Electronic Colloquium on Computational Complexity, Revision 2 of Report No. 3 (2018)

Proving that $prBPP = prP$ is as hard as "almost" proving that $P \neq NP$

Roei Tell *

January 28, 2018

Abstract

What circuit lower bounds are *necessary* in order to prove that *promise-*BPP $=$ *promise-*P? The main result in this paper is that if *promise-*BPP $=$ *promise-*P, then polynomial-sized circuits cannot simulate non-deterministic machines that run in arbitrarily small super-polynomial time (i.e., $NTIME[n^{f(n)}] \not\subseteq P/\text{poly}$, for essentially any $f(n) = \omega(1)$). The super-polynomial time bound in the conclusion of the foregoing conditional statement cannot be improved (to conclude that $NP \not\subseteq P/\text{poly}$) without *unconditionally* proving that $P \neq NP$.

This paper is a direct follow-up to the very recent breakthrough of Murray and Williams (ECCC, 2017), in which they proved a new "easy witness lemma" for $NTIME[o(2^n)]$. Our main contribution is in highlighting the *strong "barriers" for proving* $prBPP = prP$ that can be demonstrated using their results (and, as it turns out, also using previous results). We include three proofs of the main theorem: Two proofs that rely on various results from the work of Murray and Williams, and yield stronger forms of the main theorem (i.e., either use a weaker hypothesis or deduce a stronger conclusion); and a third proof that only relies on a generalization of the well-known lower bound of Santhanam (SICOMP, 2009).

*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, a version of the conjecture is *equivalent* to the conjectured existence of an algorithm for a single, specific problem (i.e., the *circuit acceptance probability problem*; see Proposition 5). However, as will be discussed next, it has also been known for at least two decades that the conjecture is actually intimately related to *circuit lower bounds*; that is, the conjecture is related to lower bounds for non-uniform models of computation. The attempts to prove such lower bounds have long been considered a prominent path to make progress on the $\mathcal{P} \neq \mathcal{NP}$ conjecture.

Informally, following a very recent breakthrough by Murray and Williams [MW17], the main result in this paper considerably strengthens the knowns connection between the $\mathcal{BPP} = \mathcal{P}$ conjecture and circuit lower bounds. To present the new result, let us first spell out the latter connections. On the one hand, any proof of sufficiently strong circuit lower bounds would also prove the conjecture; specifically, if there is a function in $\mathcal{E}$ that requires exponential-sized circuits, then $\mathcal{BPP} = \mathcal{P}$ (and even $pr\mathcal{BPP} = pr\mathcal{P}$, i.e. the promise-problem versions of $\mathcal{BPP}$ and of $\mathcal{P}$ are equal; see [IW99], which relies on the hardness-randomness paradigm [Yao82; BM84; NW94]). On the other hand, any proof that $pr\mathcal{P} = pr\mathcal{BPP}$ (or even of much weaker instances of this conjecture) implies long-sought circuit lower bounds (see, e.g., [IW98; IKW02; KI04; Wil13]).

A specific celebrated example of such a connection is that any proof that $pr\mathcal{BPP} = pr\mathcal{P}$ implies that there exists a function in $\mathcal{NEXP}$ that cannot be computed by any polynomial-sized circuit family [IKW02]. Very recently, Murray and Williams [MW17] proved a breakthrough result that allows to significantly strengthen this statement: In particular, we observe that an immediate corollary of [MW17, Thm 1.2] is that *if* $pr\mathcal{BPP} = pr\mathcal{P}$*, then there exists a function in* $NTIME[n^{\text{poly}\log(n)}]$ (rather than $\mathcal{NEXP}$) *that cannot be computed by any polynomial-sized circuit family*. We believe that this corollary is a fundamental result that is worth spelling out and highlighting. Furthermore, this result can be strengthened, and doing so is the technical contribution in this paper.

Specifically, the main result in the current paper is that if $pr\mathcal{BPP} = pr\mathcal{P}$, then there exists a function in $NTIME[n^{\omega(1)}]$ that cannot be computed by any polynomial-sized circuit family; that is, $NTIME[n^{\omega(1)}] \not\subseteq \mathcal{P}/\text{poly}$. Note that this circuit lower bound is so strong that an improvement in the time bound of the non-deterministic class in this bound (i.e., in $NTIME[n^{\omega(1)}] \not\subseteq \mathcal{P}/\text{poly}$) would immediately imply that $\mathcal{P} \neq \mathcal{NP}$. For further discussion of the meaning of this lower bound, see Section 1.3.

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* $NTIME[n^{f(n)}] \setminus \mathcal{P}/\text{poly}$.

One might a-priori hope to further improve the conclusion of Theorem 1, and prove a theorem of the form "if $prBPP = prP$, then $NP \not\subseteq P/\text{poly}$ (and $P \neq NP$)". However, we note that such a result cannot be proved without *unconditionally* proving that $P \neq NP$. That is, any proof of the conditional statement "$BPP = P \implies P \neq NP$" would *unconditionally* imply that $P \neq NP$ (see Proposition 11). Therefore, the conclusion of Theorem 1 is optimal in this sense.

The hypothesis in the conditional statement in Theorem 1 refers to classes of *promise problems*, and is thus stronger than the hypothesis that $BPP = P$ (which refers to decision problems with the trivial promise). However, the promise-problem version of the conjecture is natural and well-studied by itself, and moreover, the strongest evidence that currently suggests that $P = BPP$ also suggests that $prP = prBPP$ (since it is based on constructions of pseudorandom generators; see [IW99]).

We will deduce Theorem 1 as a corollary of (either of two) stronger theorems, which will be proved relying on various results of Murray and Williams [MW17]. However, we comment that Theorem 1 can also be proved without relying on the results of Murray and Williams. In particular, there exists a relatively simple proof for the theorem that is based on (a generalization of) the well-known circuit lower bound of Santhanam [San09]. We present this alternative proof of Theorem 1 in Appendix B.[1]

1.1 Weakening the hypothesis of Theorem 1

We can deduce the conclusion in Theorem 1 from a hypothesis that is much weaker than $prBPP = prP$. Specifically, similarly to [Wil13; MW17], it suffices to assume that there exists a deterministic algorithm for the *circuit acceptance probability problem* (see Definition 4) that runs in time *noticeably smaller* than the naive deterministic algorithm for this problem. (To see that this follows from $prBPP = prP$, see Proposition 5.) More accurately, Theorem 1 is a corollary of the following theorem:

Theorem 2 *(main theorem with a weaker hypothesis; informal). Assume that for some constant $\epsilon > 0$ there exists an algorithm that gets as input a circuit C of size m over v variables, runs in time $2^{(1-\epsilon)\cdot v} \cdot \text{poly}(m)$, and distinguishes between the case that the acceptance probability of C is one and the case that the acceptance probability of C is at most $1/3$. Then, for essentially any super-constant function $f(n) = \omega(1)$ there exists a set in $NTIME[n^{f(n)}] \setminus P/\text{poly}$.*

Theorem 2 has a form that is very similar to [MW17, Thms. 1.1 & 1.2], yet its parameters are different. The latter theorems can be considered as significant strengthenings of the celebrated result of Williams [Wil13], and their proofs in fact follow Williams' original proof strategy. In the original result, the hypothesis is that there exists a "weak" circuit-analysis algorithm, and the conclusion is a "weak" (yet highly non-trivial) circuit lower bound. In contrast, in Theorem 2 (as well as in [MW17]) the hypothesis is somewhat stronger, but the conclusion is considerably stronger. The key new technical component that allows to improve Williams' original result [Wil13] is the new "easy witness lemma" of Murray and Williams [MW17] (see Section 2 for details).

[1] The idea for the alternative proof was suggested to us by Igor Oliveira after a preliminary version of this paper appeared online.

The hypotheses of Theorem 2 can even be further relaxed (since the theorem's proof relies on Williams' proof strategy, which is well-known to support such further relaxations). For example, the conclusion of Theorem 2 also follows from "non-deterministic derandomization", and in particular follows from the hypothesis that $pr{-}co\mathcal{RP} \subseteq pr{-}\mathcal{NP}$. For a discussion of such relaxations see the end of Section 4.

1.2 Strengthening the conclusion of Theorem 1

The previous section (i.e., Section 1.1) demonstrated that Theorem 1 can be deduced as a corollary of a theorem that uses a weaker hypothesis (i.e., Theorem 2). In this section we show that Theorem 1 can be deduced as a corollary of a theorem that uses the same hypothesis (i.e., that $pr\mathcal{BPP} = pr\mathcal{P}$) but that has a stronger conclusion.

Recall that in Theorem 1 we concluded that there exists a set $S \in NTIME[n^{\omega(1)}]$ such that $S \notin \mathcal{P}/\text{poly}$. The assertion that $S \notin \mathcal{P}/\text{poly}$ means that any polynomial-sized circuit family that tries to decide S fails *infinitely often*. The following theorem asserts that if $pr\mathcal{BPP} = pr\mathcal{P}$, there exists $S \in NTIME[n^{\omega(1)}]$ such that any polynomial-sized circuit family fails to decide S *on a "dense" set of input lengths*; specifically, in any interval of size $n^{\omega(1)}$ there exists some input length on which the family fails to decide S.

Theorem 3 *(main theorem with a stronger conclusion; informal). Assume that $pr\mathcal{BPP} = pr\mathcal{P}$, and let $f(n) = \omega(1)$ be essentially any super-constant function. Then, there exists a set $S \in NTIME[n^{f(n)}]$ such that for any polynomial-sized circuit family $\{C_n\}_{n\in\mathbb{N}}$, where $C_n : \{0,1\}^n \to \{0,1\}$, and any sufficiently large $n \in \mathbb{N}$, there exists $m \in [n, n^{f(n)}]$ such that C_m does not decide $S \cap \{0,1\}^m$.* [2]

The proof of Theorem 3 is very different from the proof of Theorem 2, and is actually more similar to the alternative proof of Theorem 1 that is presented in Appendix B. Recall that the latter proof relies on the circuit lower bound of Santhanam [San09]; the proof of Theorem 3 relies on a strengthening of Santhanam's lower bound, which was proved by Murray and Williams (see [MW17, Thm 3.1], restated in Theorem 16).

The aforementioned strengthened circuit lower bound is a crucial component in the proof of the new easy witness lemma [MW17]. Since Theorem 3 relies directly on this lower bound, rather than on the easy witness lemma, its proof may be considered simpler (or more "direct") than the proof of Theorem 2. Moreover, the technical statement of Theorem 3 offers some improvements in low-level parameters, compared to the technical statement of Theorem 2 (see discussion after the proof of Theorem 20).

1.3 Theorem 1 as a "barrier" for proving $pr\mathcal{BPP} = pr\mathcal{P}$

Our main interpretation of Theorem 1 is as posing a strong "barrier" for proving that $pr\mathcal{BPP} = pr\mathcal{P}$: The theorem implies that proving $pr\mathcal{BPP} = pr\mathcal{P}$ is *at least as hard* as proving that $NTIME[n^{\omega(1)}] \not\subseteq \mathcal{P}/\text{poly}$. The foregoing statement significantly strengthens previously-known "barriers" for proving $pr\mathcal{BPP} = pr\mathcal{P}$ (cf., e.g., [IKW02]). Let us

[2] The actual lower bound that we deduce is even slightly stronger, since it asserts that the circuit family fails on (at least) one of the "end-points" of the interval; see Section 5 for precise details.

now try to clarify the meaning of this "barrier"; that is, we ask what are the conceptual implications of proving that $NTIME[n^{\omega(1)}] \not\subseteq \mathcal{P}/\text{poly}$. Note that the latter statement asserts that polynomial-sized circuits cannot simulate *both super-polynomial running time and non-determinism*.

On the one hand, one can consider "$NTIME[n^{\omega(1)}] \not\subseteq \mathcal{P}/\text{poly}$" to be a weaker form of "$\mathcal{NP} \not\subseteq \mathcal{P}/\text{poly}$" (i.e., of the assertion that polynomial-sized circuits cannot simulate polynomial non-determinism). From this view, Theorem 1 implies that proving that $pr\mathcal{BPP} = pr\mathcal{P}$ is as hard as "almost" proving that $\mathcal{P} \neq \mathcal{NP}$ (as suggested by the title of the paper). On the other hand, as pointed out by Ryan Williams, one could also view the statement "$NTIME[n^{\omega(1)}] \not\subseteq \mathcal{P}/\text{poly}$" as a weaker form of "$DTIME[n^{\omega(1)}] \not\subseteq \mathcal{P}/\text{poly}$". The latter statement asserts that polynomial-sized circuits cannot simulate algorithms with *super-polynomial running time*; this is a "strengthened time-hierarchy theorem" (in which we compare uniform algorithms to non-uniform circuits). From this perspective, Theorem 1 implies that proving $pr\mathcal{BPP} = pr\mathcal{P}$ is as hard as proving (a weak form of) a "strengthened time-hierarchy theorem". Needless to say, both views imply a "barrier" for proving $pr\mathcal{BPP} = pr\mathcal{P}$ that seems challenging at the current time.

1.4 Organization

In Section 2 we present high-level overviews of the proofs of Theorems 2 and 3. In Section 3 we present preliminary definitions. In Section 4 we prove Theorem 2, and in Section 5 we prove Theorem 3. Finally, an alternative proof of Theorem 1 is presented in Appendix B.

2 Overviews of the proofs

In Section 2.1 we present a very brief digest of the proofs, which is intended primarily for experts; this digest can be safely skipped. More detailed overviews, which also describe known approaches that are needed for the proofs, appear in Sections 2.2 and 2.3.

2.1 A brief digest (intended primarily for experts)

Let us first describe the proof of Theorem 2. This proof relies on the celebrated proof strategy of Williams [Wil13], who showed that any non-trivial derandomization of a circuit class implies lower bounds for that class against $\mathcal{NEXP}$. The key technical ingredient in Williams' proof strategy is an "easy witness lemma" (as in [IKW02]), which asserts that if some unexpected complexity collapse occurs (e.g., $\mathcal{NEXP} \subseteq \mathcal{P}/\text{poly}$), then there exist *small* circuits that encode *large* witnesses in proof systems (e.g., for $\mathcal{NE}$).

Very recently, Murray and Williams [MW17] proved a new easy witness lemma, with significantly improved parameters. Indeed, this new lemma allows to improve the results obtained via Williams' proof strategy. Specifically, Murray and Williams showed that any non-trivial derandomization of a circuit class implies lower bounds for this class against $NTIME[n^{\text{poly}\log(n)}]$. As mentioned in the introduction, a weak version of

Theorem 1 follows immediately as a corollary of this result (i.e., the result implies that if $prBPP = prP$, then $NTIME[n^{\text{poly}\log(n)}] \not\subseteq P/\text{poly}$).

The main observation that paves the path to Theorem 2 is that *the proof strategy of Williams can yield a stronger conclusion when starting from a stronger hypothesis*. (This observation also underlies [MW17, Thm 1.1].) Accordingly, we state and prove a parametrized "derandomization implies lower bounds" theorem (see Theorem 12), which uses the new easy witness lemma within the proof strategy of Williams with general parameters.[3] This theorem has the following form: Assuming a sufficiently strong derandomization hypothesis, and taking any two functions $t > s$ that have a sufficient "gap" between them, it holds that $NTIME[t] \not\subseteq SIZE[s]$. The required "gap" between s and t is determined (in part) by the strength of the derandomization hypothesis. In particular, under the hypothesis in Theorem 2, we can take both s and t to be arbitrarily-small super-polynomial functions (i.e., $t(n) = n^{\omega(1)}$ and $s(n) = n^{\omega(1)} \ll t(n)$), which allows us to deduce that $NTIME[t]$ is not contained in $SIZE[s] \supseteq P/\text{poly}$.

The proof of Theorem 3 does not rely on Williams' proof strategy [Wil13] or on an easy witness lemma. Instead, the starting point of the proof is the unconditional existence, for any two functions $t > s$ with sufficient "gap" between them, of a set $S \in MATIME[t]/O(\log(s))$ (i.e., S can be decided by a t-time Merlin-Arthur protocol with $O(\log(s))$ bits of non-uniform advice) such that S cannot be computed by circuits of size s, in the strong sense of Theorem 3 (see [MW17, Thm 3.1]).[4] Relying on the hypothesis that $prBPP = prP$, the Merlin-Arthur protocol can be derandomized, and thus $S \in NTIME[\text{poly}(t)]/O(\log(s))$.

The point is that after the derandomization step we can eliminate the advice; that is, a circuit lower bound against *non-deterministic machines* with advice implies a closely-related circuit lower bound against non-deterministic machines without advice. (This argument relies on a simple reduction that does not follow through for Merlin-Arthur protocols, and can thus be performed only after the "derandomization" step; see Section 2.3 for details.) Moreover, the derandomization and the elimination of advice hold even if the lower bound is in the strong sense of Theorem 3. Thus, $prBPP = prP$ implies the existence of a set in $NTIME[\text{poly}(t)]$ that cannot be computed by circuits of size s, in the strong sense of Theorem 3.

2.2 Proof overview for Theorem 2

For simplicity, we overview the proof of Theorem 2 under the stronger hypothesis that $prBPP = prP$ (i.e., under the hypothesis of Theorem 1; the actual proof of Theorem 2 is very similar, just more careful with the parameters). To do so we need to define the circuit acceptance probability problem (or CAPP, in short): Given as input the description of a 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

[3] For simplicity, in this paper we do not consider restricted classes of circuits (such as classes with constant depth or limited fan-out). However, the results in the paper extend to many restricted circuit classes in a straightforward way, using the PCP of Ben-Sasson and Viola (see discussion in [BV14]).

[4] For a formal definition of the foregoing "strong sense" of failing to compute S, see Definition 15.

most $1/3$. The problem can be easily solved using *randomness* (by sampling inputs for C), but it is not a-priori clear how to solve it *deterministically*. In fact, a deterministic polynomial-time algorithm for CAPP exists if and only if $pr\mathcal{BPP} = pr\mathcal{P}$ (see Proposition 5). Therefore (since we assume that $pr\mathcal{BPP} = pr\mathcal{P}$), the starting point of our argument is the hypothesis that CAPP can be solved in deterministic polynomial time.

The proof closely follows the celebrated proof strategy of Williams [Wil13], but with a different setting of parameters than is typically used in this proof strategy. Our goal is to prove that $NTIME[n^{O(f(n))}]$ is not contained in $\mathcal{P}/\text{poly}$, where f is some very small function (e.g., $f(n) = \log^*(n)$). Assuming towards a contradiction that $NTIME[n^{O(f(n))}] \subseteq \mathcal{P}/\text{poly}$, we will construct a non-deterministic algorithm for any $L \in NTIME[n^{f(n)}]$ that runs in time noticeably smaller than $n^{f(n)}$. This will contradict the non-deterministic time hierarchy [Coo72].

The key technical tool that allows us construct such an algorithm is called an "easy witness lemma" (see, e.g., [IKW02]). Loosely speaking, such a lemma asserts that under seemingly-unlikely hypotheses (such as $NTIME[n^{O(f(n))}] \subseteq \mathcal{P}/\text{poly}$), there exist *small* circuits that encode *large* witnesses in proof systems (e.g., for $NTIME[n^{f(n)}]$). Specifically, the proof of Theorem 1 crucially relies on a new easy witness lemma, which was very recently proved by Murray and Williams [MW17], and has significantly better parameters than previous lemmas. Given our hypothesis, the new lemma implies that for every $L \in NTIME[n^{O(f(n))}]$, every verifier V for L and every $x \in L$, there exists a circuit $P_x \in \mathcal{P}/\text{poly}$ that encodes a witness π_x such that $V(x, \pi_x)$ accepts (see Lemma 9 for a precise statement).[5] The point is that witnesses for the verifier V are a-priori of size $n^{O(f(n))}$, 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.

Williams' idea is to use the existence of such "compressible" witnesses in order to construct a quick non-deteministic machine for any $L \in NTIME[n^{f(n)}]$. Specifically, fix any $L \in NTIME[t]$, where $t(n) = n^{f(n)}$. Also fix a PCP system for L with a verifier V that runs in time $t_V = \text{poly}(n, \log(t)) = n^{o(f(n))}$ and uses $\ell = O(\log(t))$ random bits. (For concreteness, we use the PCP of Ben-Sasson and Viola [BV14], but previous ones such as [BGH+05] also suffice for the proof.)

Given input $x \in \{0,1\}^n$, the non-deterministic machine M will first guess a witness for x, and then verify the witness using the verifier V. However, it will perform both tasks in a very efficient manner. First, instead of guessing a t-bit witness, the machine M will guess a (much-smaller) description of a circuit $P_x \in \mathcal{P}/\text{poly}$ that encodes a witness; if $x \in L$, then such P_x exists by the easy witness lemma.[6] Secondly, instead of directly using the verifier V to verify P_x, the machine will construct a circuit $C_x^{P_x}$ that,

[5] 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). We mention that in the proof we will actually assume that $NTIME[n^{f(n)}]$ is not contained in $SIZE[n^{g(n)}]$, for some $g(n) \ll f(n)$, and deduce the existence of witness circuits of size $n^{g(n)}$. We ignore this minor issue in the high-level overview.

[6] Actually, to apply the easy witness lemma we 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 $2^\ell \cdot t_V = n^{O(f(n))}$, and we can invoke the lemma since we assumed that $NTIME[n^{O(f(n))}] \subseteq \mathcal{P}/\text{poly}$.

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; the machine M will then use the CAPP algorithm on the circuit $C_x^{P_x}$ to determine whether the verifier is likely to accept x or to reject x. More specifically, the machine M acts as follows:

1. The machine non-deterministically guesses a circuit $P_x \in \mathcal{P}/\text{poly}$ (in the hope that P_x represents a proof for x acceptable by V).
2. The machine constructs a circuit $C_x^{P_x}$ that gets as input $r \in \{0,1\}^{\ell(n)}$, and simulates the execution of the verifier V on input x, randomness r, and with oracle access to the proof represented by P_x; that is, $C_x^{P_x}(r) = V^{P_x}(x,r)$.
3. The machine runs the CAPP algorithm on the circuit $C_x^{P_x}$, and outputs the decision of the algorithm.

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 to the soundness of the PCP verifier), and thus the machine M will reject x. Since P_x is of polynomial size and the verifier runs in time $t_V(n)$, the size of $C_x^{P_x}$ is at most $t_V(n) \cdot \text{poly}(n) = n^{o(f(n))}$. Therefore, the CAPP algorithm that gets $C_x^{P_x}$ as input also runs in time $n^{o(f(n))}$. The machine M thus decides L in non-deterministic time $n^{o(f(n))}$, which contradicts the non-deterministic time hierarchy theorem.

2.3 Proof overview for Theorem 3

In Theorem 3 we assume that $pr\mathcal{BPP} = pr\mathcal{P}$, and deduce a lower bound for polynomial-sized circuits against a set $S \in NTIME[n^{\omega(1)}]$ that is stronger than a standard lower bound. For the purposes of the high-level overview, we say that a set S is *not* in the class $\texttt{i.o.}_{[q]}SIZE[s]$ if for every circuit family of size s, and every sufficiently large n, the circuit family fails to decide S on some input length in the interval $[n, q(n)]$.[7]

Fixing $t(n) = n^{\omega(1)}$ and $s(n) = n^{\omega(1)} \ll t(n)$, we want to prove that if $pr\mathcal{BPP} = pr\mathcal{P}$, then there exists a set $S \in NTIME[t] \setminus \texttt{i.o.}_{[\text{poly}(s)]}SIZE[s]$. The starting point of the proof is Murray and Williams' strengthening of Santhanam's lower bound, which asserts (unconditionally) that there exists a set S that can be computed by Merlin-Arthur protocols running in time t with $O(\log(s))$ bits of non-uniform advice (i.e., $S \in MATIME[t]/O(\log(s))$[8] such that $S \notin \texttt{i.o.}_{[\text{poly}(s)]}SIZE[s]$ (see Theorem 16).

The first observation in the proof is that if $pr\mathcal{BPP} = pr\mathcal{P}$, then Merlin-Arthur protocols can be derandomized, in a straightforward way: Specifically, the residual decision

[7] As mentioned in Footnote 2, the actual lower bound is even slightly stronger, since it asserts that the circuit family fails to compute S on (at least) one of the "end-points" of the interval. For further details see Section 5, in which we also formally define the notation $S \notin \texttt{i.o.}_{[q]}SIZE[s]$ accordingly (see Definition 15).

[8] For a standard definition of the class $MATIME[t]/O(\log(s))$, see Definition 10.

problem that is solved by the probabilistic t-time verifier can be solved deterministically in time $\text{poly}(t)$ (see Proposition 17). Hence, under this hypothesis, the set S is in $NTIME[\text{poly}(t)]/O(\log(s)) \setminus \texttt{i.o.}_{[\text{poly}(s)]}SIZE[s]$.

The second observation is that if $NTIME[\text{poly}(t)]/O(\log(s)) \not\subseteq \texttt{i.o.}_{[\text{poly}(s)]}SIZE[s]$, then $NTIME[\text{poly}(t)]$ (without the advice) is not contained in $\texttt{i.o.}_{[\text{poly}(s)]}SIZE[s']$, for s' that is moderately smaller than s (see Proposition 19). This "elimination of advice" uses a contrapositive argument: Assuming that $NTIME[\text{poly}(t)] \subseteq \texttt{i.o.}_{[\text{poly}(s)]}SIZE[s']$, we construct a circuit family of size s for any $S \in NTIME[\text{poly}(t)]/O(\log(s))$, as follows. Consider the non-deterministic machine M that decides S with advice $\{a_n\}$, and let $S^{\texttt{adv}}$ be the set of pairs (x, σ) such that $|\sigma| = O(\log(s(|x|)))$, and M (non-deterministically) accepts x when given advice σ. Note that $S^{\texttt{adv}}$ can be decided by a non-determinstic machine that simulates M (and requires no advice), and thus, by our hypothesis, $S^{\texttt{adv}}$ can also be decided on infinitely-many intervals of length $\text{poly}(s(n))$ by a circuit family $\{C_n\}$ of size s'.[9] By hard-wiring the "correct" advice string a_n into each C_n, we obtain a circuit family $\{C'_n\}$ of size $s > s'$ that decides S on infinitely-many intervals of length moderately smaller than $\text{poly}(s(n))$. (See Propositions 18 and 19 for precise details.)

Let us recap: The initial (unconditional) lower bound asserts that there exists $S \in MATIME[t]/O(\log(s))$ such that $S \notin \texttt{i.o.}_{[\text{poly}(s)]}SIZE[s]$. The first observation, along with the hypothesis that $pr\mathcal{BPP} = pr\mathcal{P}$, implies that $S \in NTIME[\text{poly}(t)]/O(\log(s)) \setminus \texttt{i.o.}_{[\text{poly}(s)]}SIZE[s]$. And finally, the second observation implies that $NTIME[\text{poly}(t)]$ is not contained in $\texttt{i.o.}_{[\text{poly}(s)]}SIZE[s'] \supseteq \texttt{i.o.}_{[\text{poly}(s)]}\mathcal{P}/\text{poly}$.

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, 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). Moreover, we consider some fixed standard form of representation for 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.

[9] Note that the foregoing argument only follows through after the "derandomization" (i.e., for $NTIME$ and not for $MATIME$). This is the case since when dealing with probabilistic machines, it is not clear how to define $S^{\texttt{adv}}$ 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 clearly accept or reject some inputs).

Definition 4 *(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 5 *(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 5 see any standard textbook on the subject (e.g. [Vad12, Cor. 2.31], [Gol08, Exer. 6.14]). In Proposition 5 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 [MW17]

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

Definition 6 *(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 7 *(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.*

In Definitions 6 and 7 we considered verifiers that are deterministic algorithms that get the witness as an explicit input. As outlined in Section 2, in the proof we will consider PCP verifiers (which are probabilistic algorithms, and only get oracle access to their witness). However, we will not consider witness circuits for these PCP verifiers, but rather for deterministic verifiers (with explicit inputs) that are derived from the PCP verifiers (see the proof of Theorem 12 for precise details).

Let us now state the new easy witness lemma of [MW17]. 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 8 *(sufficiently gapped functions). Let $\gamma, \gamma', \gamma'' \in \mathbb{N}$ be universal constants.[10] 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 9 *(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 8. 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., MA verifiers) that receive non-uniform advice.

Definition 10 *(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 $\Pr[V(x, w, a_{|x|}) = 1] \leq 1/3$.*

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 $MA = \bigcup_{c \in \mathbb{N}} MATIME[n^c]$, and also note that the definition of MA does not change if we insist on perfect completeness (see, e.g., [Gol08, Exer. 6.12(2)]).

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

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

Proposition 11 *(a barrier for "derandomization implies lower bounds" statements). If the conditional statement "$\mathcal{BPP} = \mathcal{P} \Longrightarrow \mathcal{P} \neq \mathcal{NP}$" holds, then $\mathcal{P} \neq \mathcal{NP}$.*

[10] 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. Assume towards a contradiction that $\mathcal{P} = \mathcal{NP}$. Then, we have that $\mathcal{P} = \mathcal{BPP}$ (since $\mathcal{BPP}$ is contained in the polynomial-time hierarchy [Sip83; Lau83], and the hierarchy collapses to $\mathcal{P}$ if $\mathcal{P} = \mathcal{NP}$). However, we can now use the hypothesized conditional statement to deduce that $\mathcal{P} \neq \mathcal{NP}$, which is a contradiction. ■

4 Proof of Theorem 2

Our first step is to prove a 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 [MW17]. (In their original paper, Murray and Williams [MW17] considered two specific parameter settings.) We then prove Theorem 2 as a corollary.

4.1 A parametrized "derandomization implies lower bounds" theorem

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

Theorem 12 *(derandomization implies lower bounds, with flexible 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 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 $\epsilon > 0$ it holds that*

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

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.

To do so, consider the PCP verifier of [BV14] for L, denoted by V. On inputs of length n, the verifier V runs in time $\text{poly}(n, \log(t'(n)))$, uses $\ell = \log(t'(n)) + O(\log\log(t'(n)))$

[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.

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 9), 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 9 to the deterministic verifier V' that enumerates the random coins of V, which runs in time $2^{\ell} \cdot \text{poly}(n, \log(t')) = \text{poly}(t') = \text{poly}(t)$. 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 CAPP algorithm that exists by the hypothesis, and outputs the decision of the algorithm. By the properties of the PCP verifier and of the CAPP algorithm, if $x \in L$ then for some guess of P_x the machine will accept x, and if $x \notin L$ then for any guess of P_x the machine 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 $m(n) = \text{poly}(n, \log(t')) \cdot \hat{s}(n)$; thus, its representation size is $\text{poly}(m(n))$. Therefore, the circuit $C_x^{P_x}$ can be constructed in time $\text{poly}(m(n))$, 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. ■

4.2 Theorem 2 as a corollary of Theorem 12

We now prove Theorem 2 as a corollary of Theorem 12. Recall that in Theorem 2 we asserted that $pr\mathcal{BPP} = pr\mathcal{P}$ implies 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. In the proof, instead of proving that $NTIME[n^{f(n)}] \not\subseteq \mathcal{P}/\text{poly}$, we will actually prove the stronger statement $NTIME[n^{f(n)}] \not\subseteq SIZE[n^{g(n)}]$, 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 13 *(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

[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.

gapped, which enforces time-constructibility constraints); for a precise (and tedious) discussion, see Appendix A. We can now formally state Theorem 2 and prove it:

Corollary 14 *(Theorem 2, restated formally). Assume that $(1,1/3)$-CAPP can be solved by a deterministic algorithm with running time $T(m,v) \leq 2^{(1-\epsilon)\cdot v} \cdot \mathrm{poly}(m)$, for some constant $\epsilon > 0$. Then, for every admissible function f there exists a set in $NTIME[n^{O(f(n))}] \setminus \mathcal{P}/\mathrm{poly}$.*

Recall that by Proposition 5, if $prBPP = prP$, then there exists a deterministic algorithm for $(1,1/3)$-CAPP running in time $\mathrm{poly}(m)$ (i.e., polynomial in the size of the circuit). Therefore, Theorem 1 follows from Corollary 14.

Proof of Corollary 14. Since f is admissible, there exists a function g that satisfies the requirements of Definition 13. We use Theorem 12 with the functions $t(n) = n^{f(n)}$ and $s(n) = n^{g(n)}$.

Let us verify that the hypotheses of Theorem 12 hold. Since f is admissible we have that s and t are sufficiently gapped, and indeed we also have that $s(n) > n$. Also, for any constants $c \in \mathbb{N}$ and $\alpha < 1$ it holds that

$$\begin{aligned} T\Big((n \cdot \hat{s}(n) \cdot \log(t(n)))^c,\ \ \alpha \cdot \log(t(n))\Big) &< \mathrm{poly}(\hat{s}(n)) \cdot \mathrm{poly}\log(t(n)) \cdot t^{(1-\epsilon)\cdot\alpha} \\ &< n^{(1-\epsilon+o(1))\cdot f(n)}\ , \end{aligned}$$

which is $t(n)^{(1-\Omega(1))\cdot\alpha}$. (The first inequality relies on the hypothesis regarding T and on the fact that $\hat{s}$ is super-constant, and the second equality relies on the hypothesis that f is admissible, which implies that $\hat{s}(n) = n^{o(f(n))}$.)

Thus, Theorem 12 implies that there exists a set in $NTIME[n^{O(f(n))}] \setminus SIZE[n^{g(n)}]$. Since $g(n) = \omega(1)$, in particular this set does not belong to $\mathcal{P}/\mathrm{poly}$. ■

Additional relaxations of the hypotheses in Theorem 12. Since the proof of Theorem 12 relies on the strategy of [Wil13], it is well-known that the hypothesis of the theorem can be further relaxed. We now mention two (known) relaxations, in the hope that they might be useful in some settings (e.g., when we are interested in proving lower bounds for restricted circuit classes; see Footnote 3).

First, we do not have to *unconditionally* assume that the CAPP algorithm exists: It suffices to assume that the algorithm exists *under the hypothesis that* $NTIME[t(n)^{c'}] \subseteq SIZE[s(n)]$. And secondly, since we are using the CAPP algorithm as a sub-routine of a non-deterministic machine, the CAPP algorithm itself can also be non-deterministic. (However, the non-determinism should help the algorithm accept every circuit with acceptance probability one, and reject every circuit with low acceptance probability; it is not a-priori clear how non-determinism can be useful for such a task.)

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 every sufficiently large $n \in \mathbb{N}$, the family fails to decide S on some input length in the interval $[n, n^{\omega(1)}]$. Our actual conclusion will be slightly stronger: We will conclude that for every sufficiently large $n \in \mathbb{N}$, the circuit family fails on at least one of the "end-points" of the interval; that is, either on input length n, or on input length $q(n) = n^{\omega(1)}$ (or on both). This is equivalent to saying that there does not exist an infinite set of pairs $(n, q(n))$ such that the circuit family correctly decides S on both input lengths in the pair. This leads to the following definition:

Definition 15 *(a stronger notion of infinitely-often computation). For $s, q : \mathbb{N} \to \mathbb{N}$ and $S \subseteq \{0,1\}^*$, we say that $S \in \mathtt{i.o.}_{[q]} SIZE[s]$ if there exists an infinite set $I \subseteq \mathbb{N}$ and a circuit family $\{C_n\}_{n \in \mathbb{N}}$ of size at most s such that for every $n \in I$, it holds that:*

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

Indeed, Definition 15 implies the following: If $S \notin \mathtt{i.o.}_{[q]} SIZE[s]$, 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 $q(n)$ (or on both).

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 [MW17], 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 16 *(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 $t = \mathrm{poly}(s(\mathrm{poly}(s)))$ (for sufficiently large polynomials that do not depend on s). Then, there exists a set $S \in MATIME[t]/O(\log(s))$ such that $S \notin \mathtt{i.o.}_{[\mathrm{poly}(s)]} SIZE[s]$.*

As mentioned in Section 2.3, the first observation in the proof is that if $pr\mathcal{BPP} = pr\mathcal{P}$ then we can derandomize MA verifiers that receive non-uniform advice.

Proposition 17 *(derandomization of MA with advice). If $pr\mathcal{BPP} = pr\mathcal{P}$, 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. Let S be a set in $MATIME[t]/\ell$, let V be the $MATIME[t]/\ell$ 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 $V(x, w, r, a_n)$. Since V is a verifier for S and a_n is the "good" advice for V, if $x \in S$ then there exists w such that the acceptance probability of C is at least $2/3$, and if $x \notin S$ then for any w the acceptance probability of C is at most $1/3$. Since $pr\mathcal{BPP} = pr\mathcal{P}$ (and using Proposition 5), the machine M can distinguish between the two foregoing cases using a deterministic polynomial-time CAPP

algorithm. The running time of the machine M is dominated by the running time of the latter algorithm, which is at most $\text{poly}(t(n))$. ■

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 the 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 15.

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 15.

Proposition 18 *(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 $S^{\mathtt{adv}}$, 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 $S^{\mathtt{adv}}$ 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 $S^{\mathtt{adv}} \in NTIME[O(t)]$, 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 $S^{\mathtt{adv}}_n$. 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 18, which considers possibly long advice strings, and refers to separations in the sense of Definition 15.

Proposition 19 *(eliminating the advice). Let $s_0, s, \ell, t, q : \mathbb{N} \to \mathbb{N}$ be functions such that t is super-linear and increasing, and q, 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 $s_0(q(n)) \geq s(2q(2n))$. Further assume that $NTIME[t]/\ell \not\subseteq \mathtt{i.o.}_{[q]} SIZE[s_0]$. Then, $NTIME[O(t)] \not\subseteq \mathtt{i.o.}_{[2q]} SIZE[s]$.*

We comment that a statement that is more general than the one in Proposition 19 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 19 suffices for our parameter setting, and for simplicity, we avoid such generalizations.

Proof of Proposition 19. Assuming that $NTIME[O(t)] \subseteq \texttt{i.o.}_{[2q]}SIZE[s]$, we prove that $NTIME[t]/\ell \subseteq \texttt{i.o.}_{[q]}SIZE[s_0]$. 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 $q(n)$.

The proof follows the same approach as the proof of Proposition 18, but the implementation is more cumbersome. The source of trouble is that we assume that $S^{\texttt{adv}} \in \texttt{i.o.}_{[2q]}SIZE[s]$ (rather than $S^{\texttt{adv}} \in SIZE[s]$), which only guarantees the existence of an *infinite set* $I \subseteq \mathbb{N}$ of input lengths for which $S^{\texttt{adv}}$ 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 has small circuits. To overcome this problem, we will "embed" all sufficiently short pairs (x, σ) (where $|\sigma| = \ell(|x|)$ and $|x| + |\sigma| < n$) into $\{0,1\}^n$, and define $S^{\texttt{adv}}_n$ such that deciding $S^{\texttt{adv}}_n$ allows to determine the output of M on input x with advice σ for all sufficiently short pairs (x, σ). Details follow.

Consider the following set $S^{\texttt{adv}}$. Let M be a t-time non-deterministic machine and let $\{a_n\}$ be a sequence of 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 $S^{\texttt{adv}}_n$ 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 $S^{\texttt{adv}}_n$ as 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 $S^{\texttt{adv}} \in NTIME[O(t)]$. 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 $NTIME[O(t)] \subseteq \texttt{i.o.}_{[2q]}SIZE[s]$, 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 $S^{\texttt{adv}}_n$; and
2. $C_{2q(n)} : \{0,1\}^{2q(n)} \to \{0,1\}$ correctly computes $S^{\texttt{adv}}_{2q(n)}$.

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

Claim 19.1. *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 $S^{\texttt{adv}}_n$. 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 $S^{\texttt{adv}}_n$. 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 19.1, 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 $\ell(m) < m/2$.)

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

2. There exists a circuit $C_{q(m)} : \{0,1\}^{q(m)} \to \{0,1\}$ of size $s_0(q(m))$ that decides $S_{q(m)}$. To see this, recall that there exists a circuit $C_{2q(n)}$ of size $s(2q(n))$ that decides $S^{\mathtt{adv}}_{2q(n)}$. We can invoke Claim 19.1 because $q(m) + 2\ell(q(m)) < 2q(m) < 2q(n)$. Also, relying on the fact that $m \geq n/2$ and on the hypotheses regarding s_0, s and q, we have that $s(2q(n)) \leq s(2q(2m)) \leq s_0(q(m))$.

It follows that $S \in \mathtt{i.o.}_{[q]} SIZE[s_0]$. ■

We now combine the foregoing ingredients into a proof of Theorem 3. The theorem that we obtain is similar in form to Theorem 12: Assuming a sufficiently strong derandomization hypothesis (in this case, that $pr\mathcal{BPP} = pr\mathcal{P}$), and taking two functions t and s with sufficient "gap" between them, we deduce that $NTIME[t] \not\subseteq \mathtt{i.o.}_{[s_0]} SIZE[s]$, where s_0 is a function moderately larger than s. (The fact that $s_0 > s$ is no coincidence; see discussion after the proof of Theorem 20.)

Theorem 20 *(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 $pr\mathcal{BPP} = pr\mathcal{P}$. Then, $NTIME[t] \not\subseteq \mathtt{i.o.}_{[\mathrm{poly}(s_0)]} SIZE[s]$.

Note that if the function s in Theorem 20 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 $NTIME[t] \not\subseteq \mathtt{i.o.}_{[\mathrm{poly}(s)]} SIZE[s]$.

Proof of Theorem 20. 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 16). By Theorem 16, there exists $S \in MATIME[t_0]/\ell \setminus \mathtt{i.o.}_{[(s_0)^c]} SIZE[s_0]$, for a sufficiently large constant $c \in \mathbb{N}$. By Proposition 17, and relying on the hypothesis that $pr\mathcal{BPP} = pr\mathcal{P}$, it holds that $S \in NTIME[\mathrm{poly}(t_0)]/\ell \setminus \mathtt{i.o.}_{[(s_0)^c]} SIZE[s_0]$.

We now want to use Proposition 19 to deduce that $NTIME[\mathrm{poly}(t_0)]$ is not contained in $\mathtt{i.o.}_{[\mathrm{poly}(s_0)]} SIZE[s]$. To do so, we just need to verify that the functions ℓ, s, s_0, and $(s_0)^c$ satisfy the hypothesis of Proposition 19. 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 $s(2s_0(2n)^c) \leq s(s_0(n)^{2c})^2 \leq s_0(s_0(n)^c)$. ■

One advantage of Theorem 20, in comparison to Theorem 12, is that the required "gap" between s and t (in order to conclude that $NTIME[t] \not\subseteq \mathtt{i.o.}_{[\mathrm{poly}(s)]} SIZE[s]$) depends on s in a milder way. In particular (and ignoring polynomial factors, for simplicity), in Theorem 23 we need that t will be larger than $s \circ s$ (i.e., than two compositions

of s), whereas in Theorem 12 we need that t will be larger than $\hat{s} \approx s \circ s \circ s$ (i.e., than three compositions of s).

As mentioned before the statement of Theorem 20, the "gap" between the input lengths n and $q(n) = \text{poly}(s_0(n))$ (on which any size-s circuit family is guaranteed to fail) in Theorem 20 is larger than the function s that bounds the size of the circuits. This is no coincidence: If the gap function q 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 $s(q^{-1}(\cdot))$).[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 also thanks Ryan Williams for several very helpful email exchanges, and in particular for stressing the point that the conclusion in the main theorem can be interpreted in two different ways (as explained in Section 1.3), and for suggesting to try and prove an "almost-everywhere" lower bound. The author is very grateful to Igor Oliveira for proposing the alternative proof of Theorem 1 (i.e., the one presented in Appendix B), and for his permission to include the alternative proof in this paper.

References

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

[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.

[BV14] 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.

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

[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.

[IW98] R. Impagliazzo and A. Wigderson. "Randomness vs. Time: De-Randomization Under a Uniform Assumption". In: *Proc. 39th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*. 1998, pp. 734–.

[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.

[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.

[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.

[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.

[Sip83] Michael Sipser. "A Complexity Theoretic Approach to Randomness". In: *Proc. 15th Annual ACM Symposium on Theory of Computing (STOC)*. 1983, pp. 330–335.

[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 Sufficient conditions for admissibility

The point of the current appendix is to show that essentially any increasing function $f(n) = \omega(1)$ such that $f(n) \leq n$ is admissible (in the sense of Definition 13).

Claim 21. *Let $f(n) = \omega(1)$ be any increasing function such that $f(n) \leq n$ 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 $s(n) \leq n^{\log\log(n)} < 2^{n/\gamma}/n$.

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 $c \geq 2$ such that $k(n) \leq \log^c(f(\log^{3c}(n)))$ (which in particular implies that $k(n) \leq \log^c(n)$), we have that

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

In particular, using Eq. (A.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. (A.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 $\hat{s}(n) < n^{\gamma' \cdot \log^8(f(\log^{12}(n)))} < n^{\gamma' \cdot \mathrm{poly}\log(f(n))} = n^{o(f(n))}$. ∎

Appendix B An alternative proof of Theorem 1

In this section we present an alternative proof of Theorem 1, which does not rely on the work of Murray and Williams [MW17], but rather on the work of Santhanam [San09]. The proof structure is very similar to the proof of Theorem 3 (which was described in Section 2.3), 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 [MW17]. Specifically, the starting point of the proof is the following:

Theorem 22 *(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 $MATIME[t]/1 \not\subseteq SIZE[s]$.*

The proof of Theorem 22 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 22, Proposition 17 (instantiated with the value $\ell = 1$), and Proposition 18.

Theorem 23 *(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 $t(n) = \text{poly}(s(\text{poly}(s(n))))$, for sufficiently large polynomials. Assume that $pr\mathcal{BPP} = pr\mathcal{P}$. Then, $NTIME[t] \not\subseteq SIZE[s]$.*

Proof. Let $s_0 = s^3$, and let $t_0 = \text{poly}(s_0(\text{poly}(s_0)))$, for sufficiently large polynomials. According to Theorem 22, there exists a set S in $MATIME[t_0]/1$ such that $S \notin SIZE[s_0]$. By Proposition 17, and relying on the hypothesis that $pr\mathcal{BPP} = pr\mathcal{P}$, it holds that $S \in NTIME[t_1]/1 \setminus SIZE[s_0]$, where $t_1 = \text{poly}(t_0)$. Using Proposition 18, it holds that $NTIME[t] \not\subseteq SIZE[s_1]$, where $t = O(t_1) = \text{poly}(s(\text{poly}(s)))$ 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 $NTIME[t] \not\subseteq SIZE[s]$. ∎

It is just left to detail the proof of Theorem 22. 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 $DSPACE[n^{\omega(1)}]$ under $n^{\omega(1)}$-time reductions.

Lemma 24 *(scaling the $\mathcal{PSPACE}$-complete set of [TV07]). There exists a set $L^{\text{TV}} \subseteq \{0,1\}^*$ 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 $L \in DSPACE[t]$ there exists a deterministic Turing machine R_L that runs in time $\text{poly}(t)$ such that for every $x \in \{0,1\}^*$ it holds that $x \in L \iff R_L(x) \in L^{\text{TV}}$.*
2. *On input $x \in \{0,1\}^*$, the machine M only issues queries of length $|x|$.*
3. *For any $x \in L^{\text{TV}}$ it holds that $\Pr[M^{\mathbf{1}_{L^{\text{TV}}}}(x) = 1] = 1$, where $\mathbf{1}_{L^{\text{TV}}} : \{0,1\}^n \to \{0,1\}$ is the indicator function of $L^{\text{TV}} \cap \{0,1\}^n$.*
4. *For any $x \notin L^{\text{TV}}$ and any $f : \{0,1\}^n \to \{0,1\}$ it holds that $\Pr[M^f(x) = 0] \geq 2/3$.*

Proof. We take L^{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^{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^{TV}. The first reduction maps $x \mapsto (x, 1^t)$, and since $L' \in \mathcal{PSPACE}$, there exists a second reduction

[15] The original statement asserts that any $x \notin L^{\text{TV}}$ 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.

of L' to L^{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 22. 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^{TV} be the set from Lemma 24. Our goal is to prove that there exists a set in $MATIME[t_2]/1$ that is not in $SIZE[t_0^{1/4}]$. The proof proceeds by a case analysis.

Case 1: $L^{\mathrm{TV}} \in SIZE[t_0]$**.** By a standard diagonalization argument, there exists a set $L^{\mathtt{diag}} \in DSPACE[t_1] \setminus SIZE[t_0]$. [16] Our main goal now will be to prove that $DSPACE[t_1] \subseteq MATIME[t_2]$, which will imply that $L^{\mathtt{diag}} \in MATIME[t_2] \setminus SIZE[t_0]$. (Indeed, in this case we are proving a stronger result, since the 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 $L \in DSPACE[t_1]$, and consider the following 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 24. Note that $n' = |x'| \leq \mathrm{poly}(t_1(n))$, and that $x \in L \iff x' \in L^{\mathrm{TV}}$. 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 24 on input x', while answering each oracle query of M using the circuit C.

Note that, since $L^{\mathrm{TV}} \in SIZE[t_0]$, there exists a circuit C over n' input bits of size $t_0(n')$ that correctly computes L^{TV} on inputs of length n'. Therefore, by Lemma 24, 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: $L^{\mathrm{TV}} \notin SIZE[t_0]$**.** In this case we show an explicit set $L^{\mathtt{pad}}$, which will be a padded version of L^{TV}, such that $L^{\mathtt{pad}}$ can be decided in $MATIME[t_2]$ with one bit of advice, but cannot be decided by circuits of size $s = t_0^{1/4}$. To do so, let $\mathtt{sz}_{\mathrm{TV}} : \mathbb{N} \to \mathbb{N}$ be such that $\mathtt{sz}_{\mathrm{TV}}(n)$ is the minimum circuit size for $L_n^{\mathrm{TV}} = L^{\mathrm{TV}} \cap \{0,1\}^n$. 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, 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:

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

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 $L_m^{\mathtt{pad}}$

[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 $L^{\mathtt{diag}} \in DSPACE[t_1]$ follows the well-known idea used in Kannan's theorem (see, e.g., [Juk12, Lem. 20.12]).

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 $L^{\mathtt{TV}}_n$, 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 $t_0(n+p) \leq \mathtt{sz}_{\mathtt{TV}}{}^3(n) < t_0(n+2p)$; there exists such a p since $t_0(n+2^{\lceil \log(n) \rceil}) \leq t_0(n)^3 < \mathtt{sz}_{\mathtt{TV}}{}^3(n)$. 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 $L^{\mathtt{TV}}_n$. 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 $t_0(n+p)^{1/3} \leq \mathtt{sz}_{\mathtt{TV}}(n)$ 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 $L^{\mathtt{pad}}_m \neq \emptyset$; 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 24 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 $L^{\mathtt{TV}}_n$. 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 $\mathrm{poly}(n) \cdot \mathrm{poly}(t_0(2m)) \leq t_2(m)$. ■

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il