Robust Simulations and Significant Separations*

Lance Fortnow†
Georgia Institute of Technology

Rahul Santhanam‡
Oxford University

Abstract

We define and study a new notion of "robust simulations" between complexity classes which is intermediate between the traditional notions of infinitely-often and almost-everywhere, as well as a corresponding notion of "significant separations". A language L has a robust simulation in a complexity class C if there is a language in C which agrees with L on arbitrarily large polynomial stretches of input lengths. There is a significant separation of L from C if there is no robust simulation of $L \in \mathsf{C}$.

The new notion of simulation is a cleaner and more natural notion of simulation than the infinitely-often notion. We show that various implications in complexity theory such as the collapse of PH if $\mathsf{NP} = \mathsf{P}$ and the Karp-Lipton theorem have analogues for robust simulations. We then use these results to prove that most known separations in complexity theory, such as hierarchy theorems, fixed polynomial circuit lower bounds, time-space tradeoffs, and the recent theorem of Williams, can be strengthened to significant separations, though in each case, an almost everywhere separation is unknown.

Proving our results requires several new ideas, including a completely different proof of the hierarchy theorem for non-deterministic polynomial time than the ones previously known.

*Results in this paper have been presented in ICALP 2011 [FS11] and TAMC 2015 [For15]. This journal article extends these results and provides full proofs.

†Much of this research done while at Northwestern University supported in part by NSF grants CCF-0829754 and DMS-0652521.

‡Much of this research done while at University of Edinburgh

1 Introduction

What does the statement "$\mathsf{P} \neq \mathsf{NP}$" really tell us? All it says is that for any polynomial-time algorithm A, A fails to solve SAT on an infinite number of inputs. These hard-to-solve inputs could be exponentially (or much worse) far from each other. Thus even a proof of $\mathsf{P} \neq \mathsf{NP}$ could leave open the possibility that SAT or any other NP-complete problem is still solvable on all inputs encountered in practice. This is unsatisfactory if we consider that one of the main motivations of proving lower bounds is to understand the limitations of algorithms.

Another important motivation for proving lower bounds is that hardness is *algorithmically useful* in the context of cryptography or derandomization. Again, if the hardness only holds for inputs or input lengths that are very far apart, this usefulness is called into question. For this reason, theorists developed a notion of almost-everywhere (a.e.) separations, and a corresponding notion of infinitely-often (i.o.) simulations. A language L is in $\mathsf{i.o.C}$ for a complexity class C if there is some $A \in \mathsf{C}$ such that A and L agree on infinitely many input lengths. A class D is almost everywhere not in C if for some language L in D, $L \notin \mathsf{i.o.C}$, that is any C-algorithm fails to solve L on all but a finite number of input lengths. As an example of applying these notions, Impagliazzo and Wigderson [IW97] show that if $\mathsf{E} \not\subseteq \mathsf{SIZE}(2^{o(n)})$ then BPP is in $\mathsf{i.o.P}$, and that if $\mathsf{E} \not\subseteq \mathsf{i.o.SIZE}(2^{o(n)})$, then $\mathsf{BPP} = \mathsf{P}$.

However, the infinitely often notion has its own issues. Ideally, we would like a notion of simulation to capture "easiness" in some non-trivial sense. Unfortunately, many problems that we consider hard have trivial infinitely often simulations. For example, consider any NP-hard problem on graphs or square matrices. The natural representation of inputs for such problems yields non-trivial instances only for input lengths that are perfect squares. In such a case, the problem has a trivial infinitely often simulation on the set of all input lengths which are not perfect squares. On the other hand, the problem could be "padded" so that it remains non-trivial on input lengths which are not perfect squares. It's rather unsatisfactory to have a notion of simulation which is so sensitive to the choice of input representation.

Not unrelated to this point is that analogues of many classical complexity results fail to hold in the infinitely often setting. For example, we do not know if $SAT \in \mathsf{i.o.P}$ implies that the entire Polynomial Hierarchy has simulations infinitely-often in polynomial time. We also don't know if in general a complete language for a class being easy infinitely-often implies that the entire class is easy infinitely-often. This is true for SAT and NP because SAT is paddable and downward self-reducible, but for general complexity classes the implication is unclear. Given that even these basic analogues are not known, it's not surprising that more involved results such as the Karp-Lipton theorem [KL82] and the theorem of Impagliazzo, Kabanets and Wigderson [IKW02] don't have known infinitely often analogues either.

In an ideal world, we would like all our algorithms to work on all input lengths, and all our separations to be almost-everywhere separations. While algorithm design does typically focus on algorithms that work on all input lengths, many of the complexity separations we know do not work in the almost everywhere setting. Separations proved using combinatorial or algebraic methods, such as Hastad's lower bound for Parity [Hås86] or Razborov's monotone circuit lower bound for Clique [Raz85] tend to be almost everywhere (in an appropriate input representation). However, such techniques typically have intrinsic limitations, as they run into the natural proofs barrier [RR97]. Many of the lower bounds proved recently have come from the use of indirect diagonalization. A contrary upper bound is assumed and this assumption is used together with various other ideas to derive a contradiction to a hierarchy theorem. These newer results include hierarchy theorems [Bar02, FS04, vMP06], time-space tradeoffs [For00, FLvMV05], and circuit lower

bounds [BFT98, Vin05, San07, Wil10a, Wil10b]. Unfortunately, *none* of these results give almost everywhere separations, and so the question immediately arises what we can say quantitatively about these separations, in terms of the frequency with which they hold.

To address all these issues, we describe a new notion of "robust simulation" and a corresponding notion of "significant separation". A language L is in $\mathsf{r.o.C}$ (robustly-often in C) if there is a language A in C such that for every k there are infinitely many m and such that A and L agree on all inputs between m and m^k. A class D has a significant separation from C if there is some L in D such that $L \notin \mathsf{r.o.C}$. This implies that for each $L' \in \mathsf{C}$, there is a constant k such that for each m, L and L' differ on at least one input length between m and m^k. Intuitively, this means that if the separation holds at some input length, there is another input length at most polynomially larger at which the separation also holds, i.e., the hardness is not too "sparsely" distributed.

Our definition of robust simulations extends the notion of uniform hardness of Downey and Fortnow [DF03]. A set A is uniformly hard in the sense of Downey and Fortnow if $A \notin \mathsf{r.o.P}$.

The notion of robust simulation is just slightly stronger than the notion of infinitely often simulation, and correspondingly the notion of significant separation is slightly weaker than that of almost everywhere separations. By making this tradeoff, however, we show that we can often achieve the best of both worlds.

We give robustly often analogues of many classical complexity results, where infinitely often analogues remain open, including

- $\mathsf{NP} \subseteq \mathsf{r.o.P}$ implies $\mathsf{PH} \subseteq \mathsf{r.o.P}$
- $\mathsf{NP} \subseteq \mathsf{r.o.SIZE}(\mathit{poly})$ implies $\mathsf{PH} \subseteq \mathsf{r.o.P}$
- $\mathsf{NEXP} \subseteq \mathsf{r.o.SIZE}(\mathit{poly})$ implies $\mathsf{NEXP} \subseteq \mathsf{r.o.MA}$

We then use these robustly often analogues together with other ideas to give several significant separations where almost everywhereseparations remain open, including

- $\mathsf{NTIME}(n^r) \not\subseteq \mathsf{r.o.NTIME}(n^s)$, when $r > s \geqslant 1$
- For each constant k, $\Sigma_2\mathsf{P} \not\subseteq \mathsf{r.o.SIZE}(n^k)$
- $SAT \not\subseteq \mathsf{r.o.DTISP}(n^\alpha, \mathsf{polylog}(n))$ when $\alpha < \sqrt{2}$
- $\mathsf{NEXP} \not\subseteq \mathsf{r.o.ACC}^0$

The robustly often notion gives us a cleaner and more powerful theory than the infinitely often notion.

1.1 Intuition and Techniques

To illustrate the advantages of the robustly often notion over the infinitely often notion, let's look at a simple example: trying to show that if SAT is easy, then all of NP is easy. Let $L \in \mathsf{NTIME}(n^k)$ be any NP language, where $k > 0$ is a constant. $\mathsf{SAT} \in \mathsf{i.o.P}$ doesn't immediately imply $L \in \mathsf{i.o.P}$, as the range of the reduction from L to SAT might only intersect input lengths where the polynomial-time algorithm for SAT is incorrect. In this case, the problem can be fixed by padding the reduced instance to polynomially many different input lengths and using downward self-reducibility to check YES answers for any of these inputs. However, this fix depends on specific properties of SAT.

Showing that $\mathsf{SAT} \in \mathsf{r.o.P}$ implies $L \in \mathsf{i.o.P}$ is an easier, more generic argument. Define a robust set of natural numbers to be any set S such that for each $k > 0$ there is an m for which

S contains all numbers between m and m^k for some m. $\mathsf{SAT} \in \mathsf{r.o.P}$ means that there is a robust set S on which the simulation works. Now the reduction from L to SAT creates instances of length $n^k\mathsf{polylog}(n)$, and it's not hard to see that this automatically implies that composing the reduction with the algorithm for SAT gives an algorithm for L which works on some robust subset S' of S. This implies $L \in \mathsf{r.o.P}$. We call a robust subset of a set a *robust refinement*. Many of our arguments will involve defining a series of robust refinements of a robust set such that the desired simulation holds on the final refinement in the series, thus implying that the simulation goes through in the robustly often setting.

Thus far, using robustly often seems easier but hasn't given us any additional power. The situation changes if we consider implications where the assumption is used *two or more* times. An example is the proof that $\mathsf{NP} \subseteq \mathsf{P}$ implies $\mathsf{PH} \subseteq \mathsf{P}$ - this is an inductive proof where the assumption is used several times. Trying to carry an infinitely often simulation through fails miserably in this setting because two infinitely often simulations do not compose - they might work on two completely different infinite sets of input lengths.

Now two robustly often simulations do not in general compose either. It is not in general true that for complexity classes B, C and D, $\mathsf{B} \subseteq \mathsf{r.o.C}$ and $C \subseteq \mathsf{r.o.D}$, then $\mathsf{B} \subseteq \mathsf{r.o.D}$. However, we *can* get these two robustly often simulations to compose when they are *both* consequences of a single robustly often assumption. The robustly often assumption gives us some robust set to work with. If we're careful we can define a single robust refinement of this set on which $\mathsf{B} \subseteq \mathsf{C}$ holds and so too does $\mathsf{C} \subseteq \mathsf{D}$, which implies $\mathsf{B} \subseteq \mathsf{D}$ holds on this refinement as well.

This is an idea that we will use again and again in our proofs. However, in order to use this idea, we need to be careful with the steps in our proof, as there are only some kinds of implications for which the idea is useful. For example, it works well with fixed polynomial-time reductions or translations with fixed polynomial advice, but not with exponential padding. More importantly, the idea only works when all the steps in the proof follow from a *single* assumption, so we need to re-formulate proofs so that they conform to this pattern. In some cases, eg. the proofs of Theorem 20 and Theorem 26, the re-formulation is non-trivial and leads to proofs that are quite a bit more involved than the originals [IKW02, Kan82].

In the case of significant hierarchies for non-deterministic time, i.e., hierarchies where the lower bound is against robust simulations, the known techniques break down entirely. The traditional argument is a "chaining argument" [Coo72, SFM78, Ž83] which uses a chain of exponentially many input lengths and cannot possibly give a hierarchy against robustly often simulations. Here, we come up with a novel idea of chaining using witnesses to get a significant hierarchy for polynomial time.

Our most technically involved result is that the recent breakthrough lower bound of Williams [Wil10a, Wil10b] can be strengthened to a significant separation. The proof of this result uses almost all of the techniques we develop, including the sophisticated use of robust refinements involved in proving Theorem 20, and a variant of the significant hierarchy for non-deterministic polynomial time.

Implicit in our paper is a certain proof system for proving complexity class separations, such that any separation proved in this system automatically yields an infinitely often separation. It's an interesting open problem to make this system explicit, and to study its power and its limitations more formally.

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, Σ_2^p,PP and their exponential-time versions.The Complexity Zoo[1] is an excellent resource for basic definitions and statements of results.

Given a complexity class C, coC is the class of languages L such that $\bar{L} \in \mathsf{C}$. Given a function $s : \mathbb{N} \to \mathbb{N}$, 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 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 CTIME(t)/a, for $a : \mathbb{N} \to \mathbb{N}$, 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.

Definition 1. *Let S be a subset of positive integers. S is* robust *if for each positive integer k, there is a positive integer $m \geqslant 2$ such that $n \in S$ for all $m \leqslant n \leqslant m^k$.*

Note that any robust set is infinite. We now define what it means to simulate a language in a complexity class on a subset of the positive integers.

Definition 2. *Let L be a language, C a complexity class, and S a subset of the positive integers. We say $L \in \mathsf{C}$ on S if there is a language $L' \in \mathsf{C}$ such that $L_n = L'_n$ for any $n \in S$.*

Using the terminology of Definition 2, $L \in$ i.o.C for a language L and complexity class C if there is some infinite set $S \subseteq \mathbb{N}$ such that $L \in \mathsf{C}$ on S. We now define our main notion of robustly-often simulations.

Definition 3. *Given a language L and complexity class C, $L \in$ r.o.C if there is a robust S such that $L \in C$ on S. In such a case, we say that there is a robustly-often (r.o.) simulation of L in C.*

[1] `http://qwiki.caltech.edu/wiki/ComplexityZoo`

We extend this notion to complexity classes in the obvious way - given complexity classes B *and* C, $\mathsf{B} \subseteq \mathsf{r.o.C}$ *if for each language* $L \in \mathsf{B}$, $L \in \mathsf{r.o.C}$. *If* $\mathsf{B} \not\subseteq \mathsf{r.o.C}$, *we say that there is a significant separation of* B *from* C.

Clearly $\mathsf{B} \subseteq \mathsf{r.o.C}$ implies $\mathsf{B} \subseteq \mathsf{i.o.C}$. Conversely, $\mathsf{B} \not\subseteq \mathsf{i.o.C}$ gives a very strong separation of B and C, i.e., an almost-everywhere separation, while a significant separation is somewhat weaker but still much more significant than simply a separation of B and C. Intuitively, a significant separation means that input lengths witnessing the separation are at most polynomially far apart.

We now define a sequence of *canonical refinements* for any given set S, which will play an important part in many of our proofs.

Definition 4. *Let* S *be a robust set. The* canonical refinement S_d *of* S *at level* d *is defined as follows for any integer* $d > 0$*:* $m \in S_d$ *iff* $m \in S$ *and* $n \in S$ *for all* n *with* $m \leqslant n \leqslant m^d$.

We repeat the definitions of robust set and robust refinement from the introduction.

Definition 5. *A set of natural numbers* S *is* robust *if for each* $k > 0$ *there is an* m *for which* S *contains all numbers between* m *and* m^k *for some* m. *We call a robust subset of a set a* robust refinement.

If S is robust then S_d is a robust refinement of S and of every $S_{d'}$ for $d' \leqslant d$.

We'll also need the notion of an honest reduction.

Definition 6. *An* polynomial-time honest m-reduction *from* A *to* B *is a polynomial-time computable function mapping* Σ^* *to* $\Sigma^* \cup \{+, -\}$ *such that*

1. *For some integer* k, *for all* $n > 1$ *and for all* x, *either* $f(x) \in \{+, -\}$ *or* $|x| \geqslant |f(x)|^k$ *where* $|x|$ *is the length of the string* x.
2. *If* x *is in* A *then* $f(x) \in B \cup \{+\}$.
3. *If* x *is not in* A *then* $f(x) \in \overline{B} \cup \{-\}$, *where* $\overline{B} = \Sigma^* - B$.

3 Robust Simulations

For any NP-complete language L the language

$$L' = \{x10^i \in L \mid |x| + 1 + i \text{ is even}\}$$

remains NP-complete but sits in $\mathsf{i.o.P}$. In contrast if any NP set complete under honest reductions sits in $\mathsf{r.o.P}$ then $\mathsf{NP} \subseteq \mathsf{r.o.P}$.

Lemma 7. *Let* L *and* L' *be languages such that* L' *reduces to* L *via a polynomial-time honest m-reduction. Let* C *be a complexity class closed under poly-time m-reductions. If there is a robust* S *such that* $L \in \mathsf{C}$ *on* S, *then there is a robust refinement* S' *of* S *such that* $L' \in \mathsf{C}$ *on* S'.

Proof. Let f be a polynomial-time honest m-reduction from L' to L. By assumption, there is a robust S such that $L \in \mathsf{C}$ on S. Let $K \in C$ be a language such that $L_n = K_n$ for all $n \in S$. We define a robust refinement S' of S as follows: $\hat{n} \in S'$ iff $\hat{n} \in S$ and for all x of length $\hat{n}$, $|f(x)| \in S$. By definition, S' is a refinement of S; we show that it is robust and that $L' \in C$ on S'.

First, we show robustness of S'. Robustness of S is equivalent to saying that for each positive integer k, there is a positive integer m_k such that $n \in S$ for all $m_k^{1/k} \leqslant n \leqslant m_k^k$. Since f is an honest

reduction, there is an integer $c > 1$ such that for all x, if $f(x) \in \Sigma^*$ then $|x|^{1/c} \leqslant |f(x)| \leqslant |x|^c$. We show that for each positive integer k, there is a positive integer $\hat{m}_k$ such that $\hat{n} \in S'$ for all $\hat{m}_k^{1/k} \leqslant \hat{n} \leqslant \hat{m}_k^k$. Simply choose $\hat{m}_k = m_{ck}$. We have that for any $\hat{n}$ such that $m_{ck}^{1/k} \leqslant \hat{n} \leqslant m_{ck}^k$, for all x of length $\hat{n}$, $|f(x)|$ is between $(\hat{n})^{1/c}$ and $(\hat{n})^c$ by assumption on $f(x)$. Hence for all x of length $\hat{n}$ with $f(x) \in \Sigma^*$, $|f(x)|$ is between $m_{ck}^{1/ck}$ and m_{ck}^{ck}, which implies $f(x) \in S$ for all such x, and hence $\hat{n} \in S'$.

Next we show $L' \in \mathsf{C}$ on S'. Define a language K' as follows: $x \in K'$ iff $f(x) \in K \cup \{+\}$. Since C is closed under poly-time m-reductions, $K' \in \mathsf{C}$. Now, on each input length $\hat{n} \in S'$, for any x of length $\hat{n}$ with $f(x) \in \Sigma^*$, $|f(x)| \in S$, and hence $L(x) = K(x)$. This implies that $L'(x) = K'(x)$ for all such x, and hence $L'_{\hat{n}} = K'_{\hat{n}}$, which finishes the proof. □

The proof ideas of Lemma 7 can be used to show that robustly often analogues of various useful implications hold. We omit the proofs of these propositions, since they follow from the definition of robustly often in much the same way as Lemma 7.

The first analogue essentially says that we can take a complete language to be representative of a complexity class, even in the context of robustly often simulations. It is an immediate consequence of Lemma 7 and the fact that SAT is NP-complete under honest reductions.

Proposition 8. *If* $\mathsf{SAT} \in \mathsf{r.o.P}$*, then* $\mathsf{NP} \subseteq \mathsf{r.o.P}$*.*

In fact, if $\mathsf{SAT} \in \mathsf{P}$ on S for some robust set S and $L \in \mathsf{NTIME}(n^d)$ for some integer d, the proof of Proposition 8 gives that $L \in \mathsf{P}$ on S_{d+1}, i.e., on the canonical refinement of S at level $d+1$.

The next proposition says that translation arguments carry through in the robustly often setting, for any "reasonable" complexity measure where "reasonable" means that the measure is closed under efficient deterministic transductions (see Van Melkebeek and Pervyshev [vMP06] for a formal definition). All complexity measures considered in this paper are reasonable in this sense.

Proposition 9. *Let* BTIME *and* CTIME *be any complexity measures closed under efficient deterministic transductions. Let g and h be time-constructibe functions, and p a polynomial. If* $\mathsf{BTIME}(g(n)) \subseteq \mathsf{r.o.CTIME}(h(n))$*, then* $\mathsf{BTIME}(g(p(n))) \subseteq \mathsf{r.o.CTIME}(h(p(n)))$*.*

As a consequence of Proposition 9, we get for example that if $\mathsf{NTIME}(n) \subseteq \mathsf{r.o.P}$, then $\mathsf{NP} \subseteq \mathsf{r.o.P}$. There are contexts where exponential padding is used in complexity theory, eg., in the proof that $\mathsf{NP} = \mathsf{P}$ implies $\mathsf{NEXP} = \mathsf{EXP}$. This result doesn't seem to translate to the robustly often setting, however a weaker version does, by noting that an exponential span of input lengths for NP translate to a single input length for NEXP in the usual translation argument.

Proposition 10. *If* $\mathsf{NP} \subseteq \mathsf{r.o.P}$*, then* $\mathsf{NEXP} \subseteq \mathsf{i.o.EXP}$*.*

The proposition below says that simulations of a syntactic class in another class can be translated to a simulation with fixed polynomial advice, even in the robustly often setting.

Proposition 11. *Let* BTIME *be a syntactic complexity measure and* CTIME *a complexity measure, such that both* BTIME *and* CTIME *are closed under efficient deterministic transductions. Let f and g be time-constructrible measures and p a polynomial. If* $\mathsf{BTIME}(f(n)) \subseteq \mathsf{r.o.CTIME}(g(n))$*, then* $\mathsf{BTIME}(f(n))/p(n) \subseteq \mathsf{r.o.CTIME}(g(n + p(n)))/p(n)$*.*

Theorem 12. *If* $\mathsf{NP} \subseteq \mathsf{r.o.P}$*, then* $\mathsf{PH} \subseteq \mathsf{r.o.P}$

Proof. We show that for any positive integer k, $\Sigma_k^p \subseteq \mathsf{r.o.P}$. The proof is by induction on k. We will need to formulate our inductive hypothesis carefully.

By assumption, $\mathsf{SAT} \in \mathsf{r.o.DTIME}(n^c)$ for some integer $c > 0$. Let S be a robust set such that $\mathsf{SAT} \in \mathsf{DTIME}(n^c)$ on S. Using the same idea as in the proof of Lemma 7, we have that for any $L \in \mathsf{NTIME}(n^d)$, $L \in \mathsf{DTIME}(n^{cd})$ on S_d, where S_d is the canonical refinement defined in Definition 4. Moreover, the conclusion holds even if the assumption is merely that $L \in \mathsf{NTIME}(n^d)$ on S_d.

Now we formulate our inductive hypothesis H_k as $\Sigma_k\mathrm{SAT} \in \mathsf{DTIME}(n^{c^{k+o(1)}})$ on $S_{c^{k+1}}$. For $k = 1$, the hypothesis holds because by assumption $\mathsf{SAT} \in DTIME(n^c)$ on S and hence on S_{c^2} since S_{c^2} is a refinement of S. If we perform the inductive step, the proof of the theorem will be complete using Lemma 7 and the fact that $\Sigma_k\mathrm{SAT}$ is complete for Σ_k^p under polynomial-time honest m-reductions.

Assume that H_k holds - we will establish H_{k+1}. Since H_k holds, we have that $\Sigma_k\mathrm{SAT} \in \mathsf{DTIME}(n^{c^{k+o(1)}})$ on $S_{c^{k+1}}$. Since SAT can be computed in nondeterministic quasilinear time, $\Sigma_{k+1}\mathrm{SAT}$ is in $\mathsf{NTIME}(n^{1+o(1)})^{\Sigma_k\mathrm{SAT}}$. Moreover, by using padding, we can assume that that all the queries of the oracle machine are of length $n^{1+o(1)}$. Now, using the same idea as in the proof of Lemma 7, we have that $\Sigma_{k+1}\mathrm{SAT} \in \mathsf{NTIME}(n^{c^{k+o(1)}})$ on $S_{c^{k+2}}$. Using again the assumption that $\mathsf{SAT} \in \mathsf{DTIME}(n^c)$ on S,we have for each language $L \in \mathsf{NTIME}(n^{c^{k+o(1)}})$ on $S_{c^{k+1}}$, $L \in \mathsf{DTIME}(n^{c^{k+1+o(1)}})$ on $S_{c^{k+1}}$ and hence on $S_{c^{k+2}}$ since $S_{c^{k+2}}$ is a refinement of $S_{c^{k+1}}$. Thus we have that $\Sigma_{k+1}\mathrm{SAT} \in \mathsf{DTIME}(n^{c^{k+1+o(1)}})$ on $S_{c^{k+1}}$, which establishes H_{k+1} and completes the proof of the theorem. □

Theorem 13. *If* $\mathsf{NP} \subseteq \mathsf{r.o.SIZE}(poly)$*, then* $\mathsf{PH} \subseteq \mathsf{r.o.SIZE}(poly)$

Theorem 14. *If* $\mathsf{NP} \subseteq \mathsf{r.o.BPP}$*, then* $\mathsf{PH} \subseteq \mathsf{r.o.BPP}$*.*

We omit the proof of Theorems 13 and 14, which closely resemble the proof of Theorem 12.

Next we show a robust analogue of the Karp-Lipton theorem [KL82]. We formulate a stronger statement which will be useful when we show significant fixed-polynomial circuit size lower bounds for Σ_2^p. The proof is simpler than for some of the other analogues, since the assumption is only used once during the proof.

Lemma 15. *If there is a constant k and a robust set S such that $\mathsf{SAT} \in \mathsf{SIZE}(n^k)$ on S, then there is a robust refinement S' of S such that $\Pi_2 SAT \in \Sigma_2\mathsf{TIME}(n^{k+1+o(1)})$.*

Proof. We follow the usual proof of the Karp-Lipton theorem. Let $\{C_n\}$ be a sequence of circuits such that C_n solves SAT correctly for $n \in S$ and $|C_n| = O(n^k)$. Using self-reducibility and paddability of SAT, we can define a sequence $\{C'_n\}$ of circuits such that C'_n outputs a satisfying assignment for all satisfiable formulae of length $n \in S$, and $|C'_n| = O(n^{k+1})$ for all n.

Now let ϕ be an instance of $\Pi_2 SAT$: $\phi \in \Pi_2 SAT$ iff $\forall\vec{x}\exists\vec{y}\phi(\vec{x},\vec{y})$. By NP-completeness and paddability of SAT, there is a polynomial-time computable function f such that $\phi \in \Pi_2 SAT$ iff $\forall\vec{x} f(< \phi, \vec{x} >) \in SAT$, and $|z| \leqslant |f(z)| \leqslant |z|^{1+o(1)}$. Consider the following Σ_2 algorithm for ϕ: it existentially guesses a circuit C' of size $O(n^{k+1})$ and universally verifies for all $\vec{x}$ that $\phi(\vec{x}, C'(f(< \phi, \vec{x} >)))$ holds. This algorithm decides $\Pi_2 SAT$ correctly on all $n \in S_2$, where S_2 is the canonical refinement of S at level 2. The time taken by the algorithm is $O(n^{k+1+o(1)})$, which gives the desired conclusion. □

The following is an immediate corollary.

Corollary 16. *If* $\mathsf{NP} \subseteq \mathsf{r.o.SIZE}(poly)$*, then* $\Sigma_2^p \subseteq \mathsf{r.o.}\Pi_2^p$*.*

Analogously, it is easy to show an robustly often version of a theorem credited to Nisan by Babai, Fortnow and Lund [BFL91]. The following lemma follows from their techniques.

Lemma 17. *Let L be a language complete for E under linear-time reductions. If there is a robust set S and a constant $k \geqslant 1$ such that $L \in \mathsf{SIZE}(n^k)$ on S, then there is a robust refinement S' of S such that $L \in \mathsf{MATIME}(n^{2k})$ on S'.*

The analogue of Nisan's result follows as a corollary.

Corollary 18. *If $\mathsf{EXP} \subseteq \mathsf{r.o.SIZE}(poly)$, then $\mathsf{EXP} \subseteq \mathsf{r.o.MA}$.*

The following can be shown using the easy witness method of Kabanets [Kab01, IKW02] and known results on pseudo-random generators [NW94, KvM99].

Lemma 19. *Let R be any robust set and $k > 1$ be any constant. Then there is a robust refinement R' of R such that either $\mathsf{NE} \subseteq \mathsf{DTIME}(2^{n^{16k^4}})$ on R or $\mathsf{MATIME}(n^{4k^2}) \subseteq \mathsf{NE}/O(n)$ on R'.*

We next derive an robustly often analogue of the main theorem of Impagliazzo, Kabanets and Wigderson [IKW02].

Theorem 20. *If $\mathsf{NEXP} \subseteq \mathsf{r.o.SIZE}(poly)$, then $\mathsf{NEXP} \subseteq \mathsf{r.o.MA}$.*

Proof. Let $k > 1$ be a constant and L be a complete set for NE under linear-time reductions for which there is a robust set S such that $L \in \mathsf{SIZE}(n^k)$ on S. This implies that there is a robust refinement S' of S such that $\mathsf{NE} \subseteq \mathsf{SIZE}(n^k)$ on S', using Lemma 7. Using Lemma 17, there is a robust refinement $S^{(2)}$ of S' such that $\mathsf{E} \subseteq \mathsf{MATIME}(n^{2k})$ on $S^{(2)}$. Using Proposition 9, we have that there is a robust refinement $S^{(3)}$ of $S^{(2)}$ such that $\mathsf{DTIME}(2^{n^{2k}}) \subseteq \mathsf{MATIME}(n^{4k^2})$ on $S^{(3)}$. Now set $R = S^{(3)}$ in Lemma 19.

There are two cases. Case 1 is that we have $\mathsf{NE} \subseteq \mathsf{DTIME}(2^{n^{16k^4}})$ on $S^{(2)}$. In this case, by using padding and applying Proposition 9, we have that $\mathsf{NE} \subseteq \mathsf{DTIME}(2^{n^{16k^4}}) \subseteq \mathsf{MA}$ on $S^{(2)}_{16k^4+1}$. This gives that $\mathsf{NE} \subseteq \mathsf{r.o.MA}$, which implies $\mathsf{NEXP} \subseteq \mathsf{r.o.MA}$ by applying Proposition 9 again, and we have the desired conclusion.

Case 2 is that $\mathsf{MATIME}(n^{4k^2}) \subseteq \mathsf{NE}/O(n)$ on some robust refinement $S^{(4)}$ of $S^{(3)}$. We will derive a contradiction in this case. Since $S^{(4)}$ is a refinement of $S^{(3)}$, we have that $\mathsf{DTIME}(2^{n^{2k}}) \subseteq \mathsf{MATIME}(n^{4k^2}) \subseteq \mathsf{NE}/O(n)$ on $S^{(4)}$. Since $S^{(4)}$ is a refinement of S', we have that $\mathsf{NE} \subseteq \mathsf{SIZE}(n^k)$ on $S^{(4)}$. Applying Proposition 11, we have that there is a robust refinement $S^{(5)}$ of $S^{(4)}$ such that $\mathsf{NE}/O(n) \subseteq \mathsf{SIZE}(n^k)$ on $S^{(5)}$. Thus $\mathsf{DTIME}(2^{n^{2k}}) \subseteq \mathsf{r.o.SIZE}(n^k)$, but this is in contradiction to the fact that $\mathsf{DTIME}(2^{n^{2k}}) \not\subseteq \mathsf{i.o.SIZE}(n^k)$ by direct diagonalization. □

4 Significant Separations

4.1 Hierarchies

The proofs of the hierarchies for deterministic time and space actually give almost-everywhere separations and therefore significant separations.

For nondeterministic time the situation is quite different. Cook [Coo72] showed that $\mathsf{NTIME}(n^r) \subsetneq \mathsf{NTIME}(n^s)$ for any reals $r < s$. Seiferas, Fischer and Meyer [SFM78] generalize this result to show that $\mathsf{NTIME}(t_1(n)) \subsetneq \mathsf{NTIME}(t_2(n))$ for $t_1(n+1) = o(t_2(n))$. Zak [ŽŚ83] gives a simpler proof of the same result. All these proofs require building an exponential (or worse) chain of equalities to

get a contradiction. Their proofs do not give almost everywhere separations or significant separations. No relativizable proof can give an i.o. hierarchy as Buhrman, Fortnow and Santhanam give a relativized world that NEXP $\subseteq$ i.o.NP.

In this section we give relativizing proofs that NEXP $\not\subseteq$ r.o.NP and that NTIME(n^r) $\not\subseteq$ r.o.NTIME(n^s) for $r > s \geqslant 1$. The latter proof requires a new proof of the traditional nondeterministic time hierarchy.

Theorem 21. NEXP $\not\subseteq$ r.o.NP.

Proof. Assume, to the contrary, that there is an infinitely-often robust simulation of NEXP by NP. Let L be a language which is paddable and complete for NE under linear-time m-reductions. By assumption, there is a robust set S and an integer k such that $L \in \mathsf{NTIME}(n^k)$ on S.

Let K be a set in $\mathsf{DTIME}(2^{n^{2k}})$ but not in i.o.NTIME(n^k) - such a set can be constructed by direct diagonalization. Consider the following padded version K' of K: $y \in K'$ iff $y = x01^{|x|^{2k}-|x|-1}$ for $x \in K$. Since $K \in \mathsf{DTIME}(2^{n^{2k}})$, we have that $K' \in \mathsf{E} \subseteq \mathsf{NE}$. Now using the assumption on L and the proof idea of Lemma 7, we have that $K' \in \mathsf{NTIME}(n^k)$ on S_2, where S_2 is the canonical refinement of S at level 2. Since there is an m-reduction from K to K' running in time $O(n^{2k})$ which only blows up the instance length, we can use the proof idea of Lemma 7 again to show that $K \in \mathsf{NTIME}(n^{2k^2})$ on S_{4k}. Now, since $K \in \mathsf{NE}$ and L is complete for NE, we can use the proof idea of Lemma 7 a third time to show that $K \in \mathsf{NTIME}(n^k)$ on S_{4k}. But since S_{4k} is infinite, this implies $K \in \mathsf{NTIME}(n^k)$ on an infinite set of input lengths, which contradicts our assumption on K.

□

You can add advice to the same proof to show

Theorem 22. NEXP $\not\subseteq$ r.o.NP/$n^{o(1)}$

Any strengthening of Theorem 22 would imply major new circuit lower bounds for NEXP.

In the purely uniform setting, we now show a stronger version of Theorem 21 in the form of a significant hierarchy for non-deterministic polynomial time.

Theorem 23. *If t_1 and t_2 are time-constructible functions such that*

- $t_1(n) = o(t_2(n))$, *and*
- $n \leqslant t_1(n) \leqslant n^c$ *for some constant c*

then NTIME($t_2(n)$) $\not\subseteq$ r.o.NTIME($t_1(n)$).

Corollary 24. *For any reals $1 \leqslant r < s$,* NTIME(n^s) $\not\subseteq$ r.o.NTIME(n^r).

Proof of Theorem 23. Let $M_1, M_2, \ldots$ be an enumeration of multitape nondeterministic machines that run in time $t_1(n)$.

Define a nondeterministic Turing machine M that on input 1^i01^m0w does as follows:

- If $|w| < t_1(i+m+2)$ accept if both $M_i(1^i01^m0w0)$ and $M_i(1^i01^m0w1)$ accepts.
- If $|w| \geqslant t_1(i+m+2)$ accept if $M_i(1^i01^m0)$ rejects on the path specified by the bits of w.

Since we can universally simulate $t(n)$-time nondeterministic multitape Turing machines on an $O(t(n))$-time 2-tape nondeterministic Turing machine, $L(M) \in \mathsf{NTIME}(O(t_1(n+1))) \subseteq \mathsf{NTIME}(t_2(n))$. Note $(n+1)^c = O(n^c)$ for any c.

Suppose $\mathsf{NTIME}(t_2(n)) \subseteq \mathsf{r.o.NTIME}(t_1(n))$. Pick a c such that $t_1(n) < n^{c-1}$. By the definition of r.o. there is some n_0 and a language $L \in \mathsf{NTIME}(t_1(n))$ such that $L(M) = L$ on all inputs of length between n_0 and n_0^c. Fix i such that $L = L(M_i)$. Then $z \in L(M_i) \Leftrightarrow z \in L(M)$ for all $z = 1^i 01^{n_0} 0w$ for $w \leqslant t_1(i + n_0 + 2)$.

By induction we have $M_i(1^i 01^{n_0} 0)$ accepts if $M_i(1^i 01^{n_0} 0w)$ accepts for all $w \leqslant t_1(i + n_0 + 2)$. So $M_i(1^i 01^{n_0} 0)$ accepts if and only $M_i(1^i 01^{n_0} 0)$ rejects on every computation path, contradicting the definition of nondeterministic time. □

4.2 Circuit Lower Bounds

Kannan [Kan82] showed how to diagonalize in Σ_4^p against circuit size n^k almost everywhere, for any constant k. He combined this result with the Karp-Lipton theorem to obtain fixed polynomial circuit size lower bounds in Σ_2^p. However, the lower bound for Σ_2^p no longer holds almost everywhere, as a consequence of the proof strategy used by Kannan. It has been open since then to derive an almost everywhere separation in this context. We manage to obtain a significant separation. We need a quantitative form of Kannan's original diagonalization result. Kannan's paper contains a slightly weaker form of this result, but his proof does yield the result below.

Theorem 25. *[Kan82] For any constant $k \geqslant 1$, $\Sigma_4\mathsf{TIME}(n^{3k}) \not\subseteq \mathsf{i.o.SIZE}(n^k)$*

Theorem 26. *For each integer $k > 1$, $\Sigma_2^p \not\subseteq \mathsf{r.o.SIZE}(n^k)$.*

Proof. Assume, contrarily, that $\Sigma_2^p \subseteq \mathsf{r.o.SIZE}(n^k)$. Let L be a set that is complete for $\Sigma_2\mathsf{TIME}(n^{4k^4})$ under linear-time reductions, and S be a robust set such that $L \in \mathsf{SIZE}(n^k)$ on S. Hence also $\Sigma_2 SAT \in \mathsf{SIZE}(n^k)$ on S. This implies $\mathsf{SAT} \in \mathsf{SIZE}(n^k)$ on S, and by Lemma 15, there is a robust refinement S' of S such that $\Pi_2 SAT \in \Sigma_2\mathsf{TIME}(n^{k+o(1)})$ on S'. Now, by using the proof idea of Theorem 12, we have that there is a robust refinement $S^{(2)}$ of S' such that $\Sigma_4 SAT \in \Sigma_2\mathsf{TIME}(n^{k^3+o(1)})$ on $S^{(2)}$. Using the fact that $\Sigma_4 SAT$ is complete for Σ_4^{P} and the proof idea of Lemma 7, we have that there is a robust refinement $S^{(3)}$ of $S^{(2)}$ such that $\Sigma_4\mathsf{TIME}(n^{3k}) \subseteq \Sigma_2\mathsf{TIME}(n^{3k^4+o(1)})$ on $S^{(3)}$. Now, by completeness of L for $\Sigma_2\mathsf{TIME}(n^{4k^4})$, we have that there is a robust refinement $S^{(4)}$ of $S^{(3)}$ such that $\Sigma_2\mathsf{TIME}(n^{3k^4+o(1)}) \subseteq \mathsf{SIZE}(n^k)$ on $S^{(4)}$. This implies $\Sigma_4\mathsf{TIME}(n^{3k}) \subseteq \mathsf{SIZE}(n^k)$ on $S^{(4)}$, which contradicts Theorem 25 since $S^{(4)}$ is an infinite set. □

Cai and Sengupta (see [Cai07]) show that if NP has polynomial-sized circuits then $\mathsf{PH} = \mathsf{S}_2^{\mathsf{P}}$ which implies via Theorem 25 that $\mathsf{S}_2^{\mathsf{P}}$ does not have fixed-polynomial sized circuits. That argument carries through for robustly often.

Theorem 27. *For any $k > 0$, $\mathsf{S}_2^{\mathsf{P}} \not\subseteq \mathsf{r.o.SIZE}(n^k)$.*

Similarly a lower bound for PP by Vinodchandran [Vin05] also carries through.

Theorem 28. *For any $k > 0$, $\mathsf{PP} \not\subseteq \mathsf{r.o.SIZE}(n^k)$.*

The following strong separation can be shown by direct diagonalization.

Proposition 29. *For any integer $k \geqslant 1$, $\mathsf{DTIME}(2^{n^{2k}}) \not\subseteq \mathsf{i.o.SIZE}(n^k)$.*

We use Proposition 29 to give an analogue of the result of Buhrman, Fortnow and Thierauf [BFT98] that $\mathsf{MAEXP} \not\subseteq \mathsf{SIZE}(\mathsf{poly})$ in the robustly often setting. We give a circuit lower bound for promise problems in MAEXP rather than languages. Intuitively, the fact that MATIME is a semantic measure causes problems in extending the proof of Buhrman, Fortnow and Thierauf to the robustly often setting; however, these problems can be circumvented if we consider promise problems.

Theorem 30. *$\mathsf{Promise} - \mathsf{MAEXP} \not\subseteq \mathsf{r.o.SIZE}(poly)$.*

Proof. Let Q be a promise problem complete for $\mathsf{Promise} - \mathsf{MAE}$ under linear-time reductions. Assume, for the purpose of contradiction, that $\mathsf{Promise} - \mathsf{MAEXP} \subseteq \mathsf{r.o.SIZE}(poly)$. Then there is a robust set S and an integer $k > 0$ such that $Q \in \mathsf{SIZE}(n^k)$ on S. Let L be a language complete for E under linear-time reductions. Since there is a linear-time reduction from L to Q, it follows that there is a robust refinement S' of S such that $L \in \mathsf{SIZE}(n^k)$ on S. Using Lemma 7, there is a robust refinement $S^{(2)}$ of S' such that $L \in \mathsf{MATIME}(n^{2k})$ on $S^{(2)}$. Using a simple padding trick, there is a robust refinement $S^{(3)}$ of $S^{(2)}$ such that for each language $L' \in \mathsf{DTIME}(2^{n^{2k}})$, $L' \in \mathsf{MATIME}(n^{4k^2})$ on $S^{(3)}$. Using completeness of Q again, there is a robust refinement $S^{(4)}$ of $S^{(3)}$ such that $L' \in SIZE(n^k)$ on $S^{(4)}$ for any $L' \in \mathsf{DTIME}(2^{n^{2k}})$, which contradicts Proposition 29. □

Our most technically involved result is that the recent lower bound of Williams [Wil10b] that $\mathsf{NEXP} \not\subseteq \mathsf{ACC}^0$ extends to the robustly often setting. His proof uses the nondeterministic time hierarchy and the proof of Impagliazzo, Kabanets and Wigderson [IKW02], neither of which may hold in the infinitely-often setting. So to get a robustly-often result we require variants of our Theorems 23 and 20. To save space, we will focus on the new ingredients, and abstract out what we need from Williams' paper.

We first need the following simultaneous resource-bounded complexity class.

Definition 31. *$\mathsf{NTIMEGUESS}(T(n), g(n))$ is the class of languages accepted by NTMs running in time $O(T(n))$ and using at most $O(g(n))$ non-deterministic bits.*

We have the following variant of Theorem 23, which has a similar proof.

Lemma 32. *For any constant k, $\mathsf{NTIME}(2^n) \not\subseteq \mathsf{r.o.NTIMEGUESS}(2^n/n, n^k)$.*

We also need the following robustly often analogue of a theorem of Williams [Wil10a], which uses the proof idea of Theorem 20. The problem SUCCINCT3SAT (the satisfiability of an exponentially long 3CNF described by a circuit) is complete for NEXP under polynomial-time m-reductions.

Lemma 33. *If $\mathsf{NE} \subseteq \mathsf{r.o.ACC}^0$ on S for some robust set S, then there is a constant c and a refinement S' of S such that* SUCCINCT3SAT *has succinct satisfying assignments computed by ACC^0 circuits of size n^c on S'.*

Proof. The proof of Theorem 20 gives that if $\mathsf{NE} \subseteq \mathsf{ACC}^0$ on S, then there is a constant d and a robust refinement R of S such that SUCCINCT3SAT has succinct satisfying assignments that are circuits of size n^d on R. Since $\mathsf{P} \subseteq \mathsf{ACC}^0$ on S and using Proposition 11, we get that there is a constant c and a robust refinement S' of R such that SUCCINCT3SAT has succinct satisfying assignments that are ACC^0 circuits of size n^c on S'. □

Now we are ready to prove the robustly often analogue of Williams' main result [Wil10b].

Theorem 34. $\mathsf{NEXP} \not\subseteq \mathsf{r.o.ACC}^0$.

Proof Sketch. Assume, to the contrary, that $\mathsf{SUCCINCT3SAT} \in \mathsf{ACC}^0$ on R for some robust R. By completeness of $\mathsf{SUCCINCT3SAT}$, it follows that there is a robust refinement S of R and a constant $k' > 1$ such that NE has ACC^0 circuits of size $n^{k'}$. Let $L \in \mathsf{NTIME}(2^n)$ but not in $\mathsf{r.o.NTIMEGUESS}(2^n/n, n^k)$, where k will be chosen large enough as a function of k'. Existence of L is guaranteed by Lemma 32. We will show $L \in \mathsf{r.o.NTIMEGUESS}(2^n/n, n^k)$ and obtain a contradiction.

The proof of Theorem 3.2 in Williams' paper gives an algorithm for determinining if $x \in L$. The algorithm non-deterministically guesses and verifies a "small" (size $n^{O(c)}$) ACC^0 circuit which is equivalent to the $\mathsf{SUCCINCT3SAT}$ instance to which x reduces, within time $2^n/\omega(n)$ by using Williams' new algorithm for ACC^0-SAT together with the assumption that NEXP and hence P is in ACC^0 on S. This guess-and-verification procedure works correctly on some robust refinement of S. Then, the algorithm uses the existence guarantee of Lemma 33 to guess and verify a succint witness, again using Williams' algorithm for ACC^0-SAT. This further guess-and-verification procedure works correctly on some further robust refinement $S^{(2)}$ of S. In total, the algorithm uses at most $n^{dk'}$ non-deterministic bits for some constant d, runs in time at most $2^n/n$ and decides L correctly on $S^{(2)}$. By choosing $k > dk'$, we get the desired contradiction. □

Williams' work still leaves open whether $\mathsf{NEXP} \subseteq \mathsf{SIZE}(poly)$. Using the same ideas as in the proof of Theorem 34, we can show that an algorithm for $\mathsf{CircuitSAT}$ that improves slightly on brute force search robustly often would suffice to get such a separation.

Theorem 35. *If for each polynomial p, $\mathsf{CircuitSAT}$ can be solved in time $2^{n-\omega(\log(n))}$ robustly often on instances where the circuit size is at most $p(n)$, then $\mathsf{NEXP} \not\subseteq \mathsf{SIZE}(poly)$.*

4.3 Time-Space Tradeoffs

Fortnow, Lipton, van Melkebeek and Viglas [FLvMV05] showed that SAT cannot be computed by a random-access Turing machine simultaneously using n^α time and polylogarithmic space for any $\alpha < \sqrt{2}$. We show the same result holds for robustly often.

Proposition 36. *Let t and T be time-constructible functions such that $t = o(T)$. Then $\mathsf{NTIME}(T) \not\subseteq \mathsf{i.o.coNTIME}(t)$.*

Theorem 37. *Let $\alpha < \sqrt{2}$ be any constant. $\mathsf{SAT} \notin \mathsf{r.o.DTISP}(n^\alpha, \mathsf{polylog}(n))$.*

Proof. Assume, to the contrary, that $\mathsf{SAT} \in \mathsf{r.o.DTISP}(n^\alpha, \mathsf{polylog}(n))$ on S for some robust set S and constant $\alpha < \sqrt{2}$. Since $\mathsf{DTISP}(n^\alpha, \mathsf{polylog}(n)) \subseteq \Pi_2\mathsf{TIME}(n^{\alpha/2+o(1)})$ [FLvMV05] and using that SAT is complete for $\mathsf{NTIME}(n\ \mathsf{polylog}(n))$ under quasilinear-time length-increasing reductions [Coo88], we have that there is a robust refinement S' of S such that $\mathsf{NTIME}(n) \subseteq \mathsf{DTISP}(n^\alpha, \mathsf{polylog}(n)) \subseteq \Pi_2\mathsf{TIME}(n^{\alpha/2+o(1)})$ on S'. By padding, there is a robust refinement $S^{(2)}$ of S' such that $\mathsf{NTIME}(n^2) \subseteq \Pi_2\mathsf{TIME}(n^{\alpha+o(1)})$ on $S^{(2)}$. By using the assumption that $\mathsf{NTIME}(n) \subseteq \mathsf{DTIME}(n^\alpha)$ on S' again to eliminate the existential quantifier in the Π_2 simulation, we have that there is a robust refinement $S^{(3)}$ of $S^{(2)}$ such that $\mathsf{NTIME}(n^2) \subseteq \mathsf{coNTIME}(n^{\alpha^2+o(1)})$ on $S^{(3)}$. But this is a contradiction to Proposition 36, since $S^{(3)}$ is infinite. □

Similar ideas can be used to show the best known time-space tradeoffs for SAT - the key is that the proofs of these tradeoffs proceed by indirect diagonalization, using the contrary assumption and polynomial padding a constant number of times and deriving a contradiction to a strong hierarchy theorem.

5 Conclusion and Open Problems

Our paper makes significant progress in remedying one of the main issues with proofs using indirect diagonalization - the fact that they don't give almost everywhere separations. We have shown that most interesting results proved using this technique can be strengthened at least to give significant separations.

There are still many separations which we don't know how to make significant: eg., the main results of [BFS09, San07, BFT98]. The reasons why we're unable to get significant separations are different in different cases: the inability to formulate the proof as following from a single assumption about robustly often simulations in the first case, the use of superpolynomial amount of padding in the second, and the fact that complete languages for MAEXP are unknown in the last. It's not clear whether there are inherent limitations to getting significant separations in these cases.

Other variants of robustly often might be interesting to study, such as a "linear" variant where the simulation is only required to hold on arbitrarily large linear stretches of inputs rather than polynomial stretches. Separations against this notion of simulation would be stronger than significant separations.

References

[Bar02] Boaz Barak. A probabilistic-time hierarchy theorem for "Slightly Non-uniform" algorithms. *Lecture Notes in Computer Science*, 2483:194–208, 2002.

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

[BFS09] Harry Buhrman, Lance Fortnow, and Rahul Santhanam. Unconditional lower bounds against advice. In *Proceedings of 36th International Colloquium on Automata, Languages and Programming*, pages 195–209, 2009.

[BFT98] Harry Buhrman, Lance Fortnow, and Thomas Thierauf. Nonrelativizing separations. In *Proceedings of 13th Annual IEEE Conference on Computational Complexity*, pages 8–12, 1998.

[Cai07] Jin-Yi Cai. $S_2^P \subseteq ZPP^{NP}$. *Journal of Computer and System Sciences*, 73(1):25 – 35, 2007.

[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.

[Coo88] S. Cook. Short propositional formulas represent nondeterministic computations. *Information Processing Letters*, 26:269–270, 1988.

[DF03] Rod Downey and Lance Fortnow. Uniformly hard languages. *Theoretical Computer Science*, 298(2):303 – 315, 2003.

[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.

[For00] L. Fortnow. Time-space tradeoffs for satisfiability. *Journal of Computer and System Sciences*, 60(2):337–353, April 2000.

[For15] Lance Fortnow. Nondeterministic separations. In Rahul Jain, Sanjay Jain, and Frank Stephan, editors, *Theory and Applications of Models of Computation*, volume 9076 of *Lecture Notes in Computer Science*, pages 10–17. Springer International Publishing, 2015.

[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.

[FS11] Lance Fortnow and Rahul Santhanam. Robust simulations and significant separations. In Luca Aceto, Monika Henzinger, and Ji Sgall, editors, *Automata, Languages and Programming*, volume 6755 of *Lecture Notes in Computer Science*, pages 569–580. Springer Berlin Heidelberg, 2011.

[Hås86] Johan Håstad. Almost optimal lower bounds for small depth circuits. In *Proceedings of the 18th Annual ACM Symposium on Theory of Computing*, pages 6–20, 1986.

[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.

[IW97] Russell Impagliazzo and Avi Wigderson. P = BPP if E requires exponential circuits: Derandomizing the XOR lemma. In *Proceedings of the 29th Annual ACM Symposium on the Theory of Computing*, pages 220–229, 1997.

[Kab01] Valentine Kabanets. Easiness assumptions and hardness tests: Trading time for zero error. *Journal of Computer and System Sciences*, 63(2):236–252, 2001.

[Kan82] Ravi Kannan. Circuit-size lower bounds and non-reducibility to sparse sets. *Information and Control*, 55(1):40–56, 1982.

[KL82] Richard Karp and Richard Lipton. Turing machines that take advice. *L'Enseignement Mathématique*, 28(2):191–209, 1982.

[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.

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

[Raz85] Alexander Razborov. Lower bounds for the monotone complexity of some boolean functions. *Soviet Mathematics Doklady*, 31:354–357, 1985.

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

[San07] Rahul Santhanam. Circuit lower bounds for Merlin-Arthur classes. In *Proceedings of 39th Annual Symposium on Theory of Computing*, pages 275–283, 2007.

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

[Vin05] Variyam Vinodchandran. A note on the circuit complexity of PP. *Theoretical Computer Science*, 347(1-2):415–418, 2005.

[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.

[Wil10a] Ryan Williams. Improving exhaustive search implies superpolynomial lower bounds. In *Proceedings of the 42nd Annual ACM Symposium on Theory of Computing*, pages 231–240, 2010.

[Wil10b] Ryan Williams. Non-uniform ACC circuit lower bounds. Manuscript, 2010.

[ŽŚ3] Stanislav Žák. A Turing machine time hierarchy. *Theoretical Computer Science*, 26(3):327–333, October 1983.