Unconditional Lower Bounds Against Advice

Harry Buhrman* Lance Fortnow† Rahul Santhanam‡

Abstract

We show several unconditional lower bounds for exponential time classes against polynomial time classes with advice, including:

1. For any constant c, $\mathsf{NEXP} \not\subseteq \mathsf{P}^{\mathsf{NP}[n^c]}/n^c$
2. For any constant c, $\mathsf{MAEXP} \not\subseteq \mathsf{MA}/n^c$
3. $\mathsf{BPEXP} \not\subseteq \mathsf{BPP}/n^{o(1)}$

It was previously unknown even whether $\mathsf{NEXP} \subseteq \mathsf{NP}/n^{0.01}$. For the probabilistic classes, no lower bounds for uniform exponential time against advice were known before.

We also consider the question of whether these lower bounds can be made to work on almost all input lengths rather than on infinitely many. We give an oracle relative to which $\mathsf{NEXP} \subseteq \mathsf{i.o.NP}$, which provides evidence that this is not possible with current techniques.

1 Introduction

Lower bounds are the holy grail of complexity theory. Showing $\mathsf{P} \neq \mathsf{NP}$ or separating BPP from NEXP requires one to establish unconditional super polynomial lower bounds, which are currently beyond our abilities. Most techniques we currently have run into "obstacles" such as relativization [BGS75], natural proofs [RR97] and algebrization [AW08]. Consequently research has focused mainly on conditional results, such as most work in derandomization and PCPs.

In this paper, we show unconditional lower bounds for exponential-time classes such as NEXP, MAEXP, BPEXP and REXP against their polynomial-time versions with advice. We describe our results in more detail in the next subsection.

Lower bounds against advice are closely tied to derandomization. By showing strong enough lower bounds, we hope to get new and interesting derandomization results. Indeed, we illustrate this by showing that our results imply a new separation of a probabilistic class from NEXP.

Unconditional separations are valuable not just in and of themselves, but as components of more involved arguments. A number of major results in complexity theory in recent years have required a hierarchy theorem or unconditional separation to finish the proof [AG94, FLvMV05, IKW02, KI03]. By giving stronger versions of such separations, we hope to derive tighter results. We illustrate this in our paper by deriving an improved result of the form that derandomization implies circuit lower bounds using the machinery of Impagliazzo, Kabanets and Wigderson [IKW02].

Our results for probabilistic classes have an added significance. Probabilistic classes are instances of semantic classes–classes for which there is no recursive enumeration of machines defining languages in the classes and may not have complete sets. We don't understand semantic classes very well, indeed we do not have a strict time hierarchy theorem for any semantic class that is not known to be equal to a syntactic class. Much recent work [FS04, FST05, vMP06] has been focused on showing hierarchies for semantic classes with advice. However, none of the separation results in this line of work hold for uniform semantic classes. By

*CWI, Amsterdam. Email: buhrman@cwi.nl

†Northwestern University. Email: fortnow@eecs.northwestern.edu

‡University of Edinburgh. Email: rsanthan@inf.ed.ac.uk

showing separation results for uniform semantic classes here, we extend our understanding of them and are led to pose further questions which might lead to more progress in this area.

How do our techniques face up to the lower bound obstacles mentioned earlier? The natural proofs obstacle does not trouble us because our proofs ultimately rely on diagonalization, which does not naturalize. Diagonalization does relativize, however our proofs of the lower bounds for BPEXP and MAEXP use an indirect diagonalization technique which takes advantage of the non-relativizing PCP machinery [BFL91]. Thus our proofs of these results do not relativize. We suspect that all our proofs do algebrize in the sense of Aaronson and Wigderson [AW08]. This is not necessarily cause for pessimism since the Aaronson-Wigderson results do not rule out the possibility of using non-relativizing results in more than one place in a proof to derive a stronger lower bound, perhaps even one as strong as $\mathsf{NEXP} \not\subseteq \mathsf{SIZE}(poly)$.

1.1 Results

It is straightforward to show that $\mathsf{NEXP} \neq \mathsf{NP}$, either by using the non-deterministic hierarchy theorem or even more simply by a translation argument. This translation argument can be pushed to give $\mathsf{NEXP} \not\subseteq \mathsf{NP}/n^{o(1)}$. However, this is the limit of this translation argument, and it was unknown whether $\mathsf{NEXP} \not\subseteq \mathsf{NP}/n^{\alpha}$ for any fixed constant $\alpha > 0$. We settle this, and in fact prove a much stronger result.

Theorem. For any constant c, $\mathsf{NEXP} \not\subseteq \mathsf{P}^{\mathsf{NP}[n^c]}/n^c$.

Our proof relies on a combination of ideas, including careful use of the facts that NE has a complete set with respect to linear-time reductions and that SAT is NP-complete, together with translation, a census trick and diagonalization.

As a consequence of our lower bound for NEXP, we get that for each c, the class $\mathsf{BPTIME}(n^c)^{\mathsf{NP}}$ is different from NEXP–a previously unknown separation. We also get that even a mild derandomization of MA in NP with n^c bits of advice for some fixed c, or even MA in $\mathsf{P}^{\mathsf{NP}[n^c]}/n^c$, implies super-polynomial circuit lower bounds for NEXP.

For probabilistic classes, translation arguments with advice are not known to work. Furthermore, these classes are not known to have complete languages either. Thus we are forced to use very different techniques to prove analogous results for these classes.

We use indirect arguments which take advantage of advice elimination for carefully chosen complexity classes D. An advice elimination result for D with respect to a class C says that if D is solvable in C with a certain amount of advice, then D is in C uniformly. Among our contributions is a new advice elimination result with respect to MA. Our techniques allow us to derive the following separations.

Theorem. For any constant c, $\mathsf{MAEXP} \not\subseteq \mathsf{MA}/n^c$

Theorem. $\mathsf{BPEXP} \not\subseteq \mathsf{BPP}/n^{o(1)}$

The proofs of the two results above do not relativize. We also get a relativizing separation for REXP which is somewhat weaker in terms of the advice lower bound.

Finally, we consider the question of whether our lower bounds can be strengthened by making them hold on almost all input lengths rather than on infinitely many. We give some evidence that this is hard with current techniques by constructing an oracle with respect to which NEXP is infinitely often in NP, even without advice. Note that all known techniques for separating non-deterministic time classes relativize.

Theorem. There is an oracle respect to which $\mathsf{NEXP} \subseteq \mathsf{i.o.NP}$.

The rest of the paper is organized as follows. We give definitions and preliminaries in Section 2, our lower bounds for NEXP and their consequences in Section 3, our lower bounds for probabilistic classes in Section 4, and our oracle results in Section 5. We state a few open questions in Section 6.

2 Preliminaries

2.1 Complexity Classes, Promise Problems and Advice

We assume a basic familiarity with complexity classes such as P, RP, BPP, NP, MA, AM, and their exponential-time versions. The Complexity Zoo (which can be found at `http://qwiki.caltech.edu/wiki/ComplexityZoo`) is an excellent resource for basic definitions and statements of results.

The class $\mathsf{P}^{\mathsf{NP}[q(n)]}$ is the class of languages accepted by polynomial-time oracle machines making at most $q(n)$ queries to an NP oracle on any input of length n.

Given a complexity class C, coC is the class of languages L such that $\bar{L} \in \mathsf{C}$. Given a function $s : \mathbb{N} \to \mathbb{N}$, $\mathsf{SIZE}(s)$ is the class of Boolean functions $f = \{f_n\}$ such that for each n, f_n has Boolean circuits of size $O(s(n))$. Given a language L and an integer n, $L_n = L \cap \{0,1\}^n$. Given a class C, i.o.C is the class of languages L for which there is a language $L' \in \mathsf{C}$ such that $L_n = L'_n$ for infinitely many length n.

In order to deal with promise classes in a general way, we take as fundamental the notion of a complexity measure. A complexity measure CTIME is a mapping which assigns to each pair (M, x), where M is a time-bounded machine (here a time function $t_M(x)$ is implicit) and x an input, one of three values "0" (accept), "1" (reject) and "?" (failure of CTIME promise). We distinguish between *syntactic* and *semantic* complexity measures. Syntactic measures have as their range $\{0,1\}$ while semantic measures may map some machine-input pairs to "?". The complexity measures DTIME and NTIME are syntactic (each halting deterministic or non-deterministic machine either accepts or rejects on each input), while complexity measures such as BPTIME and MATIME are semantic (a probabilistic machine may accept on an input with probability 1/2, thus failing the bounded-error promise). For syntactic measures, any halting machine defines a language, while for semantic measures, only a subset of halting machines define languages.

A promise problem is a pair (Y, N), where $Y, N \subseteq \{0,1\}^*$ and $Y \cap N = \varnothing$. We say that a promise problem (Y, N) belongs to a class $\mathsf{CTIME}(t)$ if there is a machine M halting in time t on all inputs of length n such that M fulfils the CTIME promise on inputs in $Y \cup N$, accepting on inputs in Y and rejecting on inputs in N.

A language L is in $\mathsf{CTIME}(t)/a$ if there is a machine M halting in time $t(\cdot)$ taking an auxiliary *advice* string of length $a(\cdot)$ such that for each n, there is some advice string $b_n, |b_n| = a(n)$ such that M fulfils the CTIME promise for each input x with advice string b_n and accepts x iff $x \in L$.

For syntactic classes, a lower bound with advice or for the promise version of the class translates to a lower bound for the class itself.

3 Lower Bounds for NEXP

It can be shown that NEXP $\neq$ NP by using the non-deterministic time hierarchy theorem [Coo72, SFM78, Ž83], or even by a simpler translation argument. In fact, a modification of the translation argument also gives a separation against $n^{o(1)}$ bits of advice.

Proposition 1. $\mathsf{NEXP} \not\subseteq \mathsf{NP}/n^{o(1)}$.

Until now, it's been open whether the lower bound in terms of advice can be pushed to linear or higher. The methods used in the translation argument or in the proof of the non-deterministic hierarchy theorem do not give this. Here, we prove the lower bound by an application of the fact that NE has complete sets with respect to linear-time reductions.

Our method yields somewhat more general results. We state the simpler result with proof first, and then show how to generalize it.

We first need the following lemma (a slightly stronger version of a result in [SS95]) about lower bounds for deterministic exponential time against advice. The proof we give is folklore.

Lemma 2. *For any constant d,* $\mathsf{EXP} \not\subseteq \mathsf{i.o.DTIME}(2^{n^d})/n^d$.

Proof. The proof is by diagonalization. We define a diagonalizing language L which is not in $\text{i.o.}\mathsf{DTIME}(2^{n^d})/n^d$ by defining a machine M which runs in exponential time and decides L.

M operates as follows on input x of length n. It enumerates advice taking machines $M_1, M_2 \ldots M_{\log(n)}$ each running in time at most 2^{n^d} and taking advice of length n^d. It then enumerates all $\log(n)2^{n^d}$ truth tables computed by these machines when every possible string of length n^d is given as advice. It then computes a truth table of an n-bit function f which is distinct from all the truth tables enumerated so far–this can be done in exponential time by a simple pruning strategy. Finally it outputs $f(x)$.

□

Now we are ready to state and prove our lower bound for NEXP.

Theorem 3. *For any constant c, $\mathsf{NEXP} \not\subseteq \mathsf{NP}/n^c$.*

Proof. We will show that either $\mathsf{NEXP} \not\subseteq \mathsf{NP}/poly$ or $\mathsf{NEXP} \not\subseteq \mathsf{NE}/n^c$. From this, the result follows.

Assume, to the contrary, that both these inclusions hold, i.e., $\mathsf{NEXP} \subseteq \mathsf{NP}/poly$ and $\mathsf{NEXP} \subseteq \mathsf{NE}/n^c$. We will derive a contradiction. Let L be a complete language for NE with respect to linear-time reductions. Since $\mathsf{NEXP} \subseteq \mathsf{NP}/poly$, we get that $L \in \mathsf{NTIME}(n^k)/n^k$ for some constant k. Since L is complete for NE with respect to linear-time reductions, we get that $\mathsf{NE} \subseteq \mathsf{NTIME}(n^k)/O(n^k)$.

By translation, we get that $\mathsf{NE}/n^c \subseteq \mathsf{NTIME}(n^{kc})/O(n^{kc})$. To see this, let L' be a language in NE/n^c, and let M' be an advice-taking NE machine accepting L' with advice length n^c. Define a language $L'' \in \mathsf{NE}$ as follows: a string $< x, a >$ is in L'' iff M' accepts x with advice a. Since M' is an NE machine, it follows that $L'' \in \mathsf{NE}$. Thus, by assumption $L'' \in \mathsf{NTIME}(m^k)/O(m^k)$, where m is the input length for L''. Let M'' be an advice-taking machine solving L'' using resources as stated. Now we can solve L' in $\mathsf{NTIME}(n^{kc})/O(n^{kc})$ as follows. The advice-taking machine M we construct for solving L' interprets its advice as consisting of two parts: the first part is an advice string a of length n^c, where n is the input size, and the second part is an advice string b of length $O((n+n^c)^k) = O(n^{kc})$. M simulates M'' on input $< x, a >$ with advice string b, where x is the input for L'. M accepts iff M'' accepts. M operates within time $O(n^{kc})$ (since it simulates an $O(n^k)$ time machine on an input of length $O(n^c)$), uses advice of length $O(n^{kc})$, and decides L' correctly, by definition of L'' and the assumption on M''.

Thus, we have $\mathsf{NEXP} \subseteq \mathsf{NE}/n^c$ and $\mathsf{NE}/n^c \subseteq \mathsf{NTIME}(n^{kc})/O(n^{kc})$, which together imply $\mathsf{NEXP} \subseteq \mathsf{NTIME}(n^{kc})/O(n^{kc})$. But since $\mathsf{EXP} \subseteq \mathsf{NEXP}$ and $\mathsf{NTIME}(n^{kc})/O(n^{kc}) \subseteq \mathsf{DTIME}(2^{n^{kc}})/O(n^{kc})$, we get that $\mathsf{EXP} \subseteq \mathsf{DTIME}(2^{n^{kc}})/O(n^{kc})$, which is a contradiction to Lemma 2. □

Note that Theorem 3 is nearly optimal both with respect to the advice, and with respect to the class for which we show a separation, modulo our inability to prove superpolynomial circuit lower bounds for NEXP. If the advice allowed could be increased to an arbitrary polynomial, we would obtain non-trivial derandomizations of MA and AM, which is a long-standing open problem. In terms of proving a separation for a weaker class, if we could separate say NE from NP/n^c for all constants $c > 0$, this would also imply a superpolynomial circuit lower bound for NEXP.

We generalize Theorem 3 in two ways. First we observe that an analogous result holds for any syntactic complexity measure CTIME which is stronger than deterministic time and can be simulated by deterministic exponential time.

Theorem 4. *Let* CTIME *be any syntactic complexity measure such that for any constructible t, $\mathsf{DTIME}(t) \subseteq \mathsf{CTIME}(t) \subseteq \mathsf{DTIME}(2^{O(t)})$. Then, for any constant c, $\mathsf{CEXP} \not\subseteq \mathsf{CP}/n^c$.*

Proof. The proof goes through exactly like the proof of Theorem 3, since CE has a linear-time complete set, and the same kind of translation argument can be applied as CTIME is syntactic. □

Theorem 4 can also be extended to a separation for promise classes satisfying very general properties.

For the specific case of non-deterministic time, the lower bound holds not just against NP with advice, but against $P^{\mathsf{NP}[n^c]}$ with advice, where $\mathsf{P}^{\mathsf{NP}[n^c]}$ is the class of languages accepted by polynomial-time oracle Turing machines making at most n^c queries to an NP oracle. Here c is a fixed constant.

To derive this extension, we'll need to use the following lemma.

Lemma 5. *For any* $c \geqslant 1$, $\mathsf{E}^{SAT[O(n^c)]} \subseteq \mathsf{NTIME}(2^{O(n^c)})/O(n^c)$

Proof. Let L be any language in $\mathsf{E}^{SAT[O(n^c)]}$, where $c \geqslant 1$. Let M be an exponential linear-time oracle machine making $O(n^c)$ queries to SAT on inputs of length n and deciding L. We define an advice-taking non-deterministic machine M' which operates in time $O(2^{n^c})$ and decides L using at most $O(n^c)$ bits of advice.

On an input y of length n, M' first generates the query tree for M for every possible input x of length n. By a query tree for x, we mean the tree whose nodes are queries with the first query at the root, and for each query q, the children of that query are the queries asked depending on whether query q is answered 0 or 1. For each x of length n, the query tree for x has size at most $2^{O(n^c)}$. Since $c \geqslant 1$, the total number of queries asked in any query tree for an input of length n is $2^{O(n^c)}$. The advice for M' specifies how many of these queries are in SAT–this is a number describable in $O(n^c)$ bits. In a computation path of M', for each query q in a query tree for some x of length n in turn, M' guesses a witness for q and verifies that $q \in SAT$. If the number of correct verifications is not the number coded in the advice string, M' rejects. Otherwise, M' stores a list of all queries that have been verified to be in SAT on the current computation path, and then runs M on y. If M makes a query, M' takes the answer to be yes if the query is in its list, otherwise it takes the answer to be no. Finally, M' accepts iff M accepts.

Assuming the advice is correct, M' will have an accepting computation on y iff the oracle machine M accepts y. This accepting computation will correspond to forming the correct list of queries in SAT among all queries asked in query trees of length n inputs. It can be verified that M' runs in time $2^{O(n^c)}$. □

We now use this fact to generalize Theorem 3. The following result also strengthens a result of Mocas [Moc96] separating NEXP from $\mathsf{P}^{\mathsf{NP}[n^c]}$ for fixed c.

Theorem 6. *For any* c, $\mathsf{NEXP} \not\subseteq \mathsf{P}^{\mathsf{NP}[n^c]}/n^c$.

Proof. On the contrary, assume $\mathsf{NEXP} \subseteq \mathsf{P}^{\mathsf{NP}[n^c]}/n^c$. We derive a contradiction. Since SAT is complete for NP under m-reductions, we get that $\mathsf{NEXP} \subseteq \mathsf{P}^{SAT[n^c]}/n^c$. Now let L be a language complete for NE under linear-time reductions. By assumption on NEXP, there is a constant k such that $L \in \mathsf{DTIME}(n^k)^{SAT[n^c]}/n^c$. Since L is complete for NE under linear-time reductions, it follows that $\mathsf{NE} \subseteq \mathsf{DTIME}(n^k)^{SAT[O(n^c)]}/O(n^c)$.

Now we use the assumption on NEXP again, in a different way. Since $\mathsf{NEXP} \subseteq \mathsf{P}^{\mathsf{NP}[n^c]}/n^c$, we get that $\mathsf{NEXP} \subseteq \mathsf{P}^{SAT[n^c]}/n^c$, by NP-completeness of SAT. Hence $\mathsf{NEXP} \subseteq \mathsf{E}^{SAT[n^c]}/n^c$, which implies $\mathsf{NEXP} \subseteq \mathsf{NTIME}(2^{O(n^{c^2})})/O(n^{c^2})$, using Lemma 5 and a translation argument exactly as in the proof of Theorem 3. From the conclusion in the previous paragraph, and using a translation argument again, we get that $\mathsf{NEXP} \subseteq \mathsf{DTIME}(n^{kc^2})^{SAT[O(n^{c^3})]}/O(n^{c^3})$. But the latter class is in $\mathsf{DTIME}(2^{n^{kc^2+1}})/O(n^{c^3})$, just by answering all SAT queries by exhaustive search, since the length of any SAT query asked by a machine running in time $O(n^{kc^2})$ is $O(n^{kc^2})$. Thus we get $\mathsf{EXP} \subseteq \mathsf{DTIME}(2^{n^{kc^2+1}})/O(n^{c^3})$, which is a contradiction to Lemma 2. □

Theorems 3 and 6 have some interesting consequences for the connection between circuit lower bounds and derandomization. Impagliazzo, Kabanets and Wigderson [IKW02] showed that if MA is in $\mathsf{NSUBEXP}$, then there is a language in NEXP that is not computable with polynomial-size circuits. However, their result does not say anything about a derandomization of MA where the simulating algorithm uses advice.

We will need to use the main result of [IKW02].

Theorem 7. *[IKW02] if* $\mathsf{NEXP} \subseteq \mathsf{SIZE}(poly)$, *then* $\mathsf{NEXP} = \mathsf{MA}$.

Theorem 8. *For any constant* c, *if* $\mathsf{MA} \subseteq \mathsf{NP}/n^c$, *then* $\mathsf{NEXP} \not\subseteq \mathsf{SIZE}(poly)$.

Proof. If $\mathsf{MA} \subseteq \mathsf{NP}/n^c$, then $\mathsf{NEXP} \not\subseteq \mathsf{MA}$, since otherwise we have a contradiction to Theorem 3. But this implies $\mathsf{NEXP} \not\subseteq \mathsf{SIZE}(poly)$ by Theorem 7, and thus the result follows. □

In fact a circuit lower bound follows even from a simulation of MA in $\mathsf{P}^{\mathsf{NP}[n^c]}/n^c$, by using Theorem 6 rather than Theorem 3.

It is known that $\mathsf{MA} \subseteq \mathsf{NP}/poly$, since the randomness of a Merlin-Arthur machine can be simulated by a polynomial-size advice string. Theorem 8 shows that this simulation is essentially optimal with respect to advice–if we could simulate MA in NP with *fixed* polynomial advice, that would imply a long sought-after circuit lower bound for NEXP.

Theorem 6 gives a separation for NEXP and against fixed polynomial advice, but does it imply anything new for separating uniform classes? The answer is yes: consider the class $\mathsf{BPTIME}(n^c)^{\mathsf{NP}}$ of languages accepted by probabilistic oracle machines running in time n^c for some fixed c, and with access to an NP oracle. This class lies between NP and $\mathsf{BPP}^{\mathsf{NP}}$, and seems incomparable to BPP. It is contained in NEXP, but it's unclear if the containment is strict. This is because the NP oracle could have arbitrarily high non-deterministic polynomial time complexity.

Theorem 9. *For any c, $\mathsf{NEXP} \not\subseteq \mathsf{BPTIME}(n^c)^{\mathsf{NP}}$.*

Proof. $\mathsf{BPTIME}(n^c)^{\mathsf{NP}} \subseteq \mathsf{DTIME}(n^{c+1})^{\mathsf{NP}}/O(n^{c+1})$. This follows simply by amplifying the acceptance probability of the probabilistic oracle machine so that it is at most 2^{-n} or at least $1 - 2^{-n}$, and then using Adleman's trick to encode the randomness in the advice. Note that this transformation relativizes. By Theorem 6, we have that $\mathsf{NEXP} \not\subseteq \mathsf{DTIME}(n^{c+1})^{\mathsf{NP}}/O(n^{c+1})$, which yields the desired separation. □

4 Lower Bounds for Probabilistic Exponential Time Classes

Here we prove results analogous to those in the previous section for probabilistic classes. Unlike in the case of non-deterministic time, it is not easy to show even that BPEXP is not in $\mathsf{BPP}/1$. It can be shown using a translation argument that BPEXP is not in BPP [KV87], but this translation argument does not extend to showing a lower bound against advice. Since BPP and BPEXP are semantic classes, it is unknown whether for instance $\mathsf{BPEXP} \subseteq \mathsf{BPP}/1$ implies $\mathsf{BPEXP}/1 \subseteq \mathsf{BPP}/2$.

Recently, there's been a lot of work on hierarchies for semantic classes with advice, and one might ask whether similar techniques are applicable here. The generic methods used in [FST04, vMP06] fail to work here for two reasons. First, their upper bounds always require advice. Second, those methods are inherently incapable of accommodating more than $\log(n)$ bits of advice in the lower bound.

We circumvent both these difficulties by using specific properties of the semantic measures for which we prove bounds. Our general strategy is as follows. Let's say we're trying to show $\mathsf{CEXP} \not\subseteq \mathsf{CP}/a(n)$, where CEXP and CP are the exponential-time and polynomial-time versions respectively of a semantic measure CTIME, and $a(n) \geqslant \log(n)$ is an advice bound. We choose a syntactic class D such that $\mathsf{D} \subseteq \mathsf{CEXP}$. We then argue that if $\mathsf{D} \subseteq \mathsf{CP}/a(n)$, then the advice can be eliminated to place D uniformly in CTIME with some sub-exponential time bound. Now there are two cases: if $\mathsf{D} \not\subseteq \mathsf{CP}/a(n)$, we're done since $\mathsf{D} \subseteq \mathsf{CEXP}$. In case $\mathsf{D} \subseteq \mathsf{CP}/a(n)$, we can use the advice elimination together with a translation argument to diagonalize in CEXP against $\mathsf{CP}/a(n)$.

Our proofs vary depending on the class D, which needs to be chosen judiciously, and the specific form of the advice elimination, which also dictates how much advice we can accommodate in our lower bound.

We now show how this general strategy works for the classes $\mathsf{BPEXP}, \mathsf{MAEXP}$ and REXP.

4.1 Lower Bound for BPEXP

Before we discuss the advice elimination strategy for BPEXP, we need a definition.

Definition 10. *A language L is said to have instance-checkers if there is a probabilistic polynomial-time oracle machine M outputting 1, 0 or ? such that:*

1. *If M is given L as oracle, then $M(x) = L(x)$ with probability 1*

2. *For any oracle A, when M is given A as oracle, for any input x, either $M(x) = L(x)$ or $M(x) =?$ with probability at least $1 - 2^{-\Omega(n)}$.*

It follows from the work of Babai, Fortnow and Lund [BFL91] on multi-prover interactive protocols that EXP-complete languages have instance checkers.

Theorem 11. *[BFL91] All* EXP*-complete languages have instance-checkers.*

We will use Theorem 11 to eliminate the advice from a probabilistic polynomial-time machine accepting an EXP-complete language. We use Theorem 11 in the same way as Trevisan and Vadhan [TV02], but our choice of parameters is different.

Lemma 12. *If* $\mathsf{EXP} \subseteq \mathsf{BPP}/n^{o(1)}$*, then* $\mathsf{EXP} \subseteq \mathsf{BPSUBEXP}$*.*

Proof. Let L be an EXP-complete language. By Theorem 11, L has instance-checkers. Assume $L \in \mathsf{BPP}/n^{o(1)}$, and let M' be a probabilistic polynomial-time machine deciding L with $n^{o(1)}$ bits of advice and error bound $\leqslant 2^{-\omega(n)}$. Let M be an oracle machine witnessing the fact that L is instance-checkable. We define a probabilistic sub-exponential time machine N deciding L. On input x, N simulates the oracle machine M. Let n^k be a bound on the size of queries asked by M on inputs of length n. N uses as oracle O for M each possible sequence of advice strings $a_1, a_2 \ldots a_{n^k}$ for M' in turn, where for each i, $|a_i| \leqslant i^{o(1)}$. Namely, when M asks a query of size i to the oracle, N simulates M' with advice a_i to answer the query. When the simulation of the oracle machine M is finished for a certain oracle O, N returns an answer if the answer is either 0 or 1, otherwise it moves on to the next possible sequence of advice strings. If M still doesn't give a 0 or 1 answer after all possible sequences of advice strings have been used as oracles, N outputs an arbitrary value, say 0.

Since there are at most sub-exponentially many sequences of advice strings and each simulation of M with a sequence of advice strings takes polynomial time, N halts in sub-exponential time, i.e. time $2^{n^{o(1)}}$. We need to argue that N decides L with low error. For this, we simply use the properties of the instance-checker M. When the wrong sequence of advice strings is used as oracle, the probability that M outputs a 0 or 1 answer wrongly is $2^{-\Omega(n)}$, hence by a union bound, the probability that M outputs a 0 or 1 answer wrongly on *some* sequence of advice strings is also $2^{-\Omega(n)}$. When the right sequence of advice strings is used, M outputs the correct answer with probability at least $1 - 2^{-\Omega(n)}$, since the error bound of M' is at most $2^{-\Omega(n)}$. Thus the right answer is always output with probability at least $1 - 2^{-\Omega(n)}$.

Thus we get $L \in \mathsf{BPSUBEXP}$, and since L is complete for EXP under polynomial-time reductions, we also get $\mathsf{EXP} \subseteq \mathsf{BPSUBEXP}$. □

Now we are ready to prove our separation.

Theorem 13. $\mathsf{BPEXP} \not\subseteq \mathsf{BPP}/n^{o(1)}$*.*

Proof. We consider two cases: either $\mathsf{EXP} \subseteq \mathsf{BPP}/n^{o(1)}$ or not. In the first case, by Lemma 12, we have that $\mathsf{EXP} \subseteq \mathsf{BPSUBEXP}$. By translation, this implies that there is a super-exponential time bound t such that $\mathsf{DTIME}(t) \subseteq \mathsf{BPEXP}$. But we can diagonalize in $\mathsf{DTIME}(t)$ against $\mathsf{BPP}/n^{o(1)}$, since $\mathsf{BPP}/n^{o(1)} \subseteq \mathsf{EXP}/n^{o(1)}$. Hence, in this case, we are done.

If $\mathsf{EXP} \not\subseteq \mathsf{BPP}/n^{o(1)}$, we are done immediately, since $\mathsf{EXP} \subseteq \mathsf{BPEXP}$. □

4.2 Lower bound for MAEXP

The key to our separation for MAEXP is the following advice elimination result, which involves strengthening work of Impagliazzo, Kabanets and Wigderson [IKW02].

Lemma 14. *For any constant $c > 0$, if* $\mathsf{NEXP} \subseteq \mathsf{MA}/n^c$*, then* $\mathsf{NEXP} \subseteq \mathsf{MA}$*.*

Proof. We show that $\mathsf{NEXP} \subseteq \mathsf{MA}/n^c$ implies $\mathsf{NEXP} \subseteq \mathsf{SIZE}(poly)$. $\mathsf{NEXP} \subseteq \mathsf{SIZE}(poly)$ implies $\mathsf{NEXP} = \mathsf{MA}$, by Theorem 7.

Assume, to the contrary, that $\mathsf{NEXP} \subseteq \mathsf{MA}/n^c$ and $\mathsf{NEXP} \not\subseteq \mathsf{SIZE}(poly)$. We derive a contradiction. Since $\mathsf{NEXP} \subseteq \mathsf{MA}/n^c$, we have that $\mathsf{NEXP} \subseteq \mathsf{EXP}/n^c$. Thus we have $\mathsf{EXP}/n^c \not\subseteq \mathsf{SIZE}(poly)$. But this is equivalent to saying that $\mathsf{EXP} \not\subseteq \mathsf{SIZE}(poly)$. To see this, let L be a language in EXP/n^c such that $L \notin \mathsf{SIZE}(poly)$, and let M be a deterministic exponential-time machine taking n^c bits of advice and deciding L. We define a new language $L' \in \mathsf{EXP}$ such that $L' \notin \mathsf{SIZE}(poly)$. L' consists of all pairs $< x, a >$ such that M accepts x given advice a. $L' \in EXP$ since deciding L merely involves simulating the exponential-time machine M. Now suppose $L' \in \mathsf{SIZE}(poly)$. Then we could define polynomial-size circuits for L which simulate the polynomial-size circuits for L' on $< x, a >$, where a is the correct advice for L at length $|x|$ - a is of polynomial size in n and depends only on n.

If $\mathsf{EXP} \not\subseteq \mathsf{SIZE}(poly)$, we have that $\mathsf{MA}/n^c \subseteq \mathsf{i.o.NSUBEXP}/n^c \subseteq \mathsf{i.o.NE}/n^c$. This follows essentially from the connection between circuit lower bounds and derandomization [NW94, KvM99] because circuit lower bounds derandomize not just MA but also the promise version of MA. Now we use the assumption that $\mathsf{NEXP} \subseteq \mathsf{EXP}/n^c$ again, along with the fact that there is a language Q which is complete for NE with respect to linear-time reductions. Since $Q \in \mathsf{EXP}/n^c$, $Q \in \mathsf{DTIME}(2^{n^k})/n^c$ for some fixed k. By the fact that Q is complete for NE with respect to linear-time reductions, we have that $\mathsf{NE} \subseteq \mathsf{DTIME}(2^{n^k})/O(n^c)$. By using the same translation argument as in the proof of Theorem 3, we get that $\mathsf{NE}/n^c \subseteq \mathsf{DTIME}(2^{n^{kc}})/O(n^{c^2})$. Hence $i.o.\mathsf{NE}/n^c \subseteq i.o.\mathsf{DTIME}(2^{n^{kc}})/O(n^{c^2})$. By combining this with the derandomization result, we get that $\mathsf{MA}/n^c \subseteq i.o.\mathsf{DTIME}(2^{n^{kc}})/O(n^{c^2})$. By the assumption that $\mathsf{NEXP} \subseteq \mathsf{MA}/n^c$, we get that $\mathsf{NEXP} \subseteq i.o.\mathsf{DTIME}(2^{n^{kc}})/O(n^{c^2})$, which is a contradiction to Lemma 2. □

Note that Theorem 14 is an unusually strong advice elimination result. Typically, for polynomial-time classes, advice elimination cannot handle more than $\Omega(\log(n))$ advice bits without blowing up the time to super-polynomial. Also, Theorem 14 requires the assumption to hold for *all* languages in NEXP, not just a complete language. Indeed, for a NEXP-complete language to be contained in MA/n^c for a fixed c is equivalent to $\mathsf{NEXP} \subseteq \mathsf{NP}/poly$.

Theorem 15. *For any constant c,* $\mathsf{MAEXP} \not\subseteq \mathsf{MA}/n^c$.

Proof. We consider two cases. The first case is that $\mathsf{NEXP} \not\subseteq \mathsf{MA}/n^c$. In this case, we have that $\mathsf{MAEXP} \not\subseteq \mathsf{MA}/n^c$, since $\mathsf{NEXP} \subseteq \mathsf{MAEXP}$.

The other case is that $\mathsf{NEXP} \subseteq \mathsf{MA}/n^c$. In this case, by Lemma 14, $\mathsf{NEXP} = \mathsf{MA}$. By translation, we get that $\mathsf{NEEXP} = \mathsf{MAEXP}$. In this case too, $\mathsf{MAEXP} \not\subseteq \mathsf{MA}/n^c$, since we can diagonalize against MA/n^c even in deterministic double-exponential time. □

Theorems 13 and 15 can be strengthened so that the lower bound holds against certain superpolynomial time bounds, rather than just polynomial time. However, these time bounds are not easy to state, so we defer the statement of this extension to the journal version of the paper.

4.3 Lower Bound for REXP

The result we obtain for REXP which is somewhat weaker with respect to the advice bound.

Theorem 16. $\mathsf{REXP} \not\subseteq \mathsf{RP}/O(\log(n))$.

Proof. We consider two cases: $\mathsf{NP} \subseteq \mathsf{BPP}/O(\log(n))$ or not. In the first case, we can use the downward self-reducibility of SAT to eliminate the advice and get $\mathsf{NP} \subseteq \mathsf{BPP}$, but this implies $\mathsf{NP} = \mathsf{RP}$, again using downward self-reducibility to eliminate wrong accepting paths. Thus we get $\mathsf{NEXP} = \mathsf{REXP}$, and in this case we even have $\mathsf{REXP} \not\subseteq \mathsf{RP}/n^c$ by using Theorem 3.

If $\mathsf{NP} \not\subseteq \mathsf{BPP}/O(\log(n))$, then we directly have $\mathsf{REXP} \not\subseteq \mathsf{RP}/O(\log(n))$, since $\mathsf{NP} \subseteq \mathsf{REXP}$ and $\mathsf{RP}/O(\log(n)) \subseteq \mathsf{BPP}/O(\log(n))$. □

5 Relativizations

The results in Section 4 rely on tools from the theory of interactive proofs that don't relativize. But our results (and all other known results) on separating nondeterministic and deterministic time exponential and polynomial-time classes do relativize. By examining known and new oracle results we see that we've come close to the limit of what can be done using these relativizable techniques.

In Theorem 6 we showed that for any constant c, $\mathsf{NEXP} \not\subseteq \mathsf{P}^{\mathsf{NP}[n^c]}/n^c$. One cannot remove the n^c limit on queries even without the advice using non-relativizable techniques because of the following result due to Buhrman, Fenner, Fortnow and Torenvliet [BFFT01].

Theorem 17 (BFFT)**.** *There exists a relativized world where* $\mathsf{NEXP} = \mathsf{P}^{\mathsf{NP}}$.

Eric Allender asked whether even Theorem 3 ($\mathsf{NEXP} \not\subseteq \mathsf{NP}/n^c$) can be strengthened to a lower bound that works on almost all input lengths, rather than on infinitely many. Direct diagonalizations tend to work on almost all input lengths–our separation is indirect, and technique does not give this stronger property. We give a new relativized world showing that relativizing techniques cannot get the stronger separation even without the advice.

Theorem 18. *There exists a relativized world such that* $\mathsf{NEXP} \subseteq \mathsf{i.o.NP}$.

Proof. Let M_i be a standard enumeration of non-deterministic relativized Turing machines that runs in time at most 2^{n^i}. Since these machines are paddable, for any A and any $L \in \mathsf{NEXP}^A$ there will some i such that $L = L(M_i^A)$. We will create A such that for every i there are an infinite number of n such that for all x of length n,

$$x \in L(M_i^A) \Leftrightarrow \text{ there exists a } y \text{ with } |y| = 2|x|^i \text{ and } (i, x, y) \in A$$

which immediately implies Theorem 18.

Start with $A = \varnothing$. We construct A in stages (i, j) chosen in any order that cover all possible (i, j).

Stage (i, j): Pick n such that n is larger than any frozen string as well as the n chosen in any previous stage.

Set all strings x of length n to be unmarked.

Repeat the following as long as there is an unmarked x of length n such that $M_i^A(x)$ accepts: Fix an accepting path of $M_i^A(x)$ and freeze every string queried along that path. Mark x. Pick a y, $|y| = 2|x|^i$ such that (i, x, y) is not frozen and let $A = A \cup \{(i, x, y)\}$.

We can always find such a y since we have 2^{2n^i} possible (i, x, y) and at this point since we have frozen at most 2^{n^i} strings for at most 2^n possible x's for a total of $2^{n^i}2^n < 2^{2n^i}$ frozen strings. □

By adding every (i, x, y) that is non frozen in the proof above one can get an even stronger oracle.

Corollary 19. *There exists a relativized world such that* $\mathsf{NEXP} \subseteq \mathsf{i.o.RP}$.

Theorem 18 has independent interest. Consider the nondeterministic time hierarchy [Coo72, SFM78, Žák83].

Theorem 20 (Cook,Seiferas-Fischer-Meyer,Žák)**.** *For any time-constructible functions* t_1 *and* t_2 *such that* $t_1(n+1) = o(t_2(n))$,

$$\mathsf{DTIME}(t_1(n)) \subsetneq \mathsf{DTIME}(t_2(n)).$$

The proofs of Theorem 20 work by looking at collapsing several input lengths. While Rackoff and Seiferas [**?**] showed that the bounds are tight in relativized worlds, Theorem 18 directly shows that one needs to look at many input lengths for a relativizable diagonalization.

Theorem 18 should be contrasted with the fact that by direct diagonalization, it is easy to prove that $\mathsf{NEXP} \not\subseteq \mathsf{i.o.coNP}$ and even that $\mathsf{NEXP} \not\subseteq \mathsf{i.o.coNP}/(n - o(n))$. However, even the latter separation is nearly tight with respect to advice.

Corollary 21. *There exists a relativized world such that* $\mathsf{NEXP} \subseteq \mathsf{i.o.coNP}/n + 1$.

This corollary immediately follows from Theorem 18 and the following folklore theorem.

Theorem 22. $\mathsf{NEXP} \subseteq \mathsf{coNEXP}/n+1$.

Proof. Let L be a language in NEXP and let the advice for length n be the number of strings in L of length n. Our NEXP algorithm with advice for $\overline{L}$ simply guesses and verifies all the strings in L of length n (since we know how many there are) and accepts x is x is not among them. □

6 Conclusions and Open Questions

There are several open questions that remain. Are there explicit languages that witness the lower bounds we obtain? Is $\mathsf{BPEXP} \subseteq \mathsf{i.o.BPP}$, or $\mathsf{MAEXP} \subseteq \mathsf{i.o.MA}$? Is $\mathsf{NEXP} \subseteq \mathsf{NE}/n^c$, for any fixed c? Is there a generic method for obtaining separations for exponential time against polynomial time with advice that works for semantic as well as syntactic classes? Can the advice bounds against which separations are obtained for probabilistic time with two-sided error and one-sided error be improved?

We showed a new advice elimination result for MA in this paper, but there are several other semantic classes for which there are no advice elimination results known: ZPP, $\mathsf{NP} \cap \mathsf{coNP}$, AM etc.

Acknowledgments

We thank Eric Allender for helpful discussions and the question about infinitely-often separation that led to Theorem 18. We also thank many other participants of Dagstuhl Seminar 08381 on the Computational Complexity of Discrete Problems, discussions with whom helped shape this paper.

References

[AG94] Eric Allender and Vivek Gore. A uniform circuit lower bound for the permanent. *SIAM Journal on Computing*, 23(5):1026–1049, 1994.

[AW08] Scott Aaronson and Avi Wigderson. Algebrization: A new barrier in complexity theory. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing (STOC '08)*, 2008. To appear.

[BFFT01] Harry Buhrman, Steve Fenner, Lance Fortnow, and Leen Torenvliet. Two oracles that force a big crunch. *Computational Complexity*, 10(2):93–116, 2001.

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

[BGS75] Theodore Baker, John Gill, and Robert Solovay. Relativizations of the P =? NP question. *SIAM Journal on Computing*, 4(4):431–442, 1975.

[Coo72] Stephen Cook. A hierarchy for nondeterministic time complexity. In *Conference Record, Fourth Annual ACM Symposium on Theory of Computing*, pages 187–192, Denver, Colorado, 1–3 May 1972.

[FLvMV05] Lance Fortnow, Richard Lipton, Dieter van Melkebeek, and Anastasios Viglas. Time-space lower bounds for satisfiability. *Journal of the ACM*, 52(6):833–865, 2005.

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

[FST04] Lance Fortnow, Rahul Santhanam, and Luca Trevisan. Promise hierarchies. *Electronic Colloquium on Computational Complexity(ECCC)*, 11(98), 2004.

[FST05] Lance Fortnow, Rahul Santhanam, and Luca Trevisan. Hierarchies for semantic classes. In *Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing*, 2005.

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

[KI03] Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. In *Proceedings of the 35th Annual ACM Symposium on the Theory of Computing*, pages 355–364, 2003.

[KV87] Marek Karpinski and Rutger Verbeek. On the monte carlo space constructible functions and separation results for probabilistic complexity classes. *Information and Computation*, 75, 1987.

[KvM99] Adam Klivans and Dieter van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing*, pages 659–667, 1999.

[Moc96] Sarah E. Mocas. Separating classes in the exponential-time hierarchy from classes in PH. *Theoretical Computer Science*, 158(1–2):221–231, May 1996.

[NW94] Noam Nisan and Avi Wigderson. Hardness vs randomness. *Journal of Computer and System Sciences*, 49(2):149–167, 1994.

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

[SFM78] Joel Seiferas, Michael Fischer, and Albert Meyer. Separating nondeterministic time complexity classes. *Journal of the ACM*, 25(1):146–167, January 1978.

[SS95] S.Homer and S.Mocas. Nonuniform lower bounds for exponential time classes. In *20th International Symposium on Mathematical Foundations of Computer Science*, pages 159–168, 1995.

[TV02] Luca Trevisan and Salil Vadhan. Pseudorandomness and average-case complexity via uniform reductions. In *Proceedings of the 17th Annual IEEE Conference on Computational Complexity*, volume 17, 2002.

[vMP06] Dieter van Melkebeek and Konstantin Pervyshev. A generic time hierarchy for semantic models with one bit of advice. In *Proceedings of 21st Annual IEEE Conference on Computational Complexity*, pages 129–144, 2006.

[ŽŠ3] Stanislav Žák. A Turing machine time hierarchy. *Theoretical Computer Science*, 26(3):327–333, October 1983.