of much smaller size $t_0^{.001}$. We note that the main "bottleneck" in the proof that requires using $t = s^{O(1)} \circ s^{O(1)} \circ s^{O(1)}$ (rather than, say, $t = \mathrm{poly}(s)$) is the new "easy witness lemma".

Let us now construct the non-deterministic machine for $L \in NTIME[t_0]$, relying on the existence of the foregoing "compressible" witnesses. We first fix a PCP system for L with a verifier V that runs in time $t_V = \mathrm{poly}(n, \log(t_0))$ and uses $\ell = \log(t_0) + O(\log\log(t_0))$ random bits. (For concreteness, we use the PCP of Ben-Sasson and Viola [BSV14], but previous ones such as [BGH+05] also suffice for the proof.) Using the new easy witness lemma, for every $x \in L$ there exists a circuit of size $t_0(|x|)^{.001}$ that encodes a valid proof for x in this PCP system.[7]

Now, given input $x \in \{0,1\}^n$, the non-deterministic machine M first guesses a circuit P_x of size $t_0(n)^{.001}$, in the hope that such a circuit encodes a valid proof for x. Then, the machine constructs a circuit $C_x^{P_x}$ that, when given $r \in \{0,1\}^{\ell}$ as input, simulates the execution of V on x using randomness r when V is given oracle access to the witness P_x (i.e., $C_x^{P_x}(r) = V^{P_x}(x,r)$). Finally, the machine M uses the CAPP algorithm on the circuit $C_x^{P_x}$ to determine whether the verifier is accepts x with high probability over r or rejects x with high probability over r.

Note that if $x \in L$, then for *some* guess of P_x it holds that $C_x^{P_x}$ has acceptance probability one, and thus the machine M will accept x. On the other hand, if $x \notin L$, then for *any* guess of P_x it holds that $C_x^{P_x}$ has low acceptance probability (corresponding

---

[6] A circuit $P_x : \{0,1\}^{\log(|\pi_x|)} \to \{0,1\}$ encodes a string π_x if for every $i \in [|\pi_x|]$ it holds that $P_x(i)$ is the i^{th} bit of π_x (equivalently, π_x is the truth-table of P_x).

[7] To apply the easy witness lemma, consider the deterministic verifier V' that, when given input and a proof, enumerates the random coins of V and decides by a majority vote. This verifier runs in time $2^{\ell}\cdot t_V < (t_0)^2$, so we can apply the lemma to L with this verifier.

to the soundness of the PCP verifier), and thus the machine M will reject x.

The point is that all the operations of the machine happened in time much shorter than $t_0(n)$. Specifically, the size of P_x is $t_0(n)^{.001}$, and the size of $C_x^{P_x}$ is $m < t_V(n) \cdot t_0(n)^{.001} < t_0(n)^{.002}$; thus, guessing P_x and constructing $C_x^{P_x}$ can be done in time $\text{poly}(m) \ll \sqrt{t_0(n)}$. Now, note that $C_x^{P_x}$ has $\ell = \log(t_0) + O(\log\log(t_0))$ variables; thus, when the CAPP algorithm is given $C_x^{P_x}$ it runs in time

$$2^{.99 \cdot \ell} \cdot m^2 < t_0(n)^{.995} \cdot \left(t_0(n)^{.002}\right)^2 = (t_0(n))^{1-\Omega(1)} ,$$

and we get a contradiction.

As mentioned in the introduction, the hypothesis in this proof strategy can be further relaxed in various (known) ways. For details of these relaxations, see the statement of Theorem 15 and the remark following the theorem's proof.

2.2 Proof overview for Theorem 3

The proof of Theorem 3 is very different than the proof of Theorem 2, and in particular does *not* rely on the proof strategy of Williams [Wil13] or on an "easy witness lemma". As a first step, let us prove a statement that is weaker than that of Theorem 3: We prove that if $pr\text{-}co\mathcal{RP} \subseteq pr\mathcal{NP}$, then for essentially any $s : \mathbb{N} \to \mathbb{N}$ it holds that $NTIME[s \circ s] \not\subseteq SIZE[s]$ (without claiming that failure happens in every "small" interval).

Recall that a standard approach to prove "derandomization implies lower bounds" theorems is to rely on unconditional lower bounds for $\mathcal{MA}$ protocols: Specifically, if we start from a lower bound $MATIME[t] \not\subseteq SIZE[s]$, for some $t > s$, and assume a derandomization hypothesis $MATIME[t] \subseteq NTIME[t']$, for some $t' \geq t$, then under the derandomization hypothesis we have that $NTIME[t'] \not\subseteq SIZE[s]$. In our case, we can rely on a generalization of the lower bounds of Santhanam [San09] for $\mathcal{MA}$ protocols with non-uniform advice: For $t \approx s \circ s$ and a small function $\ell = O(\log(s))$ it holds that $MATIME[t]/\ell \not\subseteq SIZE[s]$ (see Appendix A for a proof of this statement).[8] Now, our derandomization hypothesis implies that $MATIME[t]/\ell \subseteq NTIME[t']/\ell$, where $t' = \text{poly}(t)$, and thus we can conclude that $NTIME[t']/\ell \not\subseteq SIZE[s]$.

The second observation is that if $NTIME[\text{poly}(t)]/\ell \not\subseteq SIZE[s]$, then $NTIME[\text{poly}(t)]$ (without the advice) is not contained in $SIZE[s']$, for s' that is moderately smaller than s. To see this, assume towards a contradiction that $NTIME[\text{poly}(t)] \subseteq SIZE[s']$. For any $S \in NTIME[\text{poly}(t)]/\ell$ we construct a family of size-s circuits that decides S. Consider a non-deterministic machine M that decides S with advice $\{a_n\}$, and let S^{adv} be the set of pairs (x, σ) such that $|\sigma| = \ell(|x|)$ and M (non-deterministically) accepts x when given advice σ. Note that S^{adv} can be decided by a non-determinstic machine that simulates M (and requires no advice), and thus, by our hypothesis, S^{adv} can be solved by a circuit family $\{C_n\}$ of size s'. By hard-wiring the "good" advice a_n into

[8] In fact, this lower bound can be improved by reducing the advice length ℓ from logarithmic to a single bit. But since our proof of Theorem 3 (rather than the weaker version) will later use the lower bound of [MW18], in which $\ell = O(\log(s))$, let us assume at this point already that ℓ is logarithmic.

each C_n, we obtain a circuit family $\{C'_n\}$ of size s' that decides S. Note that the size of the circuit is still s', but it is now a function of a smaller input length, since we "hard-wired" the advice in place of input bits; however, since the advice is relatively short (i.e., $\ell = O(\log(s))$), the new size function, denoted s, is not much larger than s' (see Proposition 23). The crucial point is that the foregoing "advice elimination" argument only follows through *after the derandomization step* (i.e., for $NTIME$ and not for $MATIME$). This is because when dealing with probabilistic machines, it is not clear how to define S^{adv} in a way that will allow a probabilistic machine without advice to decide it (since a probabilistic machine that is given a "wrong" advice might not "distinctly" accept or reject some inputs).

This proves the weaker version of Theorem 3, which does not assert that every circuit of size-s' fails to compute the hard function in any "small" interval. To prove the stronger version, our starting point is the strengthening by Murray and Williams [MW18] of the lower bound from [San09] for $\mathcal{MA}$ protocols with advice: They showed unconditionally that there exists $S \in MATIME[t]/\ell$ (where $\ell = O(\log(s))$ as above) such that $S \notin \mathtt{i.o.}_{[\mathrm{poly}(s)]}\text{-}SIZE[s]$ (see Theorem 20).

Going through the proof above, note that the first step (i.e., the derandomization step) preserves the failure of small circuits in every "small" interval; and thus we need to show that the second step (i.e., the "advice elimination" argument) also preserves this property. The source of trouble is that now our "towards-a-contradiction" hypothesis only implies that $S^{\mathsf{adv}} \in \mathtt{i.o.}_{[\mathrm{poly}(s)]}\text{-}SIZE[s]$, which only guarantees the existence of an infinite "dense" set $I \subseteq \mathbb{N}$ of input lengths for which S^{adv} has small circuits. In particular, we have no guarantee that every $n \in I$ is of the form $m + \ell(m)$, which is what we need to deduce that $S_m = S \cap \{0,1\}^m$ has small circuits. To overcome this problem, we "embed" all pairs (x, σ) such that $|\sigma| = \ell(|x|)$ and $|x| + |\sigma| < n$ into $\{0,1\}^n$, and define $S^{\mathsf{adv}}_n = S^{\mathsf{adv}} \cap \{0,1\}^n$ such that deciding S^{adv}_n allows to determine the output of M on (x, σ) for all pairs satisfying $|x| + |\sigma| < n$. Thus, for any $n \in I$, a circuit of size $s(n)$ that decides S^{adv}_n allows us to solve S_m where $m + \ell(m) < n$. And similarly to above, since the advice is relatively small (i.e., $\ell(m) = O(\log(s(m))) < m$), both the size $s(n)$ of the circuit and the interval length $\mathrm{poly}(s(n))$ in which failure is guaranteed are not too large as a function of m. For precise details see Proposition 23.

3 Preliminaries

We assume familiarity with basic notions of complexity theory; for background see, e.g., [Gol08; AB09]. Throughout the paper, fix any standard model of a Turing machine (we need a fixed model since we discuss time-constructible functions).

Whenever we refer to circuits (without qualifying which type), we mean non-uniform circuit families over the De-Morgan basis (i.e., AND/OR/NOT gates) with fan-in at most two and unlimited fan-out, and without any specific structural restrictions (e.g., without any limitation on their depth). The size of a circuit is the number of its gates. Moreover, we consider some fixed standard form of representation for such circuits, where the representation size is polynomial in the size of the circuit.

3.1 Circuit acceptance probability problem

We now formally define the circuit acceptance probability problem (or CAPP, in short); this well-known problem is also sometimes called Circuit Derandomization, Approx Circuit Average, and GAP-SAT or GAP-UNSAT.

Definition 5 *(CAPP). The* circuit acceptance probability problem *with parameters* $\alpha, \beta \in [0,1]$ *such that* $\alpha > \beta$ *(or* (α, β)*-CAPP, in short) is the following promise problem:*

- *The YES instances are (representations of) circuits that accept at least* α *of their inputs.*
- *The NO instances are (representations of) circuits that accept at most* β *of their inputs.*

We define the CAPP problem (i.e., omitting α *and* β*) as the* $(2/3, 1/3)$*-CAPP problem.*

It is well-known that CAPP is complete for $pr\mathcal{BPP}$ under deterministic polynomial-time reductions; in particular, CAPP can be solved in deterministic polynomial time *if and only if* $pr\mathcal{BPP} = pr\mathcal{P}$.

Proposition 6 *(CAPP is equivalent to* $pr\mathcal{BPP} = pr\mathcal{P}$*). The circuit acceptance probability problem can be solved in deterministic polynomial time* if and only if $pr\mathcal{BPP} = pr\mathcal{P}$.

For a proof of Proposition 6 see any standard textbook on the subject (e.g. [Vad12, Cor. 2.31], [Gol08, Exer. 6.14]). In Proposition 6 we considered the complexity of CAPP as a function of the input size, which is the size of the (description of the) circuit. However, following [Wil13], it can also be helpful to consider the complexity of CAPP as a function of both the circuit size m (which corresponds to the input size) and of the number v of input variables to the circuit. In this case, a naive deterministic algorithm can solve the problem in time $2^v \cdot \text{poly}(m)$, whereas the naive probabilistic algorithm solves the problem in time $v \cdot \text{poly}(m) \leq \text{poly}(m)$.

3.2 Witness circuits and the new easy witness lemma of [MW18]

We now recall the definition of witness circuits for a proof system.

Definition 7 *(verifiers and witnesses). Let* $t : \mathbb{N} \to \mathbb{N}$ *be a time-constructible, non-decreasing function, and let* $L \subseteq \{0,1\}^*$*. An algorithm* $V(x,y)$ *is a* t-time verifier for L *if* V *runs in time at most* $t(|x|)$ *and satisfies the following: For all strings* x *it holds that* $x \in L$ *if and only if there exists* a witness y *such that* $V(x,y)$ *accepts.*

Definition 8 *(witness circuits). Let* $t : \mathbb{N} \to \mathbb{N}$ *be a time-constructible, non-decreasing function, let* $w : \mathbb{N} \to \mathbb{N}$*, and let* $L \subseteq \{0,1\}^*$*. We say that a* t*-time verifier* V *has* witness circuits of size w *if for every* $x \in L$ *there exists a witness* y_x *such that* $V(x, y_x)$ *accepts and there exists a circuit* $C_{y_x} : \{0,1\}^{\log(|y_x|)} \to \{0,1\}$ *of size* $w(|x|)$ *such that* $C_{y_x}(i)$ *is the* i^{th} *bit of* y_x*. We say that* $NTIME[t]$ has witness circuits of size w *if for every* $L \in NTIME[t]$*, every* t*-time verifier for* L *has witness circuits of size* w*.*

Let us now state the new easy witness lemma of [MW18]. Loosely speaking, the lemma asserts that for any two functions $t(n) \gg s(n)$ with sufficient "gap" between them, if $NTIME[\mathrm{poly}(t)] \subseteq SIZE[s]$, then $NTIME[t]$ has witness circuits of size $\hat{s}$, where $\hat{s}(n) > s(n)$ is the function s with some "overhead". To more conveniently account for the exact parameters, we introduce some auxiliary technical notation:

Definition 9 *(sufficiently gapped functions). Let $\gamma, \gamma', \gamma'' \in \mathbb{N}$ be universal constants.*[9] *For any function $s : \mathbb{N} \to \mathbb{N}$, let $s' : \mathbb{N} \to \mathbb{N}$ be the function $s'(n) = (s(\gamma \cdot n))^{\gamma}$, and let $\hat{s} : \mathbb{N} \to \mathbb{N}$ be the function $\hat{s}(n) = (s'(s'(s'(n))))^{\gamma'}$. We say that two functions $s, t : \mathbb{N} \to \mathbb{N}$ are* sufficiently gapped *if both functions are increasing and time-constructible, and s' is also time-constructible, and $s(n) < 2^{n/\gamma}/n$, and $t(n) \geq (\hat{s}(n))^{\gamma''}$.*

Lemma 10 *(easy witnesses for low nondeterministic time [MW17, Lem. 4.1]). Let $s, t : \mathbb{N} \to \mathbb{N}$ be sufficiently gapped functions, and assume that $NTIME[O(t(n))^{\gamma}] \subset SIZE[s]$, where γ is the constant from Definition 9. Then, $NTIME[t]$ has witness circuits of size $\hat{s}$.*

3.3 Merlin-Arthur protocols

We recall the standard definition of Merlin-Arthur protocols (i.e., $\mathcal{MA}$ verifiers) that receive non-uniform advice.

Definition 11 *($\mathcal{MA}$ verifiers with non-uniform advice). For $t, \ell : \mathbb{N} \to \mathbb{N}$, a set $S \subseteq \{0,1\}^*$ is in $MATIME[t]/\ell$ if there exists a probabilistic machine V, called a* verifier, *such that the following holds: The verifier V gets input $x \in \{0,1\}^*$, and a witness $w \in \{0,1\}^*$, and an advice string $a \in \{0,1\}^*$, and runs in time $t(|x|)$; and there exists a sequence $\{a_n\}_{n \in \mathbb{N}}$ of advice such that $|a_n| = \ell(n)$ and:*

1. *For every $x \in S$ there exists $w \in \{0,1\}^{t(|x|)}$ such that $\Pr[V(x, w, a_{|x|}) = 1] \geq 2/3$.*
2. *For every $x \notin S$ and every $w \in \{0,1\}^{t(|x|)}$ it holds that $\Pr[V(x, w, a_{|x|}) = 1] \leq 1/3$.*

If in Item (1) the probability that $V(x, w, a_{|x|}) = 1$ is one (i.e., the verifier has perfect completeness when given "good" advice), then we denote $S \in MATIME_0[t]/\ell$.

It is common to denote by $MATIME[t]$ the class $MATIME[t]/0$ (i.e., when the verifier receives no non-uniform advice). Note that $\mathcal{MA} = \bigcup_{c \in \mathbb{N}} MATIME[n^c]$.

It is well-known that any Merlin-Arthur verifier can be modified to have perfect completeness, at the cost of a polynomial overhead in the running time, using the ideas of [Lau83]. Moreover, this statement also holds in the setting when the verifier relies on non-uniform advice. We include a proof of this fact for completeness:

Theorem 12 *($\mathcal{MA}$ verifiers have perfect completeness, wlog). There exists a universal constant $c > 1$ such that the following holds. Let $t, \ell : \mathbb{N} \to \mathbb{N}$ such that t is time-constructible, and let $S \subseteq \{0,1\}^*$ such that $S \in MATIME[t]/\ell$. Then, $S \in MATIME_0[t^c]/\ell$.*

[9] Specifically, the values of these constants are $\gamma = e$ and $\gamma' = 2g$ and $\gamma'' = d$, where e, g, and d are the universal constants from Lemma 4.1 in [MW17].

Proof. Let V be the $MATIME[t]/\ell$ verifier (with two-sided error) for S, and let $\{a_n\}_{n\in\mathbb{N}}$ be a sequence of "good" advice strings for V. By standard error-reduction, we can assume that V runs in time $t' = \mathrm{poly}(t)$ and (when given the "good" advice) has error probability at most $1/3t'$. Denote by $V(x, w, r, a')$ the decision of V on input x with proof w and randomness r and advice a'.

We construct a verifier V' with perfect completeness for S. On input $x \in \{0,1\}^n$ and with advice $a' \in \{0,1\}^{\ell(n)}$, the verifier V' expects to receive as proof both a string $w \in \{0,1\}^{t(n)}$ and $t' = t'(n)$ strings $\bar{s} = s_1, ..., s_{t'} \in \{0,1\}^{t'}$. The verifier V' uniformly chosses $r \in \{0,1\}^{t'}$, and accepts $x \in \{0,1\}^n$ if and only if there exists $i \in [t']$ such that $V(x, w, r \oplus s_i, a') = 1$. Now, when a' equals the "good" advice a_n, the following holds:

- If $x \in S$, then there exists $(w, \bar{s})$ such that for every $r \in \{0,1\}^{t'}$ there exists $i \in [t']$ satisfying $V(x, w, r \oplus s_i, a_n) = 1$; this is because for any w such that $\Pr_r[V(x, w, r, a_n) = 1] \geq 1 - 1/3t'$ we have that

$$\begin{aligned}\Pr_{\bar{s}}&[\exists r \in \{0,1\}^{t'} \text{ st } \forall i \in [t'], V(x, w, r \oplus s_i, a_n) = 0] \\ &\leq \sum_{r\in\{0,1\}^{t'}} \Pr_{\bar{s}}[\forall i \in [t'], V(x, w, r \oplus s_i, a_n) = 0] \\ &\leq 2^{t'} \cdot (1/3t')^{t'} ,\end{aligned}$$

  which is less than one.

- If $x \notin S$ then for every $(w, \bar{s})$ that the prover sends it holds that $\Pr_r[\exists i \in [t'] \text{ st } V(x, w, r \oplus s_i, a_n) = 1] \leq \sum_{i\in[t']} \Pr_r[V(x, w, r \oplus s_i, a_n) = 1] \leq 1/3$. ■

3.4 A barrier for proving "$pr\mathcal{BPP} = pr\mathcal{P} \implies \mathcal{P} \neq \mathcal{NP}$"

We note that it is impossible to prove the statement "if $pr\mathcal{P} = pr\mathcal{BPP}$ then $\mathcal{P} \neq \mathcal{NP}$" without *unconditionally* proving that $\mathcal{P} \neq \mathcal{NP}$.

Proposition 13 *(a barrier for "derandomization implies lower bounds" statements). If the conditional statement "$pr\mathcal{BPP} = pr\mathcal{P} \implies \mathcal{P} \neq \mathcal{NP}$" holds, then $\mathcal{P} \neq \mathcal{NP}$.*

Proof. Assume towards a contradiction that $\mathcal{P} = \mathcal{NP}$. Then, the polynomial-time hierarchy collapses to $\mathcal{P}$, and similarly the promise-problem version of the polynomial-time hierarchy collapses to $pr\mathcal{P}$.[10] Now, since $pr\mathcal{BPP}$ is contained in the promise-problem version of the polynomial-time hierarchy (e.g., by adapting the well-known argument of Lautemann [Lau83]), it follows that $pr\mathcal{BPP} = pr\mathcal{P}$. Finally, we can use the hypothesized conditional statement to deduce that $\mathcal{P} \neq \mathcal{NP}$, which is a contradiction. ■

[10] To see that this is the case, let $\Pi = (Y, N) \subseteq \{0,1\}^* \times \{0,1\}^*$ be a promise problem in $pr\Sigma_k$, for some $k \in \mathbb{N}$. Then, there exists a polynomial-time algorithm A such that for every $x \in Y$ it holds that $\exists y_1, \forall y_2, ..., y_k : A(x, y_1, ..., y_k) = 1$, and for every $x \in N$ it does not hold that $\exists y_1, \forall y_2, ..., y_k : A(x, y_1, ..., y_k) = 1$. We define a set $S = S_A$ that consists of all strings x such that $\exists y_1, \forall y_2, ..., y_k : A(x, y_1, ..., y_k) = 1$. Note that $S \supseteq Y$, and that $S \cap N = \emptyset$, and that $S \in \Sigma_k$ (using the algorithm A). By our assumption that the polynomial-time hierarchy collapses, there exists a polynomial-time algorithm A' that decides S. It follows that A' solves the problem Π.

4 Proof of Theorems 1 and 2

We will first prove a general and parametrized "derandomization implies lower bounds" theorem. This theorem is obtained by using the proof strategy of Williams [Wil13] with general parameters, while leveraging the new easy witness lemma of Murray and Williams [MW18]. We then prove Theorems 1 and 2 as corollaries. Towards presenting the proofs, we first need the following auxiliary definition:

Definition 14 *(non-deterministically solving CAPP). We say that* $(1, 1/3)$-CAPP can be solved in non-deterministic time $T : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ *if there exists a non-deterministic machine that, when given as input a circuit C of size m over v variables, runs in time $T(m, v)$ and satisfies the following: If C has acceptance probability one, then for some non-deterministic choice the machine accepts; and if C has acceptance probability at most $1/3$, then the machine always rejects (regardless of the non-deterministic choices).*

4.1 A parametrized "derandomization implies lower bounds" theorem

Loosely speaking, in the following theorem statement we assume that CAPP can be solved in non-deterministic time $T(m, v)$, and deduce that for any two functions $t(n) \gg s(n)$ such that $T\Big(\mathrm{poly}(n, \hat{s}(n), \log(t(n))),\ \log(t(n))\Big) \ll t(n)$ it holds that $NTIME[\mathrm{poly}(t(n))]$ does not have circuits of size $s(n)$.

Theorem 15 *(derandomization implies lower bounds, with general parameters). There exist constants $c, c' \in \mathbb{N}$ and $\alpha < 1$ such that the following holds. For $T : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, assume that $(1, 1/3)$-CAPP on circuits of size m with at most v input variables can be solved in non-deterministic time $T(m, v)$. Let $s, t : \mathbb{N} \to \mathbb{N}$ be sufficiently gapped functions such that $s(n) > n$ and for some constant $\epsilon > 0$ and any constant $\alpha > 0$ it holds that*

$$T\Big((n \cdot \hat{s}(n) \cdot \log(t(n)))^c,\ \ \alpha \cdot \log(t(n))\Big) \leq t(n)^{(1-\epsilon)\cdot\alpha}\ ,$$

where $\hat{s}$ is defined as in Definition 9. Then, $NTIME[t(n)^{c'}] \not\subseteq SIZE[s(n)]$.

Proof. The starting point of the proof is the non-deterministic time hierarchy [Coo72]: For an appropriate function $t' = t'(n)$ (that will be determined in a moment), there exists a set $L \in NTIME[t']$ that cannot be decided by non-deterministic machines running in time $(t')^{1-\Omega(1)}$. Specifically, for a sufficiently small constant $\alpha > 0$, let $t'(n) = (t(n))^{(1-\epsilon/2)\cdot\alpha}$, and let $L \in NTIME[t'] \setminus NTIME\left[(t')^{\frac{1-\epsilon}{1-\epsilon/2}}\right]$.[11] Now, for a sufficiently large constant c', assume towards a contradiction that $NTIME[t(n)^{c'}] \subseteq SIZE[s(n)]$. Our goal is to construct a non-deterministic machine that decides L in time $(t')^{\frac{1-\epsilon}{1-\epsilon/2}}$, which will yield a contradiction.

[11] Such a function exists by standard non-deterministic time hierarchy theorems (e.g., [Coo72]), since $t'(n) > n^{\Omega(1)}$, which implies that the gap between t' and $(t')^{1-\Omega(1)}$ is sufficiently large.

To do so, consider the PCP verifier of [BSV14] for L, denoted by V. On inputs of length n, the verifier V runs in time $\mathrm{poly}(n, \log(t'(n)))$, uses $\ell = \log(t'(n)) + O(\log\log(t'(n)))$ bits of randomness, and has perfect completeness and soundness (much) lower than $1/3$. [12] Furthermore, using the hypothesis that $NTIME[t(n)^{c'}] \subseteq SIZE[s(n)]$ and the "easy witness lemma" (i.e., Lemma 10), for every $x \in L$ there exists a circuit $P_x \in SIZE[\hat{s}(n)]$ such that $\Pr_r[V^{P_x}(x,r) \text{ accepts}] = 1$. (We actually apply Lemma 10 to the deterministic verifier V' that enumerates the random coins of V, which runs in time $2^\ell \cdot \mathrm{poly}(n, \log(t')) = \mathrm{poly}(t') = \mathrm{poly}(t)$. We can use the lemma since we assumed that $NTIME[t(n)^{c'}] \subseteq SIZE[s(n)]$, for a sufficiently large c'.)

Given input $x \in \{0,1\}^n$, the non-deterministic machine M acts as follows. The machine non-deterministically guesses a (description of a) circuit P_x of size $\hat{s}(n)$, and constructs a circuit $C_x^{P_x} : \{0,1\}^\ell \to \{0,1\}$ such that $C_x^{P_x}(r) = V^{P_x}(x,r)$. Then, the machine feeds the description of $C_x^{P_x}$ as input to the machine M_{CAPP} that solves CAPP in non-deterministic time T and exists by the hypothesis, and outputs the decision of M_{CAPP}. By the properties of the PCP verifier and of M_{CAPP}, if $x \in L$ then for some guess of P_x and for some non-deterministic choices of M_{CAPP}, the machine M will accept x; and if $x \notin L$, then for any guess of P_x and any non-deterministic choices of M_{CAPP}, the machine M will reject x.

To conclude let us upper-bound the running-time of the machine M. The circuit $C_x^{P_x}$ has $\ell = \log(t') + O(\log\log(t')) < \alpha \cdot \log(t)$ input bits, and its size is $m(n) = \mathrm{poly}(n, \log(t')) \cdot \hat{s}(n)$; thus, its representation size is $\mathrm{poly}(m(n))$. Therefore, the circuit $C_x^{P_x}$ can be constructed in time $\mathrm{poly}(m(n))$, and the CAPP algorithm runs in time $T(m(n), \ell)$. The total running-time of the non-deterministic machine M is thus at most $T\left((n \cdot \hat{s}(n) \cdot \log(t(n)))^c, \alpha \cdot \log(t)\right)$, for some constant c. By our hypothesized upper-bound on T, the running time of M is at most $t(n)^{(1-\epsilon)\cdot\alpha} = (t')^{\frac{1-\epsilon}{1-\epsilon/2}}$, which yields a contradiction. ■

Additional relaxations of the hypothesis in Theorem 15. Since the proof of Theorem 15 relies on the strategy of [Wil13], it is well-known that the hypothesis of the theorem can be further relaxed. First, we do not have to *unconditionally* assume that the non-deterministic machine for CAPP exists, and it suffices to assume that the machine exists *under the hypothesis that* $NTIME[t(n)^{c'}] \subseteq SIZE[s(n)]$ (this is the case since we are only using the existence of the machine to contradict the latter hypothesis). And secondly, the non-deterministic machine that solves CAPP can use (sub-linearly many) bits of non-uniform advice; this follows by using a strengthened non-deterministic time hierarchy theorem, which was proved by Fortnow and Santhanam [FS16] (see [MW18, Remark 1] for details).

[12] Note that the only upper-bound that we need on the number of oracle queries issued by V is the trivial bound given by the running time of V.

4.2 Theorems 1 and 2 as corollaries

We now prove Theorem 2 as a corollary of Theorem 15. As detailed in Section 2.1, we start from the hypothesis that $(1, 1/3)$-CAPP can be solved in non-deterministic time $T(m, v) = 2^{.99 \cdot v} \cdot \text{poly}(m)$ (which is weaker than the hypothesis $pr\mathcal{BPP} = pr\mathcal{P}$). The proof amounts to verifying that, given such a CAPP algorithm, the hypothesis of Theorem 15 holds for essentially any s and $t \approx s^{O(1)} \circ s^{O(1)} \circ s^{O(1)}$.

Corollary 16 *(Theorem 2, restated). Assume that $(1, 1/3)$-CAPP can be solved in non-deterministic time $T(m, v) \leq 2^{(1-\epsilon) \cdot v} \cdot \text{poly}(m)$, for some constant $\epsilon > 0$. Then, there exists a constant $k \in \mathbb{N}$ such that for any two sufficiently gapped functions $s : \mathbb{N} \to \mathbb{N}$ and $t : \mathbb{N} \to \mathbb{N}$ it holds that $NTIME[t(n)^k] \not\subseteq SIZE[s]$.*

Proof. Let $k' > 1$ be such that $T(m, v) \leq 2^{(1-\epsilon) \cdot v} \cdot m^{k'}$. We invoke Theorem 15 with the sufficiently gapped functions s and $t_1(n) = t(n)^{k''}$, where $k'' > 1$ is a sufficiently large constant that depends on k'. Note that for any $\alpha > 0$ it holds that

$$\begin{aligned}
T\Big((n \cdot \hat{s}(n) \cdot \log(t_1(n)))^c,\ \ \alpha \cdot \log(t_1(n)) \Big) & \\
&\leq (n \cdot \log(t_1(n)))^{c \cdot k'} \cdot (t_1(n))^{\epsilon/2} \cdot t_1(n)^{(1-\epsilon) \cdot \alpha} && (\hat{s}(n)^{c \cdot k'} < t_1(n)^{\epsilon/2}) \\
&\leq (t_1(n))^{1-\epsilon/3}, && (n^{c \cdot k'} < s(n)^{c \cdot k'} < t_1(n)^{\epsilon/12})
\end{aligned}$$

where both inequalities relied on the hypothesis that k'' is sufficiently large. Thus, we conclude that $NTIME[t_2] \not\subseteq SIZE[s]$, where $t_2(n) = t_1(n)^{c'} = t(n)^{c' \cdot k''}$. ■

Finally, we prove Theorem 1 as a corollary of Corollary 16. Recall that the conclusion in Theorem 1 is that $NTIME[n^{f(n)}] \not\subseteq \mathcal{P}/\text{poly}$ for "essentially" any super-constant function f. We now specify exactly what this means. Our goal is to deduce that $NTIME[n^{f(n)}] \not\subseteq SIZE[n^{g(n)}]$, where $g(n) \ll f(n)$ and $g(n) = \omega(1)$. Therefore, the proof works for any f such that a suitable g exists. We note in advance that this minor technical detail imposes no meaningful restrictions on f (see next).

Definition 17 *(admissible functions). We say that a function $f : \mathbb{N} \to \mathbb{N}$ is* admissible *if f is super-constant (i.e. $f(n) = \omega(1)$), and if there exists another super-constant function $g : \mathbb{N} \to \mathbb{N}$ that satisfies the following: The function g is super-constant, and $t(n) = n^{f(n)}$ and $s(n) = n^{g(n)}$ are sufficiently gapped, and $\hat{s}(n) = n^{o(f(n))}$.*

Essentially any increasing function $f(n) = \omega(1)$ such that $f(n) \leq n$ is admissible, where the only additional constraints that the admissibility condition imposes are time-constructibility of various auxiliary functions (we require t and s to be sufficiently gapped, which enforces time-constructibility constraints); for a precise (and tedious) discussion, see Appendix B. We can now formally state Theorem 1 and prove it:

Corollary 18 *(Theorem 1, restated). Assume that $(1, 1/3)$-CAPP can be solved in non-deterministic time $T(m, v) \leq 2^{(1-\epsilon) \cdot v} \cdot \text{poly}(m)$, for some constant $\epsilon > 0$. Then, for every admissible function f there exists a set in $NTIME[n^{O(f(n))}] \setminus \mathcal{P}/\text{poly}$.*

Proof. Since f is admissible, there exists a function g that satisfies the requirements of Definition 17. We thus invoke Corollary 16 with the functions $t(n) = n^{f(n)}$ and $s(n) = n^{g(n)}$, and conclude that there exists a set in $NTIME[n^{O(f(n))}] \setminus SIZE[n^{g(n)}]$. Since $g(n) = \omega(1)$, the latter set does not belong to $\mathcal{P}/\text{poly}$. ■

5 Proof of Theorem 3

In this section we prove Theorem 3. Throughout the section, for a set $S \subseteq \{0,1\}^*$ and $n \in \mathbb{N}$, we denote $S_n = S \cap \{0,1\}^n$.

Recall that the conclusion in Theorem 3 is that there exists a set S such that for every polynomial-sized circuit family and sufficiently large $n \in \mathbb{N}$, the family fails to decide S on some input length in the interval $[n, s_I(n)]$. Our actual conclusion will be slightly stronger: We will conclude that for every sufficiently large $n \in \mathbb{N}$, the circuit family fails to decide S on at least one of the "end-points" of the interval; that is, it fails either on input length n, or on input length $s_I(n)$ (or on both).

This leads us to the following definition. Intuitively, the following definition asserts that $S \in \texttt{i.o.}_{[s_I]}\text{-}SIZE[s]$ if there exists a circuit family of size s that, on infinitely-many input lengths $n \in \mathbb{N}$, manages to decide S correctly both on inputs of length n and on inputs of length $s_I(n)$. Indeed, it follows that if $S \notin \texttt{i.o.}_{[s_I]}\text{-}SIZE[s]$, then every circuit family $\{C_n\}$ of size s that tries to decide S fails, for every sufficiently large $n \in \mathbb{N}$, either on inputs on size n or on inputs of size $s_I(n)$ (or on both).

Definition 19 *(a stronger notion of infinitely-often computation). For $s, s_I : \mathbb{N} \to \mathbb{N}$ and $S \subseteq \{0,1\}^*$, we say that $S \in \texttt{i.o.}_{[s_I]}\text{-}SIZE[s]$ if there exists an infinite set $\mathcal{N} \subseteq \mathbb{N}$ and a circuit family $\{C_n\}_{n\in\mathbb{N}}$ of size at most s such that for every $n \in \mathcal{N}$, it holds that:*

1. *The circuit $C_n : \{0,1\}^n \to \{0,1\}$ computes S_n.*
2. *The circuit $C_{s_I(n)} : \{0,1\}^{s_I(n)} \to \{0,1\}$ computes $S_{s_I(n)}$.*

Definition 19 is reminiscent of the notion of robust simulations by Fortnow and Santhanam [FS17], but the two definitions have significant differences. Recall that a set $S \subseteq \{0,1\}^*$ can be *robustly simulated* by a circuit family if for every polynomial p there are infinitely-many integers n such that the family correctly decides S on all input lengths in the interval $[n, p(n)]$. In contrast, in Definition 19 we consider a *fixed* interval length s_I (which may also be super-polynomial), but only require the circuit to decide S on the end-points of the interval.

The starting point of the proof of Theorem 3 is Murray and Williams' [MW17, Thm 3.1] strengthening of Santhanam's [San09] circuit lower bound. Following [MW18], we say that a function $s : \mathbb{N} \to \mathbb{N}$ is a circuit-size function if s is increasing, time-constructible, and for all sufficiently large $n \in \mathbb{N}$ it holds that $s(n) < 2^n/(2n)$.

Theorem 20 *(Murray and Williams' [MW17, Thm 3.1] strengthening of Santhanam's [San09] lower bound). Let s be a super-linear circuit-size function, and let $t = \text{poly}(s(\text{poly}(s)))$ (for sufficiently large polynomials that do not depend on s). Then, there exists a set $S \in MATIME_0[t]/O(\log(s))$ such that $S \notin \texttt{i.o.}_{[\text{poly}(s)]}\text{-}SIZE[s]$.*

Note that in [MW18] the "hard" set S is stated to be in $MATIME[t]/O(\log(s))$, rather than in $MATIME_0[t]/O(\log(s))$ (i.e., the verifier in [MW18] has two-sided error). However, using Theorem 12, we can assume that $S \in MATIME_0[t]/O(\log(s))$, where $t = \mathrm{poly}(s(\mathrm{poly}(s)))$.

As mentioned in Section 2.2, the first observation in the proof is that if $pr\text{-}co\mathcal{RP} \subseteq pr\mathcal{NP}$ then we can derandomize MA verifiers that receive non-uniform advice.

Proposition 21 *(derandomization of MA with advice). If $pr\text{-}co\mathcal{RP} \subseteq pr\mathcal{NP}$, then for any $t, \ell : \mathbb{N} \to \mathbb{N}$ such that t is time-constructible it holds that $MATIME[t]/\ell \subseteq NTIME[\mathrm{poly}(t)]/\ell$.*

Proof. Note that $(1, 1/3)$-CAPP is in $pr\text{-}co\mathcal{RP}$. Thus, relying on the hypothesis that $pr\text{-}co\mathcal{RP} \subseteq pr\mathcal{NP}$, there exists a non-deterministic polynomial-time machine M_{CAPP} that gets as input a Boolean circuit C and satisfies the following: If the acceptance probability of C is one, then for some non-deterministic choices M_{CAPP} accepts; and if the acceptance probability of C is at most $1/3$ then M_{CAPP} rejects, regardless of the non-deterministic choices.

Now, let S be a set in $MATIME_0[t]/\ell$, let V be an $MATIME_0[t]/\ell$ verifier for S, and let $\{a_n\}_{n\in\mathbb{N}}$ be a sequence of "good" advice that allows V to decide S. We want to construct a non-deterministic machine M that runs in time $\mathrm{poly}(t)$ and decides S with ℓ bits of non-uniform advice. Given input $x \in \{0,1\}^n$ and advice a_n, the machine M guesses a witness $w \in \{0,1\}^{t(n)}$, and constructs a circuit $C = C_{V,x,w,a_n} : \{0,1\}^{t(n)} \to \{0,1\}$ that gets as input $r \in \{0,1\}^{t(n)}$ and computes $V(x, w, r, a_n)$. Then, the machine M feeds C to the machine M_{CAPP}, and outputs the decision of M_{CAPP}. The running time of the machine M is dominated by the running time of M_{CAPP}, which is at most $\mathrm{poly}(t(n))$. Now, since a_n is the "good" advice for V, if $x \in S$ then there exists w such that the acceptance probability of C is one, which means that there exist non-deterministic choices for M_{CAPP} such that M_{CAPP} will accept C; on the other hand, if $x \notin S$ then for any w the acceptance probability of C is at most $1/3$, which means that for any non-deterministic choices for M_{CAPP} it holds that M_{CAPP} rejects C. ■

The second observation in the proof is that if $NTIME[t]/\ell$ is not contained in a non-uniform class of circuits, then $NTIME[O(t)]$ (i.e., without non-uniform advice) is also not contained in a (related) non-uniform class of circuits. Moreover, this assertion still holds if the "separation" between the classes is in the sense of Definition 19.

We first prove a simpler form of this statement, which showcases the main idea but is much less cumbersome. In the following statement, we only consider a single bit of advice, and do not refer to separations in the sense of Definition 19.

Proposition 22 *(eliminating the advice). Let $s_0, s, t : \mathbb{N} \to \mathbb{N}$ such that t is increasing, and for all sufficiently large $n \in \mathbb{N}$ it holds that $s_0(n) \geq s(n+1)$. If $NTIME[t]/1 \not\subseteq SIZE[s_0]$, then $NTIME[O(t)] \not\subseteq SIZE[s]$.*

Proof. We prove the contrapositive statement: If $NTIME[O(t)] \subseteq SIZE[s]$, then $NTIME[t]/1 \subseteq SIZE[s_0]$. To do so, fix any $S \in NTIME[t]/1$, and let us construct a circuit family of size s_0 that decides S.

To construct the circuit family we consider an auxiliary set S^{adv}, which is defined as follows. Let M be a t-time non-deterministic machine and let $\{a_n\}$ be a sequence of advice bits such that M correctly decides S when given advice $\{a_n\}$. Let S^{adv} be the set of pairs (x, σ), where $x \in \{0,1\}^*$ and $\sigma \in \{0,1\}$, such that M (non-deterministically) accepts x when given advice σ. Note that $S^{\mathrm{adv}} \in NTIME[O(t)]$, because a non-deterministic machine that gets input (x, σ) simulate the machine M on input x with advice σ and decide according to the output of M.

Relying on the hypothesis that $NTIME[O(t)] \subseteq SIZE[s]$, there exists a circuit family $\{C_n\}$ of size s such that each C_n decides S_n^{adv}. By hard-wiring the "correct" advice bit a_n in place of the last input bit into every C_n, we obtain a circuit family $\{C'_n\}$ such that each C'_n decides S_n, and its size is at most $s(n+1) \leq s_0(n)$. ■

The following proposition is a stronger form of Proposition 22, which considers possibly long advice strings, and refers to separations in the sense of Definition 19.

Proposition 23 *(eliminating the advice). Let $s_0, s, \ell, t, s_I : \mathbb{N} \to \mathbb{N}$ be functions such that t is super-linear and increasing, and s_I, s_0 and s are increasing, and the mapping $1^n \mapsto 1^{\ell(n)}$ is computable in time $O(n + \ell(n))$. Assume that for every sufficiently large $n \in \mathbb{N}$ it holds that $\ell(n) < n/2$ and $s_0(n) \geq s(2n)$ and $s_0(s_I(n)) \geq s(2s_I(2n))$. Further assume that $NTIME[t]/\ell \not\subseteq \mathtt{i.o.}_{[s_I]}\text{-}SIZE[s_0]$. Then, $NTIME[O(t)] \not\subseteq \mathtt{i.o.}_{[2s_I]}\text{-}SIZE[s]$.*

We comment that a statement that is more general than the one in Proposition 23 can be proved, foregoing some of the requirements (e.g., on ℓ) while allowing potential degradation in the parameters of the conclusion. Since the statement of Proposition 23 suffices for our parameter setting, and for simplicity, we avoid such generalizations.

Proof of Proposition 23. Assuming that $NTIME[O(t)] \subseteq \mathtt{i.o.}_{[2s_I]}\text{-}SIZE[s]$, we prove that $NTIME[t]/\ell \subseteq \mathtt{i.o.}_{[s_I]}\text{-}SIZE[s_0]$. Fixing any $S \in NTIME[t]/\ell$, let us construct a circuit family of size s_0 that decides S infinitely-often on inputs of length n and $s_I(n)$.

We first define a set S^{adv} as follows. Let M be a t-time non-deterministic machine and let $\{a_n\}$ be a sequence of "good" advice strings of length $|a_n| = \ell(n)$ such that M correctly decides S when given advice $\{a_n\}$. For every $n \in \mathbb{N}$, the set S_n^{adv} will include representations of all pairs (x, σ), where $|\sigma| = \ell(|x|)$ and $|x| + 2|\sigma| < n$, such that M accepts x when given advice σ. Specifically, we define S_n^{adv} to be the set of all n-bit strings of the form $1^t 0^{|\sigma|} 1 x \sigma$, where $t = n - (|x| + 2|\sigma| + 1)$, such that M accepts x when given advice σ.[13]

Note that $S^{\mathrm{adv}} \in NTIME[O(t)]$. This is the case since a non-deterministic machine that gets input $z \in \{0,1\}^n$ can first verify that z can be parsed as $z = 1^t 0^{|\sigma|} 1 x \sigma$ such that $|\sigma| = \ell(|x|)$ (and reject z if the parsing fails); and then simulate the machine M on input x with advice σ, in time $O(t(|x|)) = O(t(n))$, and decide according to the output of M. Now, since we assume that $NTIME[O(t)] \subseteq \mathtt{i.o.}_{[2s_I]}\text{-}SIZE[s]$, there exists an infinite set $I \subseteq \mathbb{N}$ and a circuit family $\{C_n\}$ of size s such that for every $n \in I$:

1. $C_n : \{0,1\}^n \to \{0,1\}$ correctly computes S_n^{adv}; and

[13]The $0^{|\sigma|}$ term facilitates the parsing of the suffix of the n-bit string as a pair $x\sigma$.

2. $C_{2s_I(n)} : \{0,1\}^{2s_I(n)} \to \{0,1\}$ correctly computes $S^{\mathtt{adv}}_{2s_I(n)}$.

We transform $\{C_n\}$ into a circuit family of size s_0 that decides S infinitely-often on inputs of length both n and $s_I(n)$. To do so, we rely on the following simple claim:

Claim 23.1. *Let $n, m \in \mathbb{N}$ such that $m + 2\ell(m) < n$. Assume that there exists a circuit of size $s(n)$ that decides $S^{\mathtt{adv}}_n$. Then, there exists a circuit of size $s(n)$ that decides S_m.*

Proof. Let C_n be the circuit of size $s(n)$ for $S^{\mathtt{adv}}_n$. The circuit C_m for S_m is obtained by hard-wiring into C_n the "correct" advice a_m instead of the last $\ell(m)$ input bits, and the correct initial padding $1^{n-m-2\ell(m)-1}0^{\ell(m)}1$ instead of the first $n - m - \ell(m)$ input bits. □

For every $n \in I$, let $m = m(n)$ be the largest integer such that $m + 2\ell(m) + 1 \leq n$. Let $I' = \{m(n)\}_{n \in \mathbb{N}}$, and note that I' is infinite. For every sufficiently large $m \in I'$, relying on the fact that $m = m(n)$ for some $n \in I$ and on Claim 23.1, we have that:

1. There exists a circuit $C_m : \{0,1\}^m \to \{0,1\}$ of size $s(n) \leq s_0(\lceil n/2 \rceil) \leq s_0(m)$ that decides S_m. (We relied on the fact that $m \geq n/2$, since $\ell(m) < m/2$.)

2. There exists a circuit $C_{s_I(m)} : \{0,1\}^{s_I(m)} \to \{0,1\}$ of size $s_0(s_I(m))$ that decides $S_{s_I(m)}$. To see this, recall that there exists a circuit $C_{2s_I(n)}$ of size $s(2s_I(n))$ that decides $S^{\mathtt{adv}}_{2s_I(n)}$. We can invoke Claim 23.1 because $s_I(m) + 2\ell(s_I(m)) < 2s_I(m) < 2s_I(n)$. Also, relying on the fact that $m \geq n/2$ and on the hypotheses regarding s_0, s and s_I, we have that $s(2s_I(n)) \leq s(2s_I(2m)) \leq s_0(s_I(m))$.

It follows that $S \in \mathtt{i.o.}_{[s_I]}\text{-}\mathcal{SIZE}[s_0]$. ■

We now combine the foregoing ingredients into a proof of Theorem 3.

Theorem 24 *(Theorem 3, restated). There exists a constant $\epsilon > 0$ such that the following holds.*

- *Let $s : \mathbb{N} \to \mathbb{N}$ be an increasing, super-linear and time-constructible function such that for all sufficiently large $n \in \mathbb{N}$ it holds that $s(n) \leq 2^{\epsilon \cdot n}$ and that $s(2n) \leq s(n)^2$.*
- *Let $t = \mathrm{poly}(s(\mathrm{poly}(s)))$, for sufficiently large polynomials (that do not depend on s).*
- *Let $s_0 : \mathbb{N} \to \mathbb{N}$ be an increasing and time-constructible function such that for all sufficiently large $n \in \mathbb{N}$ it holds that $s_0(n) \geq s(n^2)^2$ and that $s_0(2n) \leq s_0(n)^2$.*

Assume that $pr\mathcal{BPP} \subseteq pr\mathcal{NP}$. Then, $\mathcal{NTIME}[t] \not\subseteq \mathtt{i.o.}_{[\mathrm{poly}(s_0)]}\text{-}\mathcal{SIZE}[s]$.

Note that if the function s in Theorem 24 satisfies $s(n^2) < s(n)^k$, for a sufficiently large constant $k \in \mathbb{N}$, then we can use the function $s_0(n) = s(n)^{2k}$, and deduce that $\mathcal{NTIME}[t] \not\subseteq \mathtt{i.o.}_{[\mathrm{poly}(s)]}\text{-}\mathcal{SIZE}[s]$.

Proof of Theorem 24. Let $t_0 = \mathrm{poly}(s_0(\mathrm{poly}(s_0)))$, for sufficiently large polynomials, and let $\ell = O(\log(s_0))$ (the universal constant hidden in the O-notation is the one from Theorem 20). By Theorem 20, there exists $S \in \mathcal{MATIME}[t_0]/\ell \setminus \texttt{i.o.}_{[(s_0)^c]}\text{-}\mathcal{SIZE}[s_0]$, for a sufficiently large constant $c \in \mathbb{N}$. By Proposition 21, and relying on the hypothesis that $pr\mathcal{BPP} \subseteq pr\mathcal{NP}$, it holds that $S \in \mathcal{NTIME}[\mathrm{poly}(t_0)]/\ell \setminus \texttt{i.o.}_{[(s_0)^c]}\text{-}\mathcal{SIZE}[s_0]$.

We now want to use Proposition 23 to deduce that $\mathcal{NTIME}[\mathrm{poly}(t_0)]$ is not contained in $\texttt{i.o.}_{[\mathrm{poly}(s_0)]}\text{-}\mathcal{SIZE}[s]$, and thus we need to verify that the functions ℓ, s, s_0, and $(s_0)^c$ satisfy the hypothesis of Proposition 23. This is indeed the case since for all sufficiently large $n \in \mathbb{N}$ it holds that $\ell(n) < n/2$ (assuming that ϵ is sufficiently small); and since $s_0(n) > s(n^2)^2 \geq s(2n)$, and $s(2s_0(2n)^c) \leq s(s_0(n)^{2c})^2 \leq s_0(s_0(n)^c)$. ■

As mentioned before the statement of Theorem 24, the "gap" between the input lengths n and $s_I(n) = \mathrm{poly}(s_0(n))$ (on which any size-s circuit family is guaranteed to fail) in Theorem 24 is larger than the function s that bounds the size of the circuits. This is no coincidence: If the gap function s_I would have been *significantly smaller* than the bound s on the circuit size, then we would have obtained an "almost-everywhere" lower bound (for circuits of size about $s(s_I^{-1})$).[14]

Acknowledgements

The author thanks his advisor, Oded Goldreich, for his close guidance in the research and writing process, and for very useful comments on drafts of the paper. The author thanks Ryan Williams for several very helpful email exchanges and conversations, and in particular for stressing the point that the conclusion in the main theorem can be interpreted as a weak form of a "strengthened time-hierarchy" theorem (that contrasts circuits and uniform algorithms), and for suggesting to try and prove an "almost-everywhere" lower bound as a consequence of $pr\mathcal{BPP} = pr\mathcal{P}$. The author is very grateful to Igor Oliveira for proposing the alternative proof of Theorem 1 (i.e., the one presented in Appendix A), and for his permission to include the alternative proof in this paper. The author also thanks anonymous reviewers for helpful comments.

The author is partially supported by Irit Dinur's ERC-CoG grant 772839. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 819702).

[14] To see this, assume that $S \notin \texttt{i.o.}_{[s_I]}\text{-}\mathcal{SIZE}[s]$, for $s_I \ll s$. We define a set $S^{\mathtt{emb}}$ by "embedding" all strings in S of length $n-1$ and $s_I^{-1}(n-1)$ into $\{0,1\}^n$: For each $n \in \mathbb{N}$, let $S_n^{\mathtt{emb}}$ consist of all n-bit strings $0^{n-|x|}1x$ such that $x \in S$. Since $S \notin \texttt{i.o.}_{[s_I]}\text{-}\mathcal{SIZE}[s]$, for every sufficiently large $n \in \mathbb{N}$ the circuit complexity of $S_n^{\mathtt{emb}}$ is larger either than $s(n-1)$ or than $s(s_I^{-1}(n-1))$. In natural cases where $s(s_I^{-1}(n-1)) < s(n-1)$, we obtain an "almost-everywhere" lower bound for circuits of size about $s(s_I^{-1})$.

References

[AB09] Sanjeev Arora and Boaz Barak. *Computational complexity: A modern approach*. Cambridge University Press, Cambridge, 2009.

[Bab+93] László Babai, Lance Fortnow, Noam Nisan, and Avi Wigderson. "BPP has subexponential time simulations unless EXPTIME has publishable proofs". In: *Computational Complexity* 3.4 (1993), pp. 307–318.

[BFT98] Harry Buhrman, Lance Fortnow, and Thomas Thierauf. "Nonrelativizing separations". In: *Proc. 13th Annual IEEE Conference on Computational Complexity (CCC)*. 1998, pp. 8–12.

[BGH+05] Eli Ben-Sasson, Oded Goldreich, Prahladh Harsha, Madhu Sudan, and Salil Vadhan. "Short PCPs Verifiable in Polylogarithmic Time". In: *Proc. 20th Annual IEEE Conference on Computational Complexity (CCC)*. 2005, pp. 120–134.

[BM84] Manuel Blum and Silvio Micali. "How to Generate Cryptographically Strong Sequences of Pseudo-random Bits". In: *SIAM Journal of Computing* 13.4 (1984), pp. 850–864.

[BSV14] Eli Ben-Sasson and Emanuele Viola. "Short PCPs with projection queries". In: *Proc. 41st International Colloquium on Automata, Languages and Programming (ICALP)*. 2014, pp. 163–173.

[Coo72] Stephen A. Cook. "A Hierarchy for Nondeterministic Time Complexity". In: *Proc. 4th Annual ACM Symposium on Theory of Computing (STOC)*. 1972, pp. 187–192.

[FS16] Lance Fortnow and Rahul Santhanam. "New non-uniform lower bounds for uniform classes". In: *Proc. 31st Annual IEEE Conference on Computational Complexity (CCC)*. 2016, Art. No. 19, 14.

[FS17] Lance Fortnow and Rahul Santhanam. "Robust simulations and significant separations". In: *Information and Computation* 256 (2017), pp. 149–159.

[Gol08] Oded Goldreich. *Computational Complexity: A Conceptual Perspective*. New York, NY, USA: Cambridge University Press, 2008.

[IKW02] Russell Impagliazzo, Valentine Kabanets, and Avi Wigderson. "In search of an easy witness: exponential time vs. probabilistic polynomial time". In: *Journal of Computer and System Sciences* 65.4 (2002), pp. 672–694.

[IW99] Russell Impagliazzo and Avi Wigderson. "P = BPP if E requires exponential circuits: derandomizing the XOR lemma". In: *Proc. 29th Annual ACM Symposium on Theory of Computing (STOC)*. 1999, pp. 220–229.

[JS12] Maurice Jansen and Rahul Santhanam. "Stronger lower bounds and randomness-hardness trade-offs using associated algebraic complexity classes". In: *Proc. 29th Symposium on Theoretical Aspects of Computer Science (STACS)*. Vol. 14. 2012, pp. 519–530.

[Juk12] Stasys Jukna. *Boolean function complexity*. Springer, Heidelberg, 2012.

[KI04] Valentine Kabanets and Russell Impagliazzo. "Derandomizing Polynomial Identity Tests Means Proving Circuit Lower Bounds". In: *Computational Complexity* 13.1-2 (2004), pp. 1–46.

[KMS12] Jeff Kinne, Dieter van Melkebeek, and Ronen Shaltiel. "Pseudorandom generators, typically-correct derandomization, and dircuit lower bounds". In: *Computational Complexity* 21.1 (2012), pp. 3–61.

[Lau83] Clemens Lautemann. "BPP and the polynomial hierarchy". In: *Information Processing Letters* 17.4 (1983), pp. 215–217.

[MW17] Cody Murray and Ryan Williams. "Circuit Lower Bounds for Nondeterministic Quasi-Polytime: An Easy Witness Lemma for NP and NQP". In: *Electronic Colloquium on Computational Complexity: ECCC* 24 (2017), p. 188.

[MW18] Cody Murray and Ryan Williams. "Circuit Lower Bounds for Nondeterministic Quasi-Polytime: An Easy Witness Lemma for NP and NQP". In: *Proc. 50th Annual ACM Symposium on Theory of Computing (STOC)*. 2018.

[NW94] Noam Nisan and Avi Wigderson. "Hardness vs. randomness". In: *Journal of Computer and System Sciences* 49.2 (1994), pp. 149–167.

[San09] Rahul Santhanam. "Circuit lower bounds for Merlin-Arthur classes". In: *SIAM Journal of Computing* 39.3 (2009), pp. 1038–1061.

[TV07] Luca Trevisan and Salil P. Vadhan. "Pseudorandomness and Average-Case Complexity Via Uniform Reductions". In: *Computational Complexity* 16.4 (2007), pp. 331–364.

[Vad12] Salil P. Vadhan. *Pseudorandomness*. Foundations and Trends in Theoretical Computer Science. Now Publishers, 2012.

[Wil13] Ryan Williams. "Improving Exhaustive Search Implies Superpolynomial Lower Bounds". In: *SIAM Journal of Computing* 42.3 (2013), pp. 1218–1244.

[Yao82] Andrew C. Yao. "Theory and Application of Trapdoor Functions". In: *Proc. 23rd Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. 1982, pp. 80–91.

Appendix A An alternative proof of Theorem 2

In this section we present an alternative proof of Theorem 1, which does not rely on the work of Murray and Williams [MW18], but rather on the work of Santhanam [San09]. The idea for this alternative proof was suggested to us by Igor Oliveira (after a preliminary version of this paper appeared online).

The structure of this alternative proof is very similar to the proof of Theorem 3 (which was described in Section 2.2), but uses as a starting point a generalization of the circuit lower bound proved by Santhanam [San09], instead of its subsequent strengthening by Murray and Williams [MW18]. Specifically, the starting point of the proof is the following:

Theorem 25 *(a generalization of [San09, Thm. 1]). Let $s : \mathbb{N} \to \mathbb{N}$ be an increasing, super-linear and time-computable function such that for all sufficiently large $n \in \mathbb{N}$ it holds that $s(3n) \leq s(n)^3$. Then, for $t : \mathbb{N} \to \mathbb{N}$ such that $t(n) = \mathrm{poly}(s(\mathrm{poly}(s(n))))$ it holds that $\mathcal{MATIME}[t]/1 \not\subseteq \mathcal{SIZE}[s]$.*

The proof of Theorem 25 imitates the original argument from [San09], but uses more general parameters. We include the full proof for completeness, but since it requires no new significant ideas, we defer its presentation to the end of the appendix. The alternative proof of Theorem 1 follows by combining Theorem 25, Proposition 21 (instantiated with the value $\ell = 1$), and Proposition 22.

Theorem 26 *(Theorem 1, an alternative technical statement). Let $s : \mathbb{N} \to \mathbb{N}$ be an increasing, super-linear and time-computable function such that for all sufficiently large $n \in \mathbb{N}$ it holds that $s(3n) \leq s(n)^3$, and let $t : \mathbb{N} \to \mathbb{N}$ such that $t(n) = \mathrm{poly}(s(\mathrm{poly}(s(n))))$, for sufficiently large polynomials. Assume that $pr\mathcal{BPP} = pr\mathcal{P}$. Then, $\mathcal{NTIME}[t] \not\subseteq \mathcal{SIZE}[s]$.*

Proof. Let $s_0 = s^3$, and let $t_0 = \mathrm{poly}(s_0(\mathrm{poly}(s_0)))$, for sufficiently large polynomials. According to Theorem 25, there exists a set S in $\mathcal{MATIME}[t_0]/1$ such that $S \notin \mathcal{SIZE}[s_0]$. By Proposition 21, and relying on the hypothesis that $pr\mathcal{BPP} = pr\mathcal{P}$, it holds that $S \in \mathcal{NTIME}[t_1]/1 \setminus \mathcal{SIZE}[s_0]$, where $t_1 = \mathrm{poly}(t_0)$. Using Proposition 22, it holds that $\mathcal{NTIME}[t] \not\subseteq \mathcal{SIZE}[s_1]$, where $t = O(t_1) = \mathrm{poly}(s(\mathrm{poly}(s)))$ and $s_1(n) = s_0(n-1)$. Finally, since s is increasing and $s(n) \leq s(\lceil n/3 \rceil)^3$, we have that $s_1(n) = s_0(n-1) \geq s_0(\lceil n/3 \rceil) \geq s(n)$, and hence $\mathcal{NTIME}[t] \not\subseteq \mathcal{SIZE}[s]$. ■

It is just left to detail the proof of Theorem 25. The first technical ingredient in the proof is the $\mathcal{PSPACE}$-complete set of Trevisan and Vadhan [TV07]. We use this set, but instead of relying on the fact that the set is $\mathcal{PSPACE}$-complete, we will use padding to claim that the set is complete for $\mathcal{DSPACE}[n^{\omega(1)}]$ under $n^{\omega(1)}$-time reductions.

Lemma 27 *(scaling the $\mathcal{PSPACE}$-complete set of [TV07]). There exists a set $L^{\mathtt{TV}} \subseteq \{0,1\}^*$ and a probabilistic polynomial-time oracle Turing machine M that satisfy the following:*

1. *Let $t : \mathbb{N} \to \mathbb{N}$ be a super-linear, time-computable function. Then, for every set $L \in \mathcal{DSPACE}[t]$ there exists a deterministic Turing machine R_L that runs in time $\mathrm{poly}(t)$ such that for every $x \in \{0,1\}^*$ it holds that $x \in L \iff R_L(x) \in L^{\mathtt{TV}}$.*

2. *On input $x \in \{0,1\}^*$, the machine M only issues queries of length $|x|$.*

3. *For any $x \in L^{\mathtt{TV}}$ it holds that $\Pr[M^{\mathbf{1}_{L^{\mathtt{TV}}}}(x) = 1] = 1$, where $\mathbf{1}_{L^{\mathtt{TV}}} : \{0,1\}^n \to \{0,1\}$ is the indicator function of $L^{\mathtt{TV}} \cap \{0,1\}^n$.*

4. *For any $x \notin L^{\mathtt{TV}}$ and any $f : \{0,1\}^n \to \{0,1\}$ it holds that $\Pr[M^f(x) = 0] \geq 2/3$.*

Proof. We take $L^{\mathtt{TV}}$ to be the $\mathcal{PSPACE}$-complete set from [San09, Lem. 12], which is the same set constructed in [TV07]. Items (2) – (4) follow immediately from the original statement in [San09].[15] Item (1) follows since $L^{\mathtt{TV}}$ is $\mathcal{PSPACE}$-complete, and using a padding argument. Specifically, for any t and L, consider the machine R_L that combines a reduction of L to $L' = \{(x, 1^t) : x \in L\}$ with a reduction of L' to $L^{\mathtt{TV}}$. The first reduction maps $x \mapsto (x, 1^t)$, and since $L' \in \mathcal{PSPACE}$, there exists a second reduction of L' to $L^{\mathtt{TV}}$ that can be computed in time $\mathrm{poly}(t + |x|) < \mathrm{poly}(t)$ (the inequality is since t is super-linear). ■

Proof of Theorem 25. Let $t_0 : \mathbb{N} \to \mathbb{N}$ such that $t_0(n) = s^4(n)$, and let $t_1 = \mathrm{poly}(t_0)$ and $t = t_2 = \mathrm{poly}(t_0(\mathrm{poly}(t_0)))$, for sufficiently large polynomials. Let $L^{\mathtt{TV}}$ be the set from Lemma 27. Our goal is to prove that there exists a set in $\mathcal{MATIME}[t_2]/1$ that is not in $\mathcal{SIZE}[t_0^{1/4}]$. The proof proceeds by a case analysis.

Case 1: $L^{\mathtt{TV}} \in \mathcal{SIZE}[t_0]$. By a standard diagonalization argument, there exists a set $L^{\mathtt{diag}} \in \mathcal{DSPACE}[t_1] \setminus \mathcal{SIZE}[t_0]$.[16] Our main goal now will be to prove that $\mathcal{DSPACE}[t_1] \subseteq \mathcal{MATIME}[t_2]$, which will imply that $L^{\mathtt{diag}} \in \mathcal{MATIME}[t_2] \setminus \mathcal{SIZE}[t_0]$. (Indeed, in this case we are proving a stronger result, since the $\mathcal{MA}$ verifiers do not need advice, and since the circuits are of size t_0 rather than $s = t_0^{1/4}$.)

To do so, let $L \in \mathcal{DSPACE}[t_1]$, and consider the following $\mathcal{MA}$ verifier for L. On input $x \in \{0,1\}^n$, the verifier computes $x' = R_L(x)$, where R_L is the machine from Lemma 27. Note that $n' = |x'| \leq \mathrm{poly}(t_1(n))$, and that $x \in L \iff x' \in L^{\mathtt{TV}}$. Now, the verifier parses the witness $w \in \{0,1\}^{\mathrm{poly}(t_0(n'))}$ as a description of a circuit $C : \{0,1\}^{n'} \to \{0,1\}$ of size $t_0(n')$, and runs the machine M from Lemma 27 on input x', while answering each oracle query of M using the circuit C.

Note that, since $L^{\mathtt{TV}} \in \mathcal{SIZE}[t_0]$, there exists a circuit C over n' input bits of size $t_0(n')$ that correctly computes $L^{\mathtt{TV}}$ on inputs of length n'. Therefore, by Lemma 27, when $x \in L$ there exists a witness such that the verifier accepts x with probability one, whereas the verifier rejects any $x \notin L$ with probability at least $2/3$, regardless of the witness. The total running time of the verifier is dominated by the time it takes to simulate M using the circuit C, which is at most $\mathrm{poly}(n') \cdot \mathrm{poly}(t_0(n')) \leq t_2(n)$.

Case 2: $L^{\mathtt{TV}} \notin \mathcal{SIZE}[t_0]$. In this case we show an explicit set $L^{\mathtt{pad}}$, which will be a padded version of $L^{\mathtt{TV}}$, such that $L^{\mathtt{pad}}$ can be decided in $\mathcal{MATIME}[t_2]$ with one bit of advice, but cannot be decided by circuits of size $s = t_0^{1/4}$. To do so, let $\mathtt{sz}_{\mathtt{TV}} : \mathbb{N} \to \mathbb{N}$ be such that $\mathtt{sz}_{\mathtt{TV}}(n)$ is the minimum circuit size for $L^{\mathtt{TV}}_n = L^{\mathtt{TV}} \cap \{0,1\}^n$. Also, for any integer m, let $p(m) = 2^{\lfloor \log(m) \rfloor}$ be the largest power of two that is not larger than m,

[15] The original statement asserts that any $x \notin L^{\mathtt{TV}}$ is rejected with probability at least $1/2$ (rather than $2/3$), but this probability can be amplified to $2/3$ using standard error-reduction.

[16] For example, $L^{\mathtt{diag}} = \{x : C_{|x|}(x) = 1\}$, where C_n is the lexicographically-first circuit over n bits of size at most $t_0^2(n)$ that decides a set whose circuit complexity is more than $t_0(n)$. The proof that $L^{\mathtt{diag}} \in \mathcal{DSPACE}[t_1]$ follows the well-known idea used in Kannan's theorem (see, e.g., [Juk12, Lem. 20.12]).

and let $n(m) = m - p(m)$. We think of $n(m)$ as the "effective input length" indicated by m, and on $p(m)$ as the length of padding. We define the set $L^{\mathtt{pad}}$ as follows:

$$L^{\mathtt{pad}} = \Big\{ (x, 1^p) :\ x \in L^{\mathtt{TV}}, \text{ and } |x| = n(|x| + p), \\ \text{and } t_0(|x| + p) \le \mathtt{sz}_{\mathtt{TV}}(|x|)^3 < t_0(|x| + 2p) \Big\} .$$

Let us first see that $L^{\mathtt{pad}}$ cannot be decided by circuits of size $t_0^{1/4}$. Assume towards a contradiction that there exists a circuit family $\{C_m\}$ of size $t_0^{1/4}$ that decides $L_m^{\mathtt{pad}}$ correctly for every m. Since $L^{\mathtt{TV}} \notin SIZE[t_0]$, there exists an infinite set $I \subseteq \mathbb{N}$ such that for every $n \in I$ it holds that $\mathtt{sz}_{\mathtt{TV}}(n) > t_0(n)$. For a sufficiently large $n \in I$, we will construct a circuit $C'_n : \{0,1\}^n \to \{0,1\}$ of size less than $\mathtt{sz}_{\mathtt{TV}}(n)$ that computes $L_n^{\mathtt{TV}}$, which yields a contradiction to the definition of $\mathtt{sz}_{\mathtt{TV}}$.

Specifically, consider the circuit $C'_n : \{0,1\}^n \to \{0,1\}$ that acts as follows. Let p be a power of two such that $t_0(n+p) \le \mathtt{sz}_{\mathtt{TV}}{}^3(n) < t_0(n+2p)$; there exists such a p since $t_0(n + 2^{\lceil \log(n) \rceil}) \le t_0(n)^3 < \mathtt{sz}_{\mathtt{TV}}{}^3(n)$. The value of this p is hard-coded into C'_n. Given $x \in \{0,1\}^n$, the circuit C'_n pads x with 1^p, simulates the circuit C_m on $(x, 1^p)$ (where $m = n + p$), and outputs $C_m(x, 1^p)$. By the definition of $L^{\mathtt{pad}}$ it holds that C'_n correctly computes $L_n^{\mathtt{TV}}$. The size of C'_n is dominated by the size of C_m, and is thus at most $O(t_0(n+p)^{1/4}) = o(t_0(n+p)^{1/3})$. Since $t_0(n+p)^{1/3} \le \mathtt{sz}_{\mathtt{TV}}(n)$ and n is sufficiently large, the size of C'_n is less than $\mathtt{sz}_{\mathtt{TV}}(n)$, which yields a contradiction.

Let us now see that $L^{\mathtt{pad}}$ can be decided by an MA verifier that runs in time t_2 and uses one bit of advice. Given an input z of length m, the advice bit is set to one if and only if $L_m^{\mathtt{pad}} \neq \emptyset$; if the advice is zero, the verifier immediately rejects. Otherwise, the verifier computes $n = n(m)$ and $p = p(m)$, and parses the input z as $(x, 1^p)$ where $|x| = n$ (if the verifier fails to parse the input, it immediately rejects). The verifier parses the witness $w \in \{0,1\}^{\mathrm{poly}(t_0(n+2p))}$ as a circuit $C : \{0,1\}^n \to \{0,1\}$ of size at most $t_0(n+2p)^{1/3}$, and emulates the machine M from Lemma 27 on input x, answering each oracle query of M using the circuit C. The verifier outputs the decision of M.

Since $\mathtt{sz}_{\mathtt{TV}}(n) < t_0(n+2p)^{1/3}$, there exists a circuit C of size at most $t_0(n+2p)^{1/3}$ that computes $L_n^{\mathtt{TV}}$. For any $z \in L^{\mathtt{pad}}$, when the witness represents this circuit, the verifier accepts z with probability one. Also, for any $z \notin L^{\mathtt{TV}}$, the verifier rejects x with probability $2/3$, regardless of the witness. Finally, note that the running time of the verifier is dominated by the time that it takes to run the machine M while simulating the oracle answers, which is at most $\mathrm{poly}(n) \cdot \mathrm{poly}(t_0(2m)) \le t_2(m)$. ■

Appendix B Sufficient conditions for admissibility

The point of the current appendix is to show that essentially any increasing function $f(n) = \omega(1)$ such that $f(n) \le n$ is admissible (in the sense of Definition 17).

Claim 28. *Let $f(n) = \omega(1)$ be any increasing function such that $f(n) \le n$ for all n, and*

$t(n) = n^{f(n)}$ is time-constructible, and $s(n) = n^{\log(f(\log(n)))}$ is time-constructible, and $s'(n)$ is time-constructible. Then, f is admissible.

Proof. Let $g(n) = \log(f(\log(n)))$ and let $s(n) = n^{g(n)}$. We need to verify that g is super-constant (which holds because f is super-constant), and that t and s are sufficiently gapped, and that $\hat{s}(n) = n^{o(f(n))}$. To see that t and s are sufficiently gapped, first note that both functions are increasing (since f is increasing, and hence g is also increasing) and are time-constructible, as is s' (we assumed time-constructibility in the hypothesis). Also note that $s(n) \le n^{\log\log(n)} < 2^{n/\gamma}/n$.

Thus, it is left to verify that $\hat{s}(n) = n^{o(f(n))}$. The proof of this fact amounts to the following elementary calculation. First note that

$$s'(n) = (s(\gamma \cdot n))^{\gamma} = (\gamma \cdot n)^{\gamma \cdot \log(f(\log(\gamma \cdot n)))} < n^{\log^2(f(\log^2(n)))} .$$

Thus, for any function $k = k(n)$ and constant $c \ge 2$ such that $k(n) \le \log^c(f(\log^{3c}(n)))$ (which in particular implies that $k(n) \le \log^c(n)$), we have that

$$s'(n^k) < n^{k \cdot \log^2(f(\log^2(n^k)))} \le n^{\log^{2c}(f(\log^{3c}(n)))} . \tag{1}$$

In particular, using Eq. (1) with $k(n) = \log^2(f(\log^2(n)))$ and $c = 2$, we deduce that $s'(s'(n)) < n^{\log^4(f(\log^6(n)))}$. Then, using Eq. (1) again with $k(n) = \log^4(f(\log^6(n)))$ and $c = 4$, we deduce that $s'(s'(s'(n))) < n^{\log^8(f(\log^{12}(n)))}$. Therefore, we have that $\hat{s}(n) < n^{\gamma' \cdot \log^8(f(\log^{12}(n)))} < n^{\gamma' \cdot \mathrm{poly}\log(f(n))} = n^{o(f(n))}$. ■

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il