Pseudorandom generators and the BQP vs. PH problem

Bill Fefferman* Chris Umans†

December 21, 2010

Abstract

It is a longstanding open problem to devise an oracle relative to which BQP does not lie in the Polynomial-Time Hierarchy (PH). We advance a natural conjecture about the capacity of the Nisan-Wigderson pseudorandom generator [NW94] to fool AC_0, with MAJORITY as its hard function. Our conjecture is essentially that the loss due to the hybrid argument (which is a component of the standard proof from [NW94]) can be avoided in this setting. This is a question that has been asked previously in the pseudorandomness literature [BSW03]. We then make three main contributions:

1. We show that our conjecture implies the existence of an oracle relative to which BQP is not in the PH. This entails giving an explicit construction of unitary matrices, realizable by small quantum circuits, whose row-supports are "nearly-disjoint."
2. We give a simple framework (generalizing the setting of Aaronson [Aar10b]) in which any efficiently quantumly computable unitary gives rise to a distribution that can be distinguished from the uniform distribution by an efficient quantum algorithm. When applied to the unitaries we construct, this framework yields a problem that can be solved quantumly, and which forms the basis for the desired oracle.
3. We prove that Aaronson's "GLN conjecture" [Aar10b] implies our conjecture; our conjecture is thus formally easier to prove. The GLN conjecture was recently proved false for depth greater than 2 [Aar10a], but it remains open for depth 2. If true, the depth-2 version of either conjecture would imply an oracle relative to which BQP is not in AM, which is itself an outstanding open problem.

Taken together, our results have the following interesting interpretation: they give an instantiation of the Nisan-Wigderson generator that can be broken by quantum computers, but not by the relevant modes of classical computation, if our conjecture is true.

*Department of Computing and Mathematical Sciences, Caltech, Pasadena, CA 91125. Supported by IQI.

†Department of Computing and Mathematical Sciences, Caltech, Pasadena, CA 91125. Supported by NSF CCF-0846991.

1 Introduction

Let U_t denote a random variable uniformly distributed on t-bit strings. A *pseudorandom generator* (PRG) is a function

$$f : \{0,1\}^t \to \{0,1\}^m$$

that stretches a short "seed" into a longer output string, with the property that $f(U_t)$ is *computationally indistinguishable* from the uniform distribution.

There is a vast literature constructing PRGs that achieve computational indistinguishability against a wide variety of computational models (e.g. small circuits, small nondeterministic circuits, small branching programs, small constant-depth circuits). These constructions are typically "hardness vs. randomness" tradeoffs in the sense that they make use of a hard function (either unconditionally hard, or hard conditioned on a complexity assumption), and their proof of correctness takes the form of a reduction that transforms a computationally efficient *distinguisher* into an efficient algorithm for the hard function (thereby deriving a contradiction). This transformation entails the use of the *hybrid argument* [GM84, Yao82] which incurs a loss of a factor $1/m$ in going from a distinguisher (with gap ε) to a *predictor* (with advantage ε/m) and from there to an efficient algorithm (with advantage ε/m).

A question that has been raised in the pseudorandomness literature is whether this loss of a factor of $1/m$ can be avoided (for an explicit framing of this question, and a discussion of its motivation, see [BSW03]). In certain settings, the answer is known to be "yes" – when the notion of "efficient" is small PH circuits, or bounded-width branching programs [BSW03]. In the present paper, we identify a setting in which this question has surprising connections to a central unresolved question in quantum complexity: whether there exists an oracle relative to which BQP is not in the PH.

Our setting is a familiar one: we will work with the ubiquitous Nisan-Wigderson PRG [NW94], against AC_0 circuits, with MAJORITY as its hard function. We need a precise statement for the discussion below, which can be given via two standard definitions:

Definition 1.1 ([NW94])**.** *A set family $\mathcal{D} = \{S_1, S_2, \ldots, S_m\}$ is an (ℓ, p) design if every set in the family has cardinality ℓ, and for all $i \neq j$, $|S_i \cap S_j| \leqslant p$.*

Definition 1.2 ([NW94])**.** *Given a function $f : \{0,1\}^\ell \to \{0,1\}$ and an (ℓ, p) design $\mathcal{D} = \{S_1, S_2, \ldots, S_m\}$ in a universe of size t, the function $NW^f_{\mathcal{D}} : \{0,1\}^t \to \{0,1\}^m$ is given by*

$$NW^f_{\mathcal{D}}(x) = \left(f_1(x_{|S_1}), f_2(x_{|S_2}), f_3(x_{|S_3}), \ldots, f_m(x_{|S_m})\right),$$

where each f_i is the function f with a fixed set of its inputs negated[1]*, and $x_{|S}$ denotes the projection of x to the coordinates in the set S.*

Generally speaking, the function $NW^f_{\mathcal{D}}$ is a PRG against a class of distinguishers as long as f is hard on average for that class of distinguishers. Recall that the majority function on ℓ bits is known to be hard for AC_0: no polynomial-size (or even quasi-polynomial-size), constant-depth circuit can compute majority correctly on more than a $1/2 + \widetilde{O}(1/\sqrt{\ell})$ fraction of the inputs [Smo93, Hås87], and this is essentially tight, since the function that simply outputs the first bit of the input is correct on a random input with probability $1/2 + \Theta(1/\sqrt{\ell})$. We make the following quantitative conjecture:

[1] The standard setup has each $f_i = f$; we need the additional freedom in this paper for technical reasons. We know of no settings in which this alteration affects the analysis of the NW generator.

Conjecture 1. *Let $\mathcal{D} = \{S_1, S_2, \ldots, S_m\}$ be an $(\ell, O(1))$-design in a universe of size $t \leqslant \mathrm{poly}(\ell)$, with $m \leqslant \mathrm{poly}(\ell)$. Then for every constant-depth circuit of size at most $\exp(\mathrm{poly}\log m)$,*

$$|\Pr[C(U_{t+m}) = 1] - \Pr[C(U_t, NW_{\mathcal{D}}^{\text{MAJORITY}}(U_t)) = 1]| \leqslant o(1).$$

In this work we abuse notation and refer to constant depth circuits of size at most $\exp(\mathrm{poly}\log m)$ as "AC_0."

By the standard argument from [NW94, Nis92], a distinguishing circuit C with gap ε can be converted to a *predictor* with advantage ε/m and then a slightly larger circuit that computes MAJORITY with success rate $1/2 + \varepsilon/m$. Thus the above statement is true for $m \ll \sqrt{\ell}$; if the $1/m$ loss from the hybrid argument can be avoided (or reduced), it would be true for m as large as $\mathrm{poly}(\ell)$ (and even larger) as we conjecture is true. In Section 6 we discuss intuition supporting this conjecture that relates specifically to the hardness of MAJORITY for AC_0.

This paper contains three main results, which together make Conjecture 1 interesting and worthy of further study:

- We show that our conjecture implies the existence of an oracle relative to which BQP is not in the PH, and would thus resolve a major question in quantum complexity. We are encouraged by the fact that our conjecture is recognizable as a natural question in pseudorandomness that has been previously and independently studied (e.g., in [BSW03]).

  The crucial component in showing that our conjecture is sufficient for the existence of an oracle relative to which BQP is not in the PH, is an explicit construction of unitary matrices whose row-supports form an (ℓ, p)-design. We give such a construction and show how to realize these matrices with small quantum circuits in Section 4. This is the technical core of the paper.

- We generalize the setting of [Aar10b] (which proposed a so-called *forrelated* distribution as one that is easy to distinguish from uniform by a quantum computer, but possibly hard for AC_0) to a simple framework in which any efficiently quantumly computable unitary U gives rise to a distribution that can be distinguished from uniform by a quantum computer (and Aaronson's setup is recovered by choosing U to be a DFT matrix).

  Together with our construction of explicit unitaries whose row-supports form an (ℓ, p)-design, this framework has the following interesting interpretation: it gives an instantiation of the Nisan-Wigderson generator that can be broken by quantum computers, but not by the relevant modes of classical computation, if Conjecture 1 is true.

  Also of independent interest is the fact the unitaries that form the basis of our quantum algorithms don't seem to resemble the DFT matrices for problems in the Hidden Subgroup framework, or even the few other unitaries used in known quantum algorithms. But they possess natural extremal combinatorial (as opposed to algebraic) properties, and we wonder if they can be useful elsewhere in the quantum realm.

- We show that the "Nisan-Wigderson" distribution $(U_t, NW_{\mathcal{D}}^{\text{MAJORITY}}(U_t))$ is ε-almost k-wise independent, in the sense of Aaronson [Aar10b], whose "GLN conjecture" asserted that all such distributions fool AC_0; a depth-3 counterexample was later found [Aar10a]. Whether all such distributions fool depth-2 AC_0 remains open. A distribution in our general framework (thus efficiently quantumly distinguishable from uniform) that fools depth-2 AC_0 would imply an oracle relative to which BQP is not in AM, a weaker (and still unresolved) version of the BQP vs. PH problem. Thus there are

two potential routes to resolving this weaker version of the main problem (the depth-2 version of our conjecture, and the depth-2 version of the GLN conjecture); ours is formally easier, and arguably conceptually easier because its connection to the pseudorandomness literature suggests initial lines of attack.

Finally, since [Aar10b] has shown that the classes SZK and BPP_{path} require exponentially many queries to distinguish ε-almost k-wise independent distributions from uniform, our constructions *unconditionally* yield oracles relative to which BQP does not lie in either of these classes (and MA as well, since $MA \subseteq BPP_{\text{path}}$), just as Aaronson's construction does.

1.1 The BQP vs. PH question

The quest for an oracle relative to which BQP is not in the PH dates to the foundational papers of the field; the question was first asked by Bernstein and Vazirani [BV93] in the early 1990's. They also gave an oracle problem, RECURSIVE FOURIER SAMPLING, that is regarded as a promising candidate (but there have been as yet no real inroads on a potential proof). Currently, oracles are known relative to which BQP is not in MA [Wat00], but no relativized worlds are known in which BQP is not in AM. Obtaining an oracle relative to which BQP is not in the PH thus represents a stubborn, longstanding and fundamental problem whose resolution would help clarify the relationship between BQP and classical complexity classes. In recent progress, Aaronson [Aar10b] devised a *relation* oracle problem that lies in the function version of BQP but not in the function version of the PH, but this still leaves the original problem open. Aaronson's work [Aar10b] also has a detailed account of the many motivations for revisiting (and hopefully resolving!) this problem, and we refer the interested reader to the introduction of [Aar10b] for many more details.

In this paper we will find it convenient to speak almost exclusively about the "scaled down" version of the problem, which is equivalent via the well-known connection between PH and AC_0. In it, the goal is to design a promise problem (rather than an oracle) that lies in (promise)-BQLOGTIME but not (promise)-AC_0 We will drop the cumbersome "promise" modifiers when they are clear from context. The class BQLOGTIME is the class of languages decidable by quantum computers that have random access to an N-bit input, and use only $O(\log N)$ steps.

Definition 1.3 (BQLOGTIME)**.** *A language L is in* BQLOGTIME *if it can be decided by a LOGTIME-uniform family of circuits $\{C_n\}$, where each C_n is a quantum circuit on n qubits. On an $(N = 2^n)$-bit input x, circuit C_n applies $O(\log N)$ gates, with each gate being either a* query *gate which applies the map $|i\rangle|z\rangle \mapsto |i\rangle|z \oplus x_i\rangle$, or a standard quantum gate (from a fixed, finite basis). It is equivalent (by polynomially padding the number of qubits) to allow $\operatorname{poly}\log(N)$ gates.*

Following Aaronson, our goal will be to design, for each input length N, a *distribution* on N-bit strings that can be distinguished from the uniform distribution by a BQLOGTIME predicate, but not by an AC_0 circuit. As described in Appendix C, such a distribution can be easily converted to a proper oracle O for which $BQP^O \not\subset PH^O$.

1.2 Techniques

In this section we briefly discuss the techniques we use for each of the main results listed above.

Showing that our NW distribution is ε-almost k-wise independent. We prove that whenever $\mathcal{D}$ is an (ℓ, p) design in a universe of size t, the random variable $(U_t, NW_{\mathcal{D}}^{\text{MAJORITY}}(U_t))$ is $O(pk^2/\sqrt{\ell})$-almost k-wise independent, for $k < o(\ell^{1/4}p^{-1/2})$. The relevant definition of almost-k-wise independence (which we inherit from [Aar10b]) appears in Definition 2.1. Recall that this property of our distribution is the technical basis of the SZK and BPP_{path} results, as well as the connections to the depth-2 GLN conjecture.

This statement amounts to the assertion that after conditioning on the value of up to $k-1$ coordinates, the bias (away from $1/2$) of any specified k-th coordinate is at most $O(pk/\sqrt{\ell})$. This is an easy calculation when the conditioned coordinates all lie among the first t coordinates (since the k-th coordinate is either completely independent, if it lies among the first t coordinates, or else it is MAJORITY applied to a subset of ℓ of the first t coordinates, of which up to $k-1$ may be fixed). In the actual proof, when some conditioned coordinates lie *outside* the first t coordinates (which would otherwise be difficult to analyze), we use the following simple trick to reduce to the easy case: we replace conditioning on coordinate $t+i$ with conditioning on *all* of the coordinates in set S_i of the (ℓ, p)-design (which determine it). Since at most p of these can affect the bias of the k-th coordinate, we are back in the easy case with up to $p(k-1)$ bits fixed instead of $(k-1)$.

Showing that our conjecture is sufficient to resolve the BQP vs. PH question. In order to show that our conjecture is sufficient to imply an oracle relative to which BQP is not in the PH, we need to discuss the quantum part of the argument. Conjecture 1 implies that the NW generator with certain parameters fools AC_0, which is one part of the overall argument. The other part is to exhibit a BQLOGTIME algorithm that "breaks" this instantiation of the NW generator. Generalizing [Aar10b], our quantum algorithm[2] will receive a random string $x \in \{+1, -1\}^t$ (which should be thought of as the input to the NW generator) as the first half of its input, and as the second half of its input, *either*

1. a second random string in $\{+1, -1\}^t$, *or*
2. a string containing the *signs* of a unitary U (with entries in $\{0, 1, -1\}$) applied to x.

The algorithm distinguishes the two cases (roughly) by querying x into the phases, applying U, multiplying the second string into the phases, and measuring in the Hadamard basis.

Note that in case (2), each coordinate of the second string is the sign of a $+1/-1$ weighted sum of certain coordinates of x; i.e., it computes MAJORITY (with a fixed pattern of inputs negated) on this subset of the coordinate of x. Thus, if we can construct a unitary U whose row-supports form an (ℓ, p) design $\mathcal{D}$ in a universe of size t, then case (2) will be the distribution $(U_t, NW_{\mathcal{D}}^{\text{MAJORITY}}(U_t))$, and case (1) will be the uniform distribution. The parameters of this instantiation of the NW generator will be such that Conjecture 1 implies that it fools AC_0. Our task becomes to construct such a unitary U.

Note that it is *not* possible to simply take an existing (ℓ, p) design (random, or other explicit constructions that appear in the literature [NW94, HR03]) and attach $+/-$ signs to the elements of the sets so as to make their characteristic vectors pairwise orthogonal, which is what is needed for them to come from the rows of a unitary U. On the other hand we have a different setting of the parameters in mind than usual: we want p to be unusually small (a constant), but the number of sets in the design is also unusually small (only $\text{poly}(\ell)$ instead of $\exp(\ell)$). For these parameters we manage to obtain the required (ℓ, p) design using a geometric construction, in which the sets are the characteristic vectors of pairs of lines in an affine plane. The strong symmetries in this construction allow us to assign $+/-$ signs to the elements of each set to achieve pairwise orthogonality of their characteristic vectors. In fact these set systems have only $t/2$ (rather than t) sets in

[2] We ignore normalization factors in this discussion.

them, so the resulting unitaries will have the required properties only among half of their rows, but a small modification of the distribution given to the quantum algorithm in case (2) above can handle this without difficulty.

In Section 4.2 we give a *local decomposition* (see Section 3.1 for the formal definition) of these unitaries, which is necessary to have an *efficient* quantum algorithm. This is the most technically involved part of the paper. We also describe a modification of our construction that is *extremal* in the sense that it optimizes all relevant parameters simultaneously: *all* rows of the unitary participate, we have $p \leqslant 2$, and $t \leqslant \ell^2$. This is not required for our results, but it is aesthetically pleasing. We have been unable to find a local decomposition that would enable us to actually use this construction as the basis of an efficient quantum algorithm, and we leave finding such a decomposition as an intriguing open problem.

2 NW distributions are ε-almost k-wise independent

Aaronson [Aar10b] used the following definition of ε-almost k-wise independence in order to formulate his "Generalized Linial-Nisan" (GLN) conjecture.

Definition 2.1. *A random variable D distributed on $\{0,1\}^r$ is* ε-almost k-wise independent *if for every k distinct indices $i_1, i_2, \ldots, i_k \in [r]$, and every $\alpha_1, \alpha_2, \ldots, \alpha_k \in \{0,1\}$ we have:*

$$1 - \varepsilon \leqslant \frac{\Pr[D_{i_1} = \alpha_1 \wedge D_{i_2} = \alpha_2 \wedge \cdots \wedge D_{i_k} = \alpha_k]}{2^{-k}} \leqslant 1 + \varepsilon.$$

The following is the GLN conjecture, which is now known to be false for depth 3 and higher [Aar10a], but remains open for depth 2:

Conjecture 2 ([Aar10b])**.** *Let D be any random variable distributed on $\{0,1\}^r$ that is $1/r^{\Omega(1)}$-almost $r^{\Omega(1)}$-wise independent*[3]*. Then for every constant-depth circuit C of size at most $m = 2^{r^{o(1)}}$,*

$$|\Pr[C(D) = 1] - \Pr[C(U_r) = 1]| \leqslant o(1).$$

We now show that certain instantiations of the NW generator, including the ones in our Conjecture 1, are ε-almost k-wise independent, with parameters such that the GLN conjecture implies ours.

Theorem 2.2. *Let $\mathcal{D} = \{S_1, S_2, \ldots, S_m\}$ be an (ℓ, p) design in a universe of size t. Then for every $k < o(\ell^{1/4} p^{-1/2})$, the jointly distributed random variable*

$$(C, D) = (U_t, NW_{\mathcal{D}}^{\text{MAJORITY}}(U_t))$$

is $O(pk^2/\sqrt{\ell})$-almost k-wise independent.

Proof. Fix k_1 distinct indices $i_1, i_2, \ldots, i_{k_1} \in [t]$ and k_2 distinct indices $j_1, j_2, \ldots, j_{k_2} \in [m]$ with $k_1 + k_2 \leqslant k$, and fix $\alpha_1, \alpha_2, \ldots, \alpha_{k_1}, \beta_1, \beta_2, \ldots, \beta_{k_2} \in \{0,1\}$.

We compute the probability

$$\rho = \Pr[C_{i_1} = \alpha_1 \wedge C_{i_2} = \alpha_2 \wedge \cdots \wedge C_{i_{k_1}} = \alpha_{k_1} \wedge D_{j_1} = \beta_1 \wedge D_{j_2} = \beta_2 \wedge \cdots \wedge D_{j_{k_2}} = \beta_{k_2}],$$

[3] One might expect to see $k = \operatorname{poly}\log(r)$ independence rather than $k = r^{\Omega(1)}$, in analogy with the Linial-Nisan conjecture. Aaronson uses the stronger parameter setting (making the GLN conjecture easier) because it is sufficient for his construction; it is for ours too.

which we write as

$$\begin{aligned}\rho &= \left(\prod_{w=1}^{k_1} \Pr[C_{i_w} = \alpha_w | C_{i_1} = \alpha_1 \wedge C_2 = \alpha_2 \wedge \cdots \wedge C_{i_{w-1}} = \alpha_{i_{w-1}}]\right) \\ &\times \left(\prod_{w=1}^{k_2} \Pr[D_{j_w} = \beta_j | C_{i_1} = \alpha_1 \wedge C_2 = \alpha_2 \wedge \cdots \wedge C_{i_{k_1}} = \alpha_{i_{k_1}} \right. \\ &\qquad \left. \wedge D_{j_1} = \beta_{j_1} \wedge D_{j_2} = \beta_{j_2} \wedge \cdots \wedge D_{j_{w-1}} = \beta_{w-1}]\right).\end{aligned}$$

Clearly the first k_1 terms of the product are exactly $1/2$, since C is uniform on t-bit strings. Now, consider the w-th factor, denoted ρ_w, in the second part of the product. The key maneuver is to replace the conditioning on D_{j_v} (for $v < w$) with conditioning on D_s for $s \in S_w \cap S_v$. This is permissible because D_{j_v} can affect D_{j_w} only through the common elements of their associated sets S_v and S_w. Note that because $|S_w \cap S_v| \leqslant p$, the total number of coordinates that are being conditioned upon is $\leqslant pk$.

Recall that $|S_w| = \ell$, and that the bit D_w is the majority (with certain inputs negated) of the specified ℓ coordinates of C. Without conditioning, we could compute $\Pr[D_w = 1]$ by

$$\frac{1}{2^\ell} \cdot \sum_{r=\lceil \ell/2 \rceil}^{\ell} \binom{\ell}{r}.$$

We want to compute instead ρ_w, which is the same probability conditioned on up to pk of the coordinates of C. The maximum value of ρ_w is thus

$$\rho_w \leqslant \frac{1}{2^\ell} \cdot \sum_{r=\lceil \ell/2 \rceil - pk}^{\ell} \binom{\ell}{r}.$$

A simple calculation using Stirling's Approximation shows that $\binom{\ell}{r} \leqslant O(\frac{2^\ell}{\sqrt{\ell}})$ for all r, so we obtain the upper bound of

$$\rho_w \leqslant \frac{1}{2} + O(pk/\sqrt{\ell}).$$

A symmetric argument shows that

$$\rho_w \geqslant \frac{1}{2} - O(pk/\sqrt{\ell}).$$

Thus we conclude (using that $k < o(\sqrt{\ell}/(pk))$):

$$\rho \leqslant \left(1/2 + O(pk/\sqrt{\ell})\right)^k \leqslant \left[(1/2)\left(1 + O(pk/\sqrt{\ell})\right)\right]^k \leqslant 2^{-k}\left(1 + O(pk^2/\sqrt{\ell})\right),$$

and

$$\rho \geqslant \left(1/2 - O(pk/\sqrt{\ell})\right)^k \geqslant \left[(1/2)\left(1 - O(pk/\sqrt{\ell})\right)\right]^k \geqslant 2^{-k}\left(1 - O(pk^2/\sqrt{\ell})\right),$$

as required. □

3 A general framework

In this section we describe how to turn any efficiently quantumly computable unitary into a distribution that can be distinguished from uniform by a BQLOGTIME machine. Our framework generalizes the setup in [Aar10b]. The "quantum part" of the paper is almost entirely contained within this section, so we review some relevant preliminaries below before describing the main result.

3.1 Quantum preliminaries

A *unitary* matrix is a square matrix U with complex entries such that $UU^* = I$, where U^* is the conjugate transpose. Equivalently, its rows (and columns) form an orthonormal basis. We name the standard basis vectors of the $N = 2^n$-dimensional vectorspace underlying an n-qubit system by $|v\rangle$ for $v \in \{0,1\}^n$. A *local* unitary is a unitary that operates only on $b = O(1)$ qubits; i.e. after a suitable renaming of the standard basis by reordering qubits, it is the matrix $U \otimes I_{2^{n-b}}$, where U is a $2^b \times 2^b$ unitary U. A local unitary can be applied in a single step of a quantum computer. A *local decomposition* of a unitary is a factorization into local unitaries. We say an $N \times N$ unitary is *efficiently quantumly computable* if this factorization has at most $\mathrm{poly}(n)$ factors.

A *quantum circuit* applies a sequence of local unitaries ("gates") where each gate is drawn from a fixed, finite set of gates. There are universal finite gate sets for which any efficiently quantumly computable unitary can be realized (up to exponentially small error) by a $\mathrm{poly}(n)$-size quantum circuit [KSV02].

In this paper, the only manner in which our BQLOGTIME algorithm will access the input string x is the following operation, which "multiplies x into the phases". There are three steps: (1) query with the query register clean, which applies the map $|i\rangle|0\rangle \mapsto |i\rangle|0 \oplus x_i\rangle$ (note each x_i is in $\{0,1\}$); (2) apply to the last qubit the map $|0\rangle \mapsto -|0\rangle, |1\rangle \mapsto |1\rangle$; (3) query again to uncompute the last qubit. When we speak of "multiplying x into the phase" it will be linguistically convenient to speak about x as a vector with entries from $\{+1,-1\}$, even though one can see from this procedure that the actual input is a $0/1$ vector.

The following lemma will be useful repeatedly. It states (essentially) that a block diagonal matrix, all of whose blocks are efficiently quantumly computable, is itself efficiently quantumly computable. This is trivial when all of the blocks are identical, but not entirely obvious in general. The proof is in Appendix A

Lemma 3.1. *Fix $N = 2^n$ and $M = 2^m$. Let U be an $N \times N$ block diagonal matrix composed of the blocks $U_1, U_2, \ldots, U_M$, where each U_i is a $N/M \times N/M$ matrix that has a $\mathrm{poly}(n)$-size quantum circuit, a description of which is generated by a uniform $\mathrm{poly}(n)$ time procedure, given input i. Then given three registers of m qubits, $n-m$ qubits, and $\mathrm{poly}(n)$ qubits, respectively, with the third register initialized to $|000\cdots0\rangle$, there is a $\mathrm{poly}(n)$ size uniform quantum circuit that applies U to the first two registers and leaves the third unchanged.*

3.2 The quantum algorithm

Let A be any $N \times N$ matrix with entries[4] in $\{0,1,-1\}$ and pairwise orthogonal rows, and define $S(A,i)$ to be the support of the i-th row of matrix A. Define $\overline{A}$ to be the matrix A with entries in row i scaled by $1/\sqrt{|S(A,i)|}$, and observe that $\overline{A}$ is a unitary matrix.

Define the random variable $D_{A,M} = (x,z)$ distributed on $\{+1,-1\}^{2N}$ by picking $x \in \{+1,-1\}^N$ uniformly, and setting the next N bits to be $z \in \{+1,-1\}^N$ defined by $z_i = \mathrm{sgn}((Ax)_i) = \mathrm{sgn}((\overline{A}x)_i)$ for $i \leqslant M$ and z_i independently and uniformly random in $\{+1,-1\}$ for $i > M$.

It will be convenient to think of $M = N$ initially; we analyze the general case because we will eventually need to handle $M = N/2$. Below, we use U_{2N} to denote the random variable uniformly distributed on $\{+1,-1\}^{2N}$.

Theorem 3.2. *Let $N = 2^n$ for an integer $n > 0$, and let $M = \Omega(N)$. For every matrix $A \in \{0,1,-1\}^{N \times N}$ with pairwise orthogonal rows, there is a BQLOGTIME algorithm Q_A that distinguishes $D_{A,M}$ from U_{2N};*

[4]We could extend this framework to matrices with general entries, but we choose to present this restriction since it is all we need.

i.e., there is some constant $\varepsilon > 0$ for which:

$$|\Pr[Q_A(D_{A,M}) = 1] - \Pr[Q_A(U_{2N}) = 1]| > \varepsilon.$$

The algorithm is uniform if A comes from a uniform family of matrices.

Proof. The input to the algorithm is a pair of strings $x, z \in \{+1, -1\}^N$.

The algorithm performs the following steps:

1. Enter a uniform superposition $\frac{1}{\sqrt{N}} \sum_{i \in \{0,1\}^n} |i\rangle$ and multiply x into the phase to obtain $\frac{1}{\sqrt{N}} \sum_{i \in \{0,1\}^n} x_i |i\rangle$.
2. Apply $\overline{A}$ to obtain $\frac{1}{\sqrt{N}} \sum_{i \in \{0,1\}^n} (\overline{A}x)_i |i\rangle$.
3. Multiply z into the phase to obtain $\frac{1}{\sqrt{N}} \sum_{i \in \{0,1\}^n} z_i (\overline{A}x)_i |i\rangle$.
4. Define vector w by $w_i = \frac{1}{\sqrt{N}} z_i (\overline{A}x)_i$. Apply the $N \times N$ Hadamard[5] H to obtain $\sum_{i \in \{0,1\}^n} (Hw)_i |i\rangle$, and measure in the computational basis. Accept iff the outcome is 0^n.

We first argue that the acceptance probability is small in case (x, z) is distributed as U_{2N}. This follows from a symmetry argument: for fixed x, and w as defined in Step 4 above, the vector Hw above has every entry identically distributed, because z is independently chosen uniformly from $\{-1, +1\}^N$ and every row of H is a vector in $\{-1, +1\}^N$. In particular this implies that the random variable $(Hw)_i^2$ is identically distributed for all i. Together with the fact that $\sum_i (Hw)_i^2 = 1$, we conclude that $E[(Hw)_i^2] = 1/N$. Then by Markov, with probability at least $1 - 1/\sqrt{N}$ we accept with probability at most $\sqrt{N}/N$, for an overall acceptance probability of at most $2/\sqrt{N}$.

Next, we argue that the acceptance probability is large in case (x, z) is distributed as $D_{A,M}$. Here we observe that for $i \leqslant M$, $w_i = \frac{1}{\sqrt{N}} |(\overline{A}x)_i|$ and hence $E[w_i] = \frac{1}{\sqrt{N \cdot |S(A,i)|}} \Omega(\sqrt{|S(A,i)|}) = \Omega(1/\sqrt{N})$ (since before scaling, w_i is just the distance from the origin of a random walk on the line, with $|S(A,i)|$ steps). For $i > M$, we simply have $E[w_i] = 0$. Then $E[\sum_i w_i] = M \cdot \Omega(1/\sqrt{N}) = \Omega(\sqrt{N})$, so $E[(Hw)_{0^n}] = \Omega(1)$. Since the random variable $(Hw)_{0^n}$ is always bounded above by 1, we can apply Markov to its negation to conclude that with constant probability, it is *at least* a constant ε (and in such cases the acceptance probability is at least ε^2). Overall, the acceptance probability is $\Omega(1)$. □

The BQLOGTIME algorithm for what Aaronson calls FOURIER CHECKING in [Aar10b] is recovered from the above framework by taking A to be a DFT matrix (and $M = N$).

4 Unitary matrices with large, nearly-disjoint row supports

In this section we construct unitary matrices A with the additional property that all or "almost all" of the row supports $S(A, i)$ are large and have bounded intersections. We also show that these unitaries are efficiently quantumly computable. This is the final part of our main result: the distribution $D_{A,M}$ (it will turn out that M will be half the underlying dimension) is distinguishable from uniform by a BQLOGTIME algorithm by Theorem 3.2, and at the same time $D_{A,M}$ can be seen as an NW distribution that by Conjecture 1 fools AC_0 (see Section 5 for the precise statement).

[5] This is the matrix H whose rows and columns are indexed by $\{0,1\}^n$, with entry (i, j) equal to $-1^{\langle i,j \rangle}/\sqrt{N}$.

4.1 The paired-lines construction

We describe a collection of $q^2/2$ pairwise-orthogonal rows, each of which is a vector in $\{0, +1, -1\}^{q^2}$. We identify q^2 with the affine plane $\mathbb{F}_q \times \mathbb{F}_q$, where $q = 2^n$ for an integer $n > 0$. Let B_1, B_2 be an equipartition of $\mathbb{F}_q$, and let $\phi : B_1 \to B_2$ be an arbitrary bijection. Our vectors are indexed by a pair $(a, b) \in \mathbb{F}_q \times B_1$, and their coordinates are naturally identified with $\mathbb{F}_q \times \mathbb{F}_q$:

$$v_{a,b}[x, y] = \begin{cases} -1 & y = ax + b \\ +1 & y = ax + \phi(b) \end{cases} \tag{1}$$

Notice that $v(a, b)$ is -1 on exactly the points of $\mathbb{F}_q \times \mathbb{F}_q$ corresponding to the line with slope a and y-intercept b, and $+1$ on exactly the points of $\mathbb{F}_q \times \mathbb{F}_q$ corresponding to the line with slope a and y-intercept $\phi(b)$. So each $v(a, b)$ is supported on exactly a pair of parallel lines. Orthogonality will follow from the fact that every two non-parallel line-pairs intersect in exactly one point, as argued in the proof of the next lemma.

Lemma 4.1. *The vectors defined in Eq. (1) are pairwise orthogonal, and their supports form a* $(2q, 4)$ *design.*

Proof. Consider $(a, b) \neq (a', b')$. If $a = a'$ then the supports of $v(a, b)$ and $v(a, b')$ are disjoint. Otherwise $a \neq a'$ and there are exactly four intersection points (obtained by solving linear equations over $\mathbb{F}_q$):

- $(x = (b' - b)/(a - a'), y = ax + b) = (x = (b' - b)/(a - a'), y = a'x + b')$, which contributes $(-1) \cdot (-1) = 1$ to the inner product, and
- $(x = (b' - \phi(b))/(a - a'), y = ax + \phi(b)) = (x = (b' - \phi(b))/(a - a'), y = a'x + b')$, which contributes $(+1) \cdot (-1) = -1$ to the inner product, and
- $(x = (\phi(b') - b)/(a - a'), y = ax + b) = (x = (\phi(b') - b)/(a - a'), y = a'x + \phi(b'))$, which contributes $(-1) \cdot (+1) = -1$ to the inner product, and
- $(x = (\phi(b') - \phi(b))/(a - a'), y = ax + \phi(b)) = (x = (\phi(b') - \phi(b))/(a - a'), y = a'x + \phi(b'))$, which contributes $(+1) \cdot (+1) = 1$ to the inner product.

The sum of the contributions to the inner product from these four points is zero. The computation of the support size is straightforward. □

In Appendix B, we give another construction (which is not needed for our main result) in which the number of vectors is exactly equal to the dimension of the underlying space (giving rise to a unitary in which "all rows participate" instead of only half of the rows).

4.2 A local decomposition

We new describe an $q^2 \times q^2$ unitary matrix that is efficiently quantumly computable and has the (normalized) vectors $v(a, b)$ from Eq. (1) as $q^2/2$ of its q^2 rows. We recall that $q = 2^n$ for an integer $n > 0$.

Proposition 4.2. *The following $q \times q$ unitary matrices are efficiently quantumly computable:*

- *The DFT matrix F with respect to the additive group of $\mathbb{F}_q$.*
- *The inverse DFT matrix F^{-1} with respect to the additive group of $\mathbb{F}_q$.*

- *The $q \times q$ unitary matrix B with $\frac{1}{\sqrt{2}}(I_{q/2}| - I_{q/2})$ as its first $q/2$ rows, $\frac{1}{\sqrt{4}}(I_{q/4}| - I_{q/4}|I_{q/4}| - I_{q/4})$ as its next $q/4$ rows, $\frac{1}{\sqrt{8}}(I_{q/8}| - I_{q/8}|I_{q/8}| - I_{q/8}|I_{q/8}| - I_{q/8}|I_{q/8}| - I_{q/8})$ as its next $q/8$ rows, etc... and whose last row is $\frac{1}{\sqrt{N}}(1, 1, 1, \ldots, 1)$.*

Proof. The first two matrices are well-known to be efficiently quantumly computable. For the last one we make use of the Hadamard matrix

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & -1 \\ 1 & 1 \end{pmatrix}.$$

Let B_i be the $q \times q$ identity matrix with its lower right $2^i \times 2^i$ submatrix replaced by the matrix $H \otimes I_{2^{i-1}}$. Each B_i is efficiently quantumly computable by Lemma 3.1. It is then easy to verify that $B = B_1 B_2 B_3 \cdots B_n$. □

Lemma 4.3. *Let α be a generator of the multiplicative group of $\mathbb{F}_q$. For $c \in \mathbb{F}_q$, let D_c denote the $q \times q$ diagonal matrix*

$$\frac{1}{\sqrt{q}} \cdot \mathit{diag}\left(\sqrt{q}, (-1)^{\mathrm{Tr}\,(\alpha^1 \cdot c)}, (-1)^{\mathrm{Tr}\,(\alpha^2 \cdot c)}, (-1)^{\mathrm{Tr}\,(\alpha^3 \cdot c)}, \ldots, (-1)^{\mathrm{Tr}\,(\alpha^{q-1} \cdot c)}\right),$$

and let D'_c denote the $q \times q$ diagonal matrix

$$\frac{1}{\sqrt{q}} \cdot \mathit{diag}\left(0, (-1)^{\mathrm{Tr}\,(\alpha^1 \cdot c)}, (-1)^{\mathrm{Tr}\,(\alpha^2 \cdot c)}, (-1)^{\mathrm{Tr}\,(\alpha^3 \cdot c)}, \ldots, (-1)^{\mathrm{Tr}\,(\alpha^{q-1} \cdot c)}\right).$$

Then the $q^2 \times q^2$ matrix D whose (i, j) block (with $i, j \in \mathbb{F}_q$) equals D_{ij} if $i = j$ and D'_{ij} otherwise, is efficiently quantumly computable.

Proof. Consider the $q^2 \times q^2$ block-diagonal matrix that has as its (k, k) block the matrix whose (i, j) entry is $(-1)^{\mathrm{Tr}\,(ij\alpha^k)}$ for $k \in \{1, 2, \ldots, q-1\}$ and whose $(0, 0)$ block is I_q. Each such block except the $(0, 0)$ block is the DFT matrix F with its rows (or equivalently, columns) renamed according to the map $j \mapsto j\alpha^k$. The F matrix is efficiently quantumly computable and the map $j \mapsto j\alpha^k$ is classically and reversibly (and thus quantumly) efficiently computable. Thus each $q \times q$ block on the diagonal is efficiently quantumly computable. By Lemma 3.1 the entire matrix is efficiently quantumly computable.

If we index columns by $(i, i') \in (\mathbb{F}_q)^2$ and rows by $(j, j') \in (\mathbb{F}_q)^2$, then the desired matrix D is the above block-diagonal matrix with the order of the two indexing coordinates for the rows transposed, and the order of the two indexing coordinates for the columns transposed. □

Theorem 4.4. *The $q^2 \times q^2$ matrix $(I_q \otimes B) \cdot (I_q \otimes F) \cdot D \cdot (I_q \otimes F^{-1})$, which is efficiently quantumly computable, has the vectors $v(a, b)$ from Eq. (1) as $q^2/2$ of its rows*[6].

Proof. Let S_c be the $q \times q$ permutation matrix S_c that (when multiplied on the right) shifts columns, identified with $\mathbb{F}_q$, by the map $x \mapsto x + c$. Let J be the all-ones matrix. The main observation is that

$$F D_c F^{-1} = \frac{1}{\sqrt{q}} S_c - \frac{\sqrt{q} - 1}{q} J,$$

and that

$$F D'_c F^{-1} = \frac{1}{\sqrt{q}} S_c - \frac{1}{\sqrt{q}} J.$$

[6]To be precise, these are the $v(a, b)$ with respect to *some* equipartition B_1, B_2 and *some* bijection ϕ.

Thus the final matrix has in its (i,j) block (with $i,j \in \mathbb{F}_q$) the matrix

$$B \cdot \left(\frac{1}{\sqrt{q}} S_{ij} - \frac{\sqrt{q}-1}{q} J \right)$$

if $i = j$, and

$$B \cdot \left(\frac{1}{\sqrt{q}} S_{ij} - \frac{1}{\sqrt{q}} J \right)$$

otherwise. Observe that BJ has all zero entries except for the last row, so in particular, the first $q/2$ rows of the (i,j) block are $(1/\sqrt{2q})(I_{q/2}| - I_{q/2})S_{ij}$. Therefore the $q/2$ rows of the entire $q^2 \times q^2$ matrix corresponding to the top halves of blocks (i,j) as j varies, give the vectors $v(i,b)$ for $b \in B_1$, if we identify columns with $\mathbb{F}_q \times \mathbb{F}_q$ as follows: columns of the j-th block are identified with $\{j\} \times \mathbb{F}_q$, and within the j-th block, B_1 is the first $q/2$ columns and B_2 is the next $q/2$ columns (and the bijection ϕ maps the element associated with the b-th column to the element associated with the $(b + q/2)$-th column).

Then, as i varies over $\mathbb{F}_q$, we find all of the vectors from Eq. (1) as the "top-halves" of each successive set of q rows of the large matrix. □

5 Putting it all together

Let A be the matrix of Theorem 4.4, and set $N = q^2$ and $M = N/2$. By Theorem 3.2, there is a BQLOG-TIME algorithm that distinguishes $D_{A,M}$ from the the uniform distribution U_{2N}.

By Lemma 4.1, the first M rows of A have supports forming a $(2\sqrt{N}, 4)$-design $\mathcal{D}$. It is also clear that for $i \leqslant M$, the $(N+i)$-th bit of $D_{A,M}$ computes MAJORITY (with a fixed pattern of inputs negated) on those among the first N bits that lie in $S(A,i)$. Thus $D_{A,M}$ is exactly the distribution $(U_N, NW_{\mathcal{D}}^{\text{MAJORITY}}(U_N))$ followed by $N/2$ additional independent random bits (which can have no impact on the distinguishability of the distribution from uniform). Thus by Conjecture 1, no constant-depth, polynomial-size circuit can distinguish $D_{A,M}$ from U_{2N}, which completes the argument.

We briefly describe why the standard NW argument fails (and why we must rely on Conjecture 1). The standard argument proceeds as follows: define $2N+1$ hybrid distributions $D_{A,M} = H_0, H_1, \ldots, H_{2N} = U_{2N}$, that interpolate between $D_{A,M}$ and U_{2N}. Given a distinguishing circuit $C : \{0,1\}^{2N} \to \{0,1\}$ for which

$$|\Pr[C(D_{A,M}) = 1] - \Pr[C(U_{2N}) = 1]| \geqslant \varepsilon,$$

we argue that for some i

$$|\Pr[C(H_i) = 1] - \Pr[C(H_{i+1}) = 1]| \geqslant \varepsilon/M$$

by the triangle inequality (and here we are making the additional observation that $H_0 = H_1 = \cdots = H_N$ and $H_{N+M+1} = H_{N+M+2} = \cdots = H_{2N}$ so the gap of ε must be spread over only M differences). From here, we obtain a next-bit-predictor with advantage ε/M and hardwire at most M lookup tables of size 2^p, to obtain a circuit of size $|C| + O(2N) + O(2^p M)$ that computes MAJORITY (on $2\sqrt{N}$ bits) with success probability $1/2 + \varepsilon/M$. The problem is that this advantage over random guessing is not sufficient to obtain a contradiction for the function MAJORITY, which can be computed easily with success probability $1/2 + \Omega(N^{1/4})$, for the parameters coming from the unitary A from Theorem 4.4.

Even if we had a unitary whose rows formed an (ℓ, p)-design with better parameters, the standard argument fails. This is because it must be that $\ell \leqslant N$, and yet we must also have $M \gg \sqrt{N}$ for $D_{A,M}$ to be even *statistically* noticably different from uniform. But the trivial circuit that outputs an arbitrary bit of the input succeeds with probability $1/2 + \Omega(1/\sqrt{\ell})$ which is larger than the $1/2 + \varepsilon/M$ that comes out of the standard NW argument above.

6 Our conjecture: discussion

We believe that Conjecture 1 is quite approachable, given the large literature and variety of proof techniques concerning pseudorandom generators and related objects. As examples, we mention two ideas from the literature that seem relevant (although obviously they haven't yet led to a solution).

The first is the analysis by Sudan, Trevisan, and Vadhan [STV01] of the NW PRG when applied to a "mildly hard" predicate (i.e., one for which small circuits fail on only a δ fraction of the inputs). They prove that the output distribution is computationally indistinguishable from a distribution having high entropy by invoking Impagliazzo's hard-core lemma [Imp95], and arguing that output bits of the NW PRG "often" fall in a hard core that is considerably harder on average than the original mildly hard predicate.

We also have a hard predicate whose average-case hardness falls short of what we would need for Conjecture 1 to be true via the standard argument; i.e., if MAJORITY on ℓ bits were $1/2+1/\mathrm{poly}(\ell)$ hard, we would be done. The high-level message of Sudan, Trevisan and Vadhan is that this hardness can be achieved (essentially) at the price of comparing to a high-entropy distribution rather than the uniform distribution. Our BQP algorithm is fairly robust and would likely still work on a sufficiently high entropy distribution (it is only necessary to "kill" correlations with a particular element of the Hadamard basis). However, the central technical component of the proof in [STV01] is the Impagliazzo hard-core lemma [Imp95], and a sufficiently strong hardcore lemma is not known for AC_0. In fact, the function MAJORITY has no hard core:

Proposition 6.1. *No subset of* MAJORITY *is ε-hardcore for AC_0, for any $\varepsilon < 1/n$.*

Proof. Given a $x \in \{0,1\}^n$, the randomized procedure that picks a random one of the n input bits and outputs it succeeds in computing $\textsc{majority}(x)$ with probability at least $1/2 + 1/n$. This procedure has the same success probability over any subset $S \subseteq \{0,1\}^n$. For any fixed S, there is a fixing of the random bits that preserves this success probability, and which results in a circuit of size 1 (it just outputs x_i for some fixed i). □

Nevertheless, it may be that replacing the uniform distribution with a high minentropy one can be useful in circumventing the loss from the hybrid argument.

The second approach is to directly circumvent the loss due to the hybrid argument. This is explicitly addressed in [BSW03], where they show that the loss can indeed be avoided in certain computational models. One of these models is "PH circuits," which sounds superficially like it might be relevant to our setting. What is actually needed to use their ideas is the ability to approximately count an efficiently recognizable set, in the same class that recognizes the set. Such a statement is not known (or expected) for AC_0, but it is still possible that other ideas could circumvent the hybrid argument for AC_0.

However, any route to proving Conjecture 1 faces the same challenge discussed in [Aar10b]: the proof must be "non-black-box" in the sense that it can't apply to arbitrary low-degree polynomial functions in addition to its native Boolean setting. This is because the quantum algorithm of Theorem 3.2 implies (via [BHC$^+$01]) the existence of a constant-degree, multivariate real polynomial computing the acceptance probability (and hence distinguishing the NW distribution from uniform). A black-box reduction would transform a distinguisher of this form to a similarly low-degree polynomial approximating MAJORITY, but we know that no such polynomial for approximating MAJORITY can exist [Smo93]. So any proof of Conjecture 1 must prove that the distribution in question fools AC_0 in some way that does *not* replace AC_0 circuits by low degree approximating polynomials and then argue about those.

Here are some ideas that could plausibly form the basis of a proof of Conjecture 1. We consider the simpler situation in which the distributions being compared are N^2 independent copies of the random variable D – where $D = (U_N, \textsc{majority}(U_N))$ – and N^2 independent copies of the random variable U_{N+1}

distributed uniformly on $N+1$ bits. This corresponds to the NW construction we have been working with, if the underlying nearly-disjoint sets are taken to be *completely disjoint*. AC_0 should be incapable of distinguishing these distributions; here is the intuition. First, observe that there are no correlations between blocks, so the hypothetical distinguisher must examine each block separately. Since AC_0 cannot approximate majority well, we know that the only "accessible" information about each block is a "noisy bit" saying whether it is distributed according to D or U_{N+1} – in the case of uniform, this bit is 1 with probability $1/2$, and in the case of distribution D, this bit is 1 with probability $1/2 + \Theta(1/\sqrt{N})$. How can a hypothetical distinguisher aggregate these noisy bits across the N^2 independent copies? In one case, the expected sum of these noisy bits $(1/2)N^2$ and in the other case it is $(1/2+\Theta(1/\sqrt{N}))N^2$, and by concentration of measure, the sum is highly likely to be close to these expectations. So the hypothetical distinguisher only needs to tell the difference between N^2 fair coin flips versus N^2 slightly biased coin flips. But exactly this task is hard for AC_0 (which can be seen by reduction from MAJORITY, as written down in Corollary 12 of [Aar10b]). So, it seems that either the distinguisher must approximate MAJORITY better than allowed (to get less noisy bits), or it must be detecting very small bias in a sequence of coin flips. In upcoming work [FSUV10], we are able to show that indeed AC_0 cannot distinguish these two distributions. This is encouraging because it shows that the aforementioned "non-black-box" requirement is not insurmountable. Extending this result to the not-completely-disjoint case still seems challenging, however.

Acknowledgements. We thank Scott Aaronson, Yi-Kai Liu, and Emanuele Viola for helpful discussions.

References

[Aar10a] S. Aaronson. A counterexample to the Generalized Linial-Nisan Conjecture. In *ECCCTR: Electronic Colloquium on Computational Complexity, technical reports*, number 109, 2010.

[Aar10b] Scott Aaronson. BQP and the polynomial hierarchy. In Leonard J. Schulman, editor, *STOC*, pages 141–150. ACM, 2010.

[BHC+01] R. Beals, H.Buhrman, R. Cleve, M. Mosca, and R. de Wolf. Quantum lower bounds by polynomials. *J. ACM*, 48(4):778–797, 2001.

[BSW03] B. Barak, R. Shaltiel, and A. Wigderson. Computational analogues of entropy. In Sanjeev Arora, Klaus Jansen, José D. P. Rolim, and Amit Sahai, editors, *RANDOM-APPROX*, volume 2764 of *Lecture Notes in Computer Science*, pages 200–215. Springer, 2003.

[BV93] E. Bernstein and U. V. Vazirani. Quantum complexity theory. In *STOC*, pages 11–20, 1993.

[FSUV10] B. Fefferman, R. Shaltiel, C. Umans, and E. Viola. On beating the hybrid argument. Submitted, 2010.

[GM84] S. Goldwasser and S. Micali. Probabilistic encryption. *J. Comput. Syst. Sci.*, 28(2):270–299, 1984.

[Hås87] Johan Håstad. *Computational limitations of small-depth circuits*. MIT Press, 1987.

[HR03] T. Hartman and R. Raz. On the distribution of the number of roots of polynomials and explicit weak designs. *Random Struct. Algorithms*, 23(3):235–263, 2003.

[Imp95] R. Impagliazzo. Hard-core distributions for somewhat hard problems. In *FOCS*, pages 538–545, 1995.

[KSV02] A.Y Kitaev, A.H Shen, and M.N Vyalyi. *Classical and Quantum Computation*. AMS, 2002.

[Nis92] N. Nisan. Pseudorandom generators for space-bounded computation. *Combinatorica*, 12(4):449–461, 1992.

[NW94] N. Nisan and A. Wigderson. Hardness vs randomness. *J. Comput. Syst. Sci.*, 49(2):149–167, 1994.

[Smo93] R. Smolensky. On representations by low-degree polynomials. In *FOCS*, pages 130–138. IEEE, 1993.

[STV01] M. Sudan, L. Trevisan, and S. P. Vadhan. Pseudorandom generators without the XOR lemma. *J. Comput. Syst. Sci.*, 62(2):236–266, 2001.

[Wat00] J. Watrous. Succinct quantum proofs for properties of finite groups. In *FOCS*, pages 537–546, 2000.

[Yao82] A. Yao. Theory and applications of trapdoor functions (extended abstract). In *FOCS*, pages 80–91. IEEE, 1982.

A Omitted proofs

Proof. (Of Lemma 3.1) Fix a finite universal set of quantum gates, of cardinality d, each of which operates on at most b qubits. A convenient notion will be that of an *oblivious* circuit, in which we fix an ordering (say, lexicographic) on $[n]^b$, and the steps of the circuit are identified with $\mathrm{poly}(n)$ cycles through this list: when we are on step $(a_1, a_2, \ldots, a_b) \in [n]^b$ in one of these cycles, we operate on qubits $a_1, a_2, \ldots, a_b$. Clearly, any (uniform) quantum circuit can be converted to a (uniform) "oblivious" circuit with at most an n^b blowup by inserting dummy identity gates.

Let n^k be an upper bound on the size of the oblivious circuits obtained in this way for the various U_i. The circuit for each U_i is now a sequence

$$j^{(i)} = \left(j_1^{(i)}, j_2^{(i)}, j_3^{(i)}, \ldots, j_{n^k}^{(i)}\right),$$

with each $j_\ell^{(i)} \in [d]$ specifying which gate to apply at step ℓ in the oblivious circuit for U_i (and because the circuit is oblivious, the qubits to which this gate should be applied are easily determined from ℓ). Let $f : [M] \to [d]^{n^k}$ be the function that maps i to the vector $j^{(i)}$.

Now we describe the promised efficient quantum procedure:

1. Apply the map derived from f that takes $|i\rangle|z\rangle$ to $|i\rangle|z \oplus f(i)\rangle$, to the first and third register. We view the contents of the third register as a vector in $[d]^{n^k}$.

2. Repeat for $\ell = 1, 2, 3, \ldots, n^k$: apply the "controlled unitary" that consults the ℓ-th component of the third register, and applies the specified gate to qubits $(a_1, a_2, \ldots, a_b)$ of the second register (again, $(a_1, a_2, \ldots, a_b)$ are easily determined from ℓ because the circuit is oblivious). The important observation is that this "controlled unitary" operates on only constantly many qubits.

3. Repeat step 1 to uncompute the auxiliary information in the third register.

□

B A unitary in which all rows participate

There is a tension between the triple goals of (1) having many pairwise orthogonal vectors, (2) maintaining bounded pairwise intersections of the supports, and (3) having the supports large. It is natural to wonder whether the above construction (in which we found a number of vectors equal to $1/2$ the dimension of the underlying space) is in some sense optimal. For example, is there some barrier to simultaneously optimizing all three goals?

Here we show that one can indeed optimize all three goals at the same time, by specifying a construction that builds on the "paired-lines" construction. Our construction will have as many pairwise orthogonal vectors as the dimension of the underlying space (which is obviously as many as is possible); it will have intersections sizes bounded above by 2 (the upper bound cannot be 0 without constraining the product of the number of rows and the support sizes to be at most the dimension of the underlying space, and no pairwise intersections can have cardinality one without violating orthogonality); the support sizes will be at least the square root of the dimension of the underlying space (and one can't exceed that without having larger intersection sizes).

This construction is not needed for our main results, but we find it aesthetically pleasing that one can optimize all three parameters in this way. We *don't* know of a local decomposition for this matrix, and we leave finding one as an intriguing open problem.

While the construction of Section 4.1 needed characteristic two, the present construction needs odd characteristic. We fix $\mathbb{F}_q$ with q an odd prime power, and we choose a subset $Q \subseteq \mathbb{F}_q^*$ of size $(q-1)/2$ for which $Q \cap -Q = \emptyset$, where $-Q = \{-x : x \in Q\}$. Our vectors will have $q^2 - 1$ coordinates, identified with the *punctured plane* $P = \mathbb{F}_q \times \mathbb{F}_q \setminus \{(0,0)\}$.

We have three types of vectors in $\{0,-1,+1\}^P$: first, for all $a \in \mathbb{F}_q$ and $b \in Q$

$$v_{a,b}[x,y] = \begin{cases} +1 & x = 0, y = b \\ +1 & x \in Q, y = ax + b \\ -1 & x \in Q, y = ax - b \\ 0 & \text{otherwise} \end{cases}, \tag{2}$$

second, for all $a \in \mathbb{F}_q$ and $b \in -Q$

$$v_{a,b}[x,y] = \begin{cases} +1 & x = 0, y = b \\ +1 & x \in -Q, y = ax + b \\ -1 & x \in -Q, y = ax - b \\ 0 & \text{otherwise} \end{cases}, \tag{3}$$

and finally, for each $c \in \mathbb{F}_q^*$

$$u_c[x,y] = \begin{cases} +1 & x = c, y \in \mathbb{F}_q \\ 0 & \text{otherwise} \end{cases}. \tag{4}$$

Lemma B.1. *The vectors defined in Eqs. (2), (3) and (4) are pairwise orthogonal and their supports form a* $(q,2)$*-design.*

Proof. It is an easy computation to see that the support of each of the vectors has cardinality q. We now argue that they are pairwise orthogonal. There are several cases depending on the two rows under consideration:

1. $v_{a,b}$ and $v_{a',b'}$: if one comes from Eq. (2) and the other from Eq. (3) then the supports are disjoint. So we assume both come from Eq. (2) or both come from Eq. (3).

   (a) Both come from Eq. (2) and $b = b'$: we have one intersection $(0, b)$ (which contributes $+1$ to the inner product) and exactly one of the following two intersection points: $(x = -2b/(a - a'), ax + b = a'x - b)$ or $(x = 2b/(a - a'), ax - b = a'x + b)$, which contributes -1 to the inner product. We have exactly one because the two x-values are negations of each other, and non-zero, so exactly one is in Q.

   (b) Both come from Eq. (2) and $b \neq b'$: we have exactly one of the following two intersection points: $(x = (b' - b)/(a - a'), ax + b = a'x + b')$ or $(x = (-b' + b)/(a - a'), ax - b = a'x - b')$, which contributes $+1$ to the inner product, and exactly one of the following two intersection points: $(x = (b' + b)/(a - a'), ax - b = a'x + b')$ or $(x = (-b' - b)/(a - a'), ax + b = a'x - b')$, which contributes -1 to the inner product. For each pair, there is exactly one of the pair of possible intersection points because the two x-values are negations of each other, and non-zero, so exactly one is in Q.

   (c) Both come from Eq. (3) and $b = b'$: identical to case (1a) above, with $-Q$ in place of Q.

   (d) Both come from Eq. (3) and $b \neq b'$: identical to case (1b) above, with $-Q$ in place of Q.

2. u_c and u'_c: these have disjoint supports for $c \neq c'$.

3. $v_{a,b}$ and u_c: if $c \in Q$, then the support of u_c intersects the support of $v_{a,b}$ only if $v_{a,b}$ comes from Eq. (2), and then we get one intersection at point $(x = c, ax + b)$ which contributes a $+1$ to the inner product, and one intersection at point $(x = c, ax - b)$ which contributes a -1 to the inner product. If $c \in Q$, then the support of u_c intersects the support of $v_{a,b}$ only if $v_{a,b}$ comes from Eq. (3), and we have an identical argument, with $-Q$ in place of Q.

This is a complete enumeration of cases, and in no case did we have more than 2 intersection points. □

We conclude this section with a question: are these matrices related in some way to the DFT matrix over some family of non-abelian groups (e.g. the affine group $\mathbb{F}_q^* \ltimes \mathbb{F}_q$), or are they indeed completely different from the unitaries seen before in quantum algorithms?

C Converting a distributional oracle problem into a standard oracle

We include this section for completeness, a similar proof appears in [Aar10a].

We have two ensembles of random variables $D_1 = \{D_{1,n}\}, D_2 = \{D_{2,n}\}$ over $(N = 2^n)$-bit strings for which BQLOGTIME can distinguish the two distributions but AC_0 cannot. Then when D_1 and D_2 are viewed as distributions on (truth-tables of) *oracles*, there is a BQP oracle machine that distinguishes the two distributions, but no PH oracle machine can distinguish them. Specifically, we have that there exists a BQP oracle machine A for which

$$\Pr[A^{D_1}(1^n) = 1] - \Pr[A^{D_2}(1^n) = 1] \geqslant \varepsilon$$

while for every PH oracle machine M,

$$\Pr[M^{D_1}(1^n) = 1] - \Pr[M^{D_2}(1^n) = 1] \leqslant \delta < \varepsilon,$$

(here we use standard techniques – see, e.g., [Hås87] – which show that on any fixed input, the output of a PH machine as a function of the oracle can be seen as an AC_0[7] circuit) and we have $\varepsilon > \delta$ for sufficiently large $n \geqslant n_0$.

We now convert the distributions on oracles into a single oracle O for which $BQP^O \not\subseteq PH^O$. Let L be a uniformly random unary language in $\{1\}^*$. For each n, if $1^n \in L$, sample a 2^n-bit string x from D_1 and define oracle O restricted to length n so that x is its truth table; otherwise sample a 2^n-bit string x from D_2 and define oracle O restricted to length n so that x is its truth table.

First, note that

$$\Pr[A^O(1^n) = L(1^n)] \quad = \quad (1/2) \cdot \Pr[A^{D_1}(1^n) = 1] + (1/2) \cdot \Pr[A^{D_2}(1^n) = 0] \geqslant 1/2 + \varepsilon/2.$$

Now fix any PH machine M, and note that for sufficiently large n,

$$\Pr[M^O(1^n) = L(1^n)] \quad = \quad (1/2) \cdot \Pr[M^{D_1}(1^n) = 1] + (1/2) \cdot \Pr[M^{D_2}(1^n) = 0] \leqslant 1/2 + \delta/2.$$

Consequently, since $\varepsilon > \delta$ there is a fixed choice for the oracle at length n such that $L(1^n) = A^O(1^n) \neq M^O(1^n)$, for sufficiently large n.

Fix such a choice for the oracle at length n, and consider another PH machine M'. By the same argument, we can find another sufficiently large input length n' where $L(1^{n'}) = A^O(1^{n'}) \neq M^O(1^{n'})$.[8]

Continuing in this way, we obtain a single oracle such that for any PH machine M there exists some n for which $A^O(1^n) \neq M^O(1^n)$.

[7] Recall that we are using "AC_0" to refer to size $\exp(\mathrm{poly}\log n)$-size constant depth circuits in this paper.

[8] We have assumed that our machines, on an input of length n, only query the oracle at inputs of length n. This can be ensured by working with input lengths that are sufficiently spread out (so that the machine cannot afford to formulate queries to the next largest length, and so that the oracle at shorter lengths can be hardcoded.)