Electronic Colloquium on Computational Complexity, Report No. 164 (2021)

The Acrobatics of BQP

Scott Aaronson* DeVon Ingram† William Kretschmer‡

Abstract

We show that, in the black-box setting, the behavior of quantum polynomial-time (BQP) can be remarkably decoupled from that of classical complexity classes like NP. Specifically:

- There exists an oracle relative to which $\mathsf{NP}^{\mathsf{BQP}} \not\subset \mathsf{BQP}^{\mathsf{PH}}$, resolving a 2005 problem of Fortnow. Interpreted another way, we show that AC^0 circuits cannot perform useful homomorphic encryption on instances of the Forrelation problem. As a corollary, there exists an oracle relative to which $\mathsf{P} = \mathsf{NP}$ but $\mathsf{BQP} \neq \mathsf{QCMA}$.
- Conversely, there exists an oracle relative to which $\mathsf{BQP}^{\mathsf{NP}} \not\subset \mathsf{PH}^{\mathsf{BQP}}$.
- Relative to a random oracle, $\mathsf{PP} = \mathsf{PostBQP}$ is not contained in the "QMA hierarchy" $\mathsf{QMA}^{\mathsf{QMA}^{\mathsf{QMA}^{\cdots}}}$, and more generally $\mathsf{PP} \not\subset (\mathsf{MIP}^*)^{(\mathsf{MIP}^*)^{(\mathsf{MIP}^*)^{\cdots}}}$ (!), despite the fact that $\mathsf{MIP}^* = \mathsf{RE}$ in the unrelativized world. This result shows that there is no black-box quantum analogue of Stockmeyer's approximate counting algorithm.
- Relative to a random oracle, $\Sigma_{k+1}^{\mathsf{P}} \not\subset \mathsf{BQP}^{\Sigma_k^{\mathsf{P}}}$ for every k.
- There exists an oracle relative to which $\mathsf{BQP} = \mathsf{P}^{\#\mathsf{P}}$ and yet PH is infinite. (By contrast, if $\mathsf{NP} \subseteq \mathsf{BPP}$, then PH collapses relative to all oracles.)
- There exists an oracle relative to which $\mathsf{P} = \mathsf{NP} \neq \mathsf{BQP} = \mathsf{P}^{\#\mathsf{P}}$.

To achieve these results, we build on the 2018 achievement by Raz and Tal of an oracle relative to which $\mathsf{BQP} \not\subset \mathsf{PH}$, and associated results about the Forrelation problem. We also introduce new tools that might be of independent interest. These include a "quantum-aware" version of the random restriction method, a concentration theorem for the block sensitivity of AC^0 circuits, and a (provable) analogue of the Aaronson-Ambainis Conjecture for sparse oracles.

*University of Texas at Austin. Email: scott@scottaaronson.com. Supported by a Vannevar Bush Fellowship from the US Department of Defense, a Simons Investigator Award, and the Simons "It from Qubit" collaboration.
†University of Chicago. Email: dingram@uchicago.edu. Supported by an NSF Graduate Research Fellowship.
‡University of Texas at Austin. Email: kretsch@cs.utexas.edu. Supported by an NDSEG Fellowship.

ISSN 1433-8092

Contents

1 Introduction

The complexity-theoretic study of quantum computation is often dated from 1993, when Bernstein and Vazirani [BV97] defined BQP, or Bounded-Error Quantum Polynomial-Time: the class of languages that admit efficient quantum algorithms. Then as now, a central concern was how BQP relates to classical complexity classes, such as P, NP, and PH. Among the countless questions that one could raise here, let us single out three as especially fundamental:

(1) Can quantum computers efficiently solve any problems that classical computers cannot? In other words, does $\mathsf{BPP} = \mathsf{BQP}$?

(2) Can quantum computers solve NP-complete problems in polynomial time? In other words, is $\mathsf{NP} \subseteq \mathsf{BQP}$?

(3) What is the best classical upper bound on the power of quantum computation? Is $\mathsf{BQP} \subseteq \mathsf{NP}$? Is $\mathsf{BQP} \subseteq \mathsf{PH}$?

Three decades later, all three of these still stand as defining questions of the field. Nevertheless, from the early 2000s onwards, it became rare for work in quantum computing theory to address any of these questions directly, perhaps simply because it became too hard to say anything new about them. A major recent exception was the seminal work of Raz and Tal [RT19], who gave an oracle relative to which $\mathsf{BQP} \not\subset \mathsf{PH}$, by completing a program proposed by one of us [Aar10]. In this paper, we take the Raz-Tal breakthrough as a starting point. Using it, together with new tools that we develop, we manage to prove many new theorems about the power of BQP—at least in the black-box setting where much of our knowledge of quantum algorithms resides.

Before discussing the black-box setting or Raz-Tal, though, let's start by reviewing what is known in general about BQP. Bernstein and Vazirani [BV97] showed that $\mathsf{BPP} \subseteq \mathsf{BQP} \subseteq \mathsf{P}^{\#\mathsf{P}}$, and Adleman, DeMarrais, and Huang [ADH97] improved the upper bound to $\mathsf{BQP} \subseteq \mathsf{PP}$, giving us the following chain of inclusions:

$$\mathsf{P} \subseteq \mathsf{BPP} \subseteq \mathsf{BQP} \subseteq \mathsf{PP} \subseteq \mathsf{P}^{\#\mathsf{P}} \subseteq \mathsf{PSPACE} \subseteq \mathsf{EXP}.$$

Fortnow and Rogers [FR98] slightly strengthened the inclusion $\mathsf{BQP} \subseteq \mathsf{PP}$, to show for example that $\mathsf{PP}^{\mathsf{BQP}} = \mathsf{PP}$. This complemented the result of Bennett, Bernstein, Brassard, and Vazirani [BBBV97] that $\mathsf{BQP}^{\mathsf{BQP}} = \mathsf{BQP}$: that is, BQP is "self-low," or "the BQP hierarchy collapses to BQP."

1.1 The Contrast with BPP

Meanwhile, though, the relationships between BQP and complexity classes like NP, PH, and $\mathsf{P/poly}$ have remained mysterious. Besides the fundamental questions mentioned above—is $\mathsf{NP} \subseteq \mathsf{BQP}$? is $\mathsf{BQP} \subseteq \mathsf{NP}$? is $\mathsf{BQP} \subseteq \mathsf{PH}$?—one could ask other questions:

(i) In a 2005 blog post, Fortnow [For05] raised the question of whether $\mathsf{NP}^{\mathsf{BQP}} \subseteq \mathsf{BQP}^{\mathsf{NP}}$. Do we even have $\mathsf{NP}^{\mathsf{BQP}} \subseteq \mathsf{BQP}^{\mathsf{PH}}$? I.e., when quantum computation is combined with classical nondeterminism, how does the order of combination matter?

(ii) What about the converse: is $\mathsf{BQP}^{\mathsf{NP}} \subseteq \mathsf{PH}^{\mathsf{BQP}}$?

(iii) Suppose $\mathsf{NP} \subseteq \mathsf{BQP}$. Does it follow that $\mathsf{PH} \subseteq \mathsf{BQP}$ as well?

(iv) Suppose $\mathsf{NP} \subseteq \mathsf{BQP}$. Does it follow that PH collapses?

(v) Is $\mathsf{BQP} \subset \mathsf{P/poly}$?

(vi) Suppose $\mathsf{P} = \mathsf{NP}$. Does it follow that BQP is "small" (say, not equal to EXP)?

(vii) Suppose $\mathsf{P} = \mathsf{NP}$. Does it follow that $\mathsf{BQP} = \mathsf{QCMA}$, where QCMA (Quantum Classical Merlin Arthur) is the analogue of NP with a BQP verifier?

What is particularly noteworthy about the questions above is that, if we replace BQP by BPP, then positive answers are known to all of them:

(i) $\mathsf{NP}^{\mathsf{BPP}} \subseteq \mathsf{AM} \subseteq \mathsf{BPP}^{\mathsf{NP}}$.

(ii) $\mathsf{BPP}^{\mathsf{NP}} \subseteq \mathsf{PH} = \mathsf{PH}^{\mathsf{BPP}}$.

(iii) If $\mathsf{NP} \subseteq \mathsf{BPP}$, then $\mathsf{PH} = \mathsf{BPP}$—this is sometimes given as a homework exercise in complexity theory courses, and also follows from (i).

(iv) If $\mathsf{NP} \subseteq \mathsf{BPP}$, then $\mathsf{PH} = \Sigma_2^{\mathsf{P}}$—this follows from (iii) and the Sipser-Lautemann Theorem [Sip83, Lau83].

(v) $\mathsf{BPP} \subset \mathsf{P/poly}$ is Adleman's Theorem [Adl78].

(vi) If $\mathsf{P} = \mathsf{NP}$, then $\mathsf{P} = \mathsf{BPP}$ and hence $\mathsf{BPP} \neq \mathsf{EXP}$, by the time hierarchy theorem.

(vii) If $\mathsf{P} = \mathsf{NP}$, then of course $\mathsf{BPP} = \mathsf{MA}$.

So what is it that distinguishes BPP from BQP in these cases? In all of the above examples, the answer turns out to be one of the fundamental properties of classical randomized algorithms: namely, that one can always "pull the randomness out" from such algorithms, viewing them as simply deterministic algorithms that take a uniform random string r as an auxiliary input, in addition to their "main" input x. This, in turn, enables one to play all sorts of tricks with such an algorithm $M(x, r)$—from using approximate counting to estimate the fraction of r's that cause $M(x, r)$ to accept, to moving r from inside to outside a quantifier, to hardwiring r as advice. By contrast, there is no analogous notion of "pulling the randomness (or quantumness) out of a quantum algorithm." In quantum computation, randomness is just an intrinsic part of the model that rears its head at the *end* (rather than the beginning) of a computation, when we take the squared absolute values of amplitudes to get probabilities.

This difference between randomized and quantum algorithms is crucial to the analysis of the so-called "sampling-based quantum supremacy experiments"—for example, those recently carried out by Google [AAB+19] and USTC [ZWD+20]. The theoretical foundations of these experiments were laid a decade ago, in the work of Aaronson and Arkhipov [AA13] on BosonSampling, and (independently) Bremner, Jozsa, and Shepherd [BJS10] on the commuting Hamiltonians or IQP model. Roughly speaking, the idea is that, by using a quantum computer, one can efficiently sample a probability distribution $\mathcal{D}$ over n-bit strings such that even *estimating* the probabilities of the outcomes is a $\#\mathsf{P}$-hard problem. Meanwhile, though, if there were a polynomial-time classical randomized algorithm $M(x, r)$ to sample from the same distribution $\mathcal{D}$, then one could use the "pulling out r" trick to estimate the probabilities of M's outcomes in PH. But this would put $\mathsf{P}^{\#\mathsf{P}}$ into PH, thereby collapsing PH by Toda's Theorem [Tod91].

More generally, with any of the apparent differences between quantum algorithms and classical randomized algorithms, the question is: how can we prove that the difference is genuine, that no trick will ever be discovered that makes BQP behave more like BPP? For questions like whether

$\mathsf{NP} \subseteq \mathsf{BQP}$ or whether $\mathsf{BQP} \subseteq \mathsf{NP}$, the hard truth here is that not only have we been unable to resolve these questions in the unrelativized world, we've been able to say little more about them than certain "obvious" implications. For example, suppose $\mathsf{NP} \subseteq \mathsf{BQP}$ *and* $\mathsf{BQP} \subseteq \mathsf{AM}$. Then since BQP is closed under complement, we would also have $\mathsf{coNP} \subseteq \mathsf{BQP}$, and hence $\mathsf{coNP} \subseteq \mathsf{AM}$, which is known to imply a collapse of PH [BHZ87]. And thus, if PH is infinite, then either $\mathsf{NP} \not\subset \mathsf{BQP}$ *or* $\mathsf{BQP} \not\subset \mathsf{AM}$. How can we say anything more interesting and nontrivial?

1.2 Relativization

Since the work of Baker, Gill, and Solovay [BGS75], whenever complexity theorists were faced with an impasse like the one above, a central tool has been *relativized* or *black-box* complexity: in other words, studying what happens when all the complexity classes one cares about are fed some specially-constructed oracle. Much like perturbation theory in physics, relativization lets us make well-defined progress even when the original questions we wanted to answer are out of reach. It is well-known that relativization is an imperfect tool—the $\mathsf{IP} = \mathsf{PSPACE}$ [Sha92], $\mathsf{MIP} = \mathsf{NEXP}$ [BFL91], and more recently, $\mathsf{MIP}^* = \mathsf{RE}$ [JNV+20] theorems provide famous examples where complexity classes turned out to be equal, even in the teeth of oracles relative to which they were unequal. On the other hand, so far, almost all such examples have originated from a single source: namely, the use of algebraic techniques in interactive proof systems. And if, for example, we want to understand the consequences of $\mathsf{NP} \subseteq \mathsf{BQP}$, then arguably it makes little sense to search for nonrelativizing consequences if we don't even understand yet what the relativizing consequences (that is, the consequences that hold relative to all oracles) are or are not.

In quantum complexity theory, even more than in classical complexity theory, relativization has been an inextricable part of progress from the very beginning. The likely explanation is that, even when we just count queries to an oracle, in the quantum setting we need to consider algorithms that query all oracle bits in superposition—so that even in the most basic scenarios, it is already unintuitive what can and cannot be done, and so oracle results must do much more than formalize the obvious.

More concretely, Bernstein and Vazirani [BV97] introduced some of the basic techniques of quantum algorithms in order to prove, for the first time, that there exists an oracle A such that $\mathsf{BPP}^A \neq \mathsf{BQP}^A$. Shortly afterward, Simon [Sim97] gave a quantitatively stronger oracle separation between BPP and BQP, and then Shor [Sho99] gave a still stronger separation, along the way to his famous discovery that FACTORING is in BQP.

On the negative side, Bennett, Bernstein, Brassard, and Vazirani [BBBV97] showed that there exists an oracle relative to which $\mathsf{NP} \not\subset \mathsf{BQP}$: indeed, relative to which there are problems that take n time for an NP machine but $\Omega\left(2^{n/2}\right)$ time for a BQP machine. Following the discovery of Grover's algorithm [Gro96], which quantumly searches any list of N items in $O\left(\sqrt{N}\right)$ queries, the result of Bennett, Bernstein, Brassard, and Vazirani gained the interpretation that *Grover's algorithm is optimal*. In other words, any quantum algorithm for NP-complete problems that gets more than the square-root speedup of Grover's algorithm must be "non-black-box." It must exploit the structure of a particular NP-complete problem much like a classical algorithm would have to, rather than treating the problem as just an abstract space of 2^n possible solutions.

Meanwhile, clearly there are oracles relative to which $\mathsf{P} = \mathsf{BQP}$—for example, a PSPACE-complete oracle. But we can ask: would such oracles necessarily collapse the hierarchy of classical complexity classes as well? In a prescient result that provided an early example of the sort of thing we do in this paper, Fortnow and Rogers [FR98] showed that there exists an oracle relative to which $\mathsf{P} = \mathsf{BQP}$ and yet PH is infinite. In other words, if $\mathsf{P} = \mathsf{BQP}$ would imply a collapse of the polynomial hierarchy, then it cannot be for a relativizing reason. Aaronson and Chen [AC17] extended this to show that there exists an oracle relative to which *sampling-based quantum supremacy is impossible*—i.e., any probability distribution approximately samplable in quantum polynomial time is also approximately samplable in classical polynomial time—and yet PH is infinite. In other words, if it is possible to prove the central theoretical conjecture of quantum supremacy—namely, that there are noisy quantum sampling experiments that cannot be simulated in classical polynomial time unless PH collapses—then nonrelativizing techniques will be needed there as well.

What about showing the power of BQP, by giving oracle obstructions to containments like $\mathsf{BQP} \subseteq \mathsf{NP}$, or $\mathsf{BQP} \subseteq \mathsf{PH}$? There, until recently, the progress was much more limited. Watrous [Wat00] showed that there exists an oracle relative to which $\mathsf{BQP} \not\subset \mathsf{NP}$ and even $\mathsf{BQP} \not\subset \mathsf{MA}$ (these separations could also have been shown using the Recursive Fourier Sampling problem, introduced by Bernstein and Vazirani [BV97]). But extending this further, to get an oracle relative to which $\mathsf{BQP} \not\subset \mathsf{PH}$ or even $\mathsf{BQP} \not\subset \mathsf{AM}$, remained an open problem for two decades. Aaronson [Aar10] proposed a program for proving an oracle separation between BQP and PH, involving a new problem he introduced called Forrelation:

Problem 1 (Forrelation)**.** *Given black-box access to two Boolean functions $f, g : \{0,1\}^n \to \{1, -1\}$, and promised that either*

(i) f and g are uniformly random and independent, or

(ii) f and g are uniformly random individually, but g has $\Omega(1)$ correlation with $\hat{f}$, the Boolean Fourier transform of f (i.e., f and g are "Forrelated"),

decide which.

Aaronson [Aar10] showed that Forrelation is solvable, with constant bias, using only a single quantum query to f and g (and $O(n)$ time). By contrast, he showed that any classical randomized algorithm for the problem needs $\Omega\left(2^{n/4}\right)$ queries—improved by Aaronson and Ambainis [AA18] to $\Omega\left(\frac{2^{n/2}}{n}\right)$ queries, which is essentially tight. The central conjecture, which Aaronson left open, said that Forrelation $\notin \mathsf{PH}$—or equivalently, that there are no AC^0 circuits for Forrelation of constant depth and $2^{\text{poly}(n)}$ size.

Finally, Raz and Tal [RT19] managed to prove Aaronson's conjecture, and thereby obtain the long-sought oracle separation between BQP and PH.[1] Raz and Tal achieved this by introducing new techniques for constant-depth circuit lower bounds, involving Brownian motion and the L_1-weight of the low-order Fourier coefficients of AC^0 functions. Relevantly for us, Raz and Tal actually proved the following stronger result:

Theorem 2 ([RT19])**.** *A PH machine can guess whether f and g are uniform or Forrelated with bias at most $2^{-\Omega(n)}$.*

Recall that before Raz and Tal, we did not even have an oracle relative to which $\mathsf{BQP} \not\subset \mathsf{AM}$. Notice that, if $\mathsf{BQP} \subseteq \mathsf{AM}$, then many other conclusions would follow in a relativizing way. For example, we would have:

- $\mathsf{P} = \mathsf{NP}$ implies $\mathsf{P} = \mathsf{BQP}$,
- $\mathsf{NP}^{\mathsf{BQP}} \subseteq \mathsf{NP}^{\mathsf{AM} \cap \mathsf{coAM}} \subseteq \mathsf{BPP}^{\mathsf{NP}} \subseteq \mathsf{BQP}^{\mathsf{NP}}$,

[1] Strictly speaking, they did this for a variant of Forrelation where the correlation between g and $\hat{f}$ is only $\sim \frac{1}{n}$, and thus a quantum algorithm needs $\sim n$ queries to solve the problem, but this will not affect anything that follows.

- If $\mathsf{NP} \subseteq \mathsf{BQP}$, then $\mathsf{NP}^{\mathsf{NP}} \subseteq \mathsf{NP}^{\mathsf{BQP}} \subseteq \mathsf{BQP}^{\mathsf{NP}} = \mathsf{BQP}^{\mathsf{BQP}} = \mathsf{BQP}$, and
- If $\mathsf{NP} \subseteq \mathsf{BQP}$, then $\mathsf{NP} \subseteq \mathsf{coAM}$, which implies that PH collapses.

Looking at it a different way, our inability even to separate BQP from AM by an oracle served as an obstruction to numerous other oracle separations.

The starting point of this paper was the following question: in a "post-Raz-Tal world," can we at last completely "unshackle" BQP from P, NP, and PH, by showing that there are no relativizing obstructions to any possible answers to questions like the ones we asked in Section 1.1?

1.3 Our Results

We achieve new oracle separations that show an astonishing range of possible behaviors for BQP and related complexity classes—in at least one case, resolving a longstanding open problem in this topic. Our title, "The Acrobatics of BQP," comes from a unifying theme of the new results being "freedom." We will show that, as far as relativizing techniques can detect, collapses and separations of classical complexity classes place surprisingly few constraints on the power of quantum computation.

As we alluded to earlier, many of our new results would not have been possible without Raz and Tal's analysis of FORRELATION [RT19], which we rely on extensively. We will treat FORRELATION no longer as just an isolated problem, but as a sort of cryptographic code, by which an oracle can systematically make certain information available to BQP machines while keeping the information hidden from classical machines.

Having said that, very few of our results will follow from Raz-Tal in any straightforward way. Most often we need to develop other lower bound tools, in addition to or instead of Raz-Tal. Our new tools, which seem likely to be of independent interest, include a random restriction lemma for quantum query algorithms, a concentration theorem for the block sensitivity of AC^0 functions, and a provable analogue of the Aaronson-Ambainis conjecture [AA14] for certain sparse oracles.

Perhaps our single most interesting result is the following.

Theorem 3 (Corollary 43, restated)**.** *There exists an oracle relative to which $\mathsf{NP}^{\mathsf{BQP}} \not\subset \mathsf{BQP}^{\mathsf{NP}}$, and indeed $\mathsf{NP}^{\mathsf{BQP}} \not\subset \mathsf{BQP}^{\mathsf{PH}}$.*

As mentioned earlier, Theorem 3 resolves an open problem of Fortnow [For05], and demonstrates another clear difference between BPP and BQP.

As a straightforward byproduct of Theorem 3, we are also able to prove the following:

Theorem 4 (Corollary 45, restated)**.** *There exists an oracle relative to which $\mathsf{P} = \mathsf{NP}$ but $\mathsf{BQP} \neq \mathsf{QCMA}$.*

Conversely, it will follow from one of our later results, Theorem 9, that there exists an oracle relative to which $\mathsf{P} \neq \mathsf{NP}$ and yet $\mathsf{BQP} = \mathsf{QCMA} = \mathsf{QMA}$. In other words, as far as relativizing techniques are concerned, the classical and quantum versions of the P vs. NP question are completely uncoupled from one another.

Theorem 3 also represents progress toward a proof of the following conjecture, which might be the most alluring open problem that we leave.

Conjecture 5. *There exists an oracle relative to which $\mathsf{NP} \subseteq \mathsf{BQP}$ but $\mathsf{PH} \not\subset \mathsf{BQP}$. Indeed, for every $k \in \mathbb{N}$, there exists an oracle relative to which $\Sigma_k^{\mathsf{P}} \subseteq \mathsf{BQP}$ but $\Sigma_{k+1}^{\mathsf{P}} \not\subset \mathsf{BQP}$.*

Conjecture 5 would provide spectacularly fine control over the relationship between BQP and PH, going far beyond Raz-Tal to show how BQP could, e.g., swallow the first 18 levels of PH without swallowing the 19th. To see the connection between Theorem 3 and Conjecture 5, suppose $\mathsf{NP}^{\mathsf{BQP}} \subseteq \mathsf{BQP}^{\mathsf{NP}}$, and suppose also that $\mathsf{NP} \subseteq \mathsf{BQP}$. Then, as observed by Fortnow [For05], this would imply

$$\mathsf{NP}^{\mathsf{NP}} \subseteq \mathsf{NP}^{\mathsf{BQP}} \subseteq \mathsf{BQP}^{\mathsf{NP}} \subseteq \mathsf{BQP}^{\mathsf{BQP}} = \mathsf{BQP},$$

(and so on, for all higher levels of PH), so that $\mathsf{PH} \subseteq \mathsf{BQP}$ as well. Hence, any oracle that witnesses Conjecture 5 also witnesses Theorem 3, so our proof of Theorem 3 is indeed a prerequisite to Conjecture 5.

Our proof of Theorem 3 also reveals additional remarkable structure (or a lack thereof) involving the relation between the FORRELATION problem and AC^0 circuits. At a high level, we prove Theorem 3 by showing that no $\mathsf{BQP}^{\mathsf{PH}}$ algorithm can solve the OR $\circ$ FORRELATION problem, in which one is given a long list of FORRELATION instances, and is tasked with distinguishing whether (1) all of the instances are uniformly random, or (2) at least one of the instances is Forrelated.

A first intuition is that PH machines should gain no useful information from the input, just because FORRELATION "looks random" (by Raz-Tal), and hence a $\mathsf{BQP}^{\mathsf{PH}}$ machine should have roughly the same power as a BQP machine at deciding OR $\circ$ FORRELATION. If one could show this, then completing the theorem would amount to showing that OR $\circ$ FORRELATION is hard for BQP machines, which easily follows from the BBBV Theorem [BBBV97].

Alas, initial attempts to formalize this intuition fail for a single, crucial reason: homomorphic encryption exists! The Raz-Tal Theorem merely proves that FORRELATION is a strong form of encryption against PH algorithms. But to rule out a $\mathsf{BQP}^{\mathsf{PH}}$ algorithm for OR $\circ$ FORRELATION, we *also* have to show that a collection of PH algorithms cannot take a collection of FORRELATION instances, and transform them into a single FORRELATION instance whose solution is the OR of the solutions to the input instances. Put another way, we must show that AC^0 circuits of constant depth and $2^{\mathrm{poly}(n)}$ size cannot homomorphically evaluate the OR function, when the encryption is done via the FORRELATION problem.

More generally, we even have to show that AC^0 circuits cannot transform the ciphertext into *any* string that could later be decoded by an efficient quantum algorithm. Theorem 3 accomplishes this with the help of an additional structural property of AC^0 circuits: our concentration theorem for block sensitivity. Loosely speaking, the concentration theorem implies that, with overwhelming probability, any small AC^0 circuit is insensitive to toggling between a yes-instance and a neighboring no-instance of the OR $\circ$ FORRELATION problem. This, together with the BBBV Theorem [BBBV97], then implies that such "homomorphic encryption" is impossible.

We also achieve the following converse to Theorem 3:

Theorem 6 (Corollary 51, restated)**.** *There exists an oracle relative to which* $\mathsf{BQP}^{\mathsf{NP}} \not\subset \mathsf{PH}^{\mathsf{BQP}}$*, and even* $\mathsf{BQP}^{\mathsf{NP}} \not\subset \mathsf{PH}^{\mathsf{PromiseBQP}}$*.*

Note that an oracle relative to which $\mathsf{BQP}^{\mathsf{NP}} \not\subset \mathsf{NP}^{\mathsf{BQP}}$ is almost trivial to achieve, for example by considering any problem in coNP. However, $\mathsf{BQP}^{\mathsf{NP}} \not\subset \mathsf{PH}^{\mathsf{BQP}}$ is much harder. At a high level, rather than considering the composed problem OR $\circ$ FORRELATION, we now need to consider the reverse composition: FORRELATION $\circ$ OR, a problem that's clearly in $\mathsf{BQP}^{\mathsf{NP}}$, but plausibly not in $\mathsf{PH}^{\mathsf{BQP}}$. The key step is to show that, when solving FORRELATION $\circ$ OR, *any* $\mathsf{PH}^{\mathsf{BQP}}$ *machine can be simulated by a* PH *machine*: the BQP oracle is completely superfluous! Once we've shown that, FORRELATION $\circ$ OR $\notin \mathsf{PH}$ then follows immediately from Raz-Tal.

For our next result, recall that QMA, or *Quantum Merlin-Arthur*, is the class of problems for which a yes-answer can be witnessed by a polynomial-size quantum state. Perhaps our second most interesting result is this:

Theorem 7 (Corollary 63, restated)**.** PP *is not contained in the* "QMA hierarchy"*, consisting of constant-depth towers of the form* $\mathsf{QMA}^{\mathsf{QMA}^{\mathsf{QMA}^{\cdots}}}$*, with probability* 1 *relative to a random oracle.*[2]

Before this paper, to our knowledge, it was not even known how to construct an oracle relative to which $\mathsf{PP} \not\subset \mathsf{BQP}^{\mathsf{NP}}$, let alone classes like $\mathsf{BQP}^{\mathsf{NP}^{\mathsf{BQP}^{\mathsf{NP}^{\cdots}}}}$ or $\mathsf{QCMA}^{\mathsf{QCMA}^{\mathsf{QCMA}^{\cdots}}}$, which are contained in the QMA hierarchy. The closest result we are aware of is due to Kretschmer [Kre21], who gave a *quantum* oracle relative to which $\mathsf{BQP} = \mathsf{QMA} \neq \mathsf{PostBQP}$.

Perhaps shockingly, our proof of Theorem 7 can be extended even to show that PP is not in, say, $(\mathsf{MIP}^*)^{(\mathsf{MIP}^*)^{(\mathsf{MIP}^*)^{\cdots}}}$ relative to a random oracle, where MIP^* means multi-prover interactive proofs with entangled provers. This is despite the recent breakthrough of Ji, Natarajan, Vidick, Wright, and Yuen [JNV+20], which showed that in the *unrelativized* world, $\mathsf{MIP}^* = \mathsf{RE}$ (where RE means Recursively Enumerable), so in particular, MIP^* contains the halting problem. This underscores the dramatic extent to which results like $\mathsf{MIP}^* = \mathsf{RE}$ are nonrelativizing!

We mention another interesting interpretation of Theorem 7: it can be understood as showing that in the black box setting, there is no quantum analogue of Stockmeyer's approximate counting algorithm [Sto83]. For a probabilistic machine M that runs in $\text{poly}(n)$ time and an error bound $\varepsilon \geq \frac{1}{\text{poly}(n)}$, the approximate counting problem is to estimate the acceptance probability of M up to a multiplicative factor of $1 + \varepsilon$. Stockmeyer's algorithm [Sto83] gives a $\text{poly}(n)$-time reduction from the approximate counting problem to a problem in the third level of the polynomial hierarchy, and moreover, this reduction relativizes.

One might wonder: is there a version of Stockmeyer's algorithm for the *quantum* approximate counting problem, where we instead wish to approximate the acceptance probability of a quantum algorithm?[3] By well-known reductions involving Aaronson's postselection theorem [Aar05], all problems in $\mathsf{PP} = \mathsf{PostBQP}$ efficiently reduce to the quantum approximate counting problem (see e.g. [Kup15, Proposition 2.14]). Hence, Theorem 7 implies that the quantum approximate counting problem cannot reduce efficiently to the QMA hierarchy in a black box manner, giving some sense in which a quantum analogue of Stockmeyer's algorithm is impossible.

Notably, our proof of Theorem 7 does not appeal to Raz-Tal at all, but instead relies on a new random restriction lemma for the acceptance probabilities of quantum query algorithms. Our random restriction lemma shows that if one randomly fixes most of the inputs to a quantum query algorithm, then the algorithm's behavior on the unrestricted inputs can be approximated by a "simple" function (say, a small decision tree or small DNF formula). We then use this random restriction lemma to generalize the usual random restriction proof that, for example, PARITY $\notin \mathsf{AC}^0$ [Hås87].

Here is another noteworthy result that we are able to obtain, by combining random restriction arguments with lower bounds on quantum query complexity:

Theorem 8 (Corollary 36, restated)**.** *For every* $k \in \mathbb{N}$*,* $\Sigma_{k+1}^{\mathsf{P}} \not\subset \mathsf{BQP}^{\Sigma_k^{\mathsf{P}}}$ *with probability* 1 *relative to a random oracle.*

[2]Actually, our formal definition of the QMA hierarchy is more general than the version given here, in order to accommodate recursive queries to QMA promise problems. This only makes our separation stronger. See Section 2.2 for details.

[3]We thank Patrick Rall (personal communication) for bringing this question to our attention.

Theorem 8 extends the breakthrough of Håstad, Rossman, Servedio, and Tan [HRST17], who (solving an open problem from the 1980s) showed that PH is infinite relative to a random oracle with probability 1. Our result shows, not only that a random oracle creates a gap between every two successive levels of PH, but that quantum computing fails to bridge that gap.

Again, Theorem 8 represents a necessary step toward a proof of Conjecture 5, because if we had $\Sigma_{k+1}^{\mathsf{P}} \subseteq \mathsf{BQP}^{\Sigma_k^{\mathsf{P}}}$, then clearly $\Sigma_k^{\mathsf{P}} \subseteq \mathsf{BQP}$ would imply $\Sigma_{k+1}^{\mathsf{P}} \subseteq \mathsf{BQP}^{\mathsf{BQP}} = \mathsf{BQP}$.

Our last two theorems return to the theme of the autonomy of BQP.

Theorem 9 (Theorem 26, restated)**.** *There exists an oracle relative to which* $\mathsf{NP} \subseteq \mathsf{BQP}$*, and indeed* $\mathsf{BQP} = \mathsf{P}^{\#\mathsf{P}}$*, and yet* PH *is infinite.*

As a simple corollary (Corollary 28), we also obtain an oracle relative to which $\mathsf{BQP} \not\subset \mathsf{NP}/\mathsf{poly}$.

For three decades, one of the great questions of quantum computation has been whether it can solve NP-complete problems in polynomial time. Many experts guess that the answer is no, for similar reasons as they guess that $\mathsf{P} \neq \mathsf{NP}$—say, the BBBV Theorem [BBBV97], combined with our failure to find any promising leads for evading that theorem's assumptions in the worst case. But the fact remains that we have no structural evidence connecting the $\mathsf{NP} \not\subset \mathsf{BQP}$ conjecture to any "pre-quantum" beliefs about complexity classes. No one has any idea how to show, for example, that if $\mathsf{NP} \subseteq \mathsf{BQP}$ then $\mathsf{P} = \mathsf{NP}$ as well, or anything even remotely in that direction.

Given the experience of classical complexity theory, it would be reasonable to hope for a theorem showing that, if $\mathsf{NP} \subseteq \mathsf{BQP}$, then PH collapses—analogous to the Karp-Lipton Theorem [KL80], that if $\mathsf{NP} \subset \mathsf{P}/\mathsf{poly}$ then PH collapses, or the Boppana-Håstad-Zachos Theorem [BHZ87], that if $\mathsf{NP} \subseteq \mathsf{coAM}$ then PH collapses. But again, no such result is known for $\mathsf{NP} \subseteq \mathsf{BQP}$. Theorem 9 helps to explain this situation, by showing that any proof of such a conditional collapse would have to be nonrelativizing. The proof of Theorem 9 builds, again, on the Raz-Tal Theorem. And this is easily seen to be necessary, since as we pointed out earlier, if $\mathsf{BQP} \subseteq \mathsf{AM}$, then $\mathsf{NP} \subseteq \mathsf{BQP}$ really *would* imply a collapse of PH.

Theorem 10 (Theorem 29, restated)**.** *There exists an oracle relative to which* $\mathsf{P} = \mathsf{NP} \neq \mathsf{BQP} = \mathsf{P}^{\#\mathsf{P}}$*.*

Theorem 10 says, in effect, that there is no relativizing obstruction to BQP being inordinately powerful even while NP is inordinately weak. It substantially extends the Raz-Tal Theorem, that there is an oracle relative to which $\mathsf{BQP} \not\subset \mathsf{PH}$, to show that in some oracle worlds, BQP doesn't go just *slightly* beyond the power of PH (which, if $\mathsf{P} = \mathsf{NP}$, is simply the power of P), but *vastly* beyond it. We conjecture that Theorem 10 could be extended yet further, to give an oracle relative to which $\mathsf{P} = \mathsf{NP}$ and yet $\mathsf{BQP} = \mathsf{EXP}$, but we leave that problem to future work.

1.4 Proof Techniques

We now give rough sketches of the important ideas needed to prove our results. Here, in contrast to Section 1.3, we present the results in the order that they appear in the main text, which is roughly in order of increasing technical difficulty.

Our proofs of Theorem 9 and Theorem 10 serve as useful warm-ups, giving a flavor for how we use the Raz-Tal Theorem and oracle construction techniques in later proofs. In Theorem 9, to construct an oracle where $\mathsf{BQP} = \mathsf{P}^{\#\mathsf{P}}$ but PH is infinite, we start by taking a random oracle, which by the work of Håstad, Rossman, Servedio, and Tan [HRST17, RST15] is known to make PH infinite. Then, we recursively add into the oracle the answers to all possible $\mathsf{P}^{\#\mathsf{P}}$ queries, hiding the answers in instances of the Forrelation problem. This gives a BQP machine the power to decide any $\mathsf{P}^{\#\mathsf{P}}$ language.

It remains to argue that adding these Forrelation instances does not collapse PH. We want to show that relative to our oracle, for every k, there exists a language in $\Sigma_{k+1}^{\mathsf{P}}$ that is not in Σ_k^{P}. This is where we leverage the Raz-Tal Theorem: because the Forrelation instances look random to PH, we can show, by a hybrid argument, that a Σ_k^{P} algorithm's probability of correctly deciding a target function in $\Sigma_{k+1}^{\mathsf{P}}$ is roughly unchanged if we replace the Forrelation instances with uncorrelated, uniformly random bits. But auxiliary random bits cannot possibly improve the success probability, and so a simple appeal to [HRST17] implies that the $\Sigma_{k+1}^{\mathsf{P}}$ language remains hard for Σ_k^{P}.

The proof of Theorem 10, giving an oracle where $\mathsf{P} = \mathsf{NP} \neq \mathsf{BQP} = \mathsf{P}^{\#\mathsf{P}}$, follows a similar recipe to the proof of Theorem 9. We start with a random oracle, which separates PH from $\mathsf{P}^{\#\mathsf{P}}$, and then we add a second region of the oracle that puts $\mathsf{P}^{\#\mathsf{P}}$ into BQP by encoding all $\mathsf{P}^{\#\mathsf{P}}$ queries in instances of the Forrelation problem. Next, we add a third region of the oracle that answers all NP queries, which has the effect of collapsing PH to P. Finally, we again leverage the Raz-Tal Theorem to argue that the Forrelation instances have no effect on the separation between PH and $\mathsf{P}^{\#\mathsf{P}}$, because the Forrelation instances look random to PH algorithms.

We next prove Theorem 8, that $\Sigma_{k+1}^{\mathsf{P}} \not\subset \mathsf{BQP}^{\Sigma_k^{\mathsf{P}}}$ relative to a random oracle. Our proof builds heavily on the proof by [HRST17] that $\Sigma_{k+1}^{\mathsf{P}} \not\subset \Sigma_k^{\mathsf{P}}$ relative to a random oracle. Indeed, our proof is virtually identical, except for a single additional step.

[HRST17]'s proof involves showing that there exists a function $\textsc{Sipser}_d$ that is computable by a small AC^0 circuit of depth d (which corresponds to a $\Sigma_{d-1}^{\mathsf{P}}$ algorithm), but such that any small AC^0 circuit of depth $d-1$ (which corresponds to a $\Sigma_{d-2}^{\mathsf{P}}$ algorithm) computes $\textsc{Sipser}_d$ on at most a $\frac{1}{2} + o(1)$ fraction of random inputs. This proof uses random restrictions, or more accurately, a generalization of random restrictions called *random projections* by [HRST17]. Roughly speaking, the proof constructs a distribution $\mathcal{R}$ over random projections with the following properties:

(i) Any small AC^0 circuit C of depth $d-1$ "simplifies" with high probability under a random projection drawn from $\mathcal{R}$, say, by collapsing to a low-depth decision tree.

(ii) The target $\textsc{Sipser}_d$ function "retains structure" with high probability under a random projection drawn from $\mathcal{R}$.

(iii) The structure retained in (ii) implies that the original unrestricted circuit C fails to compute the $\textsc{Sipser}_d$ function on a large fraction of inputs.

To prove Theorem 8, we generalize step (i) above from $\Sigma_{d-2}^{\mathsf{P}}$ algorithms to $\mathsf{BQP}^{\Sigma_{d-2}^{\mathsf{P}}}$ algorithms. That is, if we have a quantum algorithm that queries arbitrary depth-$(d-1)$ AC^0 functions of the input, then we show that this algorithm's acceptance probability also "simplifies" under a random projection from $\mathcal{R}$. We prove this by combining the BBBV Theorem [BBBV97] with [HRST17]'s proof of step (i).

We next move on to the proof of Theorem 3, where we construct an oracle relative to which $\mathsf{NP}^{\mathsf{BQP}} \not\subset \mathsf{BQP}^{\mathsf{PH}}$. Recall that we prove Theorem 3 by showing that no $\mathsf{BQP}^{\mathsf{PH}}$ algorithm can solve the $\mathrm{OR} \circ \textsc{Forrelation}$ problem. To establish this, imagine that we fix a "no" instance x of the $\mathrm{OR} \circ \textsc{Forrelation}$ problem, meaning that x consists of a list of $\sim 2^n$ Forrelation instances that are all uniformly random (i.e. non-Forrelated). We can turn x into an adjacent "yes" instance y by randomly choosing one of the Forrelation instances of x and changing it to be Forrelated.

Our proof amounts to showing that with high probability over x, an $\mathsf{AC^0}$ circuit of size $2^{\text{poly}(n)}$ is unlikely (over y) to distinguish x from y. Then, applying the BBBV Theorem [BBBV97], we can show that for most choices of x, a $\mathsf{BQP^{PH}}$ algorithm is unlikely to distinguish x from y, implying that it could not have solved the OR ∘ Forrelation problem.

Next, we notice that it suffices to consider what happens when, instead of choosing y by randomly flipping one of the Forrelation instances of x from uniformly random to Forrelated, we instead choose a string z by randomly resampling one if the instances of x from the uniform distribution. This is because, as a straightforward consequence of the Raz-Tal Theorem (Theorem 2), if f is an $\mathsf{AC^0}$ circuit of size $2^{\text{poly}(n)}$, then $|\Pr_y[f(x) \neq f(y)] - \Pr_z[f(x) \neq f(z)]| \leq 2^{-\Omega(n)}$.

Our key observation is that the quantity $\Pr_z[f(x) \neq f(z)]$ is proportional to a sort of "block sensitivity" of f on x. More precisely, it is proportional to an appropriate averaged notion of block sensitivity, where the average is taken over collections of blocks that respect the partition into separate Forrelation instances. This is where our block sensitivity concentration theorem comes into play:

Theorem 11 (Corollary 39, informal)**.** *Let $f : \{0,1\}^N \to \{0,1\}$ be an $\mathsf{AC^0}$ circuit of size* quasipoly(N) *and depth $O(1)$, and let $B = \{B_1, B_2, \ldots, B_k\}$ be a collection of disjoint subsets of $[N]$. Then for any t,*

$$\Pr_{x \sim \{0,1\}^N}[\mathsf{bs}_B^x(f) \geq t] \leq 4N \cdot 2^{-\Omega\left(\frac{t}{\text{polylog}(N)}\right)},$$

where $\mathsf{bs}_B^x(f)$ denotes the block sensitivity of f on x with respect to B.

Informally, Theorem 11 says that the probability that an $\mathsf{AC^0}$ circuit has B-block sensitivity $t \gg \text{polylog}(N)$ on a random input x decays exponentially in t. This generalizes the result of Linial, Mansour, and Nisan [LMN93] that the *average* sensitivity of $\mathsf{AC^0}$ circuits is at most polylog(N). It also generalizes a concentration theorem for the sensitivity of $\mathsf{AC^0}$ circuits that appeared implicitly in the work of Gopalan, Servedio, Tal, and Wigderson [GSTW16], by taking B to be the partition into singletons.[4] In fact, we derive Theorem 11 as a simple corollary of such a sensitivity tail bound for $\mathsf{AC^0}$. For completeness, we will also prove our own sensitivity tail bound, rather than appealing to [GSTW16]. Our sensitivity tail bound follows from an $\mathsf{AC^0}$ random restriction lemma due to Rossman [Ros17].

To prove Theorem 4, which gives an oracle relative to which $\mathsf{P} = \mathsf{NP}$ but $\mathsf{BQP} \neq \mathsf{QCMA}$, we use a similar technique to the proof of Theorem 10. We first take the oracle constructed in Theorem 3 that contains instances of the OR ∘ Forrelation problem. Next, we add a second region of the oracle that answers all NP queries. This collapses PH to P. Finally, we use Theorem 3 to argue that these NP queries do not enable a BQP algorithm to solve the OR ∘ Forrelation problem, which is in QCMA.

We now move on to the proof of Theorem 6, that there exists an oracle relative to which $\mathsf{BQP^{NP}} \not\subset \mathsf{PH^{BQP}}$. Recall that our strategy is to show that no $\mathsf{PH^{BQP}}$ algorithm can solve the Forrelation ∘ OR problem. We prove this by showing that with high probability, a $\mathsf{PH^{BQP}}$ algorithm on a random instance of the Forrelation ∘ OR problem can be simulated by a PH algorithm, from which a lower bound easily follows from the Raz-Tal Theorem. This simulation hinges on the following theorem, which seems very likely to be of independent interest:

[4] Interestingly, [GSTW16]'s goal, in proving their concentration theorem for the sensitivity of $\mathsf{AC^0}$, was to make progress toward a proof of the famous *Sensitivity Conjecture*—a goal that Huang [Hua19] achieved shortly afterward using completely different methods. One happy corollary of this work is that, nevertheless, [GSTW16]'s attempt on the problem was not entirely in vain.

Theorem 12 (Theorem 48, informal)**.** *Consider a quantum algorithm Q that makes T queries to an $M \times N$ array of bits x, where each length-N row of x contains a single uniformly random 1 and 0s everywhere else. Then for any $\varepsilon \gg \frac{T}{\sqrt{N}}$ and $\delta > 0$, there exists a deterministic classical algorithm that makes $O\left(\frac{T^5}{\varepsilon^4} \log \frac{T}{\delta}\right)$ queries to x, and approximates Q's acceptance probability to within additive error ε on a $1 - \delta$ fraction of such randomly chosen x's.*

Informally, Theorem 12 says that any fast enough quantum algorithm can be simulated by a deterministic classical algorithm, with at most a polynomial blowup in query complexity, on almost all sufficiently sparse oracles. The crucial point here is that the classical simulation still needs to work, even in most cases where the quantum algorithm is lucky enough to find many '1' bits. We prove Theorem 12 via a combination of tail bounds and the BBBV hybrid argument [BBBV97].

In the statement of Theorem 12, we do not know whether the exponent of 5 on T is tight, and suspect that it isn't. We only know that the exponent needs to be at least 2, because of Grover's algorithm [Gro96].

We remark that Theorem 12 bears similarity to a well-known conjecture that involves simulation of quantum query algorithms by classical algorithms. A decade ago, motivated by the question of whether $\mathsf{P} = \mathsf{BQP}$ relative to a random oracle with probability 1, Aaronson and Ambainis [AA14] proposed the following conjecture:

Conjecture 13 ([AA14, Conjecture 1.5]; attributed to folklore)**.** *Consider a quantum algorithm Q that makes T queries to $x \in \{0,1\}^N$. Then for any $\varepsilon, \delta > 0$, there exists a deterministic classical algorithm that makes $\operatorname{poly}\left(T, \frac{1}{\varepsilon}, \frac{1}{\delta}\right)$ queries to x, and approximates Q's acceptance probability to within additive error ε on a $1 - \delta$ fraction of uniformly randomly inputs x.*

While Conjecture 13 has become influential in Fourier analysis of Boolean functions,[5] it remains open to this day. Theorem 12 could be seen as *the analogue of Conjecture 13 for sparse oracles*—an analogue that, because of the sparseness, turns out to be much easier to prove.

We conclude with the proof of Theorem 7, showing that PP is not contained in the QMA hierarchy relative to a random oracle. This is arguably the most technically involved part of this work. Recall that our key contribution, and the most important step of our proof, is a random restriction lemma for quantum query algorithms. In fact, we even prove a random restriction lemma for functions with low *quantum Merlin-Arthur* (QMA) *query complexity*: that is, functions f where a verifier, given an arbitrarily long "witness state," can become convinced that $f(x) = 1$ by making few queries to x. Notably, our definition of QMA query complexity does not care about the length of the witness, but only on the number of queries made by the verifier. This property allows us to extend our results to complexity classes beyond QMA, such as MIP^*.

An informal statement of our random restriction lemma is given below:

Theorem 14 (Theorem 57, informal)**.** *Consider a partial function $f : \{0,1\}^N \to \{0,1,\perp\}$ with QMA query complexity at most $\operatorname{polylog}(N)$. For some $p = \frac{1}{\sqrt{N}\operatorname{polylog}(N)}$, let ρ be a random restriction that leaves each variable unrestricted with probability p. Then f_ρ is $\frac{1}{\operatorname{quasipoly}(N)}$-close, in expectation over ρ, to a $\operatorname{polylog}(N)$-width DNF formula.*[6]

[5] In the context of Fourier analysis, the Aaronson-Ambainis Conjecture usually refers to a closely-related conjecture about influences of bounded low-degree polynomials; see e.g. [Mon12, OZ16]. Aaronson and Ambainis [AA14] showed that this related conjecture implies Conjecture 13.

[6] By saying that f_ρ is "close" to a DNF formula, we mean that there exists a DNF g depending on ρ such that the fraction of inputs on which f_ρ and g agree is $1 - \frac{1}{\operatorname{quasipoly}(N)}$, in expectation over ρ. In Section 5.2, we introduce some additional notation and terminology that makes it easier to manipulate such expressions, but we will not use them in this exposition.

An unusual feature of Theorem 14 is that we can only show that f_ρ is *close* to a simple function *in expectation*. By contrast, Håstad's switching lemma for DNF formulas [Hås87] shows that the restricted function reduces to a simple function *with high probability*, so in some sense our result is weaker. Additionally, unlike the switching lemma, our result has a quantitative dependence on the number of inputs N. Whether this dependence can be removed (so that the bound depends only on the number of queries) remains an interesting problem for future work.

With Theorem 14 in hand, proving that $\mathsf{PP} \not\subset \mathsf{QMA}^{\mathsf{QMA}^{\mathsf{QMA}^{\cdots}}}$ relative to a random oracle is conceptually analogous to the proof that $\mathsf{PP} \not\subset \mathsf{PH}$ relative to a random oracle [Hås87]. We first view a $\mathsf{QMA}^{\mathsf{QMA}^{\mathsf{QMA}^{\cdots}}}$ algorithm as a small constant-depth circuit in which the gates are functions of low QMA query complexity. Then we want to argue that the probability that such a circuit agrees with the Parity function on a random input is small. We accomplish this via repeated application of Theorem 14, interleaved with Håstad's switching lemma for DNF formulas [Hås87].

To elaborate further, we first take a random restriction that, by Theorem 14, turns all of the bottom-layer QMA gates into DNF formulas. Next, we apply another random restriction and appeal to the switching lemma to argue that these DNFs reduce to functions of low decision tree complexity, which can be absorbed into the next layer of QMA gates. Finally, we repeat as many times as needed until the entire circuit collapses to a low-depth decision tree. Since the Parity function reduces to another Parity function under any random restriction, we conclude that this decision tree will disagree with the reduced Parity function on a large fraction of inputs, and hence the original circuit must have disagreed with the Parity function on a large fraction of inputs as well.

Of course, the actual proof of Theorem 7 is more complicated because of the accounting needed to bound the error introduced from Theorem 14, but all of the important concepts are captured above.

We end with a few remarks on the proof ideas needed for Theorem 14. Essentially, the first step involves proving that if we take a function f computed by a quantum query algorithm Q, a random restriction ρ, and a uniformly random input x to f_ρ, then x likely contains a small set K of "influential" variables. These influential variables have the property that for any string y that agrees with x on K, $|\Pr[Q(x) = 1] - \Pr[Q(y) = 1]|$ is bounded by a small constant. Hence, K serves as a certificate for f_ρ's behavior on x.

Proving that such a K usually exists amounts to a careful application of the BBBV Theorem [BBBV97]; the reader may find the details in Theorem 52. Finally, we generalize from quantum query algorithms to arbitrary QMA query algorithms by observing that we only need to keep track of the certificates for inputs x such that $f_\rho(x) = 1$. The DNF we obtain in Theorem 14 is then simply the OR of all of these small 1-certificates.

2 Preliminaries

2.1 Notation and Basic Tools

We denote by $[N]$ the set $\{1, 2, \ldots, N\}$. For a finite set S, $|S|$ denotes the size of S. If $\mathcal{D}$ is a probability distribution, then $x \sim \mathcal{D}$ means that x is a random variable sampled from $\mathcal{D}$. If v is a real or complex vector, then $||v||$ denotes the Euclidean norm of v.

We use $\text{poly}(n)$ to denote an arbitrary polynomially-bounded function of n, i.e. a function f for which there is a constant c such that $f(n) \le n^c$ for all sufficiently large n. Likewise, we use $\text{polylog}(n)$ for an arbitrary f satisfying $f(n) \le \log(n)^c$ for all sufficiently large n, and $\text{quasipoly}(n)$ for an arbitrary f satisfying $f(n) \le 2^{\log(n)^c}$ for all sufficiently large n.

For a string $x \in \{0,1\}^N$, $|x|$ denotes the length of x. Additionally, if $i \in [N]$, then $x^{\oplus i}$ denotes the string obtained from x by flipping the ith bit. Similarly, if $S \subseteq [N]$, then $x^{\oplus S}$ denotes the string obtained from x by flipping the bits corresponding to all indices in S. For sets $S \subseteq [N]$, we sometimes use $\{0,1\}^S$ to denote mappings from S to $\{0,1\}$; these may equivalently be identified with strings in $\{0,1\}^{|S|}$ obtained by concatenating the bits of the mapping in order. We denote by $x|_S$ the string in $\{0,1\}^S$ obtained by restricting x to the bits indexed by S.

We view partial Boolean functions as functions of the form $f : \{0,1\}^N \to \{0,1,\perp\}$, where the domain of f is $f^{-1}(\{0,1\})$. We $\perp$ (instead of $*$) to refer to the evaluation of f on inputs outside the domain so as to avoid conflicting with our notation for random restrictions; see Section 2.4 below.

We use the following forms of the Chernoff bound:

Fact 15 (Chernoff bound)**.** *Suppose $X_1, \ldots, X_n$ are independent identically distributed random variables where $X_i = 1$ with probability p and $X_i = 0$ with probability $1-p$. Let $X = \sum_{i=1}^n X_i$ and let $\mu = \mathbb{E}[X] = pn$. Then for all $\delta \geq 0$ it holds that:*

$$\Pr\left[X \geq (1+\delta)\mu\right] \leq e^{-\frac{\delta^2 \mu}{2+\delta}},$$
$$\Pr\left[X \leq (1-\delta)\mu\right] \leq e^{-\frac{\delta^2 \mu}{2}},$$

Additionally, if $\delta \leq 1$, then we may use the weaker bound:

$$\Pr\left[X \geq (1+\delta)\mu\right] \leq e^{-\frac{\delta^2 \mu}{3}}.$$

We also require Hoeffding's inequality, which generalizes Fact 15 to sums of arbitrary independent bounded random variables:

Fact 16 (Hoeffding's inequality)**.** *Suppose $X_1, \ldots, X_n$ are independent random variables subject to $a_i \leq X_i \leq b_i$ for all i. Let $X = \sum_{i=1}^n X_i$ and let $\mu = \mathbb{E}[X]$. Then for all $\delta \geq 0$ it holds that:*

$$\Pr\left[X \geq (1+\delta)\mu\right] \leq \exp\left(-\frac{2\delta^2\mu^2}{\sum_{i=1}^n (b_i - a_i)^2}\right).$$

2.2 Complexity Classes

We assume familiarity with basic complexity classes, including: P, NP, $\mathsf{PH} = \bigcup_{k=0}^{\infty} \Sigma_k^{\mathsf{P}}$, PP, $\mathsf{P}^{\#\mathsf{P}}$, BQP, QCMA, and QMA; see e.g. the Complexity Zoo[7] for definitions. We also consider the relativized versions of these complexity classes augmented with oracles, where we use the standard notation that $\mathcal{C}^{\mathcal{O}}$ denotes a complexity class $\mathcal{C}$ augmented with oracle $\mathcal{O}$. A *random oracle* $\mathcal{O}$ is a uniformly random language where for each $x \in \{0,1\}^*$, $\mathcal{O}(x) = 0$ or 1 with probability $\frac{1}{2}$ (independently for each x).

We define a quantum analogue of the polynomial hierarchy that we call QMAH, and denote by $\mathsf{PromiseQMAH}$ the promise version of this class. Let $\mathsf{PromiseQMAH}_0 = \mathsf{PromiseP}$, and for $k \in \mathbb{N}$ we recursively define:

$$\mathsf{PromiseQMAH}_k := \mathsf{PromiseQMA}^{\mathsf{PromiseQMAH}_{k-1}},$$

Then, analogous to PH, we take:

$$\mathsf{PromiseQMAH} := \bigcup_{k=1}^{\infty} \mathsf{PromiseQMAH}_k$$

[7] https://complexityzoo.net/Complexity_Zoo

QMAH denotes the set of languages in PromiseQMAH. In the above definition, we define queries to a promise problem in the standard way, as follows. Recall that a promise problem can be viewed as a partial function $\Pi : \{0,1\}^* \to \{0,1,\perp\}$. We say that a language $A : \{0,1\}^* \to \{0,1\}$ *extends* Π if for all $x \in \{0,1\}^*$, $\Pi(x) \neq \perp \implies \Pi(x) = A(x)$. Then, a promise problem P is in $\mathsf{PromiseQMA}^{\Pi}$ if there exists a single QMA verifier V such that, for every language A that extends Π, V^A decides P. For a class $\mathcal{C}$ of promise problems, we let $\mathsf{PromiseQMA}^{\mathcal{C}} := \bigcup_{\Pi \in \mathcal{C}} \mathsf{PromiseQMA}^{\Pi}$. This is essentially the same definition that was given by Aaronson and Drucker [AD14].

We remark that it is not clear to us if this is the "right" way to define quantum queries to a promise problem. If we have query access to some quantum algorithm that "solves" a promise problem, that algorithm could conceivably behave arbitrarily (even non-unitarily) on the inputs outside of the promise: there is no guarantee that it decides some language, as we assume above. However, since we are chiefly interested in proving *lower bounds* on QMAH complexity, this distinction makes little difference to us: our choice of definition could only possibly make the class $\mathsf{PromiseQMA}^{\Pi}$ *more* powerful than if the oracle could have worse behavior on non-promise inputs. Whether complexity classes such as BQP^{Π} and QMA^{Π} are robust with respect to the notion of promise problem queries remains an interesting question for future work.

We note that there are many other possible ways to define a quantum analogue of the polynomial hierarchy (see [GSS+18]), and that our definition appears to differ from all others that we are aware of. The definition we give is closest in spirit to a class called BQPH by Vinkhuijzen [Vin18], except that we allow recursive queries to PromiseQMA instead of just QMA.

2.3 Query Complexity and Related Measures

We assume some familiarity with quantum and classical query complexity. We recommend a survey by Ambainis [Amb18] for additional background and definitions. A quantum query to a string $x \in \{0,1\}^N$ is implemented via the unitary transformation U_x that acts on basis states of the form $|i\rangle |w\rangle$ as $U_x |i\rangle |w\rangle = (-1)^{x_i} |i\rangle |w\rangle$, where $i \in [N]$ and w is an index over a workspace register.

Following standard notation, for a (possibly partial) function $f : \{0,1\}^N \to \{0,1,\perp\}$, we denote by $\mathsf{Q}(f)$ the bounded-error quantum query complexity of f, which means the fewest number of queries made by a quantum algorithm that computes f with error probability at most $\frac{1}{3}$. We denote by $\mathsf{D}(f)$ the deterministic query complexity of f, which is also sometimes called decision tree complexity.

We define a notion of QMA (Quantum Merlin-Arthur) query complexity as follows.

Definition 17 (QMA query complexity)**.** *The* (bounded-error) QMA query complexity *of a function $f : \{0,1\}^N \to \{0,1,\perp\}$, denoted* $\mathsf{QMA}(f)$*, is the fewest number of queries made by any quantum query algorithm $\mathcal{V}(|\psi\rangle, x)$ that takes an auxiliary input state $|\psi\rangle$ and satisfies, for all $x \in \{0,1\}^N$:*

- *(Completeness) If $f(x) = 1$, then there exists a state $|\psi\rangle$ such that $\Pr[\mathcal{V}(|\psi\rangle, x) = 1] \geq \frac{2}{3}$, and*
- *(Soundness) If $f(x) = 0$, then for every state $|\psi\rangle$, $\Pr[\mathcal{V}(|\psi\rangle, x) = 1] \leq \frac{1}{3}$.*

The algorithm $\mathcal{V}$ is sometimes called the verifier*, and the state $|\psi\rangle$ a* witness*.*

Note that in contrast to most previous works (c.f. [RS04, AKKT20, ST19]), our definition of QMA query complexity completely ignores the number of qubits in the witness state $|\psi\rangle$.

We next define sensitivity, and the related B-block sensitivity.

Definition 18 (Sensitivity)**.** *The* sensitivity *of a function $f : \{0,1\}^N \to \{0,1\}$ on input x is defined as:*

$$\mathsf{s}^x(f) := \left| \left\{ i \in [n] : f(x) \neq f\left(x^{\oplus i}\right) \right\} \right|.$$

The sensitivity of f *is defined as:*

$$\mathsf{s}(f) \coloneqq \max_{x\in\{0,1\}^N} \mathsf{s}^x(f).$$

Definition 19 (B-block sensitivity)**.** *Let $B = \{S_1, S_2, \ldots, S_k\}$ be collection of disjoint subsets of $[N]$. The* block sensitivity *of a function $f : \{0,1\}^N \to \{0,1\}$ on input $x \in \{0,1\}^N$ with respect to B is defined as:*

$$\mathsf{bs}_B^x(f) \coloneqq \left|\left\{i \in [k] : f(x) \neq f\left(x^{\oplus S_i}\right)\right\}\right|.$$

The block sensitivity of f *with respect to B is defined as:*

$$\mathsf{bs}_B(f) \coloneqq \max_{x\in\{0,1\}^N} \mathsf{bs}_B^x(f).$$

Note that sensitivity is the special case of B-block sensitivity in which B is the partition into singletons.

We require a somewhat unusual definition of certificate complexity. Our definition agrees with the standard definition for *total* functions, but may differ for partial functions.

Definition 20 (Certificate complexity)**.** *Let $f : \{0,1\}^N \to \{0,1,\bot\}$, and suppose $x \in \{0,1\}^N$ satisfies $f(x) \in \{0,1\}$. A* certificate for x on f*, also called an* $f(x)$-certificate*, is a set $K \subseteq [N]$ such that for any $y \in \{0,1\}^N$ satisfying $y_i = x_i$ for all $i \in K$, either $f(y) = f(x)$ or $f(y) = \bot$.*

The certificate complexity of f on x*, denoted $\mathsf{C}^x(f)$, is the minimum size of any certificate for f on x. The* certificate complexity of f *is defined as:*

$$\mathsf{C}(f) \coloneqq \max_{x\in\{0,1\}^N} \mathsf{C}^x(f).$$

Intuitively, in this definition of certificate complexity, a b-certificate for $b \in \{0,1\}$ witnesses that $f(x) \neq 1-b$, in contrast to the standard definition where a b-certificate witnesses that $f(x) = b$.

2.4 Random Restrictions

A *restriction* is a function $\rho : [N] \to \{0,1,*\}$. A *random restriction* with $\Pr[*] = p$ is a distribution over restrictions in which, for each $i \in [N]$, we independently sample:

$$\rho(i) = \begin{cases} 0 & \text{with probability } \frac{1-p}{2} \\ 1 & \text{with probability } \frac{1-p}{2} \\ * & \text{with probability } p. \end{cases}$$

If $f : \{0,1\}^N \to \{0,1,\bot\}$ is a function and ρ is a restriction where $S = \{i \in [N] : \rho(i) = *\}$, we denote by $f_\rho : \{0,1\}^S \to \{0,1,\bot\}$ the function obtained from f by fixing the inputs where $\rho(i) \in \{0,1\}$. We call the remaining variables the *unrestricted variables*. We sometimes apply restrictions to functions f_i that have a subscript in the name, in which case we denote the restricted function by $f_{i|\rho}$ for notational clarity.

In this work, we also make use of *projections*, which are a generalization of restrictions that were introduced in [HRST17]. The exact definition of projections is unimportant for us, but intuitively, they are restrictions where certain unrestricted variables may be mapped to each other; see [HRST17] for a more precise definition. We use the same notation f_ρ for applying a projection ρ to a function f as we do for restrictions.

2.5 Circuit Complexity

In this work, we consider Boolean circuits where the gates can be arbitrary partial Boolean functions $f : \{0,1\}^N \to \{0,1,\perp\}$. On input $x \in \{0,1,\perp\}^N$, a circuit gate labeled by f evaluates to $b \in \{0,1\}$ if, for all $y \in \{0,1\}^N$ that agree with x (meaning, for all $i \in [n]$, $x_i \in \{0,1\}$ implies $y_i = x_i$), we have $f(y) = b$; otherwise, the gate evaluates to $\perp$.

We always specify the basis of gates allowed in Boolean circuits. Most commonly, we will consider Boolean circuits with AND, OR, and NOT gates where the AND and OR gates can have unbounded fan-in, but we will also consider e.g. circuits where the gates can be arbitrary functions of low QMA query complexity.

The *size* of a circuit is the number of gates of fan-in larger than 1 (i.e. excluding NOT gates). The *depth* of circuit is the length of the longest path from an input variable to the output gate, ignoring gates of fan-in 1.

An AND/OR/NOT circuit is *alternating* if all NOT gates are directly above the input, and all paths from the inputs to the output gate alternate between AND and OR gates. A circuit is *layered* if for each gate g in the circuit, the distance from g to the output gate is the same along all paths. We denote by $\mathsf{AC}^0[s,d]$ the set of alternating, layered AND/OR/NOT circuits of size at most s and depth at most d. By a folklore result, any AND/OR/NOT circuit can be turned into an alternating, layered circuit of the same depth at the cost of a small (constant multiplicative) increase in size.

A *DNF formula*, also just called a DNF, is a depth-2 AC^0 circuit where the top gate is an OR gate (i.e. the circuit is an OR of ANDs). The *width* of a DNF is the maximum fan-in of any of the AND gates.

We require the following results in circuit complexity.

Theorem 21 ([Hås87, Lemma 7.8])**.** *For every constant d, there exists a constant c such that for all sufficiently large N, for all $C \in \mathsf{AC}^0\left[2^{N^c}, d\right]$, one has:*

$$\Pr_{x \sim \{0,1\}^N}\left[C(x) = \textsc{Parity}_N(x)\right] \leq 0.6,$$

where $\textsc{Parity}_N$ is the parity function on N bits.

Theorem 22 ([HRST17, Proof of Theorem 10.1])**.** *For all constant $d \geq 2$ and all sufficiently large $m \in \mathbb{N}$, there exists a function $\textsc{Sipser}_d$ with $N = 2^{\Theta(m)}$ inputs, and a class $\mathcal{R}$ of random projections such that the following hold:*

(a) For some value $b = 2^{-m}\left(1 - O\left(2^{-m/2}\right)\right)$, for any function $f : \{0,1\}^N \to \{0,1,\perp\}$,

$$\Pr_{x \sim \{0,1\}^N}\left[f(x) = \textsc{Sipser}_d(x)\right] = \Pr_{x \sim D, \rho \sim \mathcal{R}}\left[f_\rho(x) = \textsc{Sipser}_{d|\rho}(x)\right],$$

where D is the distribution over bit strings in which each coordinate is 0 with probability b and 1 with probability $1-b$.

Additionally, let $C \in \mathsf{AC}^0[s, d-1]$. If we sample $\rho \sim \mathcal{R}$, then:

(b) Except with probability at most $s2^{-2^{m/2-4}}$, $\mathsf{D}\left(C_\rho\right) \leq 2^{m/2-4}$.

(c) Except with probability at most $O\left(2^{-m/2}\right)$, $\textsc{Sipser}_{d|\rho}$ is reduced to an AND gate of fan-in $(\ln 2) \cdot 2^m \cdot \left(1 \pm O\left(2^{-m/4}\right)\right)$.

[HRST17] roughly explains the intuitive meaning of the above theorem as follows. Property (a) guarantees that the distribution $\mathcal{R}$ of random projections completes to the uniform distribution. Property (b) shows that the circuit C simplifies with high probability under a random projection, while property (c) shows that $\textsc{Sipser}_d$ retains structure under this distribution of projections with high probability. A simple corollary of these properties is the following:

Corollary 23 ([HRST17, Theorem 10.1])**.** *Let $\textsc{Sipser}_d$ be the function defined in Theorem 22 on $N = 2^{\Theta(m)}$ bits. Let $C \in \mathsf{AC}^0[s, d-1]$. Then, for all sufficiently large m, we have:*

$$\Pr_{x \sim \{0,1\}^N}\left[C(x) = \textsc{Sipser}_d(x)\right] \leq \frac{1}{2} + O\left(2^{-m/4}\right) + s2^{-2^{m/2-4}}.$$

2.6 Other Background

The form of the Raz-Tal Theorem stated below forms the basis for several of our results. It states that there exists a distribution that looks pseudorandom to small constant-depth AC^0 circuits, but that is easily distinguishable from random by an efficient quantum algorithm.

Theorem 24 ([RT19, Theorem 1.2])**.** *For all sufficiently large N, there exists an explicit distribution $\mathcal{F}_N$ that we call the* Forrelation distribution *over $\{0,1\}^N$ such that:*

1. *There exists a quantum algorithm $\mathcal{A}$ that makes* $\mathrm{polylog}(N)$ *queries and runs in time* $\mathrm{polylog}(N)$ *such that:*
$$\left|\Pr_{x \sim \mathcal{F}_N}\left[\mathcal{A}(x) = 1\right] - \Pr_{y \sim \{0,1\}^N}\left[\mathcal{A}(y) = 1\right]\right| \geq 1 - \frac{1}{N^2}.$$
2. *For any $C \in \mathsf{AC}^0[\mathrm{quasipoly}(N), O(1)]$:*
$$\left|\Pr_{x \sim \mathcal{F}_N}\left[C(x) = 1\right] - \Pr_{y \sim \{0,1\}^N}\left[C(y) = 1\right]\right| \leq \frac{\mathrm{polylog}(N)}{\sqrt{N}}.$$

Note that, by standard amplification techniques, the $\frac{1}{N^2}$ in the above theorem can be replaced by any $\delta \leq 2^{-\mathrm{polylog}(N)}$ at a cost of $\mathrm{polylog}(N)$ in the other parameters. For our purposes, the above theorem suffices as written. In some cases where N is clear from context, we omit the subscript and write the distribution as $\mathcal{F}$. Additionally, in a slight abuse of notation, we sometimes informally call the decision problem of distinguishing a sample from $\mathcal{F}_N$ from a sample from the uniform distribution the $\textsc{Forrelation}$ problem.

The next lemma was essentially shown in [BBBV97]. We provide a proof for completeness.

Lemma 25. *Consider a quantum algorithm Q that makes T queries to $x \in \{0,1\}^N$. Write the state of the quantum algorithm immediately after t queries to x as:*

$$|\psi_t\rangle = \sum_{i=1}^{N} \sum_{w} \alpha_{i,w,t} |i, w\rangle,$$

where w are indices over a workspace register. Define the query magnitude *q_i of an input $i \in [N]$ by:*

$$q_i \coloneqq \sum_{t=1}^{T} \sum_{w} |\alpha_{i,w,t}|^2.$$

Then for any $y \in \{0,1\}^N$, we have:

$$\left|\Pr\left[Q(x) = 1\right] - \Pr\left[Q(y) = 1\right]\right| \leq 8\sqrt{T} \cdot \sqrt{\sum_{i : x_i \neq y_i} q_i}.$$

Proof. Denote by $|\psi'_t\rangle$ the state of the quantum algorithm after t queries, where the first $t-1$ queries are to x and the tth query is to y. For $t > 0$, we have:

$$\begin{aligned} ||\,|\psi_t\rangle - |\psi'_t\rangle\,|| &= \left\| 2 \sum_{i:x_i \neq y_i} \sum_w \alpha_{i,w,t} |i,w\rangle \right\| \\ &= 2\sqrt{\sum_{i:x_i \neq y_i} \sum_w |\alpha_{i,w,t}|^2}. \end{aligned}$$

Hence, if we denote by $|\varphi_t\rangle$ the state of the quantum algorithm after t queries, where all t queries are to y, then:

$$\begin{aligned} |\Pr[Q(x) = 1] - \Pr[Q(y) = 1]| &\leq 4||\,|\psi_T\rangle - |\varphi_T\rangle\,|| \\ &\leq \sum_{t=1}^{T} 8\sqrt{\sum_{i:x_i \neq y_i} \sum_w |\alpha_{i,w,t}|^2} \\ &\leq 8\sqrt{T} \cdot \sqrt{\sum_{t=1}^{T} \sum_{i:x_i \neq y_i} \sum_w |\alpha_{i,w,t}|^2} \\ &= 8\sqrt{T} \cdot \sqrt{\sum_{i:x_i \neq y_i} q_i}. \end{aligned}$$

Above, the first line holds by [BV97, Lemma 3.6]; the second line is valid by the BBBV hybrid argument used in [BBBV97, Theorem 3.3]; the third line applies the Cauchy-Schwarz inequality, viewing the summation as the inner product between the all 1s vector and the terms of the sum; and the last line substitutes the definition of q_i. □

3 Consequences of the Raz-Tal Theorem

In this section, we prove several oracle separations that build on the Raz-Tal Theorem (Theorem 24) and other known circuit lower bounds.

3.1 Relativizing (Non-)Implications of $\mathsf{NP} \subseteq \mathsf{BQP}$

Our first result proves the following:

Theorem 26. *There exists an oracle relative to which* $\mathsf{BQP} = \mathsf{P}^{\#\mathsf{P}}$ *and* PH *is infinite.*

The proof idea is as follows. First, we take a random oracle, which makes PH infinite [HRST17, RST15]. Then, we encode the answers to all possible $\mathsf{P}^{\#\mathsf{P}}$ queries in instances of the FORRELATION problem, allowing a BQP machine to efficiently decide any $\mathsf{P}^{\#\mathsf{P}}$ language. We then leverage Theorem 24 to argue that adding these FORRELATION instances does not collapse PH, because the FORRELATION instances look random to PH algorithms. The formal proof is given below.

Proof of Theorem 26. We will inductively construct this oracle $\mathcal{O}$, which will consist of two parts, A and B. Denote the first region of the oracle A, and let this be a random oracle. For each $t \in \mathbb{N}$, we will add a region of B called B_t that will depend on the previously constructed parts of the oracle. For convenience, we let A_t denote the region of A corresponding to inputs of length t, and we write $\mathcal{O}_t = (A_t, B_t)$.

Let S_t be the set of all ordered pairs of the form $\langle M, x\rangle$ such that:

1. M is a $\mathsf{P}^{\#\mathsf{P}}$ oracle machine and x is an input to M,

2. $\langle M, x\rangle$ takes less than t bits to specify, and

3. M is syntactically restricted to run in less than t steps, and to query only the $\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_{\lfloor\sqrt{t}\rfloor}$ regions of the oracle.

Note that there are at most 2^t elements in S_t. Let $M_1, M_2, \ldots, M_{2^t}$ be an enumeration of S_t. For each M_i in S_t, we add a function $f_i : \{0,1\}^{t^2} \to \{0,1\}$ into B_t. The function f_i is chosen subject to the following rules:

1. If M_i accepts, then f_i is drawn from the Forrelation distribution $\mathcal{F}_{2^{t^2}}$ (given in Theorem 24).

2. If M_i rejects, then f_i is uniformly random.

Let $\mathcal{D}$ be the resulting distribution over oracles $\mathcal{O} = (A, B)$.

Claim 27. $\mathsf{BQP}^{\mathcal{O}} = \mathsf{P}^{\#\mathsf{P}^{\mathcal{O}}}$ *with probability* 1 *over* $\mathcal{O}$.

Proof of Claim. It suffices to show that $\mathsf{P}^{\#\mathsf{P}^{\mathcal{O}}} \subseteq \mathsf{BQP}^{\mathcal{O}}$, as the reverse containment holds relative to all oracles. Let M be any $\mathsf{P}^{\#\mathsf{P}}$ oracle machine. Then given an input x of size n, a quantum algorithm can decide whether $M^{\mathcal{O}}(x)$ accepts in poly(n) time by looking up the appropriate B_t, the one that contains a FORRELATION instance $f_i : \{0,1\}^{t^2} \to \{0,1\}$ encoding the behavior of $\langle M, x\rangle$, and then deciding whether f_i is Forrelated or random by using the algorithm $\mathcal{A}$ from Theorem 24.

In more detail, by Theorem 24 we know that:

$$\Pr_{\mathcal{O}\sim\mathcal{D}}\left[\mathcal{A}(f_i) \neq M^{\mathcal{O}}(x)\right] \leq 2^{-2t^2},$$

where the probability in the above expression is also taken over the randomness of $\mathcal{A}$. By Markov's inequality, we may conclude:

$$\Pr_{\mathcal{O}\sim\mathcal{D}}\left[\Pr\left[\mathcal{A}(f_i) \neq M^{\mathcal{O}}(x)\right] \geq 1/3\right] \leq 3 \cdot 2^{-2t^2}.$$

Hence, the BQP promise problem defined by $\mathcal{A}$ agrees with the $\mathsf{P}^{\#\mathsf{P}}$ language on $\langle M, x\rangle$, except with probability at most $3 \cdot 2^{-2t^2}$.

We now appeal to the Borel-Cantelli Lemma to argue that, with probability 1 over $\mathcal{O} \sim \mathcal{D}$, $\mathcal{A}$ correctly decides $M^{\mathcal{O}}(x)$ for all but finitely many $x \in \{0,1\}^*$. Since there are at most 2^t inputs $\langle M, x\rangle$ that take less than t bits to specify, we have:

$$\sum_{\langle x,M\rangle \in \{0,1\}^*} \Pr_{\mathcal{O}\sim\mathcal{D}}\left[\mathcal{A}^{\mathcal{O}} \text{ does not decide } M^{\mathcal{O}}(x)\right] \leq \sum_{t=1}^{\infty} 2^t \cdot 3 \cdot 2^{-2t^2} < \infty$$

Therefore, the probability that $\mathcal{A}$ fails on infinitely many inputs $\langle M, x\rangle$ is 0. Hence, $\mathcal{A}$ can be modified into a BQP algorithm that decides $M^{\mathcal{O}}(x)$ for *all* $x \in \{0,1\}^*$, with probability 1 over $\mathcal{O} \sim \mathcal{D}$. □

Now, we must show that $\mathsf{PH}^{\mathcal{O}}$ is infinite. We will accomplish this by proving, for all $k \in \mathbb{N}$, $\Sigma_k^{\mathsf{P}^{\mathcal{O}}} \neq \Sigma_{k-1}^{\mathsf{P}^{\mathcal{O}}}$ with probability 1 over the choice of $\mathcal{O}$. Let $L^{\mathcal{O}}$ be the unary language used for the same purpose as in [RST15, HRST17]. That is, $L^{\mathcal{O}}$ consists of strings 0^n such that, if we treat n as an index into a portion of the random oracle A_n that encodes a size 2^n instance of the SIPSER$_{k+1}$ function, then that instance evaluates to 1. By construction, $L^{\mathcal{O}} \in \Sigma_k^{\mathsf{P}^{\mathcal{O}}}$ [RST15, HRST17]. Furthermore, [RST15, HRST17] show that $L^{\mathcal{O}}$ is not in $\Sigma_{k-1}^{\mathsf{P}^A}$ with probability 1 over the random oracle A. We need to argue that adding B has probability 0 of changing this situation. Fix any $\Sigma_{k-1}^{\mathsf{P}^{\mathcal{O}}}$ oracle machine M. By the union bound, it suffices to show that

$$\Pr_{\mathcal{O}\sim\mathcal{D}}\left[M^{\mathcal{O}} \text{ decides } L^{\mathcal{O}}\right] = 0.$$

Let $n_1 < n_2 < \cdots$ be an infinite sequence of input lengths, spaced far enough apart (e.g. $n_{i+1} = 2^{n_i}$) such that $M\left(0^{n_i}\right)$ can query the oracle on strings of length n_{i+1} or greater for at most finitely many values of i. Next, let

$$p(M,i) \coloneqq \Pr_{\mathcal{O}\sim\mathcal{D}}\left[M^{\mathcal{O}} \text{ correctly decides } 0^{n_i} | M^{\mathcal{O}} \text{ correctly decided } 0^{n_1},\ldots,0^{n_{i-1}}\right]$$

Then we have that

$$\Pr_{\mathcal{O}\sim\mathcal{D}}\left[M^{\mathcal{O}} \text{ decides } L^{\mathcal{O}}\right] \leq \prod_{i=1}^{\infty} p(M,i)$$

Thus it suffices to show that, for every fixed M, we have $p(M,i) \leq 0.7$ for all but finitely many i. To do this, we will consider a new quantity $q(M,i)$, which is defined exactly the same way as $p(M,i)$, except that now the oracle is chosen from a different distribution, which we call D_i. This D_i is defined identically to $\mathcal{D}$ on A and $B_1,\ldots,B_{n_i}$, but is uniformly random on B_m for all $m > n_i$. It suffices to prove the following: for any fixed M,

(a) $q(M,i) \leq 0.6$ for all but finitely many values of i, and

(b) $|q(M,i) - p(M,i)| \leq 0.1$ for all but finitely many values of i.

Statement (a) essentially follows from the work of [HRST17]. The key observation is that the only portion of $\mathcal{O}$ that can depend on whether 0^{n_i} is in $L^{\mathcal{O}}$ is the input to the size-2^{n_i} $\textsc{Sipser}_{k+1}$ function that is encoded in A. All other portions of $\mathcal{O}$ are sampled independently from this region under D_i:

1. The rest of A is sampled uniformly at random,
2. $B_1,\ldots,B_{n_i}$ are drawn from a distribution that cannot depend on any queries to A on inputs of length $\lfloor\sqrt{n_i}\rfloor$ or greater (so in particular, they cannot depend on A_{n_i}), and
3. $B_{n_i+1}, B_{n_i+2},\ldots$ are sampled uniformly at random.

Hence, M is forced to evaluate the size-2^{n_i} $\textsc{Sipser}_{k+1}$ function using only auxilliary and uncorrelated random bits. By the well-known connection between $\Sigma_{k-1}^{\mathsf{P}}$ query algorithms and constant-depth circuits [FSS84], M's behavior on this size-2^{n_i} string can be computed by an $\mathsf{AC^0}\left[2^{\mathrm{poly}(n_i)}, k\right]$ circuit. Corollary 23 shows that such a circuit correctly evaluates this $\textsc{Sipser}_{k+1}$ function with probability greater than (say) 0.6 for at most finitely many i. This even holds conditioned on $M^{\mathcal{O}}$ correctly deciding $0^{n_1},\ldots,0^{n_{i-1}}$, because the size-$2^{n_i}$ $\textsc{Sipser}_{k+1}$ instance is chosen independently from the smaller instances, and because $M\left(0^{n_i}\right)$ can query the oracle on strings of length n_{i+1} or greater for at most finitely many values of i.

For statement (b), we will prove this claim using a hybrid argument. We consider an infinite sequence of hybrids $\{D_{i,j} : j \in \mathbb{N}\}$ between $D_i = D_{i,0}$ and $\mathcal{D}$, where in the jth hybrid $D_{i,j}$ we sample A and $B_1,\ldots,B_{n_i+j}$ according to $\mathcal{D}$ and $B_{n_i+j+1}, B_{n_i+j+2},\ldots$ uniformly at random. The change between each $D_{i,j-1}$ and $D_{i,j}$ may be further decomposed into a sequence of smaller changes: from the uniform distribution $\mathcal{U}$ to the Forrelated $\mathcal{F}$, for each function $f : \{0,1\}^{(n_i+j)^2} \to \{0,1\}$ corresponding to a Σ_k^{P} oracle machine that happens to accept.

Suppose we fix the values of $\mathcal{O}$ on everything except for f. Theorem 24 implies that:

$$\left| \Pr_{f \sim \mathcal{F}} \left[M^{\mathcal{O}} \left(0^{n_i} \right) = 1 \right] - \Pr_{f \sim \mathcal{U}} \left[M^{\mathcal{O}} \left(0^{n_i} \right) = 1 \right] \right| \leq \frac{\text{poly}(n_i)}{2^{(n_i+j)^2/2}}. \tag{1}$$

This is because, again using [FSS84], there exists an $\mathsf{AC}^0 \left[2^{\text{poly}(n_i)}, k \right]$ circuit that takes the oracle string as input and evaluates to $M^{\mathcal{O}} \left(0^{n_i} \right)$. In fact, (1) *also* holds even if the parts of $\mathcal{O}$ other than f are not necessarily fixed, but are drawn from some distribution, by convexity (so long as the distribution is the same in both of the probabilities in (1)). In particular, using the fact that the n_is are far enough apart for sufficiently large i, (1) also holds (for all sufficiently large i) when the parts of $\mathcal{O}$ other than f are drawn from the distribution conditioned on $M^{\mathcal{O}}$ correctly deciding $0^{n_1}, \ldots, 0^{n_{i-1}}$.

Now, recall that there are at most 2^t Forrelation instances in the B_t part of the oracle. By the triangle inequality, bounding over each of these instances yields:

$$\left| \Pr_{\mathcal{O} \sim D_{i,j-1}} \left[M^{\mathcal{O}} \left(0^{n_i} \right) = 1 \right] - \Pr_{\mathcal{O} \sim D_{i,j}} \left[M^{\mathcal{O}} \left(0^{n_i} \right) = 1 \right] \right| \leq 2^{n_i+j} \cdot \frac{\text{poly}(n_i)}{2^{(n_i+j)^2/2}},$$

where we implicitly condition on $M^{\mathcal{O}}$ correctly deciding $0^{n_1}, \ldots, 0^{n_{i-1}}$ in both of the probabilities above, omitting it as written purely for notational simplicity. Hence, when we change *all* of the hybrids, we obtain:

$$\begin{aligned} |q(M,i) - p(M,i)| &= \left| \Pr_{\mathcal{O} \sim D_i} \left[M^{\mathcal{O}} \left(0^{n_i} \right) = 1 \right] - \Pr_{\mathcal{O} \sim \mathcal{D}} \left[M^{\mathcal{O}} \left(0^{n_i} \right) = 1 \right] \right| \\ &\leq \sum_{j=1}^{\infty} 2^{n_i+j} \cdot \frac{\text{poly}(n_i)}{2^{(n_i+j)^2/2}} \\ &\leq \frac{\text{poly}(n_i)}{2^{\Omega(n_i^2)}} \\ &\leq 0.1 \end{aligned}$$

for all but at most finitely many i. □

We conclude this section with a simple corollary.

Corollary 28. *There exists an oracle relative to which* $\mathsf{BQP} \not\subset \mathsf{NP/poly}$.

Proof. It is known that for all oracles $\mathcal{O}$, $\mathsf{coNP}^{\mathcal{O}} \subset \mathsf{NP}^{\mathcal{O}}/\mathsf{poly}$ implies that $\mathsf{PH}^{\mathcal{O}}$ collapses to the third level [Yap83]. Let $\mathcal{O}$ be the oracle used in Theorem 26. Since $\mathsf{PH}^{\mathcal{O}}$ is infinite, $\mathsf{coNP}^{\mathcal{O}} \not\subset \mathsf{NP}^{\mathcal{O}}/\mathsf{poly}$. On the other hand, $\mathsf{coNP}^{\mathcal{O}} \subseteq \mathsf{BQP}^{\mathcal{O}}$, and hence $\mathsf{BQP}^{\mathcal{O}} \not\subset \mathsf{NP}^{\mathcal{O}}/\mathsf{poly}$. □

3.2 Weak NP, Strong BQP

In this section, we prove the following:

Theorem 29. *There exists an oracle relative to which* $\mathsf{P} = \mathsf{NP} \neq \mathsf{BQP} = \mathsf{P}^{\#\mathsf{P}}$.

We follow a similar proof strategy to Theorem 26, with some additional steps. First, we take a random oracle, which separates PH from $\mathsf{P}^{\#\mathsf{P}}$ [Hås87]. We encode the answers to all possible $\mathsf{P}^{\#\mathsf{P}}$ queries in instances of the Forrelation problem, allowing a BQP machine to efficiently decide any $\mathsf{P}^{\#\mathsf{P}}$ language. Then, we add a region of the oracle that answers all NP queries, which collapses PH to P. Finally, we leverage Theorem 24 to argue that the Forrelation instances have no effect on the separation between PH and $\mathsf{P}^{\#\mathsf{P}}$, because the Forrelation instances look random to PH algorithms. The formal proof is given below.

Proof of Theorem 29. This oracle $\mathcal{O}$ will consist of three parts: a random oracle A, and oracles B and C that we will construct inductively. For each $t \in \mathbb{N}$, we will add regions B_t and C_t that will depend on the previously constructed parts of the oracle. For convenience, we let A_t denote the region of A corresponding to inputs of length t, and we write $\mathcal{O}_t = (A_t, B_t, C_t)$.

We first describe B, which will effectively collapse $\mathsf{P}^{\#\mathsf{P}}$ to BQP. For $t \in \mathbb{N}$, let S_t be the set of all ordered pairs of the form $\langle M, x \rangle$ such that:

1. M is a $\mathsf{P}^{\#\mathsf{P}}$ oracle machine and x is an input to M,
2. $\langle M, x \rangle$ takes less than t bits to specify, and
3. M is syntactically restricted to run in less than t steps, and to query only the $\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_{\lfloor \sqrt{t} \rfloor}$ regions of the oracle.

Note that there are at most 2^t elements in S_t. Let $M_1, M_2, \ldots, M_{2^t}$ be an enumeration of S_t. For each M_i in S_t, we add a function $f_i : \{0,1\}^{t^2} \to \{0,1\}$ into B_t. The function f_i is chosen subject to the following rules:

1. If M_i accepts, then f_i is drawn from the Forrelation distribution $\mathcal{F}_{2^{t^2}}$ (given in Theorem 24).
2. If M_i rejects, then f_i is uniformly random.

We next describe C, which will effectively collapse NP to P. For $t \in \mathbb{N}$, define T_t similarly to S_t, except that we take NP oracle machines instead of $\mathsf{P}^{\#\mathsf{P}}$ oracle machines. For each M_i in T_t, we add a bit into C_t that returns $M(x)$.

Let $\mathcal{D}$ be the resulting distribution over oracles $\mathcal{O} = (A, B, C)$. We will show that the statement of the theorem holds with probability 1 over $\mathcal{O}$ sampled from $\mathcal{D}$.

Claim 30. $\mathsf{P}^{\mathcal{O}} = \mathsf{NP}^{\mathcal{O}}$ *with probability* 1 *over* $\mathcal{O}$.

Claim 31. $\mathsf{BQP}^{\mathcal{O}} = \mathsf{P}^{\#\mathsf{P}^{\mathcal{O}}}$ *with probability* 1 *over* $\mathcal{O}$.

The proof of Claim 30 is trivial: given an $\mathsf{NP}^{\mathcal{O}}$ machine M and input x, a polynomial time algorithm can decide $M(x)$ by simply looking up the bit in C that encodes $M(x)$. The proof of Claim 31 is completely analogous to the proof of Claim 27 in Theorem 26, so we omit it.

To complete the proof, we will show that $\mathsf{NP}^{\mathcal{O}} \neq \mathsf{P}^{\#\mathsf{P}^{\mathcal{O}}}$ with probability 1 over $\mathcal{O}$. Let $L^{\mathcal{O}}$ be the following unary language: $L^{\mathcal{O}}$ consists of strings 0^n such that, if we treat n as an index into a portion of the random oracle A_n of size 2^n, then the parity of that length-2^n string is 1. By construction, $L^{\mathcal{O}} \in \mathsf{P}^{\#\mathsf{P}^{\mathcal{O}}}$. We will show that $L^{\mathcal{O}} \notin \mathsf{NP}^{\mathcal{O}}$ with probability 1 over $\mathcal{O}$.

Fix any $\mathsf{NP}^{\mathcal{O}}$ oracle machine M. By the union bound, it suffices to show that

$$\Pr_{\mathcal{O} \sim \mathcal{D}} \left[M^{\mathcal{O}} \text{ decides } L^{\mathcal{O}} \right] = 0.$$

Let $n_1 < n_2 < \cdots$ be an infinite sequence of input lengths, spaced far enough apart (e.g. $n_{i+1} = 2^{n_i}$) such that $M\left(0^{n_i}\right)$ can query the oracle on strings of length n_{i+1} or greater for at most finitely many values of i. Next, let

$$p(M, i) := \Pr_{\mathcal{O} \sim \mathcal{D}} \left[M^{\mathcal{O}} \text{ correctly decides } 0^{n_i} | M^{\mathcal{O}} \text{ correctly decided } 0^{n_1}, \ldots, 0^{n_{i-1}} \right]$$

Then we have that

$$\Pr_{\mathcal{O} \sim \mathcal{D}} \left[M^{\mathcal{O}} \text{ decides } L^{\mathcal{O}} \right] \leq \prod_{i=1}^{\infty} p(M, i)$$

Thus it suffices to show that, for every fixed M, we have $p(M,i) \leq 0.7$ for all but finitely many i. To do this, we will consider a new quantity $q(M,i)$, which is defined exactly the same way as $p(M,i)$, except that now the oracle is chosen from a different distribution, which we call D_i. This D_i is defined identically to $\mathcal{D}$ on A, C, and $B_1,\ldots,B_{n_i}$, but is uniformly random on B_m for all $m > n_i$. It suffices to prove the following: for any fixed M,

(a) $q(M,i) \leq 0.6$ for all but finitely many values of i, and

(b) $|q(M,i) - p(M,i)| \leq 0.1$ for all but finitely many values of i.

To prove these, we first need the following lemma, which essentially states that for any $t' \leq \text{poly}(t)$, any bit in $C_{t'}$ can be computed by a small (i.e. quasipolynomial in the input length) constant-depth AC^0 circuit whose inputs do not depend on C_i for any $i > t$.

Lemma 32. *Fix $t, d \in \mathbb{N}$, and let $t' \leq t^{2^d}$. For each $\langle M, x\rangle \in S_{t'}$, there exists an AND/OR/NOT circuit of size at most $2^{1+t^{2^d}}$ and depth $2d$ that takes as input $A_1, A_2, \ldots, A_{t^{2^{d-1}}}$; $B_1, B_2, \ldots, B_{t^{2^{d-1}}}$; and $C_1, C_2, \ldots, C_t$, and outputs $M(x)$.*

Proof of Lemma. Assume $t \geq 2$ (otherwise the theorem is trivial). We proceed by induction on d. Consider the base case $d = 1$. By definition of $S_{t'}$, $\langle M, x\rangle$ is an NP oracle machine that runs in less than t' steps and queries only the $\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_t$ regions of the oracle. Hence, $M(x)$ computes a function of certificate complexity (Definition 20) at most t' in the bits of $\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_t$. This function may thus be expressed as a DNF formula of width at most t', which is in turn an AND/OR/NOT circuit of depth 2 and size at most $2^{t'} + 1 \leq 2^{t^2} + 1 \leq 2^{1+t^2}$.

For the inductive step, let $d \geq 2$. Similar to the base case, we use the definition of $S_{t'}$ to obtain a circuit of depth 2 and size at most $2^{t'} + 1$ that takes as input $\mathcal{O}_1, \mathcal{O}_2, \ldots, \mathcal{O}_{t^{2^{d-1}}}$ and outputs $M(x)$. To complete the theorem, we use the inductive hypothesis to replace each of the inputs to this DNF formula from the regions $C_{t+1}, C_{t+2}, \ldots, C_{t^{2^{d-1}}}$ with the respective circuits that compute them. This yields a circuit of depth $2d$, and by the inductive hypothesis, the total number of gates in this circuit is upper bounded by:

$$\begin{aligned}
\left(2^{t'} + 1\right) + \sum_{i=t+1}^{t^{2^{d-1}}} 2^i \cdot 2^{1+t^{2^{d-1}}} &\leq \left(2^{t^{2^d}} + 1\right) + \sum_{i=t+1}^{t^{2^{d-1}}} 2^i \cdot 2^{1+t^{2^{d-1}}} \\
&\leq 2^{t^{2^d}} + \sum_{i=1}^{t^{2^{d-1}}} 2^i \cdot 2^{1+t^{2^{d-1}}} \\
&\leq 2^{t^{2^d}} + 2^{1+t^{2^{d-1}}} \cdot 2^{1+t^{2^{d-1}}} \\
&= 2^{t^{2^d}} + 2^{2+2t^{2^{d-1}}} \\
&\leq 2^{t^{2^d}} + 2^{4t^{2^{d-1}}} \\
&\leq 2^{t^{2^d}} + 2^{t^{2+2^{d-1}}} \\
&\leq 2^{t^{2^d}} + 2^{t^{2^d}} \\
&= 2^{1+t^{2^d}}.
\end{aligned}$$

Above, the first inequality holds because $t' \leq t^{2^d}$; the second inequality simply expands the range of the sum (which certainly increases the sum by at least 1); the third inequality applies $\sum_{i=1}^{j} 2^i \leq 2^{j+1}$; and the remaining inequalities hold because $t \geq 2$ and $d \geq 2$. ☐

Note that Lemma 32 does not depend on the distribution of A and B, but only on the way C is defined recursively in terms of A and B. Hence, it holds for both $\mathcal{O}$ drawn from $\mathcal{D}$ or drawn from any D_i. We also note that the circuit given in Lemma 32 is not in $\mathsf{AC^0}$ normal form (i.e. it is not necessarily alternating and layered), but can be made so at the cost of a small increase in size.

With Lemma 32 in hand, the remainder of the proof is largely analogous to the proof of Theorem 26. Statement (a) essentially follows from the work of [Hås87]. Let $p(n)$ be a polynomial upper bound on the running time of M on inputs of length n, and also on the number of bits needed to specify $\langle M, 0^n \rangle$. This also means that M is restricted to query only $\mathcal{O}_1, \ldots, \mathcal{O}_{p(n)}$. Let $t = n_i$ and $t' = p(n_i)^2$. Then, by Lemma 32, for all sufficiently large i, there exists an $\mathsf{AC^0}\left[2^{p(n_i)^{O(1)}}, O(1)\right]$ circuit that takes as input A, B, and $C_1, \ldots, C_{n_i}$ and computes $M\left(0^{n_i}\right)$.

The key observation is that the only portion of the input to this circuit that can depend on whether 0^{n_i} is in $L^{\mathcal{O}}$ is the input to the size-2^{n_i} parity function that is encoded in A. All other portions of the input are sampled independently from this region under D_i:

1. The rest of A is sampled uniformly at random,
2. $B_1, \ldots, B_{n_i}$ and $C_1, \ldots, C_{n_i}$ are drawn from a distribution that cannot depend on any queries to A on inputs of length $\lfloor \sqrt{n_i} \rfloor$ or greater (so in particular, they cannot depend on A_{n_i}), and
3. $B_{n_i+1}, B_{n_i+2}, \ldots$ are sampled uniformly at random.

Hence, M is forced to evaluate the size-2^{n_i} parity function using only auxiliary and uncorrelated random bits. By Theorem 21, M can do this with probability greater than 0.6 for at most finitely many i. This even holds conditioned on $M^{\mathcal{O}}$ correctly deciding $0^{n_1}, \ldots, 0^{n_{i-1}}$, because the size-$2^{n_i}$ parity instance is chosen independently from the smaller instances, and because $M\left(0^{n_i}\right)$ can query the oracle on strings of length n_{i+1} or greater for at most finitely many values of i.

The proof of statement (b) is essentially the same as the proof of the analogous statement in Theorem 26: we use Lemma 32 to argue that there exists an $\mathsf{AC^0}\left[2^{\mathrm{poly}(n_i)}, O(1)\right]$ circuit that takes the truth table of a Forrelation instance as input and evaluates to $M^{\mathcal{O}}\left(0^{n_i}\right)$, and then we invoke Theorem 24 and a hybrid argument to argue that this circuit cannot detect the changes to B_m for $m > n_i$ when switching between $\mathcal{D}$ and D_i. □

4 Fine Control over BQP and PH

4.1 $\mathsf{BQP^{PH}}$ Lower Bounds for Sipser Functions

In this section, we prove that $\mathsf{BQP}^{\Sigma_k^{\mathsf{P}}}$ does not contain $\Sigma_{k+1}^{\mathsf{P}}$ relative to a random oracle, generalizing the known result that PH is infinite relative to a random oracle [HRST17, RST15].

We first require the following form of the BBBV Theorem [BBBV97]. It essentially states that a quantum algorithm that makes few queries to its input is unlikely to detect small random changes to the input. Viewed another way, Lemma 33 is just a probabilistic version of Lemma 25.

Lemma 33. *Consider a quantum algorithm Q that makes T queries to $x \in \{0,1\}^N$. Let $y \in \{0,1\}^N$ be drawn from some distribution such that, for all $i \in [N]$, $\Pr_y[x_i \neq y_i] \leq p$. Then for any $r > 0$:*

$$\Pr_y\left[\left|\Pr[Q(y) = 1] - \Pr[Q(x) = 1]\right| \geq r\right] \leq \frac{64pT^2}{r^2}$$

Proof. Let Q be the quantum query algorithm corresponding to f. By Lemma 25, we have that for any fixed y:

$$|\Pr[Q(x)=1] - \Pr[Q(y)=1]| \leq 8\sqrt{T} \cdot \sqrt{\sum_{i: x_i \neq y_i} q_i},$$

where we recall the definition of the *query magnitudes* q_i which depend on the algorithm's behavior on input x, and which satisfy $\sum_{i=1}^n q_i = T$. This implies that:

$$\begin{aligned}
\Pr_y[|\Pr[Q(y)=1] - \Pr[Q(x)=1]| \geq r] &\leq \Pr_y\left[8\sqrt{T} \cdot \sqrt{\sum_{i: x_i \neq y_i} q_i} \geq r\right] \\
&= \Pr_y\left[\sum_{i: x_i \neq y_i} q_i \geq \frac{r^2}{64T}\right] \\
&\leq \frac{64T}{r^2} \cdot \mathop{\mathbb{E}}_y\left[\sum_{i: x_i \neq y_i} q_i\right] \\
&= \frac{64T}{r^2} \cdot \sum_{i=1}^n q_i \cdot \Pr_y[y_i \neq x_i] \\
&\leq \frac{64pT^2}{r^2},
\end{aligned}$$

where the third line applies Markov's inequality (the q_is are nonnegative), and the last two lines use linearity of expectation along with the fact that the q_is sum to T. □

Corollary 34. *Let $f : \{0,1\}^N \to \{0,1,\perp\}$ be a partial function with $\mathsf{Q}(f) \leq T$. Fix $x \in \{0,1\}^N$, and let $y \in \{0,1\}^N$ be drawn from some distribution such that, for all $i \in [N]$, $\Pr_y[x_i \neq y_i] \leq p$. Then for some $i \in \{0,1\}$, $\Pr_y[f(y)=i] \leq 2304pT^2$.*

Proof. Let Q be the quantum query algorithm corresponding to f. We choose $i = 1$ if $\Pr[Q(x)=1] \leq \frac{1}{2}$, and $i = 0$ otherwise. Then the claim follows from Lemma 33 with $r = \frac{1}{6}$, just because Q computes f with error at most $\frac{1}{3}$. □

We now prove a query complexity version of the main result of this section.

Theorem 35. *Let $\textsc{Sipser}_d$ be the function defined in Theorem 22 for some choice of d, m, and N. Let $f : \{0,1\}^N \to \{0,1,\perp\}$ be computable by a depth-d circuit of size s in which the top gate has bounded-error quantum query complexity T, and all of the sub-circuits of the top gate are depth-$(d-1)$ AC^0 circuits. Then:*

$$\Pr_{x \sim \{0,1\}^N}[f(x) = \textsc{Sipser}_d(x)] \leq \frac{1}{2} + O\left(2^{-m/4}\right) + s2^{-2^{m/2-4}} + 2304T^2 2^{-m/2-4}.$$

The proof of this theorem largely follows Theorem 10.1 of [HRST17], the relevant parts of which are quoted in Theorem 22. We take a distribution $\rho \sim \mathcal{R}$ of random projections with the property that (a) $\mathcal{R}$ completes to the uniform distribution, (b) f_ρ simplifies with high probability over ρ, and (c) $\textsc{Sipser}_{d|\rho}$ retains structure with high probability over ρ. Essentially the only difference compared to [HRST17] is that we must incorporate the BBBV Theorem (in the form of Corollary 34) in order to argue (b).

Proof of Theorem 35. By Theorem 22(a),

$$\Pr_{x\sim\{0,1\}^N}[f(x)=\textsc{Sipser}_d(x)] = \Pr_{x\sim D,\rho\sim\mathcal{R}}\left[f_\rho(x)=\textsc{Sipser}_{d|\rho}(x)\right].$$

Theorem 22(b) and a union bound over all of the sub-circuits of the top gate imply that, except with probability at most $s2^{-2^{m/2-4}}$ over $\rho\sim\mathcal{R}$, f_ρ can be computed by a depth-2 circuit where the top gate has bounded-error quantum query complexity T, and all of the gates below the top gate have deterministic query complexity at most $2^{m/2-4}$. In this case, we say for brevity that f_ρ "simplifies". By Theorem 22(c) and a union bound, except with probability at most $O\left(2^{-m/2}\right)+s2^{-2^{m/4-2}}$ over $\rho\sim\mathcal{R}$, f_ρ simplifies and $\textsc{Sipser}_{d|\rho}$ is an AND of fan-in $(\ln 2)\cdot 2^m\cdot\left(1\pm O\left(2^{-m/4}\right)\right)$.

An AND of fan-in $(\ln 2)\cdot 2^m\cdot\left(1\pm O\left(2^{-m/4}\right)\right)$ evaluates to 1 with probability $\frac{1}{2}\left(1\pm O\left(2^{-m/4}\right)\right)$ on an input sampled from D. On the other hand, if f_ρ simplifies, then for each sub-circuit C of the top gate, $\Pr_{x\sim D}\left[C(x)\neq C\left(1^{|x|}\right)\right]\leq b2^{m/2-4}$, just because $\mathsf{D}(C)\leq 2^{m/2-4}$ and each bit of x is 0 with probability at most b. Hence, by Corollary 34 with $p=b2^{m/2-4}$, if f_ρ simplifies, then for some $i\in\{0,1\}$, $\Pr_{x\sim D}[f_\rho(x)=i]\leq 2304bT^2 2^{m/2-4}$. Since $b\leq 2^{-m}$, putting these together gives us that:

$$\Pr_{x\sim\mathcal{D},\rho\sim\mathcal{R}}\left[f_\rho(x)=\textsc{Sipser}_{d|\rho}(x)\right]\leq\frac{1}{2}+O\left(2^{-m/4}\right)+s2^{-2^{m/2-4}}+2304T^2 2^{-m/2-4}.$$ □

By standard techniques, we obtain the main result of this section, stated in terms of oracles instead of query complexity.

Corollary 36. *For all k, $\Sigma_{k+1}^{\mathsf{P}^{\mathcal{O}}}\not\subset\mathsf{BQP}^{\Sigma_k^{\mathsf{P}^{\mathcal{O}}}}$ with probability 1 over a random oracle $\mathcal{O}$.*

Proof sketch. Let $d=k+2$. Then this follows from Theorem 35 and the well-known connection between constant-depth circuits and the polynomial hierarchy [FSS84]. Analogous to [RST15], we take the unary language:

$$L^{\mathcal{O}}\coloneqq\left\{0^n:\textsc{Sipser}_d\left(\mathcal{O}(y^{1,n}),\mathcal{O}(y^{2,n}),\cdots,\mathcal{O}(y^{N,n}\right)=1\right\},$$

where $\textsc{Sipser}_d$ has N inputs, $n=\lceil\log_2 N\rceil$, and $y^{i,n}$ denotes the lexicographically ith string of length n. Then $L^{\mathcal{O}}$ is computable in $\Sigma_{d-1}^{\mathsf{P}^{\mathcal{O}}}$ [RST15, HRST17]. On the other hand, Theorem 35 with $m=\Theta(n)$, $T=\mathrm{poly}(n)$, and $s=2^{\mathrm{poly}(n)}$ implies that any $\mathsf{BQP}^{\Sigma_{d-2}^{\mathsf{P}^{\mathcal{O}}}}$ machine decides $L^{\mathcal{O}}(0^n)$ with probability at most $\frac{1}{2}+o(1)$ over $\mathcal{O}$, for infinitely many n, even conditioned on correctly deciding $L^{\mathcal{O}}(0^\ell)$ for all $\ell<n$. In particular, for every $\mathsf{BQP}^{\Sigma_{d-2}^{\mathsf{P}^{\mathcal{O}}}}$ machine, with probability 1 there exists *some* n on which it fails. As there are only countably many $\mathsf{BQP}^{\Sigma_{d-2}^{\mathsf{P}^{\mathcal{O}}}}$ machines, we conclude that $L^{\mathcal{O}}\notin\mathsf{BQP}^{\Sigma_{d-2}^{\mathsf{P}^{\mathcal{O}}}}$ with probability 1 over $\mathcal{O}$. □

4.2 $\mathsf{BQP}^{\mathsf{PH}}$ Lower Bounds for $\mathrm{OR}\circ\textsc{Forrelation}$

In this section, we use tail bounds on the sensitivity of AC^0 circuits to construct an oracle relative to which $\mathsf{NP}^{\mathsf{BQP}}\not\subset\mathsf{BQP}^{\mathsf{PH}}$. Such bounds are given implicitly in Section 3 of [GSTW16]. For completeness, we derive our own bound on the sensitivity tails of AC^0 circuits, though our bound is probably quantitatively suboptimal.

To prove that the sensitivity of AC^0 circuits concentrates well, the first ingredient we need is a random restriction lemma for AC^0 circuits. We use the following form, due to Rossman [Ros17].

Theorem 37 ([Ros17])**.** *Let $f \in \mathsf{AC}^0[s,d]$, and let ρ be a random restriction with $\Pr[*] = p$. Then for any $t > 0$:*

$$\Pr_{\rho}\left[\mathsf{D}(f_\rho) \geq t\right] \leq \left(p \cdot O\left(\log s\right)^{d-1}\right)^t.$$

With this in hand, it is straightforward to derive our sensitivity tail bound.

Lemma 38. *Let $f : \{0,1\}^N \to \{0,1\}$ be a circuit in $\mathsf{AC}^0[s,d]$. Then for any $t > 0$,*

$$\Pr_{x \sim \{0,1\}^N}\left[\mathsf{s}^x(f) \geq t\right] \leq 2N \cdot 2^{-\Omega\left(\frac{t}{(\log s)^{d-1}}\right)}.$$

Proof. Let ρ be a random restriction with $\Pr[*] = p$, for some p to be chosen later. It will be convenient to view the choice of ρ as follows: we choose a string $x \in \{0,1\}^N$ uniformly at random, and then we choose a set $S \subseteq [N]$ wherein each $i \in [N]$ is included in S independently with probability p. Then, we take ρ to be:

$$\rho(i) = \begin{cases} * & i \in S \\ x_i & i \notin S. \end{cases}$$

Thus, by definition, it holds that $f_\rho(x|_S) = f(x)$.

Observe that for any fixed $x \in \{0,1\}^N$ and $j > 0$, we have:

$$\begin{aligned} \mathsf{s}^x(f) &= \frac{1}{p}\mathop{\mathbb{E}}_{S}\left[\mathsf{s}^{x|_S}(f_\rho)\right] \\ &\leq \frac{1}{p}\mathop{\mathbb{E}}_{S}\left[\mathsf{s}(f_\rho)\right] \\ &\leq \frac{1}{p}\mathop{\mathbb{E}}_{S}\left[\mathsf{D}(f_\rho)\right] \\ &\leq \frac{1}{p}\cdot\left(j + N \cdot \Pr_{S}\left[\mathsf{D}(f_\rho) \geq j\right]\right), \end{aligned} \tag{2}$$

where the first line holds because each sensitive bit of f on x is kept unrestricted with probability p; the second line holds by the definition of sensitivity; the third line holds by known relations between query measures; and the last line holds because $\mathsf{D}(f_\rho) \leq N$ always holds.

With this in hand, we derive:

$$\begin{aligned} \Pr_{x}\left[\mathsf{s}^x(f) \geq t\right] &\leq \Pr_{x}\left[\frac{1}{p}\cdot\left(j + N \cdot \Pr_{S}\left[\mathsf{D}(f_\rho) \geq j\right]\right) \geq t\right] \\ &= \Pr_{x}\left[\Pr_{S}\left[\mathsf{D}(f_\rho) \geq j\right] \geq \frac{pt - j}{N}\right] \\ &\leq \Pr_{x,S}\left[\mathsf{D}(f_\rho) \geq j\right] \cdot \frac{N}{pt - j} \\ &\leq \left(p \cdot O\left(\log s\right)^{d-1}\right)^j \cdot \frac{N}{pt - j}. \end{aligned}$$

Above, the first line applies (2); the third line holds by Markov's inequality; and the last line applies Theorem 37.

Choose $p = O\left(\log s\right)^{1-d}$ so that the above expression simplifies to $2^{-j} \cdot \frac{N}{pt-j}$. Then, set $j = pt - 1$ and the corollary follows. □

The sensitivity tail bound above immediately implies a tail bound on the *block* sensitivity of $\mathsf{AC^0}$ circuits. We thank Avishay Tal for providing us with a proof of this fact.

Corollary 39. *Let $f : \{0,1\}^N \to \{0,1\}$ be a circuit in $\mathsf{AC^0}[s,d]$, and let $B = \{B_1, B_2, \ldots, B_k\}$ be a collection of disjoint subsets of $[N]$. Then for any t,*

$$\Pr_{x\sim\{0,1\}^N}\left[\mathsf{bs}_B^x(f) \geq t\right] \leq 4N \cdot 2^{-\Omega\left(\frac{t}{(\log(s+N))^d}\right)}.$$

Proof. Consider the function $g : \{0,1\}^{N+k}$ defined by

$$g(y,z) \coloneqq f\left(y \oplus z_1 \cdot B_1 \oplus z_2 \cdot B_2 \oplus \ldots \oplus z_k \cdot B_k\right),$$

where $z_i \cdot B_i$ denotes the all zeros string if $z_i = 0$, and otherwise is the indicator string of B_i.

We claim that $g \in \mathsf{AC^0}[s + O(N), d+1]$. Let $x = y \oplus z_1 \cdot B_1 \oplus z_2 \cdot B_2 \oplus \ldots \oplus z_k \cdot B_k$. Notice that each bit of x is either a bit in y, or else the XOR of a bit of y with a bit of z. Hence, we can compute g by feeding in at most N XOR gates and their negations into f. The XOR function can be written as either an OR of ANDs or an AND of ORs: $a \oplus b = (a \vee b) \wedge (\neg a \vee \neg b) = (a \wedge \neg b) \vee (\neg a \wedge b)$. Hence, we can absorb one layer of AND or OR gates into the bottom layer of the circuit that computes f, thus obtaining a circuit of depth $d+1$.

Notice that for any $x \in \{0,1\}^N$, there are exactly 2^k strings $(y,z) \in \{0,1\}^{N+k}$ such that $x = y \oplus z_1 \cdot B_1 \oplus z_2 \cdot B_2 \oplus \ldots \oplus z_k \cdot B_k$. Moreover, for any such (y,z) we have that $\mathsf{bs}_B^x(f) \leq \mathsf{s}^{(y,z)}(g)$. Thus, Lemma 38 implies that:

$$\Pr_{x\sim\{0,1\}^N}\left[\mathsf{bs}_B^x(f) \geq t\right] \leq \Pr_{(y,z)\sim\{0,1\}^{N+k}}\left[\mathsf{s}^{(y,z)}(g) \geq t\right] \leq 2(N+k) \cdot 2^{-\Omega\left(\frac{t}{(\log(s+N))^d}\right)},$$

and the corollary follows because $k \leq N$. □

The rough idea of the proof going forward is as follows: an $\mathsf{NP^{BQP}}$ algorithm can easily distinguish (1) a uniformly random $M \times N$ array of bits, and (2) an $M \times N$ array which contains a single row drawn from the Forrelation distribution, and is otherwise random. We want to show that a $\mathsf{BQP^{PH}}$ algorithm cannot distinguish (1) and (2). To prove this, we first use our block sensitivity tail bound to argue in Lemma 40 below that for most uniformly random strings x, an $\mathsf{AC^0}$ circuit is unlikely to detect a change to x made by uniformly randomly resampling a single row of x. Then, we use the Raz-Tal Theorem to argue in Lemma 41 that the same holds if we instead resample a single row of x from the Forrelation distribution, rather than the uniform distribution. Finally, in Theorem 42 we apply the BBBV Theorem to argue that a $\mathsf{BQP^{PH}}$ query algorithm cannot distinguish cases (1) and (2).

Lemma 40. *Let $f : \{0,1\}^{MN} \to \{0,1\}$ be a circuit in $\mathsf{AC^0}[s,d]$. Let $x \in \{0,1\}^{MN}$ be an input, viewed as an $M \times N$ array with M rows and N columns. Let y be sampled depending on x as follows: uniformly select one of the rows of x, randomly reassign all of the bits of that row, and leave the other rows of x unchanged. Then for any $\varepsilon > 0$:*

$$\Pr_{x\sim\{0,1\}^{MN}}\left[\Pr_y\left[f(x) \neq f(y)\right] \geq \varepsilon\right] \leq 8M^2N \cdot 2^{-\Omega\left(\frac{\varepsilon M}{(\log(s+MN))^d}\right)}.$$

Proof. Let $\mathcal{B}$ be the distribution over collections $B = \{S_1, \ldots, S_M\}$ of subsets of $[MN]$ wherein each S_i is a uniformly random subset of the ith row. Notice that for any fixed $x \in \{0,1\}^{MN}$ and

$j > 0$, we have:

$$
\begin{aligned}
\Pr_y[f(x) \neq f(y)] &= \frac{1}{M} \cdot \mathop{\mathbb{E}}_{B \sim \mathcal{B}}[\mathsf{bs}_B^x(f)] \\
&\leq \frac{1}{M} \cdot \left(j + M \cdot \Pr_{B \sim \mathcal{B}}[\mathsf{bs}_B^x(f) \geq j]\right) \\
&= \frac{j}{M} + \Pr_{B \sim \mathcal{B}}[\mathsf{bs}_B^x(f) \geq j], \qquad (3)
\end{aligned}
$$

just because one can sample y by drawing $B \sim \mathcal{B}$, $i \sim [M]$, and taking $y = x^{\oplus S_i}$. The inequality in the second line holds because $\mathsf{bs}_B^x(f) \leq |B| = M$.

With this in hand, we derive:

$$
\begin{aligned}
\Pr_{x \sim \{0,1\}^{MN}}\left[\Pr_y[f(x) \neq f(y)] \geq \varepsilon\right] &\leq \Pr_x\left[\frac{j}{M} + \Pr_{B \sim \mathcal{B}}[\mathsf{bs}_B^x(f) \geq j] \geq \varepsilon\right] \\
&= \Pr_x\left[\Pr_{B \sim \mathcal{B}}[\mathsf{bs}_B^x(f) \geq j] \geq \varepsilon - \frac{j}{M}\right] \\
&\leq \Pr_{x,B}[\mathsf{bs}_B^x(f) \geq j] \cdot \frac{M}{\varepsilon M - j} \\
&\leq 4MN \cdot 2^{-\Omega\left(\frac{j}{(\log(s+MN))^d}\right)} \cdot \frac{M}{\varepsilon M - j} \\
&\leq \frac{4M^2N}{\varepsilon M - j} \cdot 2^{-\Omega\left(\frac{j}{(\log(s+MN))^d}\right)}.
\end{aligned}
$$

Above, the first line applies (3); the third line holds by Markov's inequality; and the fourth line applies Corollary 39. Choosing $j = \varepsilon M - 1$ completes the proof. □

Lemma 41. *Let $M \leq \mathrm{quasipoly}(N)$, and suppose that $f : \{0,1\}^{MN} \to \{0,1\}$ is a circuit in $\mathsf{AC}^0[\mathrm{quasipoly}(N), O(1)]$. Let $x \in \{0,1\}^{MN}$ be an input, viewed as an $M \times N$ array with M rows and N columns. Let y be sampled depending on x as follows: uniformly select one of the rows of x, randomly sample that row from the Forrelation distribution $\mathcal{F}_N$, and leave the other rows of x unchanged. Then for some $\varepsilon = \frac{\mathrm{polylog}(N)}{\sqrt{N}}$, we have:*

$$
\Pr_{x \sim \{0,1\}^{MN}}\left[\Pr_y[f(x) \neq f(y)] \geq \varepsilon\right] \leq 8M^2N \cdot 2^{-\Omega\left(\frac{M}{\sqrt{N}\mathrm{polylog}(N)}\right)}.
$$

Proof. Consider a Boolean function $C(x, z, i)$ that takes inputs $x \in \{0,1\}^{MN}$, $z \in \{0,1\}^N$, and $i \in [M]$. Let $\tilde{y}$ be the string obtained from x by replacing the ith row with z. Let C output 1 if $f(x) \neq f(\tilde{y})$, and 0 otherwise. Clearly, $C \in \mathsf{AC}^0[\mathrm{quasipoly}(N), O(1)]$. Observe that for any fixed x:

$$
\Pr_{i \sim [M], z \sim \mathcal{F}_N}[C(x, z, i) = 1] = \Pr_y[f(x) \neq f(y)]. \qquad (4)
$$

By Theorem 24, for some $\varepsilon = \frac{\mathrm{polylog}(N)}{\sqrt{N}}$ we have:

$$
\left|\Pr_{i \sim [M], z \sim \mathcal{F}_N}[C(x, z, i) = 1] - \Pr_{i \sim [M], z \sim \{0,1\}^N}[C(x, z, i) = 1]\right| \leq \frac{\varepsilon}{2}. \qquad (5)
$$

Putting these together, we obtain:

$$\begin{aligned}
\Pr_{x\sim\{0,1\}^{MN}}\left[\Pr_{y}\left[f(x)\neq f(y)\right]\geq\varepsilon\right] &= \Pr_{x\sim\{0,1\}^{MN}}\left[\Pr_{i\sim[M],z\sim\mathcal{F}_N}\left[C(x,z,i)=1\right]\geq\varepsilon\right]\\
&\leq \Pr_{x\sim\{0,1\}^{MN}}\left[\Pr_{i\sim[M],z\{0,1\}^N}\left[C(x,z,i)=1\right]\geq\frac{\varepsilon}{2}\right]\\
&= \Pr_{x\sim\{0,1\}^{MN}}\left[\Pr_{i\sim[M],z\sim\{0,1\}^N}\left[f(x)\neq f(\tilde{y})\right]\geq\frac{\varepsilon}{2}\right]\\
&\leq 8M^2N\cdot 2^{-\Omega\left(\frac{\varepsilon M}{(\log(s+MN))^d}\right)}\\
&\leq 8M^2N\cdot 2^{-\Omega\left(\frac{M}{\sqrt{N}\mathrm{polylog}(N)}\right)},
\end{aligned}$$

where the first line substitutes (4); the second line holds by (5) and the triangle inequality; the third line holds by the definition of C and $\tilde{y}$ in terms if i and z; the fourth line invokes Lemma 40 for some $s=\mathrm{quasipoly}(N)$ and $d=O(1)$; and the last line uses these bounds on s and d along with the assumption that $M\leq\mathrm{quasipoly}(N)$. □

The next theorem essentially shows that no $\mathsf{BQP}^{\mathsf{PH}}$ query algorithm can solve the OR ∘ FORRELATION problem (i.e. given a list of FORRELATION instances, decide if one of them is Forrelated, or if they are all uniform).

Theorem 42. *Let $M\leq\mathrm{quasipoly}(N)$, and let $f:\{0,1\}^{MN}\to\{0,1,\perp\}$ be computable by a circuit of size $\mathrm{quasipoly}(N)$ in which the top gate has bounded-error quantum query complexity T, and all of the sub-circuits of the top gate are $\mathsf{AC}^0[\mathrm{quasipoly}(N),O(1)]$ circuits.*

Let $b\sim\{0,1\}$ be a uniformly random bit. Suppose $z\in\{0,1\}^{MN}$ is sampled such that:

- *If $b=0$, then z is uniformly random.*
- *If $b=1$, then a single uniformly chosen row of z is sampled from the Forrelation distribution $\mathcal{F}_N$, and the remaining $M-1$ rows of z are uniformly random.*

Then:

$$\Pr_{b,z}[f(z)=b]\leq\frac{1}{2}+\mathrm{quasipoly}(N)\cdot 2^{-\Omega\left(\frac{M}{\sqrt{N}\mathrm{polylog}(N)}\right)}+\frac{T^2\mathrm{polylog}(N)}{\sqrt{N}}.$$

Proof. We can think of sampling z as follows. First, we choose a string $x_0\sim\{0,1\}^{MN}$. Then, we sample x_1 by uniformly at random replacing one of the rows of x_0 with a sample from $\mathcal{F}_N$. Finally, we sample $b\sim\{0,1\}$ and set $z=x_b$.

Call a fixed x_0 "bad" if, for one of the sub-circuits C of the top gate, we have $\Pr_{x_1}[C(x_0)\neq C(x_1)]\geq\varepsilon$, where $\varepsilon\leq\frac{\mathrm{polylog}(N)}{\sqrt{N}}$ is the parameter given in Lemma 41. Lemma 41, combined with a union bound over the $\mathrm{quasipoly}(N)$-many such sub-circuits, implies that:

$$\Pr_{x_0\sim\{0,1\}^{MN}}[x_0\text{ is bad}]\leq\mathrm{quasipoly}(N)\cdot 2^{-\Omega\left(\frac{M}{\sqrt{N}\mathrm{polylog}(N)}\right)}.$$

Clearly, it holds that:

$$\Pr_{b,z}[f(z)=b]\leq\Pr_{x_0\sim\{0,1\}^{MN}}[x_0\text{ is bad}]+\Pr_{x_1,b}\left[f(x_b)=b|x_0\text{ is good}\right].$$

b is uniformly random, even conditioned on x_0 being good. On the other hand, Corollary 34 implies that for some $i \in \{0,1\}$ (depending on x_0), $\Pr_{x_1,b}[f(x_b) = i | x_0 \text{ is good}] \leq 2304\varepsilon T^2$. Thus, it holds that:

$$\Pr_{x_1,b}[f(x_b) = b | x_0 \text{ is good}] \leq \frac{1}{2} + \frac{T^2 \text{polylog}(N)}{\sqrt{N}}.$$

Putting these bounds together implies the statement of the theorem. □

Via standard complexity-theoretic techinques, Theorem 42 implies the following oracle separation, which resolves the question of Fortnow [For05].

Corollary 43. *There exists an oracle relative to which* $\mathsf{NP}^{\mathsf{BQP}} \not\subset \mathsf{BQP}^{\mathsf{PH}}$.

Proof sketch. We construct an oracle A as follows. Let L^A be a uniformly random unary language. For each $n \in \mathbb{N}$, add into A a function $f_n : \{0,1\}^{2n^2} \to \{0,1\}$ as follows:

- If $L^A(0^n) = 0$, then f_n is uniformly random.
- If $L^A(0^n) = 1$, then viewing the truth table of f_n as consisting of 2^{n^2} rows of length 2^{n^2}, we pick a single row at random and sample it from the Forrelation distribution $\mathcal{F}_{2^{t^2}}$, and sample the remaining $2^{n^2} - 1$ rows from the uniform distribution.

We first show that $L^A \in \mathsf{NP}^{\mathsf{BQP}^A}$ with probability 1 over A. To do so, we define a language M^A as follows: for a string $x \in \{0,1\}^*$, $M^A(x) = 1$ if $|x| = n^2$ and the xth row of f_n was drawn from the Forrelation distribution; otherwise $M^A(x) = 0$. Clearly, $L^A \in \mathsf{NP}^{M^A}$: to determine if $0^n \in L^A$, nondeterministically guess a string $x \in \{0,1\}^{n^2}$ and check if $x \in M^A$. Thus, it suffices to show that $M^A \in \mathsf{BQP}^A$, which we do below:

Claim 44. $M^A \in \mathsf{BQP}^A$ *with probability* 1 *over* A.

Proof of Claim. The proof is essentially the same as Claim 27, just with slightly different parameters. We use the algorithm $\mathcal{A}$ from Theorem 24 for distinguishing the Forrelated and uniform distributions to decide M^A. For the last step in which we apply the Borel-Cantelli Lemma to argue that, with probability 1 over A, $\mathcal{A}$ correctly decides $M^A(x)$ for all but finitely many $x \in \{0,1\}^*$, we have:

$$\sum_{x \in \{0,1\}^*} \Pr_A\left[\mathcal{A}^A \text{ does not decide } M^A(x)\right] \leq \sum_{n=1}^{\infty} \sum_{x \in \{0,1\}^{n^2}} 3 \cdot 2^{-2n^2} \leq \sum_{n=1}^{\infty} 2^{n^2} \cdot 3 \cdot 2^{-2n^2} < \infty. \quad \square$$

It remains to show that $L^A \notin \mathsf{BQP}^{\mathsf{PH}^A}$ with probability 1 over A. By analogy with Corollary 36, this follows from Theorem 42 with $M = N = 2^{n^2}$ and $T = \text{poly}(n)$, which shows that any $\mathsf{BQP}^{\mathsf{PH}^A}$ machine will correctly decide $L^A(0^n)$ with probability at most $\frac{1}{2} + o(1)$ for infinitely many n. □

Using techniques analogous to Theorem 29, we obtain the following stronger oracle separation.

Corollary 45. *There exists an oracle relative to which* $\mathsf{P} = \mathsf{NP}$ *but* $\mathsf{BQP} \neq \mathsf{QCMA}$.

Proof sketch. This oracle $\mathcal{O}$ will consist of two parts: an oracle A drawn from the same distribution as the oracle A in Corollary 43, and an oracle B that we will construct inductively. For each $t \in \mathbb{N}$, we add a region B_t that will depend on the previously constructed parts of the oracle. For convenience, we denote by A_t the region of A corresponding to inputs of length t, and write $\mathcal{O}_t = (A_t, B_t)$.

Similarly to Theorem 29, we define S_t as the set of all NP machines that take less than t bits to specify, run in at most t steps, and query only the $\mathcal{O}_1, \ldots, \mathcal{O}_{\lfloor\sqrt{t}\rfloor}$ regions of the oracle. Then, we encode into B_t the answers to all machines in S_t. This has the effect of making $\mathsf{P}^{\mathcal{O}} = \mathsf{NP}^{\mathcal{O}}$, as a polynomial-time algorithm can decide the behavior of any $\mathsf{NP}^{\mathcal{O}}$ machine M by looking up the relevant bit in B that encodes M's behavior.

It remains to show that $\mathsf{BQP}^{\mathcal{O}} \neq \mathsf{QCMA}^{\mathcal{O}}$ with probability 1 over $\mathcal{O}$. We achieve this by taking the language L^A defined in Corollary 43, which is clearly in $\mathsf{QCMA}^{\mathcal{O}}$, and showing that $L^A \notin \mathsf{BQP}$ with probability 1 over $\mathcal{O}$.

Analogous to Lemma 32, one can show that for any $t' \leq \mathrm{poly}(t)$, any bit of $B_{t'}$ can be computed by an $\mathsf{AC^0}\left[2^{\mathrm{poly}(t)}, O(1)\right]$ circuit whose inputs depend only on A and $B_1, B_2, \ldots, B_t$. Hence, any $\mathsf{BQP}^{\mathcal{O}}$ machine that runs in time $\mathrm{poly}(t)$ can be computed by a circuit of size $2^{\mathrm{poly}(t)}$ in which the top gate has bounded-error quantum query complexity $\mathrm{poly}(t)$, all of the sub-circuits of the top gate are $\mathsf{AC^0}\left[2^{\mathrm{poly}(t)}, O(1)\right]$ circuits, and the inputs are A and $B_1, B_2, \ldots, B_t$. In particular, if $t = n$, then all of the inputs are uncorrelated with $L^A(0^n)$, except for A_{2n^2}, the region whose $\mathrm{OR} \circ \textsc{Forrelation}$ instance encodes $L^A(0^n)$. But in that case, we can again appeal to Theorem 42 with $M = N = 2^{n^2}$ and $T = \mathrm{poly}(n)$ to argue that such an algorithm correctly decides $L^A(0^n)$ with probability at most $\frac{1}{2} + o(1)$ for infinitely many n. ☐

4.3 $\mathsf{PH}^{\mathsf{BQP}}$ Lower Bounds for $\textsc{Forrelation} \circ \mathrm{OR}$

In this section, we construct an oracle relative to which $\mathsf{BQP}^{\mathsf{NP}} \not\subset \mathsf{PH}^{\mathsf{PromiseBQP}}$.

Within this section, for a string $z \in \{0,1\}^M$ and some choice of N, let $\mathcal{D}_{z,N}$ denote the following distribution over $\{0,1\}^{MN}$. View $x \sim \mathcal{D}_{z,N}$ as an $M \times N$ array of bits sampled as follows: if $z_i = 0$, then the ith row of x is all 0s, while if $z_i = 1$, then the ith row of x has a single 1 chosen uniformly at random and 0s everywhere else.

Our first key lemma shows that, for a string $x \sim \mathcal{D}_{z,N}$, a quantum algorithm that queries x can be efficiently simulated by a classical query algorithm, with high probability over x. We start with a version of this lemma in which the quantum algorithm makes only a single query.

Lemma 46. *Consider a quantum algorithm Q that makes 1 query to $x \in \{0,1\}^{MN}$ to produce a state $|\psi\rangle$. Then for any $K \in \mathbb{N}$, there exists a deterministic classical algorithm that makes K queries to x, and outputs a description of a state $|\varphi\rangle$ such that for any $\alpha \geq \sqrt{\frac{8}{N}}$ and any $z \in \{0,1\}^M$:*

$$\Pr_{x \sim \mathcal{D}_{z,N}}\left[\| |\psi\rangle - |\varphi\rangle \| \geq \alpha\right] \leq e^{-\frac{\alpha^4 K}{32}}.$$

Proof. Analogous to Lemma 25, let $q_{i,j}$ be the query magnitude (i.e. probability) with which Q queries $x_{i,j}$ during its single query. That is, if the initial state $|\psi_0\rangle$ of Q has the form:

$$|\psi_0\rangle = \sum_{i=1}^{M} \sum_{j=1}^{N} \alpha_{i,j,w} |i, j, w\rangle,$$

where w are indices over a workspace register, then

$$q_{i,j} \coloneqq \sum_w |\alpha_{i,j,w}|^2,$$

so that $\sum_{i=1}^M \sum_{j=1}^N q_{i,j} = 1$.

The classical algorithm is simply the following: query every $x_{i,j}$ such that $q_{i,j} \geq \frac{1}{K}$. Clearly there are at most K such bits, so the algorithm makes at most K queries. Then, calculate Q's post-query state, assuming that all of the unqueried bits are 0. Call this state $|\varphi\rangle$.

We now argue that the classical algorithm achieves the desired approximation to $|\psi\rangle$ with the correct probability. Fix some $z \in \{0,1\}^M$. For each row i with $z_i = 1$, let $j(i,x)$ be the unique column j such that $x_{i,j} = 1$. Now define a random variable $w(i,x)$ that measures the contribution of row i to the error of our classical simulation:

$$w(i,x) := \begin{cases} q_{i,j(i,x)} & z_i = 1 \text{ and } q_{i,j(i,x)} < \frac{1}{K}, \\ 0 & \text{otherwise.} \end{cases} \tag{6}$$

Note that the $w(i,x)$'s are independent random variables, and also satisfy

$$\begin{aligned} \mathop{\mathbb{E}}_x \left[\sum_{i=1}^{M} w(i,x) \right] &\leq \mathop{\mathbb{E}}_x \left[\sum_{i:z_i=1} q_{i,j(i,x)} \right] \\ &\leq \mathop{\mathbb{E}}_x \left[\frac{1}{N} \sum_{i=1}^{M} \sum_{j=1}^{N} q_{i,j} \right] \\ &\leq \frac{1}{N} \end{aligned}$$

by (6) and linearity of expectation. We also have $w(i,x) \leq \frac{1}{K}$ for all i, but we will actually need a stronger upper bound: namely $w(i,x) \leq m_i$, where

$$m_i := \min \left\{ \frac{1}{K},\ \max_j q_{i,j} \right\}.$$

Note that $m_i \leq \frac{1}{K}$ for all i, and also that

$$\sum_{i=1}^{M} m_i \leq \sum_{i=1}^{M} \sum_{j=1}^{N} q_{i,j} = 1,$$

which together imply that

$$\sum_{i=1}^{M} m_i^2 \leq \sum_{i=1}^{M} m_i \cdot \frac{1}{K} \leq \frac{1}{K}. \tag{7}$$

Recall that we wish to bound the distance between $|\psi\rangle$ and $|\varphi\rangle$. (6) implies that

$$\| \, |\psi\rangle - |\varphi\rangle \, \| = 2\sqrt{\sum_{i=1}^{M} w(i,x)}, \tag{8}$$

and therefore

$$\Pr_{x \sim \mathcal{D}_{z,N}} \left[\| \, |\psi\rangle - |\varphi\rangle \, \| \geq \alpha \right] = \Pr_{x \sim \mathcal{D}_{z,N}} \left[\sum_{i=1}^{M} w(i,x) \geq \frac{\alpha^2}{4} \right].$$

We finally appeal to Hoeffding's inequality (Fact 16) to bound this quantity. Set $\mu := \frac{1}{N}$ and $\delta := \frac{\alpha^2 N}{4} - 1$. Then

$$\begin{aligned}
\Pr_{x \sim \mathcal{D}_{z,N}}\left[\| \ket{\psi} - \ket{\varphi} \| \geq \alpha\right] &= \Pr_{x \sim \mathcal{D}_{z,N}}\left[\sum_{i=1}^{M} w(i,x) \geq (1+\delta)\mu\right] \\
&\leq \exp\left(-\frac{2\delta^2\mu^2}{\sum_{i=1}^{M} m_i^2}\right) \\
&\leq \exp\left(-\frac{2\left(\frac{\alpha^2 N}{4} - 1\right)^2 \left(\frac{1}{N}\right)^2}{\frac{1}{K}}\right) \\
&\leq \exp\left(-\frac{2\left(\frac{\alpha^2 N}{8}\right)^2 \left(\frac{1}{N}\right)^2}{\frac{1}{K}}\right) \\
&= e^{-\frac{\alpha^2 K}{32}},
\end{aligned}$$

where the first line applies (8); the second line applies Fact 16; the third line substitutes (7); and the fourth line uses the assumption that $\alpha \geq \sqrt{\frac{8}{N}}$. □

Next, by repeated application of Lemma 46, we generalize Lemma 46 to quantum algorithms that make multiple queries.

Lemma 47. *Consider a quantum algorithm Q that makes T queries to $x \in \{0,1\}^{MN}$ to produce a state $\ket{\psi_T}$. Then for any $K \in \mathbb{N}$, there exists a classical algorithm that makes KT queries to x, and outputs a description of a state $\ket{\varphi_T}$ such that for any $\alpha \geq \sqrt{\frac{8}{N}}$ and any $z \in \{0,1\}^M$:*

$$\Pr_{x \sim \mathcal{D}_{z,N}}\left[\| \ket{\psi_T} - \ket{\varphi_T} \| \geq \alpha T\right] \leq T \cdot e^{-\frac{\alpha^4 K}{32}}.$$

Proof. View Q as a sequence of T 1-query algorithms $Q_1, Q_2, \ldots, Q_T$, where each Q_t is a unitary transformation. For $t \leq T$, define $\ket{\psi_t}$ as the state of Q after t queries. This is to say, if the initial state of Q is $\ket{\psi_0}$, then for $t > 0$, $\ket{\psi_t} = Q_t \ket{\psi_{t-1}}$.

Intuitively speaking, the classical simulation algorithm simply applies the algorithm from Lemma 46 T times consecutively. In slightly more detail, let $\ket{\varphi_0} := \ket{\psi_0}$. For $t > 0$, define $\ket{\varphi_t}$ inductively as the state obtained after applying the classical algorithm from Lemma 46 corresponding to $Q_t \ket{\psi_{t-1}}$. Then clearly $\ket{\varphi_T}$ is computable by a classical algorithm that makes KT queries to x.

It remains to show that $\ket{\psi_T}$ and $\ket{\varphi_T}$ are close with high probability. For $t \leq T$, define $\ket{\gamma_t}$ as the state obtained by applying the classical algorithm for the first t steps and the quantum algorithm for the remaining $T - t$ steps, i.e.

$$\ket{\gamma_t} = Q_T Q_{T-1} \cdots Q_{t+1} \ket{\varphi_t}.$$

Note that $|\gamma_0\rangle = |\psi_T\rangle$ and $|\gamma_T\rangle = |\varphi_T\rangle$. From this, we may bound:

$$\begin{aligned} || \, |\psi_T\rangle - |\varphi_T\rangle \, || &= || \, |\gamma_0\rangle - |\gamma_T\rangle \, || \\ &\leq \sum_{t=1}^{T} || \, |\gamma_{t-1}\rangle - |\gamma_t\rangle \, || \\ &= \sum_{t=1}^{T} ||Q_t \, |\varphi_{t-1}\rangle - |\varphi_t\rangle \, ||, \end{aligned} \tag{9}$$

where the second line holds by the triangle inequality, and the last line holds because the Q_t's are unitary transformations. Lemma 46 implies that all of the terms in this sum are bounded with high probability. In particular, we conclude:

$$\begin{aligned} \Pr_{x \sim \mathcal{D}_{z,N}} [|| \, |\psi_T\rangle - |\varphi_T\rangle \, || \geq \alpha T] &\leq \Pr_{x \sim \mathcal{D}_{z,N}} \left[\sum_{t=1}^{T} ||Q_t \, |\varphi_{t-1}\rangle - |\varphi_t\rangle \, || \geq \alpha T \right] \\ &\leq \sum_{t=1}^{T} \Pr_{x \sim \mathcal{D}_{z,N}} [||Q_t \, |\varphi_{t-1}\rangle - |\varphi_t\rangle \, || \geq \alpha] \\ &\leq T \cdot e^{-\frac{\alpha^4 k}{32}}, \end{aligned}$$

where the first line applies (9), the second line holds by a union bound, and the last line holds by Lemma 46. □

The next theorem is essentially just a restatement of Lemma 47 with cleaner parameters. It can be understood as a version of the Aaronson-Ambainis conjecture [AA14, Conjecture 1.5] for sparse oracles.

Theorem 48. *Consider a quantum algorithm Q that makes T queries to $x \in \{0,1\}^{MN}$. Then for any $\varepsilon \geq 4T\sqrt{\frac{8}{N}}$ and $\delta > 0$, there exists a classical algorithm that makes $O\left(\frac{T^5}{\varepsilon^4} \log \frac{T}{\delta}\right)$ queries to x, and that outputs an estimate p such that for any $z \in \{0,1\}^M$:*

$$\Pr_{x \sim \mathcal{D}_{z,N}} [|\Pr[Q(x) = 1] - p| \geq \varepsilon] \leq \delta.$$

Proof. Let Q be the quantum algorithm corresponding to f, and let $|\psi\rangle$ be the output state of Q on input x immediately before measurement. For some K to be chosen later, consider the classical algorithm corresponding to Q from Lemma 47 that makes KT queries and produces a classical description of a quantum state $|\varphi\rangle$ on input x. Let p be the probability that the first bit of $|\varphi\rangle$ is measured to be 1 in the computational basis.

[BV97, Lemma 3.6] tells us that on any input x:

$$|\Pr[Q(x) = 1] - p| \leq 4|| \, |\psi\rangle - |\varphi\rangle \, ||.$$

Choose $\alpha = \frac{\varepsilon}{4T}$, which, by the assumption of the theorem, must also satisfy $\alpha \geq \sqrt{\frac{8}{N}}$. By appealing to Lemma 47, we conclude that

$$\begin{aligned} \Pr_{x \sim \mathcal{D}_{z,N}} [|\Pr[Q(x) = 1] - p| \geq \varepsilon] &\leq \Pr_{x \sim \mathcal{D}_{z,N}} \left[|| \, |\psi\rangle - |\varphi\rangle \, || \geq \frac{\varepsilon}{4} \right] \\ &= \Pr_{x \sim \mathcal{D}_{z,N}} [|| \, |\psi\rangle - |\varphi\rangle \, || \geq \alpha T] \\ &\leq T \cdot e^{-\frac{\alpha^4 K}{32}}. \end{aligned}$$

Thus, we just need to choose K such that $T \cdot e^{-\frac{\alpha^4 K}{32}} \leq \delta$, or equivalently:

$$\frac{\alpha^4 K}{32} \geq \log T + \log \frac{1}{\delta}.$$

Choosing $K = O\left(\frac{T^4}{\varepsilon^4} \log \frac{T}{\delta}\right)$ completes the theorem, as the classical algorithm makes KT queries. ☐

As a straightforward corollary, we obtain the following functional version of Theorem 48.

Corollary 49. *Let $f : \{0,1\}^{MN} \to \{0,1,\perp\}$ be a function with $\mathsf{Q}(f) \leq T$ for some $T \leq \sqrt{\frac{N}{4608}}$. Then for any $\delta > 0$, there exists a function $g : \{0,1\}^{MN} \to \{0,1\}$ with $\mathsf{D}(g) \leq O\left(T^5 \log \frac{T}{\delta}\right)$ such that for any $z \in \{0,1\}^M$:*

$$\Pr_{x \sim \mathcal{D}_{z,N}} [f(x) \in \{0,1\} \text{ and } f(x) \neq g(x)] \leq \delta.$$

Proof. Let Q be the quantum query algorithm corresponding to f. Choose $\varepsilon = \frac{1}{6}$, and consider running the classical algorithm from Theorem 48 that produces an estimate p of Q's acceptance probability. (The condition of Theorem 48 is satisfied because $4T\sqrt{\frac{8}{N}} \leq \frac{1}{6}$).

Define g by:

$$g(x) = \begin{cases} 1 & p \geq \frac{1}{2}, \\ 0 & p < \frac{1}{2}. \end{cases}$$

We want to show that g usually agrees with f on inputs drawn from $\mathcal{D}_{z,N}$. Because Q computes f with error at most $\frac{1}{3}$, we have:

$$\begin{aligned} \Pr_{x \sim \mathcal{D}_{z,N}} [f(x) \in \{0,1\} \text{ and } f(x) \neq g(x)] &\leq \Pr_{x \sim \mathcal{D}_{z,N}} \left[|\Pr[Q(x) = 1] - p| \geq \frac{1}{6} \right] \\ &\leq \delta, \end{aligned}$$

by Theorem 48. Additionally, $\mathsf{D}(g) \leq O\left(T^5 \log \frac{T}{\delta}\right)$ because g depends only on p. ☐

The next theorem essentially shows that no $\mathsf{PH}^{\mathsf{PromiseBQP}}$ query algorithm can solve the Forrelation ∘ OR problem (i.e. given an input divided into rows, decide if the ORs of the rows are Forrelated or uniformly random).

Theorem 50. *Let M, N satisfy* $\mathrm{quasipoly}(M) = \mathrm{quasipoly}(N)$ *(i.e. $M \leq \mathrm{quasipoly}(N)$ and $N \leq \mathrm{quasipoly}(M)$). Let $f : \{0,1\}^{MN} \to \{0,1,\perp\}$ be computable by a depth-2 circuit of size* $\mathrm{quasipoly}(N)$ *in which the top gate is a function in $\mathsf{AC}^0[\mathrm{quasipoly}(N), O(1)]$, and all of the bottom gates are functions with bounded-error quantum query complexity at most* $\mathrm{polylog}(N)$.

Let $b \sim \{0,1\}$ be a uniformly random bit. Suppose $z \in \{0,1\}^M$ is sampled such that:

- *If $b = 0$, then z is uniformly random.*
- *If $b = 1$, then z is drawn from the Forrelation distribution $\mathcal{F}_M$.*

Then:

$$\Pr_{b,z,x \sim \mathcal{D}_{z,N}} [f(x) = b] \leq \frac{1}{2} + \frac{\mathrm{polylog}(N)}{\sqrt{M}}.$$

Proof. Suppose the bottom-level gates all have quantum query complexity at most T, and that there are at most s such gates. Let $\delta = \frac{1}{s\sqrt{M}}$. Consider the function $g : \{0,1\}^{MN} \to \{0,1\}$ obtained by replacing all of the bottom-level gates of f with the corresponding decision trees from Corollary 49 that have depth $d \leq O\left(T^5 \log \frac{T}{\delta}\right)$). (The condition of Corollary 49 is satisfied for sufficiently large N, as $T \leq \text{polylog}(N) \ll \sqrt{N}$.)

Note that $g \in \mathsf{AC}^0[\text{quasipoly}(N), O(1)]$: a depth-$d$ decision tree can be computed by a width-d DNF formula, and since $d \leq \text{polylog}(N) \cdot \log\left(\text{quasipoly}(N) \cdot \sqrt{M}\right) \leq \text{polylog}(N)$ and $s \leq \text{quasipoly}(N)$, the total number of gates needed to evaluate all s decision trees is at most $\text{quasipoly}(N)$.

By a union bound over all of the bottom-level gates, observe that

$$\begin{aligned}\Pr_{b,z,x\sim\mathcal{D}_{z,N}}[f(x) = b] &\leq \Pr_{b,z,x\sim\mathcal{D}_{z,N}}[g(x) = b] + \Pr_{b,z,x\sim\mathcal{D}_{z,N}}[f(x) \in \{0,1\} \text{ and } f(x) \neq g(x)] \\ &\leq \Pr_{b,z,x\sim\mathcal{D}_{z,N}}[g(x) = b] + \frac{1}{\sqrt{M}}, \end{aligned} \tag{10}$$

from the assumption of Corollary 49, just because g disagrees with f only if at least one of the decision trees disagrees with its corresponding quantum query algorithm.

Consider a Boolean function $C(z, i_1, \ldots, i_M)$ that takes inputs $z \in \{0,1\}^M$ and $i_1, \ldots, i_M \in [N]$. Let $\tilde{x} \in \{0,1\}^{MN}$ be the string in which for each row $j \in [M]$:

- If $z_j = 0$, then the jth row of $\tilde{x}$ is all zeros.
- If $z_j = 1$, then the jth row of $\tilde{x}$ contains a single 1 in the i_jth position.

Let C compute $g(\tilde{x})$. Clearly, $C \in \mathsf{AC}^0[\text{quasipoly}(N), O(1)]$. Moreover, if $i_1, \ldots, i_M$ are chosen randomly, then C simulates the behavior of g:

$$\Pr_{b,z,x\sim\mathcal{D}_{z,N}}[g(x) = b] = \Pr_{b,z,i_1,\ldots,i_M}[C(z, i_1, \ldots, i_M) = b]. \tag{11}$$

Putting these together, we find that:

$$\begin{aligned}\Pr_{b,z,x\sim\mathcal{D}_{z,N}}[f(x) = b] &\leq \Pr_{b,z,x\sim\mathcal{D}_{z,N}}[g(x) = b] + \frac{1}{\sqrt{M}} \\ &= \Pr_{b,z,i_1,\ldots,i_M}[C(z, i_1, \ldots, i_M) = b] + \frac{1}{\sqrt{M}} \\ &\leq \frac{1}{2} + \frac{\text{polylog}(M)}{\sqrt{M}} \\ &\leq \frac{1}{2} + \frac{\text{polylog}(N)}{\sqrt{M}}, \end{aligned}$$

where the first two lines apply (10) and (11), the third line holds by Theorem 24, and the last line uses the fact that $M \leq \text{quasipoly}(N)$. ☐

By straightforward techniques, Theorem 50 can be extended to a proof of the following oracle result. We omit the proof, as it is conceptually identical to the proof of Corollary 43 that follows from Theorem 42.

Corollary 51. *There exists an oracle relative to which* $\mathsf{BQP}^{\mathsf{NP}} \not\subset \mathsf{PH}^{\mathsf{PromiseBQP}}$.

5 Limitations of the QMA Hierarchy (And Beyond)

In this section, we use random restriction arguments to prove that $\mathsf{PP} \not\subset \mathsf{QMAH}$ relative to a random oracle.

5.1 The Basic Random Restriction Argument

The most basic form of our random restriction argument, though not necessarily its most easily applicable, is given below. The theorem can be understood as stating that if we choose a random subset S of the bits of some input x, then S usually contains a small set K such that the quantum algorithm's acceptance probability cannot change much when any bits of $S \setminus K$ are flipped. In particular, K serves as a sort of "certificate" of the quantum algorithm's behavior when the bits of $S \setminus K$ are unrestricted.

Theorem 52 (Random restriction for BQP)**.** *Consider a quantum algorithm Q that makes T queries to $x \in \{0,1\}^N$. Choose $k \in \mathbb{N}$. If $S \subseteq [N]$ is sampled such that each $i \in [N]$ is in S with probability p, then with probability at least $1 - 2e^{-k/6}$, there exists a set $K \subseteq S$ of size at most k such that for every $y \in \{0,1\}^N$ with $\{i \in [N] : x_i \neq y_i\} \subseteq S \setminus K$, we have:*

$$|\Pr[Q(x) = 1] - \Pr[Q(y) = 1]| \leq 16Tp\sqrt{N/k}$$

Proof. We proceed in cases. Suppose $k > 2pN$. Then, we may simply take the set $K = S$, which satisfies the theorem whenever $|S| \leq k$. By a Chernoff bound (Fact 15) with $\delta = \frac{k}{pN} - 1$, the probability that this condition is violated is upper bounded by:

$$\Pr[|S| \geq (1+\delta)pN] \leq e^{-\frac{\delta^2 pN}{2+\delta}} \leq e^{-\frac{(1+\delta)pN}{6}} = e^{-k/6},$$

where we use the inequality $\frac{\delta^2}{2+\delta} \geq \frac{1+\delta}{6}$ which holds for all $\delta \geq 1$.

In the complementary case, suppose $k \leq 2pN$. Recall the definition of the *query magnitudes* q_i from Lemma 25, which are defined in terms of the behavior of Q on x. Note that $\sum_{i=1}^N q_i = T$. Let $\tau = \frac{2Tp}{k}$. Since all q_is are nonnegative, $|\{i \in [N] : q_i > \tau\}| \leq \frac{T}{\tau}$. Choose $K = \{i \in S : q_i > \tau\}$. By a Chernoff bound (Fact 15),

$$\Pr[|K| \geq k] \leq e^{-k/6},$$

using the fact that $\mathbb{E}[|K|] = p \cdot |\{i \in [N] : q_i > \tau\}| \leq \frac{pT}{\tau} = \frac{k}{2}$. Additionally,

$$\Pr[|S| \geq 2pN] \leq e^{-pN/3} \leq e^{-k/6},$$

by another Chernoff bound. Suppose $|K| \leq k$ and $|S| \leq 2pN$, which happens with probability at least $1 - 2e^{-k/6}$. Then we have:

$$\begin{aligned}
|\Pr[Q(x) = 1] - \Pr[Q(y) = 1]| &\leq 8\sqrt{T} \cdot \sqrt{\sum_{i : x_i \neq y_i} q_i} \\
&\leq 8\sqrt{T} \cdot \sqrt{\sum_{i \in S \setminus K} q_i} \\
&\leq 8\sqrt{T} \cdot \sqrt{|S|\tau} \\
&\leq 16Tp\sqrt{N/k}.
\end{aligned}$$

Above, the first line applies Lemma 25; the second line holds by the assumption that $\{i \in [N] : x_i \neq y_i\} \subseteq S \setminus K$; the third line applies the definition of K to conclude that $q_i \leq \tau$ for all $i \in S \setminus K$; and the last line substitutes $|S| \leq 2pN$ and $\tau = \frac{2Tp}{k}$. ☐

5.2 Measuring Closeness of Functions

In order to better make sense of Theorem 52, we introduce some language that allows us to quantify how "close" a pair of partial functions are.

Definition 53. *Let $f, g : \{0,1\}^N \to \{0,1,\perp\}$ be partial functions. We say that g* disagrees with f on x *if $f(x) \in \{0,1\}$ and $g(x) \neq f(x)$. The* disagreement of g with respect to f, *denoted $\mathrm{disagr}_f(g)$, is the fraction of inputs on which f and g disagree:*

$$\mathrm{disagr}_f(g) := \Pr_{x \sim \{0,1\}^N} [g \text{ disagrees with } f \text{ on } x]\,.$$

If $\mathcal{C}$ is a class of partial functions, the disagreement of $\mathcal{C}$ with respect to f, *denoted $\mathrm{disagr}_f(\mathcal{C})$, is the minimum disagreement of any function in $\mathcal{C}$ with f:*

$$\mathrm{disagr}_f(\mathcal{C}) := \min_{g \in \mathcal{C}} \mathrm{disagr}_f(g).$$

Note that the above definition is not symmetric in f and g. We typically think of f as some "target" function, and g as some algorithm that tries to compute f on the inputs where f is defined. The goal is for g to be consistent with f with good probability; thus we only penalize g if it reports an incorrect answer when f takes a value in $\{0,1\}$.

This next few propositions show that disagreement behaves intuitively in various ways. First, we show that disagreement satisfies a sort of "triangle inequality".

Proposition 54. *Let $f, g, h : \{0,1\}^N \to \{0,1,\perp\}$. Then $\mathrm{disagr}_f(h) \le \mathrm{disagr}_f(g) + \mathrm{disagr}_g(h)$.*

Proof. This follows from Definition 53 and a union bound:

$$\begin{aligned}
\mathrm{disagr}_f(h) &= \Pr_{x \sim \{0,1\}^N} [h \text{ disagrees with } f \text{ on } x] \\
&\le \Pr_{x \sim \{0,1\}^N} [g \text{ disagrees with } f \text{ on } x \text{ OR } h \text{ disagrees with } g \text{ on } x] \\
&\le \mathrm{disagr}_f(g) + \mathrm{disagr}_g(h).
\end{aligned}$$

□

The next two propositions show that disagreement behaves intuitively with respect to random restrictions. First, we show that disagreement is preserved, in expectation, under random restrictions.

Proposition 55. *Let $f, g : \{0,1\}^N \to \{0,1,\perp\}$. Consider a random restriction ρ with $\Pr[*] = p$. Then $\mathbb{E}_\rho \left[\mathrm{disagr}_{f_\rho}(g_\rho)\right] = \mathrm{disagr}_f(g)$.*

Proof. Let $S = \{i \in [N] : \rho(i) = *\}$, and let $y \in \{0,1\}^{[N] \setminus S}$ be the assignment of non-$*$ variables under ρ. Then:

$$\begin{aligned}
\mathbb{E}_{\rho} \left[\mathrm{disagr}_{f_\rho}(g_\rho)\right] &= \mathbb{E}_{\rho} \left[\Pr_{z \in \{0,1\}^S} [g_\rho \text{ disagrees with } f_\rho \text{ on } z]\right] \\
&= \mathbb{E}_{y \in \{0,1\}^{[N] \setminus S}} \left[\Pr_{z \in \{0,1\}^S} [g \text{ disagrees with } f \text{ on } (y, z)]\right] \\
&= \Pr_{x \in \{0,1\}^N} [g \text{ disagrees with } f \text{ on } x] \\
&= \mathrm{disagr}_f(g).
\end{aligned}$$

□

Finally, we show that if we perform a sequence of random restrictions, each of which incurs some cost in disagreement, then the disagreement accumulates additively.

Proposition 56. *Let $f : \{0,1\}^N \to \{0,1,\bot\}$, and let ρ, σ be random restrictions with* $\Pr[*] = p, q$ *respectively. Suppose there exist classes of functions $\mathcal{C}, \mathcal{D}$ such that:*

(a) $\mathbb{E}_\rho \left[\operatorname{disagr}_{f_\rho}(\mathcal{C})\right] \le \varepsilon$.

(b) For all $g \in \mathcal{C}$, $\mathbb{E}_\sigma \left[\operatorname{disagr}_{g_\sigma}(\mathcal{D})\right] \le \delta$.

Then $\mathbb{E}_{\rho\sigma} \left[\operatorname{disagr}_{f_{\rho\sigma}}(\mathcal{D})\right] \le \varepsilon + \delta$.

Proof. Let $g \in \mathcal{C}$ be the function (depending on ρ) that minimizes $\operatorname{disagr}_{f_\rho}(g)$, and let $h \in \mathcal{D}$ be the function (depending on ρ and σ) that minimizes $\operatorname{disagr}_{g_\sigma}(h)$. Then we have:

$$\begin{aligned}
\mathop{\mathbb{E}}_{\rho,\sigma}\left[\operatorname{disagr}_{f_{\rho\sigma}}(h)\right] &\le \mathop{\mathbb{E}}_{\rho,\sigma}\left[\operatorname{disagr}_{f_{\rho\sigma}}(g_\sigma) + \operatorname{disagr}_{g_\sigma}(h)\right] \\
&= \mathop{\mathbb{E}}_{\rho}\left[\operatorname{disagr}_{f_\rho}(g)\right] + \mathop{\mathbb{E}}_{\rho,\sigma}\left[\operatorname{disagr}_{g_\sigma}(h)\right] \\
&\le \varepsilon + \delta,
\end{aligned}$$

where the first line holds by Proposition 54; the second line applies Proposition 55 and linearity of expectation; and the last line holds because of assumptions (a) and (b). ☐

5.3 Random Restriction for QMA Query Algorithms

With the tools introduced in the previous section, we can state a more intuitive and useful form of our random restriction argument for QMA query algorithms. It states that a random restriction of a QMA query algorithm is close in expectation to a small-width DNF formula.

Theorem 57. *Consider a partial function $f : \{0,1\}^N \to \{0,1,\bot\}$ with $\mathsf{QMA}(f) \le T$. Set $p = \frac{\sqrt{k}}{64T\sqrt{N}}$ for some $k \in \mathbb{N}$. Let ρ be a random restriction with* $\Pr[*] = p$. *Let $\mathcal{DNF}_k$ denote the set of width-k DNFs. Then* $\mathbb{E}_\rho \left[\operatorname{disagr}_{f_\rho}(\mathcal{DNF}_k)\right] \le 2e^{-k/6}$.

Proof. It will be convenient to view the choice of ρ as follows: we choose a string $z \in \{0,1\}^N$ uniformly at random, and then we choose a set $S \subseteq [N]$ wherein each $i \in [N]$ is included in S independently with probability p. Then, we take ρ to be:

$$\rho(i) = \begin{cases} * & i \in S \\ z_i & i \notin S. \end{cases}$$

Thus, by definition, it holds that $f_\rho(z|_S) = f(z)$.

Choose $g \in \mathcal{DNF}_k$ as follows, depending on the choice of ρ. For each $x \in \{0,1\}^S$ with $f_\rho(x) \in \{0,1\}$, choose a certificate K_x for f_ρ on x of minimal size. Define g by:

$$g(y) \coloneqq \bigvee_{\substack{x \in f_\rho^{-1}(1) \\ \mathsf{C}^x(f) \le k}} \bigwedge_{i \in K_x} y_i = x_i.$$

This is to say that we take g to be the OR of all of the chosen 1-certificates of f_ρ that have size at most k. Clearly, g is computable by a width-k DNF, so it remains to show that g has small disagreement with respect to f in expectation.

Call the pair (z, S) "good" if either $f(z) \ne 1$ or $\mathsf{C}^{z|_S}(f_\rho) \le k$. Observe that g agrees with f_ρ on the input $z|_S$ whenever (z, S) is good:

- If $f(z) = 0$, then $z|_S$ cannot contain a 1-certificate for f_ρ, and so $g(z|_S) = f_\rho(z|_S) = 0$.
- If $f(z) = \perp$ then f_ρ always agrees with g on $z|_S$.
- Lastly, if $f(z) = 1$ and $\mathsf{C}^{z|_S}(f_\rho) \leq k$, then $z|_S$ certainly contains the certificate $K_{z|_S}$ and hence $g(z|_S) = f_\rho(z|_S) = 1$.

Thus we can see that the expected disagreement of g with respect to f satisfies:

$$\mathop{\mathbb{E}}_{\rho}\left[\mathrm{disagr}_{f_\rho}(g)\right] = \Pr_{z,S}[g \text{ disagrees with } f_\rho \text{ on } z|_S] \leq \Pr_{z,S}[(z, S) \text{ is not good}].$$

It remains to prove that most (z, S) are good. Let $V(|\psi\rangle, z)$ be the QMA verifier corresponding to f on input z, where $|\psi\rangle$ is the witness. Fix z, and let $|\psi_z\rangle$ be the witness that maximizes $\Pr[V(|\psi\rangle_z, z) = 1]$. By Theorem 52, for any $z \in \{0,1\}^N$, with probability at least $1 - 2e^{-k/6}$ over S, there exists a set $K \subseteq S$ of size at most k such that for every $y \in \{0,1\}^N$ with $\{i \in [N] : z_i \neq y_i\} \in S \setminus K$ we have:

$$|\Pr[V(|\psi\rangle_z, z) = 1] - \Pr[V(|\psi\rangle_z, y) = 1]| \leq \frac{1}{4}.$$

In particular, if $f(z) = 1$ then $\Pr[V(|\psi\rangle_z, y) = 1] \geq \frac{2}{3} - \frac{1}{4} > \frac{1}{3}$, and so $f(y) \neq 0$. This is to say that K is a certificate for f_ρ on $z|_S$, and therefore $\mathsf{C}^{z|_S}(f_\rho) \leq k$ and (z, S) is good.

□

5.4 Application to QMAH

In order to generalize Theorem 57 to QMAH algorithms, we require the following form of Håstad's switching lemma for DNF formulas [Hås87]. The statement given below, and arguably its simplest proof, are given in an exposition by Thapen [Tha09]. Technically, this is just a weaker statement of Theorem 37, though we prefer the version given here because it makes the constant factor explicit.

Lemma 58 (Switching Lemma)**.** *Let f be a width-k DNF. If ρ is a random restriction with $\Pr[*] = q < \frac{1}{9}$, then for any $t > 0$, $\Pr[\mathsf{D}(f_\rho) > t] \leq (9qk)^t$.*

Corollary 59. *Let f be a width-k DNF. Denote by $\mathcal{D}_t$ the set of functions that have deterministic query complexity at most t. If ρ is a random restriction with $\Pr[*] = q < \frac{1}{9}$, then for any $t > 0$, we have $\mathbb{E}_\rho[\mathrm{disagr}_{f_\rho}(\mathcal{D}_t)] \leq (9qk)^t$.*

Proof. Take $g = f_\rho$ if $\mathsf{D}(f_\rho) \leq t$, and otherwise let g be the all-zeros function. Then $g \in \mathcal{D}_t$, and by Lemma 58, $\mathbb{E}_\rho[\mathrm{disagr}_{f_\rho}(g)] \leq (9qk)^t$. □

With all of these tools in hand, we prove in our next theorem that under an appropriately chosen random restriction, a circuit composed of QMA query gates simplifies to a function that is close (in expectation) to a function with low deterministic query complexity. The proof amounts to a recursive application of Theorem 57 combined with Corollary 59.

Theorem 60. *Let $f : \{0,1\}^N \to \{0,1,\perp\}$ be computable by a size-s depth-d circuit where each gate is a (possibly partial) function with QMA query complexity at most R. Fix $k \in \mathbb{N}$, and consider a random restriction ρ with*

$$\Pr[*] = \left(1024R^2kN\right)^{2^{-d}-1} \cdot \left(\frac{e^{-1/6}}{18k}\right)^d,$$

Denote by $\mathcal{D}_k$ the set of functions that have deterministic query complexity at most k. Then $\mathbb{E}_\rho\left[\mathrm{disagr}_{f_\rho}(\mathcal{D}_k)\right] \leq 4se^{-k/6}$.

Proof. For convenience, define $\alpha = \frac{1}{64R\sqrt{k}}$. Let $N_0 = N$, and for $i \in [d]$ define p_i and N_i recursively by:

$$\begin{aligned} p_i &= \frac{\alpha}{\sqrt{N_{i-1}}} \\ N_i &= 2p_i N_{i-1}. \end{aligned}$$

This recursive definition implies that:

$$\prod_{i=1}^{d} p_i = \frac{N_d}{2^d N}. \tag{12}$$

Additionally, a simple inductive calculation shows that N_i takes the closed form:

$$N_i = (2\alpha)^{2(1-2^{-i})} N^{2^{-i}}. \tag{13}$$

We view ρ as a sequence of restrictions $\rho_1, \ldots, \rho_d$ in which ρ_i has $\Pr[*] = p_i q$, where $q = \frac{e^{-1/6}}{9k}$ (one can easily verify from (12) and (13) that $\prod_{i=1}^{d} p_i q$ equals the probability given in the statement in the theorem). We view each ρ_i itself as the composition of two random restrictions, one with $\Pr[*] = p_i$ and one with $\Pr[*] = q$.

We proceed in cases. Suppose $k > N_d$. Let $S = \{i \in [N] : \rho(i) = *\}$ denote the set of unrestricted variables. Notice that $\mathrm{disagr}_{f_\rho}(\mathcal{D}_k) = 0$ whenever $|S| \le k$, just because $\mathcal{D}_k$ contains all total functions on at most k bits. Let $\delta = \frac{k}{N \cdot \Pr[*]} - 1$. By (12),

$$k > N_d = \frac{2^d N \cdot \Pr[*]}{q^d} \ge 2^d N \cdot \Pr[*] \ge 2N \cdot \Pr[*],$$

which implies that $\delta \ge 1$. By a Chernoff bound (Fact 15), this implies that:

$$\mathop{\mathbb{E}}_{\rho}\left[\mathrm{disagr}_{f_\rho}(\mathcal{D}_k)\right] \le \Pr\left[|S| \ge (1+\delta)N \cdot \Pr[*]\right] \le e^{-\frac{\delta^2 N \cdot \Pr[*]}{2+\delta}} \le e^{-\frac{(1+\delta)N \cdot \Pr[*]}{6}} = e^{-k/6},$$

where we use the inequality $\frac{\delta^2}{2+\delta} \ge \frac{1+\delta}{6}$ which holds for all $\delta \ge 1$. Thus, the theorem is proved in this $k > N_d$ case.

In the complementary case, suppose $k \le N_d$. Let $\mathcal{C}_i$ be the class of functions such that for all $g \in \mathcal{C}_i$:

(a) g is computable by a circuit with the same structure as f, except that the gates at distance[8] at most i from the input are eliminated and the gates at distance $i+1$ make at most Rk queries (or, if $i = d$, $\mathsf{D}(g) \le k$).

(b) g depends on at most N_i inputs.

By convention, let $\mathcal{C}_0 = \{f\}$. With this definition, the statement of the theorem follows from Proposition 56 and an inductive application of the following claim:

Claim 61. *For all $g \in \mathcal{C}_{i-1}$, $\mathbb{E}_{\rho_i}\left[\mathrm{disagr}_{g_{\rho_i}}(\mathcal{C}_i)\right] \le 4e^{-k/6} \cdot s_i$, where s_i is the number of gates at distance exactly i from the inputs in the circuit that computes f.*

[8] Here, distance is defined as the length of the *longest* path from that gate to any of the inputs.

Proof of claim. Consider applying ρ_i to g. After the random restriction with $\Pr[*] = p_i$, by a Chernoff bound (Fact 15) and because g depends on at most N_{i-1} variables, the resulting function depends on at most $N_i = 2p_i N_{i-1}$ variables, except with probability at most $e^{-N_i/6} \le e^{-N_d/6} \le e^{-k/6}$ over this first restriction.[9] Additionally, by Theorem 57 with $T = Rk$ and $p = p_i$, because g is a function of at most N_{i-1} variables, there exist width-k DNFs that each have expected disagreement at most $2e^{-k/6}$ with respect to the corresponding bottom-layer QMA gates of the circuit that computes g_{ρ_i}.

After the next random restriction with $\Pr[*] = q$, by Corollary 59, these width-k DNFs each have expected disagreement at most $e^{-k/6}$ from functions of deterministic query complexity at most k. Hence, by Proposition 56, viewing ρ_i as the composition of these two random restrictions, each bottom-layer QMA gate in the circuit that computes g_{ρ_i} has expected disagreement at most $3e^{-k/6}$ from a function of deterministic query complexity at most k.

Let h be a function depending on ρ_i, chosen as follows. Take h to be the all zeros function if g_{ρ_i} depends on more than N_i variables; otherwise let h be the function obtained from g by replacing the bottom-level gates of the circuit that computes g with the corresponding functions of deterministic query complexity k. We verify that $h \in \mathcal{C}_i$:

(a) We can absorb the functions of query complexity k into the next layer of QMA gates, increasing the query complexity of each gate by a multiplicative factor of k. Alternatively, if $i = d$, then $\mathsf{D}(h) \le k$ just because g consists of a single gate.

(b) Either g_{ρ_i} depends on at most N_i variables, or else h is trivial; in either case h depends on at most N_i inputs.

We now demonstrate that $\mathbb{E}_{\rho_i}\left[\operatorname{disagr}_{g_{\rho_i}}(h)\right] \le 4e^{-k/6} \cdot s_i$. Notice that h never disagrees with g_{ρ_i} on input x, unless either (1) g_{ρ_i} depends on more than N_i variables, or (2) one of the functions of deterministic query complexity k disagrees with its corresponding QMA gate on x. Hence, by a union bound, $\mathbb{E}_{\rho_i}\left[\operatorname{disagr}_{g_{\rho_i}}(h)\right] \le e^{-k/6} + 3e^{-k/6} \cdot s_i \le 4e^{-k/6} \cdot s_i$. ☐

This completes the theorem in the $k \le N_d$ case. ☐

As a corollary, we obtain the following result, which shows that small circuits composed of functions with low QMA query complexity cannot compute the Parity function.

Corollary 62. *Let $f : \{0,1\}^N \to \{0,1,\perp\}$ be computed by a circuit of size $s = \operatorname{quasipoly}(N)$ and depth $d = O(1)$, where each gate has QMA query complexity at most $R \le \operatorname{polylog}(N)$. Let $\textsc{Parity}_N$ be the parity function on N bits. Then for any $\varepsilon = \frac{1}{\operatorname{quasipoly}(N)}$ and sufficiently large N, $\operatorname{disagr}_{\textsc{Parity}_N}(f) \ge \frac{1}{2} - \varepsilon$.*

Proof. Choose $k = \lceil 6 \ln(5s/\varepsilon) \rceil \le \operatorname{polylog}(N)$. Let ρ be a random restriction where $p = \Pr[*]$ is the probability given in the statement of Theorem 60. A simple calculation shows that $pN \ge \frac{N^{\Omega(1)}}{\operatorname{polylog}(N)}$; hence $k \le \frac{pN}{2}$ for sufficiently large N. Let $g \in \mathcal{D}_k$ be the function (depending on ρ) that minimizes

[9] Actually, a careful inspection of the steps leading up to this proof reveals that this Chernoff bound is unnecessary: we already account for this "bad" event (the number of unrestricted variables being larger than $2p_i N_{i-1}$) in Theorem 52. We only write it this way to make each step of the proof is as self-contained as possible.

$\text{disagr}_{f_\rho}(g)$. With this, we have the following chain of inequalities:

$$\begin{aligned}
\text{disagr}_{\text{PARITY}_N}(f) &= \mathop{\mathbb{E}}_{\rho}\left[\text{disagr}_{\text{PARITY}_{N|\rho}}(f_\rho)\right] \\
&\geq \mathop{\mathbb{E}}_{\rho}\left[\text{disagr}_{\text{PARITY}_{N|\rho}}(g) - \text{disagr}_{f_\rho}(g)\right] \\
&\geq \mathop{\mathbb{E}}_{\rho}\left[\text{disagr}_{\text{PARITY}_{N|\rho}}(g)\right] - 4se^{-k/6} \\
&\geq \frac{1}{2}\Pr_{\rho}\left[|\{i \in [N] : \rho(i) = *\}| > k\right] - 4se^{-k/6} \\
&\geq \frac{1}{2}\left(1 - e^{-k/4}\right) - 4se^{-k/6} \\
&\geq \frac{1}{2} - 5se^{-k/6} \\
&\geq \frac{1}{2} - \varepsilon.
\end{aligned}$$

Above, the first line holds by Proposition 55; the second line holds by Proposition 54; the third line applies linearity of expectation along with the bound from Theorem 60; the fourth line uses the fact that any function of deterministic query complexity k disagrees with the $(k+1)$-bit parity function on exactly half of all inputs; the fifth line uses a Chernoff bound (Fact 15) and $k \leq \frac{pN}{2}$; the sixth line substitutes $\frac{e^{-k/4}}{2} \leq e^{-k/4} \leq e^{-k/6} \leq se^{-k/6}$; and the last line substitutes the definition of k. □

Via standard complexity-theoretic techniques (c.f. Corollary 36), this implies the following:

Corollary 63. $\mathsf{PP} \not\subset \mathsf{QMAH}$ *with probability* 1 *relative to a random oracle.*

Proof sketch. Note that $\mathsf{PP} \subseteq \mathsf{QMAH}$ if and only if $\mathsf{P}^{\#\mathsf{P}} \subseteq \mathsf{QMAH}$, just because QMAH is closed under polynomial-time reductions. Hence, it suffices to show that $\mathsf{P}^{\#\mathsf{P}} \not\subset \mathsf{QMAH}$ relative to a random oracle.

Call the random oracle $\mathcal{O}$, and let $L^{\mathcal{O}}$ be the language consisting of strings 0^n such that, if we treat n as an index into a portion of $\mathcal{O}$ of size 2^n, then the parity of that length-2^n string is 1. Then $L^{\mathcal{O}} \in \mathsf{P}^{\#\mathsf{P}^{\mathcal{O}}}$. On the other hand, Corollary 62 implies that with probability 1, any $\mathsf{QMAH}^{\mathcal{O}}$ machine fails to decide $L^{\mathcal{O}}(0^n)$ on infinitely many n. Hence, $L^{\mathcal{O}} \notin \mathsf{QMAH}^{\mathcal{O}}$. □

More generally, one even obtains separations such as $\mathsf{PP} \not\subset (\mathsf{MIP}^*)^{(\mathsf{PromiseMIP}^*)^{(\mathsf{PromiseMIP}^*)^{\cdots}}}$ relative to a random oracle—despite the fact that $\mathsf{MIP}^* = \mathsf{RE}$ in the unrelativized world [JNV+20]! This follows from the fact that our definition of QMA query complexity (Definition 17) only depends on the number of queries made by the verifier, and not on the length of the witness state. Hence, it upper bounds the relativized power of essentially *any* complexity class that involves interactive proofs with a polynomial-time quantum verifier, including MIP^*.

6 Open Problems

6.1 Oracles where $\mathsf{BQP} = \mathsf{EXP}$

We construct oracles relative to which $\mathsf{BQP} = \mathsf{P}^{\#\mathsf{P}}$ and yet either PH is infinite (Theorem 26), or $\mathsf{P} = \mathsf{NP}$ (Theorem 29). Can these be strengthened to oracles where we also have $\mathsf{BQP} = \mathsf{EXP}$? The main challenge in generalizing our proofs is that $\mathsf{P}^{\#\mathsf{P}}$ machines, unlike EXP machines, have a polynomial upper bound on the length of the queries they can make. This property allowed us to encode the behavior of a $\mathsf{P}^{\#\mathsf{P}}$ machine M into a part of the oracle that M cannot query, but that a BQP machine with a larger polynomial running time *can* query. Alas, such a simple trick will not work when M is an EXP machine. Nevertheless, there exist alternative tools that can collapse EXP to such weaker complexity classes. For instance, Heller [Hel86] gives an oracle relative to which $\mathsf{BPP} = \mathsf{EXP}$. Beigel and Maciel [BM99] even construct an oracle relative to which $\mathsf{P} = \mathsf{NP}$ and $\oplus\mathsf{P} = \mathsf{EXP}$, using AC^0 circuit lower bounds for the Parity problem that are analogous to the lower bounds we use for Forrelation.

6.2 Finer Control over BQP and PH

Recall Conjecture 5, which states that for every k, there exists an oracle relative to which $\Sigma_k^{\mathsf{P}} \subseteq \mathsf{BQP}$ but $\Sigma_{k+1}^{\mathsf{P}} \not\subset \mathsf{BQP}$. We conjecture more strongly that a small modification of the oracle $\mathcal{O}$ constructed in Theorem 26 achieves this. Recall that $\mathcal{O}$ consists of a random oracle A, and an oracle B that recursively hides the answers to all possible $\mathsf{P}^{\#\mathsf{P}^{\mathcal{O}}}$ queries in instances of the Forrelation problem. The idea is simply to modify the definition of B so that it instead encodes the outputs of Σ_k^{P} machines instead of $\mathsf{P}^{\#\mathsf{P}}$ machines.

Our intuition is that, because the Forrelation instances look random to Σ_k^{P} machines, a Σ_k^{P} machine should not be able to recursively reason about B. Thus, a BQP machine that queries $\mathcal{O} = (A, B)$ should be effectively no more powerful than a $\mathsf{BQP}^{\Sigma_k^{\mathsf{P}}}$ machine that queries only A. If this intuition can be made precise, then one could possibly appeal to our proof that $\Sigma_{k+1}^{\mathsf{P}} \not\subset \mathsf{BQP}^{\Sigma_k^{\mathsf{P}}}$ relative to a random oracle. Of course, we could not get this proof strategy to work—otherwise, we would not have needed the machinery surrounding sensitivity concentration of AC^0 circuits in order to get an oracle where $\mathsf{NP}^{\mathsf{BQP}} \not\subset \mathsf{BQP}^{\mathsf{NP}}$!

We now sketch what we consider a viable alternative approach towards showing that our conjectured oracle separation holds. Instead of the "top-down" view taken above, where one tries to argue that a Σ_k^{P} machine gains no benefit from making recursive queries to B, one might instead attempt a "bottom-up" approach, where one uses the structure of the target $\Sigma_{k+1}^{\mathsf{P}}$ problem (the $\textsc{Sipser}_{k+2}$ function) to argue that each bit of B has only minimal correlation with the answer, starting with the parts of B that are constructed first. Very roughly speaking, our idea would be to combine the random projection technique of [HRST17] with some generalization of the AC^0 sensitivity concentration bounds that we prove in Section 4.2.

In slightly more detail, we would first hit A with a random projection, one that with high probability turns the $\textsc{Sipser}_{k+2}$ function into an AND of large fan-in, while turning any $\Sigma_k^{\mathsf{P}^A}$ algorithm into a low-depth decision tree. Then, we would want to argue that if we fix the unrestricted variables of A to all 1s, and choose Forrelation instances in B consistent with this, then each bit of B is unlikely to flip if we instead randomly change a few bits of A to 0s, and resample the Forrelation instances of B corresponding to Σ_k^{P} machines that return different answers. If this could be shown, then as in Theorem 35, an appeal to Lemma 33 (which is a modification of the BBBV Theorem [BBBV97]) ought to be sufficient to argue that a $\mathsf{BQP}^{\mathcal{O}}$ algorithm could not compute the $\textsc{Sipser}_{k+2}$ function.

For the bits of B corresponding to the bottom-level Σ_k^{P} machines that only query A directly, this is easy to show, as a low-depth decision tree is unlikely to query any 0s under a distribution of mostly 1s. However, for the higher levels of B corresponding to Σ_k^{P} machines that can query the earlier bits of B, this becomes more challenging: we have to argue that a Σ_k^{P} algorithm that queries a long list of Forrelation instances is unlikely to return a different answer when we randomly flip a few of the instances between the uniform and Forrelated distributions. This might require a generalization of Lemma 41 in which (1) the string x is not just uniformly random, but is an arbitrary sequence of Forrelated and uniformly random rows, and (2) instead of flipping a single random row of x from uniformly random to Forrelated, we flip an arbitrary subset of the rows between random and Forrelated, subject only to the constraint that the probability of any individual row being chosen is small.

If this problem is too difficult, it remains interesting, in our view, to give an oracle where $\mathsf{NP} \subseteq \mathsf{BQP}$ but $\mathsf{PH} \not\subset \mathsf{BQP}$. This would merely require proving our proposed generalization of Lemma 41 for low-width DNF formulas, as opposed to arbitrary AC^0 circuits of quasipolynomial size.

6.3 Stronger Random Restriction Lemmas

Can one prove a sharper version of our random restriction lemma for QMA query algorithms (Theorem 57)? Unlike the switching lemma for DNF formulas (Lemma 58), our result has a quantitative dependence on the number of inputs N. Thus, whereas a $\operatorname{polylog}(N)$-width DNF simplifies (to a low-depth decision tree, with high probability) under a random restriction with $\Pr[*] = \frac{1}{\operatorname{polylog}(N)}$, we can only show that a $\operatorname{polylog}(N)$-query QMA algorithm simplifies under a random restriction with $\Pr[*] = \frac{1}{\sqrt{N}\operatorname{polylog}(N)}$, which leaves much fewer unrestricted variables. We see no reason why such a dependence on N should be necessary, and we conjecture that a $\operatorname{polylog}(N)$-query QMA algorithm should simplify greatly under a random restriction with $\Pr[*] = \frac{1}{\operatorname{polylog}(N)}$. It would be interesting to see whether one could prove this even without a bound on the QMA witness length, as we do in our proofs.

It is also worth exploring whether our random restriction lemma could be generalized to other classes of functions. Our argument works for functions of low quantum query complexity, so it is natural to ask: is there a comparable random restriction lemma for bounded low-degree polynomials, and thus functions of low *approximate degree*? Kabanets, Kane, and Lu [KKL17] exhibit a random restriction lemma for *polynomial threshold functions*, an even stronger class of functions, though their bounds become very weak when the degree is much larger than $\sqrt{\log N}$. We conjecture that an analogue of Theorem 57 should hold if we replace low QMA query complexity by low approximate degree, perhaps even with better quantitative parameters.[10]

6.4 Collapsing QMAH to P

In Corollary 63, we gave an oracle relative to which $\mathsf{PP} \not\subset \mathsf{QMAH}$ (indeed, we showed that this holds even for a random oracle). Can one generalize this to an oracle relative to which $\mathsf{P} = \mathsf{QMA} = \mathsf{QMAH} \neq \mathsf{PP}$? A priori, it might seem that one could use techniques similar to the ones we used in Theorem 29 to set $\mathsf{P} = \mathsf{NP}$ while still keeping $\mathsf{P} \neq \mathsf{P}^{\#\mathsf{P}}$. That is, the idea would be to start with a random oracle A, then inductively construct an oracle B, recursively encoding into B answers to all QMA machines that query earlier parts of A and B. One would then hope to prove an analogue of Lemma 32, showing that the bits of B can be computed by small low-depth circuits where the gates are functions of low QMA query complexity, and the inputs are in A. Finally, one could appeal to Corollary 62 to argue that such a circuit cannot compute PARITY.

The main issue is that QMA is a semantic complexity class, in contrast to NP, which is a syntactic complexity class. This is to say that every NP machine defines a language, whereas a QMA machine only defines a promise problem. Hence, it is not clear how B should answer on machines that fail to satisfy the QMA promise without "leaking" information that would otherwise be difficult

[10] One could conceivably even show this by simply proving that *every* partial function with low approximate degree also has low QMA query complexity, made easier by the fact that our definition of QMA query complexity allows for unbounded witness length. This is an easier version of the problem of showing whether approximate degree and quantum query complexity are polynomially related for all partial functions, which remains an open problem.

to compute. Even if we, say, assign those bits of B randomly, we can no longer argue that those bits are computable by a QMA query algorithm, which would break our idea for generalizing Lemma 32.

To illustrate the difficulty in constructing such an oracle, we describe an example of an oracle $\mathcal{O} = (A, B)$ that *fails* to put PP outside QMAH. We start by taking a random oracle A. Then, we inductively construct B, where each bit of B encodes the behavior of a $\mathsf{QMA}^{\mathcal{O}}$ verifier $\langle M, x\rangle$, where M can query the previously constructed parts of the oracle, as follows. We let $p := \max_{|\psi\rangle} \Pr[M(x, |\psi\rangle)] = 1$, and then we randomly choose the encoded bit to be 1 with probability p and 0 with probability $1 - p$. This is to say that we set the bit to 1 with probability equaling the acceptance probability of the QMA verifier, maximized over all possible witness states $|\psi\rangle$.

Unfortunately, while one can easily show that $\mathsf{BPP}^{\mathcal{O}} = \mathsf{QMA}^{\mathcal{O}}$, $\mathcal{O}$ also allows an algorithm to "pull the randomness out" of a quantum algorithm, which makes $\mathcal{O}$ much more powerful than it seems! By padding $\langle M, x\rangle$ with extra bits, one can obtain from the oracle arbitrarily many independent bits sampled with bias p. Because $\mathsf{PH}^{\mathcal{O}} \subseteq \mathsf{BPP}^{\mathcal{O}}$, a $\mathsf{BPP}^{\mathcal{O}}$ machine can run Stockmeyer's algorithm [Sto83] on these samples to obtain a multiplicative approximation of any such p. In particular, this implies that the quantum approximate counting problem, defined in Section 1.3, is in $\mathsf{BPP}^{\mathcal{O}}$. But the quantum approximate counting problem is $\mathsf{PP}^{\mathcal{O}}$-hard [Kup15], so we also have $\mathsf{BPP}^{\mathcal{O}} = \mathsf{PP}^{\mathcal{O}}$. Hence, any oracle that makes $\mathsf{P} = \mathsf{QMA} \neq \mathsf{PP}$ would have to choose a more careful encoding of the answers to QMA problems than the one described here.

7 Acknowledgments

We thank Lance Fortnow, Greg Kuperberg, Patrick Rall, and Avishay Tal for helpful conversations. We are especially grateful to Avishay Tal for providing us with a proof of Corollary 39.

References

[AA13] Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. *Theory of Computing*, 9(4):143–252, 2013. doi:10.4086/toc.2013.v009a004. [p. 4]

[AA14] Scott Aaronson and Andris Ambainis. The need for structure in quantum speedups. *Theory of Computing*, 10(6):133–166, 2014. doi:10.4086/toc.2014.v010a006. [pp. 7, 13, 37]

[AA18] Scott Aaronson and Andris Ambainis. Forrelation: A problem that optimally separates quantum from classical computing. *SIAM Journal on Computing*, 47(3):982–1038, 2018. doi:10.1137/15M1050902. [p. 6]

[AAB+19] Frank Arute, Kunal Arya, Ryan Babbush, et al. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019. doi:10.1038/s41586-019-1666-5. [p. 4]

[Aar05] Scott Aaronson. Quantum computing, postselection, and probabilistic polynomial-time. *Proceedings of the Royal Society A*, 461:3473–3482, 2005. doi:10.1098/rspa.2005.1546. [p. 9]

[Aar10] Scott Aaronson. BQP and the polynomial hierarchy. In *Proceedings of the Forty-Second ACM Symposium on Theory of Computing*, STOC '10, pages 141–150, New York, NY, USA, 2010. Association for Computing Machinery. doi:10.1145/1806689.1806711. [pp. 3, 6]

[AC17] Scott Aaronson and Lijie Chen. Complexity-Theoretic Foundations of Quantum Supremacy Experiments. In Ryan O'Donnell, editor, *32nd Computational Complexity Conference (CCC 2017)*, volume 79 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 22:1–22:67, Dagstuhl, Germany, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.CCC.2017.22. [p. 6]

[AD14] Scott Aaronson and Andrew Drucker. A full characterization of quantum advice. *SIAM Journal on Computing*, 43(3):1131–1183, 2014. doi:10.1137/110856939. [p. 16]

[ADH97] Leonard M. Adleman, Jonathan DeMarrais, and Ming-Deh A. Huang. Quantum computability. *SIAM Journal on Computing*, 26(5):1524–1540, 1997. doi:10.1137/S0097539795293639. [p. 3]

[Adl78] Leonard Adleman. Two theorems on random polynomial time. In *19th Annual Symposium on Foundations of Computer Science (sfcs 1978)*, pages 75–83, 1978. doi:10.1109/SFCS.1978.37. [p. 4]

[AKKT20] Scott Aaronson, Robin Kothari, William Kretschmer, and Justin Thaler. Quantum Lower Bounds for Approximate Counting via Laurent Polynomials. In Shubhangi Saraf, editor, *35th Computational Complexity Conference (CCC 2020)*, volume 169 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 7:1–7:47, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CCC.2020.7. [p. 16]

[Amb18] Andris Ambainis. Understanding quantum algorithms via query complexity. In *Proceedings of the 2018 International Congress of Mathematicians*, volume 3, pages 3249–3270, 2018. [p. 16]

[BBBV97] Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26(5):1510–1523, 1997. doi:10.1137/S0097539796300933. [pp. 3, 5, 8, 10, 11, 12, 13, 14, 19, 20, 26, 47]

[BFL91] László Babai, Lance Fortnow, and Carsten Lund. Non-deterministic exponential time has two-prover interactive protocols. *computational complexity*, 1(1):3–40, 1991. doi:10.1007/BF01200056. [p. 5]

[BGS75] Theodore Baker, John Gill, and Robert Solovay. Relativizations of the P=?NP question. *SIAM Journal on Computing*, 4(4):431–442, 1975. doi:10.1137/0204037. [p. 5]

[BHZ87] Ravi B. Boppana, Johan Håstad, and Stathis Zachos. Does co-NP have short interactive proofs? *Inf. Process. Lett.*, 25(2):127–132, May 1987. doi:10.1016/0020-0190(87)90232-8. [pp. 5, 10]

[BJS10] Michael J. Bremner, Richard Jozsa, and Dan J. Shepherd. Classical simulation of commuting quantum computations implies collapse of the polynomial hierarchy. *Proceedings of the Royal Society A*, 467:459–472, 2010. doi:10.1098/rspa.2010.0301. [p. 4]

[BM99] R. Beigel and A. Maciel. Circuit lower bounds collapse relativized complexity classes. In *Proceedings. Fourteenth Annual IEEE Conference on Computational Complexity (Formerly: Structure in Complexity Theory Conference) (Cat.No.99CB36317)*, pages 222–226, 1999. doi:10.1109/CCC.1999.766280. [p. 47]

[BV97] Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997. doi:10.1137/S0097539796300921. [pp. 3, 5, 6, 20, 37]

[For05] Lance Fortnow. Pulling out the quantumness [online]. December 2005. URL: https://blog.computationalcomplexity.org/2005/12/pulling-out-quantumness.html. [pp. 3, 7, 8, 33]

[FR98] Lance Fortnow and John Rogers. Complexity limitations on quantum computation. In *Proceedings. Thirteenth Annual IEEE Conference on Computational Complexity (Formerly: Structure in Complexity Theory Conference) (Cat. No.98CB36247)*, pages 202–209, 1998. doi:10.1109/CCC.1998.694606. [pp. 3, 5]

[FSS84] Merrick Furst, James B. Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1):13–27, 1984. doi:10.1007/BF01744431. [pp. 22, 23, 28]

[Gro96] Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, pages 212–219, New York, NY, USA, 1996. Association for Computing Machinery. doi:10.1145/237814.237866. [pp. 5, 13]

[GSS+18] Sevag Gharibian, Miklos Santha, Jamie Sikora, Aarthi Sundaram, and Justin Yirka. Quantum Generalizations of the Polynomial Hierarchy with Applications to QMA(2). In Igor Potapov, Paul Spirakis, and James Worrell, editors, *43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018)*, volume 117 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 58:1–58:16, Dagstuhl, Germany, 2018. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.MFCS.2018.58. [p. 16]

[GSTW16] Parikshit Gopalan, Rocco Servedio, Avishay Tal, and Avi Wigderson. Degree and sensitivity: tails of two distributions, 2016. Earlier version in CCC 2016. arXiv:1604.07432. [pp. 12, 28]

[Hås87] Johan Håstad. *Computational Limitations of Small-Depth Circuits*. MIT Press, Cambridge, MA, USA, 1987. [pp. 9, 14, 18, 23, 26, 43]

[Hel86] Hans Heller. On relativized exponential and probabilistic complexity classes. *Information and Control*, 71(3):231–243, 1986. doi:10.1016/S0019-9958(86)80012-2. [p. 47]

[HRST17] Johan Håstad, Benjamin Rossman, Rocco A. Servedio, and Li-Yang Tan. An average-case depth hierarchy theorem for Boolean circuits. *J. ACM*, 64(5), August 2017. doi:10.1145/3095799. [pp. 10, 11, 17, 18, 19, 20, 21, 22, 26, 27, 28, 47]

[Hua19] Hao Huang. Induced subgraphs of hypercubes and a proof of the sensitivity conjecture. *Annals of Mathematics*, 190(3):949–955, 2019. [p. 12]

[JNV+20] Zhengfeng Ji, Anand Natarajan, Thomas Vidick, John Wright, and Henry Yuen. MIP*=RE, 2020. arXiv:2001.04383. [pp. 5, 9, 46]

[KKL17] Valentine Kabanets, Daniel M. Kane, and Zhenjian Lu. A polynomial restriction lemma with applications. In *Proceedings of the 49th Annual ACM SIGACT Symposium *on Theory of Computing*, STOC 2017, pages 615–628, New York, NY, USA, 2017. Association for Computing Machinery. doi:10.1145/3055399.3055470. [p. 48]

[KL80] Richard M. Karp and Richard J. Lipton. Some connections between nonuniform and uniform complexity classes. In *Proceedings of the Twelfth Annual ACM Symposium on Theory of Computing*, STOC '80, pages 302–309, New York, NY, USA, 1980. Association for Computing Machinery. doi:10.1145/800141.804678. [p. 10]

[Kre21] William Kretschmer. Quantum Pseudorandomness and Classical Complexity. In Min-Hsiu Hsieh, editor, *16th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2021)*, volume 197 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 2:1–2:20, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.TQC.2021.2. [p. 9]

[Kup15] Greg Kuperberg. How hard is it to approximate the Jones polynomial? *Theory of Computing*, 11(6):183–219, 2015. doi:10.4086/toc.2015.v011a006. [pp. 9, 49]

[Lau83] Clemens Lautemann. BPP and the polynomial hierarchy. *Information Processing Letters*, 17(4):215–217, 1983. doi:https://doi.org/10.1016/0020-0190(83)90044-3. [p. 4]

[LMN93] Nathan Linial, Yishay Mansour, and Noam Nisan. Constant depth circuits, Fourier transform, and learnability. *J. ACM*, 40(3):607–620, July 1993. doi:10.1145/174130.174138. [p. 12]

[Mon12] Ashley Montanaro. Some applications of hypercontractive inequalities in quantum information theory. *Journal of Mathematical Physics*, 53(12):122206, 2012. doi:10.1063/1.4769269. [p. 13]

[OZ16] Ryan O'Donnell and Yu Zhao. Polynomial Bounds for Decoupling, with Applications. In Ran Raz, editor, *31st Conference on Computational Complexity (CCC 2016)*, volume 50 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 24:1–24:18, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.CCC.2016.24. [p. 13]

[Ros17] Benjamin Rossman. An entropy proof of the switching lemma and tight bounds on the decision-tree size of AC0. Manuscript, 2017. URL: https://users.cs.duke.edu/~br148/logsize.pdf. [pp. 12, 28, 29]

[RS04] Ran Raz and Amir Shpilka. On the power of quantum proofs. In *Proceedings. 19th IEEE Annual Conference on Computational Complexity, 2004.*, pages 260–274, 2004. doi:10.1109/CCC.2004.1313849. [p. 16]

[RST15] Benjamin Rossman, Rocco A. Servedio, and Li-Yang Tan. Complexity theory column 89: The polynomial hierarchy, random oracles, and Boolean circuits. *SIGACT News*, 46(4):50–68, December 2015. doi:10.1145/2852040.2852052. [pp. 10, 20, 21, 22, 26, 28]

[RT19] Ran Raz and Avishay Tal. Oracle separation of BQP and PH. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, pages 13–23, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3313276.3316315. [pp. 3, 6, 7, 19]

[Sha92] Adi Shamir. IP = PSPACE. *J. ACM*, 39(4):869–877, October 1992. doi:10.1145/146585.146609. [p. 5]

[Sho99] Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Review*, 41(2):303–332, 1999. doi:10.1137/S0036144598347011. [p. 5]

[Sim97] Daniel R. Simon. On the power of quantum computation. *SIAM Journal on Computing*, 26(5):1474–1483, 1997. doi:10.1137/S0097539796298637. [p. 5]

[Sip83] Michael Sipser. A complexity theoretic approach to randomness. In *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*, STOC '83, pages 330–335, New York, NY, USA, 1983. Association for Computing Machinery. doi:10.1145/800061.808762. [p. 4]

[ST19] Alexander A. Sherstov and Justin Thaler. Vanishing-error approximate degree and QMA complexity, 2019. arXiv:1909.07498. [p. 16]

[Sto83] Larry Stockmeyer. The complexity of approximate counting. In *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*, STOC '83, pages 118–126, New York, NY, USA, 1983. Association for Computing Machinery. doi:10.1145/800061.808740. [pp. 9, 49]

[Tha09] Neil Thapen. Notes on switching lemmas. *Unpublished manuscript*, 2009. URL: https://users.math.cas.cz/~thapen/switching.pdf. [p. 43]

[Tod91] Seinosuke Toda. PP is as hard as the polynomial-time hierarchy. *SIAM Journal on Computing*, 20(5):865–877, 1991. doi:10.1137/0220053. [p. 4]

[Vin18] Lieuwe Vinkhuijzen. A quantum polynomial hierarchy and a simple proof of Vyalyi's theorem. Master's thesis, Leiden University, 2018. URL: https://theses.liacs.nl/pdf/2017-2018-VinkhuijzenLieuwe.pdf. [p. 16]

[Wat00] John Watrous. Succinct quantum proofs for properties of finite groups. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 537–546. IEEE, 2000. doi:10.1109/SFCS.2000.892005. [p. 6]

[Yap83] Chee-Keng Yap. Some consequences of non-uniform conditions on uniform classes. *Theor. Comput. Sci.*, 26:287–300, 1983. doi:10.1016/0304-3975(83)90020-8. [p. 23]

[ZWD+20] Han-Sen Zhong, Hui Wang, Yu-Hao Deng, Ming-Cheng Chen, Li-Chao Peng, Yi-Han Luo, Jian Qin, Dian Wu, Xing Ding, Yi Hu, Peng Hu, Xiao-Yan Yang, Wei-Jun Zhang, Hao Li, Yuxuan Li, Xiao Jiang, Lin Gan, Guangwen Yang, Lixing You, Zhen Wang, Li Li, Nai-Le Liu, Chao-Yang Lu, and Jian-Wei Pan. Quantum computational advantage using photons. *Science*, 370(6523):1460–1463, 2020. doi:10.1126/science.abe8770. [p. 4]

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il