Electronic Colloquium on Computational Complexity, Report No. 104 (2009)

BQP and the Polynomial Hierarchy

Scott Aaronson[*]

Abstract

The relationship between BQP and PH has been an open problem since the earliest days of quantum computing. We present evidence that quantum computers can solve problems outside the entire polynomial hierarchy, by relating this question to topics in circuit complexity, pseudorandomness, and Fourier analysis.

First, we show that there exists an oracle relation problem (i.e., a problem with many valid outputs) that is solvable in BQP, but not in PH. This also yields a non-oracle relation problem that is solvable in quantum *logarithmic* time, but not in AC^0.

Second, we show that an oracle *decision* problem separating BQP from PH would follow from the *Generalized Linial-Nisan Conjecture*, which we formulate here and which is likely of independent interest. The original Linial-Nisan Conjecture (about pseudorandomness against constant-depth circuits) was recently proved by Braverman, after being open for twenty years.

Contents

[*] MIT. Email: aaronson@csail.mit.edu. Supported by an NSF CAREER Award, a DARPA YFA grant, MIT CSAIL, and the Keck Foundation.

ISSN 1433-8092

1 Introduction

A central task of quantum computing theory is to understand how BQP—meaning Bounded-Error Quantum Polynomial-Time, the class of all problems feasible for a quantum computer—fits in with classical complexity classes. In their original 1993 paper defining BQP, Bernstein and Vazirani [11] showed that $\mathsf{BPP} \subseteq \mathsf{BQP} \subseteq \mathsf{P}^{\#\mathsf{P}}$.[1] Informally, this says that quantum computers are at least as fast as classical probabilistic computers and no more than exponentially faster (indeed, they can be simulated using an oracle for counting). Bernstein and Vazirani also gave evidence that $\mathsf{BPP} \neq \mathsf{BQP}$, by exhibiting an oracle problem called Recursive Fourier Sampling that requires $n^{\Omega(\log n)}$ queries on a classical computer but only n queries on a quantum computer.[2] The evidence for the power of quantum computers became dramatically stronger a year later, when Shor [33] (building on work of Simon [34]) showed that Factoring and Discrete Logarithm are in BQP. On the other hand, Bennett et al. [9] gave oracle evidence that $\mathsf{NP} \not\subset \mathsf{BQP}$, and while no one regards such evidence as decisive, today it seems extremely unlikely that quantum computers can solve NP-complete problems in polynomial time. A vast body of research, continuing to the present, has sought to map out the detailed boundary between those NP problems that are feasible for quantum computers and those that are not.

However, there is a complementary question that—despite being universally recognized as one of the "grand challenges" of the field—has had essentially zero progress over the last sixteen years:

> *Is* BQP *in* NP*? More generally, is* BQP *contained anywhere in the polynomial hierarchy* $\mathsf{PH} = \mathsf{NP} \cup \mathsf{NP}^{\mathsf{NP}} \cup \mathsf{NP}^{\mathsf{NP}^{\mathsf{NP}}} \cup \cdots$*?*

The "default" conjecture is presumably $\mathsf{BQP} \not\subset \mathsf{PH}$, since no one knows what a simulation of BQP in PH would look like. Before this work, however, there was no formal evidence for or against that conjecture. Almost all the problems for which we have quantum algorithms—including Factoring and Discrete Logarithm—are easily seen to be in $\mathsf{NP} \cap \mathsf{coNP}$.[3] One notable exception is Recursive Fourier Sampling, the problem that Bernstein and Vazirani [11] originally used to construct an oracle A relative to which $\mathsf{BPP}^A \neq \mathsf{BQP}^A$. One can show, without too much difficulty, that Recursive Fourier Sampling yields oracles A relative to which $\mathsf{BQP}^A \not\subset \mathsf{NP}^A$ and indeed $\mathsf{BQP}^A \not\subset \mathsf{MA}^A$. However, while it is reasonable to conjecture that Recursive Fourier Sampling (as an oracle problem) is not in PH, it is open even to show that this problem (or any other BQP oracle problem) is not in AM! Recall that $\mathsf{AM} = \mathsf{NP}$ under

[1] The upper bound was later improved to $\mathsf{BQP} \subseteq \mathsf{PP}$ by Adleman, DeMarrais, and Huang [3].

[2] For more about Recursive Fourier Sampling see Aaronson [2].

[3] Here we exclude BQP-complete problems such as approximating the Jones polynomial [5], which, by the very fact of being BQP-complete, seem hard to interpret as "evidence" for $\mathsf{BQP} \not\subset \mathsf{PH}$.

plausible derandomization assumptions [26]. Thus, until we solve the problem of constructing an oracle A such that $\mathsf{BQP}^A \not\subset \mathsf{AM}^A$, we cannot even claim to have oracle evidence (which is itself, of course, a weak form of evidence) that $\mathsf{BQP} \not\subset \mathsf{NP}$.

Before going further, we should clarify that there are two questions here: whether $\mathsf{BQP} \subseteq \mathsf{PH}$ and whether $\mathsf{PromiseBQP} \subseteq \mathsf{PromisePH}$. In the unrelativized world, it is entirely possible that quantum computers can solve promise problems outside the polynomial hierarchy, but that all *languages* in BQP are nevertheless in PH. However, for the specific purpose of constructing an oracle A such that $\mathsf{BQP}^A \not\subset \mathsf{PH}^A$, the two questions are equivalent, basically because one can always "offload" a promise into the construction of the oracle A.[4]

1.1 Motivation

There are at least four reasons why the BQP versus PH question is so interesting. At a basic level, it is both theoretically and practically important to understand what classical resources are needed to simulate quantum physics. For example, when a quantum system evolves to a given state, is there always a short classical proof that it does so? Can one estimate quantum amplitudes using approximate counting (which would imply $\mathsf{BQP} \subseteq \mathsf{BPP}^{\mathsf{NP}}$)? If something like this were true, then while the exponential speedup of Shor's factoring algorithm might stand, quantum computing would nevertheless seem much less different from classical computing than previously thought.

Second, if $\mathsf{BQP} \not\subset \mathsf{PH}$, then many possibilities for new quantum algorithms might open up to us. One often hears the complaint that there are too few quantum algorithms, or that progress on quantum algorithms has slowed since the mid-1990s. In our opinion, the real issue here has nothing to do with quantum computing, and is simply that there are too few natural NP-intermediate problems for which there plausibly *could be* quantum algorithms! In other words, instead of focussing on Graph Isomorphism and a small number of other NP-intermediate problems, it might be fruitful to look for quantum algorithms solving completely different types of problems—problems that are not necessarily even in PH. In this paper, we will see a new example of such a quantum algorithm, which solves a problem called Fourier Checking.

Third, it is natural to ask whether the $\mathsf{P} \overset{?}{=} \mathsf{BQP}$ question is related to that *other* fundamental question of complexity theory, $\mathsf{P} \overset{?}{=} \mathsf{NP}$. More concretely, is it possible that quantum computers could provide exponential speedups even if $\mathsf{P} = \mathsf{NP}$? If $\mathsf{BQP} \subseteq \mathsf{PH}$, then certainly the answer to that question is no (since $\mathsf{P} = \mathsf{NP} \implies \mathsf{P} = \mathsf{PH}$). Therefore, if we want evidence that quantum computing could survive a collapse of P and NP, we must also seek evidence that $\mathsf{BQP} \not\subset \mathsf{PH}$.

Fourth, a major challenge for quantum computing research is to *get better evidence that quantum computers cannot solve* NP*-complete problems in polynomial time.* As an example, could we show that if $\mathsf{NP} \subseteq \mathsf{BQP}$, then the polynomial hierarchy collapses? At first glance, this seems like a wild hope; certainly we have no idea at present how to prove anything of the kind. However, notice

[4] Here is a simple proof: let $\Pi = (\Pi_{\text{YES}}, \Pi_{\text{NO}})$ be a promise problem in $\mathsf{PromiseBQP}^A \setminus \mathsf{PromisePH}^A$, for some oracle A. Then clearly, every $\mathsf{PromisePH}^A$ machine M fails to solve Π on infinitely many inputs x in $\Pi_{\text{YES}} \cup \Pi_{\text{NO}}$. This means that we can produce an infinite sequence of inputs $x_1, x_2, \ldots$ in $\Pi_{\text{YES}} \cup \Pi_{\text{NO}}$, whose lengths $n_1, n_2, \ldots$ are spaced arbitrarily far apart, such that every $\mathsf{PromisePH}^A$ machine M fails to solve Π on at least one x_i. Now let B be an oracle that is identical to A, except that for each input length n, it reveals (i) whether $n = n_i$ for some i and (ii) if so, what the corresponding x_i is. Also, let L be the unary language that contains 0^n if and only if (i) $n = n_i$ for some i and (ii) $x_i \in \Pi_{\text{YES}}$. Then L is in BQP^B but not PH^B.

that if $\mathsf{BQP} \subseteq \mathsf{AM}$, then the desired implication would follow immediately! For in that case,

$$\begin{aligned} \mathsf{NP} \subseteq \mathsf{BQP} &\Longrightarrow \mathsf{coNP} \subseteq \mathsf{BQP} \\ &\Longrightarrow \mathsf{coNP} \subseteq \mathsf{AM} \\ &\Longrightarrow \mathsf{PH} = \Sigma_2^{\mathsf{P}} \end{aligned}$$

where the last implication was shown by Boppana, Håstad, and Zachos [12]. Similar remarks apply to the questions of whether $\mathsf{NP} \subseteq \mathsf{BQP}$ would imply $\mathsf{PH} \subseteq \mathsf{BQP}$, and whether the folklore result $\mathsf{NP}^{\mathsf{BPP}} \subseteq \mathsf{BPP}^{\mathsf{NP}}$ has the quantum analogue $\mathsf{NP}^{\mathsf{BQP}} \subseteq \mathsf{BQP}^{\mathsf{NP}}$. In each of these cases, we find that understanding some other issue in quantum complexity theory requires first coming to grips with whether BQP is contained in some level of the polynomial hierarchy.

1.2 Our Results

This paper presents the first formal evidence for the possibility that $\mathsf{BQP} \not\subset \mathsf{PH}$. Perhaps more importantly, it places the relativized BQP versus PH question at the frontier of (classical) circuit lower bounds. The heart of the problem, we will find, is to extend Braverman's spectacular recent proof [13] of the Linial-Nisan Conjecture, in ways that would reveal a great deal of information about small-depth circuits independent of the implications for quantum computing.

We have two main contributions. First, we achieve an oracle separation between BQP and PH for the case of *relation problems*. A relation problem is simply a problem where the desired output is an n-bit string (rather than a single bit), and any string from some nonempty set S is acceptable. Relation problems arise often in theoretical computer science; one well-known example is finding a Nash equilibrium (shown to be PPAD-complete by Daskalakis et al. [15]). Within quantum computing, there is considerable precedent for studying relation problems as a warmup to the harder case of decision problems. For example, in 2004 Bar-Yossef, Jayram, and Kerenidis [6] gave a relation problem with quantum one-way communication complexity $O(\log n)$ and randomized one-way communication complexity $\Omega(\sqrt{n})$. It took several more years for Gavinsky et al. [20] to achieve the same separation for decision problems, and the proof was much more complicated. The same phenomenon has arisen many times in quantum communication complexity [17, 18, 19, 21, 22], though to our knowledge, this is the first time it has arisen in quantum query complexity.

Formally, our result is as follows:

Theorem 1 *There exists an oracle A relative to which $\mathsf{FBQP}^A \not\subset \mathsf{FBPP}^{\mathsf{PH}^A}$, where FBQP and FBPP are the relation versions of BQP and BPP respectively.*[5]

Underlying Theorem 1 is a new lower bound against AC^0 circuits (constant-depth circuits composed of AND, OR, and NOT gates). The close connection between AC^0 and the polynomial hierarchy that we exploit is not new. In the early 1980s, Furst-Saxe-Sipser [16] and Yao [39] noticed that, if we have a PH machine M that computes (say) the Parity of a 2^n-bit oracle string, then by simply reinterpreting the existential quantifiers of M as OR gates and the universal quantifiers as AND gates, we obtain an AC^0 circuit of size $2^{\mathrm{poly}(n)}$ solving the same problem. It follows that, if we can prove a $2^{\omega(\mathrm{polylog}\, n)}$ lower bound on the size of AC^0 circuits computing Parity, we

[5] Confusingly, the F stands for "function"; we are simply following the standard naming convention for classes of relation problems (FP, FNP, etc).

can construct an oracle A relative to which $\oplus\mathsf{P}^A \not\subset \mathsf{PH}^A$. The idea is the same for constructing an A relative to which $\mathcal{C}^A \not\subset \mathsf{PH}^A$, where $\mathcal{C}$ is any complexity class.

Indeed, the relation between PH and AC^0 is so direct that we get the following as a more-or-less immediate counterpart to Theorem 1:

Theorem 2 *In the unrelativized world (with no oracle), there exists a relation problem solvable in quantum logarithmic time but not in nonuniform* AC^0.

The relation problem that we use to separate BQP from PH, and $\mathsf{BQLOGTIME}$ from AC^0, is called Fourier Fishing. The problem can be informally stated as follows. We are given oracle access to n Boolean functions $f_1, \ldots, f_n : \{0,1\}^n \rightarrow \{-1,1\}$, which we think of as chosen uniformly at random. The task is to output n strings, $z_1, \ldots, z_n \in \{0,1\}^n$, such that the corresponding squared Fourier coefficients $\widehat{f_1}(z_1)^2, \ldots, \widehat{f_n}(z_n)^2$ are "often much larger than average." Notice that if f_i is a random Boolean function, then each of its Fourier coefficients $\widehat{f_i}(z)$ follows a normal distribution—meaning that with overwhelming probability, a constant fraction of the Fourier coefficients will be a constant factor larger than the mean. Furthermore, it is straightforward to create a quantum algorithm that samples each z with probability proportional to $\widehat{f_i}(z)^2$, so that larger Fourier coefficients are more likely to be sampled than smaller ones.

On the other hand, computing any *specific* $\widehat{f_i}(z)$ is easily seen to be equivalent to summing 2^n bits. By well-known lower bounds on the size of AC^0 circuits computing the Majority function (see Håstad [36] for example), it follows that, for any fixed z, computing $\widehat{f_i}(z)$ cannot be in PH as an oracle problem. Unfortunately, this does not directly imply any separation between BQP and PH, since the quantum algorithm does not compute $\widehat{f_i}(z)$ either: it just samples a z with probability proportional to $\widehat{f_i}(z)^2$. However, we will show that, if there exists a $\mathsf{BPP}^{\mathsf{PH}}$ machine M that even *approximately* simulates the behavior of the quantum algorithm, then one can solve Majority by means of a *nondeterministic* reduction—which uses approximate counting to estimate $\Pr[M \text{ outputs } z]$, and adds a constant number of layers to the AC^0 circuit. The central difficulty is that, if M knew the specific z for which we were interested in estimating $\widehat{f_i}(z)$, then it could choose adversarially never to output that z. To solve this, we will show that we can "smuggle" a Majority instance into the estimation of a *random* Fourier coefficient $\widehat{f_i}(z)$, in such a way that it is information-theoretically impossible for M to determine which z we care about.

Our second contribution is to define and study a new black-box decision problem, called Fourier Checking. Informally, in this problem we are given oracle access to *two* Boolean functions $f, g : \{0,1\}^n \rightarrow \{-1,1\}$, and are promised that either

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

(ii) f is uniformly random, while g is extremely well correlated with f's Fourier transform over $\mathbb{Z}_2^n$ (which we call "forrelated").

The problem is to decide whether (i) or (ii) is the case.

It is not hard to show that Fourier Checking is in BQP: basically, one can prepare a uniform superposition over all $x \in \{0,1\}^n$, then query f, apply a quantum Fourier transform, query g, and check whether one has recovered something close to the uniform superposition. On the other hand, being forrelated seems like an extremely "global" property of f and g: one that would not be apparent from querying *any* small number of $f(x)$ and $g(y)$ values, regardless of the outcomes of those queries. And thus, one might conjecture that Fourier Checking (as an oracle problem) is not in PH.

In this paper, we adduce strong evidence for that conjecture. Specifically, we show that for every $k \leq 2^{n/4}$, the forrelated distribution over $\langle f, g\rangle$ pairs is $O\left(k^2/2^{n/2}\right)$*-almost k-wise independent.* By this we mean that, if one had $1/2$ prior probability that f and g were uniformly random, and $1/2$ prior probability that f and g were forrelated, then even conditioned on any k values of f and g, the posterior probability that f and g were forrelated would still be

$$\frac{1}{2} \pm O\left(\frac{k^2}{2^{n/2}}\right).$$

We conjecture that this almost k-wise independence property is enough, by itself, to imply that an oracle problem is not in PH. We call this the *Generalized Linial-Nisan Conjecture.*

Without the $\pm O\left(k^2/2^{n/2}\right)$ error term, our conjecture would be equivalent[6] to a famous conjecture in circuit complexity made by Linial and Nisan [28] in 1990. Their conjecture stated that *polylogarithmic independence fools* AC^0: in other words, every probability distribution over N-bit strings that is *uniform* on every small subset of bits, is indistinguishable from the truly uniform distribution by AC^0 circuits. When we began investigating this topic a year ago, even the original Linial-Nisan Conjecture was still open. Since then, Braverman [13] (building on earlier work by Bazzi [7] and Razborov [30]) has given a beautiful proof of that conjecture. In other words, to construct an oracle relative to which $\mathsf{BQP} \not\subset \mathsf{PH}$, it now suffices to generalize Braverman's Theorem from k-wise independent distributions to almost k-wise independent ones. We believe that this is by far the most promising approach to the BQP versus PH problem.

Alas, generalizing Braverman's proof is much harder than one might have hoped. To prove the original Linial-Nisan Conjecture, Braverman showed that every AC^0 function $f : \{0,1\}^n \rightarrow \{0,1\}$ can be well-approximated, in the L_1-norm, by *low-degree sandwiching polynomials*: real polynomials $p_\ell, p_u : \mathbb{R}^n \rightarrow \mathbb{R}$, of degree $O(\operatorname{polylog} n)$, such that $p_\ell(x) \leq f(x) \leq p_u(x)$ for all $x \in \{0,1\}^n$. Since p_ℓ and p_u trivially have the same expectation on any k-wise independent distribution that they have on the uniform distribution, one can show that f must have almost the same expectation as well. To generalize Braverman's result from k-wise independence to almost k-wise independence, we will show that it suffices to construct low-degree sandwich polynomials that satisfy a certain additional condition. This new condition (which we call "low-fat") basically says that p_ℓ and p_u must be representable as linear combinations of *terms* (that is, products of x_i's and $(1 - x_i)$'s), in such a way that the sum of the absolute values of the coefficients is bounded—thereby preventing "massive cancellations" between positive and negative terms. Unfortunately, while we know two techniques for approximating AC^0 functions by low-degree polynomials—that of Linial-Mansour-Nisan [27] and that of Razborov [29] and Smolensky [35]—neither technique provides anything like the control over coefficients that we need. To construct low-fat sandwiching polynomials, it seems necessary to reprove the LMN and Razborov-Smolensky theorems in a more "conservative," less "profligate" way. And such an advance seems likely to lead to breakthroughs in circuit complexity and computational learning theory having nothing to do with quantum computing.

Let us mention two further applications of Fourier Checking:

(1) If the Generalized Linial-Nisan Conjecture holds, then just like with Fourier Fishing, we can "scale down by an exponential," to obtain a promise problem that is in $\mathsf{BQLOGTIME}$ but not in AC^0.

[6] Up to unimportant variations in the parameters

(2) Without any assumptions, we can prove the new results that there exist oracles relative to which $\mathsf{BQP} \not\subset \mathsf{BPP}_{\mathsf{path}}$ and $\mathsf{BQP} \not\subset \mathsf{SZK}$. We can also reprove all previous oracle separations between BQP and classical complexity classes in a unified fashion.

To summarize our conclusions:

Theorem 3 *Assuming the Generalized Linial-Nisan Conjecture, there exists an oracle A relative to which $\mathsf{BQP}^A \not\subset \mathsf{PH}^A$, and there also exists a promise problem in $\mathsf{BQLOGTIME} \setminus \mathsf{AC}^0$. Unconditionally, there exists an oracle A relative to which $\mathsf{BQP}^A \not\subset \mathsf{BPP}^A_{\mathsf{path}}$ and $\mathsf{BQP}^A \not\subset \mathsf{SZK}^A$.*

As a candidate problem, Fourier Checking has at least five advantages over the Recursive Fourier Sampling problem of Bernstein and Vazirani [11]. First, it is much simpler to define and reason about. Second, Fourier Checking has the almost k-wise independence property, which is not shared by Recursive Fourier Sampling, and which immediately connects the former to general questions about pseudorandomness against constant-depth circuits. Third, Fourier Checking can yield exponential separations between quantum and classical models, rather than just quasipolynomial ones. Fourth, one can hope to use Fourier Checking to give an oracle relative to which BQP is not in $\mathsf{PH}[n^c]$ (or PH with n^c alternations) for any fixed c; by contrast, Recursive Fourier Sampling is in $\mathsf{PH}[\log n]$. Finally, it is at least conceivable that the quantum algorithm for Fourier Checking is *good* for something. We leave the challenge of finding an explicit computational problem that "instantiates" Fourier Checking, in the same way that Factoring and Discrete Logarithm instantiated Shor's period-finding problem.

1.3 In Defense of Oracles

This paper is concerned with finding oracles relative to which BQP outperforms classical complexity classes. As such, it is open to the usual objections: "But don't oracle results mislead us about the 'real' world? What about non-relativizing results like $\mathsf{IP} = \mathsf{PSPACE}$ [32]?"

In our view, it is most helpful to think of oracle separations, not as strange metamathematical claims, but as *lower bounds in a concrete computational model that is natural and well-motivated in its own right.* The model in question is *query complexity*, where the resource to be minimized is the number of accesses to a very long input string. When someone gives an oracle A relative to which $\mathcal{C}^A \not\subset \mathcal{D}^A$, what they really mean is simply that they have found a problem that $\mathcal{C}$ machines can solve using superpolynomially fewer queries than $\mathcal{D}$ machines. In other words, $\mathcal{C}$ has has "cleared the first possible obstacle"—the query complexity obstacle—to having capabilities beyond those of $\mathcal{D}$. Of course, it could be (and sometimes is) that $\mathcal{C} \subseteq \mathcal{D}$ for other reasons, but if we do not *even* have a query complexity lower bound, then proving one is in some sense the obvious place to start.

Oracle separations have played a role in many of the central developments of both classical and quantum complexity theory. As mentioned earlier, proving query complexity lower bounds for PH machines is essentially equivalent to proving size lower bounds for AC^0 circuits—and indeed, the pioneering AC^0 lower bounds of the early 1980s were explicitly motivated by the goal of proving oracle separations for PH.[7] Within quantum computing, oracle results have played an even more decisive role: the first evidence for the power of quantum computers came from the oracle

[7] Yao's paper [39] was entitled "Separating the polynomial-time hierarchy by oracles"; the Furst-Saxe-Sipser paper [16] was entitled "Parity, circuits, and the polynomial time hierarchy."

separations of Bernstein-Vazirani [11] and Simon [34], and Shor's algorithm [33] contains an oracle algorithm (for the PERIOD-FINDING problem) at its core.

Having said all that, if for some reason one still feels averse to the language of oracles, then (as mentioned before) one is free to scale everything down by an exponential, and to reinterpret a relativized separation between BQP and PH as an *un*relativized separation between $\mathsf{BQLOGTIME}$ and AC^0.

2 Preliminaries

It will be convenient to consider Boolean functions of the form $f : \{0,1\}^n \rightarrow \{-1,1\}$. Throughout this paper, we let $N = 2^n$; we will often view the truth table of a Boolean function as an "input" of size N. Given a Boolean function $f : \{0,1\}^n \rightarrow \{-1,1\}$, the Fourier transform of f is defined as

$$\widehat{f}(z) := \frac{1}{\sqrt{N}} \sum_{x \in \{0,1\}^n} (-1)^{x \cdot z} f(x).$$

Recall Parseval's identity:

$$\sum_{x \in \{0,1\}^n} f(x)^2 = \sum_{z \in \{0,1\}^n} \widehat{f}(z)^2 = N.$$

2.1 Problems

We first define the FOURIER FISHING problem, in both "distributional" and "promise" versions. In the distributional version, we are given oracle access to n Boolean functions $f_1, \ldots, f_n : \{0,1\}^n \rightarrow \{-1,1\}$, which are chosen uniformly and independently at random. The task is to output n strings, $z_1, \ldots, z_n \in \{0,1\}^n$, at least 75% of which satisfy $\left|\widehat{f_i}(z_i)\right| \geq 1$ and at least 25% of which satisfy $\left|\widehat{f_i}(z_i)\right| \geq 2$. (Note that these thresholds are not arbitrary, but were carefully chosen to produce a separation between the quantum and classical models!)

We now want a version of FOURIER FISHING that removes the need to assume the f_i's are uniformly random, replacing it with a worst-case promise on the f_i's. Call an n-tuple $\langle f_1, \ldots, f_n \rangle$ of Boolean functions *good* if

$$\sum_{i=1}^{n} \sum_{z_i : \left|\widehat{f_i}(z_i)\right| \geq 1} \widehat{f_i}(z_i)^2 \geq 0.8Nn,$$

$$\sum_{i=1}^{n} \sum_{z_i : \left|\widehat{f_i}(z_i)\right| \geq 2} \widehat{f_i}(z_i)^2 \geq 0.26Nn.$$

(We will show in Lemma 8 that the vast majority of $\langle f_1, \ldots, f_n \rangle$ are good.) In PROMISE FOURIER FISHING, we are given oracle access to Boolean functions $f_1, \ldots, f_n : \{0,1\}^n \rightarrow \{-1,1\}$, which are promised to be good. The task, again, is to output strings $z_1, \ldots, z_n \in \{0,1\}^n$, at least 75% of which satisfy $\left|\widehat{f_i}(z_i)\right| \geq 1$ and at least 25% of which satisfy $\left|\widehat{f_i}(z_i)\right| \geq 2$.

Next we define a decision problem called FOURIER CHECKING. Here we are given oracle access to two Boolean functions $f, g : \{0,1\}^n \rightarrow \{-1,1\}$. We are promised that either

(i) $\langle f,g\rangle$ was drawn from the uniform distribution $\mathcal{U}$, which sets every $f(x)$ and $g(y)$ by a fair, independent coin toss.

(ii) $\langle f,g\rangle$ was drawn from the "forrelated" distribution $\mathcal{F}$, which is defined as follows. First choose a random real vector $v=(v_x)_{x\in\{0,1\}^n}\in\mathbb{R}^N$, by drawing each entry independently from a Gaussian distribution with mean 0 and variance 1. Then set $f(x):=\operatorname{sgn}(v_x)$ and $g(x):=\operatorname{sgn}(\widehat{v}_x)$ for all x. Here

$$\operatorname{sgn}(\alpha):=\begin{cases}1 & \text{if } \alpha\geq 0\\ -1 & \text{if } \alpha<0\end{cases}$$

and $\widehat{v}$ is the Fourier transform of v over $\mathbb{Z}_2^n$:

$$\widehat{v}_y:=\frac{1}{\sqrt{N}}\sum_{x\in\{0,1\}^n}(-1)^{x\cdot y}v_x.$$

In other words, f and g *individually* are still uniformly random, but they are no longer independent: now g is now extremely well correlated with the Fourier transform of f (hence "forrelated").

The problem is to accept if $\langle f,g\rangle$ was drawn from $\mathcal{F}$, and to reject if $\langle f,g\rangle$ was drawn from $\mathcal{U}$. Note that, since $\mathcal{F}$ and $\mathcal{U}$ overlap slightly, we can only hope to succeed with overwhelming probability over the choice of $\langle f,g\rangle$, not for every $\langle f,g\rangle$ pair.

We can also define a promise-problem version of Fourier Checking. In Promise Fourier Checking, we are promised that the quantity

$$p(f,g):=\frac{1}{N^3}\left(\sum_{x,y\in\{0,1\}^n}f(x)(-1)^{x\cdot y}g(y)\right)^2$$

is either at least 0.05 or at most 0.01. The problem is to accept in the former case and reject in the latter case.

2.2 Complexity Classes

See the Complexity Zoo[8] for the definitions of standard complexity classes, such as BQP, AM, and PH. When we write $\mathcal{C}^{\mathsf{PH}}$ (i.e., a complexity class $\mathcal{C}$ with an oracle for the polynomial hierarchy), we mean $\cup_{k\geq 1}\mathcal{C}^{\Sigma_k^{\mathsf{P}}}$.

We will consider not only decision problems, but also *relation problems* (also called *function problems*). In a relation problem, the output is not a single bit but a $\operatorname{poly}(n)$-bit string y. There could be many valid y's for a given instance, and the algorithm's task is to output any one of them.

The definitions of FP and FNP (the relation versions of P and NP) are standard. We now define FBPP and FBQP, the relation versions of BPP and BQP.

Definition 4 FBPP *is the class of relations $R\subseteq\{0,1\}^*\times\{0,1\}^*$ for which there exists a probabilistic polynomial-time algorithm A that, given any input $x\in\{0,1\}^n$, produces an output y such that*

$$\Pr[(x,y)\in R]=1-o(1),$$

[8] www.complexityzoo.com

where the probability is over A's internal randomness. (In particular, this implies that for every x, there exists at least one y such that $(x,y) \in R$.) FBQP *is defined the same way, except that A is a quantum algorithm rather than a classical one.*

An important point about FBPP and FBQP is that, as far as we know, these classes do not admit amplification. In other words, the value of an algorithm's success probability might actually matter, not just the fact that the probability is bounded above $1/2$. This is why we adopt the convention that an algorithm "succeeds" if it outputs $(x,y) \in R$ with probability $1 - o(1)$. In practice, we will give oracle problems for which the FBQP algorithm succeeds with probability $1 - 1/\exp(n)$, while any $\mathsf{FBPP}^{\mathsf{PH}}$ algorithm succeeds with probability at most (say) 0.99. How far the constant in this separation can be improved is an open problem.

Another important point is that, while $\mathsf{BPP}^{\mathsf{PH}} = \mathsf{P}^{\mathsf{PH}}$ (which follows from $\mathsf{BPP} \subseteq \Sigma_2^{\mathsf{P}}$), the class $\mathsf{FBPP}^{\mathsf{PH}}$ is strictly larger than $\mathsf{FP}^{\mathsf{PH}}$. To see this, consider the relation

$$R = \{(0^n, y) : K(y) \geq n\},$$

where we are given n, and asked to output any string of Kolmogorov complexity at least n. Clearly this problem is in FBPP: just output a random $2n$-bit string. On the other hand, just as obviously the problem is not in $\mathsf{FP}^{\mathsf{PH}}$. This is why we need to construct an oracle A such that $\mathsf{FBQP}^A \not\subset \mathsf{FBPP}^{\mathsf{PH}^A}$: because constructing an oracle A such that $\mathsf{FBQP}^A \not\subset \mathsf{FP}^{\mathsf{PH}^A}$ is trivial and not even related to quantum computing.

We now discuss some "low-level" complexity classes. AC^0 is the class of problems solvable by a nonuniform family of AND/OR/NOT circuits, with depth $O(1)$, size $\operatorname{poly}(n)$, and unbounded fanin. When we say "AC^0 circuit," we mean a constant-depth circuit of AND/OR/NOT gates, not necessarily of polynomial size. Any such circuit can be made into a *formula* (i.e., a circuit of fanout 1) with only a polynomial increase in size. The circuit has *depth d* if it consists of d alternating layers of AND and OR gates (without loss of generality, the NOT gates can all be pushed to the bottom, and we do not count them towards the depth). For example, a DNF (Disjunctive Normal Form) formula is just an AC^0 circuit of depth 2.

We will also be interested in quantum *logarithmic* time, which can be defined naturally as follows:

Definition 5 $\mathsf{BQLOGTIME}$ *is the class of languages $L \subseteq \{0,1\}^*$ that are decidable, with bounded probability of error, by a* $\mathsf{LOGTIME}$*-uniform family of quantum circuits $\{C_n\}_n$ such that each C_n has $O(\log n)$ gates, and can include gates that make random-access queries to the input string $x = x_1 \ldots x_n$ (i.e., that map $|i\rangle |z\rangle$ to $|i\rangle |z \oplus x_i\rangle$ for every $i \in [n]$).*

One other complexity class that arises in this paper, which is less well known than it should be, is $\mathsf{BPP_{path}}$. Loosely speaking, $\mathsf{BPP_{path}}$ can be defined as the class of problems that are solvable in probabilistic polynomial time, given the ability to "postselect" (that is, discard all runs of the computation that do not produce a desired result, even if such runs are the overwhelming majority). Formally:

Definition 6 $\mathsf{BPP_{path}}$ *is the class of languages $L \subseteq \{0,1\}^*$ for which there exists a* BPP *machine M, which can either "succeed" or "fail" and conditioned on succeeding either "accept" or "reject," such that for all inputs x:*

(i) $\Pr[M(x) \text{ succeeds}] > 0$.

(ii) $x \in L \Longrightarrow \Pr[M(x) \text{ accepts} \mid M(x) \text{ succeeds}] \geq \frac{2}{3}$.

(iii) $x \notin L \Longrightarrow \Pr[M(x) \text{ accepts} \mid M(x) \text{ succeeds}] \leq \frac{1}{3}$.

$\mathsf{BPP_{path}}$ was defined by Han, Hemaspaandra, and Thierauf [25], who also showed that $\mathsf{MA} \subseteq \mathsf{BPP_{path}}$ and $\mathsf{P}_{||}^{\mathsf{NP}} \subseteq \mathsf{BPP_{path}} \subseteq \mathsf{BPP}_{||}^{\mathsf{NP}}$. Using Fourier Checking, we will construct an oracle A relative to which $\mathsf{BQP}^A \not\subset \mathsf{BPP}_{\mathsf{path}}^A$. This result might not sound amazing, but (i) it is new, (ii) it does not follow from the "standard" quantum algorithms, such as those of Simon [34] and Shor [33], and (iii) it supersedes almost all previous oracle results placing BQP outside classical complexity classes.[9] As another illustration of the versatility of Fourier Checking, we use it to give an A such that $\mathsf{BQP}^A \not\subset \mathsf{SZK}^A$, where SZK is Statistical Zero Knowledge. The opposite direction—an A such that $\mathsf{SZK}^A \not\subset \mathsf{BQP}^A$—was shown by Aaronson [1] in 2002.

3 Quantum Algorithms

In this section, we show that Fourier Fishing and Fourier Checking both admit simple quantum algorithms.

3.1 Quantum Algorithm for Fourier Fishing

Here is a quantum algorithm, `FF-ALG`, that solves Fourier Fishing with overwhelming probability in $O(n^2)$ time and n quantum queries (one to each f_i). For $i := 1$ to n, first prepare the state

$$\frac{1}{\sqrt{N}} \sum_{x \in \{0,1\}^n} f_i(x) |x\rangle ,$$

then apply Hadamard gates to all n qubits, then measure in the computational basis and output the result as z_i.

Intuitively, `FF-ALG` samples the Fourier coefficients of each f_i under a distribution that is skewed towards larger coefficients; the algorithm's behavior is illustrated pictorially in Figure 1. We now give a formal analysis. Recall the definition of a "good" tuple $\langle f_1, \ldots, f_n \rangle$ from Section 2.1. Assuming $\langle f_1, \ldots, f_n \rangle$ is good, it is easy to analyze `FF-ALG`'s success probability.

Lemma 7 *Assuming $\langle f_1, \ldots, f_n \rangle$ is good, `FF-ALG` succeeds with probability $1 - 1/\exp(n)$.*

Proof. Let $\langle z_1, \ldots, z_n \rangle$ be the algorithm's output. For each i, let X_i be the event that $\left|\widehat{f}_i(z_i)\right| \geq 1$ and let Y_i be the event that $\left|\widehat{f}_i(z_i)\right| \geq 2$. Also let $p_i := \Pr[X_i]$ and $q_i := \Pr[Y_i]$, where the

[9] The one exception is the result of Green and Pruim [24] that there exists an A relative to which $\mathsf{BQP}^A \not\subset \mathsf{P}^{\mathsf{NP}^A}$, but that can also be easily reproduced using Fourier Checking.

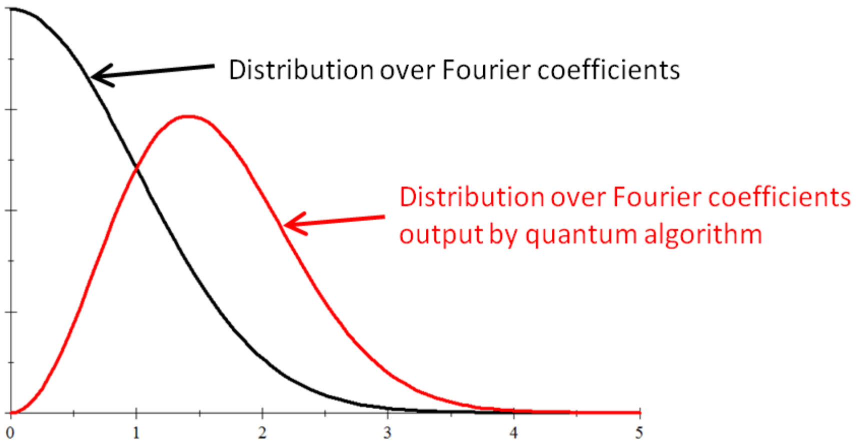

Figure 1: The Fourier coefficients of a random Boolean function follow a Gaussian distribution, with mean 0 and variance 1. However, larger Fourier coefficients are more likely to be observed by the quantum algorithm.

probability is over `FF-ALG`'s internal (quantum) randomness. Then clearly

$$p_i = \frac{1}{N} \sum_{z_i : \left|\widehat{f}_i(z_i)\right| \geq 1} \widehat{f}_i(z_i)^2,$$

$$q_i = \frac{1}{N} \sum_{z_i : \left|\widehat{f}_i(z_i)\right| \geq 2} \widehat{f}_i(z_i)^2.$$

So by assumption,

$$p_1 + \cdots + p_n \geq 0.8n,$$
$$q_1 + \cdots + q_n \geq 0.26n.$$

By a Chernoff/Hoeffding bound, it follows that

$$\Pr[X_1 + \cdots + X_n \geq 0.75n] > 1 - \frac{1}{\exp(n)},$$
$$\Pr[Y_1 + \cdots + Y_n \geq 0.25n] > 1 - \frac{1}{\exp(n)}.$$

Hence `FF-ALG` succeeds with $1 - 1/\exp(n)$ probability by the union bound. ■

We also have the following:

Lemma 8 *$\langle f_1, \ldots, f_n \rangle$ is good with probability $1 - 1/\exp(n)$, if the f_i's are chosen uniformly at random.*

Proof. Choose $f : \{0,1\}^n \rightarrow \{-1,1\}$ uniformly at random. Then for each z, the Fourier coefficient $\widehat{f}(z)$ follows a normal distribution, with mean 0 and variance 1. So in the limit of large N,

$$\begin{aligned}
\operatorname*{E}_f\left[\sum_{z:\left|\widehat{f}(z)\right|\geq 1}\widehat{f}(z)^2\right] &= \sum_{z\in\{0,1\}^n}\Pr\left[\left|\widehat{f}(z)\right|\geq 1\right]\operatorname{E}\left[\widehat{f}(z)^2 \mid \left|\widehat{f}(z)\right|\geq 1\right] \\
&\approx \frac{2N}{\sqrt{2\pi}}\int_1^\infty e^{-x^2/2}x^2dx \\
&\approx 0.801N.
\end{aligned}$$

Likewise,

$$\begin{aligned}
\operatorname*{E}_f\left[\sum_{z:\left|\widehat{f}(z)\right|\geq 2}\widehat{f}(z)^2\right] &\approx \frac{2N}{\sqrt{2\pi}}\int_2^\infty e^{-x^2/2}x^2dx \\
&\approx 0.261N.
\end{aligned}$$

Since the f_i's are chosen independently of one another, it follows by a Chernoff bound that

$$\begin{aligned}
\sum_{i=1}^n\sum_{z_i:\left|\widehat{f}_i(z_i)\right|\geq 1}\widehat{f}_i(z_i)^2 &\geq 0.8Nn, \\
\sum_{i=1}^n\sum_{z_i:\left|\widehat{f}_i(z_i)\right|\geq 2}\widehat{f}_i(z_i)^2 &\geq 0.26Nn
\end{aligned}$$

with probability $1-1/\exp(n)$ over the choice of $\langle f_1,\ldots,f_n\rangle$. ■

Combining Lemmas 7 and 8, we find that `FF-ALG` succeeds with probability $1-1/\exp(n)$, where the probability is over both $\langle f_1,\ldots,f_n\rangle$ and `FF-ALG`'s internal randomness.

3.2 Quantum Algorithm for Fourier Checking

We now turn to Fourier Checking, the problem of deciding whether two Boolean functions f,g are independent or forrelated. Here is a quantum algorithm, `FC-ALG`, that solves Fourier Checking with constant error probability using $O(1)$ queries. First prepare a uniform superposition over all $x\in\{0,1\}^n$. Then query f in superposition, to create the state

$$\frac{1}{\sqrt{N}}\sum_{x\in\{0,1\}^n}f(x)\left|x\right\rangle$$

Then apply Hadamard gates to all n qubits, to create the state

$$\frac{1}{N}\sum_{x,y\in\{0,1\}^n}f(x)(-1)^{x\cdot y}\left|y\right\rangle.$$

Then query g in superposition, to create the state

$$\frac{1}{N}\sum_{x,y\in\{0,1\}^n}f(x)(-1)^{x\cdot y}g(y)\left|y\right\rangle.$$

Then apply Hadamard gates to all n qubits again, to create the state

$$\frac{1}{N^{3/2}} \sum_{x,y,z \in \{0,1\}^n} f(x)(-1)^{x \cdot y} g(y)(-1)^{y \cdot z} |z\rangle .$$

Finally, measure in the computational basis, and "accept" if and only if the outcome $|0\rangle^{\otimes n}$ is observed. If needed, repeat the whole algorithm $O(1)$ times to boost the success probability.

It is clear that the probability of observing $|0\rangle^{\otimes n}$ (in a single run of `FC-ALG`) equals

$$p(f,g) := \frac{1}{N^3} \left(\sum_{x,y \in \{0,1\}^n} f(x)(-1)^{x \cdot y} g(y) \right)^2 .$$

Recall that Promise Fourier Checking was the problem of deciding whether $p(f,g) \geq 0.05$ or $p(f,g) \leq 0.01$, promised that one of these is the case. Thus, we immediately get a quantum algorithm to solve Promise Fourier Checking, with constant error probability, using $O(1)$ queries to f and g.

For the distributional version of Fourier Checking, we also need the following theorem.

Theorem 9 *If $\langle f,g \rangle$ is drawn from the uniform distribution $\mathcal{U}$, then*

$$\operatorname*{E}_{\mathcal{U}}[p(f,g)] = \frac{1}{N}.$$

If $\langle f,g \rangle$ is drawn from the forrelated distribution $\mathcal{F}$, then

$$\operatorname*{E}_{\mathcal{F}}[p(f,g)] > 0.07.$$

Proof. The first part follows immediately by symmetry (i.e., the fact that all $N = 2^n$ measurement outcomes of the quantum algorithm are equally likely).

For the second part, let $v \in \mathbb{R}^N$ be the vector of independent Gaussians used to generate f and g, let $w = v / \|v\|_2$ be v scaled to have unit norm, and let H be the n-qubit Hadamard matrix. Also let $\operatorname{flat}(w)$ be the unit vector whose x^{th} entry is $\operatorname{sgn}(w_x)/\sqrt{N} = f(x)/\sqrt{N}$, and let $\operatorname{flat}(Hw)$ be the unit vector whose x^{th} entry is $\operatorname{sgn}(\widehat{v}_x)/\sqrt{N} = g(x)/\sqrt{N}$. Then $p(f,g)$ equals

$$\left(\operatorname{flat}(w)^T H \operatorname{flat}(Hw) \right)^2 ,$$

or the squared inner product between the vectors $\operatorname{flat}(w)$ and $H \operatorname{flat}(Hw)$. Note that $w^T \cdot HHw = w^T w = 1$. So the whole problem is to understand the "discretization error" incurred in replacing w^T by $\operatorname{flat}(w)^T$ and HHw by $H \operatorname{flat}(Hw)$. By the triangle inequality, the angle between $\operatorname{flat}(w)$ and $H \operatorname{flat}(Hw)$ is at most the angle between $\operatorname{flat}(w)$ and w, plus the angle between w and $H \operatorname{flat}(Hw)$. In other words:

$$\arccos\left(\operatorname{flat}(w)^T H \operatorname{flat}(Hw) \right) \leq \arccos\left(\operatorname{flat}(w)^T w \right) + \arccos\left(w^T H \operatorname{flat}(Hw) \right) .$$

Now,

$$\begin{aligned}
\operatorname{flat}(w)^T w &= \sum_{x\in\{0,1\}^n} w_x \cdot \frac{1}{\sqrt{N}} \frac{|w_x|}{w_x} \\
&= \frac{1}{\sqrt{N}} \sum_{x\in\{0,1\}^n} |w_x| \\
&= \frac{\sum_{x\in\{0,1\}^n} |v_x|}{\sqrt{N}\,\|v\|_2}.
\end{aligned}$$

Recall that each v_x is an independent real Gaussian with mean 0 and variance 1, meaning that each $|v_x|$ is an independent nonnegative random variable with expectation $\sqrt{2/\pi}$. So by standard tail bounds, for all constants $\varepsilon > 0$ we have

$$\begin{aligned}
\Pr_v \left[\sum_{x\in\{0,1\}^n} |v_x| \leq \left(\sqrt{\frac{2}{\pi}} - \varepsilon \right) N \right] &\leq \frac{1}{\exp(N)}, \\
\Pr \left[\|v\|_2^2 \geq (1+\varepsilon) N \right] &\leq \frac{1}{\exp(N)}.
\end{aligned}$$

So by the union bound,

$$\Pr_v \left[\operatorname{flat}(w)^T w \leq \sqrt{\frac{2}{\pi}} - \varepsilon \right] \leq \frac{1}{\exp(N)}.$$

Since H is unitary, the same analysis applies to $w^T H \operatorname{flat}(Hw)$. Therefore, for all constants $\varepsilon > 0$, with $1 - 1/\exp(N)$ probability we have

$$\begin{aligned}
\arccos\left(\operatorname{flat}(w)^T w \right) &\leq \left(\arccos \sqrt{\frac{2}{\pi}} \right) + \varepsilon, \\
\arccos\left(w^T H \operatorname{flat}(Hw) \right) &\leq \left(\arccos \sqrt{\frac{2}{\pi}} \right) + \varepsilon.
\end{aligned}$$

So setting $\varepsilon = 0.0001$,

$$\begin{aligned}
\arccos\left(\operatorname{flat}(w)^T H \operatorname{flat}(Hw) \right) &\leq \arccos\left(\operatorname{flat}(w)^T w \right) + \arccos\left(w^T H \operatorname{flat}(Hw) \right) \\
&\leq 2 \left(\arccos \sqrt{\frac{2}{\pi}} \right) + 2\varepsilon \\
&\leq 1.3
\end{aligned}$$

Therefore, with $1 - 1/\exp(N)$ probability over $\langle f, g \rangle$ drawn from $\mathcal{F}$,

$$\operatorname{flat}(w)^T H \operatorname{flat}(Hw) \geq \cos 1.3,$$

in which case $p(f,g) \geq (\cos 1.3)^2 \approx 0.072$. ■

Combining Theorem 9 with Markov's inequality, we immediately get the following:

Corollary 10

$$\Pr_{\langle f,g\rangle\sim\mathcal{U}}\left[p\left(f,g\right)\geq 0.01\right]\leq\frac{100}{N},$$
$$\Pr_{\langle f,g\rangle\sim\mathcal{D}}\left[p\left(f,g\right)\geq 0.05\right]\geq\frac{1}{50}.$$

4 The Classical Complexity of Fourier Fishing

In Section 3.1, we gave a quantum algorithm for Fourier Fishing that made only one query to each f_i. By contrast, it is not hard to show that any classical algorithm for Fourier Fishing requires exponentially many queries to the f_i's. In this section, we prove a much stronger result: that Fourier Fishing is not in $\mathsf{FBPP}^{\mathsf{PH}}$. This result does not rely on any unproved conjectures.

4.1 Constant-Depth Circuit Lower Bounds

Our starting point will be the following AC^0 lower bound, which can be found in the book of Håstad [36] for example.

Theorem 11 ([36]) *Any depth-d circuit that accepts all n-bit strings of Hamming weight $n/2+1$, and rejects all strings of Hamming weight $n/2$, has size* $\exp\left(\Omega\left(n^{1/(d-1)}\right)\right)$.

We now give a corollary of Theorem 11, which (though simple) seems to be new, and might be of independent interest. Consider the following problem, which we call ε-Bias Detection. We are given a string $y=y_1\ldots y_m\in\{0,1\}^m$, and are promised that each bit y_i is 1 with independent probability p. The task is to decide whether $p=1/2$ or $p=1/2+\varepsilon$.

Corollary 12 *Let $\mathcal{U}[\varepsilon]$ be the distribution over $\{0,1\}^m$ where each bit is 1 with independent probability $1/2+\varepsilon$. Then any depth-d circuit C such that*

$$\left|\Pr_{\mathcal{U}[\varepsilon]}[C]-\Pr_{\mathcal{U}[0]}[C]\right|=\Omega\left(1\right)$$

has size $\exp\left(\Omega\left(1/\varepsilon^{1/(d+2)}\right)\right)$.

Proof. Suppose such a distinguishing circuit C exists, with depth d and size S, for some $\varepsilon>0$ (the parameter m is actually irrelevant). Let $n=1/\varepsilon$, and assume for simplicity that n is an integer. Using C, we will construct a new circuit C' with depth $d'=d+3$ and size $S'=O\left(nS\right)+\text{poly}\left(n\right)$, which accepts all strings $x\in\{0,1\}^n$ of Hamming weight $n/2+1$, and rejects all strings of Hamming weight $n/2$. By Theorem 11, this will imply that the original circuit C must have had size

$$\begin{aligned}S&=\frac{1}{n}\exp\left(\Omega\left(n^{1/(d'-1)}\right)\right)-\text{poly}\left(n\right)\\&=\exp\left(\Omega\left(1/\varepsilon^{1/(d+2)}\right)\right).\end{aligned}$$

So fix an input $x\in\{0,1\}^n$, and suppose we choose m bits $x_{i_1},\ldots,x_{i_m}$ from x, with each index i_j chosen uniformly at random with replacement. Call the resulting m-bit string y. Observe that if

x had Hamming weight $n/2$, then y will be distributed according to $\mathcal{U}[0]$, while if x had Hamming weight $n/2+1$, then y will be distributed according to $\mathcal{U}[\varepsilon]$. So by assumption,

$$\Pr\left[C(y) \mid |x| = n/2\right] = \alpha,$$
$$\Pr\left[C(y) \mid |x| = n/2+1\right] = \alpha + \delta$$

for some constants α and $\delta \neq 0$ (we can assume $\delta > 0$ without loss of generality).

Now suppose we repeat the above experiment $T = kn$ times, for some constant $k = k(\alpha, \delta)$. That is, we create T strings $y_1, \ldots, y_T$ by choosing random bits of x, so that each y_i is distributed independently according to either $\mathcal{U}[0]$ or $\mathcal{U}[\varepsilon]$. We then apply C to each y_i. Let

$$Z = C(y_1) + \cdots + C(y_T)$$

be the number of C invocations that accept. Then by a Chernoff bound, if $|x| = n/2$ then

$$\Pr\left[Z > \alpha T + \frac{\delta}{3} T\right] < \exp(-n),$$

while if $|x| = n/2 + 1$ then

$$\Pr\left[Z < \alpha T + \frac{2\delta}{3} T\right] < \exp(-n).$$

By taking k large enough, we can make both of these probabilities less than 2^{-n}. By the union bound, this implies that there must exist a way to choose $y_1, \ldots, y_T$ so that

$$|x| = \frac{n}{2} \Longrightarrow Z \leq \alpha T + \frac{\delta}{3} T,$$
$$|x| = \frac{n}{2} + 1 \Longrightarrow Z \geq \alpha T + \frac{2\delta}{3} T$$

for *every* x with $|x| \in \{n/2, n/2+1\}$ simultaneously. In forming the circuit C', we simply hardwire that choice.

The last step is to decide whether $Z \leq \alpha T + \frac{\delta}{3} T$ or $Z \geq \alpha T + \frac{2\delta}{3} T$. This can be done using an AC^0 circuit for the Approximate Majority problem (see Viola [37] for example), which has depth 3 and size $\operatorname{poly}(T)$. The end result is a circuit C' to distinguish $|x| = n/2$ from $|x| = n/2+1$, which has depth $d+3$ and size $TS + \operatorname{poly}(T) = O(nS) + \operatorname{poly}(n)$. ■

4.2 Secretly Biased Fourier Coefficients

In this section, we prove two lemmas indicating that one can *slightly* bias one of the Fourier coefficients of a random Boolean function $f : \{0,1\}^n \to \{-1,1\}$, and yet still have f be information-theoretically indistinguishable from a random Boolean function (so that, in particular, an adversary has no way of knowing which Fourier coefficient was biased). These lemmas will play a key role in our reduction from ε-Bias Detection to Fourier Fishing.

Fix a string $s \in \{0,1\}^n$. Let $\mathcal{A}[s]$ be the probability distribution over functions $f : \{0,1\}^n \to \{-1,1\}$ where each $f(x)$ is 1 with independent probability $\frac{1}{2} + (-1)^{s \cdot x} \frac{1}{2\sqrt{N}}$, and let $\mathcal{B}[s]$ be the distribution where each $f(x)$ is 1 with independent probability $\frac{1}{2} - (-1)^{s \cdot x} \frac{1}{2\sqrt{N}}$. Then let $\mathcal{D}[s] = \frac{1}{2}(\mathcal{A}[s] + \mathcal{B}[s])$ (that is, an equal mixture of $\mathcal{A}[s]$ and $\mathcal{B}[s]$).

Lemma 13 *Suppose Alice chooses $s \in \{0,1\}^n$ uniformly at random, then draws f according to $\mathcal{D}[s]$. She keeps s secret, but sends the truth table of f to Bob. After examining f, Bob outputs a string z such that $\left|\widehat{f}(z)\right| \geq \beta$. Then*

$$\Pr[s = z] \geq \frac{e^{\beta} + e^{-\beta}}{2\sqrt{e}N}.$$

where the probability is over all runs of the protocol.

Proof. By Yao's principle, we can assume without loss of generality that Bob's strategy is deterministic. For each z, let $\mathcal{F}[z]$ be the set of all f's that cause Bob to output z. Then the first step is to lower-bound $\Pr_{\mathcal{D}[z]}[f]$, for some fixed z and $f \in \mathcal{F}[z]$. Let $N_f[z]$ be the number of inputs $x \in \{0,1\}^n$ such that $f(x) = (-1)^{z \cdot x}$. It is not hard to see that $N_f[z] = \frac{N}{2} + \frac{\sqrt{N}\widehat{f}(z)}{2}$. So

$$\begin{aligned}
\Pr_{\mathcal{D}[z]}[f] &= \frac{1}{2}\left(\Pr_{\mathcal{A}[z]}[f] + \Pr_{\mathcal{B}[z]}[f]\right) \\
&= \frac{1}{2}\left(\prod_{x \in \{0,1\}^n}\left(\frac{1}{2} + \frac{(-1)^{z \cdot x} f(x)}{2\sqrt{N}}\right) + \prod_{x \in \{0,1\}^n}\left(\frac{1}{2} - \frac{(-1)^{z \cdot x} f(x)}{2\sqrt{N}}\right)\right) \\
&= \frac{1}{2^{N+1}}\left(\left(1 + \frac{1}{\sqrt{N}}\right)^{N_f[z]}\left(1 - \frac{1}{\sqrt{N}}\right)^{N - N_f[z]} + \left(1 - \frac{1}{\sqrt{N}}\right)^{N_f[z]}\left(1 + \frac{1}{\sqrt{N}}\right)^{N - N_f[z]}\right) \\
&= \frac{1}{2^{N+1}}\left(\frac{\left(1 + 1/\sqrt{N}\right)^{\left(\sqrt{N}\widehat{f}(z) + N\right)/2}}{\left(1 - 1/\sqrt{N}\right)^{\left(\sqrt{N}\widehat{f}(z) - N\right)/2}} + \frac{\left(1 - 1/\sqrt{N}\right)^{\left(\sqrt{N}\widehat{f}(z) + N\right)/2}}{\left(1 + 1/\sqrt{N}\right)^{\left(\sqrt{N}\widehat{f}(z) - N\right)/2}}\right) \\
&= \frac{1}{2^{N+1}}\left(1 - \frac{1}{N}\right)^{N/2}\left(\left(\frac{1 + 1/\sqrt{N}}{1 - 1/\sqrt{N}}\right)^{\sqrt{N}\widehat{f}(z)/2} + \left(\frac{1 - 1/\sqrt{N}}{1 + 1/\sqrt{N}}\right)^{\sqrt{N}\widehat{f}(z)/2}\right) \\
&= \frac{1}{2\sqrt{e}2^N}\left(e^{\widehat{f}(z)} + e^{-\widehat{f}(z)}\right) \\
&\geq \frac{e^{\beta} + e^{-\beta}}{2\sqrt{e}2^N}.
\end{aligned}$$

Here the second-to-last line takes the limit as $N \to \infty$, while the last line follows from the assumption $\left|\widehat{f}(z)\right| \geq \beta$, together with the fact that $e^y + e^{-y}$ increases monotonically away from $y = 0$.

Summing over all z and f,

$$\begin{aligned}\Pr[s=z] &= \sum_{z\in\{0,1\}^n}\sum_{f\in\mathcal{F}[z]}\Pr[f]\cdot\Pr[s=z \mid f] \\ &= \sum_{z\in\{0,1\}^n}\sum_{f\in\mathcal{F}[z]}\Pr[f]\cdot\frac{\Pr[f \mid s=z]\Pr[s=z]}{\Pr[f]} \\ &= \frac{1}{N}\sum_{z\in\{0,1\}^n}\sum_{f\in\mathcal{F}[z]}\Pr_{\mathcal{D}[z]}[f] \\ &\geq \frac{e^{\beta}+e^{-\beta}}{2\sqrt{e}N}.\end{aligned}$$

■

Now let $\mathcal{D}=\mathrm{E}_s[\mathcal{D}[s]]$ (that is, an equal mixture of all the $\mathcal{D}[s]$'s). We claim that $\mathcal{D}$ is extremely close in variation distance to $\mathcal{U}$, the uniform distribution over all Boolean functions $f:\{0,1\}^n\rightarrow\{-1,1\}$.

Lemma 14 $\|\mathcal{D}-\mathcal{U}\|\leq\frac{e-1}{2\sqrt{2eN}}$.

Proof. By a calculation from Lemma 13, for all f and s we have

$$\Pr_{\mathcal{D}[s]}[f]=\frac{1}{2\sqrt{e}2^N}\left(e^{\widehat{f}(s)}+e^{-\widehat{f}(s)}\right)$$

in the limit of large N. Hence

$$\Pr_{\mathcal{D}}[f]=\mathop{\mathrm{E}}_{s}\left[\Pr_{\mathcal{D}[s]}[f]\right]=\frac{1}{2\sqrt{e}N2^N}\sum_{s\in\{0,1\}^n}\left(e^{\widehat{f}(s)}+e^{-\widehat{f}(s)}\right).$$

Clearly $\mathrm{E}_f[\Pr_{\mathcal{D}}[f]]=1/2^N$. Our goal is to upper-bound the variance $\mathrm{Var}_f[\Pr_{\mathcal{D}}[f]]$, which measures the distance from $\mathcal{D}$ to the uniform distribution. In the limit of large N, we have

$$\begin{aligned}\mathop{\mathrm{E}}_{f}\left[\Pr_{\mathcal{D}}[f]^2\right] &= \frac{1}{4eN^2 2^{2N}}\left(\sum_s\mathop{\mathrm{E}}_{f}\left[\left(e^{\widehat{f}(s)}+e^{-\widehat{f}(s)}\right)^2\right]+\sum_{s\neq t}\mathop{\mathrm{E}}_{f}\left[\left(e^{\widehat{f}(s)}+e^{-\widehat{f}(s)}\right)\left(e^{\widehat{f}(t)}+e^{-\widehat{f}(t)}\right)\right]\right) \\ &= \frac{1}{4eN^2 2^{2N}}\left(\begin{array}{c}\sum_s\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}e^{-x^2/2}\left(e^x+e^{-x}\right)^2dx \\ +\sum_{s\neq t}\left[\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty}e^{-x^2/2}\left(e^x+e^{-x}\right)dx\right]^2\end{array}\right) \\ &= \frac{1}{4eN^2 2^{2N}}\left[\left(2e^2+2\right)N+4eN(N-1)\right] \\ &= \frac{1}{2^{2N}}\left(1+\frac{(e-1)^2}{2eN}\right).\end{aligned}$$

Hence

$$\mathop{\mathrm{Var}}_{f}\left[\Pr_{\mathcal{D}}[f]\right]=\mathop{\mathrm{E}}_{f}\left[\Pr_{\mathcal{D}}[f]^2\right]-\mathop{\mathrm{E}}_{f}\left[\Pr_{\mathcal{D}}[f]\right]^2=\frac{(e-1)^2}{2eN2^{2N}}.$$

So by Cauchy-Schwarz,

$$\operatorname*{E}_{f}\left[\left|\Pr_{\mathcal{D}}[f]-\Pr_{\mathcal{U}}[f]\right|\right]\leq\sqrt{\operatorname*{Var}_{f}\left[\Pr_{\mathcal{D}}[f]\right]}=\frac{e-1}{\sqrt{2eN}}\cdot\frac{1}{2^{N}}$$

and

$$\|\mathcal{D}-\mathcal{U}\|\leq\frac{e-1}{2\sqrt{2eN}}.$$

■

An immediate corollary of Lemma 14 is that, if a Fourier Fishing algorithm succeeds with probability p on $\langle f_1,\ldots,f_n\rangle$ drawn from $\mathcal{U}^n$, then it also succeeds with probability at least

$$p-\|\mathcal{D}^n-\mathcal{U}^n\|\geq p-\frac{(e-1)\,n}{2\sqrt{2eN}}$$

on $\langle f_1,\ldots,f_n\rangle$ drawn from $\mathcal{D}^n$.

4.3 Putting It All Together

Using the results of Sections 4.1 and 4.2, we are now ready to prove a lower bound on the constant-depth circuit complexity of Fourier Fishing.

Theorem 15 *Any depth-d circuit that solves the* Fourier Fishing *problem, with probability at least* 0.99 *over $f_1,\ldots,f_n$ chosen uniformly at random, has size* $\exp\left(\Omega\left(N^{1/(2d+8)}\right)\right)$.

Proof. Let C be a circuit of depth d and size s. Let G be the set of all $\langle f_1,\ldots,f_n\rangle$ on which C *succeeds*: that is, for which it outputs $z_1,\ldots,z_n$, at least 75% of which satisfy $\left|\widehat{f}_i(z_i)\right|\geq 1$ and at least 25% of which satisfy $\left|\widehat{f}_i(z_i)\right|\geq 2$. Suppose

$$\Pr_{\mathcal{U}^n}[\langle f_1,\ldots,f_n\rangle\in G]\geq 0.99.$$

Then by Lemma 14, we also have

$$\Pr_{\mathcal{D}^n}[\langle f_1,\ldots,f_n\rangle\in G]\geq 0.99-\frac{(e-1)\,n}{2\sqrt{2eN}}\geq 0.98$$

for sufficiently large n.

Using the above fact, we will convert C into a new circuit C' that solves the ε-Bias Detection problem of Corollary 12, with $\varepsilon:=\frac{1}{2\sqrt{N}}$. This C' will have depth $d'=d+2$ and size $S'=O(NS)$. By Corollary 12, this will imply that C itself must have had size

$$\begin{aligned}S&=\exp\left(\Omega\left(1/\varepsilon^{1/(d'+2)}\right)\right)\\&=\exp\left(\Omega\left(N^{1/(2d+8)}\right)\right).\end{aligned}$$

Let $M=N^2n$, and let $R=r_1\ldots r_M\in\{0,1\}^M$ be a string of bits where each r_j is 1 with independent probability p. We want to decide whether $p=1/2$ or $p=1/2+\varepsilon$—that is, whether R

was drawn from $\mathcal{U}[0]$ or $\mathcal{U}[\varepsilon]$. We can do this as follows. First, choose strings $s_1, \ldots, s_n \in \{0,1\}^n$, bits $b_1, \ldots, b_n \in \{0,1\}$, and an integer $k \in [n]$ uniformly at random. Next, define Boolean functions $f_1, \ldots, f_n : \{0,1\}^n \to \{-1,1\}$ using the first Nn bits of R, like so:

$$f_i(x) := (-1)^{r_{(i-1)N+x} + s_i \cdot x + b_i}.$$

Finally, feed $\langle f_1, \ldots, f_n \rangle$ as input to C, and consider z_k, the k^{th} output of C (discarding the other $n-1$ outputs). We are interested in $\Pr[z_k = s_k]$, where the probability is over R, $s_1, \ldots, s_n$, $b_1, \ldots, b_n$, and k.

If $p = 1/2$, notice that $f_1, \ldots, f_n$ are independent and uniformly random regardless of $s_1, \ldots, s_n$. So C gets no information about s_k, and $\Pr[z_k = s_k] = 1/N$.

On the other hand, if $p = 1/2 + \varepsilon$, then each f_i is drawn independently from the distribution $\mathcal{D}[s]$ studied in Lemma 13. So by the Lemma, for every $i \in [n]$, if $\left|\widehat{f_i}(z_i)\right| \geq \beta$ then

$$\Pr_{f_i}[z_i = s_i] \geq \frac{e^{\beta} + e^{-\beta}}{2\sqrt{e}N}.$$

So assuming C succeeds (that is, $\langle f_1, \ldots, f_n \rangle \in G$), we have

$$\Pr_{f_1,\ldots,f_n,k}[z_k = s_k] \geq \frac{1}{4}\left(\frac{e^2 + e^{-2}}{2\sqrt{e}N}\right) + \frac{1}{2}\left(\frac{e^1 + e^{-1}}{2\sqrt{e}N}\right) \geq \frac{1.038}{N}.$$

So for a *random* $\langle f_1, \ldots, f_n \rangle$ drawn according to $\mathcal{D}^n$,

$$\Pr_{f_1,\ldots,f_n,k}[z_k = s_k] \geq 0.98\left(\frac{1.038}{N}\right) \geq \frac{1.017}{N}.$$

Notice that this is bounded above $1/N$ by a multiplicative constant.

Now let us repeat the above experiment N times. That is, for all $j := 1$ to N, we generate Boolean functions $f_{j1}, \ldots, f_{jn} : \{0,1\}^n \to \{-1,1\}$ by the same probabilistic procedure as before, but each time using a new Nn-bit substring of R_j of R, as well as new s, b, and k values (denoted $s_{j1}, \ldots, s_{jn}$, $b_{j1}, \ldots, b_{jn}$, and k_j). We then apply C to each n-tuple $\langle f_{j1}, \ldots, f_{jn} \rangle$. Let z_j be the k_j^{th} string that C outputs when run on $\langle f_{j1}, \ldots, f_{jn} \rangle$. Then by the above, for each $j \in [N]$ we have

$$p = \frac{1}{2} \Longrightarrow \Pr\left[z_j = s_{jk_j}\right] = \frac{1}{N},$$
$$p = \frac{1}{2} + \varepsilon \Longrightarrow \Pr\left[z_j = s_{jk_j}\right] \geq \frac{1.017}{N}.$$

Furthermore, these probabilities are independent across the different j's. So let E be the event that there *exists* a $j \in [N]$ such that $z_j = s_{jk_j}$. Then if $p = 1/2$ we have

$$\Pr[E] = 1 - \left(1 - \frac{1}{N}\right)^N \approx 1 - \frac{1}{e} \leq 0.633,$$

while if $p = 1/2 + \varepsilon$ we have

$$\Pr[E] \geq 1 - \left(1 - \frac{1.017}{N}\right)^N \geq 0.638.$$

It should now be clear how to create the circuit C', which distinguishes $R \in \{0,1\}^M$ drawn from $\mathcal{U}[0]$ from R drawn from $\mathcal{U}[\varepsilon]$ with constant bias. For each $j \in [N]$, generate an n-tuple of Boolean functions $\langle f_{j1}, \ldots, f_{jn} \rangle$ from R and apply C to it; then check whether there exists a $j \in [N]$ such that $z_j = s_{jk_j}$. This checking step can be done by a depth-2 circuit of size $O(Nn)$. Therefore, C' will have depth $d' = d + 2$ and size $s' = O(Ns)$. A technicality is that our choices of the s_{ji}'s, b_{ji}'s, and k_j's were made randomly. However, by Yao's principle, there clearly *exist* s_{ji}'s, b_{ji}'s, and k_j's such that

$$\Pr_{\mathcal{U}[\varepsilon]}\left[C'(R)\right] - \Pr_{\mathcal{U}[0]}\left[C'(R)\right] \geq 0.638 - 0.633 = 0.005.$$

So in forming C', we simply hardwire those choices. ■

Combining Theorem 15 with standard diagonalization tricks, we can now prove an oracle separation (in fact, a *random* oracle separation) between the complexity classes FBQP and $\mathsf{FBPP}^{\mathsf{PH}}$.

Theorem 16 $\mathsf{FBQP}^A \not\subset \mathsf{FBPP}^{\mathsf{PH}^A}$ *with probability* 1 *for a random oracle* A.

Proof. We interpret the oracle A as encoding n random Boolean functions $f_{n1}, \ldots, f_{nn} : \{0,1\}^n \rightarrow \{-1,1\}$ for each positive integer n. Let R be the relational problem where we are given 0^n as input, and *succeed* if and only if we output strings $z_1, \ldots, z_n \in \{0,1\}^n$, at least 3/4 of which satisfy $\left|\widehat{f}_{ni}(z_i)\right| \geq 1$ and at least 1/4 of which satisfy $\left|\widehat{f}_{ni}(z_i)\right| \geq 2$. Then by Lemmas 7 and 8, there exists an FBQP^A machine M such that for all n,

$$\Pr\left[M(0^n) \text{ succeeds}\right] \geq 1 - \frac{1}{\exp(n)},$$

where the probability is over both A and the quantum randomness. Hence $\Pr\left[M(0^n) \text{ succeeds}\right] \geq 1 - 1/\exp(n)$ on all but finitely many n, with probability 1 over A. Since we can simply hardwire the answers on the n's for which M fails, it follows that $R \in \mathsf{FBQP}^A$ with probability 1 over A.

On the other hand, let M be an $\mathsf{FBPP}^{\mathsf{PH}^A}$ machine. Then by the standard conversion between PH and AC^0, for every n there exists a probabilistic AC^0 circuit $C_{M,n}$, of size $2^{\mathrm{poly}(n)} = 2^{\mathrm{polylog}(N)}$, that takes A as input and simulates $M(0^n)$. By Yao's principle, we can assume without loss of generality that $C_{M,n}$ is deterministic, since the oracle A is already random. Then by Theorem 15,

$$\Pr_A\left[C_{M,n} \text{ succeeds}\right] < 0.99$$

for all sufficiently large n. By the independence of the f_{ni}'s, this is true even if we condition on $C_{M,1}, \ldots, C_{M,n-1}$ succeeding. So as in the standard random oracle argument of Bennett and Gill [10], for every fixed M we have

$$\Pr_A\left[C_{M,1}, C_{M,2}, C_{M,3}, \ldots \text{ succeed}\right] = 0.$$

So by the union bound,

$$\Pr_A\left[\exists M : C_{M,1}, C_{M,2}, C_{M,3}, \ldots \text{ succeed}\right] = 0$$

as well. It follows that $\mathsf{FBQP}^A \not\subset \mathsf{FBPP}^{\mathsf{PH}^A}$ with probability 1 over A. ■

If we "scale down by an exponential," then we can eliminate the need for the oracle A, and get a relation problem that is solvable in quantum *logarithmic* time but not in AC^0.

Theorem 17 *There exists a relation problem solvable in* $\mathsf{BQLOGTIME}$ *but not in* AC^0.

Proof. In our relation problem R, the input (of size $M = 2^n n$) will encode the truth tables of n Boolean functions, $f_1, \ldots, f_n : \{0,1\}^n \rightarrow \{-1,1\}$, which are promised to be "good" as defined in Section 2.1. The task is to solve Promise Fourier Fishing on $\langle f_1, \ldots, f_n \rangle$.

By Lemma 7, there exists a quantum algorithm that runs in $O(n) = O(\log M)$ time, making random accesses to the truth tables of $f_1, \ldots, f_n$, that solves R with probability $1 - 1/\exp(n) = 1 - 1/M^{\Omega(1)}$.

On the other hand, suppose R is in AC^0. Then we get a nonuniform circuit family $\{C_n\}_n$, of depth $O(1)$ and size $\operatorname{poly}(M) = 2^{O(n)}$, that solves Fourier Fishing on all tuples $\langle f_1, \ldots, f_n \rangle$ that are good. Recall that by Lemma 8, a $1 - 1/\exp(n)$ fraction of $\langle f_1, \ldots, f_n \rangle$'s are good. Therefore $\{C_n\}_n$ actually solves Fourier Fishing with probability $1 - 1/\exp(n)$ on $\langle f_1, \ldots, f_n \rangle$ chosen uniformly at random. But this contradicts Theorem 15.

Hence $R \in \mathsf{FBQLOGTIME} \setminus \mathsf{FAC}^0$ (where $\mathsf{FBQLOGTIME}$ and FAC^0 are the relation versions of $\mathsf{BQLOGTIME}$ and AC^0 respectively). ■

5 The Classical Complexity of Fourier Checking

Section 4 settled the relativized BQP versus PH question, if we are willing to talk about relation problems. Ultimately, though, we also care about decision problems. So in this section we consider the Fourier Checking problem, of deciding whether two Boolean functions f, g are independent or forrelated. In Section 3.2, we saw that Fourier Checking has quantum query complexity $O(1)$. What is its classical query complexity?[10]

It is not hard to give a classical algorithm that solves Fourier Checking using $O\left(\sqrt{N}\right) = O\left(2^{n/2}\right)$ queries. The algorithm is as follows: for some $K = \Theta\left(\sqrt{N}\right)$, first choose sets $X = \{x_1, \ldots, x_K\}$ and $Y = \{y_1, \ldots, y_K\}$ of n-bit strings uniformly at random. Then query $f(x_i)$ and $g(y_i)$ for all $i \in [K]$. Finally, compute

$$Z := \sum_{i,j=1}^{K} f(x_i)(-1)^{x_i \cdot y_j} g(y_j),$$

accept if $|Z|$ is greater than some cutoff cK, and reject otherwise. For suitable K and c, one can show that this algorithm accepts a forrelated $\langle f, g \rangle$ pair with probability at least $2/3$, and accepts a random $\langle f, g \rangle$ pair with probability at least $1/3$. We omit the details of the analysis, as they are tedious and not needed elsewhere in the paper.

In the next section, we will show that Fourier Checking has a property called *almost k-wise independence*, which immediately implies a lower bound of $\Omega\left(\sqrt[4]{N}\right) = \Omega\left(2^{n/4}\right)$ on its classical query complexity (as well as exponential lower bounds on its MA, $\mathsf{BPP_{path}}$, and SZK query complexities). Indeed, we conjecture that almost k-wise independence is enough to imply that Fourier Checking is not in PH. We discuss the status of that conjecture in Section 6.

[10] So long as we consider the distributional version of Fourier Checking, the deterministic and randomized query complexities are the same (by Yao's principle).

5.1 Almost k-Wise Independence

Let $Z = z_1 \ldots z_M \in \{-1,1\}^M$ be a string. Then a *literal* is a term of the form $\frac{1 \pm z_i}{2}$, and a *k-term* is a product of k literals (each involving a different z_i), which is 1 if the literals all take on prescribed values and 0 otherwise.

Let $\mathcal{U}$ be the uniform distribution over $\{-1,1\}^M$. The following definition will play a major role in this work.

Definition 18 *A distribution $\mathcal{D}$ over $\{-1,1\}^M$ is ε-almost k-wise independent if for every k-term C,*

$$1 - \varepsilon \leq \frac{\Pr_{\mathcal{D}}[C]}{\Pr_{\mathcal{U}}[C]} \leq 1 + \varepsilon.$$

(Note that $\Pr_{\mathcal{U}}[C]$ is just 2^{-k}.)

Now let $M = 2^{n+1} = 2N$, and let $\mathcal{F}$ be the forrelated distribution over pairs of Boolean functions $f, g : \{0,1\}^n \rightarrow \{-1,1\}$. That is, we sample $\langle f, g \rangle \in \mathcal{F}$ by first choosing a vector $v = (v_x)_{x \in \{-1,1\}^n} \in \mathbb{R}^N$ of independent $\mathcal{N}(0,1)$ Gaussians, then setting $f(x) := \operatorname{sgn}(v_x)$ for all x and $g(y) := \operatorname{sgn}(\widehat{v}_y)$ for all y.

Theorem 19 *For all $k \leq \sqrt[4]{N}$, the forrelated distribution $\mathcal{F}$ is $O\left(k^2/\sqrt{N}\right)$-almost k-wise independent.*

Proof. As a first step, we will prove an analogous statement for the real-valued functions $F(x) := v_x$ and $G(y) := \widehat{v}_y$; then we will generalize to the discrete versions $f(x)$ and $g(y)$. Let $\mathcal{U}'$ be the probability measure over $\langle F, G \rangle$ that corresponds to case (i) of Fourier Checking: that is, we choose each $F(x)$ and $G(y)$ independently from the Gaussian measure $\mathcal{N}(0,1)$. Let $\mathcal{F}'$ be the probability measure over $\langle F, G \rangle$ that corresponds to case (ii) of Fourier Checking: that is, we choose each $F(x)$ independently from $\mathcal{N}(0,1)$, then set $G(y) := \widehat{F}(y)$ where

$$\widehat{F}(y) = \frac{1}{\sqrt{N}} \sum_{x \in \{0,1\}^n} (-1)^{x \cdot y} F(x)$$

is the Fourier transform of F. Observe that since the Fourier transform is unitary, G has the same marginal distribution as F under $\mathcal{F}'$: namely, a product of independent $\mathcal{N}(0,1)$ Gaussians.

Fix inputs $x_1, \ldots, x_K \in \{0,1\}^n$ of F and $y_1, \ldots, y_L \in \{0,1\}^n$ of G, for some $K, L \leq N^{1/4}$. Then given constants $a_1, \ldots, a_K, b_1, \ldots, b_L \in \mathbb{R}$, let S be the set of all $\langle F, G \rangle$ that satisfy the $K + L$ equations

$$\begin{aligned} F(x_i) &= a_i \text{ for all } 1 \leq i \leq K, \\ G(y_j) &= b_j \text{ for all } 1 \leq j \leq L. \end{aligned} \qquad (1)$$

Clearly S is a $(2N - K - L)$-dimensional affine subspace of $\mathbb{R}^{2N}$. The *measure* of S, under some probability measure μ on $\mathbb{R}^{2N}$, is defined in the usual way as

$$\mu(S) := \int_{\langle F, G \rangle \in S} \mu(F, G) \, d\langle F, G \rangle.$$

Now let

$$\Delta_S := a_1^2 + \cdots + a_K^2 + b_1^2 + \cdots + b_L^2$$

be the squared distance between S and the origin (that is, the minimum squared 2-norm of any point in S). Then by the spherical symmetry of the Gaussian measure, it is not hard to see that S has measure

$$\mathcal{U}'(S) = \frac{e^{-\Delta_S/2}}{\sqrt{2\pi}^{K+L}}$$

under $\mathcal{U}'$. Our key claim is that

$$1 - O\left(\frac{(K+L)\,\Delta_S}{\sqrt{N}}\right) \leq \frac{\mathcal{F}'(S)}{\mathcal{U}'(S)} \leq 1 + O\left(\frac{(K+L)\,\Delta_S}{\sqrt{N}}\right).$$

To prove this claim: recall that the probability measure over F induced by $\mathcal{F}'$ is just a spherical Gaussian $\mathcal{G}$ on $\mathbb{R}^N$, and that $G = \widehat{F}$ uniquely determines F and vice versa. So consider the $(N-K-L)$-dimensional affine subspace T of $\mathbb{R}^N$ defined by the $K+L$ equations

$$\begin{aligned} F(x_i) &= a_i \text{ for all } 1 \leq i \leq K, \\ \widehat{F}(y_j) &= b_j \text{ for all } 1 \leq j \leq L. \end{aligned}$$

Then $\mathcal{F}'(S) = \mathcal{G}(T)$: that is, to compute how much measure $\mathcal{F}'$ assigns to S, it suffices to compute how much measure $\mathcal{G}$ assigns to T. We have

$$\mathcal{G}(T) = \frac{e^{-\Delta_T/2}}{\sqrt{2\pi}^{K+L}},$$

where Δ_T is the squared Euclidean distance between T and the origin. Thus, our problem reduces to minimizing

$$\Delta_F := \sum_{x \in \{0,1\}^n} F(x)^2$$

over all $F \in T$. By a standard fact about quadratic optimization, the minimal $F \in T$ will have the form

$$F(x) = \alpha_1 E_1(x) + \cdots + \alpha_K E_K(x) + \beta_1 \chi_1(x) + \cdots + \beta_L \chi_L(x)$$

where

$$E_i(x) := \begin{cases} 1 & \text{if } x = x_i \\ 0 & \text{otherwise} \end{cases}$$

is an indicator function, and

$$\chi_j(x) := \frac{(-1)^{x \cdot y_j}}{\sqrt{N}}$$

is the y_j^{th} Fourier character evaluated at x. Furthermore, the coefficients $\{\alpha_i\}_{i\in[K]}, \{\beta_j\}_{j\in[L]}$ can be obtained by solving the linear system

$$\underbrace{\begin{pmatrix} 1 & 0 & 0 & \pm1/\sqrt{N} & \cdots & \pm1/\sqrt{N} \\ 0 & \ddots & 0 & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & \pm1/\sqrt{N} & \cdots & \pm1/\sqrt{N} \\ \pm1/\sqrt{N} & \cdots & \pm1/\sqrt{N} & 1 & 0 & 0 \\ \vdots & \ddots & \vdots & 0 & \ddots & 0 \\ \pm1/\sqrt{N} & \cdots & \pm1/\sqrt{N} & 0 & 0 & 1 \end{pmatrix}}_{A} \underbrace{\begin{pmatrix} \alpha_1 \\ \vdots \\ \alpha_K \\ \beta_1 \\ \vdots \\ \beta_L \end{pmatrix}}_{u} = \underbrace{\begin{pmatrix} a_1 \\ \vdots \\ a_K \\ b_1 \\ \vdots \\ b_L \end{pmatrix}}_{w}$$

Here A is simply a matrix of covariances: the top left block records the inner product between each E_i and E_j (and hence is a $K \times K$ identity matrix), the bottom right block records the inner product between each χ_i and χ_j (and hence is an $L \times L$ identity matrix), and the remaining two blocks of size $K \times L$ record the inner product between each E_i and χ_j.

Thus, to get the vector of coefficients $u \in \mathbb{R}^{K+L}$, we simply need to calculate $A^{-1}w$. Define $B := I - A$. Then by Taylor series expansion,

$$A^{-1} = (I - B)^{-1} = I + B + B^2 + B^3 + \cdots$$

Notice that every entry of B is at most $1/\sqrt{N}$ in absolute value. This means that, for all positive integers t, every entry of B^t is at most

$$\frac{(K+L)^{t-1}}{N^{t/2}}$$

in absolute value. Since $K + L \ll \sqrt{N}$, this in turn means that every entry of $I - A^{-1}$ has absolute value $O\left(1/\sqrt{N}\right)$. So A^{-1} is exponentially close to the identity matrix. Hence, when we compute the vector $u = A^{-1}w$, we find that

$$\alpha_i = a_i + \varepsilon_i \text{ for all } 1 \leq i \leq K,$$
$$\beta_j = b_j + \delta_j \text{ for all } 1 \leq j \leq L,$$

for some small error terms ε_i and δ_j. Specifically, each ε_i and δ_j is the inner product of w, a $(K + L)$-dimensional vector of length $\sqrt{\Delta_S}$, with a vector every entry of which has absolute value $O\left(1/\sqrt{N}\right)$. By Cauchy-Schwarz, this implies that

$$|\varepsilon_i|, |\delta_j| = O\left(\frac{\sqrt{(K+L)\,\Delta_S}}{\sqrt{N}}\right)$$

for all i, j. So

$$\begin{aligned}
\Delta_T &= \min_{F \in T} \sum_{x \in \{0,1\}^n} F(x)^2 \\
&= \sum_{i=1}^{K} \alpha_i^2 + \sum_{j=1}^{L} \beta_j^2 + 2 \sum_{i=1}^{K} \sum_{j=1}^{L} \frac{\alpha_i \beta_j}{\sqrt{N}} \\
&= \sum_{i=1}^{K} (a_i + \varepsilon_i)^2 + \sum_{j=1}^{L} (b_j + \delta_j)^2 + 2 \sum_{i=1}^{K} \sum_{j=1}^{L} \frac{(a_i + \varepsilon_i)(b_j + \delta_j)}{\sqrt{N}} \\
&= \Delta_S \pm O\left(\frac{(K+L)\Delta_S}{\sqrt{N}} + \frac{(K+L)^2 \Delta_S}{N} + \frac{(K+L)^3 \Delta_S}{N^{3/2}} \right) \\
&= \Delta_S \left(1 \pm O\left(\frac{K+L}{\sqrt{N}} \right) \right),
\end{aligned}$$

where the fourth line made repeated use of Cauchy-Schwarz, and the fifth line used the fact that $K + L \ll \sqrt{N}$. Hence

$$\begin{aligned}
\frac{\mathcal{F}'(S)}{\mathcal{U}'(S)} &= \frac{e^{-\Delta_T/2} / \sqrt{2\pi}^{K+L}}{e^{-\Delta_S/2} / \sqrt{2\pi}^{K+L}} \\
&= \exp\left(\frac{\Delta_S - \Delta_T}{2} \right) \\
&= \exp\left(\pm O\left(\frac{(K+L)\Delta_S}{\sqrt{N}} \right) \right) \\
&= 1 \pm O\left(\frac{(K+L)\Delta_S}{\sqrt{N}} \right)
\end{aligned}$$

which proves the claim.

To prove the theorem, we now need to generalize to the discrete functions f and g. Here we are given a term C that is a conjunction of $K + L$ inequalities: K of the form $F(x_i) \leq 0$ or $F(x_i) \geq 0$, and L of the form $G(y_j) \leq 0$ or $G(y_j) \geq 0$. If we fix $x_1, \ldots, x_K$ and $y_1, \ldots, y_L$, we can think of C as just a convex region of $\mathbb{R}^{K+L}$. Then given an affine subspace S as defined by equation (1), we will (abusing notation) write $S \in C$ if the vector $(\alpha_1, \ldots, \alpha_K, \beta_1, \ldots, \beta_L)$ is in C: that is, if S is compatible with the $K + L$ inequalities that define C. We need to show that the ratio

$\Pr_{\mathcal{F}}[C] / \Pr_{\mathcal{U}}[C]$ is close to 1. We can do so using the previous result, as follows:

$$\begin{aligned}
\frac{\Pr_{\mathcal{F}}[C]}{\Pr_{\mathcal{U}}[C]} &= \frac{\int_{S\in C} \mathcal{F}'(S)\, dS}{\int_{S\in C} \mathcal{U}'(S)\, dS} \\
&= \frac{\int_{S\in C} \mathcal{U}'(S) \left[1 \pm O\left(\frac{(K+L)\Delta_S}{\sqrt{N}}\right)\right] dS}{\int_{S\in C} \mathcal{U}'(S)\, dS} \\
&= \frac{\int_{S\in C} \left[e^{-\Delta_S/2} / \sqrt{2\pi}^{K+L}\right] \left[1 \pm O\left(\frac{(K+L)\Delta_S}{\sqrt{N}}\right)\right] dS}{\int_{S\in C} \left[e^{-\Delta_S/2} / \sqrt{2\pi}^{K+L}\right] dS} \\
&= \frac{(1/2)^{K+L} \pm O\left(\int_{S\in C} \left[e^{-\Delta_S/2} / \sqrt{2\pi}^{K+L}\right] \frac{(K+L)\Delta_S}{\sqrt{N}} dS\right)}{(1/2)^{K+L}} \\
&= 1 \pm \frac{2^{K+L}(K+L)}{\sqrt{N}} O\left(\int_{S\in C} \frac{e^{-\Delta_S/2}}{\sqrt{2\pi}^{K+L}} \Delta_S dS\right) \\
&= 1 \pm \frac{K+L}{\sqrt{N}} O\left(\int_{S} \frac{e^{-\Delta_S/2}}{\sqrt{2\pi}^{K+L}} \Delta_S dS\right) \\
&= 1 \pm O\left(\frac{(K+L)^2}{\sqrt{N}}\right).
\end{aligned}$$

Setting $k := K + L$, this completes the proof. ■

5.2 Oracle Separation Results

The following lemma shows that *any* almost k-wise independent distribution is indistinguishable from the uniform distribution by $\mathsf{BPP_{path}}$ or SZK machines.

Lemma 20 *Suppose a probability distribution $\mathcal{D}$ over oracle strings is $1/t(n)$-almost* poly (n)*-wise independent, for some superpolynomial function t. Then no $\mathsf{BPP_{path}}$ machine or SZK protocol can distinguish $\mathcal{D}$ from the uniform distribution $\mathcal{U}$ with non-negligible bias.*

Proof. Let M be a $\mathsf{BPP_{path}}$ machine, and let $p_{\mathcal{D}}$ be the probability that M accepts an oracle string drawn from distribution $\mathcal{D}$. Then $p_{\mathcal{D}}$ can be written as $a_{\mathcal{D}}/s_{\mathcal{D}}$, where $s_{\mathcal{D}}$ is the fraction of M's computation paths that are postselected, and $a_{\mathcal{D}}$ is the fraction of M's paths that are both postselected and accepting. Since each computation path can examine at most poly (n) bits and $\mathcal{D}$ is $1/t(n)$-almost poly (n)-wise independent, we have

$$1 - \frac{1}{t(n)} \leq \frac{a_{\mathcal{D}}}{a_{\mathcal{U}}} \leq 1 + \frac{1}{t(n)} \quad \text{and} \quad 1 - \frac{1}{t(n)} \leq \frac{s_{\mathcal{D}}}{s_{\mathcal{U}}} \leq 1 + \frac{1}{t(n)}.$$

Hence

$$\left(1 - \frac{1}{t(n)}\right)^2 \leq \frac{a_{\mathcal{D}}/s_{\mathcal{D}}}{a_{\mathcal{U}}/s_{\mathcal{U}}} \leq \left(1 + \frac{1}{t(n)}\right)^2.$$

Now let P be an SZK protocol. Then by a result of Sahai and Vadhan [31], there exist polynomial-time samplable distributions A and A' such that if P accepts, then $\|A - A'\| \leq 1/3$, while if P rejects, then $\|A - A'\| \geq 2/3$. But since each computation path can examine at most $\operatorname{poly}(n)$ oracle bits and $\mathcal{D}$ is $1/t(n)$-almost $\operatorname{poly}(n)$-wise independent, we have $\|A_{\mathcal{D}} - A_{\mathcal{U}}\| \leq 1/t(n)$ and $\|A'_{\mathcal{D}} - A'_{\mathcal{U}}\| \leq 1/t(n)$, where the subscript denotes the distribution from which the oracle string was drawn. Hence

$$\left| \|A_{\mathcal{D}} - A'_{\mathcal{D}}\| - \|A_{\mathcal{U}} - A'_{\mathcal{U}}\| \right| \leq \|A_{\mathcal{D}} - A_{\mathcal{U}}\| + \|A'_{\mathcal{D}} - A'_{\mathcal{U}}\| \leq \frac{2}{t(n)}$$

and no SZK protocol exists. ■

We now combine Lemma 20 and Theorem 19 with standard diagonalization tricks, to obtain an oracle relative to which $\mathsf{BQP} \not\subset \mathsf{BPP}_{\mathsf{path}}$ and $\mathsf{BQP} \not\subset \mathsf{SZK}$.

Theorem 21 *There exists an oracle A relative to which $\mathsf{BQP}^A \not\subset \mathsf{BPP}^A_{\mathsf{path}}$ and $\mathsf{BQP}^A \not\subset \mathsf{SZK}^A$.*

Proof. The oracle A will encode the truth tables of Boolean functions $f_1, f_2, \ldots$ and $g_1, g_2, \ldots$, where $f_n, g_n : \{0,1\}^n \to \{-1,1\}$ are on n variables each. For each n, with 1/2 probability we draw $\langle f_n, g_n \rangle$ from the uniform distribution $\mathcal{U}$, and with 1/2 probability we draw $\langle f_n, g_n \rangle$ from the forrelated distribution $\mathcal{F}$. Let L be the unary language consisting of all 0^n for which $\langle f_n, g_n \rangle$ was drawn from $\mathcal{F}$.

By Theorem 9, there exists a BQP^A machine M that decides L on all but finitely many values of n, with probability 1 over A. Since we can simply hardwire the values of n on which M fails, it follows that $L \in \mathsf{BQP}^A$ with probability 1 over A.

On the other hand, we showed in Theorem 19 that $\mathcal{F}$ is $O\left(p(n)^2 / 2^{n/2}\right)$-almost $p(n)$-wise independent for all polynomials p. Hence, by Lemma 20, no $\mathsf{BPP}_{\mathsf{path}}$ machine can distinguish $\mathcal{F}$ from $\mathcal{U}$ with non-negligible bias. Let $E_n(M)$ be the event that the $\mathsf{BPP}^A_{\mathsf{path}}$ machine M correctly decides whether $0^n \in L$. Then

$$\Pr_A [E_n(M)] \leq \frac{1}{2} + o(1),$$

and moreover this is true even conditioning on $E_1(M), \ldots, E_{n-1}(M)$. So as in the standard random oracle argument of Bennett and Gill [10], for every fixed M we have

$$\Pr_A [E_1(M) \wedge E_2(M) \wedge \cdots] = 0.$$

So by the union bound,

$$\Pr_A [\exists M : E_1(M) \wedge E_2(M) \wedge \cdots] = 0$$

as well. It follows that $\mathsf{BQP}^A \not\subset \mathsf{BPP}^A_{\mathsf{path}}$ with probability 1 over A. By exactly the same argument, we also get $\mathsf{BQP}^A \not\subset \mathsf{SZK}^A$ with probability 1 over A. ■

Since $\mathsf{BPP} \subseteq \mathsf{MA} \subseteq \mathsf{BPP}_{\mathsf{path}}$, Theorem 21 supersedes the previous results that there exist oracles A relative to which $\mathsf{BPP}^A \neq \mathsf{BQP}^A$ [11] and $\mathsf{BQP}^A \not\subset \mathsf{MA}^A$ [38].

6 The Generalized Linial-Nisan Conjecture

In 1990, Linial and Nisan [28] famously conjectured that "polylogarithmic independence fools AC^0"—or loosely speaking, that every probability distribution $\mathcal{D}$ over n-bit strings that is uniform on all small subsets of bits, is *indistinguishable* from the uniform distribution by polynomial-size, constant-depth circuits. We now state a variant of the Linial-Nisan Conjecture, not with the best possible parameters but with weaker, easier-to-understand parameters that suffice for our application.

Conjecture 22 (Linial-Nisan Conjecture) *Let $\mathcal{D}$ be an $n^{\Omega(1)}$-wise independent distribution over $\{0,1\}^n$, and let $f : \{0,1\}^n \rightarrow \{0,1\}$ be computed by an AC^0 circuit of size $2^{n^{o(1)}}$ and depth $O(1)$. Then*

$$\left| \Pr_{x \sim \mathcal{D}} [f(x)] - \Pr_{x \sim \mathcal{U}} [f(x)] \right| = o(1).$$

After seventeen years of almost no progress, in 2007 Bazzi [7] finally proved Conjecture 22 for the special case of depth-2 circuits (also called DNF formulas). Bazzi's proof was about 50 pages, but it was dramatically simplified a year later, when Razborov [30] discovered a 3-page proof. Then in 2009, Braverman [13] gave a breakthrough proof of the full Linial-Nisan Conjecture.

Theorem 23 (Braverman's Theorem [13]) *Let $f : \{0,1\}^n \rightarrow \{0,1\}$ be computed by an AC^0 circuit of size S and depth d, and let $\mathcal{D}$ be a $\left(\log \frac{S}{\varepsilon}\right)^{7d^2}$-wise independent distribution over $\{0,1\}^n$. Then for all sufficiently large S,*

$$\left| \Pr_{x \sim \mathcal{D}} [f(x)] - \Pr_{x \sim \mathcal{U}} [f(x)] \right| \leq \varepsilon.$$

We conjecture a modest-seeming extension of Braverman's Theorem, which says (informally) that *almost* k-wise independent distributions fool AC^0 as well.

Conjecture 24 (Generalized Linial-Nisan or GLN Conjecture) *Let $\mathcal{D}$ be a $1/n^{\Omega(1)}$-almost $n^{\Omega(1)}$-wise independent distribution over $\{0,1\}^n$, and let $f : \{0,1\}^n \rightarrow \{0,1\}$ be computed by an AC^0 circuit of size $2^{n^{o(1)}}$ and depth $O(1)$. Then*

$$\left| \Pr_{x \sim \mathcal{D}} [f(x)] - \Pr_{x \sim \mathcal{U}} [f(x)] \right| = o(1).$$

By the usual correspondence between AC^0 and PH, the GLN Conjecture immediately implies the following counterpart of Lemma 20.

> *Suppose a probability distribution $\mathcal{D}$ over oracle strings is $1/t(n)$-almost* $\operatorname{poly}(n)$*-wise independent, for some superpolynomial function t. Then no PH machine can distinguish $\mathcal{D}$ from the uniform distribution $\mathcal{U}$ with non-negligible bias.*

And thus we get the following implication:

Theorem 25 *Assuming the GLN Conjecture, there exists an oracle A relative to which $\mathsf{BQP}^A \not\subset \mathsf{PH}^A$.*

Proof. The proof is the same as that of Theorem 21; the only difference is that the GLN Conjecture now plays the role of Lemma 20. ■

Likewise:

Theorem 26 *Assuming the GLN Conjecture for the special case of depth-*2 *circuits (i.e., DNF formulas), there exists an oracle A relative to which $\mathsf{BQP}^A \not\subset \mathsf{AM}^A$.*

Proof. Just like in Theorem 21, define an oracle A and an associated language L using the Fourier Checking problem. Then $L \in \mathsf{BQP}^A$, with probability 1 over the choices made in constructing A. On the other hand, suppose $L \in \mathsf{AM}^A$ with probability 1 over A. Then we claim that Fourier Checking can also be solved by a family of DNF formulas $\{\varphi_n\}_{n\geq 1}$ of size $2^{\operatorname{poly}(n)}$:

$$\left| \Pr_{\langle f,g\rangle \sim \mathcal{F}} [\varphi_n(f,g)] - \Pr_{\langle f,g\rangle \sim \mathcal{U}} [\varphi_n(f,g)] \right| = \Omega(1).$$

But since $\mathcal{F}$ is $O\left(k^2/2^{n/2}\right)$-almost k-wise independent (by Theorem 19), such a family φ_n would violate the depth-2 case of the GLN Conjecture.

We now prove the claim. For simplicity, fix an input length n, and let A refer to a single instance $\langle f,g\rangle$ of Fourier Checking.[11] Let P be an AM protocol that successfully distinguishes the forrelated distribution $\mathcal{F}$ over $\langle f,g\rangle$ pairs from the uniform distribution $\mathcal{U}$. We can assume without loss of generality that P is *public-coin* [23]. In other words, Arthur first sends a random challenge $r \in \{0,1\}^{\operatorname{poly}(n)}$ to Merlin, then Merlin responds with a witness $w \in \{0,1\}^{\operatorname{poly}(n)}$, then Arthur runs a deterministic polynomial-time verification procedure $V^A(r,w)$ to decide whether to accept. By the assumption that P succeeds,

$$\left| \Pr_{A\sim\mathcal{D},r} \left[\exists w : V^A(r,w)\right] - \Pr_{A\sim\mathcal{D},r} \left[\exists w : V^A(r,w)\right] \right| = \Omega(1).$$

So by Yao's principle, there exists a *fixed* challenge r^* such that

$$\left| \Pr_{A\sim\mathcal{D}} \left[\exists w : V^A(r^*,w)\right] - \Pr_{A\sim\mathcal{D}} \left[\exists w : V^A(r^*,w)\right] \right| = \Omega(1).$$

Now let $Q_{A,w}$ be the set of all queries that $V^A(r^*,w)$ makes to A, and let $C_{A,w}(A')$ be a term (i.e., a conjunction of 1's and 0's) that returns TRUE if and only if A' agrees with A on all queries in $Q_{A,w}$. Then we can assume without loss of generality that $C_w := C_{A,w}$ depends only on w, not on A—since Merlin can simply *tell* Arthur what queries V is going to make and what their outcomes will be, and Arthur can reject if Merlin is lying. Let W be the set of all witnesses w such that Arthur accepts if $C_w(A)$ returns TRUE. Consider the DNF formula

$$\varphi(A) := \bigvee_{w\in W} C_w(A),$$

which expresses that there exists a w causing $V^A(r^*,w)$ to accept. Then φ contains at most $2^{\operatorname{poly}(n)}$ terms with $\operatorname{poly}(n)$ literals each, and

$$\left| \Pr_{A\sim\mathcal{D}} [\varphi(A)] - \Pr_{A\sim\mathcal{D}} [\varphi(A)] \right| = \Omega(1).$$

■

[11] It is straightforward to generalize to the case where Arthur can query other instances, besides the one he is trying to solve.

As a side note, it is conceivable that one could prove

$$\Pr_{x\sim\mathcal{D}}\left[\varphi\left(x\right)\right]-\Pr_{x\sim\mathcal{U}}\left[\varphi\left(x\right)\right]=o\left(1\right)$$

for every almost k-wise independent distribution $\mathcal{D}$ and small *CNF* formula φ, without getting the same result for *DNF* formulas (or vice versa). However, since BQP is closed under complement, even such an asymmetric result would imply an oracle A relative to which $\mathsf{BQP}^A \not\subset \mathsf{AM}^A$.

If the GLN Conjecture holds, then we can also "scale down by an exponential," to obtain an *unrelativized* decision problem that is solvable in quantum logarithmic time but not in AC^0.

Theorem 27 *Assuming the GLN Conjecture, there exists a promise problem in* $\mathsf{BQLOGTIME}$ *that is not in* AC^0.

Proof. In our promise problem $\Pi = (\Pi_{\text{YES}}, \Pi_{\text{NO}})$, the inputs (of size $M = 2^{n+1}$) will encode pairs of Boolean functions $f, g : \{0,1\}^n \to \{-1,1\}$, such that

$$p\left(f,g\right) := \frac{1}{N^3}\left(\sum_{x,y\in\{0,1\}^n} f\left(x\right)\left(-1\right)^{x\cdot y} g\left(y\right)\right)^2$$

is either at least 0.05 or at most 0.01. The problem is to accept in the former case and reject in the latter case.

Using the algorithm `FC-ALG` from Section 3.2, it is immediate that $\Pi \in \mathsf{BQLOGTIME}$. On the other hand, suppose $\Pi \in \mathsf{AC}^0$. Then we get a nonuniform circuit family $\{C_n\}_n$, of depth $O\left(1\right)$ and size $\operatorname{poly}\left(M\right) = 2^{O(n)}$, that solves FOURIER CHECKING on all pairs $\langle f,g\rangle$ such that (i) $p\left(f,g\right) \leq 0.01$ or (ii) $p\left(f,g\right) \geq 0.05$. By Corollary 10, the class (i) includes the overwhelming majority of $\langle f,g\rangle$'s drawn from the uniform distribution $\mathcal{U}$, while the class (ii) includes a constant fraction of $\langle f,g\rangle$'s drawn from the forrelated distribution $\mathcal{F}$. Therefore, we actually obtain an AC^0 circuit family that distinguishes $\mathcal{U}$ from $\mathcal{F}$ with constant bias. But this contradicts Theorem 19 together with the GLN Conjecture. ■

6.1 Low-Fat Polynomials

Given that the GLN Conjecture would have such remarkable implications for quantum complexity theory, the question arises of how we can go about proving it. As we are indebted to Louay Bazzi for pointing out to us, the GLN Conjecture is *equivalent* to the following conjecture, about approximating AC^0 functions by low-degree polynomials.

Conjecture 28 (Low-Fat Sandwich Conjecture) *For every function* $f : \{0,1\}^n \to \{0,1\}$ *computable by an* AC^0 *circuit, there exist polynomials* $p_\ell, p_u : \mathbb{R}^n \to \mathbb{R}$ *of degree* $k = n^{o(1)}$ *that satisfy the following three conditions.*

(i) ***Sandwiching:*** $p_\ell\left(x\right) \leq f\left(x\right) \leq p_u\left(x\right)$ *for all* $x \in \{0,1\}^n$.

(ii) L_1***-Approximation:*** $\operatorname{E}_{x\sim\mathcal{U}}\left[p_u\left(x\right) - p_\ell\left(x\right)\right] = o\left(1\right)$.

(iii) ***Low-Fat:*** $p_\ell\left(x\right)$ *and* $p_u\left(x\right)$ *can be written as linear combinations of terms,* $p_\ell\left(x\right) = \sum_C \alpha_C C\left(x\right)$ *and* $p_u\left(x\right) = \sum_C \beta_C C\left(x\right)$ *respectively, such that* $\sum_C \left|\alpha_C\right| 2^{-|C|} = n^{o(1)}$ *and* $\sum_C \left|\beta_C\right| 2^{-|C|} = n^{o(1)}$. *(Here a term is a product of literals of the form* x_i *and* $1 - x_i$*.)*

If we take out condition (iii), then Conjecture 28 becomes equivalent to the *original* Linial-Nisan Conjecture (see Bazzi [7] for a proof). And indeed, all progress so far on "Linial-Nisan problems" has crucially relied on this connection with polynomials. Bazzi [7] and Razborov [30] proved the depth-2 case of the LN Conjecture by constructing low-degree, approximating, sandwiching polynomials for every DNF, while Braverman [13] proved the full LN Conjecture by constructing such polynomials for every AC^0 circuit.[12] Given this history, proving Conjecture 28 would seem like the "obvious" approach to proving the GLN Conjecture.

Below we prove one direction of the equivalence: that to prove the GLN Conjecture, it suffices to construct low-fat sandwiching polynomials for every AC^0 circuit. The other direction—that the GLN Conjecture implies Conjecture 28, and hence, there is no loss of generality in working with polynomials instead of probability distributions—follows from a linear programming duality calculation that we omit.

Theorem 29 *The Low-Fat Sandwich Conjecture implies the GLN Conjecture.*

Proof. Given an AC^0 function f, let p_ℓ, p_u be the low-fat sandwiching polynomials of degree k that are guaranteed by Conjecture 28. Also, let $\mathcal{D}$ be an ε-almost k-wise independent distribution over $\{0,1\}^n$, for some $\varepsilon = 1/n^{\Omega(1)}$. Then

$$\begin{aligned}
\Pr_{x\sim\mathcal{D}}[f(x)] - \Pr_{x\sim\mathcal{U}}[f(x)] &\leq \mathop{\mathrm{E}}_{\mathcal{D}}[p_u] - \mathop{\mathrm{E}}_{\mathcal{U}}[p_\ell] \\
&= \sum_C \beta_C \mathop{\mathrm{E}}_{\mathcal{D}}[C] - \mathop{\mathrm{E}}_{\mathcal{U}}[p_\ell] \\
&\leq \sum_C \frac{\beta_C + |\beta_C|\,\varepsilon}{2^{|C|}} - \mathop{\mathrm{E}}_{\mathcal{U}}[p_\ell] \\
&= \mathop{\mathrm{E}}_{\mathcal{U}}[p_u - p_\ell] + \varepsilon \sum_C \frac{|\beta_C|}{2^{|C|}} \\
&= o(1) + \frac{n^{o(1)}}{n^{\Omega(1)}} \\
&= o(1).
\end{aligned}$$

Likewise,

$$\begin{aligned}
\Pr_{x\sim\mathcal{U}}[f(x)] - \Pr_{x\sim\mathcal{D}}[f(x)] &\leq \mathop{\mathrm{E}}_{\mathcal{U}}[p_u] - \mathop{\mathrm{E}}_{\mathcal{D}}[p_\ell] \\
&= \mathop{\mathrm{E}}_{\mathcal{U}}[p_u] - \sum_C \alpha_C \mathop{\mathrm{E}}_{\mathcal{D}}[C] \\
&\leq \mathop{\mathrm{E}}_{\mathcal{U}}[p_u] - \sum_C \frac{\alpha_C - |\alpha_C|\,\varepsilon}{2^{|C|}} \\
&= \mathop{\mathrm{E}}_{\mathcal{U}}[p_u - p_\ell] + \varepsilon \sum_C \frac{|\alpha_C|}{2^{|C|}} \\
&= o(1).
\end{aligned}$$

■

[12] Strictly speaking, Braverman constructed approximating polynomials with slightly different (though still sufficient) properties. We know from Bazzi [7] that it must be possible to get sandwiching polynomials as well.

7 Discussion

We now take a step back, and use our results to address some conceptual questions about the relativized BQP versus PH question, the GLN Conjecture, and what makes them so difficult.

The first question is an obvious one. Complexity theorists have known for decades how to prove constant-depth circuit lower bounds, and how to use those lower bounds to give oracles A relative to which (for example) $\mathsf{PP}^A \not\subset \mathsf{PH}^A$ and $\oplus\mathsf{P}^A \not\subset \mathsf{PH}^A$. So why should it be so much harder to give an A relative to which $\mathsf{BQP}^A \not\subset \mathsf{PH}^A$? What makes this AC^0 lower bound different from other AC^0 lower bounds?

The answer seems to be that, while we have powerful techniques for proving that a function f is not in AC^0, *all of those techniques, in one way or another, involve arguing that f is not approximated by a low-degree polynomial.* The Razborov-Smolensky technique [29, 35] argues this explicitly, while even the random restriction technique [16, 39, 36] argues it "implicitly," as shown by Linial, Mansour, and Nisan [27]. And this is a problem, if f is also computed by an efficient quantum algorithm. For Beals et al. [8] proved the following in 1998:

Lemma 30 ([8]) *Suppose a quantum algorithm Q makes T queries to a Boolean input $X \in \{0,1\}^N$. Then Q's acceptance probability is a real multilinear polynomial $p(X)$, of degree at most $2T$.*

In other words, if a function f is in BQP, then for that very reason, f *has* a low-degree approximating polynomial! As an example, we already saw that the following polynomial p, of degree 4, successfully distinguishes the forrelated distribution $\mathcal{F}$ from the uniform distribution $\mathcal{U}$:

$$p(f,g) := \frac{1}{N^3}\left(\sum_{x,y\in\{0,1\}^n} f(x)(-1)^{x\cdot y} g(y)\right)^2. \tag{2}$$

Therefore, we cannot hope to prove a lower bound for FOURIER CHECKING, by any argument that would also imply that such a p cannot exist.

This brings us to a second question. If

(i) every known technique for proving $f \notin \mathsf{AC}^0$ involves showing that f is not approximated by a low-degree polynomial, but

(ii) every function f with low quantum query complexity *is* approximated by a low-degree polynomial,

does that mean there is no hope of solving the relativized BQP versus PH problem using polynomial-based techniques?

We believe the answer is no. The essential point here is that an AC^0 function can be approximated by different *kinds* of low-degree polynomials. For example, Linial, Mansour, and Nisan [27] showed that, if $f : \{0,1\}^n \to \{0,1\}$ is in AC^0, then there exists a real polynomial $p : \mathbb{R}^n \to \mathbb{R}$, of degree polylog n, such that

$$\operatorname*{E}_{x\in\{0,1\}^n}\left[(p(x) - f(x))^2\right] = o(1).$$

By comparison, Razborov [29] and Smolensky [35] showed that if $f \in \mathsf{AC}^0$, then there exists a polynomial $p : \mathbb{F}^n \rightarrow \mathbb{F}$ over *any* field $\mathbb{F}$ (finite or infinite), of degree $\operatorname{polylog} N$, such that

$$\Pr_{x \in \{0,1\}^n} [p(x) \neq f(x)] = o(1).$$

Furthermore, to show that $f \notin \mathsf{AC}^0$, it suffices to show that f is not approximated by a low-degree polynomial in *any one* of these senses. For example, even though the Parity function has degree 1 over the finite field $\mathbb{F}_2$, Razborov and Smolensky showed that over other fields (such as $\mathbb{F}_3$), any degree-$o(\sqrt{n})$ polynomial disagrees with Parity on a large fraction of inputs—and that is enough to imply that Parity$\notin \mathsf{AC}^0$. In other words, we simply need to find a *type* of polynomial approximation that works for AC^0 circuits, but does not work for the Fourier Checking problem. If true, Conjecture 28 (the Low-Fat Sandwich Conjecture) provides exactly such a type of approximation.

But this raises another question: what is the significance of the "low-fat" requirement in Conjecture 28? Why, of all things, do we want our approximating polynomial p to be expressible as a linear combination of terms, $p(x) = \sum_C \alpha_C C(x)$, such that $\sum_C |\alpha_C| 2^{-|C|} = n^{o(1)}$?

The answer takes us to the heart of what an oracle separation between BQP and PH would have to accomplish. Notice that, although the polynomial p from equation (2) solved the Fourier Checking problem, it did so only by *cancelling massive numbers of positive and negative terms,* then representing the answer by the tiny residue left over. Not coincidentally, this sort of cancellation is a central feature of quantum algorithms. By contrast, Theorem 29 essentially says that, if a polynomial p does *not* involve such massive cancellations, but is instead more "conservative" and "reasonable" (like the polynomials that arise from classical decision trees), then p cannot distinguish almost k-wise independent distributions from the uniform distribution, and therefore cannot solve Fourier Checking. If Conjecture 28 holds, then every small-depth circuit can be approximated, not just by any low-degree polynomial, but by a "conservative," "reasonable" low-degree polynomial—one with a bound on the coefficients that prevents massive cancellations. This would prove that Fourier Checking has no small constant-depth circuits, and hence that there exists an oracle separating BQP from PH.

This brings us to the fourth and final question: how might one prove Conjecture 28? In particular, is it possible that some trivial modification of Braverman's proof [13] would give low-fat sandwiching polynomials, thereby establishing the GLN Conjecture?

While we cannot rule this out, we believe that the answer is no. For examining Braverman's proof, we find that it combines two kinds of polynomial approximations of AC^0 circuits: that of Linial-Mansour-Nisan [27], and that of Razborov [29] and Smolensky [35]. Unfortunately, *neither LMN nor Razborov-Smolensky gives anything like the control over the approximating polynomial's coefficients that Conjecture 28 demands.* LMN simply takes the Fourier transform of an AC^0 function and deletes the high-order coefficients; while Razborov-Smolensky approximates each OR gate by a product of randomly-chosen linear functions. Both techniques produce approximating polynomials with a huge number of monomials, and no reasonable bound on their coefficients. While it is conceivable that those polynomials satisfy the low-fat condition anyway—because of some non-obvious representation as a linear combination of terms—certainly neither LMN nor Razborov-Smolensky gives any idea what that representation would look like. Thus, we suspect that, to get the desired control over the coefficients, one will need more "constructive" proofs of both the LMN and Razborov-Smolensky theorems. Such proofs would likely be of great interest to circuit complexity and computational learning theory for independent reasons.

8 Open Problems

First, of course, prove the GLN Conjecture, or prove the existence of an oracle A relative to which $\mathsf{BQP}^A \not\subset \mathsf{PH}^A$ by some other means. A natural first step would be to prove the GLN Conjecture for the special case of DNFs: as shown in Theorem 26, this would imply an oracle A relative to which $\mathsf{BQP}^A \not\subset \mathsf{AM}^A$. We have offered a \$200 prize for the PH case and a \$100 prize for the AM case.[13]

Second, it would be of interest to prove the GLN Conjecture for classes of functions weaker than (or incomparable with) DNFs: for example, monotone DNFs, read-once formulas, and read-k-times formulas.

Third, can we give an example of a Boolean function $f : \{0,1\}^n \rightarrow \{-1,1\}$ that is well-approximated by a low-degree polynomial, but *not* by a low-degree low-fat polynomial? Here is a more concrete version of the challenge: let

$$\|f - p\| := \operatorname*{E}_{x\in\{0,1\}^n}\left[(f(x) - p(x))^2\right].$$

Then find a Boolean function f for which

(i) there exists a degree-$n^{o(1)}$ polynomial $p : \mathbb{R}^n \rightarrow \mathbb{R}$ such that $\|f - p\| = o(1)$, but

(ii) there does *not* exist a degree-$n^{o(1)}$ polynomial $q : \mathbb{R}^n \rightarrow \mathbb{R}$ such that $\|f - q\| = o(1)$ and q can be written as a linear combination of terms, $q(x) = \sum_C \alpha_C C(x)$, with $\sum_C |\alpha_C| 2^{-|C|} = n^{o(1)}$.

Fourth, can we give an oracle relative to which $\mathsf{BQP} \not\subset \mathsf{IP}$? What about an oracle relative to which $\mathsf{BQP} \neq \mathsf{IP}_{\mathsf{BQP}}$, where $\mathsf{IP}_{\mathsf{BQP}}$ is the class of problems that admit an interactive protocol with a BPP verifier and a BQP prover?[14]

Fifth, what other implications does the GLN Conjecture have? If we assume it, can we address other longstanding open questions in quantum complexity theory, such as those discussed in Section 1.1? For example, can we give an oracle relative to which $\mathsf{NP} \subseteq \mathsf{BQP}$ but $\mathsf{PH} \not\subset \mathsf{BQP}$, or an oracle relative to which $\mathsf{NP} \subseteq \mathsf{BQP}$ and PH is infinite?

Sixth, how much can we say about the BQP versus PH question in the unrelativized world? As one concrete challenge, can we find a nontrivial way to "realize" the FOURIER CHECKING oracle (in other words, an explicit computational problem that is solvable using FOURIER CHECKING)?

Seventh, how far can the gap between the success probabilities of FBQP and $\mathsf{FBPP}^{\mathsf{PH}}$ algorithms be improved? Theorem 15 gave a relation for which a quantum algorithm succeeds with probability $1 - c^{-n}$, whereas any $\mathsf{FBPP}^{\mathsf{PH}}$ algorithm succeeds with probability at most 0.99. By changing the success criterion for FOURIER FISHING—basically, by requiring the classical algorithm to output $z_1, \ldots, z_n$ such that $\widehat{f_1}(z_1)^2, \ldots, \widehat{f_n}(z_n)^2$ are distributed "almost exactly as they would be in the quantum algorithm"—one can improve the 0.99 to $1/2 + \varepsilon$ for any $\varepsilon > 0$. However, improving the constant further might require a direct product theorem for AC^0 circuits solving FOURIER FISHING.

[13] See http://scottaaronson.com/blog/?p=381

[14] If we let the verifier transmit unentangled qubits to the prover, then the resulting class $\mathsf{IP}_{\mathsf{BQP}}^{|\theta\rangle}$ actually equals BQP, as recently shown by Broadbent, Fitzsimons, and Kashefi [14] (see also Aharonov, Ben-Or, and Eban [4]). It is not known whether this $\mathsf{IP}_{\mathsf{BQP}}^{|\theta\rangle} = \mathsf{BQP}$ result relativizes; we conjecture that it does not.

9 Acknowledgments

I thank Louay Bazzi for reformulating the GLN Conjecture as the Low-Fat Sandwich Conjecture; and Andy Drucker, Lance Fortnow, and Sasha Razborov for helpful discussions.

References

[1] S. Aaronson. Quantum lower bound for the collision problem. In *Proc. ACM STOC*, pages 635–642, 2002. quant-ph/0111102.

[2] S. Aaronson. Quantum lower bound for recursive Fourier sampling. *Quantum Information and Computation*, 3(2):165–174, 2003. ECCC TR02-072, quant-ph/0209060.

[3] L. Adleman, J. DeMarrais, and M.-D. Huang. Quantum computability. *SIAM J. Comput.*, 26(5):1524–1540, 1997.

[4] D. Aharonov, M. Ben-Or, and E. Eban. Interactive proofs for quantum computations. arXiv:0810.5375, 2008.

[5] D. Aharonov, V. Jones, and Z. Landau. A polynomial quantum algorithm for approximating the Jones polynomial. In *Proc. ACM STOC*, pages 427–436, 2006. quant-ph/0511096.

[6] Z. Bar-Yossef, T. S. Jayram, and I. Kerenidis. Exponential separation of quantum and classical one-way communication complexity. *SIAM J. Comput.*, 38(1):366–384, 2008. Conference version in ACM STOC 2004. ECCC TR04-036.

[7] L. Bazzi. Polylogarithmic independence can fool DNF formulas. In *Proc. IEEE FOCS*, pages 63–73, 2007.

[8] R. Beals, H. Buhrman, R. Cleve, M. Mosca, and R. de Wolf. Quantum lower bounds by polynomials. *J. ACM*, 48(4):778–797, 2001. Earlier version in IEEE FOCS 1998, pp. 352-361. quant-ph/9802049.

[9] C. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and weaknesses of quantum computing. *SIAM J. Comput.*, 26(5):1510–1523, 1997. quant-ph/9701001.

[10] C. H. Bennett and J. Gill. Relative to a random oracle A, $P^A \neq NP^A \neq coNP^A$ with probability 1. *SIAM J. Comput.*, 10(1):96–113, 1981.

[11] E. Bernstein and U. Vazirani. Quantum complexity theory. *SIAM J. Comput.*, 26(5):1411–1473, 1997. First appeared in ACM STOC 1993.

[12] R. B. Boppana, J. Håstad, and S. Zachos. Does co-NP have short interactive proofs? *Inform. Proc. Lett.*, 25:127–132, 1987.

[13] M. Braverman. Poly-logarithmic independence fools AC^0 circuits. In *Proc. IEEE Conference on Computational Complexity*, pages 3–8, 2009. ECCC TR09-011.

[14] A. Broadbent, J. Fitzsimons, and E. Kashefi. Universal blind quantum computation. arXiv:0807.4154, 2008.

[15] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou. The complexity of computing a Nash equilibrium. *Commun. ACM*, 52(2):89–97, 2009. Earlier version in Proceedings of STOC'2006.

[16] M. Furst, J. B. Saxe, and M. Sipser. Parity, circuits, and the polynomial time hierarchy. *Math. Systems Theory*, 17:13–27, 1984.

[17] D. Gavinsky. Classical interaction cannot replace a quantum message. In *Proc. ACM STOC*, pages 95–102, 2008. quant-ph/0703215.

[18] D. Gavinsky. On the role of shared entanglement. *Quantum Information and Computation*, 8(1-2):82–95, 2008. quant-ph/0604052.

[19] D. Gavinsky. Predictive quantum learning. arXiv:0812.3429, 2009.

[20] D. Gavinsky, J. Kempe, I. Kerenidis, R. Raz, and R. de Wolf. Exponential separation for one-way quantum communication complexity, with applications to cryptography. *SIAM J. Comput.*, 38(5):1695–1708, 2008. Earlier version in STOC'2007. quant-ph/0611209.

[21] D. Gavinsky, J. Kempe, O. Regev, and R. de Wolf. Bounded-error quantum state identification and exponential separations in communication complexity. In *Proc. ACM STOC*, pages 594–603, 2006. quant-ph/0511013.

[22] D. Gavinsky and P. Pudlák. Exponential separation of quantum and classical non-interactive multi-party communication complexity. In *Proc. IEEE Conference on Computational Complexity*, pages 332–339, 2008. arXiv:0708.0859.

[23] S. Goldwasser and M. Sipser. Private coins versus public coins in interactive proof systems. In *Randomness and Computation*, volume 5 of *Advances in Computing Research*. JAI Press, 1989.

[24] F. Green and R. Pruim. Relativized separation of EQP from P^{NP}. *Inform. Proc. Lett.*, 80(5):257–260, 2001.

[25] Y. Han, L. Hemaspaandra, and T. Thierauf. Threshold computation and cryptographic security. *SIAM J. Comput.*, 26(1):59–78, 1997.

[26] A. Klivans and D. van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. *SIAM J. Comput.*, 31:1501–1526, 2002. Earlier version in ACM STOC 1999.

[27] N. Linial, Y. Mansour, and N. Nisan. Constant depth circuits, Fourier transform, and learnability. *J. ACM*, 40(3):607–620, 1993.

[28] N. Linial and N. Nisan. Approximate inclusion-exclusion. *Combinatorica*, 10(4):349–365, 1990. Earlier version in STOC'90.

[29] A. A. Razborov. Lower bounds for the size of circuits of bounded depth with basis $\{\&, \oplus\}$. *Mathematicheskie Zametki*, 41(4):598–607, 1987. English translation in *Math. Notes. Acad. Sci. USSR* 41(4):333–338, 1987.

[30] A. A. Razborov. A simple proof of Bazzi's theorem. *ACM Trans. on Computation Theory*, 1(1), 2009. ECCC TR08-081.

[31] A. Sahai and S. Vadhan. A complete promise problem for statistical zero-knowledge. *J. ACM*, 50(2):196–249, 2003. ECCC TR00-084. Earlier version in IEEE FOCS 1997.

[32] A. Shamir. IP=PSPACE. *J. ACM*, 39(4):869–877, 1992.

[33] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997. Earlier version in IEEE FOCS 1994. quant-ph/9508027.

[34] D. Simon. On the power of quantum computation. In *Proc. IEEE FOCS*, pages 116–123, 1994.

[35] R. Smolensky. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *Proc. ACM STOC*, pages 77–82, 1987.

[36] J. Håstad. *Computational Limitations for Small Depth Circuits.* MIT Press, 1987.

[37] E. Viola. On approximate majority and probabilistic time. In *Proc. IEEE Conference on Computational Complexity*, pages 155–168, 2007. Journal version to appear in Computational Complexity.

[38] J. Watrous. Succinct quantum proofs for properties of finite groups. In *Proc. IEEE FOCS*, pages 537–546, 2000. cs.CC/0009002.

[39] A. C-C. Yao. Separating the polynomial-time hierarchy by oracles (preliminary version). In *Proc. IEEE FOCS*, pages 1–10, 1985.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de