On Exponential-Time Hypotheses, Derandomization, and Circuit Lower Bounds
[Extended Abstract]

Lijie Chen*, Ron D. Rothblum†, Roei Tell‡, Eylon Yogev§
*Massachusetts Institute of Technology. Email: lijieche@mit.edu
†Technion. Email: rothblum@cs.technion.ac.il
‡Massachusetts Institute of Technology. Email: roeitell@gmail.com
§Boston University and Tel Aviv University. Email: eylony@gmail.com

2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS) | 978-1-7281-9621-3/20/$31.00 ©2020 IEEE | DOI: 10.1109/FOCS46700.2020.00010

***Abstract*—The Exponential-Time Hypothesis (ETH) is a strengthening of the $\mathcal{P} \neq \mathcal{NP}$ conjecture, stating that 3-SAT on n variables cannot be solved in (uniform) time $2^{\epsilon \cdot n}$, for some $\epsilon > 0$. In recent years, analogous hypotheses that are "exponentially-strong" forms of other classical complexity conjectures (such as $\mathcal{NP} \not\subseteq \mathcal{BPP}$ or $co\mathcal{NP} \not\subseteq \mathcal{NP}$) have also been introduced, and have become widely influential.**

In this work, we focus on the interaction of exponential-time hypotheses with the fundamental and closely-related questions of *derandomization and circuit lower bounds*. We show that even relatively-mild variants of exponential-time hypotheses have far-reaching implications to derandomization, circuit lower bounds, and the connections between the two. Specifically, we prove that:

1) **The Randomized Exponential-Time Hypothesis (rETH) implies that $\mathcal{BPP}$ can be simulated on "average-case" in *deterministic (nearly-)polynomial-time* (i.e., in time $2^{\tilde{O}(\log(n))} = n^{\log\log(n)^{O(1)}}$). The derandomization relies on a conditional construction of a pseudorandom generator with *near-exponential stretch* (i.e., with seed length $\tilde{O}(\log(n))$); this significantly improves the state-of-the-art in uniform "hardness-to-randomness" results, which previously only yielded pseudorandom generators with sub-exponential stretch from such hypotheses.**
2) **The Non-Deterministic Exponential-Time Hypothesis (NETH) implies that derandomization of $\mathcal{BPP}$ is *completely equivalent* to circuit lower bounds against $\mathcal{E}$, and in particular that pseudorandom generators are necessary for derandomization. In fact, we show that the foregoing equivalence follows from a *very weak version of* NETH, and we also show that this very weak version is necessary to prove a slightly stronger conclusion that we deduce from it.**

Lastly, we show that *disproving* certain exponential-time hypotheses requires proving breakthrough circuit lower bounds. In particular, if CircuitSAT for circuits over n bits of size $\mathrm{poly}(n)$ can be solved by *probabilistic algorithms* in time $2^{n/\mathrm{polylog}(n)}$, then $\mathcal{BPE}$ does not have circuits of quasilinear size.

***Keywords*-computational complexity**

I. Introduction

The Exponential-Time Hypothesis (ETH), introduced by Impagliazzo and Paturi [2] (and refined in [3]), conjectures that 3-SAT with n variables and $m = O(n)$ clauses cannot be deterministically solved in time less than $2^{\epsilon \cdot n}$, for some constant $\epsilon = \epsilon_{m/n} > 0$. The ETH may be viewed as an "exponentially-strong" version of $\mathcal{P} \neq \mathcal{NP}$, since it conjectures that a specific $\mathcal{NP}$-complete problem requires essentially exponential time to solve.

Since the introduction of ETH many related variants, which are also "exponentially-strong" versions of classical complexity-theoretic conjectures, have also been introduced. For example, the Randomized Exponential-Time Hypothesis (rETH), introduced in [4], conjectures that the same lower bound holds also for *probabilistic* algorithms (i.e., it is a strong version of $\mathcal{NP} \not\subseteq \mathcal{BPP}$). The Non-Deterministic Exponential-Time Hypothesis (NETH), introduced (implicitly) in [5], conjectures that co-3SAT (with n variables and $O(n)$ clauses) cannot be solved by non-deterministic machines running in time $2^{\epsilon \cdot n}$ for some constant $\epsilon > 0$ (i.e., it is a strong version of $co\mathcal{NP} \not\subseteq \mathcal{NP}$). The variations MAETH and AMETH are defined analogously (see [6][1]), and other variations conjecture similar lower bounds for seemingly-harder problems (e.g., for #3SAT; see [4]).

These Exponential-Time Hypotheses have been widely influential across different areas of complexity theory. Among the numerous fields to which they were applied so far are structural complexity (i.e., showing classes of problems that, conditioned on exponential-time hypotheses, are "exponentially-hard"), parameterized complexity, communication complexity, and fine-grained complexity; see, e.g., the surveys [7]–[10].

Exponential-time hypotheses focus on conjectured lower bounds for *uniform algorithms*. Two other fundamental questions in theoretical computer science are those of *derandomization*, which refers to the power of probabilistic algorithms; and of *circuit lower bounds*, which refers to the power of *non-uniform* circuits. Despite the central place of all three questions, the interactions of exponential-time hypotheses with derandomization and circuit lower bounds have yet to be systematically studied.

A full version of this paper is available online at ECCC [1].

[1] In [6], the introduction of these variants is credited to a private communication from Carmosino, Gao, Impagliazzo, Mihajlin, Paturi, and Schneider [5].

2575-8454/20/$31.00 ©2020 IEEE
DOI 10.1109/FOCS46700.2020.00010

A. Our results: Bird's eye

In this work we focus on the interactions between exponential-time hypotheses, derandomization, and circuit lower bounds. In a nutshell, our main contribution is showing that even *relatively-mild* variants of exponential-time hypotheses have *far-reaching consequences* on derandomization and circuit lower bounds.

Let us now give a brief overview of our specific results, before describing them in more detail in Sections I-B, I-C, and I-D. Our two main results are the following:

1) We show that rETH implies a *nearly-polynomial-time* average-case derandomization of $\mathcal{BPP}$. Specifically, assuming rETH,[2] we construct a pseudorandom generator for uniform circuits with *near-exponential stretch* (i.e., with seed length $\tilde{O}(\log(n))$) and with running time $2^{\tilde{O}(\log(n))} = n^{\log\log(n)^{O(1)}}$, and deduce that $\mathcal{BPP}$ can be decided, in average-case and infinitely-often, by deterministic algorithms that run in time $n^{\log\log(n)^{O(1)}}$ (see Theorem I.1). This significantly improves the state-of-the-art in the long line of *uniform "hardness-to-randomness"* results, which previously only yielded pseudorandom generators with at most a *sub-exponential* stretch from worst-case lower bounds for uniform probabilistic algorithms (i.e., for $\mathcal{BPTIME}$; see Section I-B for details). We also extend this result to deduce an "almost-always" derandomization of $\mathcal{BPP}$ from an "almost-always" lower bound (see Theorem I.2), which again improves on the state-of-the-art. See Section I-B for details.
2) Circuit lower bounds against $\mathcal{E}$ are well-known to yield pseudorandom generators for non-uniform circuits that can be used to derandomize $prBPP$ in the worst-case. An important open question is whether such lower bounds and pseudorandom generators are actually *necessary* for worst-case derandomization of $prBPP$. We show that a very weak version of NETH yields a positive answer to the foregoing question; specifically, to obtain a positive answer it suffices to assume that $\mathcal{E}$ cannot be computed by *small circuits* that are *uniformly generated* by a non-deterministic machine.[3] In fact, loosely speaking, we show that this weak version of NETH is both *sufficient and necessary* to show an equivalence between non-deterministic derandomization of $prBPP$ and circuit lower bounds against $\mathcal{E}$. See Section I-C for more details.

Lastly, in Section I-D we show that *disproving* a conjecture similar to rETH requires proving breakthrough circuit lower bounds. Specifically, we show that if there exists a *probabilistic algorithm* that solves CircuitSAT for circuits with n input bits and of size $\mathrm{poly}(n)$ in time $2^{n/\mathrm{polylog}(n)}$, then non-uniform circuits of quasilinear size cannot decide $\mathcal{BPE} \stackrel{\text{def}}{=\!=} \mathcal{BPTIME}[2^{O(n)}]$ (see Theorem I.6, and see the discussion in Section I-D for a comparison with the state-of-the-art).

Relation to Strong *Exponential Time Hypotheses:* The exponential-time hypotheses that we consider also have "strong" variants that conjecture a lower bound of $2^{(1-\epsilon)\cdot n}$, where $\epsilon > 0$ is arbitrarily small, for solving a corresponding problem (e.g., for solving SAT, *co*SAT, or #SAT; see, e.g., [10]). We emphasize that in this paper we focus only on the "non-strong" variants that conjecture lower bounds of $2^{\epsilon \cdot n}$ for some $\epsilon > 0$; these are indeed significantly weaker than their "strong" counterparts; in fact, some "strong" variants of standard exponential-time hypotheses are simply known to be false (see [6]).

We mention that a recent work of Carmosino, Impagliazzo, and Sabin [11] studied the implications of hypotheses in *fine-grained complexity* on derandomization. These fine-grained hypotheses are implied by the "strong" version of rETH (i.e., by rSETH), but are not known to follow from the "non-strong" versions that we consider in this paper. We will refer again to their results in Section I-B.

B. rETH *and pseudorandom generators for uniform circuits*

The first hypothesis that we study is rETH, which (slightly changing notation from above) asserts that probabilistic algorithms cannot decide if a given 3-SAT formula with v variables and $O(v)$ clauses is satisfiable in time less than $2^{\epsilon \cdot v}$, for some constant $\epsilon > 0$. Note that such a formula can be represented with $n = O(v \cdot \log(v))$ bits, and therefore the conjectured lower bound as a function of the input length is $2^{\epsilon \cdot (n/\log(n))}$.

Intuitively, using "hardness-to-randomness" results, we expect that such a strong lower bound would imply a strong derandomization result. For context, recall that in *non-uniform* hardness-to-randomness results (following [12]), lower bounds for non-uniform circuits yield pseudorandom generators (PRGs) that "fool" non-uniform distinguishers. Moreover, these results "scale smoothly" such that lower bounds for larger circuits yield PRGs with longer stretch (see [13] for an essentially optimal trade-off); at the extreme, if $\mathcal{E}$ is hard almost-always for exponential-sized circuits, then we obtain PRGs with exponential stretch and deduce that $prBPP = prP$ (see [14]).

The key problem, however, is that the long line-of-works concerning *uniform* "hardness-to-randomness" did not yield such smooth trade-offs so far (see [11], [15]–[23]). Ideally, given an exponential lower bound for uniform probabilistic

[2] We will in fact consider a hypothesis that is weaker (qualitatively and quantitatively), and conjectures that the specific $\mathcal{PSPACE}$-complete problem Totally Quantified Boolean Formula (TQBF) cannot be solved in probabilistic time $2^{n/\mathrm{polylog}(n)}$. (Recall that TQBF is the set of 3-SAT formulas φ over variables $w_1, ..., w_v$ such that $\forall w_1 \exists w_2 \forall w_3 ..., \varphi(w_1, ..., w_v) = 1$, and that 3SAT reduces to TQBF in linear time; see [1, Definition 4.6].)

[3] That is, we assume that the following statement does not hold: For any $L \in \mathcal{E}$ there is a uniform machine M_L that runs in time $\ll 2^n$ and uses its non-determinism to generate a single small circuit C_L that decides L on *all* n-bit inputs; for example, M_L runs in sub-exponential time and C_L is of polynomial size. (See Section I-C.)

algorithms (such as $\mathcal{E} \not\subseteq \texttt{i.o.}\mathcal{BPTIME}[2^{\epsilon \cdot n}]$) we would like to deduce that there exists a PRG with exponential stretch for uniform circuits, and consequently that $\mathcal{BPP} = \mathcal{P}$ in "average-case".[4] However, prior to the current work, the state-of-the-art (by Trevisan and Vadhan [20]) could at best yield PRGs with *sub-exponential stretch* (i.e., with seed length $\mathrm{polylog}(n)$), even if the hypothesis refers to an exponential lower bound. Moreover, the best currently-known PRG only works infinitely-often, even when we assume that the "hard" function cannot be computed by probabilistic algorithms on almost all input lengths.

Previous works bypassed these two obstacles in various indirect ways. Goldreich [23] relied on the (much) stronger hypothesis $pr\mathcal{BPP} = pr\mathcal{P}$ to construct an "almost-always" PRG with exponential stretch for uniform circuits. Similarly, Carmosino, Impagliazzo, and Sabin [11] relied on hypotheses from *fine-grained complexity* (recall that these are qualitatively strong, and implied by the "strong" version of rETH, i.e. by rSETH) to bypass both obstacles and derandomize $\mathcal{BPP}$ "almost-always" on average-case in polynomial time; however, their derandomization does not rely on a PRG construction, and satisfies a weaker notion of average-case derandomization than the notion that we use.[5] Gutfreund and Vadhan [22] bypassed the "almost-always" barrier by deducing (subexponential-time) derandomization of $\mathcal{RP}$ rather than of $\mathcal{BPP}$ (see details below). Lastly, a line-of-works dealing with uniform "hardness-to-randomness" for $\mathcal{AM}$ (rather than for $\mathcal{BPP}$) was able to bypass both obstacles in this context (see, e.g., [18], [19], [21]).

In this work we tackle both obstacles directly. First, we establish for the first time that hardness assumptions for $\mathcal{BPTIME}$ yield a pseudorandom generator for uniform circuits with *near-exponential stretch* (i.e., with seed length $\tilde{O}(\log(n))$), which can be used for average-case derandomization of $\mathcal{BPP}$ in nearly-polynomial-time (i.e., in time $2^{\tilde{O}(\log(n))} = n^{\mathrm{loglog}(n)^{O(1)}}$). Specifically, we start from the hypothesis that the Totally Quantified Boolean Formula (TQBF) problem cannot be solved by probabilistic algorithms that run in time $2^{n/\mathrm{polylog}(n)}$; this hypothesis is *weaker* than rETH (since 3-SAT reduces to TQBF with a linear overhead). Under this hypothesis, we show that there exists a PRG for uniform circuits with seed length $\widetilde{O}(\log(n))$ that is computable in time $2^{\tilde{O}(\log(n))} = n^{\mathrm{loglog}(n)^{O(1)}}$.

[4]Throughout the paper, when we say that a PRG is ϵ-pseudorandom for *uniform circuits*, we mean that for every efficiently-samplable distribution over circuits, the probability over choice of circuit that the circuit distinguishes the output of the PRG from uniform with advantage more than ϵ is at most ϵ (see [1, Definitions 3.6 and 3.7]). The existence of such PRGs implies an "average-case" derandomization of $\mathcal{BPP}$ in the following sense: For every $L \in \mathcal{BPP}$ there exists an efficient deterministic algorithm D such that every probabilistic algorithm that gets input 1^n and tries to find $x \in \{0,1\}^n$ such that $D(x) \neq L(x)$ has a small probability of success (see, e.g., [23, Prop. 4.4]).

[5]Specifically, they deduce an average-case derandomization of $\mathcal{BPP}$ with respect to the *uniform* distribution, rather than with respect to every polynomial-time-samplable distribution.

Theorem I.1 (rETH $\Rightarrow$ PRG with almost-exponential stretch for uniform circuits; informal)**.** *Suppose that there exists $T(n) = 2^{n/\mathrm{polylog}(n)}$ such that* TQBF $\notin \mathcal{BPTIME}[T]$. *Then, for every $t(n) = n^{\mathrm{polyloglog}(n)}$, there exists a PRG that has seed length $\widetilde{O}(\log(n))$, runs in time $n^{\mathrm{polyloglog}(n)}$, and is infinitely-often $(1/t)$-pseudorandom for every distribution over circuits that can be sampled in time t with $\log(t)$ bits of non-uniform advice.*

The proof of Theorem I.1 is based on careful refinements of the proof framework of [15], using new technical tools that we construct. The latter tools significantly refine and strengthen the technical tools that were used by [20] to obtain the previously-best uniform hardness-to-randomness tradeoff. For high-level overviews of the proof of Theorem I.1 (and of the new constructions), see Section II-A.

Overcoming the "infinitely-often" barrier: The hypothesis in Theorem I.1 is that any probabilistic algorithm that runs in time $2^{n/\mathrm{polylog}(n)}$ fails to compute TQBF *infinitely-often*, and the corresponding conclusion is that the PRG "fools" uniform circuits only *infinitely-often*. This is identical to all previous uniform "hardness-to-randomness" results that used the [15] proof framework.[6]

Gutfreund and Vadhan [22, Sec 6] showed one way to overcome this "infinitely-often" barrier, by deducing almost-always average-case derandomization of $\mathcal{RP}$ (rather than of $\mathcal{BPP}$) under an almost-always lower bound hypothesis; as in previous results, their derandomization is relatively slow (i.e., it works in sub-exponential time). Combining their ideas with the techniques underlying Theorem I.1, we prove that under the hypothesis that rETH holds almost-always, $\mathcal{RP}$ can be derandomized almost-always in average-case and in (nearly-)polynomial time (see [1, Theorem 4.14]).

In addition, their techniques can be adapted to yield an almost-always PRG (from an almost-always lower bound hypothesis) that uses $O(\log(n))$ bits of non-uniform advice. We are able to significantly improve this: Assuming that every probabilistic algorithm that runs in time $2^{n/\mathrm{polylog}(n)}$ fails to decide TQBF on almost all input lengths, we prove that $\mathcal{BPP}$ can be derandomized in average-case and almost-always, using only a *triply-logarithmic* number (i.e., $O(\mathrm{logloglog}(n))$) of advice bits.

Theorem I.2 (aa-rETH $\Rightarrow$ almost-always derandomization in time $n^{\mathrm{polyloglog}(n)}$; informal)**.** *Assume that for some $T(n) = 2^{n/\mathrm{polylog}(n)}$ it holds that* TQBF $\notin$ $\texttt{i.o.}\mathcal{BPTIME}[T]$, *and let $t(n) = n^{\mathrm{polyloglog}(n)}$. Then, for every $L \in \mathcal{BPTIME}[t]$ and every distribution ensemble $\mathcal{X} = \{\mathcal{X}_n \subset \{0,1\}^n\}$ such that $x \sim \mathcal{X}_n$ can be sampled in time $t(n)$, there exists a deterministic algorithm $D = D_{\mathcal{X}}$ that runs in time $n^{\mathrm{polyloglog}(n)}$ and uses $O(\mathrm{logloglog}(n))$ bits of non-uniform advice such that for* almost all input

[6]Other proof strategies (which use different hypotheses) were able to support an "almost-always" conclusion, albeit not necessarily a PRG, from an "almost-always" hypothesis (see [11], [19]).

lengths $n \in \mathbb{N}$ *it holds that* $\Pr_{x\sim\mathcal{X}_n}[D(x) \neq L(x)] < 1/t(n)$.

Remark: Non-deterministic extensions: We note that "scaled-up" versions of Theorems I.1 and I.2 for *non-deterministic settings* follow easily from known results; that is, assuming lower bounds for non-deterministic uniform algorithms, we can deduce strong derandomization of corresponding non-deterministic classes. First, from the hypothesis MAETH[7] we can deduce strong circuit lower bounds, and hence also *worst-case* derandomization of $pr\mathcal{BPP}$ and of $pr\mathcal{MA}$ (this uses relatively standard Karp-Lipton-style arguments, following [24]; see [1, Appendix A] for details and for a related result). Similarly, as shown by Gutfreund, Shaltiel, and Ta-Shma [19], a suitable variant of AMETH implies an average-case derandomization of $\mathcal{AM}$.

C. NETH *and an equivalence of derandomization and circuit lower bounds*

Let us now consider the Non-Deterministic Exponential-Time Hypothesis (NETH), which asserts that co-3SAT (with n variables and $O(n)$ clauses) cannot be solved by non-deterministic machines running in time $2^{\epsilon\cdot n}$ for some $\epsilon > 0$. This hypothesis is an exponential-time version of $co\mathcal{NP} \not\subseteq \mathcal{NP}$, and is therefore incomparable to rETH and weaker than MAETH.

The motivating observation for our results in this section is that NETH has an unexpected consequence to the long-standing question of whether *worst-case derandomization of $pr\mathcal{BPP}$ is equivalent to circuit lower bounds against $\mathcal{E}$*. Specifically, recall that two-way implications between derandomization and circuit lower bounds have been gradually developing since the early '90s (for surveys see, e.g., [25], [26]), and that it is a long-standing question whether the foregoing implications can be strengthened to show a *complete equivalence* between the two. One well-known implication of such an equivalence would be that any (worst-case) derandomization of $pr\mathcal{BPP}$ *necessitates* the construction of PRGs that "fool" non-uniform circuits.[8] Then, being more concrete, the motivating observation for our results in this section is that NETH *implies an affirmative answer* to the foregoing question (and this is not difficult to show; see Section II-B).

Our main contribution is in showing that, loosely speaking, even a *very weak form of* NETH suffices to answer the question of equivalence in the affirmative, and that this weak form of NETH is in some sense *inherent* (see details below). Specifically, we say that $L \subseteq \{0,1\}^*$ has $\mathcal{NTIME}[T]$-uniform circuits if there exists a non-deterministic machine M that gets input 1^n, runs in time $T(n)$, and satisfies the following: For some non-deterministic choices M outputs a *single circuit* $C\colon \{0,1\}^n \to \{0,1\}$ that *decides L on all inputs* $x \in \{0,1\}^n$, and whenever M does not output such a circuit, it outputs $\perp$. We also quantify the *size* of the output circuit, when this size is smaller than $T(n)$.

The hypotheses that will suffice to show an equivalence between derandomization and circuit lower bounds are of the form "$\mathcal{E}$ does not have $\mathcal{NTIME}[T]$-uniform circuits of size $S(n) \ll T(n)$", for values of T and S that will be specified below. In words, this hypothesis rules out a world in which every $L \in \mathcal{E}$ can be computed by *small circuits* that can be *efficiently produced by a uniform* (non-deterministic) *machine*. Indeed, this hypothesis is weaker than the NETH-style hypothesis $\mathcal{E} \not\subseteq \mathcal{NTIME}[T]$, and even than the hypothesis $\mathcal{E} \not\subseteq (\mathcal{NTIME}[T] \cap \mathcal{SIZE}[T])$. We stress that our hypothesis refers to lower bounds for *uniform* models of computation, for which strong lower bounds (compared to those for non-uniform circuits) are already known. (For example, $\mathcal{NP}$ is hard for $\mathcal{NP}$-uniform circuits of size n^k for every fixed $k \in \mathbb{N}$ (see [31]), whereas we do not even know if $\mathcal{E}^{\mathcal{NP}}$ is hard for non-uniform circuits of arbitrarily large *linear* size.) The fact that such a weak hypothesis suffices to deduce that derandomization and circuit lower bounds are equivalent can be seen as appealing evidence that the equivalence indeed holds.

Our first result is that if $\mathcal{E}$ cannot be decided by $\mathcal{NTIME}[2^{n^\delta}]$-uniform circuits of polynomial size (for some $\delta > 0$), then derandomization of $pr\mathcal{BPP}$ in sub-exponential time is equivalent to lower bounds for polynomial-sized circuits against $\mathcal{EXP}$.

Theorem I.3 (NETH $\Rightarrow$ circuit lower bounds are equivalent to derandomization; "low-end" setting)**.** *Assume that there exists $\delta > 0$ such that $\mathcal{E}$ cannot be decided by $\mathcal{NTIME}[2^{n^\delta}]$-uniform circuits of arbitrary polynomial size, even infinitely-often. Then,*

$$pr\mathcal{BPP} \subseteq \mathtt{i.o.}pr\mathcal{SUBEXP} \iff \mathcal{EXP} \not\subset \mathcal{P}/\mathrm{poly}\,.$$

Theorem I.3 also scales-up to "high-end" parameter settings, albeit not smoothly, and using different proof techniques (see [1, Section 5] for details). Nevertheless, an analogous result holds for the extreme "high-end" setting: Under the stronger hypothesis that $\mathcal{E}$ cannot be decided by $\mathcal{NTIME}[2^{\Omega(n)}]$-uniform circuits, we show that $pr\mathcal{BPP} = pr\mathcal{P}$ is equivalent to lower bounds for exponential-sized

[7] Note that indeed a non-deterministic analogue of rETH is MAETH (or, arguably, AMETH), rather than NETH, due to the use of randomness. Also recall that, while the "strong" version of MAETH is false (see [6]), there is currently no evidence against the "non-strong" version MAETH.

[8] The question of equivalence is mostly "folklore", but was mentioned several times in writing. It was asked in [27, Remark 33], who proved an analogous equivalence between non-deterministic derandomization with short advice and circuit lower bounds against non-deterministic classes (i.e., against $\mathcal{NTIME}$; see also [28]). It was also mentioned as a hypothetical possibility in [20] (referred to there as a "super-Karp-Lipton theorem"). Following the results of [29], the question was recently raised again as a conjecture in [30]. We note that in the context of uniform "hardness-to-randomness", equivalences between average-case derandomization, lower bounds for uniform classes, and PRGs for uniform circuits have long been known (see [15], [23]), but these equivalences do not involve circuit lower bounds or standard PRGs.

circuits against $\mathcal{E}$; that is:

Theorem I.4 ($\mathsf{NETH} \Rightarrow$ circuit lower bounds are equivalent to derandomization; "high-end" setting)**.** *Assume that there exists $\delta > 0$ such that $\mathcal{E}$ cannot be decided by $\mathcal{NTIME}[2^{\delta \cdot n}]$-uniform circuits, even infinitely-often. Then:*

$$pr\mathcal{BPP} = pr\mathcal{P} \iff \exists \epsilon > 0 : \mathcal{DTIME}[2^n] \not\subset \mathtt{i.o}.\mathcal{SIZE}[2^{\epsilon \cdot n}]$$

Remarkably, as mentioned above, hypotheses such as the ones in Theorems I.3 and I.4 actually yield a stronger conclusion, and are also *necessary* for that stronger conclusion. Specifically, the stronger conclusion is that even *non-deterministic derandomization of $pr\mathcal{BPP}$* (such as $pr\mathcal{BPP} \subseteq pr\mathcal{NSUBEXP}$) yields circuit lower bounds against $\mathcal{E}$, which in turn yield PRGs for non-uniform circuits.

Theorem I.5 ($\mathcal{NTIME}$-uniform circuits for $\mathcal{E}$, non-deterministic derandomization, and circuit lower bounds)**.** *Assume that there exists $\delta > 0$ such that $\mathcal{E}$ cannot be decided by $\mathcal{NTIME}[2^{n^\delta}]$-uniform circuits of arbitrary polynomial size. Then,*

$$pr\mathcal{BPP} \subseteq pr\mathcal{NSUBEXP} \Longrightarrow \mathcal{EXP} \not\subset \mathcal{P}/\mathrm{poly}\,. \quad \text{(I.1)}$$

In the other direction, if Eq. (I.1) *holds, then $\mathcal{E}$ cannot be decided by $\mathcal{NP}$-uniform circuits.*

Note that in Theorem I.5 there is a gap between the hypothesis that implies Eq. (I.1) and the conclusion from Eq. (I.1). Specifically, the hypothesis refers to $\mathcal{NTIME}[2^{n^\delta}]$-uniform circuits of polynomial size, whereas the conclusion refers to $\mathcal{NP}$-uniform circuits. By optimizing the parameters, this gap between sub-exponential and polynomial can be considerably narrowed (see [1, Theorem 5.11]).

D. Disproving a version of rETH *requires circuit lower bounds*

Lastly, we show that *disproving* a weak version of rETH requires breakthrough circuit lower bounds. Specifically, we show that a randomized algorithm that solves CircuitSAT in time $2^{n/\mathrm{polylog}(n)}$ would yield lower bounds for circuits of quasilinear size against $\mathcal{BPE} = \mathcal{BPTIME}[2^{O(n)}]$. For context, the best known lower bounds for such circuits are against Σ_2 (see [32]) or against $\mathcal{MA}/1$ (i.e., Merlin-Arthur protocols that use one bit of *non-uniform advice*; see [33]). Specifically, we prove the following:

Theorem I.6 (circuit lower bounds from non-trivial randomized CircuitSAT algorithms)**.** *For any constant $c \in \mathbb{N}$ there exists a constant $c' \in \mathbb{N}$ such that the following holds. If* CircuitSAT *for circuits over n variables and of size $n^2 \cdot (\log n)^{c'}$ can be solved in probabilistic time $2^{n/(\log n)^{c'}}$, then $\mathcal{BPE} \not\subset \mathcal{SIZE}[n \cdot (\log n)^c]$.*

Theorem I.6 constitutes progress on a well-known technical challenge. Specifically, the known arguments that deduce circuit lower bounds from "non-trivial" circuit-analysis algorithms (following Williams [34]) crucially rely on the hypothesis that the circuit-analysis algorithm is *deterministic*, and it is a well-known challenge to obtain analogous results for *randomized* algorithms, as we do in Theorem I.6. In order to prove Theorem I.6 we crucially leverage the technical tools that we develop in the proof of Theorem I.1; see Section II-C for further details and for comparison with known results.

Finally, we combine Theorem I.6 and Theorem I.1 to deduce the following unconditional Karp-Lipton style result: If $\mathcal{BPE}$ can be decided by circuits of quasilinear size, then $\mathcal{BPP}$ can be derandomized, in average-case and infinitely-often, in time $2^{\tilde{O}(\log(n))} = n^{\mathrm{polyloglog}(n)}$. (See [1, Corollary 6.6] for details and for a precise statement.)

II. Technical overview

In this section we describe the proofs of our main results, in high level. In Section II-A we describe the proofs of Theorems I.1 and I.2; in Section II-B we describe the proofs of Theorems I.3, I.4 and I.5; and in Section II-C we describe the proof of Theorem I.6, which relies on the proofs from Section II-A.

A. Near-optimal uniform hardness-to-randomness results for TQBF

Recall that in typical "hardness-to-randomness" results, a PRG is based on a hard function, and the proof amounts to showing that an efficient distinguisher for the PRG can be transformed to an efficient algorithm or circuit that computes the hard function.

In high-level, our proof strategy follows this paradigm, and relies on the classic approach of Impagliazzo and Wigderson [15] for transforming a distinguisher into an algorithm for the hard function. Loosely speaking, the latter approach works only when the hard function $f^{\mathtt{ws}} \colon \{0,1\}^* \to \{0,1\}^*$ is well-structured; the precise meaning of the term "well-structured" differs across different follow-up works, and in the current work it will also take on a new meaning, but for now let us intuitively think of $f^{\mathtt{ws}}$ as downward self-reducible and as having properties akin to random self-reducibility. Instantiating the Nisan-Wigderson PRG with a suitable encoding $\mathtt{ECC}(f^{\mathtt{ws}})$ of $f^{\mathtt{ws}}$ as the underlying function (again, the precise requirements from ECC differ across works), our goal is to show that if the PRG with stretch $t(n)$ does not "fool" uniform distinguishers even infinitely-often, then $f^{\mathtt{ws}}$ is computable in probabilistic time $t'(n) > t(n)$.

The key challenge underlying this approach is the *significant overheads* in the proof, which increase the time complexity t' of computing $f^{\mathtt{ws}}$. In the original proof of [15] this time was roughly $t'(n) \approx t(t(n))$, and the state-of-the-art prior to the current work, by Trevisan and Vadhan [20] (following [16]), yielded $t'(n) = \mathrm{poly}(t(\mathrm{poly}(n)))$. Since the relevant functions $f^{\mathtt{ws}}$ in all works are computable in

$\mathcal{E}$, proofs with such an overhead can yield at most a sub-exponential stretch $t(n) = 2^{n^{\Omega(1)}}$.

As mentioned in Section I-B, previous works bypassed this difficulty by either using stronger hypotheses, or deducing weaker conclusions, or working in different contexts (e.g., considering derandomization of $\mathcal{AM}$ rather than of $\mathcal{BPP}$). In contrast, we tackle this difficulty directly, and manage to reduce *all* of the polynomial overheads in the input length to *polylogarithmic overheads* in the input length. That is, we will show that for carefully-constructed $f^{\mathtt{ws}}$ and suitably-chosen ECC (and with some variations in the proof approach), if the PRG instantiated with $\mathtt{ECC}(f^{\mathtt{ws}})$ for stretch t does not "fool" uniform distinguishers infinitely-often, then $f^{\mathtt{ws}}$ can be computed in time $t'(n) = t(\widetilde{O}(n))^{O(1)}$.

1) The well-structured function $f^{\mathtt{ws}}$: Following Trevisan and Vadhan [20], our $f^{\mathtt{ws}}$ is an artificial $\mathcal{PSPACE}$-complete problem that we carefully construct. Their goal was to construct $f^{\mathtt{ws}}$ that will be simultaneously downward self-reducible and randomly self-reducible. They achieved this by constructing a function based on the proof of $\mathcal{IP} = \mathcal{PSPACE}$ [35], [36]: Loosely speaking, at input length $N = \mathrm{poly}(n)$ the function gets as input a 3-SAT formula φ over n variables, and outputs $P^{(\varphi,N)}(\varphi) = Q_1 \circ Q_2 \circ ... \circ Q_{\mathrm{poly}(n)} P^{(\varphi)}$, where $P^{(\varphi)}$ is an arithmetization of φ, the Q_i's are arithmetic operators from the $\mathcal{IP} = \mathcal{PSPACE}$ proof, and $P^{(\varphi,N)}(\varphi) = \mathtt{TQBF}(\varphi)$; and for $i \in [\mathrm{poly}(n)]$, at input length $N - i$, the function gets input (φ, w) and outputs $P^{(\varphi,N-i)}(\varphi, w)$, where $P^{(\varphi,N-i)}$ is the polynomial that applies one less operator to $P^{(\varphi)}$ than $P^{(\varphi,N-i+1)}$ and fixes some input variables for $P^{(\varphi)}$ according to w. Since $f^{\mathtt{ws}}$ consists of low-degree polynomials, it is randomly self-reducible; and since each $P^{(\varphi,N-i)}$ is obtained by applying a simple operator to $P^{(\varphi,N-(i-1))}$, the function $f^{\mathtt{ws}}$ is also downward self-reducible.

Going through their proof (with needed adaptations for our "high-end" parameter setting), we encounter four different polynomial overheads in the input length. The first and obvious one is that inputs of length n are mapped to inputs of length $N = \mathrm{poly}(n)$, corresponding to the number of rounds in the $\mathcal{IP} = \mathcal{PSPACE}$ protocol. The other polynomial overheads in the input length come from their reduction of TQBF to an intermediate problem that takes both φ and w as part of the input and is still amenable to arithmetization,[9] from the field size that is required for the strong random self-reducibility that is needed in our parameter setting (see below), and from the way the $\mathrm{poly}(n)$ polynomials are combined into a single Boolean function.

The main challenge is to eliminate all of the foregoing overheads *simultaneously*. We will achieve this by presenting a construction of a suitable $f^{\mathtt{ws}}$, which is a refinement of their construction, and constitutes the main technical part in the proof of Theorem I.1. We now outline (very briefly) the key points underlying the construction; for a detailed overview see [1, Section 4.1]. After the following brief outline, we will explain how we use $f^{\mathtt{ws}}$ to prove Theorem I.1.

Our first main idea is to use an $\mathcal{IP} = \mathcal{PSPACE}$ protocol with $\mathrm{polylog}(n)$ rounds instead of $\mathrm{poly}(n)$ rounds, so that the first overhead (i.e., the additive overhead in the input length caused by the number of operators) will be only $\mathrm{polylog}(n)$ instead of $\mathrm{poly}(n)$. Indeed, in such a protocol the verification time in each round is high, and therefore our downward self-reducibility algorithm is relatively slow and makes many queries; but we will be able to afford this in our proof (since eventually we only need to solve TQBF in time $2^{n/\mathrm{polylog}(n)}$). While implementing this idea, we define a different intermediate problem that is both amenable to arithmetization and reducible from TQBF in quasilinear time (see [1, Claim 4.7.1]); we move to an arithmetic setting that will support the strong random self-reducibility that we want (see details below), and arithmetize the intermediate problem in this setting (see [1, Claim 4.7.2]); we show how to execute arithmetic operators in a "batch" in this arithmetic setting (see [1, Claim 4.7.3]); and we combine the resulting collection of polynomials into a single Boolean function. We stress that we are "paying" for all the optimizations above, by the fact that the associated algorithms (for downward self-reducibility and for our notion of random self-reducibility that will be described next) now run in time $2^{n/\mathrm{polylog}(n)}$, rather than polynomial time; but again, we are able to afford this in our proof.

We obtain a function $f^{\mathtt{ws}}$ with the following properties: First, $f^{\mathtt{ws}}$ is computable in linear space; secondly, TQBF is reducible to $f^{\mathtt{ws}}$ in *quasilinear time*; thirdly, $f^{\mathtt{ws}}$ is *downward self-reducible* in time $2^{n/\mathrm{polylog}(n)}$; and lastly, $f^{\mathtt{ws}}$ is sample-aided worst-case to δ-average-case reducible, for $\delta(n) = 2^{-n/\mathrm{polylog}(n)}$. The last property, which is implicit in many works and was recently made explicit by Goldreich and G. Rothblum [37], asserts the following: There exists a uniform algorithm T that gets as input a circuit $C\colon \{0,1\}^n \to \{0,1\}^*$ that agrees with $f_n^{\mathtt{ws}}$ on at least $\delta(n)$ of the inputs, and *labeled examples* $(x, f^{\mathtt{ws}}(x))$ where $x \in \{0,1\}^n$ is uniformly-chosen, runs in time $2^{n/\mathrm{poly}\log(n)}$ and with high probability outputs a circuit $C'\colon \{0,1\}^n \to \{0,1\}^*$ that computes $f_n^{\mathtt{ws}}$ on all inputs (see [1, Definition 4.2]). Our construction of $f^{\mathtt{ws}}$ also satisfies an additional property, which will only be used in the proof of Theorem I.2 (i.e., of the "almost-always" version of the result); we will describe this property in the proof outline for Theorem I.2 below.

2) Instantiating the [15] proof framework with the function $f^{\mathtt{ws}}$: Given this construction of $f^{\mathtt{ws}}$, we now use a variant of the [15] proof framework, as follows. (For simplicity, we show how to "fool" polynomial-time distinguishers that do not use advice.) Let ECC be the Goldreich-Levin [38]

[9] Recall that the standard arithmetization of 3-SAT is a polynomial that depends on the input formula, whereas we want a single polynomial that gets both a formula and the assignment as input.

(i.e., Hadamard) encoding $\mathtt{ECC}(f^{\mathtt{ws}})(x,r) = \oplus_i f^{\mathtt{ws}}(x)_i \cdot r_i$. The argument of [15] (following [12]) shows that if for input length n there exists a uniform $\mathrm{poly}(n)$-time distinguisher A for the Nisan-Wigderson PRG (instantiated with $\mathtt{ECC}(f^{\mathtt{ws}})$) that succeeds with advantage $1/n$, then for input length $\ell = \widetilde{O}(\log(n))$ (corresponding to the set-size in the underlying combinatorial design) there is a *weak learner* for $\mathtt{ECC}(f^{\mathtt{ws}})$: That is, there exists an algorithm that gets oracle access to $\mathtt{ECC}(f^{\mathtt{ws}})$, runs in time $\mathrm{poly}(n) \approx 2^{\ell/\mathrm{polylog}(\ell)}$, and outputs a small circuit that agrees with $\mathtt{ECC}(f^{\mathtt{ws}})$ on approximately $1/2+1/n^2 \approx 1/2+\delta_0(\ell)$ of the ℓ-bit inputs, where $\delta_0(\ell) = 2^{-\ell/\mathrm{polylog}(\ell)}$.

Assuming that there exists a distinguisher for the PRG as above for every $n \in \mathbb{N}$, we deduce that a weak learner exists for every $\ell \in \mathbb{N}$. Following [15], for each input length $i = 1, ..., \ell$ we construct a circuit of size $2^{i/\mathrm{polylog}(i)}$ for $f_i^{\mathtt{ws}}$. Specifically, in iteration i we run the learner for $\mathtt{ECC}(f^{\mathtt{ws}})$ on input length $2i$, and answer its oracle queries using the downward self-reducibility of $f^{\mathtt{ws}}$, the circuit that we have for $f_{i-1}^{\mathtt{ws}}$, and the fact that $\mathtt{ECC}(f^{\mathtt{ws}})_{2i}$ is easily computable given access to $f_i^{\mathtt{ws}}$. The learner outputs a circuit of size $2^{2i/\mathrm{polylog}(2i)}$ that agrees with $\mathtt{ECC}(f^{\mathtt{ws}})$ on approximately $1/2 + \delta_0(2i)$ of the $2i$-bit inputs, and the argument of [38] allows to efficiently transform this circuit to a circuit of similar size that computes $f^{\mathtt{ws}}$ on a approximately $\delta(i) = \mathrm{poly}(\delta_0(2i))$ of the i-bit inputs. Our goal now is to transform this circuit to a circuit of similar size that computes $f^{\mathtt{ws}}$ on all i-bit inputs. Recall that in general, performing such transformations by a *uniform* algorithm is challenging (intuitively, if $f^{\mathtt{ws}}$ is a codeword in an error-correcting code, this corresponds to uniform list-decoding of a "very corrupt" version of $f^{\mathtt{ws}}$). However, in our specific setting we can produce random *labeled samples* for $f^{\mathtt{ws}}$, using its downward self-reducibility and the circuit that we have for $f_{i-1}^{\mathtt{ws}}$. Relying on the *sample-aided worst-case to average-case reducibility* of $f^{\mathtt{ws}}$, we can transform our circuit to a circuit of similar size that computes $f_i^{\mathtt{ws}}$ on all inputs.

Finally, since TQBF is reducible with quasilinear overhead to $f^{\mathtt{ws}}$, if we can compute $f^{\mathtt{ws}}$ in time $2^{n/\mathrm{polylog}(n)}$ then we can compute TQBF in such time. Moreover, since $f^{\mathtt{ws}}$ is computable in space $O(\ell) = \tilde{O}(\log(n))$ (and thus in time $n^{\mathrm{polyloglog}(n)}$), the pseudorandom generator is computable in time $n^{\mathrm{polyloglog}(n)}$.

3) The "almost-always" version: Proof of Theorem I.2: We now explain how to adapt the proof above in order to get an "almost-always" PRG with near-exponential stretch. For starters, we will use a stronger property of $f^{\mathtt{ws}}$, namely that it is downward self-reducible in a polylogarithmic number of steps; this means that for every input length ℓ there exists an input length $\ell_0 \geq \ell - \mathrm{polylog}(\ell)$ such that $f^{\mathtt{ws}}$ is efficiently-computable at input length ℓ_0 (i.e., $f_{\ell_0}^{\mathtt{ws}}$ is computable in time $2^{\ell_0/\mathrm{polylog}(\ell_0)}$ without a "downward" oracle); see [1, Section 4.1.1] for intuition and details about this property.

Now, observe that the transformation of a probabilistic distinguisher A for the PRG to a probabilistic algorithm F that computes $f^{\mathtt{ws}}$ actually gives a "point-wise" guarantee: For every input length $n \in \mathbb{N}$, if A distinguishes the PRG on a corresponding set of input lengths S_n, then F computes $f^{\mathtt{ws}}$ correctly at input length $\ell = \ell(n) = \widetilde{O}(\log(n))$; specifically, we want to use the downward self-reducibility argument for $f^{\mathtt{ws}}$ at input lengths $\ell, \ell-1, ..., \ell_0$, and S_n is the set of input lengths at which we need a distinguisher for G in order to obtain a weak learner for $\mathtt{ECC}(f^{\mathtt{ws}})$ at input lengths $\ell, \ell-1, ...\ell_0$. Moreover, since $f^{\mathtt{ws}}$ is downward self-reducible in $\mathrm{polylog}$ steps, we will only need weak learners at inputs $\ell, ..., \ell_0 = \ell - \mathrm{polylog}(\ell)$; hence, we can show that S_n is a set of $\mathrm{polylog}(\ell) = \mathrm{polyloglog}(n)$ input lengths in the interval $[n, n^2]$ (see [1, Lemma 4.9] for the precise calculation). Taking the contrapositive, if $f^{\mathtt{ws}}$ cannot be computed by F on almost all ℓ's, then for every $n \in \mathbb{N}$ there exists an input length $m \in S_n \subset [n, n^2]$ such that G fools A at input length m.[10]

Our derandomization algorithm gets input 1^n and also gets the "good" input length $m \in S_n$ *as non-uniform advice*; it then simulates $G(1^m)$ (i.e., the PRG at input length m) and truncates the output to n bits. (We can indeed show that truncating the output of our PRG preserves its pseudorandomness in a uniform setting; see [1, Proposition 4.12] for details.) The crucial point is that since $|S_n| = \mathrm{polyloglog}(n)$, the advice length is $O(\mathrm{logloglog}(n))$. Note, however, that for every potential distinguisher A there exists a *different* input length $m \in S_n$ such that G is pseudorandom for A on m. Hence, our derandomization algorithm (or, more accurately, its advice) depends on the distinguisher that it wants to "fool". Thus, for every $L \in \mathcal{BPP}$ and every efficiently-samplable distribution $\mathcal{X}$ of inputs, there exists a corresponding "almost-always" derandomization algorithm $D_{\mathcal{X}}$ (see [1, Proposition 4.12]).

B. $\mathcal{NTIME}$-uniform circuits for $\mathcal{E}$ and an equivalence between derandomization and circuit lower bounds

The proofs that we describe in the current section are significantly simpler technically than the proofs described in Sections II-A and II-C. As mentioned in Section I-C, the motivating observation is that NETH implies an equivalence between derandomization and circuit lower bounds; let us start by proving this statement:

[10] Actually, since $f^{\mathtt{ws}}$ is downward self-reducible in $\mathrm{polylog}$ steps, it can be computed relatively-efficiently on infinitely-many input lengths, and thus cannot be "hard" for almost all ℓ's. However, since TQBF can be reduced to $f^{\mathtt{ws}}$ with quasilinear overhead, if TQBF is "hard" almost-always then for every $\ell(n)$ there exists $\ell' \leq \tilde{O}(\ell(n))$ such that $f^{\mathtt{ws}}$ is "hard" on ℓ', which allows our argument to follow through, with a similar set $\overline{S_n} \subset [n, n^{\mathrm{polyloglog}(n)}]$ (see [1, Proposition 4.11] for details). For simplicity, we ignore this issue in the overview.

Proposition II.1 ("warm-up": a weaker version of Theorem I.3)**.** *Assume that* $\mathcal{EXP} \not\subset \mathtt{i.o.}\mathcal{NSUBEXP}$*. Then,* $prBPP \subseteq prSUBEXP \iff \mathcal{EXP} \not\subset \mathtt{i.o.}\mathcal{P}/\mathrm{poly}$.

Proof: The "$\Longleftarrow$" direction follows (without any assumption) from [24]. For the "$\Longrightarrow$" direction, assume that $prBPP \subseteq prSUBEXP$, and assume towards a contradiction that $\mathcal{EXP} \subset \mathtt{i.o.}\mathcal{P}/\mathrm{poly}$. The latter hypothesis implies (using the Karp-Lipton style result of [24]) that $\mathcal{EXP} \subset \mathtt{i.o.}\mathcal{MA}$. Combining this with the former hypothesis, we deduce that $\mathcal{EXP} \subset \mathtt{i.o.}\mathcal{NSUBEXP}$, a contradiction. ■

Our proofs of Theorems I.3 and I.4 will follow the same logical structure as the proof of Proposition II.1, and our goal will be to relax the hypothesis $\mathcal{EXP} \not\subset \mathtt{i.o.}\mathcal{NSUBEXP}$. We will do so by strengthening the Karp-Lipton style result that uses [24] and asserts that a joint "collapse" hypothesis and derandomization hypothesis implies that $\mathcal{EXP}$ can be decided in small non-deterministic time. We will show two different strengthenings, each referring to a different parameter setting: The first strengthening refers to a "low-end" setting, and asserts that if $\mathcal{EXP} \subset \mathcal{P}/\mathrm{poly}$ and $prBPP \subseteq prSUBEXP$ then $\mathcal{EXP}$ has $\mathcal{NSUBEXP}$-uniform circuits of polynomial size (see [1, Item (1) of Proposition 5.6]); and the second strengthening refers to a "high-end" setting, and asserts that if $\mathcal{E} \subset \mathtt{i.o.}\mathcal{SIZE}[2^{\epsilon \cdot n}]$ and $prBPP = pr\mathcal{P}$ then $\mathcal{E}$ has $\mathcal{NTIME}[2^{O(\epsilon)\cdot n}]$-uniform circuits (see [1, Proposition 5.7]). The proofs of these two different strengthenings rely on different ideas; for high-level descriptions of the proofs see [1, Section 5.1.2] and [1, Section 5.1.3], respectively.

For context, recall that (as noted by Fortnow, Santhanam, and Williams [39]), the proof of [24] already supports the stronger result that $\mathcal{EXP} \subset \mathcal{P}/\mathrm{poly} \iff \mathcal{EXP} = \mathcal{OMA}$;[11] and by adding a derandomization hypothesis (e.g., $prBPP = pr\mathcal{P}$) we can deduce that $\mathcal{EXP} = \mathcal{ONP}$. Nevertheless, our results above are stronger, because $\mathcal{NP}$-uniform circuits are an even weaker model than $\mathcal{ONP}$: This is since in the latter model the proof is verified on an input-by-input basis, whereas in the former model we only verify *once* that the proof is convincing for *all* inputs. We also stress that some lower bounds for this weaker model (i.e., for $\mathcal{NTIME}$-uniform circuits of small size) are already known: Santhanam and Williams [31] proved that for every $k \in \mathbb{N}$ there exists a function in $\mathcal{NP}$ that cannot be computed by $\mathcal{NP}$-uniform circuits of size n^k.

We also note that our proofs actually show that (conditioned on lower bounds for $\mathcal{NTIME}$-uniform circuits against $\mathcal{E}$) even a *relaxed derandomization hypothesis* is already equivalent to the corresponding circuit lower bounds. For example, in the "high-end" setting, to deduce that $\mathcal{E} \not\subset \mathcal{SIZE}[2^{\Omega(n)}]$ it suffices to assume that CAPP on v-bit circuits of size $n = 2^{\Omega(v)}$ can be solved in time $2^{\epsilon \cdot v}$, for a sufficiently small $\epsilon > 0$.[12] For more details, see [1, Section 5.2].

Proof of Theorem I.5: The first part of Theorem I.5 asserts that if $\mathcal{E}$ does not have $\mathcal{NTIME}[2^{n^\delta}]$-uniform circuits of polynomial size, then the conditional statement "$prBPP \subseteq prNSUBEXP \implies \mathcal{EXP} \not\subset \mathcal{P}/\mathrm{poly}$" holds. The proof of this statement again follows the logical structure from the proof of Proposition II.1, and relies on a further strengthening of our "low-end" Karp-Lipton style result such that the result only uses the hypothesis that $prBPP \subseteq prNSUBEXP$ rather than $prBPP \subseteq prSUBEXP$.[13]

The second part of Theorem I.5 asserts that if the conditional statement "$prBPP \subseteq prNSUBEXP \implies \mathcal{EXP} \not\subset \mathcal{P}/\mathrm{poly}$" holds, then $\mathcal{E}$ does not have $\mathcal{NP}$-uniform circuits. We will in fact prove the stronger conclusion that $\mathcal{E} \not\subseteq (\mathcal{NP} \cap \mathcal{P}/\mathrm{poly})$. (Recall that the class of problems decidable by $\mathcal{NP}$-uniform circuits is a subclass of $\mathcal{ONP} \subseteq \mathcal{NP} \cap \mathcal{P}/\mathrm{poly}$.) The proof itself is very simple: Assume towards a contradiction that $\mathcal{E} \subseteq (\mathcal{NP} \cap \mathcal{P}/\mathrm{poly})$; since $\mathcal{BPP} \subseteq \mathcal{EXP}$, it follows that $prBPP \subseteq pr\mathcal{NP}$ (see [1, Proof of Theorem 5.10]); and by the hypothesized conditional statement, we deduce that $\mathcal{EXP} \not\subset \mathcal{P}/\mathrm{poly}$, a contradiction. Indeed, the parameter choices in the foregoing proof are far from tight, and (as mentioned after the statement of Theorem I.5) the quantitative gap between the two parts of Theorem I.5 can be considerably narrowed (see [1, Theorem 5.11]).

C. Circuit lower bounds from randomized CircuitSAT algorithms

Recall that Theorem I.6 asserts that if CircuitSAT for n-bit circuits of size $\tilde{O}(n^2)$ can be solved in probabilistic time $2^{n/(\log n)^c}$, then $\mathcal{BPE} \not\subset \mathcal{SIZE}[n \cdot (\log n)^{c'}]$, where c' depends on c. The relevant context for this result is the known line of works that deduce circuit lower bounds from "non-trivial" circuit-analysis algorithms, following the celebrated result of Williams [34]. The main technical innovation in Theorem I.6 is that our hypothesis is only that there exists a *probabilistic* circuit-analysis algorithm, whereas the aforementioned known results crucially rely on the fact that the circuit-analysis algorithm is *deterministic*. On the other hand, the aforementioned known results yield new circuit lower bounds even if the running time of the algorithm is

[11] The notation $\mathcal{OMA}$ stands for "oblivious" $\mathcal{MA}$. It denotes the class of problems that can be decided by an $\mathcal{MA}$ verifier such that for every input length there is a single "good" proof that convinces the verifier on all inputs in the set (rather than a separate proof for each input); see, e.g., [39], [40].

[12] Note that the problem of solving CAPP for v-bit circuits of size $n = 2^{\Omega(v)}$ can be trivially solved in time $2^{O(v)} = \mathrm{poly}(n)$, and thus unconditionally lies in $pr\mathcal{P} \cap prBPTIME[\tilde{O}(n)]$. The derandomization problem described above simply calls for a *faster* deterministic algorithm for this problem.

[13] Intuitively, in the "low-end" Karp-Lipton result we only need to derandomize probabilistic decisions made by the non-deterministic machine that constructs the circuit, whereas the circuit itself is deterministic; thus, a non-deterministic derandomization hypothesis suffices for this result. See [1, Section 5.1.2] for details.

$2^n/n^{\omega(1)}$,[14] whereas Theorem I.6 only yields new circuit lower bounds if the running time is $2^{n/\mathrm{polylog}(n)}$.

As far as we are aware, Theorem I.6 is the first result that deduces circuit lower bounds from a near-exponential-time probabilistic algorithm for a natural circuit-analysis task. The closest result that we are aware of is by Oliveira and Santhanam [41, Theorem 14], who deduced lower bounds for circuits of size $n^{O(1)}$ against $\mathcal{BPE}$ from non-trivial probabilistic algorithms for *learning with membership queries* (rather than for a circuit-analysis task such as CircuitSAT); as explained next, we build on their techniques in our proof.[15]

Our proof strategy is indeed very different from the proof strategies underlying known results that deduce circuit lower bounds from deterministic circuit-analysis algorithms (e.g., from the "easy-witness" proof strategy [27]–[29], [34], [42], [43], or from proofs that rely on $\mathcal{MA}$ lower bounds [27, Rmk. 26], [30], [33]). In high-level, to prove our result we exploit the connection between *randomized learning algorithms* and *circuit lower bounds*, which was recently discovered by Oliveira and Santhanam [41, Sec. 5] (following [44]–[46]). Loosely speaking, their connection relies on the classical results of [15], and we are able to significantly refine this connection, using our refined version of the [15] argument that was detailed in Section II-A.

Our starting point is the observation that CircuitSAT algorithms yield learning algorithms. Specifically, fix $k \in \mathbb{N}$, and assume (for simplicity) that CircuitSAT for polynomial-sized n-bit circuits can be solved in probabilistic time $2^{n/\mathrm{polylog}(n)}$ for an arbitrarily large polylogarithmic function. We show that in this case, any function that is computable by circuits of size $n\cdot(\log n)^k$ can be learned (approximately) using membership queries in time $2^{n/\mathrm{polylog}(n)}$ (we explain below how to prove this).[16] Now, let $f^{\mathtt{ws}}$ be the well-structured function from Section II-A, and recall that $f^{\mathtt{ws}}$ is computable in linear space, and hard for linear space under quasilinear-time reductions. Then, exactly one of two cases holds:

1) The function $f^{\mathtt{ws}}$ does not have circuits of size $n \cdot (\log n)^k$. In this case a Boolean version of $f^{\mathtt{ws}}$ also does not have circuits of such size, and since this Boolean version is in $\mathcal{SPACE}[O(n)] \subseteq \mathcal{BPE}$, we are done.
2) The function $f^{\mathtt{ws}}$ has circuits of size $n \cdot (\log n)^k$. Hence, $f^{\mathtt{ws}}$ is also learnable (as we concluded above), and so the argument of [15] can be used to show that $f^{\mathtt{ws}}$ is computable by an efficient probabilistic algorithm.[17] Now, by a diagonalization argument, there exists $L^{\mathsf{diag}} \in \Sigma_4[n \cdot (\log n)^{2k}]$ that cannot be computed by circuits of size $n \cdot (\log n)^k$. We show that $L^{\mathsf{diag}} \in \mathcal{BPE}$ by first reducing L^{diag} to $f^{\mathtt{ws}}$ in time $\widetilde{O}(n)$, and then computing $f^{\mathtt{ws}}$ (using the efficient probabilistic algorithm).

Thus, in both cases we showed a function in $\mathcal{BPE} \setminus \mathcal{SIZE}[n \cdot (\log n)^k]$. The crucial point is that in the second case, our new and efficient implementation of the [15] argument (which was described in Section II-A) yields a probabilistic algorithm for $f^{\mathtt{ws}}$ with very little overhead, which allows us to indeed show that $L^{\mathsf{diag}} \in \mathcal{BPE}$. Specifically, our implementation of the argument (with the specific well-structured function $f^{\mathtt{ws}}$) shows that $f^{\mathtt{ws}}$ can be learned in time $T(n) = 2^{n/\mathrm{polylog}(n)}$, then $f^{\mathtt{ws}}$ can be computed in similar time $T'(n) = 2^{n/\mathrm{polylog}(n)}$ (see [1, Corollary 4.10]).

We thus only need to explain how a CircuitSAT algorithm yields a learning algorithm with comparable running time. The idea here is quite simple: Given oracle access to a function $f^{\mathtt{ws}}$, we generate a random sample of $r = \mathrm{poly}(n)$ labeled examples $(x_1, f^{\mathtt{ws}}(x_1)), ..., (x_r, f^{\mathtt{ws}}(x_r))$ for $f^{\mathtt{ws}}$, and we use the CircuitSAT algorithm to construct, bit-by-bit, a circuit of size $n \cdot (\log n)^k$ that agrees with $f^{\mathtt{ws}}$ on the sample. Note that the input for the CircuitSAT algorithm is a circuit of size $\mathrm{poly}(n)$ over only $n' \approx n \cdot (\log n)^{k+1}$ bits (corresponding to the size of the circuit that we wish to construct). Hence, the CircuitSAT algorithm runs in time $2^{n'/\mathrm{polylog}(n')} = 2^{n/\mathrm{polylog}(n)}$. And if the sample size $r = \mathrm{poly}(n)$ is large enough, then with high probability *any* circuit of size $n \cdot (\log n)^k$ that agrees with $f^{\mathtt{ws}}$ on the sample also agrees with $f^{\mathtt{ws}}$ on almost all inputs (i.e., by a union-bound over all circuits of such size).

ACKNOWLEDGMENT

We are grateful to Igor Oliveira for pointing us to the results in [41, Sec. 5], which serve as a basis for the proof of Theorem I.6. We thank Oded Goldreich, who provided feedback throughout the research process and detailed comments on the manuscript, both of which helped improve the work. We also thank Ryan Williams for a helpful discussion, for asking us whether a result as in Theorem I.6 can be proved, and for feedback on the manuscript. Finally, we thank an anonymous reviewer for pointing out a bug in the initial proof of Theorem I.5, which we fixed.

[14] For example, from such an algorithm they deduce the lower bound $\mathcal{NEXP} \not\subseteq \mathcal{P}/\mathrm{poly}$; and from an algorithm that runs in time $2^{n/\mathrm{polylog}(n)}$ as in Theorem I.6, their results yield the lower bound $\mathcal{NP} \not\subset \mathcal{SIZE}[n^k]$ for every fixed $k \in \mathbb{N}$.

[15] Another known result, which was communicated to us by Igor Oliveira, asserts that if CircuitSAT for circuits over n variables and of size $\mathrm{poly}(n)$ can be solved in probabilistic sub-exponential time $2^{n^{o(1)}}$, then $\mathcal{BPTIME}[2^{O(n)}] \not\subset \mathcal{P}/\mathrm{poly}$. This result can be seen as a "high-end" form of our result (i.e., of Theorem I.6), where the latter will use a weaker hypothesis but deduce a weaker conclusion.

[16] That is, there exists a probabilistic algorithm that gets input 1^n and oracle access to f, and with high probability outputs an n-bit circuit of size $n \cdot (\log n)^k$ that agrees with f on almost all inputs.

[17] Actually, our implementation of the [15] argument shows that if the function $\mathtt{ECC}(f^{\mathtt{ws}})$ (where ECC is defined as in Section II-A) can be learned, then the function $f^{\mathtt{ws}}$ can be efficiently computed. For simplicity, we ignore the difference between $f^{\mathtt{ws}}$ and $\mathtt{ECC}(f^{\mathtt{ws}})$ in the current high-level decription.

The work was initiated in the 2018 Complexity Workshop in Oberwolfach; the authors are grateful to the Mathematisches Forschungsinstitut Oberwolfach and to the organizers of the workshop for the productive and lovely work environment. Lijie Chen is supported by NSF CCF-1741615 and a Google Faculty Research Award. Ron Rothblum is supported in part by a Milgrom family grant, by the Israeli Science Foundation (Grant No. 1262/18), and the Technion Hiroshi Fujiwara cyber security research center and Israel cyber directorate. Roei Tell is supported by funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 819702). Eylon Yogev is funded by the ISF grants 484/18, 1789/19, Len Blavatnik and the Blavatnik Foundation, and The Blavatnik Interdisciplinary Cyber Research Center at Tel Aviv University. Part of this work was done while the fourth author was visiting the Simons Institute for the Theory of Computing.

REFERENCES

[1] L. Chen, R. Rothblum, R. Tell, and E. Yogev, "On exponential-time hypotheses, derandomization, and circuit lower bounds," *Electronic Colloquium on Computational Complexity: ECCC*, vol. 26, p. 169, 2019.

[2] R. Impagliazzo and R. Paturi, "On the complexity of k-SAT," *Journal of Computer and System Sciences*, vol. 62, no. 2, pp. 367–375, 2001.

[3] R. Impagliazzo, R. Paturi, and F. Zane, "Which problems have strongly exponential complexity?" *Journal of Computer and System Sciences*, vol. 63, no. 4, pp. 512–530, 2001.

[4] H. Dell, T. Husfeldt, D. Marx, N. Taslaman, and M. Wahlén, "Exponential time complexity of the permanent and the Tutte polynomial," *ACM Transactions on Algorithms*, vol. 10, no. 4, pp. Art. 21, 32, 2014.

[5] M. L. Carmosino, J. Gao, R. Impagliazzo, I. Mihajlin, R. Paturi, and S. Schneider, "Nondeterministic extensions of the strong exponential time hypothesis and consequences for non-reducibility," in *Proc. 7th Conference on Innovations in Theoretical Computer Science (ITCS)*, 2016, pp. 261–270.

[6] R. R. Williams, "Strong ETH breaks with Merlin and Arthur: short non-interactive proofs of batch evaluation," in *Proc. 31st Annual IEEE Conference on Computational Complexity (CCC)*, 2016, vol. 50, pp. Art. No. 2, 17.

[7] G. J. Woeginger, "Exact algorithms for NP-hard problems: a survey," in *Combinatorial optimization—Eureka, you shrink!*, ser. Lecture Notes in Computer Science. Springer, Berlin, 2003, vol. 2570, pp. 185–207.

[8] D. Lokshtanov, D. Marx, and S. Saurabh, "Lower bounds based on the exponential time hypothesis," *Bulletin of the European Association for Theoretical Computer Science (EATCS)*, no. 105, pp. 41–71, 2011.

[9] V. V. Williams, "Hardness of easy problems: basing hardness on popular conjectures such as the Strong Exponential Time Hypothesis," in *Proc. 10th International Symposium on Parameterized and Exact Computation*, 2015, vol. 43, pp. 17–29.

[10] ——, "On some fine-grained questions in algorithms and complexity," 2018, accessed at https://people.csail.mit.edu/virgi/eccentri.pdf, October 17, 2019.

[11] M. L. Carmosino, R. Impagliazzo, and M. Sabin, "Fine-grained derandomization: from problem-centric to resource-centric complexity," in *Proc. 45th International Colloquium on Automata, Languages and Programming (ICALP)*, 2018, pp. Art. No. 27, 16.

[12] N. Nisan and A. Wigderson, "Hardness vs. randomness," *Journal of Computer and System Sciences*, vol. 49, no. 2, pp. 149–167, 1994.

[13] C. Umans, "Pseudo-random generators for all hardnesses," *Journal of Computer and System Sciences*, vol. 67, no. 2, pp. 419–440, 2003.

[14] R. Impagliazzo and A. Wigderson, "$\mathrm{P} = \mathrm{BPP}$ if E requires exponential circuits: derandomizing the XOR lemma," in *Proc. 29th Annual ACM Symposium on Theory of Computing (STOC)*, 1999, pp. 220–229.

[15] ——, "Randomness vs. time: De-randomization under a uniform assumption," in *Proc. 39th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 1998, pp. 734–.

[16] J.-Y. Cai, A. Nerurkar, and D. Sivakumar, "Hardness and hierarchy theorems for probabilistic quasi-polynomial time," in *Proc. 31st Annual ACM Symposium on Theory of Computing (STOC))*, 1999, pp. 726–735.

[17] V. Kabanets, "Easiness assumptions and hardness tests: trading time for zero error," 2001, vol. 63, no. 2, pp. 236–252.

[18] C.-J. Lu, "Derandomizing Arthur-Merlin games under uniform assumptions," *Computational Complexity*, vol. 10, no. 3, pp. 247–259, 2001.

[19] D. Gutfreund, R. Shaltiel, and A. Ta-Shma, "Uniform hardness versus randomness tradeoffs for Arthur-Merlin games," *Computational Complexity*, vol. 12, no. 3-4, pp. 85–130, 2003.

[20] L. Trevisan and S. P. Vadhan, "Pseudorandomness and average-case complexity via uniform reductions," *Computational Complexity*, vol. 16, no. 4, pp. 331–364, 2007.

[21] R. Shaltiel and C. Umans, "Low-end uniform hardness vs. randomness tradeoffs for AM," in *Proc. 39th Annual ACM Symposium on Theory of Computing (STOC)*, 2007, pp. 430–439.

[22] D. Gutfreund and S. Vadhan, "Limitations of hardness vs. randomness under uniform reductions," in *Proc. 12th International Workshop on Randomization and Approximation Techniques in Computer Science (RANDOM)*, 2008, pp. 469–482.

[23] O. Goldreich, "In a world of P=BPP," in *Studies in Complexity and Cryptography. Miscellanea on the Interplay Randomness and Computation*, 2011, pp. 191–232.

[24] L. Babai, L. Fortnow, N. Nisan, and A. Wigderson, "BPP has subexponential time simulations unless EXPTIME has publishable proofs," *Computational Complexity*, vol. 3, no. 4, pp. 307–318, 1993.

[25] I. C. Oliveira, "Algorithms versus circuit lower bounds," *Electronic Colloquium on Computational Complexity: ECCC*, vol. 20, p. 117, 2013.

[26] R. Williams, "Algorithms for circuits and circuits for algorithms: Connecting the tractable and intractable," in *Proc. International Congress of Mathematicians (ICM)*, 2014, pp. 659–682.

[27] R. Impagliazzo, V. Kabanets, and A. Wigderson, "In search of an easy witness: exponential time vs. probabilistic polynomial time," *Journal of Computer and System Sciences*, vol. 65, no. 4, pp. 672–694, 2002.

[28] L. Chen and H. Ren, "Strong average-case circuit lower bounds from non-trivial derandomization," in *Proc. 52th Annual ACM Symposium on Theory of Computing (STOC)*, 2020.

[29] C. Murray and R. 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.

[30] R. Tell, "Proving that $prBPP = prP$ is as hard as proving that "almost NP" is not contained in P/poly," *Information Processing Letters*, vol. 152, p. 105841, 2019.

[31] R. Santhanam and R. Williams, "On medium-uniformity and circuit lower bounds," in *Proc. 28th Annual IEEE Conference on Computational Complexity (CCC)*, 2013, pp. 15–23.

[32] R. Kannan, "Circuit-size lower bounds and non-reducibility to sparse sets," *Information and Control*, vol. 55, no. 1-3, pp. 40–56, 1982.

[33] R. Santhanam, "Circuit lower bounds for Merlin-Arthur classes," *SIAM Journal of Computing*, vol. 39, no. 3, pp. 1038–1061, 2009.

[34] R. Williams, "Improving exhaustive search implies superpolynomial lower bounds," *SIAM Journal of Computing*, vol. 42, no. 3, pp. 1218–1244, 2013.

[35] C. Lund, L. Fortnow, H. Karloff, and N. Nisan, "Algebraic methods for interactive proof systems," *Journal of the Association for Computing Machinery*, vol. 39, no. 4, pp. 859–868, 1992.

[36] A. Shamir, "IP = PSPACE," *Journal of the ACM*, vol. 39, no. 4, pp. 869–877, 1992.

[37] O. Goldreich and G. N. Rothblum, "Worst-case to average-case reductions for subclasses of P," *Electronic Colloquium on Computational Complexity: ECCC*, vol. 26, p. 130, 2017.

[38] O. Goldreich and L. A. Levin, "A hard-core predicate for all one-way functions," in *Proc. 21st Annual ACM Symposium on Theory of Computing (STOC)*, 1989, pp. 25–32.

[39] L. Fortnow, R. Santhanam, and R. Williams, "Fixed-polynomial size circuit bounds," in *Proc. 24th Annual IEEE Conference on Computational Complexity (CCC)*, 2009, pp. 19–26.

[40] O. Goldreich and O. Meir, "Input-oblivious proof systems and a uniform complexity perspective on P/poly," *ACM Transactions on Computation Theory*, vol. 7, no. 4, pp. Art. 16, 13, 2015.

[41] I. C. Oliveira and R. Santhanam, "Conspiracies between learning algorithms, circuit lower bounds, and pseudorandomness," in *Proc. 32nd Annual IEEE Conference on Computational Complexity (CCC)*, 2017, vol. 79, pp. Art. No. 18, 49.

[42] L. Chen and R. R. Williams, "Stronger Connections Between Circuit Analysis and Circuit Lower Bounds, via PCPs of Proximity," in *Proc. 34th Annual IEEE Conference on Computational Complexity (CCC)*, 2019, pp. 19:1–19:43.

[43] L. Chen, "Non-deterministic quasi-polynomial time is average-case hard for ACC circuits," in *Proc. 60th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2019.

[44] L. Fortnow and A. R. Klivans, "Efficient learning algorithms yield circuit lower bounds," *Journal of Computer and System Sciences*, vol. 75, no. 1, pp. 27–36, 2009.

[45] R. C. Harkins and J. M. Hitchcock, "Exact learning algorithms, betting games, and circuit lower bounds," *ACM Transactions on Computation Theory*, vol. 5, no. 4, pp. Art. 18, 11, 2013.

[46] A. Klivans, P. Kothari, and I. Oliveira, "Constructing hard functions using learning algorithms," in *Proc. 28th Annual IEEE Conference on Computational Complexity (CCC)*, 2013, pp. 86–97.