Forrelation: A Problem that Optimally Separates Quantum from Classical Computing*

Scott Aaronson†
MIT
aaronson@csail.mit.edu

Andris Ambainis‡
University of Latvia
ambainis@lu.lv

ABSTRACT

We achieve essentially the largest possible separation between quantum and classical query complexities. We do so using a property-testing problem called FORRELATION, where one needs to decide whether one Boolean function is highly correlated with the Fourier transform of a second function. This problem can be solved using 1 quantum query, yet we show that any randomized algorithm needs $\Omega(\sqrt{N}/\log N)$ queries (improving an $\Omega(N^{1/4})$ lower bound of Aaronson). Conversely, we show that this 1 versus $\widetilde{\Omega}(\sqrt{N})$ separation is optimal: indeed, any t-query quantum algorithm whatsoever can be simulated by an $O(N^{1-1/2t})$-query randomized algorithm. Thus, resolving an open question of Buhrman et al. from 2002, there is no partial Boolean function whose quantum query complexity is constant and whose randomized query complexity is linear. We conjecture that a natural generalization of FORRELATION achieves the optimal t versus $\Omega(N^{1-1/2t})$ separation for all t. As a bonus, we show that this generalization is BQP-complete. This yields what's arguably the simplest BQP-complete problem yet known, and gives a second sense in which FORRELATION "captures the maximum power of quantum computation."

1. BACKGROUND

Since the work of Simon [15] and Shor [14] two decades ago, we have had powerful evidence that quantum computers can achieve exponential speedups over classical computers. Of course, for problems like FACTORING, these speedups are conjectural at present: we cannot rule out that a fast classical factoring algorithm might exist. But in the *black-box model*, which captures most known quantum algorithms, exponential and even larger speedups can be *proved*. We know, for example, that PERIOD-FINDING (a natural abstraction of the problem solved by Shor's algorithm) is solvable with only $O(1)$ quantum queries, but requires $N^{\Omega(1)}$ classical randomized queries, where N is the number of input elements [8, 6, 12]. We also know that SIMON'S PROBLEM is solvable with $O(\log N)$ quantum queries, but requires $\Omega(\sqrt{N})$ classical queries; and that a similar separation holds for the GLUED-TREES problem introduced by Childs et al. [7, 11].[1]

To us, these results raise an extremely interesting question:

- **"The Speedup Question."** *Within the black-box model, just how large of a quantum speedup is possible? For example, could there be a function of N bits with a quantum query complexity of 1, but a classical randomized query complexity of $\Omega(N)$?*

One may object: once we know that exponential and even larger quantum speedups are possible in the black-box model, who cares about the exact limit? In our view, the central reason to study the Speedup Question is that doing so can help us better understand the nature of quantum speedups themselves. For example, can all exponential quantum speedups be seen as originating from a common cause? Is there a single problem or technique that captures the advantages of quantum over classical query complexity, in much the same way that random sampling could be said to capture the advantages of randomized over deterministic query complexity?

As far as we know, the Speedup Question was first posed by Buhrman et al. [5] around 2002, in their study of quantum property-testing. Specifically, Buhrman et al. asked whether there is any property of N-bit strings that exhibits a "maximal" separation: that is, one that requires $\Omega(N)$ queries to test classically, but only $O(1)$ quantumly. The best separation they could find, based on Simon's problem, was "deficient" on both ends: it required $\Omega(\sqrt{N})$ queries to test classically, and $O(\log N \log\log N)$ quantumly.

Since then, there has been only sporadic progress on the Speedup Question. In 2009, Aaronson [1] introduced the FORRELATION problem—a problem that we will revisit in this paper—and showed that it was solvable with only 1

*Extended abstract. For the full version, see www.scottaaronson.com/papers/for.pdf

†Supported by an NSF Waterman Award, under grant no. 1249349.

‡The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n°600700 (QALGO), ERC Advanced Grant MQC and Latvian State Research programme NexIT project No. 1.

STOC'15, June 14–17, 2015, Portland, Oregon, USA.

http://dx.doi.org/10.1145/2746539.2746547.

[1]However, in all these cases the queries are non-Boolean. If we insist on Boolean queries, then the quantum query complexities get multiplied by an $O(\log N)$ factor.

quantum query, but required $\Omega(N^{1/4})$ classical randomized queries. In 2010, Chakraborty et al. [6] argued that PERIOD-FINDING gives a different example of an $O(1)$ versus $\widetilde{\Omega}(N^{1/4})$ quantum/classical gap; there, however, we only get an $O(1)$-query quantum algorithm if we allow non-Boolean queries.

Earlier, in 2001, de Beaudrap, Cleve, and Watrous [4] had given what they described as a black-box problem that was solvable with 1 quantum query, but that required $\Omega(N^{1/4})$ or $\Omega(\sqrt{N})$ classical randomized queries (depending on how one defines the "input size" N). However, de Beaudrap et al. were not working within the usual model of quantum query complexity. Normally, one provides "black-box access" to a function f, meaning that the quantum algorithm can apply a unitary transformation that maps basis states of the form $|x,y\rangle$ to basis states of the form $|x,y\oplus f(x)\rangle$ (or $|x\rangle$ to $(-1)^{f(x)}|x\rangle$, if f is Boolean). By contrast, for their separation, de Beaudrap et al. had to assume the ability to map basis states of the form $|x,y\rangle$ to basis states of the form $|x,\pi(y+sx)\rangle$, for some unknown permutation π and hidden shift s.

2. OUR RESULTS

This paper has two main contributions—the largest quantum black-box speedup yet known, and a proof that that speedup is essentially optimal—as well as many smaller related contributions.

2.1 Maximal Quantum/Classical Separation

We undertake a detailed study of the FORRELATION problem, which Aaronson [1] introduced for a different purpose than the one that concerns us here (he was interested in an oracle separation between BQP and the polynomial hierarchy).[2] In FORRELATION, we are given access to two Boolean functions $f,g:\{0,1\}^n\rightarrow\{-1,1\}$. We want to estimate the amount of correlation between f and the Fourier transform of g—that is, the quantity

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

It is not hard to see that $|\Phi_{f,g}|\leq 1$ for all f,g. The problem is to decide, say, whether $|\Phi_{f,g}|\leq\frac{1}{100}$ or $\Phi_{f,g}\geq\frac{3}{5}$, promised that one of these is the case.[3] Here and throughout this paper, the "input size" is taken to be $N:=2^n$.

One can give (see Section 4.3) a quantum algorithm that solves FORRELATION, with bounded probability of error, using only 1 quantum query. Intuitively, however, the property of f and g being "forrelated" (that is, having large $\Phi_{f,g}$ value) is an extremely global property, which should not be apparent to a classical algorithm until it has queried a significant fraction of the entire truth tables of f and g. And indeed, improving an $\Omega(N^{1/4})$ lower bound of Aaronson [1], in Section 4.1 we show the following:

[2]Also, in [1], the problem was called "Fourier Checking."

[3]The reason for the asymmetry—i.e., for promising that $\Phi_{f,g}$ is positive if its absolute value is large, but not if its absolute value is small—is a bit technical. On the one hand, we want the "unforrelated" case to encompass almost all randomly-chosen functions f,g. On the other hand, we also want FORRELATION to be solvable using only 1 quantum query. If we had promised $|\Phi_{f,g}|\geq\frac{3}{5}$, rather than $\Phi_{f,g}\geq\frac{3}{5}$, then we would only know a 2-query quantum algorithm. In any case, none of these choices make a big difference to our results.

THEOREM 1. *Any classical randomized algorithm for* FORRELATION *must make* $\Omega(\frac{\sqrt{N}}{\log N})$ *queries.*

Theorem 1 yields *the largest quantum versus classical separation yet known* in the black-box model. As we show in the full version, Theorem 1 also implies the largest *property-testing* separation yet known—for with some work, one can recast FORRELATION (or rather, its negation) as a property that is testable with only 1 query quantumly, but that requires $\Omega(\frac{\sqrt{N}}{\log N})$ queries to test classically.

We deduce Theorem 1 as a consequence of a more general result: namely, a lower bound on the classical query complexity of a problem called GAUSSIAN DISTINGUISHING. Here we are given oracle access to a collection of $\mathcal{N}(0,1)$ real Gaussian random variables, $x_1,\ldots,x_M$. We are asked to decide whether the variables are all independent, or alternatively, whether they lie in a known low-dimensional subspace of $\mathbb{R}^M$: one that induces a covariance of at most ε between each pair of variables, while keeping each variable an $\mathcal{N}(0,1)$ Gaussian individually. We show the following:

THEOREM 2. GAUSSIAN DISTINGUISHING *requires* $\Omega\left(\frac{1/\varepsilon}{\log(M/\varepsilon)}\right)$ *classical randomized queries.*

Theorem 1 is then simply a (discretized) special case of Theorem 2, with $M=2N$ and $\varepsilon=1/\sqrt{N}$, the latter coming from the inner product between a standard basis vector and a Fourier basis vector. Beyond that, it seems to us that Theorem 2 could have independent applications in statistics and machine learning.

2.2 Proof of Optimality

We show that the quantum/classical query complexity separation achieved by the FORRELATION problem is close to the best possible. More generally:

THEOREM 3. *Let Q be any quantum algorithm that makes $t=O(1)$ queries to an N-bit string $X\in\{0,1\}^N$. Then we can estimate $\Pr[Q \text{ accepts } X]$, to constant additive error and with high probability, by making only $O(N^{1-1/2t})$ classical randomized queries to X.*[4] *Moreover, the randomized queries are nonadaptive.*

So for example, every 1-query quantum algorithm can be simulated by an $O(\sqrt{N})$-query classical randomized algorithm, every 2-query quantum algorithm can be simulated by an $O(N^{3/4})$-query randomized algorithm, and so on. Theorem 3 resolves the open problem of Buhrman et al. [5] in the negative: it shows that there is no problem (property-testing or otherwise) with a constant versus linear quantum/classical query complexity gap. Theorem 3 does not rule out the possibility of an $O(\log N)$ versus $\widetilde{\Omega}(N)$ gap, and indeed, we conjecture that such a gap is possible.

Once again, we deduce Theorem 3 as a consequence of a more general result, which might have independent applications to classical sublinear algorithms. Namely:

THEOREM 4. *Every degree-k real polynomial $p:\{-1,1\}^N\rightarrow\mathbb{R}$ that is*

[4]The reason for the condition $t=O(1)$ is that, in the bound $O(N^{1-1/2t})$, the big-O hides a multiplicative factor of $\exp(t)$. Thus, we can obtain a nontrivial upper bound on query complexity as long as $t=o(\sqrt{\log N})$.

(i) bounded in $[-1,1]$ *at every Boolean point, and*

(ii) "block-multilinear" (that is, the variables can be partitioned into k *blocks, such that every monomial is the product of one variable from each block),*

can be approximated to within $\pm\varepsilon$*, with high probability, by nonadaptively querying only* $O\left(\left(N/\varepsilon^2\right)^{1-1/k}\right)$ *of the variables.*

In the statement of Theorem 4, we strongly conjecture that condition (ii) can be removed. If so, then we would obtain a sublinear algorithm to estimate any bounded, constant-degree real polynomial. In the full version, we show that condition (ii) can indeed be removed in the special case $k = 2$.

2.2.1 k-fold Forrelation

Next, we study a natural generalization of FORRELATION. In k-fold FORRELATION, we are given access to k Boolean functions $f_1, \ldots, f_k : \{0,1\}^n \to \{-1,1\}$. We want to estimate the "twisted sum"

$$\Phi_{f_1,\ldots,f_k} := \frac{1}{2^{(k+1)n/2}} \sum_{x_1,\ldots,x_k \in \{0,1\}^n} \begin{matrix} f_1(x_1)(-1)^{x_1 \cdot x_2} \\ f_2(x_2)(-1)^{x_2 \cdot x_3} \\ \cdots (-1)^{x_{k-1} \cdot x_k} f_k(x_k) \end{matrix}$$

It is not hard to show that $|\Phi_{f_1,\ldots,f_k}| \leq 1$ for all $f_1, \ldots, f_k$. The problem is to decide, say, whether $|\Phi_{f_1,\ldots,f_k}| \leq \frac{1}{100}$ or $\Phi_{f_1,\ldots,f_k} \geq \frac{3}{5}$, promised that one of these is the case.

One can give (see Section 4.3) a quantum algorithm that solves k-fold FORRELATION, with bounded error probability, using only $\lceil k/2 \rceil$ quantum queries. In Section 4.4, we show, conversely, that k-fold FORRELATION "captures the full power of quantum computation":

THEOREM 5. *If* $f_1, \ldots, f_k$ *are described explicitly (say, by circuits to compute them), and* $k = \text{poly}(n)$*, then* k*-fold* FORRELATION *is a* BQP*-complete promise problem.*

This gives us a particularly "self-contained" complete problem for quantum computation. Not only can one state the k-fold FORRELATION problem without any notions from quantum mechanics, one does not need any nontrivial mathematical notions, like the condition number of a matrix or the Jones polynomial of a knot.

We conjecture, moreover, that k-fold FORRELATION achieves the optimal $k/2$ versus $\widetilde{\Omega}(N^{1-1/k})$ quantum/classical query complexity separation for all even k. If so, then there are *two* senses in which k-fold FORRELATION captures the power of quantum computation.

2.3 Other Results

In the full version of this paper, we also include several other results.

First, we study the largest possible quantum versus classical separations that are achievable for *approximate sampling* and *relation* problems. We show that there exists a sampling problem—namely, FOURIER SAMPLING of a Boolean function—that is solvable with 1 quantum query, but requires $\Omega(N/\log N)$ classical queries. By our previous results, this exceeds the largest quantum/classical gap that is possible for decision problems.

Second, we generalize our result that every 1-query quantum algorithm can be simulated using $O(\sqrt{N})$ randomized queries, to show that *every bounded degree-2 real polynomial* $p : \{-1,1\}^N \to [-1,1]$ can be estimated using $O(\sqrt{N})$ randomized queries. We conjecture that this can be generalized, to show that every bounded degree-k real polynomial can be estimated using $O(N^{1-1/k})$ randomized queries.

Third, we extend our $\Omega(\frac{\sqrt{N}}{\log N})$ randomized lower bound for the FORRELATION problem, to show a $\Omega(\frac{\sqrt{N}}{\log^{7/2} N})$ lower bound for k-fold FORRELATION for any $k \geq 2$. We conjecture that the right lower bound is $\widetilde{\Omega}(N^{1-1/k})$, but even generalizing our $\widetilde{\Omega}(\sqrt{N})$ lower bound to the k-fold case is nontrivial.

3. QUERY COMPLEXITY

Briefly, by the *query complexity* of an algorithm $\mathcal{A}$, we mean the number of queries that $\mathcal{A}$ makes to its input $z = (z_1, \ldots, z_N)$, maximized over all valid inputs z.[5] The query complexity of a function F is then the minimum query complexity of *any* algorithm $\mathcal{A}$ (of a specified type—classical, quantum, etc.) that outputs $F(z)$, with bounded probability of error, given any valid input z.

One slightly unconventional choice that we make is to define "bounded probability of error" to mean "error probability at most $1/2 - \varepsilon$, for some constant $\varepsilon > 0$" rather than "error probability at most $1/3$." The reason is that we will be able to design a 1-query quantum algorithm that solves the FORRELATION problem with error probability $2/5$, but *not* one that solves it with error probability $1/3$. Of course, one can make the error probability as small as one likes using amplification, but doing so increases the query complexity by a constant factor.

We assume throughout this paper that the input $z \in \{-1,1\}^N$ is Boolean, and we typically work in the $\{-1,1\}$ basis for convenience. In the classical setting, each query returns a single bit z_i, for some index $i \in [N]$ specified by $\mathcal{A}$. In the quantum setting, each query performs a diagonal unitary transformation

$$|i, w\rangle \to z_i |i, w\rangle,$$

where w represents "workspace qubits" that do not participate in the query.[6] Between two queries, $\mathcal{A}$ can apply any unitary transformation it likes that does not depend on z.

In this paper, the input $z = (z_1, \ldots, z_N)$ will typically consist of the truth tables of one or more Boolean functions: for example, $f, g : \{0,1\}^n \to \{-1,1\}$, or $f_1, \ldots, f_k : \{0,1\}^n \to \{-1,1\}$. Throughout, we use n for the number of input bits that these Boolean functions take (which roughly corresponds to the number of *qubits* in a quantum algorithm), and we use $N = 2^n$ for the number of bits being queried in superposition. (Strictly speaking, we should set $N = k \cdot 2^n$, where k is the number of Boolean functions. But this constant-factor difference will not matter for us.) Thus, for the purposes of query complexity, N is the "input size," in terms of which we state our upper and lower bounds.

[5] If we are talking about a partial Boolean function, then a "valid" input is simply any input that satisfies the promise.

[6] For Boolean inputs z, this is well-known to be exactly equivalent to a different definition of a quantum query, wherein each basis state $|i, a, w\rangle$ gets mapped to $|i, a \oplus z_i, w\rangle$. Here a represents a 1-qubit "answer register."

4. TECHNIQUES

4.1 Randomized Lower Bound

Proving that any randomized algorithm for FORRELATION requires $\Omega(\frac{\sqrt{N}}{\log N})$ queries is surprisingly involved. As we mentioned in Section 2.1, the first step, following the work of Aaronson [1], is to convert FORRELATION into an analogous problem involving real Gaussian variables. In REAL FORRELATION, we are given oracle access to two real functions $f, g : \{0,1\}^n \rightarrow \mathbb{R}$, and are promised either that (i) every $f(x)$ and $g(y)$ value is an independent $\mathcal{N}(0,1)$ Gaussian, or else (ii) every $f(x)$ value is an independent $\mathcal{N}(0,1)$ Gaussian, while every $g(y)$ value equals $\hat{f}(y)$ (i.e., the Fourier transform of f evaluated at y). The problem is to decide which holds. Using a rounding reduction, we show that any query complexity lower bound for REAL FORRELATION implies the same lower bound for FORRELATION itself.

More formally, in the full version we prove the following.

THEOREM 6. *Suppose real functions $f, g : \{0,1\}^n \rightarrow \mathbb{R}$ are drawn according to the measure (ii) above. Define Boolean functions $F, G : \{0,1\}^n \rightarrow \{-1,1\}$ by $F(x) := \mathrm{sgn}(f(x))$ and $G(y) := \mathrm{sgn}(g(y))$. Then*

$$\mathrm{E}_{f,g}[\Phi_{F,G}] = \frac{2}{\pi} \pm O\left(\frac{\log N}{N}\right).$$

Earlier, Aaronson [1, Theorem 9] proved a variant of Theorem 6, but with a badly suboptimal constant: he was only able to show that

$$\mathrm{E}[\Phi_{F,G}] \geq \cos\left(2 \arccos\sqrt{\frac{2}{\pi}}\right) - o(1) \approx 0.273,$$

compared to the exact value of $2/\pi \approx 0.637$ that we get here.

Theorem 6 has the following corollary, as we prove in the full version.

COROLLARY 7. *Suppose there exists a T-query algorithm that solves* FORRELATION *with bounded error. Then there also exists an $O(T)$-query algorithm that solves* REAL FORRELATION *with bounded error.*

So to prove a lower bound for FORRELATION, it suffices to prove the same lower bound for REAL FORRELATION. Furthermore, because the REAL FORRELATION problem is to distinguish two probability distributions, we can assume without loss of generality that any algorithm for the latter is deterministic.

Making the problem continuous allows us to adopt a geometric perspective. In this perspective, we are given oracle access to a real vector $v \in \mathbb{R}^{2N}$, whose $2N$ coordinates consist of all values $f(x)$ and all values $g(y)$ (recall that $N = 2^n$). We are trying to distinguish the case where v is simply an $\mathcal{N}(0,1)^{2N}$ Gaussian, from the case where v is confined to an N-dimensional subspace of $\mathbb{R}^{2N}$—namely, the subspace defined by $g = \hat{f}$. Now, suppose that values $f(x_1), \ldots, f(x_t)$ and $g(y_1), \ldots, g(y_u)$ have already been queried. Then we can straightforwardly calculate the Bayesian posterior probabilities for being in case (i) or case (ii). For case (i), the probability turns out to depend solely on the squared 2-norm of the vector of empirical data seen so far:

$$\Pr[\text{case (i)}] \propto \exp\left(-\frac{\Delta_{\mathrm{i}}}{2}\right),$$

where

$$\Delta_{\mathrm{i}} = f(x_1)^2 + \cdots + f(x_t)^2 + g(y_1)^2 + \cdots + g(y_u)^2.$$

For case (ii), by contrast, the probability is proportional to $\exp(-\Delta_{\mathrm{ii}}/2)$, where Δ_{ii} is the minimum squared 2-norm of any point $f \in \mathbb{R}^N$ compatible with all the data seen so far, as well as with the linear constraint $g = \hat{f}$. Let $\mathcal{V} = \left\{|1\rangle, \ldots, |N\rangle, |\hat{1}\rangle, \ldots, |\hat{N}\rangle\right\}$ be the set of $2N$ unit vectors in $\mathbb{R}^N$ that consists of all N elements of the standard basis, together with all N elements of the Fourier basis. (For convenience, we will often use ket notation, even to represent vectors that are not quantum states.) Then Δ_{ii}, in turn, can be calculated using a process of Gram-Schmidt orthogonalization, on the vectors in $\mathcal{V}$ corresponding to the f-values and g-values that have been queried so far.

More formally, given an arbitrary collection of linearly-independent unit vectors $|v_1\rangle, |v_2\rangle \ldots$, the *Gram-Schmidt process* produces orthonormal vectors by recursively projecting each $|v_i\rangle$ onto the orthogonal complement of the subspace spanned by $|v_1\rangle$ up to $|v_{i-1}\rangle$, and then normalizing the result. That is:

$$|z_i\rangle = |v_i\rangle - \sum_{j=1}^{i-1} \langle v_i|w_j\rangle |w_j\rangle,$$
$$|w_i\rangle = \beta_i |z_i\rangle$$

where $\beta_i = \frac{1}{\sqrt{\langle z_i|z_i\rangle}}$ is a normalizing constant. Note that $\langle z_i|z_i\rangle \leq 1$ (since $|z_i\rangle$ is the projection of a unit vector onto a subspace), and hence $\beta_i \geq 1$.

To calculate Δ_{ii}, we need to understand the behavior of the Gram-Schmidt process when the $|v_i\rangle$'s are already very close to orthogonal. We can control that behavior with the help of the following lemma, proved in the full version.

LEMMA 8 (GRAM-SCHMIDT LEMMA). *Let $|v_1\rangle, \ldots, |v_t\rangle$ be unit vectors with $|\langle v_i|v_j\rangle| \leq \varepsilon$ for all $i \neq j$, and suppose $t \leq 0.1/\varepsilon$ (so in particular, $\varepsilon \leq 0.1$). Let $|w_i\rangle$ and β_i be as above. Then for all $i > j$, we have*

$$|\langle v_i|w_j\rangle| \leq \varepsilon + 2j\varepsilon^2,$$
$$\beta_i \leq 1 + 2i\varepsilon^2.$$

So in particular, under the stated hypothesis, $|\langle v_i|w_j\rangle| \leq 1.2\varepsilon$ and $\beta_i \leq 1 + 0.2\varepsilon$.

For recall: our goal is to show that, with high probability, Δ_{i} and Δ_{ii} *remain close to each other*, even after a large number of queries have been made—meaning that the algorithm has not yet succeeded in distinguishing case (i) from case (ii) with non-negligible bias. To show this, we need to use the fact that the vectors in $\mathcal{V}$ are nearly-orthogonal: that is, for all $|v\rangle, |w\rangle \in \mathcal{V}$, we have $|\langle v|w\rangle| \leq \frac{1}{\sqrt{N}}$. Intuitively, this means that, if we restrict attention to any small subset S of f-values and g-values, then while correlations exist among those values, the correlations are *weak*: "to a first approximation," we have simply asked for the projections of a Gaussian vector onto $|S|$ orthogonal directions, and have therefore received $|S|$ uncorrelated answers.

From this perspective, the key question is: how many values can we query until the "orthogonal approximation" breaks down (meaning that we notice the correlations)? In

his previous work, Aaronson [1] showed that the approximation holds until $\Omega(N^{1/4})$ queries are made. Indeed, he proved a stronger statement: even if the x's and y's are chosen *nondeterministically*, still $\Omega(N^{1/4})$ values must be revealed until we have a *certificate* showing that we are in case (i) or case (ii) with high probability.

To improve the lower bound from $\Omega(N^{1/4})$ to the optimal $\widetilde{\Omega}(\sqrt{N})$, there are several hurdles to overcome.

Aaronson had assumed, conservatively, that the deviations from orthogonality all "pull in the same direction." As a first step, we notice instead that the deviations follow an unbiased random walk, with some positive and others negative—the martingale property arising from the fact that the algorithm can control which x's and y's to query, but not the values of $f(x)$ and $g(y)$. We then use a Gaussian generalization of Azuma's inequality to upper-bound the sum of the deviations. Doing this improves the lower bound from $\Omega(N^{1/4})$ to $\widetilde{\Omega}(N^{1/3})$, but we then hit an apparent barrier.

In this work, we explain the $\Omega(N^{1/3})$ barrier, by exhibiting a "model problem" that is extremely similar to Real Forrelation (in particular, has exactly the same near-orthogonality property), yet is solvable with only $O(N^{1/3})$ queries, by exploiting adaptivity.

In more detail, recall the generalization of Real Forrelation that we call Gaussian Distinguishing. Here we are given a finite set $\mathcal{V}$ of unit vectors in $\mathbb{R}^N$, called "test vectors." (In our case, $\mathcal{V}$ happens to equal $\left\{|1\rangle, |\hat{1}\rangle, \ldots, |N\rangle, |\hat{N}\rangle\right\}$.) In each step, we are allowed to pick any test vector $|v\rangle \in \mathcal{V}$ that hasn't been picked in previous steps. We then "query" $|v\rangle$, getting back a real-valued response $a_v \in \mathbb{R}$. The problem is to distinguish the following two cases, with constant bias:

(i) Each a_v is drawn independently from $\mathcal{N}(0,1)$.

(ii) Each a_v equals $\langle \Psi | v \rangle$, where $|\Psi\rangle \in \mathbb{R}^N$ is a vector drawn from $\mathcal{N}(0,1)^N$ that is fixed throughout the algorithm.

In our case (Real Forrelation), we have $M = 2N$ and $\varepsilon = 1/\sqrt{N}$. Thus, one might hope one could prove a general lower bound on Gaussian Distinguishing, only depending on ε, of the form $\widetilde{\Omega}(1/\varepsilon)$. Unfortunately, this is impossible: one does *not* have a $\widetilde{\Omega}(1/\varepsilon)$ lower bound on query complexity independent of M, but at best a $\Omega((1/\varepsilon)^{2/3})$ lower bound.[7] In the context of Real Forrelation, this means that, if the only thing we knew about $\mathcal{V}$ was that $|\langle v|w\rangle| \leq 1/\sqrt{N}$ for all distinct $|v\rangle, |w\rangle \in \mathcal{V}$ (so in particular, we had no upper bound on $\mathcal{V}$'s cardinality), then we could not hope to prove any lower bound better than $\Omega(N^{1/3})$.[8]

However, we then break this barrier, by using the fact that, for Real Forrelation (but *not* for the model problem), the total number of vectors in $\mathcal{V}$ is only $N^{O(1)}$. This fact lets us use the Gaussian Azuma's inequality a second time, to upper-bound not only the *sum* of all the deviations from orthogonality, but the individual deviations themselves. Implementing this yields a lower bound of $\widetilde{\Omega}(N^{2/5})$: better than $\widetilde{\Omega}(N^{1/3})$, but still not all the way up to $\widetilde{\Omega}(\sqrt{N})$. However, we then notice that we can apply Azuma's inequality *recursively*—once for each layer of the Gram-Schmidt orthogonalization process—to get better and better upper bounds on the deviations from orthogonality. Doing so gives us a sequence of lower bounds $\widetilde{\Omega}(N^{3/7})$, $\widetilde{\Omega}(N^{4/9})$, $\widetilde{\Omega}(N^{5/11})$, etc., with the ultimate limit of the process being $\Omega(\frac{\sqrt{N}}{\log N})$.

Our argument actually proves a *general* lower bound for Gaussian Distinguishing, which works whenever $|\mathcal{V}|$ is not too large, and every pair of vectors in $\mathcal{V}$ is sufficiently close to orthogonal. Here is our general result:

Theorem 9. *Suppose $|\mathcal{V}| \leq M$, and $|\langle v|w\rangle| \leq \varepsilon$ for all distinct vectors $|v\rangle, |w\rangle \in \mathcal{V}$. Then any classical algorithm for* Gaussian Distinguishing *must make $\Omega\left(\frac{1/\varepsilon}{\log(M/\varepsilon)}\right)$ queries.*

4.2 Randomized Upper Bound

Why did we have to work so hard to prove a $\widetilde{\Omega}(\sqrt{N})$ lower bound on the randomized query complexity of Forrelation? Our other main result provides one possible explanation: namely, we are here scraping up against the "ceiling" of the possible separations between randomized and quantum query complexity. In particular, *any* quantum algorithm that makes 1 query to a Boolean input $X \in \{0,1\}^N$, can be simulated by a randomized algorithm (in fact, a nonadaptive randomized algorithm) that makes $O(\sqrt{N})$ queries to X. More generally, any quantum algorithm that makes $t = O(1)$ queries to X, can be simulated by a nonadaptive randomized algorithm that makes $O(N^{1-1/2t})$ queries to X.

The proof of this result consists of three steps. The first involves the simulation of quantum algorithms by low-degree polynomials. In 1998, Beals et al. [3] famously observed that, if a quantum algorithm makes t queries to a Boolean input $X \in \{-1,1\}^N$, then $p(X)$, the probability that the algorithm accepts X, can be written as a multilinear polynomial in X of degree at most $2t$. We extend this result of Beals et al., in a way that might be of independent interest for quantum lower bounds. Namely, we observe that every t-query quantum algorithm gives rise, not merely to

[7] Here is the example that shows this: let $|1\rangle, \ldots, |n\rangle$ be orthogonal unit vectors. Then for all 2^n strings $z = z_1 \cdots z_n \in \{-1,1\}^n$, let $|w_z\rangle$ be a vector such that $\langle w_z | i \rangle = z_i / n^{3/2}$ for all $i \in [n]$, and also such that the projections of the $|w_z\rangle$'s onto the orthogonal complement of $|1\rangle, \ldots, |n\rangle$ are all orthogonal to one another. Let $\mathcal{V} = \{|1\rangle, \ldots, |n\rangle\} \cup \{|w_z\rangle\}_{z \in \{-1,1\}^n}$. Then the inner product between any two distinct vectors in $\mathcal{V}$ is upper-bounded by $\varepsilon = 1/n^{3/2}$ in absolute value (the inner product between any two $|w_z\rangle$'s is at most $n/(n^{3/2})^2 = 1/n^2$). On the other hand, here is an algorithm that solves Gaussian Distinguishing using only $O(n) \ll 1/\varepsilon$ queries: first query $|1\rangle, \ldots, |n\rangle$ to obtain $a_1, \ldots, a_n$. Let $|\varphi\rangle := a_1 |1\rangle + \cdots + a_n |n\rangle$. Next, find n distinct vectors $|w_z\rangle$ that each have inner product $\Theta\left(n/n^{3/2}\right) = \Theta(1/\sqrt{n})$ with $|\varphi\rangle$ (such vectors can always be found, so long as $|a_i| = \Omega(1)$ for $\Omega(n)$ values of i), and query all of them, letting $b_1, \ldots, b_n$ be the results. In case (i), we have $\mathrm{E}[b_1 + \cdots + b_n]$ and $\mathrm{Var}[b_1 + \cdots + b_n] = n$. But in case (ii), we have $\mathrm{E}[b_1 + \cdots + b_n] = \Theta(\sqrt{n})$ and $\mathrm{Var}[b_1 + \cdots + b_n] = O(n)$, allowing the two cases to be distinguished with constant bias.

[8] In fact one *can* prove a $\widetilde{\Omega}(N^{1/3})$ lower bound even under this restriction—and more generally, in the statement of Theorem 9, one can replace the lower bound $\Omega\left(\frac{1/\varepsilon}{\log(M/\varepsilon)}\right)$ by $\Omega\left(\frac{(1/\varepsilon)^{2/3}}{(\log 1/\varepsilon)^{1/3}}\right)$, independent of M. We show how to do this in the full version.

a multilinear polynomial, but to a *block-multilinear* polynomial. By this, we mean a degree-$2t$ polynomial q that takes as input $2t$ blocks of N variables each, and whose every monomial contains exactly one variable from each block. If we repeat the input $X \in \{-1,1\}^N$ across all $2t$ blocks, then $q(X,\ldots,X)$ represents the quantum algorithm's acceptance probability on X. However, the key point is that $q(Y)$ is bounded in $[-1,1]$ for *any* Boolean input $Y \in \{-1,1\}^{2tN}$. More formally:

LEMMA 10. *Let $\mathcal{A}$ be a quantum algorithm that makes t queries to a Boolean input $x \in \{-1,1\}^N$. Then there exists a degree-$2t$ block-multilinear polynomial $p : \mathbb{R}^{2tN} \rightarrow \mathbb{R}$, with $2t$ blocks of N variables each, such that*

(i) the probability that $\mathcal{A}$ accepts x equals $p(x,\ldots,x)$ (with x repeated $2t$ times), and

(ii) $p(z) \in [-1,1]$ for all $z \in \{-1,1\}^{2tN}$.

PROOF. Assume for simplicity (and without loss of generality) that $\mathcal{A}$ involves real amplitudes only. For all $j \in [t]$ and $i \in [N]$, let $x_{j,i}$ be the value of x_i that $\mathcal{A}$'s oracle returns in response to its j^{th} query. Of course, in any "normal" run of $\mathcal{A}$, we will have $x_{j,i} = x_{j',i}$ for all j, j': that is, the value of x_i will be consistent across all t queries. But it is perfectly legitimate to ask what happens if x changes from one query to the next. In any case, $\mathcal{A}$ will have some normalized final state, of the form

$$\sum_{i,w} \alpha_{i,w}(x_{1,1},\ldots,x_{t,N})\,|i,w\rangle.$$

Furthermore, following Beals et al. [3], it is easy to see that each amplitude $\alpha_{i,w}$ can be written as a degree-t block-multilinear polynomial in the tN variables $x_{1,1},\ldots,x_{t,N}$, with one block of N variables, $R_j = \{x_{j,1},\ldots,x_{j,N}\}$, corresponding to each of the t queries. Next, for all $j \in [t]$ and $i \in [N]$, we create a *second* variable $x_{t+j,i}$, which just like $x_{j,i}$, represents the value of x_i that $\mathcal{A}$'s oracle returns in response to its j^{th} query. Let Acc be the set of all accepting basis states, and consider the polynomial

$$p(x_{1,1},\ldots,x_{2t,N}) := \sum_{(i,w)\in \mathrm{Acc}} \alpha_{i,w}(x_{1,1},\ldots,x_{t,N})\,\alpha_{i,w}(x_{t+1,1},\ldots,x_{2t,N}).$$

By construction, p is a degree-$2t$ block-multilinear polynomial in the $2tN$ variables $x_{1,1},\ldots,x_{2t,N}$, with one block of N variables, $R_j = \{x_{j,1},\ldots,x_{j,N}\}$, for each $j \in [2t]$. Furthermore, if we repeat the same input $x \in \{-1,1\}^N$ across all $2t$ blocks, then

$$p(x,\ldots,x) = \sum_{(i,w)\in \mathrm{Acc}} \alpha_{i,w}^2(x,\ldots,x)$$

is simply the probability that $\mathcal{A}$ accepts x. Finally, even if $x_{1,1},\ldots,x_{2t,N} \in \{-1,1\}^{2tN}$ is completely arbitrary, p still represents an inner product between two vectors. Since both of these vectors have norm at most 1, their inner product is bounded in $[-1,1]$. □

This leads to a new complexity measure for Boolean functions f: the *block-multilinear approximate degree* $\widetilde{\mathrm{bmdeg}}(f)$, which lower-bounds the quantum query complexity $\mathrm{Q}(f)$ just as $\widetilde{\deg}(f)$ does, but which might provide a tighter lower bound in some cases.

Once we have our quantum algorithm's acceptance probability in the form of a block-multilinear polynomial p, the second step is to *preprocess* p, to make it easier to estimate using random sampling. The basic problem is that p might be highly "unbalanced": certain variables might be hugely influential. Such variables are essential to query, but examining the form of p does not make it obvious which variables these are. To deal with this, we repeatedly perform an operation called "variable-splitting," which consists of identifying an influential variable x_i, then replacing every occurrence of x_i in p by $\frac{1}{m}(x_{i,1} + \cdots + x_{i,m})$, where $x_{i,1},\ldots,x_{i,m}$ are newly-created variables set equal to x_i. Observe that variable-splitting preserves the property that p is bounded in $[-1,1]$ at all Boolean points—for, regardless of how we set $x_{j,l_1},\ldots,x_{j,l_m}$, the new value will simply equal the value of p with $x_{j,l}$ set to $\frac{x_{j,l_1}+\cdots+x_{j,l_m}}{m}$, which in turn is a convex combination of p with $x_{j,l}$ set to -1 and p with $x_{j,l}$ set to 1. The point of doing this is that each $x_{i,j}$ will be less influential in p than x_i itself was, thereby yielding a more balanced polynomial. We show that variable-splitting can achieve the desired balance by introducing at most $\exp(t)\cdot O(N)$ new variables, which is linear in N for constant t.

More formally, suppose

$$p(x_{1,1},\ldots,x_{k,N}) = \sum_{i_1,\ldots,i_k\in[N]} a_{i_1,\ldots,i_k} x_{1,i_1}\cdots x_{k,i_k}$$

is a bounded block-multilinear polynomial of degree k. Set $\delta := \varepsilon^2/N$. Then by repeatedly splitting variables, we wish to achieve the following requirement: for every nonempty set $S \subseteq [k]$,

$$\Lambda_S := \sum_{(i_j)_{j\in S}} \left(\sum_{(i_j)_{j\notin S}} a_{i_1,\ldots,i_k} \right)^2 \leq \delta. \qquad (1)$$

Our key lemma is the following.

LEMMA 11. *Let $S \subseteq [k]$ be nonempty. Then there is a sequence of variable-splittings that introduces at most $1/\delta$ new variables, and that produces a polynomial p' that satisfies $\Lambda_S \leq \delta$.*

PROOF. We start with the case $S = [k]$. Then we have to ensure

$$\sum_{i_1,\ldots,i_k\in[N]} a_{i_1,\ldots,i_k}^2 \leq \delta, \qquad (2)$$

where $a_{i_1,\ldots,i_k}^2$ is the coefficient of $x_{1,i_1}\ldots x_{k,i_k}$. Let

$$V_i := \sum_{i_2,\ldots,i_k\in[N]} a_{i_1,i_2,\ldots,i_k}^2.$$

We now randomly set each x_{j,i_j} for $j \geq 2$, to be 1 or -1 with independent probability $1/2$. Let

$$X_i := \sum_{i_2,\ldots,i_k\in[N]} a_{i_1,i_2\ldots,i_k} x_{2,i_2}\cdots x_{k,i_k}.$$

Then $\mathrm{E}[X_i^2] = V_i$. By the concavity of the square root function, this means $\mathrm{E}[|X_i|] \geq \sqrt{V_i}$. Hence

$$E[|X_1| + \cdots + |X_N|] \geq \sqrt{V_1} + \cdots + \sqrt{V_N}.$$

If we set $x_{1,i} = 1$ whenever $X_i \geq 0$ and $x_{1,i} = -1$ otherwise, we get

$$p(x_{1,1}, \ldots, x_{k,N}) = \sum_{i=1}^{N} x_{1,i} X_i = \sum_{i=1}^{N} |X_i| .$$

Since $p(x_{1,1}, \ldots, x_{k,N})$ is bounded in $[-1, 1]$ at all Boolean points, this means that

$$\sqrt{V_1} + \cdots + \sqrt{V_N} \leq 1.$$

We now perform a sequence of variable-splittings. For each $i \in [N]$, let $m_i := \lfloor \sqrt{V_i}/\delta \rfloor$, so that

$$\delta m_i \leq \sqrt{V_i} < \delta (m_i + 1) .$$

Then we split $x_{1,i}$ into $m_i + 1$ variables. This replaces each term $a_{i_1,\ldots,i_k} x_{1,i_1} \cdots x_{k,i_k}$ with $m_i + 1$ terms that each equal $\frac{1}{m_i+1} a_{i_1,\ldots,i_k} x_{1,i_1} \cdots x_{k,i_k}$. Therefore, this variable-splitting reduces V_i to $V_i/(m_i + 1)$, and decreases the sum (2) by $\frac{m_i}{m_i+1} V_i$.

After we have performed such variable-splittings for each i, the sum (2) becomes

$$\begin{aligned} \sum_{i=1}^{N} \frac{V_i}{m_i + 1} &\leq \sum_{i=1}^{N} \frac{V_i}{\sqrt{V_i}/\delta} \\ &= \delta \left(\sqrt{V_1} + \cdots + \sqrt{V_N} \right) \\ &\leq \delta. \end{aligned}$$

The number of new variables that get introduced equals

$$\sum_{i=1}^{N} m_i \leq \sum_{i=1}^{N} \frac{\sqrt{V_i}}{\delta} \leq \frac{1}{\delta}.$$

The case $S \subset [k]$ reduces to the case $S = [k]$ in the following way. For typographical convenience, assume that $S = [\ell]$ for some ℓ. Then substituting $x_{i,j} = 1$ for $i > \ell$ transforms the polynomial $p(x_{1,1}, \ldots, x_{k,N})$ into the polynomial

$$p'(x_{1,1}, \ldots, x_{\ell,N}) = \sum_{i_1,\ldots,i_\ell \in [N]} \bar{a}_{i_1,\ldots,i_\ell} x_{1,i_1} \cdots x_{\ell,i_\ell}$$

where

$$\bar{a}_{i_1,\ldots,i_\ell} := \sum_{i_{\ell+1},\ldots,i_k \in [N]} a_{i_1,\ldots,i_k}.$$

The statement of Lemma 11 now becomes

$$\sum_{i_1,\ldots,i_\ell \in [N]} \bar{a}^2_{i_1,\ldots,i_\ell} \leq \delta$$

which can be achieved similarly to the previous case. □

Lemma 11 has the following consequence.

COROLLARY 12. *There is a sequence of variable-splittings that introduces at most $2^k/\delta$ new variables, and that produces a polynomial p' that satisfies $\Lambda_S \leq \delta$ for every nonempty subset $S \subseteq [k]$.*

PROOF. We simply apply the procedure of Lemma 11 once for each nonempty $S \subseteq [k]$, in any order. Since there are $2^k - 1$ possible choices for S, and since each iteration adds at most $1/\delta$ variables, the total number of added variables is at most $2^k/\delta$. Furthermore, we claim that later iterations can never "undo" the effects of previous iterations. This is because, if we consider how the quantity

$$\Lambda_S = \sum_{(i_j)_{j \in S}} \left(\sum_{(i_j)_{j \notin S}} a_{i_1,\ldots,i_k} \right)^2$$

is affected by variable-splittings applied to the variables in the j^{th} block, there are only two possibilities: if $j \in S$ then Λ_S can decrease, while if $j \notin S$ then Λ_S remains unchanged. □

We then apply Corollary 12 with the choice $\delta = \varepsilon^2/N$. This introduces at most $2^k N/\varepsilon^2 = O\left(N/\varepsilon^2\right)$ new variables, and achieves $\Lambda_S \leq \varepsilon^2/N$ for every S.

From now on, we use n to denote the "new" number of variables per block, which is a constant factor greater than the "old" number N.

Once we have a balanced polynomial q, the last step is to give a query-efficient randomized algorithm to estimate its value. Our algorithm is the simplest one imaginable: we simply choose $O(n^{1-1/2t})$ variables uniformly at random, query them, then form an estimate $\tilde{q}$ of q by summing only those monomials all of whose variables were queried. In more detail, let

$$b_{i_1,\ldots,i_k} := a_{i_1,\ldots,i_k} x_{1,i_1} \cdots x_{k,i_k}.$$

Then

$$q(x_{1,1}, \ldots, x_{k,n}) = \sum_{i_1,\ldots,i_k} b_{i_1,\ldots,i_k}.$$

We can estimate this sum in the following way. For each i, j_i independently, let y_{i,j_i} be a $\{0,1\}$-valued random variable with $\Pr[y_{i,j_i} = 1] = \frac{1}{n^{1/k}}$. We then take

$$P := b_{i_1,\ldots,i_k} y_{1,i_1} \cdots y_{k,i_k}$$

as our estimator.

Clearly, this is an unbiased estimator of $q(x_{1,1}, \ldots, x_{k,n})$, with expectation

$$\mathrm{E}[P] = \frac{q(x_1, \ldots, x_n)}{n}.$$

The result we need to prove is that $\mathrm{Var}[P] = O(\delta/n)$. If this is true, then performing $O(1)$ repetitions of P allows us to estimate $q(x_{1,1}, \ldots, x_{k,n})$ with precision $\sqrt{\delta n} = \sqrt{(\varepsilon^2/n) \cdot n} = \varepsilon$. This estimation can be carried out with $O(n^{1-1/k})$ queries because, to calculate P, we only need the values of $x_{i,j}$ with $y_{i,j} = 1$, and the number of such variables is $O(n^{1-1/k})$, with a very high probability.

The hard part, done in the full version, is of course to show that the variance is bounded. The proof of this makes heavy use of the balancedness property that was ensured by the preprocessing step.

Examining our estimation algorithm, an obvious question is whether it was essential that q be block-multilinear, or whether the algorithm could be extended to *all* bounded low-degree polynomials. In the full version, we take a first step toward answering that question, by giving an $O(\sqrt{N})$-query randomized algorithm to estimate any bounded degree-2 polynomial in N Boolean variables. Once we drop block-multilinearity, our variable-splitting procedure no longer works, so we rely instead on Fourier-analytic results of Dinur et al. [9] to identify influential variables which we then split.

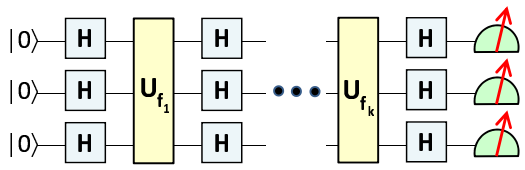

Figure 1: A quantum circuit that can be taken to define the k-fold Forrelation problem. The circuit consists of k query transformations $U_{f_1},\ldots,U_{f_k}$, which map each basis state $|x\rangle$ to $f_i(x)|x\rangle$, sandwiched between rounds of Hadamard gates.

4.3 k-fold Forrelation

The Forrelation and k-fold Forrelation problems were defined in Sections 2.1 and 2.2.1 respectively. Informally, though, one could define k-fold Forrelation simply as the problem of simulating the quantum circuit shown in Figure 1—and in particular, of estimating the amplitude, call it $\alpha_{0\cdots0}$, with which this circuit returns $|0\rangle^{\otimes n}$ as its output. Observe that $\alpha_{0\cdots0}$ is *precisely* the quantity $\Phi_{f_1,\ldots,f_k}$ defined in Section 2.2.1. From this, it follows that we can decide whether $|\Phi_{f_1,\ldots,f_k}| \leq \frac{1}{100}$ or $\Phi_{f_1,\ldots,f_k} \geq \frac{3}{5}$ with bounded probability of error, and thereby solve the k-fold Forrelation problem, by making only k quantum queries to $f_1,\ldots,f_k$.

Slightly more interesting is that we can improve the quantum query complexity further, to $\lceil k/2 \rceil$:

Proposition 13. *The k-fold Forrelation problem is solvable, with error probability 0.4, using $\lceil k/2 \rceil$ quantum queries to the functions $f_1,\ldots,f_k : \{0,1\}^n \rightarrow \{-1,1\}$, as well as $O(nk)$ quantum gates.*

Proof. Let H be the Hadamard gate, and let U_{f_i} be the query transformation that maps each computational basis state $|x\rangle$ to $f_i(x)|x\rangle$. Then to improve from k to $\lceil k/2 \rceil$ queries, we modify the circuit of Figure 1 in the following way. In addition to the initial state $|0\rangle^{\otimes n}$, we prepare a control qubit in the state $|+\rangle = \frac{|0\rangle+|1\rangle}{\sqrt{2}}$. Then, conditioned on the control qubit being $|0\rangle$, we apply the following sequence of operations to the initial state:

$$\mathrm{H}^{\otimes n} \rightarrow U_{f_1} \rightarrow \mathrm{H}^{\otimes n} \rightarrow U_{f_2} \rightarrow \cdots \rightarrow \mathrm{H}^{\otimes n} \rightarrow U_{f_{\lceil k/2 \rceil}} \rightarrow \mathrm{H}^{\otimes n}.$$

Meanwhile, conditioned on the control qubit being $|1\rangle$, we apply the following sequence of operations:

$$\mathrm{H}^{\otimes n} \rightarrow U_{f_k} \rightarrow \mathrm{H}^{\otimes n} \rightarrow U_{f_{k-1}} \rightarrow \cdots \rightarrow \mathrm{H}^{\otimes n} \rightarrow U_{f_{\lceil k/2 \rceil+1}}.$$

Finally, we measure the control qubit in the $\{|+\rangle,|-\rangle\}$ basis, and "accept" (i.e., say that $\Phi_{f_1,\ldots,f_k}$ is large) if and only if we find it in the state $|+\rangle$. It is not hard to see that the probability that this circuit accepts is exactly

$$\frac{1+\Phi_{f_1,\ldots,f_k}}{2}.$$

Thus, consider an algorithm $\mathcal{A}$ that rejects with probability 1/4, and runs the circuit with probability 3/4. We have

$$\Pr[\mathcal{A} \text{ accepts}] = \frac{3}{4}\left(\frac{1+\Phi_{f_1,\ldots,f_k}}{2}\right).$$

If $|\Phi_{f_1,\ldots,f_k}| \leq \frac{1}{100}$ then the above is less than 0.4, while if $\Phi_{f_1,\ldots,f_k} \geq \frac{3}{5}$ then it is at least 0.6. □

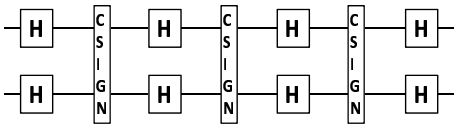

Figure 2: A 2-qubit gadget for converting an even number of layers of Hadamard gates into an odd number.

Purely from the unitarity of the quantum algorithm to compute $\Phi_{f_1,\ldots,f_k}$, we can derive some interesting facts about $\Phi_{f_1,\ldots,f_k}$ itself: for example, that $|\Phi_{f_1,\ldots,f_k}| \leq 1$.

4.4 Other Results

BQP-Completeness. The proof that the k-fold Forrelation problem is PromiseBQP-complete is simple, once one has the main idea. The sum that defines k-fold Forrelation is, itself, an output amplitude for a particular kind of quantum circuit, which consists entirely of Hadamard and f-phase gates (i.e., gates that map $|x\rangle$ to $(-1)^{f(x)}|x\rangle$ for some Boolean function f). Since the Hadamard and CCPHASE gates (corresponding to $f(x,y,z)=xyz$) are known to be universal for quantum computation, one might think that our work is done. The difficulty is that the quantum circuit for k-fold Forrelation contains a Hadamard gate on every qubit, between every pair of f-phase gates, *whether we wanted Hadamards there or not.* Thus, if we want to encode an arbitrary quantum circuit, then we need some way of *canceling* unwanted Hadamards, while leaving the wanted ones. We achieve this via a gadget construction.

In more detail, we need a gadget that lets us Hadamard some desired *subset* of the qubits, $S \subset [n]$, and not the qubits outside S. For simplicity, suppose that $|S|=2$, and let a and b be S's elements. Our gadget, shown in Figure 2, consists of three CSIGN gates (i.e., gates that map $|x,y\rangle$ to $(-1)^{xy}|x,y\rangle$) on a and b, sandwiched between Hadamard gates. Note that we can implement a CSIGN on a and b as U_{f_i}, where $f_i(z_1,\ldots,z_n) := (-1)^{z_a z_b}$. Meanwhile, the Hadamard gates are just those that are automatically applied between each U_{f_i} and $U_{f_{i+1}}$ in a quantum circuit for Forrelation. To see why the gadget works, consider the following identity:

$$\left(\frac{1}{2}\begin{pmatrix}1&1&1&1\\1&-1&1&-1\\1&1&-1&-1\\1&-1&-1&1\end{pmatrix}\begin{pmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&-1\end{pmatrix}\right)^3 = \begin{pmatrix}1&0&0&0\\0&0&1&0\\0&1&0&0\\0&0&0&1\end{pmatrix}.$$

In particular, if we let C stand for CSIGN, $\mathrm{H}^{\otimes 2}$ for Hadamards on two qubits, and S for the 2-qubit SWAP gate, then

$$\mathrm{H}^{\otimes 2}\,\mathrm{C}\,\mathrm{H}^{\otimes 2}\,\mathrm{C}\,\mathrm{H}^{\otimes 2}\,\mathrm{C}\,\mathrm{H}^{\otimes 2} = \mathrm{S}\,\mathrm{H}^{\otimes 2}.$$

Contrast this with what happens if we apply the 2-qubit identity, I, rather than C, in the inner layers:

$$\mathrm{H}^{\otimes 2}\,\mathrm{I}\,\mathrm{H}^{\otimes 2}\,\mathrm{I}\,\mathrm{H}^{\otimes 2}\,\mathrm{I}\,\mathrm{H}^{\otimes 2} = \mathrm{I}.$$

Thus, Hadamards get applied if C is chosen for the inner layers, but *not* if I is chosen. So this gadget has the ef-

fect of Hadamarding a and b, while not Hadamarding the other qubits in the circuit. Now, the gadget also has the unintended side effect of swapping a and b. But since we know this is going to happen, we can keep track of it by simply swapping the *labels* of a and b whenever the gadget is applied.

Indeed, examining the construction, we can derive a stronger consequence. Define a *depth-d quantum circuit* as one where the gates are organized into d sequential layers, with the gates within each layer all commuting with one another. Now, given a depth-d quantum circuit Q over the basis $\{\mathrm{H}, \mathrm{CCSIGN}\}$ (where H is Hadamard), let QSIM_d be the problem of deciding whether the circuit's accepting amplitude A satisfies $A \geq \frac{1}{4}$ or $|A| \leq \frac{1}{100}$, promised that one of those is the case. Then in the full version, we prove the following:

THEOREM 14. QSIM_d *is polynomial-time reducible to explicit* $(2d+1)$*-fold* FORRELATION. *(Moreover, the functions* $f_1, \ldots, f_{2d+1}$ *produced by the reduction all have the form* $f_i(x) = (-1)^{p(x)}$*, where* p *is a degree-*3 *polynomial in the input bits.)*

So for example, we find that explicit $\log n$-fold FORRELATION is a complete promise problem for $\mathsf{PromiseBQNC}^1$: the class of problems that captures what can be done using log-depth quantum circuits.

Separation for Sampling Problems. To achieve a 1 versus $\Omega(N/\log N)$ quantum/classical query complexity separation for a sampling problem, we consider FOURIER SAMPLING: the problem, given oracle access to a Boolean function $f : \{0,1\}^n \rightarrow \{-1,1\}$, of outputting a string $y \in \{0,1\}^n$ with probability approximately equal to $\hat{f}(y)^2$. This problem is trivially solvable with 1 quantum query, but proving a $\Omega(N/\log N)$ classical lower bound takes a few pages of work. The basic idea is to concentrate on the probability of a *single* string—say, $y = 0^n$—being output. Using a binomial calculation, we show that this probability cannot depend on f's truth table in the appropriate way unless $\Omega(N/\log N)$ function values are queried.

Lower Bound for k-Fold FORRELATION. Once we have a $\Omega(\frac{\sqrt{N}}{\log N})$ randomized lower bound for FORRELATION, one might think it would be trivial to prove the same lower bound for k-fold FORRELATION: just reduce one to the other! However, FORRELATION does not embed in any clear way as a subproblem of k-fold FORRELATION. On the other hand, given an instance of k-fold FORRELATION, suppose we "give away for free" the complete truth tables of all but two of the functions. In that case, we show that the induced subproblem on the remaining two functions is an instance of GAUSSIAN DISTINGUISHING to which, with high probability, our lower bound techniques can be applied. Pursuing this idea leads to our $\Omega(\frac{\sqrt{N}}{\log^{7/2} N})$ lower bound on the randomized query complexity of k-fold FORRELATION, for all $k \geq 2$.

Property-Testing Separation. To turn our quantum versus classical separation for the FORRELATION problem into a property-testing separation, we need to prove two interesting statements. The first is that function pairs $\langle f, g \rangle$ that are far in Hamming distance from the set of all pairs with low forrelation, actually have high forrelation. The second is that "generic" function pairs $\langle f, g \rangle$ and $\langle f', g' \rangle$ that have small Hamming distance from one another, are close in their forrelation values as well. In fact, in the full version, we prove both of these statements for the general case of k-fold FORRELATION.

5. DISCUSSION

To summarize, this paper proves the largest separation between classical and quantum query complexities yet known, and it also proves that that separation is in some sense optimal. These results put us in a position to pose an intriguing open question:

> Among all the problems that admit a superpolynomial quantum speedup, is there any whose classical randomized query complexity is $\gg \sqrt{N}$?

Strikingly, if we look at the known problems with superpolynomial quantum speedups, for every one of them the classical randomized lower bound seems to hit a "ceiling" at $\sqrt{N}$. Thus, SIMON'S PROBLEM has quantum query complexity $O(\log N)$ and randomized query complexity $\widetilde{\Theta}(\sqrt{N})$; the GLUED-TREES problem of Childs et al. [7] has quantum query complexity $\log^{O(1)}(N)$ and randomized query complexity $\widetilde{\Theta}(\sqrt{N})$;[9] and FORRELATION has quantum query complexity 1 and randomized query complexity $\widetilde{\Theta}(\sqrt{N})$.

If we insist on making the randomized query complexity $\Omega(N^{1/2+c})$, for some $c > 0$, and then try to minimize the quantum query complexity, then the best thing we know how to do is to take the OR of N^{2c} independent instances of FORRELATION, each of size N^{1-2c}. This gives us a problem whose quantum query complexity is $\Theta(N^c)$,[10] and whose classical randomized query complexity is $\widetilde{\Theta}(N^{1/2+c})$.[11] Of course, this is not an exponential separation.

In this paper, we gave a candidate for a problem that breaks the "$\sqrt{N}$ barrier": namely, k-fold FORRELATION. Indeed, we conjecture that k-fold FORRELATION achieves the

[9] The randomized lower bound for GLUED-TREES proved by Childs et al. [7] was only $\Omega(N^{1/6})$. However, Fenner and Zhang [11] improved the lower bound to $\Omega(N^{1/3})$; and if we allow a success probability that is merely (say) 1/3, rather than exponentially small, then their bound can be improved further, to $\Omega(\sqrt{N})$. In the other direction, we are indebted to Shalev Ben-David for proving that GLUED-TREES can be solved deterministically using only $O(\sqrt{N}\log N)$ queries (or $O(\sqrt{N}\log^2 N)$, if the queries are required to be Boolean). For his proof, see http://cstheory.stackexchange.com/questions/25279/the-randomized-query-complexity-of-the-conjoined-trees-problem

[10] Here the upper bound comes from combining Grover's algorithm with the FORRELATION algorithm: the "naïve" way of doing this would produce an additional $\log N$ factor for error reduction, but it is well-known that that log factor can be eliminated [13]. Meanwhile, the lower bound comes from the optimality of Grover's algorithm.

[11] Here the upper bound comes from simply taking the best randomized FORRELATION algorithm, which uses $O(\sqrt{N^{1-2c}})$ queries, and running it N^{2c} times, with an additional $\log N$ factor for error reduction. Meanwhile, the lower bound comes from combining this paper's $\Omega(\sqrt{N}/\log N)$ lower bound for FORRELATION, with a general result stating that the randomized query complexity of $\mathrm{OR}(f, \ldots, f)$, the OR of k disjoint copies of a function f, is $\Omega(k)$ times the query complexity of a single copy. This result can be proved by adapting ideas from a direct product theorem for randomized query complexity given by Drucker [10] (we thank A. Drucker, personal communication).

optimal separation for all $k = O(1)$, requiring $\widetilde{\Omega}(N^{1-1/k})$ classical randomized queries but only $\lceil k/2 \rceil$ quantum queries.[12] Proving this conjecture is an enticing problem. Unfortunately, k-fold FORRELATION becomes extremely hard to analyze when $k > 2$, because we can no longer view the functions $f_1, \ldots, f_k$ as confined to a low-dimensional subspace: now we have to view them as confined to a low-dimensional *manifold*, which is defined by degree-$(k-1)$ polynomials. As such, we can no longer compute posterior probabilities by simply appealing to the rotational invariance of the Gaussian measure, which made our lives easier in the $k = 2$ case. Instead we need to calculate integrals over a nonlinear manifold.

Short of proving our conjecture about k-fold FORRELATION, it would of course be nice to find *any* partial Boolean function whose quantum query complexity is polylog N, and whose randomized query complexity is $N^{1/2+\Omega(1)}$.

Another problem we leave is to generalize our $O(N^{1-1/k})$ randomized estimation algorithm from block-multilinear polynomials to *arbitrary* bounded polynomials of degree k. As we said, in the full version we achieve this in the special case $k = 2$. Achieving it for arbitrary k seems likely to require generalizing the machinery of Dinur et al. [9].

A third problem concerns the notion of block-multilinear approximate degree, $\widetilde{\text{bmdeg}}(f)$, that we introduced to prove Theorem 4. Is there any asymptotic separation between $\widetilde{\text{bmdeg}}(f)$ and ordinary approximate degree? What about a separation between $\widetilde{\text{bmdeg}}(f)$ and quantum query complexity, as Ambainis [2] showed between $\widetilde{\deg}(f)$ and quantum query complexity?

A fourth, more open-ended problem is whether there are any applications of FORRELATION, in the same sense that factoring and discrete log provide "applications" of Shor's period-finding problem. Concretely, are there any situations where one has two efficiently-computable Boolean functions $f, g : \{0,1\}^n \to \{-1,1\}$ (described, for example, by circuits), one wants to estimate how forrelated they are, *and* the structure of f and g does not provide a fast classical way to do this?

Here are five other open problems:

First, can we tighten the lower bound on the randomized query complexity of FORRELATION from $\Omega(\frac{\sqrt{N}}{\log N})$ to $\Omega(\sqrt{N})$, or give an $O(\frac{\sqrt{N}}{\log N})$ upper bound?

Second, can we generalize our results from Boolean to non-Boolean functions?

Third, what are the largest possible quantum versus classical query complexity separations for *sampling* problems? Is an $O(1)$ versus $\Omega(N)$ separation possible in this case? Also, what separations are possible for search or relation problems? (For our results on these questions, see the full version.)

Fourth, while there exists a 1-query quantum algorithm that solves FORRELATION with bounded error probability, the error probability we are able to achieve is about 0.4—more than the customary 1/3. If we want (say) a 1 versus $N^{\Omega(1)}$ quantum versus classical query complexity separation, then how small can the quantum algorithm's error be?

Fifth, in the full version we show that being "*un*forrelated"—that is, having $\Phi_{f,g} \leq \frac{1}{100}$—behaves nicely as a property-testing problem. But it would be interesting to show the same for being forrelated.

[12] And perhaps k-fold FORRELATION continues to give optimal separations, all the way up to $k = O(\log N)$.

6. ACKNOWLEDGMENTS

We are grateful to Ronald de Wolf for early discussions, Andy Drucker for discussions about the randomized query complexity of OR$(f, \ldots, f)$, Shalev Ben-David and Sean Hallgren for discussions about the glued-trees problem, Saeed Mehraban for a numerical calculation, and Man-Hong Yung for catching a calculation error in an earlier draft.

7. REFERENCES

[1] S. Aaronson. BQP and the polynomial hierarchy. In *Proc. STOC*, 2010. arXiv:0910.4698.

[2] A. Ambainis. Polynomial degree vs. quantum query complexity. *J. Comput. Sys. Sci.*, 72(2):220–238, 2006. Earlier version in FOCS 2003. quant-ph/0305028.

[3] 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 FOCS 1998, pp. 352-361. quant-ph/9802049.

[4] J. N. de Beaudrap, R. Cleve, and J. Watrous. Sharp quantum versus classical query complexity separations. *Algorithmica*, 34(4):449–461, 2002. quant-ph/0011065.

[5] H. Buhrman, L. Fortnow, I. Newman, and H. Röhrig. Quantum property testing. *SIAM J. Comput.*, 37(5):1387–1400, 2008. Earlier version in SODA 2003. quant-ph/0201117.

[6] S. Chakraborty, E. Fischer, A. Matsliah, and R. de Wolf. New results on quantum property testing. In *Proc. FSTTCS*, pages 145–156, 2010. arXiv:1005.0523.

[7] A. M. Childs, R. Cleve, E. Deotto, E. Farhi, S. Gutmann, and D. A. Spielman. Exponential algorithmic speedup by a quantum walk. In *Proc. STOC*, pages 59–68, 2003. quant-ph/0209131.

[8] R. Cleve. The query complexity of order-finding. *Inf. Comput.*, 192(2):162–171, 2004. Earlier version in CCC'2000. quant-ph/9911124.

[9] I. Dinur, E. Friedgut, G. Kindler, and R. O'Donnell. On the Fourier tails of bounded functions over the discrete cube. In *Proc. STOC*, pages 437–446, 2006.

[10] A. Drucker. Improved direct product theorems for randomized query complexity. *Computational Complexity*, 21(2):197–244, 2012. Earlier version in CCC'2011. ECCC TR10-080.

[11] S. Fenner and Y. Zhang. A note on the classical lower bound for a quantum walk algorithm. quant-ph/0312230v1, 2003.

[12] A. Montanaro and R. de Wolf. A survey of quantum property testing. arXiv:1310.2035, 2013.

[13] B. Reichardt. Reflections for quantum query algorithms. In *Proc. SODA*, pages 560–569, 2011. arXiv:1005.1601.

[14] 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 FOCS 1994. quant-ph/9508027.

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