THEORY OF COMPUTING, Volume 10 (6), 2014, pp. 133–166
www.theoryofcomputing.org

The Need for Structure in Quantum Speedups

Scott Aaronson* Andris Ambainis†

Received October 22, 2012; Revised July 10, 2014; Published August 12, 2014

Abstract: Is there a general theorem that tells us when we can hope for exponential speedups from quantum algorithms, and when we cannot? In this paper, we make two advances toward such a theorem, in the black-box model where most quantum algorithms operate.

First, we show that for any problem that is invariant under permuting inputs and outputs and that has sufficiently many outputs (like the collision and element distinctness problems), the quantum query complexity is at least the 7^{th} root of the classical randomized query complexity. (An earlier version of this paper (ICS 2011) gave the 9^{th} root.) This resolves a conjecture of Watrous from 2002.

Second, inspired by work of O'Donnell et al. (2005) and Dinur et al. (2006), we conjecture that every bounded low-degree polynomial has a "highly influential" variable. (A multivariate polynomial p is said to be *bounded* if $0 \leq p(x) \leq 1$ for all x in the Boolean cube.) Assuming this conjecture, we show that every T-query quantum algorithm can be simulated on *most* inputs by a $T^{O(1)}$-query classical algorithm, and that one essentially cannot hope to prove $\mathsf{P} \neq \mathsf{BQP}$ relative to a random oracle.

ACM Classification: F.1.2, F.1.3

AMS Classification: 81P68, 68Q12, 68Q17

Key words and phrases: decision trees, adversary method, collision problem, Fourier analysis, influences, quantum computing, query complexity

A preliminary version of this paper appeared in the Proc. 2nd "Innovations in Computer Science" Conference (ICS 2011). See Sec. 1.3 for a comparison with the present paper.

*MIT. Email: aaronson@csail.mit.edu. This material is based upon work supported by the National Science Foundation under Grant No. 0844626, by a TIBCO Career Development Chair, and by an Alan T. Waterman award.

†University of Latvia and IAS, Princeton. Email: ambainis@lu.lv. Supported by University of Latvia Research Grant ZP01-100, FP7 Marie Curie International Reintegration Grant (PIRG02-GA-2007-224886), FP7 FET-Open project QCS, FP7 FET-Proactive project QALGO and ERC Advanced Grant MQC (at the University of Latvia) and the National Science Foundation under agreement No. DMS-1128155 (at IAS, Princeton). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

© 2014 Scott Aaronson and Andris Ambainis
Licensed under a Creative Commons Attribution License (CC-BY)
DOI: 10.4086/toc.2014.v010a006

1 Introduction

Perhaps the central lesson gleaned from fifteen years of quantum algorithms research is this:

> *Quantum computers can offer superpolynomial speedups over classical computers, but only for certain "structured" problems.*

The key question, of course, is what we mean by "structured." In the context of most existing quantum algorithms, "structured" basically means that we are trying to determine some global property of an extremely long sequence of numbers, *assuming* that the sequence satisfies some global regularity. As a canonical example, consider PERIOD-FINDING, the core of Shor's algorithms for factoring and computing discrete logarithms [30]. Here we are given black-box access to an exponentially-long sequence of integers $X = (x_1, \ldots, x_N)$; that is, we can compute x_i for a given i. We are asked to find the *period* of X—that is, the smallest $k > 0$ such that $x_i = x_{i-k}$ for all $i > k$—under the promise that X is indeed periodic, with period $k \ll N$ (and also that the x_i values are approximately distinct within each period). The requirement of periodicity is crucial here: it is what lets us use the Quantum Fourier Transform to extract the information we want from a superposition of the form

$$\frac{1}{\sqrt{N}} \sum_{i=1}^{N} |i\rangle \, |x_i\rangle \; .$$

For other known quantum algorithms, X needs to be (for example) a cyclic shift of quadratic residues [17], or constant on the cosets of a hidden subgroup.

By contrast, the canonical example of an "unstructured" problem is the Grover search problem. Here we are given black-box access to an N-bit string $(x_1, \ldots, x_N) \in \{0,1\}^N$, and are asked whether there exists[1] an i such that $x_i = 1$. Grover [21] gave a quantum algorithm to solve this problem using $O(\sqrt{N})$ queries [21], as compared to the $\Omega(N)$ needed classically. However, this quadratic speedup is optimal, as shown by Bennett, Bernstein, Brassard, and Vazirani [11]. For other "unstructured" problems—such as computing the PARITY or MAJORITY of an N-bit string—quantum computers offer no asymptotic speedup at all over classical computers (see Beals et al. [9]).

Unfortunately, this "need for structure" has essentially limited the prospects for superpolynomial quantum speedups to those areas of mathematics that are liable to produce things like periodic sequences or sequences of quadratic residues.[2] *This is the fundamental reason why the greatest successes of quantum algorithms research have been in cryptography, and specifically in number-theoretic cryptography.* It helps to explain why we do not have a fast quantum algorithm to solve NP-complete problems (for example), or to break arbitrary one-way functions.

Given this history, the following problem takes on considerable importance:

Problem 1.1 (Informal)**.** For every "unstructured" problem f, are the quantum query complexity $\mathrm{Q}(f)$ and the classical randomized query complexity $\mathrm{R}(f)$ polynomially related?

[1] A variant asks us to *find* an i such that $x_i = 1$, under the mild promise that such an i exists.

[2] Here we exclude BQP-complete problems, such as simulating quantum physics (the "original" application of quantum computers), approximating the Jones polynomial [5], and estimating a linear functional of the solution of a well-conditioned linear system [22].

Despite its apparent vagueness, Problem 1.1 can be formalized in several natural and convincing ways—and under these formalizations, the problem has remained open for about a decade.

1.1 Formalizing the problem

Let $S \subseteq [M]^N$ be a collection of inputs, and let $f : S \to \{0,1\}$ be a function that we are trying to compute. In this paper, we assume for simplicity that the range of f is $\{0,1\}$; in other words, that we are trying to solve a decision problem. It will also be convenient to think of f as a function from $[M]^N$ to $\{0,1,*\}$, where $*$ means 'undefined' (that is, that a given input $X \in [M]^N$ is not in f's domain S).

We will work in the well-studied *decision-tree model*. In this model, given an input $X = (x_1, \ldots, x_N)$, an algorithm can at any time choose an i and receive x_i. We count only the number of queries the algorithm makes to the x_i, ignoring other computational steps. Then the *deterministic query complexity* of f, or $\mathrm{D}(f)$, is the number of queries made by an optimal deterministic algorithm on a worst-case input $X \in S$. The (bounded-error) *randomized query complexity* $\mathrm{R}(f)$ is the expected number of queries made by an optimal randomized algorithm that, for every $X \in S$, computes $f(X)$ with probability at least $2/3$. The (bounded-error) *quantum query complexity* $\mathrm{Q}(f)$ is the same as $\mathrm{R}(f)$, except that we allow quantum algorithms. Clearly $\mathrm{Q}(f) \leq \mathrm{R}(f) \leq \mathrm{D}(f) \leq N$ for all f. See Buhrman and de Wolf [15] for detailed definitions as well as a survey of these measures.

If $S = [M]^N$, then we say f is *total*, and if $M = 2$, then we say f is *Boolean*. The case of total f is relatively well-understood. Already in 1998, Beals et al. [9] showed the following:

Theorem 1.2 (Beals et al.)**.** $\mathrm{D}(f) = O(\mathrm{Q}(f)^6)$ *for all total Boolean functions* $f : \{0,1\}^N \to \{0,1\}$.

Furthermore, it is easy to generalize Theorem 1.2 to show that $\mathrm{D}(f) = O(\mathrm{Q}(f)^6)$ for *all* total functions $f : [M]^N \to \{0,1\}$, not necessarily Boolean.[3] In other words, for total functions, the quantum query complexity is always at least the 6^{th} root of the classical query complexity. The largest known gap between $\mathrm{D}(f)$ and $\mathrm{Q}(f)$ for a total function is quadratic, and is achieved by the OR function (because of Grover's algorithm).

On the other hand, as soon as we allow non-total functions, we can get enormous gaps. Aaronson [2] gave a Boolean function $f : S \to \{0,1\}$ for which $\mathrm{R}(f) = N^{\Omega(1)}$, yet $\mathrm{Q}(f) = O(1)$.[4] Other examples, for which $\mathrm{R}(f) = \Omega(\sqrt{N})$ and $\mathrm{Q}(f) = O(\log N \log\log N)$, follow easily from Simon's algorithm [31] and Shor's algorithm [30]. Intuitively, these functions f achieve such large separations by being highly structured: that is, their domain S includes only inputs that satisfy a stringent promise, such as encoding a periodic function, or (in the case of [2]) encoding two Boolean functions, one of which is correlated with the Fourier transform of the other one.

By contrast with these highly-structured problems, consider the *collision problem*: that of deciding whether a sequence of numbers $(x_1, \ldots, x_N) \in [M]^N$ is one-to-one (each number appears once) or two-to-one (each number appears twice). Let $\textsc{Col}(X) = 0$ if X is one-to-one and $\textsc{Col}(X) = 1$ if X is two-to-one,

[3] Theorem 1.2 is proved by combining three ingredients: $\mathrm{D}(f) = O(\mathrm{C}(f)\,\mathrm{bs}(f))$, $\mathrm{C}(f) = O(\mathrm{bs}(f)^2)$, and $\mathrm{bs}(f) = O(\mathrm{Q}(f)^2)$ (where $\mathrm{C}(f)$ is the *certificate complexity* of f and $\mathrm{bs}(f)$ is the *block sensitivity*). And all three ingredients go through with no essential change if we set $M > 2$, and define suitable M-ary generalizations of $\mathrm{C}(f)$ and $\mathrm{bs}(f)$. (We could also convert the non-Boolean function $f : [M]^N \to \{0,1\}$ to a Boolean one, but then we would lose a factor of $\log M$.)

[4] Previously, de Beaudrap, Cleve, and Watrous [10] had stated a similar randomized versus quantum separation. However, their separation applied not to the standard quantum black-box model, but to a different model in which the black box permutes the answer register $|y\rangle$ in some unknown way (rather than simply mapping $|y\rangle$ to $|y \oplus f(x)\rangle$).

under the promise that one of these is the case. Then $\textsc{Col}(X)$ is not a total function, since its definition involves a promise on X. Intuitively, however, the collision problem seems much less "structured" than Simon's and Shor's problems. One way to formalize this intuition is as follows.

Definition 1.3. Call a partial function $f : [M]^N \to \{0,1,*\}$ *permutation-invariant* if

$$f(x_1,\ldots,x_N) = f(\tau(x_{\sigma(1)}),\ldots,\tau(x_{\sigma(N)}))$$

for all inputs $X \in [M]^N$ and all permutations $\sigma \in S_N$ and $\tau \in S_M$.

Then $\textsc{Col}(X)$ is permutation-invariant: we can permute a one-to-one sequence and relabel its elements however we like, but it is still a one-to-one sequence, and likewise for a two-to-one sequence. Because of this symmetry, attempts to solve the collision problem using (for example) the Quantum Fourier Transform seem unlikely to succeed. And indeed, in 2002 Aaronson [1] proved that $\mathrm{Q}(\textsc{Col}) = \Omega(N^{1/5})$: that is, the quantum query complexity of the collision problem is at most polynomially better than its randomized query complexity of $\Theta(\sqrt{N})$. The quantum lower bound was later improved to $\Omega(N^{1/3})$ by Aaronson and Shi [4], matching an upper bound of Brassard, Høyer, and Tapp [14].

Generalizing boldly from this example, John Watrous (personal communication) conjectured that the randomized and quantum query complexities are polynomially related for *every* permutation-invariant problem:

Conjecture 1.4 (Watrous 2002)**.** $\mathrm{R}(f) \leq \mathrm{Q}(f)^{O(1)}$ *for every partial function* $f : [M]^N \to \{0,1,*\}$ *that is permutation-invariant.*

Let us make two remarks about Conjecture 1.4. First, the conjecture talks about *randomized* versus quantum query complexity, since in this setting, it is easy to find functions f for which $\mathrm{R}(f)$ and $\mathrm{Q}(f)$ are both tiny but $\mathrm{D}(f)$ is huge. As an example, consider the *Deutsch-Jozsa problem* [18]: given a Boolean input $(x_1,\ldots,x_N)$, decide whether the x_i are all equal or whether half of them are 1 and the other half are 0, under the promise that one of these is the case.

Second, if $M = 2$ (that is, f is Boolean), then Conjecture 1.4 follows relatively easily from known results: indeed, we prove in Appendix 5 that $\mathrm{R}(f) = O(\mathrm{Q}(f)^2)$ in that case. So the interesting case is when $M \gg 2$, as it is for the collision problem.

Conjecture 1.4 provides one natural way to formalize the idea that classical and quantum query complexities should be polynomially related for all "unstructured" problems. A different way is provided by the following conjecture, which we were aware of since about 1999:

Conjecture 1.5 (folklore)**.** *Let Q be a quantum algorithm that makes T queries to a Boolean input $X = (x_1,\ldots,x_N)$, and let $\varepsilon > 0$. Then there exists a deterministic classical algorithm that makes $\mathrm{poly}(T, 1/\varepsilon, 1/\delta)$ queries to the x_i, and that approximates Q's acceptance probability to within an additive error ε on a $1-\delta$ fraction of inputs.*

But what exactly does Conjecture 1.5 have to do with "the need for structure in quantum speedups"? With Conjecture 1.4, the connection to this paper's theme was more-or-less obvious, but with Conjecture 1.5, some additional explanation is probably needed.

Intuitively, we want to say the following: in order to achieve a superpolynomial speedup in the black-box model, a quantum computer needs not merely a promise problem, but a "severely constrained"

promise problem. In other words, only a *minuscule fraction* of the 2^N oracle strings $X = (x_1, \ldots, x_N)$ ought to satisfy the promise—precisely like what happens in Simon's and Shor's problems, where the promise asserts that X encodes a periodic function. If the promise is too "mild"—if, say, it holds for all X in some set $S \subseteq \{0,1\}^N$ with $|S| = \Omega(2^N)$—then we should be back in the situation studied by Beals et al. [9], where the oracle X lacked enough "structure" for a Shor-like algorithm to exploit, and as a result, the best one could hope for was a *polynomial* quantum speedup like that of Grover's algorithm.

Yet, if we interpret the above intuition too naïvely, then it is easy to find counterexamples. To illustrate, let S_1 consist of all strings $X \in \{0,1\}^N$ that encode valid inputs to Simon's problem, let S_0 consist of all $Y \in \{0,1\}^N$ that have Hamming distance at least $N/10$ from every $X \in S_1$, and let $S = S_0 \cup S_1$. Then define a Boolean function $f_{\text{Simon}} : S \to \{0,1\}$ by $f_{\text{Simon}}(X) = 1$ for all $X \in S_1$, and $f_{\text{Simon}}(X) = 0$ for all $X \in S_0$. As observed by Buhrman et al. [16] (see also Ambainis and de Wolf [7] and Hemaspaandra, Hemaspaandra, and Zimand [23]), this "property-testing version of Simon's problem" achieves an exponential separation between randomized and quantum query complexities: $\mathrm{R}(f_{\text{Simon}}) = \Omega(\sqrt{N/\log N})$ while $\mathrm{Q}(f_{\text{Simon}}) = O(\log N)$. But the promise is certainly "mild": indeed $|S| \geq 2^N - 2^{cN}$ for some constant $c < 1$.

On the other hand, examining this counterexample more closely suggests a way to salvage our original intuition. For notice that there exists a fast, deterministic classical algorithm that correctly evaluates $f_{\text{Simon}}(X)$ on *almost all inputs* $X \in S$: namely, the algorithm that always outputs 0! This algorithm errs only on the minuscule fraction of inputs $X \in S$ that happen to belong to S_1. Thus, we might conjecture that this points to a general phenomenon: namely, whenever there exists a fast quantum algorithm to compute a Boolean function $f : S \to \{0,1\}$ with $|S| = \Omega\left(2^N\right)$, there also exists a fast classical algorithm to compute $f(X)$ on *most* inputs $X \in S$. In Appendix 7, we will prove that Conjecture 1.5 is equivalent to this conjecture.

Indeed, Conjecture 1.5 readily implies a far-reaching generalization of the result of Beals et al. [9] stating that $\mathrm{D}(f) = O(\mathrm{Q}(f)^6)$ for all total Boolean functions f. In particular, define the *ε-approximate query complexity* of a Boolean function $f : \{0,1\}^N \to \{0,1\}$, or $\mathrm{D}_\varepsilon(f)$, to be the minimum number of queries made by a deterministic algorithm that evaluates f correctly on at least a $1-\varepsilon$ fraction of inputs X. Likewise, let $\mathrm{Q}_\varepsilon(f)$ be the minimum number of queries made by a quantum algorithm that evaluates f correctly on at least a $1-\varepsilon$ fraction of inputs. Then Conjecture 1.5 implies that $\mathrm{D}_\varepsilon(f)$ and $\mathrm{Q}_\delta(f)$ are polynomially related for all Boolean functions f and all constants $\varepsilon > \delta > 0$ independent of N.[5] This would provide a quantum counterpart to a beautiful 2002 result of Smyth [32], who solved an old open problem of Steven Rudich by showing that $\mathrm{D}_\varepsilon(f) = O(\mathrm{C}_{\varepsilon^3/30}(f)^2/\varepsilon^3)$ for all $\varepsilon > 0$ (where $\mathrm{C}_\delta(f)$ denotes the "δ-approximate certificate complexity" of f).

More dramatically, if Conjecture 1.5 holds, then *we basically cannot hope to prove* $\mathsf{P} \neq \mathsf{BQP}$ *relative to a random oracle.* This would answer a question raised by Fortnow and Rogers [20] in 1998, and would contrast sharply with the situation for *non*-random oracles: we have had oracles relative to which $\mathsf{P} \neq \mathsf{BQP}$, and indeed $\mathsf{BQP} \not\subset \mathsf{MA}$, since the work of Bernstein and Vazirani [12] in the early 1990s. More precisely, under some suitable complexity assumption (such as $\mathsf{P} = \mathsf{P}^{\#\mathsf{P}}$), we would get $\mathsf{BQP}^A \subset \mathsf{AvgP}^A$ with probability 1 for a random oracle A. Here AvgP is the class of languages for which there exists a polynomial-time algorithm that solves a $1-o(1)$ fraction of instances of size n. In other words, separating BQP from AvgP relative to a random oracle would be as hard as separating complexity classes in the

[5] More generally, as we will show in Corollary 3.4, the relation we obtain is $\mathrm{D}_{\varepsilon+\delta}(f) \leq (\mathrm{Q}_\varepsilon(f)/\delta)^{O(1)}$ for all $\varepsilon, \delta > 0$.

unrelativized world. This would provide a quantum counterpart to a theorem of Impagliazzo and Rudich (credited in [24]), who used the powerful results of Kahn, Saks, and Smyth [24] to show that if $\mathsf{P} = \mathsf{NP}$, then $\mathsf{NP}^A \cap \mathsf{coNP}^A \subset \mathsf{ioAvgP}^A$ with probability 1 for a random oracle A.[6]

1.2 Our results

Our main contribution in this paper is essentially to prove Watrous's conjecture (Conjecture 1.4), that randomized and quantum query complexities are polynomially related for every symmetric problem.

Theorem 1.6. $\mathrm{R}(f) = O(\mathrm{Q}(f)^7 \operatorname{polylog} \mathrm{Q}(f))$ *for every partial function* $f : [M]^N \to \{0,1,*\}$ *that is permutation-invariant.*

We conjecture that $\mathrm{R}(f)$ and $\mathrm{Q}(f)$ are polynomially related even for functions f satisfying *one* of the two symmetries: namely, $f(x_1,\ldots,x_N) = f(x_{\sigma(1)},\ldots,x_{\sigma(N)})$ for all $\sigma \in S_N$. We also conjecture that the exponent of 7 can be improved to 2: in other words, that Grover's algorithm once again provides the optimal separation between the quantum and classical models.

While the proof of Theorem 1.6 is somewhat involved, it can be entirely understood by those unfamiliar with quantum computing: the difficulties lie in getting the problem into a form where existing quantum lower bound technology can be applied to it. Let us stress that it was not at all obvious *a priori* that existing quantum lower bounds would suffice here; that they did came as a surprise to us.

We first define and analyze a simple randomized algorithm, which tries to compute $f(X)$ for a given $X = (x_1,\ldots,x_N)$ by estimating the multiplicity of each element x_i. Next, by considering where this randomized algorithm breaks down, we show that one can identify a "hard core" within f: roughly speaking, two input types $\mathcal{A}^*$ and $\mathcal{B}^*$, such that the difficulty of distinguishing $\mathcal{A}^*$ from $\mathcal{B}^*$ accounts for a polynomial fraction of the entire difficulty of computing f. The rest of the proof consists of lower-bounding the quantum query complexity of distinguishing $\mathcal{A}^*$ from $\mathcal{B}^*$. We do so using a hybrid argument: we develop a "chopping procedure" that gradually deforms $\mathcal{A}^*$ to make it more similar to $\mathcal{B}^*$, creating a sequence of intermediate input types $\mathcal{A}_0 = \mathcal{A}^*, \mathcal{A}_1, \mathcal{A}_2, \ldots, \mathcal{A}_{2L} = \mathcal{B}^*$. We then show that, for every $\ell \in [L]$, distinguishing $\mathcal{A}_\ell$ from $\mathcal{A}_{\ell-1}$ requires many quantum queries, *either* by a reduction from Zhandry's recent $\Omega(N^{1/3})$ quantum lower bound for the SETEQUALITY problem [34] (which is a nontrivial generalization of Aaronson and Shi's collision lower bound [4]), or *else* by an application of Ambainis's general quantum adversary theorem [6].

Note that, prior to Zhandry's $\Omega(N^{1/3})$ quantum lower bound for SETEQUALITY, Midrijanis [25] had proved a lower bound of $\Omega((N/\log N)^{1/5})$; the latter was the first quantum lower bound for SETEQUALITY, and the only one for nearly a decade. An earlier version of this paper [3] used Midrijanis's lower bound to show that $\mathrm{R}(f) = O(\mathrm{Q}(f)^9 \operatorname{polylog} \mathrm{Q}(f))$ for all permutation-symmetric f. The improvement to $\mathrm{R}(f) = O(\mathrm{Q}(f)^7 \operatorname{polylog} \mathrm{Q}(f))$ in the current version comes entirely from Zhandry's improvement of the SETEQUALITY lower bound to the optimal $\Omega(N^{1/3})$.

Carrying out the hybrid argument in the "obvious" way produces a bound of the form $\mathrm{R}(f) \leq \mathrm{Q}(f)^{O(1)} \operatorname{polylog} N$, which fails to imply a polynomial relationship between $\mathrm{R}(f)$ and $\mathrm{Q}(f)$ when $\mathrm{Q}(f) \leq (\log N)^{o(1)}$. However, a more sophisticated hybrid argument eliminates the $\operatorname{polylog} N$ factor.

[6] Here ioAvgP means "average-case P for infinitely many input lengths n." The reason Impagliazzo and Rudich only get a simulation in ioAvgP, rather than AvgP, has to do with the fact that Smyth's result [32] only relates $\mathrm{D}_\varepsilon(f)$ to $\mathrm{C}_{\varepsilon^3/30}(f)$, rather than $\mathrm{D}_{\varepsilon+\delta}(f)$ to $\mathrm{C}_\varepsilon(f)$ for all $\delta > 0$.

Our second contribution is more exploratory, something we put forward in the hope of inspiring followup work. We study Conjecture 1.5, which states that every T-query quantum algorithm can be simulated on *most* inputs using $T^{O(1)}$ classical queries. We relate this conjecture to a fundamental open problem in Fourier analysis and approximation theory. Given a real polynomial $p : \mathbb{R}^N \to \mathbb{R}$, let

$$\mathrm{Inf}_i[p] := \mathop{\mathrm{E}}_{X \in \{0,1\}^N} \left[(p(X) - p(X^i))^2 \right]$$

be the *influence* of the i^{th} variable, where X^i means X with the i^{th} bit flipped. Then we conjecture that *every bounded low-degree polynomial has a "highly influential" variable*. More precisely:

Conjecture 1.7 (Bounded Polynomials Have Influential Variables)**.** *Let $p : \mathbb{R}^N \to \mathbb{R}$ be a polynomial of degree d. Suppose that $0 \leq p(X) \leq 1$ for all $X \in \{0,1\}^N$, and*

$$\mathop{\mathrm{E}}_{X \in \{0,1\}^N} \left[(p(X) - \mathrm{E}[p])^2 \right] \geq \varepsilon .$$

Then there exists an i such that $\mathrm{Inf}_i[p] \geq (\varepsilon/d)^{O(1)}$.

We show the following:

Theorem 1.8. *Assume Conjecture 1.7. Then*

(i) Conjecture 1.5 holds.

(ii) $\mathrm{D}_{\varepsilon+\delta}(f) \leq (\mathrm{Q}_\varepsilon(f)/\delta)^{O(1)}$ for all Boolean functions $f : \{0,1\}^N \to \{0,1\}$ and all $\varepsilon, \delta > 0$.

(iii) If $\mathsf{P} = \mathsf{P}^{\#\mathsf{P}}$, then $\mathsf{BQP}^A \subset \mathsf{AvgP}^A$ with probability 1 for a random oracle A.

The main evidence for Conjecture 1.7—besides the fact that all the Fourier analysis experts we asked were confident of it!—is that extremely similar statements have recently been proved. Firstly, O'Donnell, Saks, Schramm, and Servedio [27] proved an analogue of Conjecture 1.7 for *decision trees*, which are a special case of bounded real polynomials:

Theorem 1.9 (O'Donnell et al. 2005)**.** *Let $f : \{0,1\}^N \to \{0,1\}$ be a Boolean function, and suppose $\Pr[f=1]\Pr[f=0] \geq \varepsilon$. Then there exists an i such that $\mathrm{Inf}_i[f] \geq 4\varepsilon/\mathrm{D}(f)$, where $\mathrm{D}(f)$ is the decision-tree complexity of f.*

Unfortunately, Theorem 1.9 does not directly imply anything about our problem, even though Beals et al. [9] showed that $\mathrm{D}(f)$ and $\mathrm{Q}(f)$ are polynomially related for all total Boolean functions f. The reason is that the acceptance probability of a quantum algorithm need not approximate a total Boolean function.

The second piece of evidence for Conjecture 1.7 comes from a powerful result of Dinur, Friedgut, Kindler, and O'Donnell [19], which implies our conjecture, *except* with $\mathrm{Inf}_i[p] \geq \varepsilon^3/2^{O(d)}$ instead of $\mathrm{Inf}_i[p] \geq (\varepsilon/d)^{O(1)}$. Let us state the special case of their result that is relevant for us:

Theorem 1.10 (Dinur et al. 2006)**.** *Let $\varepsilon > 0$, and let $p : \mathbb{R}^N \to \mathbb{R}$ be a degree-d polynomial such that $0 \leq p(X) \leq 1$ for all $X \in \{0,1\}^N$. Then there exists a $2^{O(d)}/\varepsilon^2$-junta $\widetilde{p} : \mathbb{R}^N \to \mathbb{R}$ (that is, a polynomial depending on at most $2^{O(d)}/\varepsilon^2$ variables) such that*

$$\mathop{\mathrm{E}}_{X \in \{0,1\}^N} \left[\left(\widetilde{p}(X) - p(X) \right)^2 \right] \leq \varepsilon .$$

Even though Theorem 1.10 has an exponential rather than polynomial dependence on $1/d$, we observe that it already has a nontrivial consequence for quantum computation. Namely, it implies that any T-query quantum algorithm can be simulated on *most* inputs using $2^{O(T)}$ classical queries.[7] Recall that the gaps between classical and quantum query complexities can be superexponential (and even $N^{\Omega(1)}$ versus $O(1)$, as in the example of Aaronson [2]), so even an exponential upper bound is far from obvious.

1.3 Subsequent work

Since the first version of this paper appeared on arXiv [3] was circulated, there have been at least three interesting developments (not counting the $\Omega(N^{1/3})$ quantum lower bound of Zhandry [34] for SETEQUALITY, which we incorporate here).

First, Yuen [33] adapted the hybrid argument that we used to prove Theorem 1.6, in order to show that distinguishing a random function $X : [N] \to [N]$ from a random permutation requires $\Omega(N^{1/5}/\log N)$ quantum queries. (Subsequently, however, Zhandry [34] proved a tight lower bound of $\Omega(N^{1/3})$ for the random function versus random permutation problem, using different ideas.)

Second, Montanaro [26] used a hypercontractive inequality to prove Conjecture 1.7, in the special case where p is a multilinear polynomial all of whose coefficients (when written in the Fourier basis) have the same absolute value. Currently, it remains open to generalize Montanaro's technique to arbitrary multilinear polynomials, let alone arbitrary polynomials.

Third, Bačkurs and Bavarian [8] solved a technical problem that arose from an earlier version of this paper [3]. In that version, we stated Conjecture 1.7 in terms of L_1-influences rather than L_2-influences, and we also used the L_1-norm in proving the consequences of Conjecture 1.7 for quantum query complexity. Subsequently, Bačkurs (personal communication) found an error in our proof. Fortunately, however, we noticed that (a) our proof could be fixed by simply switching from L_1-norm to L_2-norm throughout, and (b) the L_2 version of Conjecture 1.7 was, in any case, provably equivalent to our original L_1 version. So we switched to the L_2-norm. At the same time, though, we remained curious whether our original L_1-based argument *could have* worked. The question boiled down to the following: given a degree-d real polynomial $p : \mathbb{R}^N \to \mathbb{R}$, let

$$\mathrm{Inf}_i^1[p] := \mathop{\mathrm{E}}_{X \in \{0,1\}^N} \left[\left| p(X) - p(X^i) \right| \right] .$$

Then do we have $\sum_{i=1}^N \mathrm{Inf}_i^1[p] \leq d^{O(1)}$, whenever $p(X) \in [0,1]$ for all $X \in \{0,1\}^N$? Bačkurs and Bavarian [8] show that the answer is yes: indeed, the sum of the L_1-influences is upper-bounded by $O(d^3 \log d)$. Using their result, one *can* salvage our original L_1-based argument.

For simplicity, though, in this version of the paper we stick with L_2-influences. There, the analogue of Bačkurs and Bavarian's result is much easier, and states that $\sum_{i=1}^N \mathrm{Inf}_i[p] \leq d$ (we provide the folklore

[7] Indeed, in this case the classical queries are nonadaptive.

proof in Lemma 3.1). For completeness, in Appendix 6 we prove the equivalence of the L_1 and L_2 versions of Conjecture 1.7.

2 Quantum lower bound for all symmetric problems

In this section we prove Theorem 1.6: that $\mathrm{R}(f) = O(\mathrm{Q}(f)^7 \operatorname{polylog} \mathrm{Q}(f))$ for all permutation-symmetric f.

We start with a simple observation that is essential to everything that follows. Since f is symmetric, we can group the inputs $X = (x_1, \ldots, x_N)$ into equivalence classes that we call *types*.

Definition 2.1. Given an input $X = (x_1, \ldots, x_N) \in [M]^N$, the type of X is a list of positive integers $\mathcal{A} = (a_1, \ldots, a_u)$, which records the multiplicities of the integers occurring in X from most to least frequent. So in particular, $a_1 \geq \cdots \geq a_u$ and $a_1 + \cdots + a_u = N$. For convenience, we adopt the convention that $a_i = 0$ for all $i > u$.

In other words, a type is just a partition (or *Young diagram*) that records the multiplicities of the input elements. For example, a one-to-one input has type $a_1 = \cdots = a_N = 1$, while a two-to-one input has type $a_1 = \cdots = a_{N/2} = 2$. We write $X \in \mathcal{A}$ if X is of type $\mathcal{A}$. Clearly $f(X)$ depends only on the type of X. Furthermore, given a quantum query algorithm Q, we can assume without loss of generality that $\Pr[Q \text{ accepts } X]$ depends only on the type of X—since we can "symmetrize" Q (that is, randomly permute X's inputs and outputs) prior to running Q.

2.1 Randomized upper bound

Let $X = (x_1, \ldots, x_N)$ be an input. For each $j \in [M]$, let κ_j be the number of subscripts i such that $x_i = j$. Then the first step is to give a classical randomized algorithm that estimates the κ_j. This algorithm, $\mathcal{S}_T$, is an extremely straightforward sampling procedure. (Indeed, there is essentially nothing else that a randomized algorithm can do here.) $\mathcal{S}_T$ will make $O(T^{1+c} \log T)$ queries, where T is a parameter and $c \in (0,1]$ is a constant that we will choose later to optimize the final bound.

```
Set U := 21 T^{1+c} ln T
Choose U indices i_1,...,i_U ∈ [N] uniformly at random with replacement
Query x_{i_1},...,x_{i_U}
For each j ∈ [M]:
    Let z_j be the number of occurrences of j in (x_{i_1},...,x_{i_U})
    Output κ̃_j := (z_j / U) · N as the estimate for κ_j
```

We now analyze how well $\mathcal{S}_T$ works.

Lemma 2.2. *With probability* $1 - O(1/T)$*, we have*

$$\left|\widetilde{\kappa}_j - \kappa_j\right| \leq \frac{N}{T} + \frac{\kappa_j}{T^c}$$

for all $j \in [M]$.

Proof. For each $j \in [M]$, we consider four cases. First suppose $\kappa_j \geq N/T^{1-c}$. Notice that z_j is a sum of U independent Boolean variables, and that

$$\mathrm{E}[z_j] = \frac{U}{N}\mathrm{E}[\widetilde{\kappa}_j] = \frac{U}{N}\kappa_j .$$

Thus

$$\begin{aligned}
\Pr\left[\left|\widetilde{\kappa}_j - \kappa_j\right| > \frac{\kappa_j}{T^c}\right] &= \Pr\left[\left|z_j - \frac{U}{N}\kappa_j\right| > \frac{U\kappa_j}{NT^c}\right] \\
&< 2\exp\left(-\frac{U\kappa_j/N}{3T^{2c}}\right) \\
&< 2\exp\left(-\frac{U}{3T^{1+c}}\right) \\
&= 2T^{-7},
\end{aligned}$$

where the second line follows from a Chernoff bound and the third from $\kappa_j \geq N/T^{1-c}$.

Second, suppose $N/T \leq \kappa_j < N/T^{1-c}$. Then

$$\begin{aligned}
\Pr\left[\left|\widetilde{\kappa}_j - \kappa_j\right| > \frac{N}{T}\right] &= \Pr\left[\left|z_j - \frac{U}{N}\kappa_j\right| > \frac{U}{T}\right] \\
&< 2\exp\left(-\frac{U\kappa_j/N}{3}\left(\frac{N}{T\kappa_j}\right)^2\right) \\
&< 2\exp\left(-\frac{U}{3T^{1+c}}\right) \\
&= 2T^{-7}
\end{aligned}$$

where the second line follows from a Chernoff bound (which is valid because $N/(T\kappa_j) \leq 1$) and the third from $\kappa_j < N/T^{1-c}$.

Third, suppose $N/T^6 \leq \kappa_j < N/T$. Then

$$\begin{aligned}
\Pr\left[\left|\widetilde{\kappa}_j - \kappa_j\right| > \frac{N}{T}\right] &= \Pr\left[\left|z_j - \frac{U}{N}\kappa_j\right| > \frac{U}{T}\right] \\
&< \left(\frac{e^{N/(T\kappa_j)}}{(1+N/(T\kappa_j))^{1+N/(T\kappa_j)}}\right)^{U\kappa_j/N} \\
&\leq \exp\left(-\frac{N}{T\kappa_j}\cdot\frac{U\kappa_j}{N}\right) \\
&= \exp\left(-\frac{U}{T}\right) \\
&= O\left(\frac{1}{T^7}\right),
\end{aligned}$$

where the second line follows from a Chernoff bound, the third line follows from $N/(T\kappa_j) > 1$, and the last follows from $U = 21T^{1+c}\ln T$.

Fourth, suppose $\kappa_j < N/T^6$. Then

$$\begin{aligned}\Pr\left[\left|\widetilde{\kappa}_j - \kappa_j\right| > \frac{N}{T}\right] &= \Pr\left[\left|z_j - \frac{U}{N}\kappa_j\right| > \frac{U}{T}\right] \\ &\leq \Pr[z_j \geq 2] \\ &\leq \binom{U}{2}\left(\frac{\kappa_j}{N}\right)^2 \\ &\leq \frac{U^2}{T^6}\left(\frac{\kappa_j}{N}\right) \\ &\leq \frac{\kappa_j}{TN}\end{aligned}$$

for all sufficiently large T, where the second line follows from $\kappa_j < N/T^6$, the third from the union bound, the fourth from $\kappa_j < N/T^6$ (again), and the fifth from $U \leq 21T^2\ln T$.

Notice that there are at most T^6 values of j such that $\kappa_j \geq N/T^6$. Hence, putting all four cases together,

$$\begin{aligned}\Pr\left[\exists j : \left|\widetilde{\kappa}_j - \kappa_j\right| > \frac{N}{T} + \frac{\kappa_j}{T^c}\right] &\leq T^6 \cdot O\left(\frac{1}{T^7}\right) + \sum_{j:\kappa_j < N/T^6} \frac{\kappa_j}{TN} \\ &= O\left(\frac{1}{T}\right).\end{aligned}$$

□

Now call $\mathcal{A}$ a 1-*type* if $f(X) = 1$ for all $X \in \mathcal{A}$, or a 0-*type* if $f(X) = 0$ for all $X \in \mathcal{A}$. Consider the following randomized algorithm $\mathcal{R}_T$ to compute $f(X)$:

```
Run S_T to find an estimate κ̃_i for each κ_i
Sort the κ̃_i's in descending order, so that κ̃_1 ≥ ⋯ ≥ κ̃_M
If there exists a 1-type A = (a_1, a_2, ...) such that |κ̃_i − a_i| ≤ N/T + a_i/T^c
    for all i, then output f(X) = 1
Otherwise output f(X) = 0
```

Clearly $\mathcal{R}_T$ makes $O(T^{1+c}\log T)$ queries, just as $\mathcal{S}_T$ does. We now give a sufficient condition for $\mathcal{R}_T$ to succeed.

Lemma 2.3. *Suppose that for all* 1*-types* $\mathcal{A} = (a_1, a_2, \ldots)$ *and* 0*-types* $\mathcal{B} = (b_1, b_2, \ldots)$*, there exists an* i *such that*

$$|a_i - b_i| > \frac{2N}{T} + \frac{a_i + b_i}{T^c}.$$

Then $\mathcal{R}_T$ *computes* f *with bounded probability of error, and hence* $\mathrm{R}(f) = O(T^{1+c}\log T)$.

Proof. First suppose $X \in \mathcal{A}$ where $\mathcal{A} = (a_1, a_2, \ldots)$ is a 1-type. Then by Lemma 2.2, with probability $1 - O(1/T)$ we have

$$|\widetilde{\kappa}_i - a_i| \leq \frac{N}{T} + \frac{a_i}{T^c}$$

for all i. (It is easy to see that sorting the $\widetilde{\kappa}_i$ can only decrease the maximum difference.) Provided this occurs, $\mathcal{R}_T$ finds some 1-type close to $(\widetilde{\kappa}_1, \widetilde{\kappa}_2, \ldots)$ (possibly $\mathcal{A}$ itself) and outputs $f(X) = 1$.

Second, suppose $X \in \mathcal{B}$ where $\mathcal{B} = (b_1, b_2, \ldots)$ is a 0-type. Then with probability $1 - O(1/T)$ we have

$$|\widetilde{\kappa}_i - b_i| \leq \frac{N}{T} + \frac{b_i}{T^c}$$

for all i. Provided this occurs, by the triangle inequality, for every 1-type $\mathcal{A} = (a_1, a_2, \ldots)$ there exists an i such that

$$|\widetilde{\kappa}_i - a_i| \geq |a_i - b_i| - |\widetilde{\kappa}_i - b_i| > \frac{N}{T} + \frac{a_i}{T^c}.$$

Hence $\mathcal{R}_T$ does *not* find a 1-type close to $(\widetilde{\kappa}_1, \widetilde{\kappa}_2, \ldots)$, and it outputs $f(X) = 0$. □

In particular, suppose we keep decreasing T until there exists a 1-type $\mathcal{A}^* = (a_1, a_2, \ldots)$ and a 0-type $\mathcal{B}^* = (b_1, b_2, \ldots)$ such that

$$|a_i - b_i| \leq \frac{2N}{T} + \frac{a_i + b_i}{T^c} \tag{2.1}$$

for all i, stopping as soon as that happens. Then Lemma 2.3 implies that we will still have $\mathrm{R}(f) = O(T^{1+c} \log T)$. For the rest of the proof, we will fix that "almost as small as possible" value of T for which (2.1) holds, as well as the 1-type $\mathcal{A}^*$ and the 0-type $\mathcal{B}^*$ that $\mathcal{R}_T$ "just barely distinguishes" from one another.

2.2 The chopping procedure

Given two sets of inputs A and B with $A \cap B = \varnothing$, let $\mathrm{Q}(A,B)$ be the minimum number of queries made by any quantum algorithm that accepts every $X \in A$ with probability at least $2/3$, and accepts every $Y \in B$ with probability at most $1/3$. Also, let $\mathrm{Q}_\varepsilon(A,B)$ be the minimum number of queries made by any quantum algorithm that accepts every $X \in A$ with at least some probability p, and that accepts every $Y \in B$ with probability at most $p - \varepsilon$. Then we have the following basic relation:

Proposition 2.4. $\mathrm{Q}(A,B) = O(\frac{1}{\varepsilon}\mathrm{Q}_\varepsilon(A,B))$ *for all* A, B *and all* $\varepsilon > 0$.

Proof. This follows from standard amplitude estimation techniques (see Brassard et al. [13] for example). □

The rest of the proof consists of lower-bounding $\mathrm{Q}(\mathcal{A}^*, \mathcal{B}^*)$, the quantum query complexity of distinguishing inputs of type $\mathcal{A}^*$ from inputs of type $\mathcal{B}^*$. We do this via a hybrid argument. Let $L := \lceil \log_2 N \rceil + 1$. At a high level, we will construct a sequence of types $\mathcal{A}_0, \ldots, \mathcal{A}_{2L}$ such that

(i) $\mathcal{A}_0 = \mathcal{A}^*$,

(ii) $\mathcal{A}_{2L} = \mathcal{B}^*$, and

(iii) $\mathrm{Q}(\mathcal{A}_\ell, \mathcal{A}_{\ell-1})$ is large for every $\ell \in [2L]$.

Provided we can do this, it is not hard to see that we get the desired lower bound on $\mathrm{Q}(\mathcal{A}^*, \mathcal{B}^*)$. Suppose a quantum algorithm distinguishes $\mathcal{A}_0 = \mathcal{A}^*$ from $\mathcal{A}_{2L} = \mathcal{B}^*$ with constant bias. Then by the triangle inequality, it must also distinguish *some* $\mathcal{A}_\ell$ from $\mathcal{A}_{\ell+1}$ with reasonably large bias (say $\Omega(1/\log N)$). By Proposition 2.4, any quantum algorithm that succeeds with bias ε can be amplified, with $O(1/\varepsilon)$ overhead, to an algorithm that succeeds with constant bias.

Incidentally, the need, in this hybrid argument, to amplify the distinguishing bias $\varepsilon = \varepsilon_\ell$ from $\Omega(1/\log N)$ to $\Omega(1)$ is exactly what could produce an undesired $1/\log N$ factor in our final lower bound on $\mathrm{Q}(f)$, if we were not careful. (We mentioned this issue in Section 1.2.) The way we will solve this problem, roughly speaking, is to design the $\mathcal{A}_\ell$ in such a way that our lower bounds on $\mathrm{Q}(\mathcal{A}_\ell, \mathcal{A}_{\ell-1})$ increase quickly as functions of ℓ. That way, we can take the biases ε_ℓ to decrease quadratically with ℓ (thus summing to a constant), yet still have $\mathrm{Q}(\mathcal{A}_\ell, \mathcal{A}_{\ell-1})$ increasing quickly enough that

$$\mathrm{Q}_{\varepsilon_\ell}(\mathcal{A}_\ell, \mathcal{A}_{\ell-1}) = \Omega(\varepsilon_\ell \, \mathrm{Q}(\mathcal{A}_\ell, \mathcal{A}_{\ell-1}))$$

remain "uniformly large," with $1/\log T$ factors but no $1/\log N$ factor.

We now describe the procedure for creating the intermediate types $\mathcal{A}_\ell$. Intuitively, we want to form $\mathcal{A}_\ell$ from $\mathcal{A}_{\ell-1}$ by making its Young diagram more similar to that of $\mathcal{B}^*$, by decreasing the rows of $\mathcal{A}_{\ell-1}$ which are larger than the corresponding rows of $\mathcal{B}^*$ and increasing the rows of $\mathcal{A}_{\ell-1}$ which are smaller than the corresponding rows of $\mathcal{B}^*$.

More precisely, we construct the intermediate types $\mathcal{A}_1, \mathcal{A}_2, \ldots$ via the following procedure. In this procedure, $(a_1, a_2, \ldots)$ is an input type that is initialized to $\mathcal{A}^*$, and $\mathcal{B}^* = (b_1, b_2, \ldots)$.

```
let P be the first power of 2 greater than or equal to N
for ℓ := 1 to L
    let S_A be the set of i such that  a_i − b_i ≥ P/2^ℓ
    let S_B be the set of i such that  b_i − a_i ≥ P/2^ℓ
    let m := min(|S_A|, |S_B|)
    choose m elements i from S_A, set  a_i := a_i − P/2^ℓ  and remove them from S_A
    choose m elements i from S_B, set  a_i := a_i + P/2^ℓ  and remove them from S_B
    let A_{2ℓ−1} := type(a_1, a_2, ...)
    if |S_A| > 0
        let  a_i := a_i − P/2^ℓ  for all  i ∈ S_A
        choose |S_A| elements i such that a_i < b_i  and set  a_i := a_i + P/2^ℓ
    if |S_B| > 0
        let  a_i := a_i + P/2^ℓ  for all  i ∈ S_B
        choose |S_B| elements i such that a_i > b_i  and set  a_i := a_i − P/2^ℓ
    let A_{2ℓ} := type(a_1, a_2, ...)
next ℓ
```

The procedure is illustrated pictorially in Figure 1. Intuitively, the main goal we want to achieve is that, after ℓ stages (that is, for $\mathcal{A}_{2\ell}$), we have $|a_i - b_i| < P/2^\ell$ for all i. To achieve that, we increase all

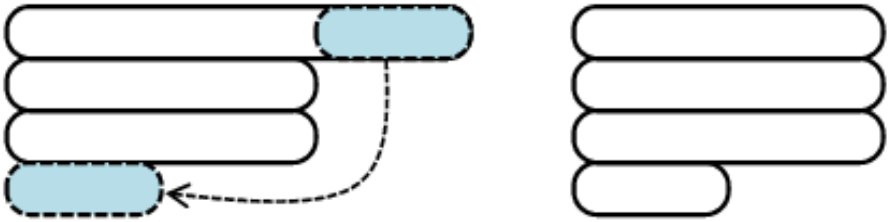

Figure 1: Chopping a row of $\mathcal{A}_\ell$'s Young diagram to make it more similar to $\mathcal{B}_\ell$.

a_i for i such that $a_i \leq b_i - P/2^\ell$ by $P/2^\ell$, and decrease all a_i for i such that $a_i \geq b_i + P/2^\ell$ by $P/2^\ell$. If the number of i such that $a_i \leq b_i - P/2^\ell$ does not equal the number of i such that $a_i \geq b_i + P/2^\ell$, we no longer have $a_1 + a_2 + \ldots = N$. To deal with that, we increase or decrease the correct number of other a_i by $P/2^\ell$, and choose those a_i so that, after the increase or decrease, we have $|a_i - b_i| < P/2^\ell$.

We start with some simple observations. First, by construction, this procedure halts after $L = O(\log N)$ iterations.

By induction, after the ℓ^{th} iteration, we have $|a_i - b_i| < P/2^\ell$ for all i. (Let a'_i be the value of a_i before the ℓ^{th} iteration. Because of the inductive assumption, we must have $|a'_i - b_i| < P/2^{\ell-1}$. If $|a'_i - b_i| \in [P/2^\ell, P/2^{\ell-1} - 1]$, then a_i is increased (decreased by $P/2^\ell$), resulting in

$$|a_i - b_i| \in \left[\frac{P}{2^\ell} - \frac{P}{2^\ell}, \frac{P}{2^{\ell-1}} - \frac{P}{2^\ell} - 1\right] = \left[0, \frac{P}{2^\ell} - 1\right].$$

If $|a'_i - b_i| < P/2^\ell$, a_i is increased only if $a'_i < b_i$ and a_i is decreased only if $a'_i > b_i$. Therefore, after the change, the sign of the difference $a_i - b_i$ flips and $|a_i - b_i| < P/2^\ell$.)

In particular, this means that, for $\ell = L$, $|a_i - b_i| < P/2^\ell \leq 1$. That is, we have $a_i = b_i$ for all i and $\mathcal{A}_{2L} = \mathcal{B}^*$.

We define

$$\|\mathcal{A} - \mathcal{B}\| := \frac{1}{2}\sum_{i=1}^{N} |a_i - b_i|.$$

Notice that $\|\mathcal{A}_{2\ell-2} - \mathcal{A}_{2\ell-1}\| + \|\mathcal{A}_{2\ell-1} - \mathcal{A}_{2\ell}\| = rP/2^{\ell'}$ where r is the number of rows that get increased (or decreased) in the ℓ^{th} iteration. We now prove an upper bound on $\|\mathcal{A}_\ell - \mathcal{A}_{\ell-1}\|$ when ℓ is small, which will be useful later.

Lemma 2.5. *If $\ell \leq (\log_2 T) - 2$, then*

$$\|\mathcal{A}_{2\ell-2} - \mathcal{A}_{2\ell-1}\| + \|\mathcal{A}_{2\ell-1} - \mathcal{A}_{2\ell}\| \leq \frac{4N}{T^c}.$$

Proof. Let $m := \max(|S_A|, |S_B|)$. Then

$$\|\mathcal{A}_{2\ell-2} - \mathcal{A}_{2\ell-1}\| + \|\mathcal{A}_{2\ell-1} - \mathcal{A}_{2\ell}\| = m\frac{P}{2^\ell}.$$

Without loss of generality, we assume that $m = |S_A|$. To show the lemma, it suffices to prove that

$$|S_A| \leq \frac{4N/T^c}{P/2^\ell}.$$

We consider the sum $\sum_{j\in R}|a_j - b_j|$ where R is the set of all j such that $|a_j - b_j| \geq P/2^\ell$, with $(a_1, a_2, \ldots)$ evolving from $\mathcal{A}_0$ to $\mathcal{A}_{2\ell-2}$ and $\mathcal{B}^* = (b_1, b_2, \ldots)$ fixed. Initially (when $(a_1, a_2, \ldots) = \mathcal{A}_0$), we have

$$\frac{P}{2^\ell} \leq |a_j - b_j| \leq \frac{2N}{T} + \frac{a_j + b_j}{T^c}$$

for each $j \in R$. Since $\ell \leq (\log_2 T) - 2$, the left inequality implies

$$|a_j - b_j| \geq \frac{4N}{T},$$

which combined with the right inequality yields

$$\frac{a_j + b_j}{T^c} \geq \frac{2N}{T}. \tag{2.2}$$

Therefore

$$\begin{aligned}\sum_{i\in R}|a_i - b_i| &\leq \sum_{i\in R}\left(\frac{2N}{T} + \frac{a_i + b_i}{T^c}\right)\\ &\leq 2\sum_{i\in R}\frac{a_i + b_i}{T^c}\\ &\leq \frac{4N}{T^c},\end{aligned}$$

where the third line uses (2.2).

The sum $\sum_{i\in R}|a_i - b_i|$ is not increased by any step of the algorithm that generates $\mathcal{A}_0, \ldots, \mathcal{A}_{2\ell-2}$. Therefore, at the beginning of the ℓ^{th} iteration, we still have $\sum_{i\in R}|a_i - b_i| \leq 4N/T^c$. This means that

$$|S_A| \leq \frac{4N/T^c}{P/2^\ell}.$$

□

2.3 Quantum lower bounds

Recall that we listed four properties that we needed the chopping procedure to satisfy. We have already seen that it satisfies properties (i)-(ii), so the remaining step is to show that it satisfies property (iii). That is, we need to lower-bound $\mathrm{Q}(\mathcal{A}_\ell, \mathcal{A}_{\ell-1})$, the bounded-error quantum query complexity of distinguishing inputs of type $\mathcal{A}_\ell$ from inputs of type $\mathcal{A}_{\ell-1}$. To do this, it will be convenient to consider two cases: first, that forming $\mathcal{A}_\ell$ involved chopping few elements of $\mathcal{A}_{\ell-1}$, and second, that it involved chopping many elements. We will show that we "win either way," by a different quantum lower bound in each case.

First consider the case that few elements were chopped. Here we prove a lower bound using Ambainis's quantum adversary method [6], in its "general" form (the one used, for example, to lower-bound the quantum query complexity of inverting a permutation). For completeness, we now state Ambainis's adversary theorem in the form we will need.

Theorem 2.6 (Ambainis [6]). *Let $A,B \subseteq [M]^N$ be two sets of inputs with $A \cap B = \varnothing$. Let $R \subseteq A \times B$ be a relation on input pairs, such that for every $X \in A$ there exists at least one $Y \in B$ with $(X,Y) \in R$ and vice versa. Given inputs $X = (x_1,\ldots,x_N)$ in A and $Y = (y_1,\ldots,y_N)$ in B, let*

$$q_{X,i} = \Pr_{Y \in B}[x_i \neq y_i \mid (X,Y) \in R],$$
$$q_{Y,i} = \Pr_{X \in A}[x_i \neq y_i \mid (X,Y) \in R].$$

Suppose that $q_{X,i}q_{y,i} \leq \alpha$ for every $(X,Y) \in R$ and every $i \in [N]$ such that $x_i \neq y_i$. Then $\mathrm{Q}(A,B) = \Omega(1/\sqrt{\alpha})$.

Using Theorem 2.6, we can prove the following lower bound on $\mathrm{Q}(\mathcal{A}_\ell, \mathcal{A}_{\ell-1})$.

Lemma 2.7. *Let $d = \|\mathcal{A}_\ell - \mathcal{A}_{\ell-1}\|$, and assume $d \leq N/2$. Then $\mathrm{Q}(\mathcal{A}_\ell, \mathcal{A}_{\ell-1}) = \Omega(\sqrt{N/d})$.*

Proof. Let $\mathcal{A}_{\ell-1} = (a_1, a_2, \ldots)$, and let $\ell' = \lceil \ell/2 \rceil$. Then in the transition from $\mathcal{A}_{\ell-1}$ to $\mathcal{A}_\ell$, we augment or chop various rows by $P/2^{\ell'}$ elements each. Let $i(1),\ldots,i(r)$ be the r rows in $\mathcal{A}_{\ell-1}$ that get chopped and let $i'(1),\ldots,i'(r)$ be the r rows in $\mathcal{A}_{\ell-1}$ that get augmented.

Fix distinct $h_1,\ldots,h_r \in [M]$ and $h'_1,\ldots,h'_r \in [M]$. Also, let us restrict ourselves to inputs such that for each $j \in [r]$, there are exactly $a_{i(j)}$ indices $i \in [N]$ satisfying $x_i = h_j$ and exactly $a_{i'(j)}$ indices $i \in [N]$ satisfying $x_i = h'_j$. Given inputs $X = (x_1,\ldots,x_N)$ in $\mathcal{A}_{\ell-1}$ and $Y = (y_1,\ldots,y_N)$ in $\mathcal{A}_\ell$, we set $(X,Y) \in R$ if and only if it is possible to transform X to Y in the following way:

(1) For each $j \in [r]$, change exactly $P/2^{\ell'}$ of the x_i that are equal to h_j to value h'_j. (Let $d := P/2^{\ell'}$ be the number of changed elements.)

(2) Swap the d elements of X that were changed in step (1) with any other d elements x_i of X, subject to the following constraints:

   (a) we do not use x_i such that $x_i = h_j$ for some j and $\dfrac{a_{i(j)}}{P/2^{\ell'}} < \dfrac{N-d}{3d}$;

   (b) we do not use x_i such that $x_i = h'_j$ for some j and $\dfrac{a_{i(j)} - P/2^{\ell'}}{P/2^{\ell'}} < \dfrac{N-d}{3d}$.

The procedure is illustrated pictorially in Figure 2. Note that we can reverse the procedure in a natural way to go from Y back to X:

(1) For each $j \in [r]$, change exactly $P/2^{\ell'}$ of the x_i that are equal to h'_j to value h_j. (Let $d := P/2^{\ell'}$.)

(2) Swap the d elements of X that were changed in step (1) with any d elements x_i of X, subject to the same constraints as in the step (2) of the $X \to Y$ conversion.

Fix any $(X,Y) \in R$, and let $i \in [N]$ be any index such that $x_i \neq y_i$. Then we claim that the parameters of Theorem 2.6 satisfy either

$$q_{X,i} \leq \frac{6d}{N-d} \qquad \text{or} \qquad q_{Y,i} \leq \frac{6d}{N-d}.$$

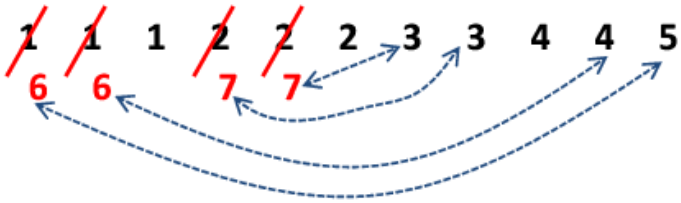

Figure 2: In this example, $N = 11$, $r = 2$, $P/2^{\ell} = 2$, and $a_1 = a_2 = 3$. So we transform X to Y by choosing $h_1 = 1$ and $h_2 = 2$, changing any two elements equal to h_1 and any two elements equal to h_2, and then swapping the four elements that we changed with four unchanged elements.

To see this, let us write $q_{X,i} = q'_{X,i} + q''_{X,i}$, where $q'_{X,i}$ is the probability that x_i is changed in step (1) of the $X \to Y$ conversion and $q''_{X,i}$ is the probability that x_i is not changed in step (1), but is swapped with some changed element in step (2). We also express $q_{Y,i}$ in a similar way, with respect to the $Y \to X$ conversion.

We consider two cases. The first case is that x_i is one of the "other d elements" with which we swap the changed elements in step (2) of the $X \to Y$ conversion. In this case, $q'_{X,i} \neq 0$ only if $x_i = h_j$ for some j. Then because of the constraint (a), we have $q'_{X,i} \leq 3d/(N-2d)$. (The probability that this particular $x_i = h_j$ is chosen for being changed is equal to $(P/2^{\ell'})/a_{i(j)}$ and, because of (a), this is at most $3d/(N-2d)$.)

We also have

$$q''_{X,i} = \Pr_{Y' \in \mathcal{A}_\ell}\left[x_i \neq y'_i \mid (X,Y') \in R\right] \leq \frac{d}{(N-d)/3} = \frac{3d}{N-d},$$

because each of the constraints (a) and (b) eliminates at most $(N-d)/3$ of the $N-d$ variables x_i that are available for swapping in step (2). Therefore, $q_{X,i} = q'_{X,i} + q''_{X,i} \leq 6d/(N-d)$.

The second case is that x_i is one of the elements that are changed in step (1) of the $X \to Y$ conversion. Then y_i is one of the "other d elements" in step (2) of the $Y \to X$ conversion. Similarly to the previous case, we can show that $q_{Y,i} \leq 6d/(N-d)$.

Since $q_{X,i} \leq 1$ and $q_{Y,i} \leq 1$, it follows that

$$q_{X,i} q_{Y,i} \leq \frac{6d}{N-d}.$$

Thus, by Theorem 2.6,

$$\mathrm{Q}(\mathcal{A}_\ell, \mathcal{A}_{\ell-1}) = \Omega\left(\frac{1}{\sqrt{q_{X,i} q_{Y,i}}}\right) = \Omega\left(\sqrt{\frac{N-d}{d}}\right) = \Omega\left(\sqrt{\frac{N}{d}}\right). \qquad \square$$

We now consider the case that many elements are chopped. Here we prove a lower bound by reduction from SETEQUALITY. Given two sequences of integers $Y \in [M]^N$ and $Z \in [M]^N$, neither with any repeats, the SETEQUALITY problem is to decide whether Y and Z are equal as sets or disjoint as sets, under the promise that one of these is the case. SETEQUALITY is similar to the collision problem studied by Aaronson and Shi [4], but it lacks permutation symmetry, making it harder to prove a lower bound by

the polynomial method. By combining the collision lower bound with Ambainis's adversary method, Midrijanis [25] was nevertheless able to show that

$$\mathrm{Q}(\textsc{SetEquality}) = \Omega\left(\left(\frac{N}{\log N}\right)^{1/5}\right).$$

Very recently, and using different ideas, Zhandry [34] managed to improve Midrijanis's lower bound to the following:

Theorem 2.8 (Zhandry [34])**.** $\mathrm{Q}(\textsc{SetEquality}) = \Omega(N^{1/3})$.

Theorem 2.8 is known to be tight, by the upper bound of Brassard, Høyer, and Tapp [14] mentioned in Section 1.1.

We will consider a modification of the SETEQUALITY problem, which we call 3SETEQUALITY. Here we are given three sequences of integers $Y,Z,W \in [M]^N$, none of which has any repeats. We are promised that Y and W are disjoint as sets, and that Z is equal either to Y or to W as a set. The task is to distinguish between those two cases.

Theorem 2.9. $\mathrm{Q}(3\textsc{SetEquality}) = \Omega(N^{1/3})$.

Proof. The theorem follows from Theorem 2.8 together with the following claim: if 3SETEQUALITY is solvable by a quantum algorithm $\mathcal{A}$ that uses T queries, then SETEQUALITY is solvable by a quantum algorithm that uses $O(T)$ queries.

To show this, let Y,W be an instance of SETEQUALITY. We produce an instance of 3SETEQUALITY by choosing Z to be either a randomly permuted version of Y or a randomly permuted version of W. We then run the algorithm for 3SETEQUALITY on that instance. If Y and W are disjoint, then the promise of 3SETEQUALITY is satisfied and the algorithm will find whether we used Y or W to generate Z. If $Y = W$, then using Y and using W results in the same probability distribution for Z; hence no algorithm will be able to guess whether we used Y or W with probability greater than $1/2$. □

We now use Theorem 2.9 to prove another lower bound on $\mathrm{Q}(\mathcal{A}_\ell, \mathcal{A}_{\ell-1})$.

Lemma 2.10. *Suppose $\mathcal{A}_\ell$ was formed from $\mathcal{A}_{\ell-1}$ by chopping r rows. Then* $\mathrm{Q}(\mathcal{A}_\ell, \mathcal{A}_{\ell-1}) = \Omega\left(r^{1/3}\right)$.

Proof. We will show how to embed a 3SETEQUALITY instance of size r into the $\mathcal{A}_\ell$ versus $\mathcal{A}_{\ell-1}$ problem.

Let $\mathcal{A}_{\ell-1} = (a_1, \ldots, a_u)$. Also, let $i(1), \ldots, i(r) \in [u]$ be the r rows that are chopped in going from $\mathcal{A}_{\ell-1}$ to $\mathcal{A}_\ell$, let $i'(1), \ldots, i'(r) \in [u]$ be the r rows that are augmented, and let $j(1), \ldots, j(u-2r) \in [u]$ be the $u-2r$ rows that are left unchanged. Recall that, in going from $\mathcal{A}_{\ell-1}$ to $\mathcal{A}_\ell$, each row $i(k)$ (or $i'(k)$) is chopped or augmented by $P/2^{\ell'}$ elements, where $\ell' = \lceil \ell/2 \rceil$.

Now let $Y = (y_1, \ldots, y_r)$, $Z = (z_1, \ldots, z_r)$, $W = (z_1, \ldots, z_r)$ be an instance of 3SETEQUALITY. Then we construct an input $X \in [M]^N$ as follows. First, for each $k \in [r]$, set $a_{i(k)} - P/2^{\ell'}$ of the x_i equal to y_k, set $P/2^{\ell'}$ of the x_i equal to z_k and set $a'_{i(k)}$ of the the x_i equal to w_k. Next, let $w_1, w_2, \ldots \in [M]$ be a list of numbers that are guaranteed *not* to be in $Y \cup Z$. Then for each $k \in [u-2r]$, set $a_{j(k)}$ of the x_i equal to w_k.

It is easy to see that, if Y and Z are equal as sets, then X will have type $\mathcal{A}_{\ell-1}$, while if Z and W are equal as sets, then X will have type $\mathcal{A}_\ell$. So in deciding whether X belongs to $\mathcal{A}_\ell$ or $\mathcal{A}_{\ell-1}$, we also decide whether $Y = Z$ or $Z = W$. The lemma now follows from Theorem 2.9. □

2.4 Putting everything together

Let $\mathcal{C}$ be a quantum query algorithm that distinguishes $\mathcal{A}_0 = \mathcal{A}^*$ from $\mathcal{A}_{2L} = \mathcal{B}^*$, and assume $\mathcal{C}$ is optimal: that is, it makes $\mathrm{Q}(\mathcal{A}^*, \mathcal{B}^*) \leq \mathrm{Q}(f)$ queries. As mentioned earlier, we can assume that $\Pr[\mathcal{C} \text{ accepts } X]$ depends only on the type of X. Thus, let

$$p_\ell := \Pr[\mathcal{C} \text{ accepts } X \in \mathcal{A}_\ell] .$$

Then by assumption, $|p_0 - p_{2L}| \geq 1/3$. Now let $\beta_\ell := 1/(10\ell^2)$, and observe that $\sum_{\ell=1}^{\infty} \beta_\ell < 1/6$. By the triangle inequality, it follows that there exists an $\ell \in [2L]$ such that $|p_\ell - p_{\ell-1}| \geq \beta_\ell$. In other words, we get a $\mathrm{Q}(f)$-query quantum algorithm that distinguishes $\mathcal{A}_\ell$ from $\mathcal{A}_{\ell-1}$ with bias β_ℓ. By Proposition 2.4, this immediately implies

$$\mathrm{Q}(\mathcal{A}_\ell, \mathcal{A}_{\ell-1}) = O\left(\frac{\mathrm{Q}(f)}{\beta_\ell}\right) \qquad \text{or equivalently} \qquad \mathrm{Q}(f) = \Omega\left(\frac{\mathrm{Q}(\mathcal{A}_\ell, \mathcal{A}_{\ell-1})}{\ell^2}\right) .$$

Now let $d = \|\mathcal{A}_\ell - \mathcal{A}_{\ell-1}\|$, and suppose $\mathcal{A}_\ell$ was produced from $\mathcal{A}_{\ell-1}$ by chopping r rows. Then $d = rP/2^{\ell'} \leq 2rN/2^{\ell'}$ where $\ell' = \lceil \ell/2 \rceil$. Combining Lemmas 2.7 and 2.10, we find that

$$\begin{aligned} \mathrm{Q}(\mathcal{A}_\ell, \mathcal{A}_{\ell-1}) &= \Omega\left(\max\left\{\sqrt{\frac{N}{d}}, r^{1/3}\right\}\right) \\ &= \Omega\left(\sqrt{\frac{2^{\ell'}}{r}} + r^{1/3}\right) \\ &= \Omega\left(2^{\ell'/5}\right), \end{aligned}$$

since the minimum occurs when r is asymptotically $2^{3\ell'/5}$. If $\ell' \leq (\log_2 T) - 2$, then combining Lemmas 2.7 and 2.5, we also have the lower bound

$$\mathrm{Q}(\mathcal{A}_\ell, \mathcal{A}_{\ell-1}) = \Omega\left(\sqrt{\frac{N}{4N/T^c}}\right) = \Omega(\sqrt{T^c}) .$$

Hence

$$\mathrm{Q}(f) = \begin{cases} \Omega\left(\frac{\sqrt{T^c}}{\ell^2}\right) & \text{if } \ell' \leq (\log_2 T) - 2 , \\ \Omega\left(2^{\ell'/5}\right) & \text{if } \ell' > (\log_2 T) - 2 . \end{cases}$$

Let us now make the choice $c = 2/5$, so that we get a lower bound of

$$\mathrm{Q}(f) = \Omega\left(\frac{T^{1/5}}{\log^2 T}\right)$$

in either case. Hence $T = O(\mathrm{Q}(f)^5 \log^{10} \mathrm{Q}(f))$. By Lemma 2.3:

$$\begin{aligned} \mathrm{R}(f) &= O(T^{1+c} \log T) \\ &= O(T^{7/5} \log T) \\ &= O(\mathrm{Q}(f)^7 \log^{15} \mathrm{Q}(f)) . \end{aligned}$$

This completes the proof of Theorem 1.6.

3 Quantum lower bounds under the uniform distribution

In this section, we consider the problems of $\mathsf{P} \stackrel{?}{=} \mathsf{BQP}$ relative to a random oracle, and of simulating a T-query quantum algorithm on *most* inputs using $T^{O(1)}$ classical queries. We show that these problems are connected to a fundamental conjecture about influences in low-degree polynomials.

Recall Conjecture 1.7, which said that *bounded polynomials have influential variables*: that is, for every degree-d polynomial $p : \mathbb{R}^N \to \mathbb{R}$ such that $0 \leq p(X) \leq 1$ for all $X \in \{0,1\}^N$, there exists an $i \in [N]$ such that $\mathrm{Inf}_i[p] \geq (\mathrm{Var}[p]/d)^{O(1)}$, where

$$\mathrm{Inf}_i[p] := \mathop{\mathrm{E}}_{X \in \{0,1\}^N} \left[(p(X) - p(X^i))^2 \right],$$
$$\mathrm{Var}[p] := \mathop{\mathrm{E}}_{X \in \{0,1\}^N} \left[(p(X) - \mathrm{E}[p])^2 \right].$$

We will show that Conjecture 1.7 has several powerful consequences for quantum complexity theory.

As a first step, let

$$\mathrm{Inf}[p] := \sum_{i=1}^{N} \mathrm{Inf}_i[p]$$

be the *total influence* of p. Then we have the following bound, versions of which have long been known in the analysis of Boolean functions community,[8] but which we prove for completeness.

Lemma 3.1 (folklore)**.** *Let $p : \mathbb{R}^N \to \mathbb{R}$ be a degree-d real polynomial such that $0 \leq p(X) \leq 1$ for all $X \in \{0,1\}^N$. Then $\mathrm{Inf}[p] \leq d$.*

Proof. Let q be the analogue of p in the Fourier representation:

$$q(x_1, \ldots, x_N) := 1 - 2p\left(\frac{1-x_1}{2}, \ldots, \frac{1-x_N}{2} \right).$$

Clearly $\deg(q) = \deg(p) = d$ and $-1 \leq q(X) \leq 1$ for all $X \in \{1,-1\}^N$. Also, defining X^i to be $X \in \{1,-1\}^N$ with x_i negated, and

$$\mathrm{Inf}_i[q] := \frac{1}{4} \mathop{\mathrm{E}}_{X \in \{1,-1\}^N} \left[(q(X) - q(X^i))^2 \right],$$

we have $\mathrm{Inf}_i[q] = \mathrm{Inf}_i[p]$.

Note that we can express q as

$$q(X) = \sum_{S \subseteq [N] \,:\, |S| \leq d} \alpha_S \chi_S(X),$$

where $\alpha_S \in \mathbb{R}$ and $\chi_S(X) := \prod_{i \in S} x_i$ is the Fourier character corresponding to the set S. Furthermore, by Parseval's identity,

$$\sum_{|S| \leq d} \alpha_S^2 = \frac{1}{2^N} \sum_{X \in \{1,-1\}^N} q(X)^2 \leq 1.$$

[8] For example, Shi [29] proved the bound for the special case of Boolean functions, and generalizing his proof to arbitrary bounded functions is straightforward.

Now, in the Fourier representation, it is known that

$$\mathrm{Inf}_i[q] = \sum_{|S|\leq d \,:\, i\in S} \alpha_S^2 .$$

Hence

$$\mathrm{Inf}[p] = \mathrm{Inf}[q] = \sum_{i\in[N]} \sum_{|S|\leq d \,:\, i\in S} \alpha_S^2 = \sum_{|S|\leq d} \sum_{i\in S} \alpha_S^2 = \sum_{|S|\leq d} |S|\,\alpha_S^2 \leq d \sum_{|S|\leq d} \alpha_S^2 \leq d$$

as claimed. □

We also need the following lemma of Beals et al. [9].

Lemma 3.2 (Beals et al.)**.** *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 $2T$.*

3.1 Consequences of our influence conjecture

We now prove our first consequence of Conjecture 1.7: namely, that it implies the folklore Conjecture 1.5.

Theorem 3.3. *Suppose Conjecture 1.7 holds, and let $\varepsilon,\delta > 0$. Then given any quantum algorithm Q that makes T queries to a Boolean input X, there exists a deterministic classical algorithm that makes $\mathrm{poly}(T, 1/\varepsilon, 1/\delta)$ queries, and that approximates Q's acceptance probability to within an additive constant ε on a $1-\delta$ fraction of inputs.*

Proof. Let $p(X)$ be the probability that Q accepts input $X = (x_1,\ldots,x_N)$. Then Lemma 3.2 says that p is a real polynomial of degree at most $2T$. Assume Conjecture 1.7. Then for every such p, there exists an index i satisfying $\mathrm{Inf}_i[p] \geq w(\mathrm{Var}[p]/T)$, for some fixed polynomial w. Under that assumption, we give a classical algorithm C that makes $\mathrm{poly}(T, 1/\varepsilon, 1/\delta)$ queries to the x_i and that approximates $p(X)$ on most inputs X. In what follows, assume $X \in \{0,1\}^N$ is uniformly random.

```
set p_0 := p
for j := 0,1,2,...:
    if Var[p_j] ≤ ε^2 δ/2
        output E_{Y∈{0,1}^{N-j}}[p_j(Y)] as approximation for p(X) and halt
    else
        find an i ∈ [N-j] such that Inf_i[p_j] > w(ε^2 δ/2T)
        query x_i, and let p_{j+1} : R^{N-j} → R be the polynomial
            induced by the answer
```

When C halts, by assumption $\mathrm{Var}[p_j] \leq \varepsilon^2\delta/2$. By Markov's inequality, this implies

$$\Pr_{X\in\{0,1\}^{N-j}}\left[\left|p_j(X) - \mathrm{E}[p_j]\right| > \varepsilon\right] < \frac{\delta}{2}$$

meaning that *when* C halts, it succeeds with probability at least $1-\delta/2$.

On the other hand, suppose $\mathrm{Var}[p_j] > \varepsilon^2\delta/2$. Then by Conjecture 1.7, there exists an index $i^* \in [N]$ such that

$$\mathrm{Inf}_{i^*}[p_j] \geq w\left(\frac{\mathrm{Var}[p_j]}{T}\right) \geq w\left(\frac{\varepsilon^2\delta}{2T}\right).$$

Thus, suppose we query x_{i^*}. Since X is uniformly random, x_{i^*} will be 0 or 1 with equal probability, even conditioned on the results of all previous queries. So after the query, our new polynomial p_{j+1} will satisfy

$$\Pr\left[p_{j+1} = p_{j|x_{i^*}=0}\right] = \Pr\left[p_{j+1} = p_{j|x_{i^*}=1}\right] = \frac{1}{2},$$

where $p_{j|x_{i^*}=0}$ and $p_{j|x_{i^*}=1}$ are the polynomials on $N-j-1$ variables obtained from p_j by restricting x_{i^*} to 0 or 1 respectively. Therefore

$$\begin{aligned}
\mathop{\mathrm{E}}_{x_{i^*}\in\{0,1\}}[\mathrm{Inf}[p_{j+1}]] &= \frac{1}{2}\left(\mathrm{Inf}\left[p_{j|x_{i^*}=0}\right] + \mathrm{Inf}\left[p_{j|x_{i^*}=1}\right]\right) \\
&= \frac{1}{2}\left(\sum_{i\neq i^*}\mathrm{Inf}_i\left[p_{j|x_{i^*}=0}\right] + \sum_{i\neq i^*}\mathrm{Inf}_i\left[p_{j|x_{i^*}=1}\right]\right) \\
&= \sum_{i\neq i^*}\mathrm{Inf}_i[p_j] \\
&= \mathrm{Inf}[p_j] - \mathrm{Inf}_{i^*}[p_j] \\
&\leq \mathrm{Inf}[p_j] - w\left(\frac{\varepsilon^2\delta}{2T}\right).
\end{aligned}$$

By linearity of expectation, this implies that for all j,

$$\mathop{\mathrm{E}}_{X\in\{0,1\}^N}[\mathrm{Inf}[p_j]] \leq \mathrm{Inf}[p_0] - jw\left(\frac{\varepsilon^2\delta}{2T}\right).$$

But recall from Lemma 3.1 that

$$\mathrm{Inf}[p_0] \leq \deg(p_0) \leq 2T.$$

It follows that C halts after an expected number of iterations that is at most

$$\frac{\mathrm{Inf}[p_0]}{w(\varepsilon^2\delta/2T)} \leq \frac{2T}{w(\varepsilon^2\delta/2T)}.$$

Thus, by Markov's inequality, the probability (over X) that C has *not* halted after

$$\frac{4T}{\delta\cdot w(\varepsilon^2\delta/2T)}$$

iterations is at most $\delta/2$. Hence by the union bound, the probability over X that C fails is at most $\delta/2+\delta/2=\delta$. Since each iteration queries exactly one variable and

$$\frac{4T}{\delta\cdot w(\varepsilon^2\delta/2T)} = \mathrm{poly}(T, 1/\varepsilon, 1/\delta),$$

this completes the proof. □

An immediate corollary is the following:

Corollary 3.4. *Suppose Conjecture 1.7 holds. Then* $\mathrm{D}_{\varepsilon+\delta}(f) \leq (\mathrm{Q}_\varepsilon(f)/\delta)^{O(1)}$ *for all Boolean functions* f *and all* $\varepsilon, \delta > 0$.

Proof. Let Q be a quantum algorithm that evaluates $f(X)$, with bounded error, on a $1-\varepsilon$ fraction of inputs $X \in \{0,1\}^N$. Let $p(X) := \Pr[Q \text{ accepts } X]$. Now run the classical simulation algorithm C from Theorem 3.3, to obtain an estimate $\widetilde{p}(X)$ of $p(X)$ such that

$$\Pr_{X \in \{0,1\}^N}\left[|\widetilde{p}(X) - p(X)| \leq \frac{1}{10}\right] \geq 1-\delta\,.$$

Output $f(X) = 1$ if $\widetilde{p}(X) \geq 1/2$ and $f(X) = 0$ otherwise. By the theorem, this requires $\mathrm{poly}(T, 1/\delta)$ queries to X, and by the union bound it successfully computes $f(X)$ on at least a $1-\varepsilon-\delta$ fraction of inputs X. □

We also get the following complexity-theoretic consequence:

Theorem 3.5. *Suppose Conjecture 1.7 holds. Then* $\mathsf{P} = \mathsf{P}^{\#\mathsf{P}}$ *implies* $\mathsf{BQP}^A \subset \mathsf{AvgP}^A$ *with probability* 1 *for a random oracle* A.

Proof. Let Q be a polynomial-time quantum Turing machine that queries an oracle A, and assume Q decides some language $L \in \mathsf{BQP}^A$ with bounded error. Given an input $x \in \{0,1\}^n$, let $p_x(A) := \Pr\left[Q^A(x) \text{ accepts}\right]$. Then clearly $p_x(A)$ depends only on some finite prefix B of A, of size $N = 2^{\mathrm{poly}(n)}$. Furthermore, Lemma 3.2 implies that p_x is a polynomial in the bits of B, of degree at most $\mathrm{poly}(n)$.

Assume Conjecture 1.7 as well as $\mathsf{P} = \mathsf{P}^{\#\mathsf{P}}$. Then we claim that there exists a deterministic polynomial-time algorithm C such that for all Q and $x \in \{0,1\}^n$,

$$\Pr_A\left[|\widetilde{p}_x(A) - p_x(A)| > \frac{1}{10}\right] < \frac{1}{n^3}, \tag{3.1}$$

where $\widetilde{p}_x(A)$ is the output of C given input x and oracle A. This C is essentially just the algorithm from Theorem 3.3. The key point is that we can implement C using not only $\mathrm{poly}(n)$ queries to A, but also $\mathrm{poly}(n)$ computation steps.

To prove the claim, let M be any of the $2^{\mathrm{poly}(n)}$ monomials in the polynomial p_j from Theorem 3.3, and let α_M be the coefficient of M. Then notice that α_M can be computed to $\mathrm{poly}(n)$ bits of precision in $\mathsf{P}^{\#\mathsf{P}}$, by the same techniques used to show $\mathsf{BQP} \subseteq \mathsf{P}^{\#\mathsf{P}}$ [12]. Therefore the expectation

$$\mathop{\mathrm{E}}_{Y \in \{0,1\}^{N-j}}[p_j(Y)] = \sum_M \frac{\alpha_M}{2^{|M|}}$$

can be computed in $\mathsf{P}^{\#\mathsf{P}}$ as well. The other two quantities that arise in the algorithm—$\mathrm{Var}[p_j]$ and $\mathrm{Inf}_i[p_j]$—can also be computed in $\mathsf{P}^{\#\mathsf{P}}$, since they are simply sums of squares of differences of the values $p_j(X)$. This means that finding an i such that $\mathrm{Inf}_i[p_j] > w(\varepsilon^2\delta/T)$ is in $\mathsf{NP}^{\#\mathsf{P}}$. But under the assumption that $\mathsf{P} = \mathsf{P}^{\#\mathsf{P}}$, we have $\mathsf{P} = \mathsf{NP}^{\#\mathsf{P}}$ as well. Therefore all of the computations needed to implement C take polynomial time.

Now let $\delta_n(A)$ be the fraction of inputs $x \in \{0,1\}^n$ such that $|\widetilde{p}_x(A) - p_x(A)| > 1/10$. Then by (3.1) together with Markov's inequality,

$$\Pr_A\left[\delta_n(A) > \frac{1}{n}\right] < \frac{1}{n^2}.$$

Since $\sum_{n=1}^{\infty} 1/n^2$ converges, it follows that $\delta_n(A) \leq 1/n$ for all but finitely many values of n, with probability 1 over A. Assuming this occurs, we can simply hardwire the behavior of Q on the remaining values n into our classical simulation procedure C. Hence $L \in \mathsf{AvgP}^A$.

Since the number of BQP^A languages is countable, the above implies that $L \in \mathsf{AvgP}^A$ for every $L \in \mathsf{BQP}^A$ *simultaneously* (that is, $\mathsf{BQP}^A \subset \mathsf{AvgP}^A$) with probability 1 over A. □

As a side note, suppose we had an extremely strong variant of Conjecture 1.7, one that implied something like

$$\Pr_A\left[|\widetilde{p}_x(A) - p_x(A)| > \frac{1}{10}\right] < \frac{1}{\exp(n)}.$$

in place of (3.1). Then we could eliminate the need for AvgP in Theorem 3.5, and show that $\mathsf{P} = \mathsf{P}^{\#\mathsf{P}}$ implies $\mathsf{P}^A = \mathsf{BQP}^A$ with probability 1 for a random oracle A.

3.2 Unconditional results

We conclude this section with some unconditional results. These results will use Theorem 1.10 of Dinur et al. [19]: that for every degree-d polynomial $p : \mathbb{R}^N \to \mathbb{R}$ such that $0 \leq p(X) \leq 1$ for all $X \in \{0,1\}^N$, there exists a polynomial $\widetilde{p}$ depending on at most $2^{O(d)}/\varepsilon^2$ variables such that $\|\widetilde{p} - p\|_2^2 \leq \varepsilon$, where

$$\|p\|_2^2 := \mathop{\mathrm{E}}_{X \in \{0,1\}^N}\left[p(X)^2\right].$$

Theorem 1.10 has the following simple corollary.

Corollary 3.6. *Suppose a quantum algorithm Q makes T queries to a Boolean input $X \in \{0,1\}^N$. Then for all $\alpha, \delta > 0$, we can approximate Q's acceptance probability to within an additive constant α, on a $1-\delta$ fraction of inputs, by making $2^{O(T)}/(\alpha^4\delta^4)$ deterministic classical queries to X. (Indeed, the classical queries are nonadaptive.)*

Proof. Let $p(X) := \Pr[Q \text{ accepts } X]$. Then p is a degree-$2T$ real polynomial by Lemma 3.2. Hence, by Theorem 1.10, there exists a polynomial $\widetilde{p}$, depending on $K = 2^{O(T)}/(\alpha^4\delta^4)$ variables $x_{i_1}, \ldots, x_{i_K}$, such that

$$\mathop{\mathrm{E}}_{X \in \{0,1\}^N}\left[(\widetilde{p}(X) - p(X))^2\right] \leq \alpha^2\delta^2.$$

By the Cauchy-Schwarz inequality, then,

$$\mathop{\mathrm{E}}_{X \in \{0,1\}^N}\left[|\widetilde{p}(X) - p(X)|\right] \leq \alpha\delta,$$

so by Markov's inequality

$$\Pr_{X \in \{0,1\}^N}\left[|\widetilde{p}(X) - p(X)| > \alpha\right] < \delta.$$

Thus, our algorithm is simply to query $x_{i_1}, \ldots, x_{i_K}$, and then output $\widetilde{p}(X)$ as our estimate for $p(X)$. □

Likewise:

Corollary 3.7. $\mathrm{D}_{\varepsilon+\delta}(f) \leq 2^{O(\mathrm{Q}_\varepsilon(f))}/\delta^4$ *for all Boolean functions* f *and all* $\varepsilon, \delta > 0$.

Proof. Set α to any constant less than $1/6$, then use the algorithm of Corollary 3.6 to simulate the ε-approximate quantum algorithm for f. Output $f(X) = 1$ if $\widetilde{p}(X) \geq 1/2$ and $f(X) = 0$ otherwise. □

Given an oracle A, let $\mathsf{BQP}^{A[\log]}$ be the class of languages decidable by a BQP machine able to make $O(\log n)$ queries to A. Also, let $\mathsf{AvgP}^A_{||}$ be the class of languages decidable, with probability $1 - o(1)$ over $x \in \{0,1\}^n$, by a P machine able to make $\mathrm{poly}(n)$ parallel (nonadaptive) queries to A. Then we get the following unconditional variant of Theorem 3.5.

Theorem 3.8. *Suppose* $\mathsf{P} = \mathsf{P}^{\#\mathsf{P}}$. *Then* $\mathsf{BQP}^{A[\log]} \subset \mathsf{AvgP}^A_{||}$ *with probability* 1 *for a random oracle* A.

Proof. The proof is essentially the same as that of Theorem 3.5, except that we use Corollary 3.6 in place of Conjecture 1.7. In the proof of Corollary 3.6, observe that the condition

$$\mathop{\mathrm{E}}_{X \in \{0,1\}^N} [|\widetilde{p}(X) - p(X)|] \leq \alpha\delta$$

implies

$$\mathop{\mathrm{E}}_{X \in \{0,1\}^N} \left[\left|p_\mu(X) - p(X)\right|\right] \leq \alpha\delta \tag{3.2}$$

as well, where $p_\mu(X)$ equals the mean of $p(Y)$ over all inputs Y that agree with X on $x_{i_1}, \ldots, x_{i_K}$. Thus, given a quantum algorithm that makes T queries to an oracle string, the computational problem that we need to solve boils down to finding a subset of the oracle bits $x_{i_1}, \ldots, x_{i_K}$ such that $K = 2^{O(T)}/(\alpha^4\delta^4)$ and (3.2) holds. Just like in Theorem 3.5, this problem is solvable in the counting hierarchy $\mathsf{CH} = \mathsf{P}^{\#\mathsf{P}} \cup \mathsf{P}^{\#\mathsf{P}^{\#\mathsf{P}}} \cup \cdots$. So if we assume $\mathsf{P} = \mathsf{P}^{\#\mathsf{P}}$, then it is also solvable in P.

In Theorem 3.5, the conclusion we got was $\mathsf{BQP}^A \subset \mathsf{AvgP}^A$ with probability 1 for a random oracle A. In our case, the number of classical queries K is exponential (rather than polynomial) in the number of quantum queries T, so we only get $\mathsf{BQP}^{A[\log]} \subset \mathsf{AvgP}^A$. On the other hand, since the classical queries are nonadaptive, we can strengthen the conclusion to $\mathsf{BQP}^{A[\log]} \subset \mathsf{AvgP}^A_{||}$. □

4 Open problems

It would be nice to improve the $\mathrm{R}(f) = O(\mathrm{Q}(f)^7 \,\mathrm{polylog}\, \mathrm{Q}(f))$ bound for all symmetric problems. As mentioned earlier, we conjecture that the right answer is $\mathrm{R}(f) = O(\mathrm{Q}(f)^2)$. In trying to improve our lower bound, it seems best to avoid the use of SETEQUALITY. After all, it is a curious feature of our proof that, to get a lower bound for symmetric problems, we need to reduce from the *non*-symmetric SETEQUALITY problem!

Another problem is to remove the assumption $M \geq N$ in our lower bound for symmetric problems. Experience with related problems strongly suggests that this can be done, but one might need to replace our chopping procedure by something different.

We also conjecture that $\mathrm{R}(f) \leq \mathrm{Q}(f)^{O(1)}$ for all partial functions f that are symmetric *only* under permuting the inputs (and not necessarily the outputs). Proving this seems to require a new approach.

Another problem, in a similar spirit, is whether $\mathrm{R}(f) \leq \mathrm{Q}(f)^{O(1)}$ for all partial functions $f : S \to \{0,1\}$ such that S (i. e., the promise on inputs) is symmetric, but f itself need not be symmetric.

It would be interesting to reprove the $\mathrm{R}(f) \leq \mathrm{Q}(f)^{O(1)}$ bound using only the polynomial method, and not the adversary method. Or, to rephrase this as a purely classical question: for all $X = (x_1, \ldots, x_N)$ in $[M]^N$, let B_X be the $N \times M$ matrix whose $(i,j)^{th}$ entry is 1 if $x_i = j$ and 0 otherwise. Then given a set $S \subseteq [M]^N$ and a function $f : S \to \{0,1\}$, let $\widetilde{\deg}(f)$ be the minimum degree of a real polynomial $p : \mathbb{R}^{MN} \to \mathbb{R}$ such that

(i) $0 \leq p(B_X) \leq 1$ for all $X \in [M]^N$, and

(ii) $|p(B_X) - f(X)| \leq \frac{1}{3}$ for all $X \in S$.

Then is it the case that $\mathrm{R}(f) \leq \widetilde{\deg}(f)^{O(1)}$ for all permutation-invariant functions f?

On the random oracle side, the obvious problem is to prove Conjecture 1.7—thereby establishing that $\mathrm{D}_\varepsilon(f)$ and $\mathrm{Q}_\delta(f)$ are polynomially related, and all the other consequences shown in Section 3. Alternatively, one could look for some technique that was tailored to polynomials p that arise as the acceptance probabilities of quantum algorithms. In this way, one could conceivably solve $\mathrm{D}_\varepsilon(f)$ versus $\mathrm{Q}_\delta(f)$ and the other quantum problems, without settling the general conjecture about bounded polynomials.

5 Appendix: The Boolean case

Given a partial Boolean function $f : \{0,1\}^N \to \{0,1,*\}$, call f *symmetric* if $f(X)$ depends only on the Hamming weight $|X| := x_1 + \cdots + x_N$. For completeness, in this appendix we prove the following basic fact:

Theorem 5.1. $\mathrm{R}(f) = O(\mathrm{Q}(f)^2)$ *for every partial symmetric Boolean function* f.

For *total* symmetric Boolean functions, Theorem 5.1 was already shown by Beals et al. [9], using an approximation theory result of Paturi [28]. Indeed, in the total case one even has $\mathrm{D}(f) = O(\mathrm{Q}(f)^2)$. So the new twist is just that f can be partial.

Abusing notation, let $f(k) \in \{0,1,*\}$ be the value of f on all inputs of Hamming weight k (where as usual, $*$ means 'undefined'). Then we have the following quantum lower bound:

Lemma 5.2. *Suppose that* $f(a) = 0$ *and* $f(b) = 1$ *or vice versa, where* $a < b$ *and* $a \leq N/2$. *Then* $\mathrm{Q}(f) = \Omega\left(\sqrt{bN}/(b-a)\right)$.

Proof. This follows from a straightforward application of Ambainis's adversary theorem (Theorem 2.6). Specifically, let $A, B \subseteq \{0,1\}^N$ be the sets of all strings of Hamming weights a and b respectively, and for all $X \in A$ and $Y \in B$, put $(X,Y) \in R$ if and only if $X \preceq Y$ (that is, $x_i \leq y_i$ for all $i \in [N]$). Then

$$\mathrm{Q}(f) = \Omega\left(\sqrt{\frac{N-a}{b-a} \cdot \frac{b}{b-a}}\right) = \Omega\left(\frac{\sqrt{bN}}{b-a}\right).$$

Alternatively, this lemma can be proved using the approximation theory result of Paturi [28], following Beals et al. [9]. □

In particular, if we set $\beta := b/N$ and $\varepsilon := (b-a)/N$, then $\mathrm{Q}(f) = \Omega(\sqrt{\beta}/\varepsilon)$. On the other hand, we also have the following randomized *upper* bound, which follows from a Chernoff bound (similar to Lemma 2.2):

Lemma 5.3. *Assume $\beta > \varepsilon > 0$. By making $O(\beta/\varepsilon^2)$ queries to an N-bit string X, a classical sampling algorithm can estimate the fraction $\beta := |X|/N$ of 1 bits to within an additive error $\pm\varepsilon/3$, with success probability at least $2/3$.*

Thus, assume the function f is non-constant, and let

$$\gamma := \max_{f(a)=0, f(b)=1} \frac{\sqrt{bN}}{b-a}. \tag{5.1}$$

Assume without loss of generality that the maximum of (5.1) is achieved when $a < b$ and $a \leq N/2$, if necessary by applying the transformations $f(X) \to 1 - f(X)$ and $f(X) \to f(N-X)$. Now consider the following randomized algorithm to evaluate f, which makes $T := O(\gamma^2)$ queries:

```
Choose indices i_1,...,i_T ∈ [N] uniformly at random with replacement
Query x_{i_1},...,x_{i_T}
Set k := (N/T)(x_{i_1} + ··· + x_{i_T})
If there exists a b ∈ {0,...,N} such that f(b) = 1 and |k − b| ≤ sqrt(bN)/(3γ)
    output f(X) = 1
Otherwise output f(X) = 0
```

By Lemma 5.3, the above algorithm succeeds with probability at least $2/3$, provided we choose T suitably large. Hence $\mathrm{R}(f) = O(\gamma^2)$. On the other hand, Lemma 5.2 implies that $\mathrm{Q}(f) = \Omega(\gamma)$. Hence $\mathrm{R}(f) = O(\mathrm{Q}(f)^2)$, completing the proof of Theorem 5.1.

6 Appendix: 1-norm versus 2-norm

As mentioned in Section 1.3, in the original version of this paper we stated Conjecture 1.7, and all our results assuming it, in terms of L_1-influences rather than L_2-influences. Subsequently, Arturs Bačkurs discovered a gap in our L_1-based argument. In recent work, Bačkurs and Bavarian [8] managed to fill the gap, allowing our L_1-based argument to proceed. Still, the *simplest* fix for the problem Bačkurs uncovered is just to switch from L_1-influences to L_2-influences, so that is what we did in Section 3 (and in our current statement of Conjecture 1.7).

Fortunately, it turns out that the L_1 and L_2 versions of Conjecture 1.7 are *equivalent*, so making this change does not even involve changing our conjecture. For completeness, in this appendix we prove the equivalence of the L_1 and L_2 versions of Conjecture 1.7.

As usual, let $p : \{0,1\}^N \to [0,1]$ be a real polynomial, let $X \in \{0,1\}^N$, and let X^i denote X with the i^{th} bit flipped. Then the L_1-*variance* $\mathrm{Vr}[p]$ of p and the L_1-*influence* $\mathrm{Inf}_i[p]$ of the i^{th} variable x_i are

defined as follows:

$$\mathrm{Vr}[p] := \mathop{\mathrm{E}}_{X\in\{0,1\}^N}[|p(X)-\mathrm{E}[p]|],$$
$$\mathrm{Inf}_i^{\mathrm{l}}[p] := \mathop{\mathrm{E}}_{X\in\{0,1\}^N}\left[\left|p(X)-p(X^i)\right|\right].$$

The L_1 analogue of Conjecture 1.7 simply replaces $\mathrm{Var}[p]$ by $\mathrm{Vr}[p]$ and $\mathrm{Inf}_i[p]$ by $\mathrm{Inf}_i^{\mathrm{l}}[p]$:

Conjecture 6.1 (Bounded Polynomials Have Influential Variables, L_1 Version)**.** *Let $p:\mathbb{R}^N\to\mathbb{R}$ be a degree-d real polynomial such that $0\le p(X)\le 1$ for all $X\in\{0,1\}^N$. Then there exists an $i\in[N]$ such that $\mathrm{Inf}_i^{\mathrm{l}}[p]\ge(\mathrm{Vr}[p]/d)^{O(1)}$.*

We now prove the equivalence:

Proposition 6.2. *Conjectures 1.7 and 6.1 are equivalent.*

Proof. First assume Conjecture 1.7. By the Cauchy-Schwarz inequality,

$$\mathrm{Inf}_i[p] = \mathop{\mathrm{E}}_{X\in\{0,1\}^N}\left[(p(X)-p(X^i))^2\right] \ge \left(\mathop{\mathrm{E}}_{X\in\{0,1\}^N}\left[\left|p(X)-p(X^i)\right|\right]\right)^2 = \mathrm{Inf}_i^{\mathrm{l}}[p]^2.$$

Also, since $p(X)\in[0,1]$,

$$\mathrm{Vr}[p] = \mathop{\mathrm{E}}_{X\in\{0,1\}^N}[|p(X)-\mathrm{E}[p]|] \ge \mathop{\mathrm{E}}_{X\in\{0,1\}^N}\left[(p(X)-\mathrm{E}[p])^2\right] = \mathrm{Var}[p].$$

Hence there exists an $i\in[N]$ such that

$$\mathrm{Inf}_i[p]\ge\mathrm{Inf}_i^{\mathrm{l}}[p]^2\ge\left(\frac{\mathrm{Vr}[p]}{d}\right)^{O(1)}\ge\left(\frac{\mathrm{Var}[p]}{d}\right)^{O(1)}$$

and Conjecture 6.1 holds.

Likewise, assume Conjecture 6.1. Then we have $\mathrm{Inf}_i^{\mathrm{l}}[p]\ge\mathrm{Inf}_i[p]$ since $p(X)\in[0,1]$, and $\mathrm{Var}[p]\ge\mathrm{Vr}[p]^2$ by the Cauchy-Schwarz inequality. Hence there exists an $i\in[N]$ such that

$$\mathrm{Inf}_i^{\mathrm{l}}[p]\ge\mathrm{Inf}_i[p]\ge\left(\frac{\mathrm{Var}[p]}{d}\right)^{O(1)}\ge\left(\frac{\mathrm{Vr}[p]}{d}\right)^{O(1)}$$

and Conjecture 1.7 holds. □

7 Appendix: Equivalent form of Conjecture 1.5

Recall Conjecture 1.5, which said (informally) that any quantum algorithm that makes T queries to $X\in\{0,1\}^N$ can be simulated to within $\pm\varepsilon$ additive error on a $1-\delta$ fraction of the inputs X by a classical algorithm that makes $\mathrm{poly}(T,1/\varepsilon,1/\delta)$ queries. In Section 1.1, we claimed that Conjecture 1.5 was equivalent to an alternative conjecture, which we now state more formally:

Conjecture 7.1. *Let $S \subseteq \{0,1\}^N$ with $|S| \geq c2^N$, and let $f : S \to \{0,1\}$. Then there exists a deterministic classical algorithm that makes $\mathrm{poly}(\mathrm{Q}(f), 1/\alpha, 1/c)$ queries, and that computes $f(X)$ on at least a $1-\alpha$ fraction of $X \in S$.*

In this appendix, we justify the equivalence claim. We first need a simple combinatorial lemma.

Lemma 7.2. *Suppose we are trying to learn an unknown real $p \in [0,1]$. There are k "hint bits" $h_1, \ldots, h_k$, where each h_i is 0 if $(i-1)/k \leq p$ or 1 if $i/k \geq p$ (and can otherwise be arbitrary). However, at most $b < k/2$ of the h_i are then corrupted by an adversary, producing the new string $h'_1, \ldots, h'_k$. Using $h'_1, \ldots, h'_k$, one can still determine p to within additive error $\pm(b+1)/k$.*

Proof. Given the string $h' = (h'_1, \ldots, h'_k)$, we apply the following correction procedure: we repeatedly search for pairs $i < j$ such that $h'_i = 1$ and $h'_j = 0$, and "delete" those pairs (that is, we set $h'_i = h'_j = *$, where $*$ means "unknown"). We continue for t steps, until no more such pairs exist. Next, we delete the rightmost $b-t$ zeroes in h' (replacing them with $*$s), and likewise delete the leftmost $b-t$ ones. Finally, as our estimate for p, we output

$$q := \frac{i^* + j^* - 1}{2k},$$

where i^* is the index of the rightmost 0 remaining in h' (or $i^* = 0$ if no 0's 0s remain), and j^* is the index of the leftmost 1 remaining (or $j^* = k+1$ if no 1s remain).

To show correctness: every time we find an $i < j$ pair such that $h'_i = 1$ and $h'_j = 0$, at least one of h'_i and h'_j must have been corrupted by the adversary. It follows that $t \leq b$, where t is the number of deleted pairs. Furthermore, after the first stage finishes, every 1 is to the right of every 0, at most $b-t$ of the remaining bits are corrupted, and the bits that *are* corrupted must be among the rightmost zeroes or the leftmost ones (or both). Hence, after the second stage finishes, every $h'_i = 0$ reliably indicates that $p \geq (i-1)/k$, and every $h'_j = 1$ reliably indicates that $p \leq j/k$. Moreover, since only $2b$ bits are deleted in total, we must have $j^* - i^* \leq 2b+1$, where i^* and j^* are as defined above. It follows that

$$|p-q| \leq \frac{j^* - i^* + 1}{2k} \leq \frac{b+1}{k}. \qquad \square$$

Theorem 7.3. *Conjectures 1.5 and 7.1 are equivalent.*

Proof. We start with the easy direction, that Conjecture 1.5 implies Conjecture 7.1. Given $f : S \to \{0,1\}$ with $|S| \geq c2^N$, let Q be a quantum algorithm that evaluates f with error probability at most $1/3$ using T queries. Let $p(X)$ be Q's acceptance probability on a given input $X \in \{0,1\}^N$ (not necessarily in S). Then by Conjecture 1.5, there exists a deterministic classical algorithm that approximates $p(X)$ to within additive error $\pm\varepsilon$ on a $1-\delta$ fraction of $X \in \{0,1\}^N$ using $\mathrm{poly}(T, 1/\varepsilon, 1/\delta)$ queries. If we set (say) $\varepsilon := 1/7$ and $\delta := \alpha c$, then such an approximation lets us decide whether $f(X) = 0$ or $f(X) = 1$ for a $1-\alpha$ fraction of $X \in S$, using $\mathrm{poly}(T, 1/\alpha, 1/c)$ queries.

We now show the other direction, that Conjecture 7.1 implies Conjecture 1.5. Let Q be a T-query quantum algorithm, let $p(X)$ be Q's acceptance probability on input X, and suppose we want to approximate $p(X)$ to within error $\pm\varepsilon$ on at least a $1-\delta$ fraction of $X \in \{0,1\}^N$. Let $\varepsilon := \varepsilon/3$. Assume for simplicity that ε has the form $1/k$ for some positive integer k; this will have no effect on the asymptotics. For each $j \in [k]$, let

$$S_j := \left\{ X : p(X) \leq \frac{j-1}{k} \text{ or } p(X) \geq \frac{j}{k} \right\},$$

and define the function $f_j : S_j \to \{0,1\}$ by

$$f_j(X) := \begin{cases} 0 & \text{if } p(X) \leq (j-1)/k, \\ 1 & \text{if } p(X) \geq j/k. \end{cases}$$

By Proposition 2.4, we have $\mathrm{Q}(f_j) = O(kT)$ for all $j \in [k]$. Also, note that

$$\mathop{\mathrm{E}}_{j}\left[\left|S_j\right|\right] \geq \left(1 - \frac{1}{k}\right) 2^n.$$

By Markov's inequality, this implies that there can be at most one $j \in [k]$ (call it j^*) such that $\left|S_j\right| < 2^{n-2}$. Likewise, note that for every $X \in \{0,1\}^N$, there is at most one $j \in [k]$ such that $X \notin S_j$.

Together with Conjecture 7.1, the above facts imply that, for all $j \neq j^*$ and $\alpha > 0$, there exists a deterministic classical algorithm $A_{j,\alpha}$, making $\mathrm{poly}(T, 1/\alpha)$ queries, that computes $f_j(X)$ on at least a $1-\alpha$ fraction of all $X \in S_j$. Suppose we run $A_{j,\alpha}$ for all $j \neq j^*$. Then by the union bound, for at least a $1-k\alpha$ fraction of $X \in \{0,1\}^N$, there can be at most two $j \in [k]$ such that $A_{j,\alpha}$ fails to compute $f_j(X)$: namely, j^*, and the unique j (call it j') such that $X \notin S_{j'}$. Thus, suppose $A_{j,\alpha}$ succeeds for all $j \notin \{j^*, j'\}$. By Lemma 7.2, this implies that $p(X)$ has been determined up to an additive error of $\pm 3\varepsilon = \pm\varepsilon$. Hence, we simply need to set $\alpha := \delta/k$, in order to get a classical algorithm that makes $k \cdot \mathrm{poly}(T, k/\delta) = \mathrm{poly}(T, 1/\varepsilon, 1/\delta)$ queries, and that approximates $p(X)$ up to additive error $\pm\varepsilon$ for at least a $1-\delta$ fraction of $X \in \{0,1\}^N$. □

8 Acknowledgments

We thank Aleksandrs Belovs, Andy Drucker, Ryan O'Donnell, and Ronald de Wolf for helpful discussions; Mark Zhandry for taking up our challenge to improve the lower bound on $\mathrm{Q}(\textsc{SetEquality})$ to the optimal $\Omega(N^{1/3})$; and Dana Moshkovitz for suggesting a proof of Lemma 7.2. We especially thank Artūrs Bačkurs, Jānis Iraids, the attendees of the quantum computing reading group at the University of Latvia, and the anonymous reviewers for their feedback, and for catching some errors in earlier versions of this paper.

References

[1] SCOTT AARONSON: Quantum lower bound for the collision problem. In *Proc. 34th STOC*, pp. 635–642. ACM Press, 2002. [doi:10.1145/509907.509999] 136, 163

[2] SCOTT AARONSON: BQP and the polynomial hierarchy. In *Proc. 42nd STOC*, pp. 141–150. ACM Press, 2010. See expanded version in ECCC 2009. [doi:10.1145/1806689.1806711] 135, 140

[3] SCOTT AARONSON AND ANDRIS AMBAINIS: The need for structure in quantum speedups. In *Proc. 2nd Innovations in Computer Science Conference (ICS'11)*, pp. 338–352, 2011. (Tsinghua). [arXiv:0911.0996] 138, 140

[4] SCOTT AARONSON AND YAOYUN SHI: Quantum lower bounds for the collision and the element distinctness problems. *J. ACM*, 51(4):595–605, 2004. Preliminary versions by Aaronson in STOC'02 (item [1]) and by Shi in FOCS'02. [doi:10.1145/1008731.1008735] 136, 138, 149

[5] DORIT AHARONOV, VAUGHAN JONES, AND ZEPH LANDAU: A polynomial quantum algorithm for approximating the jones polynomial. *Algorithmica*, 55(3):395–421, 2009. Preliminary version in STOC'06. [doi:10.1007/s00453-008-9168-0] 134

[6] ANDRIS AMBAINIS: Quantum lower bounds by quantum arguments. *J. Comput. Syst. Sci.*, 64(4):750–767, 2002. Preliminary version in STOC'00. [doi:10.1006/jcss.2002.1826] 138, 147, 148

[7] ANDRIS AMBAINIS AND RONALD DE WOLF: Average-case quantum query complexity. *Journal of Physics A: Mathematical and General*, 34(35):6741, 2001. Preliminary version in STACS'00. [doi:10.1088/0305-4470/34/35/302, arXiv:quant-ph/9904079] 137

[8] ARTŪRS BAČKURS AND MOHAMMAD BAVARIAN: On the sum of L_1 influences. In *Proc. 29th IEEE Conf. on Computational Complexity (CCC'14)*. IEEE Comp. Soc. Press, 2014. ECCC 2014. [doi:10.1109/CCC.2014.21, arXiv:1302.4625] 140, 159

[9] ROBERT BEALS, HARRY BUHRMAN, RICHARD CLEVE, MICHELE MOSCA, AND RONALD DE WOLF: Quantum lower bounds by polynomials. *J. ACM*, 48(4):778–797, 2001. Preliminary version in FOCS'98. [doi:10.1145/502090.502097] 134, 135, 137, 139, 153, 158

[10] JONATHAN NIEL DE BEAUDRAP, RICHARD CLEVE, AND JOHN WATROUS: Sharp quantum versus classical query complexity separations. *Algorithmica*, 34(4):449–461, 2002. [doi:10.1007/s00453-002-0978-1] 135

[11] CHARLES H. BENNETT, ETHAN BERNSTEIN, GILLES BRASSARD, AND UMESH V. VAZIRANI: Strengths and weaknesses of quantum computing. *SIAM J. Comput.*, 26(5):1510–1523, 1997. [doi:10.1137/S0097539796300933] 134

[12] ETHAN BERNSTEIN AND UMESH V. VAZIRANI: Quantum complexity theory. *SIAM J. Comput.*, 26(5):1411–1473, 1997. Preliminary version in STOC'93. [doi:10.1137/S0097539796300921] 137, 155

[13] GILLES BRASSARD, PETER HØYER, MICHELE MOSCA, AND ALAIN TAPP: Quantum amplitude amplification and estimation. In S. J. LOMONACO AND H. E. BRANDT, editors, *Quantum Computation and Information*, volume 305 of *Contemporary Mathematics Series*. Amer. Math. Soc., 2002. [doi:10.1090/conm/305, arXiv:quant-ph/0005055] 144

[14] GILLES BRASSARD, PETER HØYER, AND ALAIN TAPP: Quantum cryptanalysis of hash and claw free functions. *ACM SIGACT News*, 28(2):14–19, 1997. [doi:10.1145/261342.261346, arXiv:quant-ph/9705002] 136, 150

[15] HARRY BUHRMAN AND RONALD DE WOLF: Complexity measures and decision tree complexity: a survey. *Theor. Comput. Sci.*, 288(1):21–43, 2002. [doi:10.1016/S0304-3975(01)00144-X] 135

[16] HARRY BUHRMAN, LANCE FORTNOW, ILAN NEWMAN, AND HEIN RÖHRIG: Quantum property testing. *SIAM J. Comput.*, 37(5):1387–1400, 2008. Preliminary version in SODA'03. [doi:10.1137/S0097539704442416] 137

[17] WIM VAN DAM, SEAN HALLGREN, AND LAWRENCE IP: Quantum algorithms for some hidden shift problems. *SIAM J. Comput.*, 36(3):763–778, 2006. Preliminary version in SODA'03. [doi:10.1137/S009753970343141X] 134

[18] DAVID DEUTSCH AND RICHARD JOZSA: Rapid solution of problems by quantum computation. *Proc. Roy. Soc. A*, 439(1907):553–558, 1992. [doi:10.1098/rspa.1992.0167] 136

[19] IRIT DINUR, EHUD FRIEDGUT, GUY KINDLER, AND RYAN O'DONNELL: On the Fourier tails of bounded functions over the discrete cube. *Israel J. Math.*, 160(1):389–412, 2007. Preliminary version in STOC'06. [doi:10.1007/s11856-007-0068-9] 139, 156

[20] LANCE FORTNOW AND JOHN D. ROGERS: Complexity limitations on quantum computation. *J. Comput. Sys. Sci.*, 59(2):240–252, 1999. Preliminary version in CCC'98. [doi:10.1006/jcss.1999.1651, arXiv:cs/9811023] 137

[21] LOV K. GROVER: A fast quantum mechanical algorithm for database search. In *Proc. 28th STOC*, pp. 212–219. ACM Press, 1996. [doi:10.1145/237814.237866, arXiv:quant-ph/9605043] 134

[22] ARAM HARROW, AVINATAN HASSIDIM, AND SETH LLOYD: Quantum algorithm for solving linear systems of equations. *Phys. Rev. Lett.*, 15, 2009. [doi:10.1103/PhysRevLett.103.150502, arXiv:0811.3171] 134

[23] EDITH HEMASPAANDRA, LANE A. HEMASPAANDRA, AND MARIUS ZIMAND: Almost-everywhere superiority for quantum polynomial time. *Inform. and Comput.*, 175(2):171–181, 2002. [doi:10.1006/inco.2001.3110, arXiv:quant-ph/9910033] 137

[24] JEFF KAHN, MICHAEL E. SAKS, AND CLIFFORD D. SMYTH: A dual version of Reimer's inequality and a proof of Rudich's conjecture. In *Proc. 15th IEEE Conf. on Computational Complexity (CCC'00)*, pp. 98–103. IEEE Comp. Soc. Press, 2000. [doi:10.1109/CCC.2000.856739] 138

[25] GATIS MIDRIJĀNIS: A polynomial quantum query lower bound for the set equality problem. In *Proc. 31th Internat. Colloq. on Automata, Languages and Programming (ICALP'04)*, pp. 996–1005. Springer, 2004. [doi:10.1007/978-3-540-27836-8_83] 138, 150

[26] ASHLEY MONTANARO: Some applications of hypercontractive inequalities in quantum information theory. *J. Math. Phys.*, 53(12), 2012. [doi:10.1063/1.4769269, arXiv:1208.0161] 140

[27] RYAN O'DONNELL, MICHAEL E. SAKS, ODED SCHRAMM, AND ROCCO A. SERVEDIO: Every decision tree has an influential variable. In *Proc. 46th FOCS*, pp. 31–39. IEEE Comp. Soc. Press, 2005. [doi:10.1109/SFCS.2005.34, arXiv:cs/0508071] 139

[28] RYAN O'DONNELL AND ROCCO A. SERVEDIO: New degree bounds for polynomial threshold functions. *Combinatorica*, 30(3):327–358, 2010. Preliminary version in STOC'03. [doi:10.1007/s00493-010-2173-3] 158

[29] YAOYUN SHI: Lower bounds of quantum black-box complexity and degree of approximating polynomials by influence of Boolean variables. *Inform. Proc. Lett.*, 75(1-2):79–83, 2000. [doi:10.1016/S0020-0190(00)00069-7, arXiv:quant-ph/9904107] 152

[30] PETER W. SHOR: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509, 1997. Preliminary version in FOCS'94. [doi:10.1137/S0097539795293172] 134, 135

[31] DANIEL R. SIMON: On the power of quantum computation. *SIAM J. Comput.*, 26(5):1474–1483, 1997. Preliminary version at FOCS'94. [doi:10.1137/S0097539796298637] 135

[32] CLIFFORD D. SMYTH: Reimer's inequality and Tardos' conjecture. In *Proc. 34th STOC*, pp. 218–221. ACM Press, 2002. [doi:10.1145/509907.509942] 137, 138

[33] HENRY YUEN: A quantum lower bound for distinguishing random functions from random permutations. *Quantum Information & Computation*, 14(13 & 14), 2014. QIC-online. [arXiv:1310.2885] 140

[34] MARK ZHANDRY: A note on the quantum collision and set equality problems. 2013. [arXiv:1312.1027] 138, 140, 150

AUTHORS

Scott Aaronson
associate professor
Massachusetts Institute of Technology
Cambridge, MA
aaronson@csail.mit.edu
http://www.scottaaronson.com

Andris Ambainis
professor
University of Latvia and
IAS, Princeton
ambainis@lu.lv

ABOUT THE AUTHORS

SCOTT AARONSON received his bachelor's degree from Cornell University and his Ph. D. from UC Berkeley. He is known for his blog and for founding the Complexity Zoo. He publishes often in *Theory of Computing*.

ANDRIS AMBAINIS received his Ph. D. from UC Berkeley and is now Professor at the University of Latvia. His research interests include many areas of quantum computing and quantum information theory (quantum algorithms, quantum complexity theory, quantum cryptography, pseudorandom quantum states, etc.), as well as the classical theory of computation.