Quantum Information and Computation, Vol. 18, No. 11&12 (2018) 0901–0909

PDQP/qpoly = ALL

SCOTT AARONSON[a]
University of Texas at Austin
aaronson@cs.utexas.edu

Received June 5, 2018
Revised August 9, 2018

We show that combining two different hypothetical enhancements to quantum computation—namely, quantum advice and non-collapsing measurements—would let a quantum computer solve any decision problem whatsoever in polynomial time, even though neither enhancement yields extravagant power by itself. This complements a related result due to Raz. The proof uses locally decodable codes.

Keywords:
Communicated by: R Cleve & R de Wolf

We've known for a quarter-century that quantum computers could efficiently solve a few problems, like factoring and discrete logarithms, that have resisted sustained efforts to solve them classically [22]. But we've also known that the tools used to prove this don't generalize, for example, to NP-complete problems [13]. At least in the black-box setting, even a quantum computer would provide at most a quadratic speedup (i.e., the speedup of Grover's algorithm [16] for unordered search, and it would face similar limits for many other tasks.

This situation has motivated some researchers to consider speculative generalizations of known physics, which would dramatically boost quantum computers' power. In 1998, Abrams and Lloyd [10] showed that a nonlinear term in the Schrödinger equation, if one existed, generally *would* let quantum computers solve NP-complete and even harder problems in polynomial time. Others (e.g., [12,9]) pointed out similar superpowers in quantum computers equipped with closed timelike curves.

Perhaps it's no surprise that doing violence to quantum-mechanical linearity in these ways would yield inordinate computational power. What's more surprising is that there are hypothetical resources that appear to boost the power of quantum computers, *but only by a little*, rather than by "absurd" amounts. This note is concerned with perhaps the two main examples of such resources: *quantum advice* and *non-collapsing measurements*. We now discuss them in turn.

Quantum Advice. In 2003, Nishimura and Yamakami [19] defined the class BQP/qpoly, consisting of all decision problems solvable by a polynomial-time quantum algorithm that's

[a] Supported by a Vannevar Bush Fellowship from the US Department of Defense, a Simons Investigator Award, and the Simons "It from Qubit" collaboration.

given a *quantum advice state* $|\psi_n\rangle$ with $n^{O(1)}$ qubits. The advice state depends only on the input length n, rather than on the specific input $x \in \{0,1\}^n$, but can otherwise be chosen arbitrarily to help the algorithm. It's natural to wonder *how much it can help* to be given a fixed state that encodes exponentially many complex numbers, albeit not in directly measurable form. More formally: does BQP/qpoly equal BQP/poly, which is the same class except that the advice is now restricted to being classical?

Watrous [23] gave an example of a problem for which quantum advice seems to help. Given a finite group G, each of whose elements is uniquely encoded by an n-bit string, as well as a fixed subgroup $H \leq G$ (and the ability to perform group operations), suppose we want to decide whether an input element $x \in G$ belongs to H. Watrous showed that a quantum computer can solve this problem in polynomial time, for *any* $x \in G$, if given the advice state

$$|H\rangle := \frac{1}{\sqrt{|H|}} \sum_{h \in H} |h\rangle \,,$$

by estimating the overlap between $|H\rangle$ and the coset state $|Hx\rangle$ (which can be efficiently created given $|H\rangle$). It's currently unknown how to solve the problem without such an advice state. Meanwhile, Aaronson and Kuperberg [8] showed that there exists a "quantum oracle" relative to which BQP/poly $\neq$ BQP/qpoly.[b]

Conversely, we also know significant limits on the power of quantum advice. In 2004, Aaronson [1] showed that BQP/qpoly $\subseteq$ PostBQP/poly, where PostBQP means quantum polynomial-time enhanced by the ability to *postselect* (or condition) on exponentially unlikely measurement outcomes, and is known to equal the classical complexity class PP [3]. In 2010, Aaronson and Drucker [7] improved this to BQP/qpoly $\subseteq$ QMA/poly $\cap$ coQMA/poly where QMA (Quantum Merlin-Arthur) is a quantum analogue of NP. These results imply, by a counting argument, that there must be at least some languages *not* in BQP/qpoly, which is not immediate from the definition! As we'll see, there are other complexity classes $\mathcal{C}$ for which $\mathcal{C}$/qpoly *does* contain all languages.

As a corollary of his so-called *direct product theorem* for quantum search, Aaronson [1] also showed that there exists an oracle relative to which NP $\not\subset$ BQP/qpoly. This means that, in the black-box setting, even quantum advice would not let quantum computers solve NP-complete problems in polynomial time.

Non-Collapsing Measurements. In 2014, Aaronson et al. [6] defined the class PDQP (Product Dynamical Quantum Polynomial-Time), consisting of all decision problems solvable by polynomial-time quantum algorithms with a hypothetical ability to make *multiple non-collapsing measurements* of a quantum state. In other words, they considered quantum circuits that, besides 1- and 2-qubit unitary gates, are equipped with two kinds of measurements:

(i) "ordinary" measurements, which collapse the state being measured according to the usual quantum-mechanical rules, *and also*

[b] However, they also showed that Watrous's group membership problem does *not* lead to an oracle separation, because it's solvable by a quantum computer with polynomial-size classical advice as well as a polynomial number of quantum queries (albeit, possibly exponential computation time).

(ii) "non-collapsing" measurements, which return an independent sample from the appropriate output distribution every time they're applied, yet leave the state unaffected and ready to be measured again.

Here Aaronson et al. [6] were building on 2005 work by Aaronson [2], who studied the power of quantum algorithms enhanced by the hypothetical ability to inspect the entire history of a *hidden variable* (as in Bohmian mechanics). This led him to define a complexity class called DQP (Dynamical Quantum Polynomial-Time), which contains PDQP and is closely related to it. However, the later work on PDQP separated out the core complexity-theoretic issues from the technical details of hidden-variable theories, and also fixed an error that Aaronson [2] had made.[c]

Aaronson et al. [6] gave two main examples of the power of non-collapsing measurements. First, we can use non-collapsing measurements to find *collisions* in any two-to-one function $f : [N] \rightarrow [N]$—that is, pairs x, y such that $f(x) = f(y)$—almost instantly. To do so, we first prepare the state

$$\frac{1}{\sqrt{N}} \sum_{x=1}^{N} |x\rangle |f(x)\rangle .$$

We then apply an ordinary collapsing measurement to the second register, to produce $\frac{|x\rangle+|y\rangle}{\sqrt{2}}$ where $f(x) = f(y)$ in the first register. Finally, we apply non-collapsing measurements to the first register to read out both x and y. Generalizing this, Aaronson et al. [6] showed that SZK $\subseteq$ PDQP, where SZK is the class of problems—including, for example, graph isomorphism and breaking lattice-based cryptography—that admit so-called statistical zero-knowledge proof protocols.

As a second example of the power of non-collapsing measurements, Aaronson et al. [6] showed that they let us solve the Grover problem—i.e., given a black-box function $f : [N] \rightarrow \{0, 1\}$, find a "marked item" x such that $f(x) = 1$—using only $\sim N^{1/3}$ steps, as opposed to the $\sim \sqrt{N}$ steps needed by an ordinary quantum computer. To do this, we first run $T \sim N^{1/3}$ iterations of Grover's search algorithm, in order to amplify the probability of the marked item up to $\sim \frac{T^2}{N} = \frac{1}{N^{1/3}}$. We then make $\sim N^{1/3}$ non-collapsing measurements of the resulting state, until (with high probability) the marked item has been found.

Strikingly, though, and much like with quantum advice, PDQP seems to provide only "slightly" more power than ordinary quantum computing. Indeed, Aaronson et al. [6] showed that any PDQP algorithm needs at least $\sim N^{1/4}$ steps to do Grover search, and as a consequence, that there exists an oracle relative which NP $\not\subset$ PDQP. In other words: in the black-box setting, even non-collapsing measurements still wouldn't let quantum computers solve NP-complete problems in polynomial time.

This note considers what happens when we *combine* polynomial-size quantum advice with non-collapsing measurements, to obtain the complexity class PDQP/qpoly. Surprisingly, and contrary to our initial guess, we find that even though the two resources are fairly weak individually, together they let us solve *everything*. That is: PDQP/qpoly = ALL, where ALL

[c] Specifically, Aaronson [2] claimed to show that $\mathsf{NP}^A \not\subset \mathsf{DQP}^A$ relative to a suitable oracle A; in reality he showed no such thing, though the conjecture remains plausible. By contrast, Aaronson et al. [6] gave a correct proof that $\mathsf{NP}^A \not\subset \mathsf{PDQP}^A$ relative to a suitable oracle A.

is the set of all languages $L \subseteq \{0,1\}^*$ (including the halting problem and other noncomputable languages).

There are precedents for such a result in quantum complexity theory. Most notably, in 2005, Raz [20] showed that $\mathsf{QIP}(2)/\mathsf{qpoly} = \mathsf{ALL}$, where $\mathsf{QIP}(2)$ consists of all languages that have two-message quantum interactive proof systems. His protocol, though different from ours, even used the exact same quantum advice state that ours will: namely, a superposition over a low-degree polynomial extension of the Boolean function $f : \{0,1\}^n \rightarrow \{0,1\}$ that we want to evaluate.

More trivially, Aaronson [4] observed that $\mathsf{PostBQP}/\mathsf{qpoly} = \mathsf{ALL}$. This is simply because, for any Boolean function f, if given the advice state

$$\frac{1}{\sqrt{2^n}} \sum_{z \in \{0,1\}^n} |z\rangle |f(z)\rangle \qquad (*)$$

as well as an input x, we can first measure in the computational basis and then postselect on getting $z = x$. For similar reasons, we have $\mathsf{PQP}/\mathsf{qpoly} = \mathsf{ALL}$, where $\mathsf{PQP} = \mathsf{PP}$ consists of all languages that admit a polynomial-time quantum algorithm that guesses the right answer with probability greater than $1/2$. Using error-correcting codes, Aaronson [4] also observed that $\mathsf{QMA}_{\mathsf{EXP}}/\mathsf{qpoly} = \mathsf{ALL}$, where $\mathsf{QMA}_{\mathsf{EXP}}$ is the exponential-time analogue of QMA.[d]

Compared to these earlier observations, we think the main novelty here is simply that PDQP seems to be so much *weaker* than $\mathsf{QIP}(2)$, $\mathsf{PostBQP}$, PQP, or $\mathsf{QMA}_{\mathsf{EXP}}$. As we've seen, unlike those other classes, PDQP is neither known nor believed to contain NP. Intuitively, it's just a "slight generalization" of BQP itself—which is what makes it perhaps unsettling that the mere addition of quantum advice can unlock so much power.

Indeed, the fact that $\mathsf{PDQP}/\mathsf{qpoly} = \mathsf{ALL}$ could be said to have a "real-world" implication. In a forthcoming work, on a practical scheme for generating cryptographically secure random bits using quantum supremacy experiments, Aaronson [5] found that, in order to derive the soundness of such a scheme, he needed to assume (what seems plausible) the existence of pseudorandom functions that are indistinguishable from random functions by any PDQP algorithm. He then noticed that an even stronger soundness conclusion would follow, if he assumed the existence of pseudorandom functions that are indistinguishable from random by any $\mathsf{PDQP}/\mathsf{qpoly}$ algorithm. Unfortunately, by the main result of this note, the latter doesn't exist! This was the genesis of the present work: as ethereal as it sounds, the result that $\mathsf{PDQP}/\mathsf{qpoly} = \mathsf{ALL}$ rules out a natural approach to proving the soundness of randomness generation schemes against adversaries with quantum advice.

For completeness, let us now give a formal definition of $\mathsf{PDQP}/\mathsf{qpoly}$.

Definition 1 *A* PDQP *circuit, acting on* m *qubits, is just an ordinary quantum circuit, which starts with the initial state* $|0\rangle^{\otimes m}$*; and can contain* 1*- and* 2*-qubit unitary gates from some finite, computationally universal set (for example,* CNOT *plus* $\pi/8$ *rotations), as well as measurement gates, which measure a qubit in the* $\{|0\rangle, |1\rangle\}$ *basis, collapsing the qubit to* $|0\rangle$ *or* $|1\rangle$ *in the usual way. In a given run of the circuit, let* $|\phi_t\rangle$ *be the pure state of the* m

[d] Note that, as pointed out in [4] adding quantum advice need not "commute" with standard complexity class inclusions. As an example, we have $\mathsf{PP} = \mathsf{PostBQP} \subseteq \mathsf{BQPSPACE} = \mathsf{PSPACE}$, yet $\mathsf{PostBQP}/\mathsf{qpoly}$ contains all languages whereas $\mathsf{BQPSPACE}/\mathsf{qpoly} = \mathsf{PSPACE}/\mathsf{poly}$ does not.

qubits immediately after the t^{th} gate is applied (note that the $|\psi_t\rangle$'s can be different in different runs, because of the probabilistic measurement gates). Also, let $\mathcal{D}_t$ be the distribution over m-bit strings obtained by measuring $|\phi_t\rangle$ in the computational basis. Then the "output" of a T-gate PDQP *circuit is a list of m-bit strings, $y_1, \ldots, y_T$, where each y_t was sampled from $\mathcal{D}_t$, independently of $y_{t'}$ for all $t' \neq t$.*

A PDQP *algorithm is a polynomial-time classical algorithm that, given an input $x \in \{0,1\}^n$, gets to specify a single* PDQP *circuit $C = C_x$, receive a single output $Y = \langle y_1, \ldots, y_T \rangle$ of C, and finally perform classical postprocessing on Y before either accepting or rejecting.*

A PDQP/qpoly *algorithm is the same, except that it can also include a list of pure states $\{|\psi_n\rangle\}_{n \geq 1}$, where $|\psi_n\rangle$ is on $p(n)$ qubits for some polynomial p, such that when the input x has length n, the initial state of C_x has the form $|\psi_n\rangle \otimes |0 \cdots 0\rangle$ rather than just $|0\rangle^{\otimes m}$.*

PDQP/qpoly *is the class of languages $L \subseteq \{0,1\}^*$ for which there exists a* PDQP/qpoly *algorithm A such that, for all $x \in \{0,1\}^*$, if $x \in L$ then $A(x)$ accepts with probability at least $2/3$, while if $x \notin L$ then $A(x)$ accepts with probability at most $1/3$.*

We can also let PDQEXP/qpoly be the same class as PDQP/qpoly, except that now the quantum algorithm can use exponential time. Then as an easy warmup, we observe that PDQEXP/qpoly = ALL. This is simply because, given the advice state (*), as well as an input $x \in \{0,1\}^n$, a PDQEXP algorithm can keep measuring in the computational basis, over and over about 2^n times, until it happens to get the outcome $|x\rangle |f(x)\rangle$.

We now prove this note's main (only) result.

Theorem 1 PDQP/qpoly = ALL.

Proof. Fix n, and let $f : \{0,1\}^n \rightarrow \{0,1\}$ be an arbitrary Boolean function. Then it suffices to describe a quantum advice state $|\psi_f\rangle$, on $n^{O(1)}$ qubits, such that a polynomial-time quantum algorithm equipped with both $|\psi_f\rangle$ and non-collapsing measurements can evaluate $f(x)$ on any input $x \in \{0,1\}^n$ of its choice.

Let $\mathbb{F}$ be a finite field of some prime order $q \geq n+2$ (by Bertrand's postulate, we can assume $q \leq 2n+1$). Also, let $g : \mathbb{F}^n \rightarrow \mathbb{F}$ be the unique multilinear extension of f: that is, the multilinear polynomial such that $g(x) = f(x)$ for all $x \in \{0,1\}^n$. Then our advice state will simply be

$$|\psi_f\rangle := \frac{1}{\sqrt{q^n}} \sum_{z \in \mathbb{F}^n} |z\rangle |g(z)\rangle .$$

This is a state of $O(n \log n)$ qubits.

Let $R : \mathbb{F}^n \rightarrow \mathbb{F}^n$ be the function that maps each vector $y \in \mathbb{F}^n$ to the unique scalar multiple αy of y whose leftmost nonzero entry is a 1, or to 0^n if $y = 0^n$. In other words, $R(y)$ is a canonical label for the *ray* in $\mathbb{F}^n$ that y belongs to.

Our PDQP algorithm is now the following. Given an input $x \in \{0,1\}^n$, first map $|\psi_f\rangle$ to

$$\frac{1}{\sqrt{q^n}} \sum_{z \in \mathbb{F}^n} |z\rangle |g(z)\rangle |R(z-x)\rangle .$$

Then measure the third register, $|R(z-x)\rangle$, via an ordinary collapsing measurement.

If the measurement outcome happens to be 0^n, then we can immediately learn $g(x) = f(x)$ by simply measuring the second register.

In the much more likely case that measuring $|R(z-x)\rangle$ yielded a nonzero outcome, say y, the reduced state of the first two registers is now

$$|\phi\rangle := \frac{1}{\sqrt{q-1}} \sum_{j \in \mathbb{F} \setminus \{0\}} |x+jy\rangle \, |g(x+jy)\rangle .$$

Define $p : \mathbb{F} \to \mathbb{F}$ by $p(j) := g(x+jy)$. Then notice that p is a univariate polynomial in j of degree at most n, and furthermore that $p(0) = g(x) = f(x)$.

As the last step, we simply perform repeated non-collapsing measurements of $|\phi\rangle$ in the computational basis, until we have learned the values of $p(j)$ for every $j \in \mathbb{F} \setminus \{0\}$. This is an instance of the coupon collector's problem, so with overwhelming probability it takes at most $O(q \log q) = O(n \log n)$ measurements. Then, in the classical postprocessing phase, we perform polynomial interpolation on the recovered $p(j)$ values, in order to learn $p(0) = f(x)$. □

We conclude with some miscellaneous remarks and open problems about Theorem 1.

Notice that the proof of Theorem 1 did not depend on quantum mechanics in any essential way. In other words, let PDPP be the classical analogue of PDQP (see Definition 1), which consists of all languages decidable by polynomial-time randomized algorithms that can contain both "collapsing" and "non-collapsing" measurements of bits.[e] Likewise, let PDPP/rpoly be PDPP augmented by polynomial-size classical randomized advice. More formally:

Definition 2 *A* PDPP *circuit starts with the uniform distribution over inputs of the form* $(r, 0^m)$*, where* $r \in \{0,1\}^m$*; and can apply Toffoli gates as well as "measurement gates." The latter force a given bit to be either definitely* 0 *or definitely* 1*, both with the appropriate probabilities. In a given run of the circuit, let* $\mathcal{D}_t$ *be the probability distribution over the* $2m$ *bits immediately after the* t^{th} *gate is applied (note that the* $\mathcal{D}_t$*'s can be different in different runs, because of the probabilistic measurement gates). Then the "output" of a* T*-gate* PDPP *circuit is a list of* $2m$*-bit strings,* $y_1, \ldots, y_T$*, where each* y_t *was sampled from* $\mathcal{D}_t$*, independently of* $y_{t'}$ *for all* $t' \neq t$*.*

A PDPP *algorithm is a polynomial-time deterministic classical algorithm that, given an input* $x \in \{0,1\}^n$*, gets to specify a single* PDPP *circuit* $C = C_x$*, receive a single output* $Y = \langle y_1, \ldots, y_T \rangle$ *of* C*, and finally perform postprocessing on* Y *before either accepting or rejecting.*

A PDPP/rpoly *algorithm is the same, except that it can also include a list of distributions* $\{\mathcal{E}_n\}_{n \geq 1}$*, where* $\mathcal{E}_n$ *is on* $p(n)$ *bits for some polynomial* p*, such that when the input* x *has length* n*, the initial state of* C_x *has the form* $(r, 0^m, z)$*, where* z *is a sample from* $\mathcal{E}_n$*, rather than just* $(r, 0^m)$*.*

PDPP/rpoly *is the class of languages* $L \subseteq \{0,1\}^*$ *for which there exists a* PDPP/rpoly *algorithm* A *such that, for all* $x \in \{0,1\}^*$*, if* $x \in L$ *then* $A(x)$ *accepts with probability at least* $2/3$*, while if* $x \notin L$ *then* $A(x)$ *accepts with probability at most* $1/3$*.*

Then we have:

Theorem 2 PDPP/rpoly = ALL.

[e] As far as I know, the class PDPP was first suggested by Harry Altman in blog comments: see, e.g., www.scottaaronson.com/blog/?p=2096#comment-345575

Proof. The proof is literally the same as that of Theorem 1. The only difference is that, whenever the proof of Theorem 1 considers a pure state $|\phi\rangle$, we now consider instead the classical probability distribution obtained by measuring $|\phi\rangle$ in the computational basis. □

The previous results of Raz [20] and Aaronson [4], about quantum advice boosting various quantum complexity classes to unlimited power, can all similarly be "de-quantized," and stated in terms of randomized rather than quantum advice. That is,

$$\mathsf{IP}(2)/\mathsf{rpoly} = \mathsf{PostBPP}/\mathsf{rpoly} = \mathsf{PP}/\mathsf{rpoly} = \mathsf{MA_{EXP}}/\mathsf{rpoly} = \mathsf{ALL}.^{f}$$

Indeed, the only reason to state these results in terms of quantum advice in the first place, is that quantum advice has been a subject of independent interest whereas randomized advice has not.

In 2006, Aaronson [4] raised the question of whether there's *any* natural quantum complexity class $\mathcal{C}$ that quantum advice boosts to ALL, even though classical randomized advice fails to do so. As far as we know that question remains open.

The trick used to prove Theorem 1 also has an implication for communication complexity. Namely: suppose Alice has a string $x \in \{0,1\}^N$, Bob has an index $i \in [N]$, and Alice wants to send Bob a message that will enable him to learn x_i. For this so-called Index problem, it's known that even any quantum protocol requires Alice to send Bob at least $\sim N$ qubits [11]. Nevertheless, we claim that there's a protocol for this problem in which Alice sends Bob a quantum state $|\psi_x\rangle$ of only $O(\log N \log\log N)$ qubits, and then Bob learns x_i after making an ordinary collapsing measurement of $|\psi_x\rangle$ followed by $O(\log N \log\log N)$ non-collapsing measurements. This protocol is exactly the one from Theorem 1, except with x in place of the truth table of f, and i in place of x.

Any reader familiar with *Locally Decodable Codes* (LDCs) might recognize them as the central concept in the proof of Theorem 1, even though we kept the proof self-contained and never used the term. In general, an error-correcting code is a function $C : \Sigma^N \to \Sigma^M$ for some finite alphabet Σ, with the property that $C(x)$ and $C(y)$ differ on a large fraction of coordinates for all $x \neq y$. An LDC is a special kind of error-correcting code: one such that, for each entry x_i of the original string $x = x_1 \ldots x_N$, it's possible to recover x_i from any string w close to $C(x)$, with high probability, via a randomized algorithm that queries w in only r randomly chosen (but correlated) locations. Here one wants r to be as small as possible, even a constant like 2 or 3.

In a sequence of breakthroughs (see, e.g., [24, 17, 14]), it was established that for every constant $r = 2^t$, there exist r-query LDCs with linear distance and with size

$$M = \exp\exp\left((\log N)^{1/t}(\log\log N)^{1-1/t}\right).$$

For $r \geq 4$,[g] this size is less than exponential in N, albeit more than polynomial. We didn't use these sophisticated LDCs, for a combination of reasons: first, we were fine with $r = n^{O(1)}$

[f] A word of caution, though: even though Goldwasser and Sipser [15] showed that $\mathsf{IP}(2) = \mathsf{AM}$ (where AM denotes two-message, *public-coin* interactive proof systems), we do *not* have $\mathsf{AM}/\mathsf{rpoly} = \mathsf{ALL}$. Instead, $\mathsf{AM}/\mathsf{rpoly} = \mathsf{MA}/\mathsf{rpoly} = \mathsf{NP}/\mathsf{poly}$ (see [4]). This is an instance of the broader phenomenon that adding randomized and quantum advice needn't commute with standard complexity class containments.

[g] Though 4 is the smallest power of 2 for which the bound is nontrivial, with modified arguments one can also handle the case $r = 3$.

queries, which meant that a vastly simpler LDC, based on a multilinear extension of the Boolean function f, could be used instead. Second, we were *not* fine with $\log M$, the number of qubits in the advice state, being more than $(\log N)^{O(1)} = n^{O(1)}$, as it would be with the state-of-the-art constant-query LDCs.

One might ask whether, in the algorithm of Theorem 1, the number of non-collapsing measurements could be reduced from $O(n \log n)$ to a small constant r. A positive answer will follow if there turn out to exist $(r+1)$-query LDCs of constant distance and at most quasipolynomial size, which moreover are sufficiently explicit and efficient.

In this connection, it's interesting that Kerenidis and de Wolf [18] proved—as it happens, by using a quantum information argument—that there are no 2-query LDCs of subexponential size. This raises the possibility that, in any algorithm like ours, there must be at least *two* non-collapsing measurements (as well as a third and final measurement, which might as well be collapsing). This seems surprising: *a priori*, one might have guessed that a single non-collapsing measurement would already provide all the computational power that can be had from such a resource.

The open problem that interests us the most in this subject is the following. A central fact about PDQP, shown by Aaronson et al. [6], is that it contains SZK. While [6] never made this explicit, the same argument shows that PDQP contains a larger class that we could call QCSZK (Quantum Classical SZK), consisting of all languages that admit a statistical zero-knowledge proof protocol with a quantum verifier but classical communication with the prover.[h] We thus raise the following question: does QCSZK/qpoly equal ALL? Or we might as well ask the analogous classical question: does SZK/rpoly equal ALL? What about NISZK/rpoly (where NISZK means Non-Interactive SZK)?

Acknowledgments

I thank Dana Moshkovitz for helpful conversations, and the anonymous reviewers for their comments.

References

1. S. Aaronson. Limitations of quantum advice and one-way communication. *Theory of Computing*, bf 1:128, 2005. Earlier version in CCC2004. quant-ph/0402095.
2. S. Aaronson. Quantum computing and hidden variables. *Phys. Rev. A*, **71**(032325), 2005. quant-ph/0408035 and quant-ph/0408119.
3. S. Aaronson. Quantum computing, postselection, and probabilistic polynomial-time. *Proc. Roy. Soc. London*, **A461**(2063):34733482, 2005. quant-ph/0412187.
4. S. Aaronson. QMA/qpoly is contained in PSPACE/poly: de-Merlinizing quantum protocols. In *Proc. Conference on Computational Complexity*, pages 261273, 2006. quant-ph/0510230.
5. S. Aaronson. Certied randomness from quantum supremacy. To appear, 2018.
6. S. Aaronson, A. Bouland, J. Fitzsimons, and M. Lee. The space just above BQP. In *Proc. Innovations in Theoretical Computer Science (ITCS)*, pages 271280, 2016. arXiv:1412.6507.

[h] This class has the following as a complete promise problem, generalizing the SZK-complete Statistical Difference problem of Sahai and Vadhan [21]. Given as input two quantum circuits C_0 and C_1, which sample probability distributions $\mathcal{D}_0$ and $\mathcal{D}_1$ respectively over n-bit strings, decide whether $\mathcal{D}_0$ and $\mathcal{D}_1$ have variation distance at most 1/3 or at least 2/3, promised that one of these is the case.

7. S. Aaronson and A. Drucker. A full characterization of quantum advice. *SIAM J. Comput.*, **43**(3):11311183, 2014. Earlier version in STOC2010. arXiv:1004.0377.
8. S. Aaronson and G. Kuperberg. Quantum versus classical proofs and advice. *Theory of Computing*, **3**(7):129157, 2007. Earlier version in CCC2007. arXiv:quant-ph/0604056.
9. S. Aaronson and J. Watrous. Closed timelike curves make quantum and classical computing equivalent. *Proc. Roy. Soc. London*, **A465**:631647, 2009. arXiv:0808.2669.
10. D. S. Abrams and S. Lloyd. Nonlinear quantum mechanics implies polynomial-time solution for NP-complete and P problems. *Phys. Rev. Lett.*, **81**:39923995, 1998. quant-ph/9801041.
11. A. Ambainis, A. Nayak, A. Ta-Shma, and U. V. Vazirani. Quantum dense coding and quantum nite automata. *J. of the ACM*, **49**:496511, 2002. Earlier version in STOC1999, pp. 376-383. quant-ph/9804043.
12. D. Bacon. Quantum computational complexity in the presence of closed timelike curves. *Phys. Rev. A*, **70**(032309), 2004. quant-ph/0309189.
13. C. Bennett, E. Bernstein, G. Brassard, and U. Vazirani. Strengths and weaknesses of quantum computing. *SIAM J. Comput.*, **26**(5):15101523, 1997. quant-ph/9701001.
14. K. Efremenko. 3-query locally decodable codes of subexponential length. *SIAM J. Comput.*, **41**(6):16941703, 2012. Earlier version in STOC2009. ECCC TR08-069.
15. S. Goldwasser and M. Sipser. Private coins versus public coins in interactive proof systems. In *Randomness and Computation*, volume 5 of Advances in Computing Research. JAI Press, 1989.
16. L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proc. ACM STOC*, pages 212219, 1996. quant-ph/9605043.
17. T. Itoh and Y. Suzuki. Improved constructions for query-ecient locally decodable codes of subexponential length. *IEICE Transactions*, **93-D**(2):263270, 2010. arXiv:0810.4576.
18. I. Kerenidis and R. de Wolf. Exponential lower bound for 2-query locally decodable codes via a quantum argument. *J. Comput. Sys. Sci.*, **69**(3):395420, 2004. Earlier version in STOC2003. quant-ph/0208062.
19. H. Nishimura and T. Yamakami. Polynomial time quantum computation with advice. *Inform. Proc. Lett.*, **90**:195204, 2003. ECCC TR03-059, quant-ph/0305100.
20. R. Raz. Quantum information and the PCP theorem. *Algorithmica*, **55**(3):462489, 2009. Earlier version in FOCS2005. quant-ph/0504075.
21. A. Sahai and S. Vadhan. A complete promise problem for statistical zero-knowledge. *J. of the ACM*, **50**(2):196249, 2003. Earlier version in FOCS1997. ECCC TR00-084.
22. P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, **26**(5):14841509, 1997. Earlier version in FOCS1994. quant-ph/9508027.
23. J. Watrous. Succinct quantum proofs for properties of nite groups. In *Proc. IEEE FOCS*, pages 537546, 2000. cs.CC/0009002.
24. S. Yekhanin. Towards 3-query locally decodable codes of subexponential length. *J. of the ACM*, **55**:1, 2008. Earlier version in STOC2007. See also ECCC TR06-127.