On Quantum Computing and Pseudorandomness

Thesis by

William Jason Fefferman

In Partial Fulfillment of the Requirements

Masters Degree in Computer Science

California Institute of Technology

Pasadena, California

2010

(Submitted June, 2010)

Abstract

The relationship between classically efficient verification and quantum computing is one of the most important and least well-understood questions in the theory of computation. In particular, is there a problem that can be solved efficiently on a quantum computer that cannot be verified? In this thesis we give evidence that $BQP \not\subset PH$, relating the classes of languages decidable with a quantum computer to a generalization of NP. In so doing we connect a question in pseudorandomness, first studied in [BSW03] to the problem of finding an oracle relative to which $BQP \not\subset PH$. The primary technical challenge is to construct a unitary matrix, realized by an efficient quantum circuit and whose rows are supported on nearly disjoint subsets. Using this matrix and assuming the validity of the aforementioned question in pseudorandomness, we show an instantiation of the Nisan-Wigderson pseudorandom generator that can be broken with quantum computers, but not with the relevant mode of classical computation.

0.1 Introduction

Let U_t denote a random variable distributed uniformly on t-bit binary strings. It has long been a goal to explicitly construct "pseudorandom generators":

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

that stretch a short uniformly generated "seed" into a longer output string, so that $f(U_t)$ is computationally indistinguishable from U_m. Further, one can ask to what degree this computational indistinguishability is dependent on particular models of computation (with respect to existing constructions). Here we give evidence that quantum computers can distinguish particular instantiations of "pseudorandom" distributions that are provably indistinguishable from uniform by classical circuits. In so doing, we make progress on finding a relativized setting in which *BQP* $\not\subset$ *PH*, an infamous question first addressed by Bernstein and Vazirani in 1993 [BV97]. In addition, we connect this problem to that of the "Generalized Nisan-Linial" Conjecture [Aar10b] relating such results to a generalization of a recently proven result of Braverman [Bra10].

More formally, we refer to AC_0 as the class of constant depth and-or-not circuits of polynomial size and unbounded fan-in. We will work with the Nisan-Wigderson pseudorandom generator against such circuits, with MAJORITY as its hard function. In particular we will need two standard definitions:

Definition 0.1.1 ([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 0.1.2 ([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.*

In general 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. The MAJORITY function on ℓ bits is known to be hard for AC_0 circuits, and it remains hard to compute the MAJORITY correctly on more than a $1/2 + \tilde{\Theta}(\ell^{-1/2})$ fraction of the inputs [Smo93]. Furthermore such approximation of MAJORITY is realized by the simple algorithm that outputs an arbitrary bit of the input. However, we make the following conjecture:

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* $\exp(\mathrm{polylog}(m))$,

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

Note that in this work we will abuse notation slightly and refer to constant depth circuits of size at most $\exp(\mathrm{poly}\log m)$ as "AC_0". By the standard argument from [Yao82, GM84], 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 $m \leqslant o(\sqrt{\ell})$; 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. While we don't know how to prove this conjecture we believe it to be true and give intuition in our Conclusion, section 0.9.

We summarize our three main results, which together make Conjecture 1 interesting and worthy of further study:

- We make progress on finding an oracle relative to which $BQP \not\subset PH$, which is a major open question in quantum computing. We give a new approach to resolving this question (e.g.,

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

by proving Conjecture 1), which is formally easier than the previous approach of Aaronson [Aar10b]. We also give a potential line of reasoning in support of this conjecture.

- We find a surprising consequence to Aaronson's "Generalized Linial-Nisan" conjecture (for which previously little evidence was known). In particular, we show that this conjecture *implies* Conjecture 1. Thus our implication shows a possible path for proving this relativized separation, or disproving Aaronson's conjecture, by showing an AC_0 circuit for breaking the Nisan-Wigderson generator based on the MAJORITY function. Furthermore, our conjecture is a natural question in pseudorandomness that is of independent interest in many other contexts (e.g., [BSW03]).

- We generalize [Aar10b], which shows that a "forrelated" distribution can be efficiently distinguished from uniform by the quantum computer, which is possibly hard for AC_0. We show a framework for which any quantumly computable unitary gives rise to a distribution that can be distinguished from uniform quantumly. Further, we show how, based upon this framework, we can use it to break an instantiation of the Nisan-Wigderson pseudorandom generator, assuming Conjecture 1.

  We also note that the unitaries that form the basis of our quantum algorithms don't seem to resemble unitaries useful for other quantum algorithms, and show a task that gives an "exponential" quantum speedup over the best classical algorithms. Interestingly the desired property in these unitaries is precisely their natural extremal combinatorial properties, and we wonder if they can be useful elsewhere.

Finally, this exponential quantum speedup mentioned above gives us other *unconditional* oracle separations. [Aar10b] has shown that the classes SZK and BPP_{path} require exponentially many queries to distinguish ε-almost k-wise independent distributions from uniform and therefore, our constructions 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.

0.2 Quantum preliminaries

The state of an n-*qubit quantum system* is described by a unit vector in $\mathcal{H} = (\mathbb{C}^2)^{\otimes n}$, a 2^n-dimensional complex Hilbert space, endowed with the standard Hilbert-Schmidt inner product. As per the literature we will denote the standard orthogonal basis vectors of $\mathcal{H}$ by $\{|v\rangle\}$ for $v \in \{0,1\}^n$.

In accordance with the laws of quantum mechanics, transformations of states are described by unitary transformations acting on $\mathcal{H}$, where a *unitary transformation* over $\mathcal{H}$ is a linear transformation specified by a $2^n \times 2^n$ square complex matrix U, such that $UU^* = I$, where U^* is the conjugate transpose. Equivalently, the rows (and columns) of U form an orthonormal basis. 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.

We will also need the concept of *projective measurement*, which given an orthonormal basis O for $\mathcal{H}$ associates a value designated by a real number r_i for each basis vector $|v_i\rangle \in O$. Suppose our quantum system is in the state $|\phi\rangle \in \mathcal{H}$. We define $\{\Pi_{r_j}\}$ to be a collection of projection operators, that project into the subspace spanned by the designated $|v_j\rangle$ for all v_j associated to the same output value r_j. When we measure our system, we obtain the respective outcome r_j with probability $|\Pi_{r_j}|\phi\rangle|^2$ and the resulting state of the system becomes $\frac{\Pi_{r_j}|\phi\rangle}{|\Pi_{r_j}|\phi\rangle|}$.

For example suppose our Hilbert space $\mathcal{H}$ can be decomposed into orthogonal subspaces $\mathcal{H} = S_1 \oplus S_2$. When we measure $\{\Pi_1, \Pi_2\}$ which project into the orthogonal subspaces S_1 and S_2, it causes the system to collapse to $\Pi_1|\phi\rangle/|\Pi_1|\phi\rangle|$ or $\Pi_2|\phi\rangle/|\Pi_2|\phi\rangle|$ with probability $|\Pi_1|\phi\rangle|^2$ and $|\Pi_2|\phi\rangle|^2$ respectively.

An efficient *quantum circuit* consists of at most $poly(n)$ local unitaries, followed by a measurement.

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 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 next two paragraphs are from [FU10].)

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.

Lemma 0.2.1. *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 $|000\cdots 0\rangle$, there is a $\mathrm{poly}(n)$ size uniform quantum circuit that applies U to the first two registers and leaves the third unchanged.*

Proof. 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

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

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 $\ell = 1, 2, 3, \dots, n^k$: apply the "controlled unitary" that consults the ℓ-th component of the third register, and applies the specified gate to qubits $(a_1, a_2, \dots, a_b)$ of the second register (again, $(a_1, a_2, \dots, a_b)$ 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.

□

0.3 Motivation: Quantum Computing vs Classical Nondeterminism

The relation between the class of decision problems admitting efficient quantum algorithms, BQP and the class of decision problems with (classically) efficient verification procedures, NP is a question of vital importance and yet remains wide open. It is clear that the two classes have a non-trivial intersection, every problem that can be solved efficiently with a classical computer can both be verified efficiently and can be solved efficiently on a quantum computer. Additionally, Shor's famous factoring algorithm [Sho94] and other examples of hidden subgroup problems over certain classes of groups [Hal07] demonstrate that there are also problems in this intersection not known to be in P. However, can every problem that can solved efficiently on a quantum computer be verified efficiently? Conversly, can every efficiently verifiable problem be efficiently solved on a quantum computer? Frustratingly, nearly two decades after such questions were proposed (e.g.,[BV97]), little can be said in support of either direction.

In [Sho94] a first step was taken to give evidence for $BQP \not\subset NP$ in the form of an oracle separation, as a direct consequence of Shor's algorithm. Let N be some integer exponential in input length, n. The idea of this separation is to give both machines oracle access to a function, $f : [N] \rightarrow [N]$ promised to be either:

1. Periodic, so that $\forall x\ f(x) = f(x + P)$ for some period $P < N$ but still exponential in n or,
2. Completely arbitrary, with no such periodicity at all.

The decision question that obtains the separation is to distinguish which of the two is the case. Note that Shor's algorithm implies an efficient algorithm to solve this problem in $O(polylog\ N)$ queries. However the problem is not in NP. This is because there is no polynomial length witness that verifies that a given function is *not* periodic [BBBV97].

Given that we know $\exists O\ BQP^O \not\subset NP^O$, the natural open question has been to strengthen this result, separating BQP from more and more powerful classical classes. Since the best known upper bound for BQP is $P^{\sharp P}$, we are interested in classes between NP and $P^{\sharp P}$.

In 2003, Watrous showed an oracle separating BQP from MA using the "group nonmembership problem" [Wat00]. However, Babai has since shown that this problem is contained in AM and so cannot be used to show stronger separations [BBS09]. Finding such an oracle separation from AM or PH has remained wide open.

Since $NP = MA = AM$ under widely believed derandomization assumptions [NW94, KvM02], our failure to obtain these separations is yet another indication of our lack of understanding the relationship between NP and BQP.

In this paper we will find it convenient to speak 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 a quasi-polynomial sized (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 $O(\log N)$ local-unitaries.

Definition 0.3.1 ($BQLOGTIME$)**.** *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* $\text{poly}\log(N)$ *gates.*

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 0.10, such a distribution can be easily converted to a proper oracle O for which $BQP^O \not\subset PH^O$.

0.4 Framework

Here we give an informal description of the main ideas necessary to achieve our separation. In order to prove our desired separation, we need to show an oracle problem which is not in PH as per our conjecture, yet has an efficient quantum algorithm. To do this we define the following oracle problem:

MAJORITY-CHECKING : Given oracle access to a string over $\{\pm 1\}^{2t}$, which we think about as divided into two strings $x, z \in \{\pm 1\}^t$. We will think about x as the input to the NW generator. Then the problem is to distinguish between the following two distributions:

1. x and z are generated uniformily[2] over $\{\pm 1\}^{2t}$, *or*
2. x is uniform over $\{\pm 1\}^t$ and z is a vector of signs of a unitary matrix U (with entries in $\{0, 1, -1\}$)[3] applied to x

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; which is simply the MAJORITY (with a fixed pattern of inputs negated) function applied to 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 quasi-polynomial sized AC_0. We will show an explicit construction of such a unitary U.

Our Conjecture 1 implies that the NW generator with certain parameters fools AC_0, which is one part of the overall argument, readily implying MAJORITY-CHECKING (with respect to a particular U) has no AC_0 circuit. Clearly we also need to exhibit a $BQLOGTIME$ algorithm that "breaks" this instantiation of the NW generator and thus solves MAJORITY-CHECKING .

Roughly, in order to distinguish the two cases the quantum algorithm queries the x into the phases, applying U, multiplying the second string into the phases, and measuring in the Hadamard basis.

We want to construct U so that the rows are supported on subsets that are nearly disjoint, forming an $(\ell, p)-$design where p is a constant. This is a different setting of the parameters than usual, but in our construction we compensate for this because the number of sets in the design is also

[2] In fact, we will show that the quantum algorithm works even when z is distributed according to any arbitary distribution (independent from x)

[3] We ignore normalization factors in this discussion.

small ($poly(\ell)$ instead of $\exp(\ell)$), and for these parameters we present a geometric construction, where the sets are the characteristic vectors of pairs of lines in an affine plane. In prior explicit constructions of $(\ell, p)-$designs such as [NW94, HR03] we cannot simply attach $+/-$ signs to make their characteristic vectors orthogonal. However, massive symmetries in this construction allow us to assign signs to the elements of each set to achieve this pairwise orthogonality of their characteristic vectors, which results in a matrix whose row vectors are orthogonal– giving us unitarity. In our construction these set systems have only $t/2$ sets, so the resulting unitary will have the required properties among half of their rows, but we show that the quantum algorithm can be adapted so as to be resilient to this change.

In Section 0.7.2 we give a *local decomposition* (see Section 0.2 for the formal definition) of these unitaries, which is necessary to have an *efficient* quantum algorithm.

0.5 Quantum algorithm

In this section we describe a quantum algorithm that solves MAJORITY-CHECKING with high probability. Let A be any $N \times N$ real unitary matrix. For generality we consider any such A matrix, and in section 0.7 construct a particular real unitary that is relavent for our purposes (i.e., to establish classical hardness).

Define the random variable $D_{A,M} = (x, z)$ distributed on $\{\pm 1\}^{2N}$ by picking $x \in \{\pm 1\}^N$ uniformly, and setting the next N bits to be $z \in \{\pm 1\}^N$ defined by $z_i = \text{sgn}((Ax)_i)$ for $i \leqslant M$ and z_i independently and uniformly random in $\{\pm 1\}$ for $i > M$. Likewise, let X_{2N} be the random variable distributed over $\{\pm 1\}^{2N}$, so that the first half of coordinates, x is generated uniformly at random, by flipping N unbiased random coins and z is distributed according to an arbitrary distribution (but independently from x)[4] over $\{\pm 1\}^N$.

For convenience we think of $M = N$ initially; we analyze the general case because we will

[4] Note that X_{2N} generalizes the “all-uniform” distribution, U_{2N}, as promised.

eventually need to handle $M = N/2$.

Theorem 0.5.1. *Let $N = 2^n$ for an integer $n > 0$, and let $M = \Omega(N)$. For every real $N \times N$ unitary A, there is a $BQLOGTIME$ algorithm Q_A that distinguishes $D_{A,M}$ from X_{2N}; i.e., there is some constant $\varepsilon > 0$ for which:*

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

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

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

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 A to obtain $\frac{1}{\sqrt{N}} \sum_{i \in \{0,1\}^n} (Ax)_i |i\rangle$.
3. Multiply z into the phase to obtain $\frac{1}{\sqrt{N}} \sum_{i \in \{0,1\}^n} z_i (Ax)_i |i\rangle$.
4. Define vector w by $w_i = \frac{1}{\sqrt{N}} z_i (Ax)_i$. 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 is distributed uniformly and z is distributed according to an arbitrary distribution. This is a consequence of in quantum computation to cancel the changes in sign after application of the unitary A applied to x. In particular, we will show that we can distinguish the instantiation of the NW distribution, $D_{A,M}$, from a *"half-uniform"* distribution where the second half of coordinates z has all probability mass on a single string, $\tilde{z} \in \{\pm 1\}^N$. It follows that we can take any distribution on z and still distinguish the distributions.

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

Additionally, define U to be the unitary matrix obtained by multiplying the diagonal matrix with $\tilde{z}$ on the diagonal by A, e.g., U is the unitary applied to the first-half of coordinates after querying the second-half. Let u_j be the sum of the jth column of U. We proceed to show that the acceptance probability of the quantum algorithm given an *uncorrelated* distribution is smaller than any constant (the acceptance probability of the quantum algorithm after the final measurement given a NW distribution).

In this case, we are interested in the post-measurement acceptance probability, given by the square of the mass on the 0^n basis vector after step 4 in the algorithm above:

$$\begin{aligned} E\left[(Hw)_0^2\right] =& E\left[\left(\frac{1}{N}\sum_{i}^{N}\sum_{j}^{N}\tilde{z}_i a_{i,j} x_j\right)^2\right] && (1)\\ =& \frac{1}{N^2}E\left[\left(\sum_j u_j x_j\right)^2\right] && (2)\\ =& \frac{1}{N^2}E\left[\sum_{j,j'} u_j u_{j'} x_j x_{j'}\right] && (3)\\ =& \frac{1}{N^2}\sum_{j,j'} E\left[u_j u_{j'} x_j x_{j'}\right] && (4)\\ & && (5)\end{aligned}$$

Clearly, this expectation is equal 0 if $j \neq j'$, thus

$$\begin{aligned} =& \frac{1}{N^2}E\left[\sum_j {u_j}^2\right] && (6)\\ =& \frac{1}{N^2}\cdot N = \frac{1}{N} \leqslant o(1) && (7)\end{aligned}$$

Where (7) follows from unitarity.[6]

We next argue that the acceptance probability is large in case (x, z) is distributed as $D_{A,M}$. We are interested in analyzing the expectation value of each individual w_i. Note that the w-vector is uniformly distributed for $i > M$, since the corresponding entry of z is uniform. Thus for these i, $E[w_i] = 0$. For this reason we will be interested in $i \leqslant M$. Then

$$\begin{aligned} E\left[w_i\right] &= E\left[\left|\frac{1}{\sqrt{N}}\left(Ax\right)_i\right|\right] = E\left[\left|\sum_j^N \frac{1}{\sqrt{N}} a_{i,j} x_j\right|\right] && (8) \\ &= E\left[\sqrt{\frac{1}{N}\left(\sum_j a_{i,j} x_j\right)^2}\right] && (9) \\ &= \frac{1}{\sqrt{N}} E\left[\sqrt{\sum_{j,j'} a_{i,j} x_j a_{i,j'} x_{j'}}\right] && (10) \\ &= \frac{1}{\sqrt{N}} E\left[\sqrt{\sum_j a_{i,j}{}^2}\right] && (11) \\ &= \frac{1}{\sqrt{N}} E\left[1\right] = \frac{1}{\sqrt{N}} && (12) \end{aligned}$$

Where equality (11) follows from the same argument as the prior part of the proof (i.e., the expectation is easily verified to be 0 unless $j = j'$) and (12) follows from the unitarity of A.

Then $E[\sum_i w_i] = M \cdot \frac{1}{\sqrt{N}} = \Omega(\sqrt{N})$, 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 $\Omega(1)$. □

[6] More precisely, this is a consequence of unitary matrices preserving the L_2-norm. Formally, let $\mathbf{1}$ be the all-1's row-vector and let V be any complex unitary. Then we know $||\mathbf{1}V||_2 = ||\mathbf{1}||_2 \Rightarrow \sum_{j=1}^N |v_j|^2 = N$

0.6 Classical hardness

In this section we discuss the classical difficulty in distinguishing the uniform distribution from the NW distribution with MAJORITY as the hard function. Recall that there are no AC_0 circuits that can exactly compute the parity or MAJORITY function [FSS84]. These results by themselves are not useful directly for our desired separation, because it is known that any quantum circuit needs to make $\Omega(2^n)$ queries to compute the parity of n bits [BBC$^+$01]. What other tasks are hard for AC_0? Recently, Braverman [Bra10] showed that such circuits are quite limited in their ability to distinguish uniformly distributed input bits from bits that are distributed uniformly if we look at any small subset of the bits, but could potentially be correlated across larger scales.

Formally, let D be a discrete random variable with values over $\{\pm 1\}^r$. We give two defintions:

Definition 0.6.1. *D is k-wise independent if every restriction of D to k coordinates is uniform over $\{\pm 1\}^k$*

Definition 0.6.2. *A function $f : \{\pm 1\}^r \rightarrow \{\pm 1\}$ is ε-fooled by D if:*

$$|\Pr_{x \sim D}[f(x) = 1] - \Pr_{x \sim U_r}[f(x) = 1]| < \varepsilon$$

Braverman recently [Bra10] proved the ”Linial-Nisan“ Conjecture:

Theorem 0.6.3. *AC_0 circuits of size $m = poly(n)$, depth $d = O(1)$, are ε-fooled by $\log(m/\varepsilon)^{O(d^2)}-$independence.*

The proof of this theorem answered a famous conjecture of Nisan and Linial [NW94] from 1994. However, what if D is not exactly k-wise independent but can deviate by some multiplicative factor?

Definition 0.6.4. *[Aar10b] D is ε-*almost k-wise independent *if for every k distinct indices $i_1, i_2, \ldots, i_k \in [r]$, and every $\alpha_1, \alpha_2, \ldots, \alpha_k \in \{\pm 1\}$ we have:*

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

Then the Generalized Nisan-Linial Conjecture states:

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*[7]*. 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.

First, we will generalize this to the distribution NW_D^f with f a generic boolean function.

Let $f : \{\pm 1\}^n \to \{\pm 1\}$ be a boolean function, we designate the restriction $R = (\rho \subseteq [n], a : [\rho] \to \{\pm 1\})$ as the pairing of the indices we fix and the respective values assigned to the indices by the mapping, a. Given f and k, the maximum size of restriction set ρ, we're interested in the maximum bias (away from 1/2) of the function under such restrictions, where:

Definition 0.6.5. $bias(f) = \Pr[f(x)=1] - \frac{1}{2} = \frac{\Pr[f(x)=1]-\Pr[f(x)=-1]}{2} = \frac{E[f(x)]}{2}$

Let $f = \sum\limits_{S \subseteq [n]} \widehat{f}(S)\chi_S$ be the representation of f in the fourier basis,

where each χ_S is the product over the variables at indices specified by the set S. Clearly $E[f] = \widehat{f}(\emptyset)$ because $\widehat{f}(\emptyset) = \langle f, \chi_\emptyset \rangle = \frac{1}{2^n}\sum_x f(x)$.

Thus $bias(f) = \frac{\widehat{f}(\emptyset)}{2}$.

[7] 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.

We're interested in the quantity:

Definition 0.6.6. *$\phi_f(k) = max_R(|bias f_R|)$, the maximum bias of f taken over all restrictions R, i.e., over both the subset of variables and their assignments. Clearly $\phi_f(k) = max_R|\frac{\widehat{f|_R(\emptyset)}}{2}|$*

And we claim:

Lemma 0.6.7. $\widehat{f|_R}(\emptyset) = \widehat{f}(\emptyset) + \sum_{S\subseteq\rho}[\widehat{f}(S)\prod_{i\in S} a(i)]$

This is true simply because each fourier term fixed by the restriction becomes correlated with the constant fourier term, $\widehat{f}(\emptyset)$. Note that in the monotonic case the restriction R achieving the maximum of $\phi_f(k)$ is clearly the function mapping all indices of ρ to $+1$ and we need only maximize over subsets ρ but in the general case we may reach a maximum bias with mixed values of a.

Theorem 0.6.8. *Adapted from [FU10]: Let $\mathcal{D} = \{S_1, S_2, \ldots, S_m\}$ be an (ℓ, p) design in a universe of size t, and $f : \{\pm 1\}^\ell \to \{\pm 1\}$. Then for every $k < o(\ell^{1/4}p^{-1/2})$, the jointly distributed random variable*

$$(C, D) = (U_t, NW^f_{\mathcal{D}}(U_t))$$

is $O(k\phi_f(pk))$-almost k-wise independent.

We will show 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 $\phi_f(pk)$ which we can show for MAJORITY to be $O(pk/\sqrt{\ell})$. Clearly we only have to worry about the case when some of the conditioned coordinates are *outside* of the first t coordinates, since the first t coordinates are exactly independent. To analyze this case for these coordinates in position $t+i$ we replace the conditioning on this coordinate with those coordinates in the support of this coordinate. Namely, for each coordinate we condition on every coordinate in the set S_i of the $(\ell, p)-$design. This makes sense because these support coordinates completely determine the $t+i$th coordinate, and this is true for all coordinates after t. Since at most p of these can affect the bias of the k-th coordinate, we compute this quantity conditioning on up to $p(k-1)$ bits instead of $(k-1)$. Our proof is intrinsically combinatorial and involves only simple calculation, while the analogous proof in [Aar10b] showing that a *forrelated* distribution is almost-k-wise independent) takes a rather involved calculation to bound the measure of a space of independent Gaussians restricted to an affine subspace, and then converts to a discrete setting.

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

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

which we write as

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

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 $\leqslant pk$.

Then by definition, after conditioning on at most pk coordinates:

$$\rho_w \leqslant 1/2 + \phi_f(pk)$$

and

$$\rho_w \geqslant 1/2 - \phi_f(pk)$$

so,

$$\rho \leqslant (1/2 + \phi_f(pk))^k \leqslant [(1/2)(1 + 2\phi_f(pk))]^k \leqslant 2^{-k}(1 + 2k\phi_f(pk))$$

and symmetrically,

$$\rho \geqslant (1/2 - \phi_f(pk))^k \geqslant [(1/2)(1 - 2\phi_f(pk))]^k \geqslant 2^{-k}(1 - 2k\phi_f(pk))$$

Corollary 0.6.9. *$\phi_{majority}(pk) = O(pk/\sqrt{\ell})$, and it follows from Theorem 0.6.8, for every $k < o(\ell^{1/4}p^{-1/2})$, the jointly distributed random variable*

$$(C, D) = (U_t, NW_{\mathcal{D}}^{\textit{majority}}(U_t))$$

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

Let $|S_w| = \ell$, and 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

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

Using Stirling's Approximation we obtain $\binom{\ell}{r} \leqslant O(\frac{2^\ell}{\sqrt{\ell}})$ for all r, so we get the upper bound of

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

and symmetrically

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

The corollary now follows from Theorem 0.6.8, taking $\phi_{majority}(pk) = O(pk/\sqrt{\ell})$.

□

Corollary 0.6.9 immediately implies that MAJORITY-CHECKING has exponential classical query complexity. To show this we cite a theorem of Aaronson:

Theorem 0.6.10. *Lemma 20 from [Aar10b]: Suppose a probability distribution $\mathcal{D}$ over oracle strings is δ-almost k-wise independent. Then no bounded-error postselected classical machine running in less than k-steps can distinguish $\mathcal{D}$ from the uniform distribution with bias larger than 2δ.*

Since we have already demonstrated a quantum algorithm solving MAJORITY-CHECKING with constant queries, we have given an example of a problem with an "exponential quantum speedup" over randomized classical computation.

0.7 Unitary matrices with large, nearly-disjoint row supports

Note that this section is largely verbatim from [FU10]. In this section we construct unitary matrices A with the additional property that all or "almost all" of the row supports are large and have bounded intersections. Define $S(A, i)$ to be the support of the ith row of matrix A. 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 0.5.1, and at the same time $D_{A,M}$ can be seen as an NW distribution that by Conjecture 1 fools quasi-polynomial size AC_0 (see Section 0.8 for the precise statement).

0.7.1 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 $\mathbb{F}_q \times \mathbb{F}_q$:

$$v_{a,b}[x, y] = \begin{cases} -1 & y = ax + b \\ +1 & y = ax + \phi(b) \end{cases} \tag{13}$$

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 0.7.1. *The vectors defined in Eq. (13) 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. □

0.7.2 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. (13) as $q^2/2$ of its q^2 rows. We recall that $q = 2^n$ for an integer $n > 0$.

Proposition 0.7.2. *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 0.2.1. It is then easy to verify that $B = B_1 B_2 B_3 \cdots B_n$. □

Lemma 0.7.3. *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 is efficiently quantumly computable. By Lemma 0.2.1 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 0.7.4. *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. (13) as $q^2/2$ of its rows*[8].

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

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

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

and that

$$FD'_cF^{-1} = \frac{1}{\sqrt{q}}S_c - \frac{1}{\sqrt{q}}J.$$

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. (13) as the "top-halves" of each successive set of q rows of the large matrix. □

0.8 Putting everything together

Let A be the matrix of Theorem 0.7.4, and set $N = q^2$ and $M = N/2$. By Theorem 0.5.1, there is a $BQLOGTIME$ algorithm that distinguishes $D_{A,M}$ from U_{2N}, solving MAJORITY-CHECKING .

By Lemma 0.7.1, 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, quasi-polynomial-size circuit can distinguish $D_{A,M}$ from U_{2N}, which completes the argument.

Unfortunately the standard NW argument fails, which is why we must rely on Conjecture 1. The standard argument defines $2N + 1$ "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 0.7.4.

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.

0.9 Conclusion

As a result of our work and the work of [Aar10b] the Generalized Nisan-Linial conjecture yields two separate relativized settings in which quantum computers can efficiently solve problems outside of the polynomial hierarchy. Our proof that the Generalized Nisan-Linial Conjecture implies Conjecture 1 also gives us a nontrivial connection between two seemingly unrelated problems in circuit complexity– either the "hybrid loss" incurred by the standard "distinguishability implies predictability" argument [Yao82] can be avoided, or else AC_0 circuits are able to distinguish "approximate" $(polylog\ N)$-wise independence from uniformly distributed bits, even though it is known they cannot distinguish exact $(polylog\ N)$-wise independence. This gives both a new hope for proving a fundamental problem in quantum computing, or alternatively a completely new approach for disproving a natural conjecture about the classes of distributions that fool AC_0 circuits[9]

Both MAJORITY-CHECKING in this paper and FOURIER-CHECKING [Aar10b] give rare examples of oracle problems that can be solved in a constant number of quantum queries, yet requires exponential classical query complexity. Thus, these give rare examples of oracle problems where the quantum algorithm obtains an exponential speedup over classical computation. This is a stronger separation than from Recursive Fourier Sampling, the candidate problem proposed for the separation in [BV97]. It is known that this speedup can only be quasi-polynomial (i.e., $O(n^{\log n})$) [Aar02].

We believe Conjecture 1 to be true, and offer the following intuition. For simplicity, let's assume that the distributions we are trying to distinguish are N^2 copies of the random variable D where $D = (U_N, majority(U_N))$, and N^2 independent copies of the random variable U_{N+1}

[9] Note that very recently the GLN has been proven false, as a counterexample is known for depth 3 AC_0 circuits [Aar10a]. Despite this, the conjecture remains open for depth 2 circuits, and our implication may still be useful to study the GLN at this depth.

distributed uniformly on $N + 1$ bits. This is the result of taking the NW construction where the underlying subsets are *completely disjoint*. Here, unlike the prior setting, it seems that a hypothetical distinguisher must look at each block separately, since they are completely independent of each other. Now the AC_0 circuit can collect a "noisy bit" from each block which gives some information about whether it is distributed according to D or U_{N+1}– in the case of the former each bit is 1 with probability $1/2 + \Theta(1/\sqrt{N})$ and the latter case the probability is $1/2$. Note that to use this information to make a decision, it seems a distinguisher must aggregate these noisy bits across the N^2 copies– presumably this involves a task which itself is hard for AC_0.

However, it is worth noting that any proof of Conjecture 1 will need to respect a tension intrinsic in obtaining our desired separation. Most known lower bounds against AC_0 circuits are proven by showing that there is no low-degree approximating real polynomial, which by results of Razborov and Smolensky e.g., [Smo93] we know every AC_0 circuit has. However, every quantum algorithm that computes a function $f : \{\pm 1\}^n \to \{\pm 1\}$ using T-queries to f has a real polynomial p of degree $2T$ so that $|f(x) - p(x)| \leqslant o(1), \forall x \in \{\pm 1\}^n$ [BBC+01]. Thus to prove our conjecture that AC_0-circuits are fooled by such classes of distributions we cannot use any method that pertains exclusively to low-degree polynomials. This highlights a reason this problem is so notorious and likely needs dramatically new ideas for resolution.

Bibliography

[Aar02] Scott Aaronson. Quantum lower bound for recursive fourier sampling. *Electronic Colloquium on Computational Complexity (ECCC)*, (072), 2002.

[Aar10a] S. Aaronson. A counterexample to the Generalized Linial-Nisan conjecture. *ECCC Report 109*, 2010.

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

[BBBV97] 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.

[BBC$^+$01] Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. *J. ACM*, 48(4):778–797, 2001.

[BBS09] László Babai, Robert Beals, and Ákos Seress. Polynomial-time theory of matrix groups. In Michael Mitzenmacher, editor, *STOC*, pages 55–64. ACM, 2009.

[Bra10] Mark Braverman. Polylogarithmic independence fools ac^0 circuits. *J. ACM*, 57(5), 2010.

[BSW03] Boaz Barak, Ronen Shaltiel, and Avi 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.

[BV97] Ethan Bernstein and Umesh V. Vazirani. Quantum complexity theory. *SIAM J. Comput.*, 26(5):1411–1473, 1997.

[FSS84] Merrick L. Furst, James B. Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical Systems Theory*, 17(1):13–27, 1984.

[FU10] Bill Fefferman and Chris Umans. On pseudorandom generators and the bqp vs ph problem. *Manuscript*, 2010.

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

[Hal07] Sean Hallgren. Polynomial-time quantum algorithms for pell's equation and the principal ideal problem. *J. ACM*, 54(1), 2007.

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

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

[KvM02] Adam Klivans and Dieter van Melkebeek. Graph nonisomorphism has subexponential size proofs unless the polynomial-time hierarchy collapses. *SIAM J. Comput.*, 31(5):1501–1526, 2002.

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

[Sho94] Peter W. Shor. Polynominal time algorithms for discrete logarithms and factoring on a quantum computer. In Leonard M. Adleman and Ming-Deh A. Huang, editors, *ANTS*, volume 877 of *Lecture Notes in Computer Science*, page 289. Springer, 1994.

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

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

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

0.10 Converting a distributional oracle problem into a standard oracle, from [FU10]

Let $D_1 = \{D_{1,n}\}, D_2 = \{D_{2,n}\}$ be ensembles of random variables over $2^{g(n)}$-bit strings (and assume $g(n) \leqslant \mathrm{poly}(n)$ is injective and easily computable) for which BQLOGTIME can distinguish the two distributions but a quasi-polynomial 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,

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

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 $BQP^O \not\subset PH^O$. Let L be a uniformly random unary language in $\{1\}^*$. For each n, if $1^n \in L$, sample a $2^{g(n)}$-bit string x from D_1 and define oracle O restricted to length $g(n)$ so that x is its truth table; otherwise sample a $2^{g(n)}$-bit string x from D_2 and define oracle O restricted to length $g(n)$ so that x is its truth table. Fix an enumeration of all PH machines. Note that for any fixed PH oracle machine M we can find an input length n_M so that:

$$\begin{aligned} \Pr[A^O(1^{n_M}) = L(1^{n_M})] &= (1/2) \cdot \Pr[A^{D_1}(1^{n_M}) = 1] + (1/2) \cdot \Pr[A^{D_2}(1^{n_M}) = 0] \geqslant 1/2 + \varepsilon/2 \\ \Pr[M^O(1^{n_M}) = L(1^{n_M})] &= (1/2) \cdot \Pr[M^{D_1}(1^{n_M}) = 1] + (1/2) \cdot \Pr[M^{D_2}(1^{n_M}) = 0] \leqslant 1/2 + \delta/2 \end{aligned}$$

Now, there is a “gap” in the quantum and classical acceptance probabilities (because $\varepsilon > \delta$), so there exists a particular oracle O for which the corresponding quantum and classical machines disagree on input 1^{n_M}. Fix this choice of oracle up to this input length, and look the next fixed machine M'. By the same argument, we can find some $n_{M'} > n_M$ so that there exists a new oracle O' (with O as a prefix) for which $A^{O'}(1^{n_{M'}}) \neq M^{O'}(1^{n_{M'}})$

Because there at most a countably infinite number of PH machines, we can keep applying this argument, finding some fixed oracle obtaining the separation for all machines. Note that we can ensure that each successive length differs sufficiently from the prior length so that each machine cannot query inputs of the next largest length, and oracles at shorter lengths can be hardcoded.