PDQMA = DQMA = NEXP: QMA with Hidden Variables and Non-Collapsing Measurements

Scott Aaronson
University of Texas at Austin, TX, USA

Sabee Grewal
University of Texas at Austin, TX, USA

Vishnu Iyer
University of Texas at Austin, TX, USA

Simon C. Marshall
Leiden University, The Netherlands

Ronak Ramachandran
University of Texas at Austin, TX, USA

Abstract

We define and study a variant of QMA (Quantum Merlin Arthur) in which Arthur can make multiple non-collapsing measurements to Merlin's witness state, in addition to ordinary collapsing measurements. By analogy to the class PDQP defined by Aaronson, Bouland, Fitzsimons, and Lee (2014), we call this class PDQMA. Our main result is that PDQMA = NEXP; this result builds on the PCP theorem and complements the result of Aaronson (2018) that PDQP/qpoly = ALL. While the result has little to do with quantum mechanics, we also show a more "quantum" result: namely, that QMA with the ability to inspect the entire history of a hidden variable is equal to NEXP, under mild assumptions on the hidden-variable theory. We also observe that a quantum computer, augmented with quantum advice and the ability to inspect the history of a hidden variable, can solve any decision problem in polynomial time.

2012 ACM Subject Classification Theory of computation → Complexity classes; Theory of computation → Quantum computation theory; Theory of computation → Quantum complexity theory; Theory of computation → Quantum complexity theory

Keywords and phrases quantum Merlin-Arthur, non-collapsing measurements, hidden-variable theories

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2025.3

Related Version *Full Version*: https://arxiv.org/abs/2403.02543

Funding SG and RR are supported via SA from the same funding sources.
Scott Aaronson: SA is supported by a Vannevar Bush Fellowship by the US Department of Defense, the Berkeley NSF-QLCI CIQC Center, a Simons Investigator Award, and the Simons "It from Qubit" collaboration.
Vishnu Iyer: VI is supported by an NSF Graduate Research Fellowship.
Simon C. Marshall: SCM is supported by the European Union's Horizon 2020 program (NEQASQC) and a gift from Google Quantum AI.

Acknowledgements SCM thanks David Dechant for useful discussions. SG, VI, and RR thank Joshua Cook, Siddhartha Jain, and Kunal Marwaha for helpful conversations.

1 Introduction

To understand the power of quantum computation is, in large part, to understand how that power depends on the central features of quantum mechanics itself, such as linearity, unitarity, tensor products, complex numbers, the Born Rule, or the destructive nature of measurement.

45th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2025).
Editors: C. Aiswarya, Ruta Mehta, and Subhajit Roy; Article No. 3; pp. 3:1–3:17
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

But since quantum mechanics is usually presented as a "package deal," how can we pick apart these dependencies? One natural approach has been to define complexity classes based on "fantasy" versions of quantum mechanics, which change one or more of its features, and see how they relate to the standard BQP (Bounded-Error Quantum Polynomial-Time). Some representative findings of that research program over the past few decades include:

(1) Quantum mechanics over the reals or quaternions leads to the same computational power as quantum mechanics over the complex numbers, despite the theories differing in other respects [25].

(2) Quantum mechanics with a nonlinear Schrödinger equation would generically allow NP- and even #P-complete problems to be solved in polynomial time, in contrast to what is conjectured for standard (linear) quantum mechanics [11].

(3) A quantum computer with closed timelike curves could solve exactly the problems in PSPACE, same as a classical computer with closed timelike curves [10].

(4) Quantum computers with nonunitary linear evolution, or modifications of the Born rule (say, $|\psi|^3$), with normalization of probabilities imposed, yield at least the power of quantum computers with *postselected measurement outcomes* – a model that Aaronson called PostBQP and proved to coincide with the classical complexity class PP [2, 5].

(5) Generalized probabilistic theories (GPTs), a class of theories that includes quantum mechanics and classical probability theory as special cases, are as a whole characterized by the complexity class AWPP [23, 15].

(6) If we allow multiple, non-collapsing measurements of the same state – or, closely related, to see the entire history of a hidden variable as in Bohmian mechanics – we get a model of computation that seems more powerful than standard quantum computation, but only "slightly" so [4]. As examples, we can quickly find collisions in many-to-one functions (and thus, for example, solve Graph Isomorphism), and we can solve the Grover search problem in $N^{1/3}$ steps rather than $\sqrt{N}$. But we still seem unable to solve NP-complete problems in polynomial time.

Example (6) is the one of most interest to us here. To our knowledge, it is the only natural example known where changing the rules of quantum mechanics leads to complexity classes that appear only modestly larger than BQP. Much of the power comes from the combination of non-collapsing measurements with ordinary collapsing ones. As an example, given a two-to-one function $f : [N] \to [M]$, the way to find collisions is simply to prepare

$$\frac{1}{\sqrt{N}} \sum_{x \in [N]} |x\rangle|f(x)\rangle,$$

then measure the $|f(x)\rangle$ register in the ordinary way to get

$$\frac{|x\rangle + |y\rangle}{\sqrt{2}}$$

where $f(x) = f(y)$ in the first register, and finally perform multiple non-collapsing measurements on the first register in the standard basis, until both x and y are observed with high probability.[1]

[1] The reason why non-collapsing measurement allows Grover search in $N^{1/3}$ steps is simpler and does not require us to combine collapsing with non-collapsing measurements. Instead, given a unique marked item out of N, one simply runs Grover's algorithm for $T = N^{1/3}$ iterations, thereby boosting the probability of the marked item to $\sim \frac{T^2}{N} = N^{-1/3}$, and then performs $N^{1/3}$ non-collapsing measurements so that the marked item is found with constant probability.

In a hidden-variable theory, where the hidden variable is either at $|x\rangle$ or $|y\rangle$ with equal probabilities, the solution at this point is to "juggle" – for example, by repeatedly applying Fourier transforms followed by their inverses. The goal here is to cause the hidden variable to "forget" whether it was at $|x\rangle$ or $|y\rangle$, so that it must eventually visit both of them with high probability. In such a case, if (as we're imagining) we could see the whole history of the hidden variable at once, from some godlike vantage point, we would learn both $|x\rangle$ and $|y\rangle$ and thereby solve our computational problem.

The history of these ideas is a bit tangled. In 2005, Aaronson defined the class DQP (Dynamical Quantum Polynomial-Time), to capture the problems that quantum computers could efficiently solve, if only one could examine the entire history of a hidden variable (in any hidden-variable theory that satisfies reasonable axioms, called robustness and indifference). He showed that $\mathsf{SZK} \subseteq \mathsf{DQP}$, where SZK is Statistical Zero Knowledge, basically because DQP could simulate non-collapsing measurements (although he didn't formalize this). Combined with Aaronson's quantum lower bound for finding collisions [1], which implies the existence of an oracle relative to which $\mathsf{SZK} \not\subset \mathsf{BQP}$, this gives us an oracle separation between BQP and DQP. Aaronson also showed that $\mathsf{DQP} \subseteq \mathsf{EXP}$, which has not been improved since then. He claimed to give an oracle relative to which $\mathsf{NP} \not\subset \mathsf{DQP}$, although his proof had a bug [8] and the existence of such an oracle remains open. Then, in 2014, Aaronson, Bouland, Fitzsimons, and Lee [8] defined the class PDQP (Product Dynamical Quantum Polynomial-Time), to capture non-collapsing measurements specifically. They showed that PDQP also contains SZK, showed the upper bound $\mathsf{PDQP} \subseteq \mathsf{BPP}^{\mathsf{PP}}$, and gave a correct proof that there exists an oracle relative to which $\mathsf{NP} \not\subset \mathsf{PDQP}$.

Overall, as we said, these results painted a picture of DQP and PDQP as only modestly more powerful than BQP. However, a surprising new wrinkle came in 2018, when Aaronson [6] observed that $\mathsf{PDQP/qpoly} = \mathsf{ALL}$, where /qpoly means "with polynomial-sized quantum advice," and ALL is the class of all languages. This stands in contrast to 2004 results of Aaronson [3] that limit the power of BQP/qpoly: namely, that $\mathsf{BQP/qpoly} \subseteq \mathsf{PP/poly}$ (later improved by Aaronson and Drucker [9] to $\mathsf{BQP/qpoly} \subseteq \mathsf{QMA/poly}$), and that there exists an oracle relative to which $\mathsf{NP} \not\subset \mathsf{BQP/qpoly}$. In other words, quantum advice and non-collapsing measurements have a "Mentos and Coke" character, where each one is only modestly powerful in isolation, but together they trigger a complexity-theoretic explosion.

To prove the $\mathsf{PDQP/qpoly} = \mathsf{ALL}$ result, Aaronson adapted a 2005 theorem of Raz [24] that $\mathsf{QIP/qpoly} = \mathsf{ALL}$, where QIP is the class of languages that admit quantum interactive proofs. In both Raz's protocol and Aaronson's, given an arbitrary Boolean function $f : \{0,1\}^n \to \{0,1\}$ that one wants to compute, one first chooses a prime $q \gg n$. The whole truth table of f is then encoded by the quantum advice state

$$|\psi\rangle = \frac{1}{\sqrt{q^n}} \sum_{z \in \mathbb{F}_q^n} |z\rangle |p(z)\rangle,$$

where $p : \mathbb{F}_q^n \to \mathbb{F}_q$ is the unique multilinear polynomial over $\mathbb{F}_q$ such that $p(x) = f(x)$ for all $x \in \{0,1\}^n$. Next, given a point of interest $x \in \{0,1\}^n$, on which one wants to evaluate $f(x)$, one measures $|\psi\rangle$ so as to collapse it to an equal superposition

$$|\psi_\ell\rangle = \frac{1}{\sqrt{q-1}} \sum_{z \in \ell \setminus \{x\}} |z\rangle |p(z)\rangle$$

over a random line $\ell \subset \mathbb{F}_q^n$ that passes through x, minus the point x itself. Finally, one uses polynomial interpolation on this line to recover $p(x) = f(x)$. In the non-collapsing measurements model, this is done by simply measuring the state $|\psi_\ell\rangle$ over and over in the standard basis, until enough $(z, p(z))$ pairs have been observed for the interpolation to work.[2]

1.1 This Paper

Here we show a new example where non-collapsing measurements combined with one other resource yield extraordinary computational power – vastly more power than either resource in isolation.

The class QMA (Quantum Merlin Arthur) is a well-known quantum generalization of NP; it consists of all languages for which a "yes" answer can be verified in quantum polynomial time with the help of a polynomial-size quantum witness state. We define and study PDQMA (Product Dynamical QMA), or QMA augmented with non-collapsing measurements. Our main result is that PDQMA = NEXP, even if PDQMA is defined with a completeness/soundness gap of (say) $1 - 2^{-n}$ vs. 2^{-n} (Theorem 5).

Since the inclusion PDQMA $\subseteq$ NEXP is straightforward, the interesting part is NEXP $\subseteq$ PDQMA. By the celebrated PCP theorem [14, 13], it suffices to show that a two-query PCP system (Definition 3) can be simulated by PDQMA. Our proof builds on Aaronson's proof [6] that PDQP/qpoly = ALL, but with two key differences. First, to simulate a two-query PCP system, we need a witness state that can support at least *two* queries to an exponentially long truth table, rather than just one query. Second, we now need an analysis of *soundness*: why, for example, can the prover not cheat by sending a witness that causes the response to each query to depend on the other query? This problem seems particularly acute once we realize that we can no longer rely on the no-signaling principle of quantum mechanics when non-collapsing measurements are allowed. To address the first challenge, we extend Aaronson's algorithm to support multiple queries to an exponentially long truth table. To address the second challenge, we show that one can use non-collapsing measurements to perform a low-degree test, ensuring that Merlin's witness has the "error-correcting code" structure of an honest witness state.

Let us highlight our simulation of a classical PCP as additional motivation for our work. Specifically, simulating PCP systems is one of the few approaches to show that quantum Merlin-Arthur systems can solve NEXP (see e.g. [7, 22]). An important direction for future work is to adapt our simulation to quantum Merlin-Arthur systems that adhere to the laws of quantum mechanics (even if the verifier is allowed exponential time or to interact with the prover). We view any progress of this form as progress toward understanding more "reasonable" quantum Merlin-Arthur systems, such as QMA(2).

We point out some implications of our result. First, combining with the Nondeterministic Time Hierarchy Theorem (NP $\neq$ NEXP), we find that PDQMA is *unconditionally* more powerful than NP. Second, "scaling down by an exponential," we find that when non-collapsing measurements are allowed, the problem of optimizing an acceptance probability over all N-dimensional quantum states jumps from easy (a principal eigenvector problem) to NP-hard even to approximate.

Let us place our result in the context of previous work on generalizations of QMA. Aharonov and Regev [12] defined QMA+, a variant of QMA where the verifier can directly obtain the probability a given two-outcome measurement will accept, and showed QMA+ =

[2] Since p is a multilinear extension of a Boolean function on n variables, its degree is at most n. Hence, $n + 1$ pairs are needed to do polynomial interpolation, so q must be chosen to be at least $n + 2$.

QMA. More recently, Jeronimo and Wu [22] showed that $\mathsf{QMA}^+(2) = \mathsf{NEXP}$, where $\mathsf{QMA}^+(2)$ is QMA with two unentangled proofs that have nonnegative real amplitudes. Bassirian, Fefferman, and Marwaha [16] improved this to show that $\mathsf{QMA}^+ = \mathsf{NEXP}$ – i.e., nonnegative amplitudes alone suffice for the jump to NEXP. As of this writing, it remains a matter of avid speculation whether unentangled witnesses *also* suffice for the jump to NEXP: that is, whether $\mathsf{QMA}(2) = \mathsf{NEXP}$. Of course, our result implies that it would suffice to simulate non-collapsing measurements using unentangled proofs – i.e., to show $\mathsf{PDQMA} \subseteq \mathsf{QMA}(2)$.

One important observation about our PDQMA protocol is that it only ever measures the witness state in the computational basis – and hence, one could say, never exploits quantum interference. So in particular, if we defined a complexity class PDMA (Product Dynamical Merlin-Arthur) in mathematical parallel to PDQMA, where the witness was a classical probability distribution $\mathcal{D}$, and one was allowed to sample from $\mathcal{D}$ with or without doing Bayesian updating to it, we would equally have $\mathsf{PDMA} = \mathsf{NEXP}$. The main difference here is simply that it seems hard to invent a story that motivates PDMA.

In Section 4, we sharpen this point by defining and studying a class that we call DQMA (Dynamical QMA), or QMA augmented by the ability to see the entire history of a hidden variable. In particular, the verifier can perform a DQP (Dynamical Quantum Polynomial-Time) computation, a complexity class introduced by Aaronson to study hidden variable theories from a quantum computing perspective [4]. In short, DQP captures computations where one evolves a state vector unitarily, but there is a deeper "hidden variable" in a definite state that evolves stochastically. At the end of the computation, the history of the hidden variable (i.e., the sequence of definite states it occupied at each step) can be viewed when making the final accept/reject decision. DQP captures any hidden variable theory that satisfies a set of axioms (given in Section 4), including theories due to Dieks [19] and Schrödinger [26].

We show that $\mathsf{DQMA} = \mathsf{NEXP}$ (Theorem 16). Here the reasons really do depend on quantum mechanics. Specifically, they depend on the ability to "hide" crucial information in the phases of amplitudes, to prevent a hidden variable trajectory from remembering that information. If we tried to define the analogous class DMA (Dynamical MA), it would trivially coincide with MA. Assuming $\mathsf{MA} = \mathsf{NP}$, as follows from a standard derandomization assumption, this has the amusing consequence that $\mathsf{DMA} \neq \mathsf{DQMA}$ – that is, in the presence of both witness states and trajectory sampling, quantum can already be known to be stronger than classical. In the same section, we also observe that our techniques imply $\mathsf{DQP/qpoly} = \mathsf{ALL}$ (Theorem 17), complementing Aaronson's result that $\mathsf{PDQP/qpoly} = \mathsf{ALL}$ [6]. This result relies on quantum mechanics in the same way that our DQMA result does.

One might also wonder about other "fantasy" variants of QMA. In principle, any variant of BQP can be combined with QMA to yield a new complexity class. For example, consider a variant of BQP that can clone states (i.e., perform the transformation $|\psi\rangle|0^n\rangle \mapsto |\psi\rangle^{\otimes 2}$). It is easy to see that one can simulate k non-collapsing measurements of a state by cloning the state k times and then measuring each copy in the usual collapsing way. Hence, combining this BQP variant with QMA yields a complexity class equal to NEXP by our Theorem 5. Indeed, if one wants to alter Arthur's powers *without* triggering a "Mentos and Coke" effect, they would need to find a BQP variant that is only modestly more powerful, the way PDQP is.

1.2 Main Ideas

We give a high-level overview of our proof that $\mathsf{PDQMA} = \mathsf{NEXP}$ (Theorem 5). Informally, PDQMA is like QMA except the verifier can perform non-collapsing measurements in addition to the normal collapsing ones (see Definition 2 for a formal definition). The containment $\mathsf{PDQMA} \subseteq \mathsf{NEXP}$ is straightforward because NEXP can guess the exponentially-long classical description of the PDQMA witness and verify it in exponential time.

Thus, the challenge is proving $\mathsf{NEXP} \subseteq \mathsf{PDQMA}$. We show this by simulating a two-query probabilistically checkable proof (PCP) system for NEXP (see Definition 3 and Theorem 4). In short, the celebrated PCP theorem [14, 13] tells us that languages in NEXP can be decided by a verifier that queries a PCP $\pi : \{0,1\}^n \to \Sigma$ at two points of the verifier's choosing, where Σ is a constant-sized alphabet. Hence, our result follows from simulating two queries to a π.

The honest witness is the same as in [6]:

$$|\psi\rangle = \frac{1}{\sqrt{q^n}} \sum_{z \in \mathbb{F}_q^n} |z\rangle|p(z)\rangle,$$

where $p : \mathbb{F}_q^n \to \mathbb{F}_q$ is the unique degree-n multilinear extension of the PCP $\pi : \{0,1\}^n \to \Sigma$ and $q = O(n)$ is chosen to be sufficiently large. However, our setting differs from [6] in two key ways: (i) we must retrieve two values $\pi(w)$ and $\pi(w')$ for our choice of w, w', rather than one, and (ii) the quantum witness can no longer be trusted (as it can be in the advice setting).

To explain how to handle the first difference, let us briefly recall how to simulate a single query. As discussed, given a point of interest $w \in \{0,1\}^n$, on which one wants to evaluate $\pi(w)$, Aaronson [6] measures $|\psi\rangle$ so as to collapse it to an equal superposition

$$|\psi_\ell\rangle = \frac{1}{\sqrt{q-1}} \sum_{z \in \ell \setminus \{w\}} |z\rangle|p(z)\rangle$$

over a random line $\ell \subset \mathbb{F}_q^n$ that passes through w, minus the point w itself. To recover two values $w, w' \in \{0,1\}^n$, one can generalize Aaronson's procedure so that the measurement on $|\psi\rangle$ collapses it to an equal superposition over an *affine plane* that contains the points w and w', minus the unique affine line containing w and w'. (Indeed, this can be generalized to any k-dimensional affine subspace, where the measurement collapses $|\psi\rangle$ to a superposition over a random k-dimensional affine subspace containing points $w_1, \ldots, w_k$ minus the unique $(k-1)$-dimensional affine subspace containing $w_1, \ldots, w_k$.) Then, just as in [6], one performs enough non-collapsing measurements to recover $\pi(w)$ and $\pi(w')$ via polynomial interpolation.

The only remaining issue is that Merlin can potentially cheat and send a quantum state that is far from the honest witness. To address this, we use the lines-point low-degree test (Lemma 1) given by Friedl and Sudan [20]. Recall that an affine line $\ell \subset \mathbb{F}_q^n$ passing through points $x, y \in \mathbb{F}_q^n$ is the set $\{x + (y-x)t\}_{t \in \mathbb{F}_q^n}$, so a function $g : \ell \to \mathbb{F}_q$ can be viewed as a univariate polynomial in t. The lines-point low-degree test says that if a function $f : \mathbb{F}_q^n \to \mathbb{F}_q$ restricted to a randomly chosen line agrees with a degree-n univariate polynomial, then f agrees with some degree-n polynomial $h : \mathbb{F}_q^n \to \mathbb{F}_q$ on all of $\mathbb{F}_q^n$.

Observe that to perform this low-degree test, we need to make sure the function that Merlin encodes into his witness agrees with a low-degree polynomial on a randomly chosen affine line. Therefore, rather than interpolating a polynomial on a random affine plane containing the points w and w' of interest, our verification procedure interpolates a polynomial on an *affine cube*. This affine cube contains the points $w, w' \in \{0,1\}^n$ but we also ensure that it contains a point $w'' \in \mathbb{F}_q^n$ that the verifier chooses uniformly at random. This ensures that the affine cube contains a uniformly random line independent of the points w and w'. Then, if the polynomial interpolation succeeds, we can (i) recover $\pi(w)$ and $\pi(w')$ as desired and (ii) conclude that Merlin's witness encoded a low-degree polynomial because it passed the lines-point low degree test (i.e., it agreed with a low-degree polynomial on a randomly chosen line).

To summarize, the procedure works as follows. First, Merlin commits to some witness state $|\psi\rangle$. Then the PDQMA verifier simulates the PCP verifier to obtain queries $w, w' \in \{0,1\}^n$ and picks a uniformly random $w'' \in \mathbb{F}_q^n$. The verifier measures Merlin's witness state $|\psi\rangle$ to collapse it to a superposition over points in a random affine cube containing w, w', and w'' minus the affine plane containing w, w', and w''. Then the verifier uses non-collapsing measurements to collect all $(z, p(z))$ pairs in the affine cube (minus the affine plane) and interpolates a polynomial $u : \mathbb{F}_q^3 \to \mathbb{F}_q$ that fits these pairs. If the polynomial interpolation succeeds, Merlin passes the lines-point low-degree test and the verifier can learn $\pi(w)$ and $\pi(w')$ by evaluating the polynomial u.

1.3 Concurrent Work

In concurrent and independent work, Bassirian and Marwaha [17] show that QMA with non-collapsing measurements equals NEXP. They observe that the proof of $\mathsf{QMA}^+ = \mathsf{NEXP}$ [16] goes through even if one replaces the promise of a non-negative witness state with the ability for the verifier to perform non-collapsing measurements. [16] can prove a containment (for a constant completeness/soundness gap) using a constant number of non-collapsing measurements, whereas our verification procedure always uses $O(n \log n)$ non-collapsing measurements. Our approach readily extends to prove $\mathsf{DQMA} = \mathsf{NEXP}$ and $\mathsf{DQP/qpoly} = \mathsf{ALL}$.

2 Preliminaries

Throughout this work, we use the following notation. $\mathbb{N} := \{1, 2, 3, \ldots\}$ denotes the natural numbers. For a finite field $\mathbb{F}_q$, $\mathbb{F}_q^* := \mathbb{F}_q \setminus \{0\}$.

Let $\mathcal{L}$ denote the set of all affine lines (i.e., 1-dimensional affine subspaces) of $\mathbb{F}_q^n$. Recall that the line passing through points $a, b \in \mathbb{F}_q^n$ is the set $\{a + (b-a)t\}_{t \in \mathbb{F}_q}$. Therefore, a polynomial g that maps a line $\ell \in \mathcal{L}$ to $\mathbb{F}_q$ can be thought of as a univariate polynomial. We will use the following lines-point low-degree test of Friedl and Sudan [20] (see also the restatement in [21, Theorem 1.2]).

▶ **Lemma 1** (Multivariate low-degree test [20]). *Let $f : \mathbb{F}_q^n \to \mathbb{F}_q$ be any n-variate function, and let $G = \{g_\ell\}_{\ell \in \mathcal{L}}$ be a collection of degree-d polynomials $g_\ell : \ell \to \mathbb{F}_q$. There is a constant C large enough such that for any d satisfying $q > Cd$, if*

$$\Pr_{\substack{\ell \sim \mathcal{L} \\ z \sim \ell}} [f(z) \neq g_\ell(z)] \leq \delta,$$

for some $0 < \delta < 0.01$, then there exists an n-variate degree-d polynomial h such that

$$\Pr_{z \sim \mathbb{F}_q^n} [f(z) \neq h(z)] \leq 4\delta.$$

In words, Lemma 1 is saying that, if a function f restricted to a randomly chosen line ℓ disagrees with a low-degree polynomial on at most a δ fraction of points, then there must exist a globally low-degree polynomial that f disagrees with on at most a 4δ fraction of points. In short, Lemma 1 gives a local way to test that the entire function is low degree.

This work also involves affine planes and cubes (i.e., 2- and 3-dimensional affine subspaces). Recall that an affine plane is uniquely defined by three independent points $a, b, c \in \mathbb{F}_q^n$. The plane containing these points is the set

$$\{a + (b-a)t_1 + (c-a)t_2\}_{t_1, t_2 \in \mathbb{F}_q}.$$

Similarly, the unique affine cube containing the independent points $a, b, c, d \in \mathbb{F}_q^n$ is the set

$$\{a + (b - a)t_1 + (c - a)t_2 + (d - a)t_3\}_{t_1,t_2,t_3 \in \mathbb{F}_q}.$$

We now give a formal definition of PDQMA.

▶ **Definition 2.** PDQMA$_{c,s}$ *is the class of languages $L \subseteq \{0,1\}^*$ for which there exists a* PDQP *verifier V (to be defined shortly) such that, for all $x \in \{0,1\}^*$:*

- ***Completeness:*** *If $x \in L$ then there exists a witness state $|\phi\rangle$, on* poly(n) *qubits, such that $V(x, |\phi\rangle)$ accepts with probability at least c.*
- ***Soundness:*** *If $x \notin L$ then $V(x, |\phi\rangle)$ accepts with probability at most s for all witness states $|\phi\rangle$.*

A PDQP *verifier consists of two phases. In the first phase, a* P*-uniform quantum circuit C_x, depending on the input x, is applied to the initial state $|\phi\rangle|0^{p(n)}\rangle$, where $|\phi\rangle$ is the witness and p is a polynomial. This C_x can consist of three types of gates:* CNOT *gates,* 1*-qubit $\pi/8$ rotations, and measurements in the $\{|0\rangle, |1\rangle\}$ basis. The* CNOT *and $\pi/8$ gates provide universality for* BQP*, while the measurement gates introduce a probabilistic component. In the second phase, we consider the final state $|\psi\rangle$ of C_x, which depends in part on the probabilistic results of the measurement gates. Let $\mathcal{D}$ be the probability distribution induced by measuring all qubits of $|\psi\rangle$ in the computational basis and let q be a polynomial. Then a classical polynomial-time algorithm, A, receives as input x as well as $q(n)$ independent samples from $\mathcal{D}$, and then either accepts or rejects.*

Our proof that PDQMA = NEXP will rely on the characterization NEXP by probabilistically checkable proof (PCP) systems where the verifier only makes 2 queries to the PCP. Although this is well-known to classical complexity theorists, we explain this formally below for completeness. We begin by defining the complexity classes PCP.

▶ **Definition 3.** PCP$_{c,s}[r, q]_\Sigma$ *is the class of languages $L \subseteq \{0,1\}^*$ for which there exists a probabilistic polynomial-time verifier V_{PCP} which uses r random bits and makes q queries to an oracle π, each time receiving a response in some alphabet Σ such that*

1. ***Completeness:*** *If $x \in L$, then $\exists \pi$ such that $\Pr[V^{\pi}_{\mathsf{PCP}}(x) = 1] \geq c$.*
2. ***Soundness:*** *If $x \notin L$, then $\forall \pi$, $\Pr[V^{\pi}_{\mathsf{PCP}}(x) = 1] \leq s$.*

The celebrated PCP theorem [14, 13] tells us that NP = PCP$_{1,s}[O(\log(n)), 3]_{\{0,1\}}$ for any constant s. A simple consequence is that NEXP = PCP$_{1,s}[\mathrm{poly}(n), 3]_{\{0,1\}}$. It is folklore that PCP$_{c,s}[r, q]_\Sigma \subseteq$ PCP$_{c,s/q}[r + \log(q), 2]_{\Sigma^q}$, i.e., that the number of queries can be reduced to two at the cost of a larger alphabet and worse soundness error.

▶ **Theorem 4.** *For a size-*8 *alphabet Σ and any constant $s \in (0, 1)$,* PCP$_{1,s}[\mathrm{poly}(n), 2]_\Sigma$ = NEXP*.*

One can improve the soundness error to sub-constant by further increasing the alphabet size and our proofs that PDQMA = NEXP (Theorem 5) and DQMA = NEXP (Theorem 16) still go through with only slight (if any) modification.

3 QMA and Non-collapsing Measurements

We prove our main result: if QMA is modified so that Arthur can perform non-collapsing measurements in addition to standard quantum computation, then the resulting class equals NEXP.

The hard part is to show that $\mathsf{NEXP} \subseteq \mathsf{PDQMA}$, which we prove by simulating a two-query PCP system for NEXP. At a high level, the PDQMA verifier is given a PCP $\pi : \{0,1\}^n \to \Sigma$ encoded in a quantum proof, and it suffices to learn π at two points of the verifier's choosing.

Our starting point is Aaronson's result that $\mathsf{PDQP/qpoly} = \mathsf{ALL}$ [6] where he showed how to evaluate π at one point but this assumed that the verifier is provided with a *trusted* quantum advice state. Hence, our contribution is to show that one can retrieve *two* points of choice even if the quantum proof is from an *untrusted prover.*

▶ **Theorem 5.** $\mathsf{PDQMA} = \mathsf{NEXP}$.

Proof. $\mathsf{PDQMA} \subseteq \mathsf{NEXP}$ is clear, since in NEXP we can guess an exponentially-long classical description of the PDQMA witness and then verify it.

Thus, we show $\mathsf{NEXP} \subseteq \mathsf{PDQMA}$. By Theorem 4, it suffices to show $\mathsf{PCP}_{c,s}[\mathrm{poly}(n), 2]_\Sigma \subseteq \mathsf{PDQMA}$ for some constant-size alphabet Σ. In particular, we can assume the PCP verifier makes two queries to a PCP and receives responses in a constant-size alphabet. Let $\pi : \{0,1\}^n \to \Sigma$ be the Boolean function that encodes the PCP for all possible queries $x \in \{0,1\}^n$. Let $p : \mathbb{F}_q^n \to \mathbb{F}_q$ be the unique degree-n multilinear extension of π, where q is chosen so that the conditions of Lemma 1 are satisfied and $q \geq n+2$. Note that $q = O(n)$ by Bertrand's postulate. To simulate the PCP system, it suffices for the verifier to learn $\pi(w) = p(w)$ and $\pi(w') = p(w')$ at two points w and w' of the verifier's choosing, given a witness state $|\psi\rangle$ sent by Merlin.

We explain the verification procedure as we analyze the honest case (i.e., the case when there exists a π such that the PCP verifier accepts with probability at least c). Let $a, b \in \{0,1\}^n$ be distinct points and let $c \in \mathbb{F}_q^n$ be independent of a and b. Let A be the unique affine plane that contains a, b, and c. Let $C_{a,b,c}$ be the following function. For a vector $y \in \mathbb{F}_q^n$, if $y \in A$, then $C_{a,b,c}(y) = 0^n$. Otherwise, a, b, c, and y define a unique affine cube, which we denote by B. In this case, $C_{a,b,c}(y) = y' \in \mathbb{F}_q^n$ is a canonical representation of the point y, so that (a, b, c, y) and (a, b, c, y') define the same affine cube.

We describe the canonical representation in more detail. First, note that y must be one of the $q^3 - q^2$ points in the cube B that are not in the plane A (otherwise $C_{a,b,c}(y) = 0^n$). There are many ways to pick a canonical representative y'. For example, of the $q^3 - q^2$ many points, one can have $C_{a,b,c}(y)$ output the point y' with the fewest nonzero entries (and if there are ties, pick the one that comes first in lexicographic order). This ensures that all of the $q^3 - q^2$ points get mapped to the same canonical representative y', which is crucial for our verification procedure. Recall that $q = O(n)$, so $C_{a,b,c}$ can be computed efficiently.

The honest PDQMA witness is

$$\frac{1}{\sqrt{q^n}} \sum_{z \in \mathbb{F}_q^n} |z\rangle |p(z)\rangle.$$

Given this witness, the PDQP verification procedure is as follows:

(1) Simulating the PCP verifier, choose two queries $w, w' \in \{0,1\}^n$. Pick a point $w'' \in \mathbb{F}_q^n$ uniformly at random.

(2) Map the witness to

$$\frac{1}{\sqrt{q^n}} \sum_{z \in \mathbb{F}_q^n} |z\rangle |p(z)\rangle |C_{w,w',w''}(z)\rangle.$$

(3) Measure the $|C_{w,w',w''}(z)\rangle$ register in the usual collapsing way to obtain the outcome $y \in \mathbb{F}_q^n$. If the measurement outcome is 0^n, reject. Let B denote the affine cube containing w, w', w'', and y, and let A denote the affine plane containing w, w', and w''.

(4) Make $O(n^4)$ non-collapsing measurements of the $|z\rangle$ and $|p(z)\rangle$ registers.

(5) If exactly the $q^3 - q^2$ points in $B \setminus A$ are obtained and the empirical distribution over these points is $O(1/n)$-close in total variation distance to the uniform distribution, continue. Otherwise, reject.

(6) If more than one $p(z)$ value was obtained for the same z, reject.

(7) Perform polynomial interpolation to obtain trivariate polynomial $u_B : \mathbb{F}_q^3 \to \mathbb{F}_q$ of degree at most n that is consistent with p on the measured points. If this interpolation fails, then reject.

(8) Calculate $\pi(w) = p(w) = u(0,0,0)$ and $\pi(w') = p(w') = u(0,1,0)$. Plug these responses into the PCP verifier, and accept if and only if it does.

Let us analyze the verification procedure in more detail. Suppose, upon measuring the register $|C_{w,w',w''}(z)\rangle$ in Step 3, the verifier sees $y \in \mathbb{F}_q^n$. With probability q^{2-n}, we observe $y = 0^n$, because there are q^2 points on the affine plane $\{w+(w'-w)t_1+(w''-w)t_2\}_{t_1,t_2 \in \mathbb{F}_q}$. In this case, the verifier will reject (which is incorrect). Otherwise, with probability $1 - q^{2-n} = 1 - \exp(-\Omega(n))$, the verifier will see some canonical point $y \in \mathbb{F}_q^n$ so that the points (w, w', w'', y) defines an affine cube B.

The post-measurement state of the remaining two registers will then be in superposition over all points $z \in \mathbb{F}_q^n$ such that z, w, w', and w'' define the same affine cube as y, w, w', and w''. Recall that the affine cube B is the set

$$B = \{w + (y - w)t_1 + (w' - w)t_2 + (w'' - w)t_3\}_{t_1,t_2,t_3 \in \mathbb{F}_q},$$

and the affine plane $A \subseteq B$ containing w, w', and w'' is the set

$$A = \{w + (w' - w)t_1 + (w'' - w)t_2\}_{t_1,t_2 \in \mathbb{F}_q}.$$

Observe that z can be any of the $q^3 - q^2$ points in $B \setminus A$. In particular,

$$z = w + (y - w)t_1 + (w' - w)t_2 + (w'' - w)t_3$$

for any $t_1 \in \mathbb{F}_q^*$ and $t_2, t_3 \in \mathbb{F}_q$. Therefore, our post-measurement state $|\phi\rangle$ can be expressed as

$$\frac{1}{q\sqrt{q-1}} \sum_{\substack{t_1 \in \mathbb{F}_q^* \\ t_2 \in \mathbb{F}_q \\ t_3 \in \mathbb{F}_q}} |w+(y-w)t_1+(w'-w)t_2+(w''-w)t_3\rangle |p(w+(y-w')t_1+(w'-w)t_2+(w''-w)t_3)\rangle.$$

Define $u_B : \mathbb{F}_q^3 \to \mathbb{F}$ by $u(t_1, t_2, t_3) \coloneqq p(w + (y - w)t_1 + (w' - w)t_2 + (w'' - w)t_3)$, and notice that $u_B(0,0,0) = p(w)$ and $u_B(0,1,0) = p(w')$. Because p is the multilinear extension of π, we also have that $p(w) = \pi(w)$ and $p(w') = \pi(w')$.

After Step 6, the verifier has collected $q^3 - q^2$ pairs $(z, p(z))_{z \in B \setminus A}$. Collecting these pairs is an instance of the coupon collector's problem, so $O(q^3 \log q) = O(n^3 \log n)$ many samples suffice to succeed with high probability. We take more samples, which will be relevant to the soundness of our protocol. With the $(z, p(z))$ pairs, the verifier runs polynomial interpolation to learn the polynomial u_B. We note that q is chosen to be $\geq n + 2$ to ensure that $q^3 - q^2$ pairs suffice for polynomial interpolation. After learning u_B, the verifier has learned $\pi(w)$ and $\pi(w')$ as desired and will accept with the same probability as the PCP verifier.

A crucial part of our verification procedure is that the verifier tests that p is a low-degree polynomial. Because we picked a point $w'' \in \mathbb{F}_q^n$ uniformly at random, we ensure that the affine cube B contains a random affine line, independent of the points w and w'. Therefore, the polynomial interpolation succeeds if and only if the truth table of p matches a low-degree polynomial on a randomly chosen line ℓ. Hence, by Lemma 1, p must also be globally low degree. After taking into account the failures that can occur during the verification procedure, we conclude that the verifier accepts with probability at least $c - \exp(-\Omega(n))$.

We now analyze the soundness of our verification procedure. That is, suppose the PCP verifier will accept with probability at most s for all possible proofs π. We will show that the PDQMA verifier accepts with probability at most $s + \exp(-\Omega(n))$. The key insight for the soundness case is that, by deviating from the honest witness state above, Merlin only increases the probability that the polynomial interpolation will fail, causing Arthur to reject. In particular, the only way Merlin can cheat is to encode a truth table in $|\psi\rangle$ that is not degree n, but some function with much larger degree. However, the verifier will detect this with the lines-point low-degree test.

Before going through the technical details, let us emphasize that Merlin does not know the points w, w', and w'' that the verifier will select – these are chosen after Merlin commits to a witness state. Hence, Merlin must send a witness state $|\psi\rangle$ that passes all the checks in the verification procedure for all choices of w, w', and w'' and no matter the random outcome y the verifier observes in Step 3.

Formally, Merlin can send an arbitrary state:

$$|\psi\rangle = \sum_{z \in \mathbb{F}_q^n, b \in \mathbb{F}_q} \alpha_{z,b} |z\rangle |b\rangle.$$

The verifier maps the witness to

$$\sum_{z \in \mathbb{F}_q^n, b \in \mathbb{F}_q} \alpha_{z,b} |z\rangle |b\rangle |C_{w,w',w''}(z)\rangle,$$

and measures the last register. Suppose the measurement outcome is some $y \in \mathbb{F}_q^n$. If $y = 0^n$, the verifier rejects, but we will pessimistically assume this never happens. Suppose $y \neq 0^n$. As discussed previously, there are $q^3 - q^2$ points z such that z, w, w', and w'' define the same affine cube B as y, w, w', and w''. These correspond to the points in $B \setminus A$ (recall that A is the affine plane containing w, w', and w''). Define $D \subseteq B \setminus A$ to be the points $z \in B \setminus A$ for which there exists at least one nonzero $\alpha_{z,b}$ for some $b \in \mathbb{F}_q$. The post-measurement state $|\phi\rangle$ is then

$$|\phi\rangle = \sum_{z \in D, b \in \mathbb{F}_q} \widetilde{\alpha_{z,b}} |z\rangle |b\rangle,$$

where

$$\widetilde{\alpha_{z,b}} = \frac{|\alpha_{z,b}|}{\sqrt{\sum_{z \in D, b \in \mathbb{F}_q} |\alpha_{z,b}|^2}}.$$

Note that we can assume without loss of generality that $\widetilde{\alpha_{z,b}} \in \mathbb{R}$ as the verifier only performs non-collapsing measurements on $|\phi\rangle$. In fact, in Step 4, the verifier's actions can be understood as drawing samples from a classical probability distribution where each (z, b) pair has a probability of $|\widetilde{\alpha_{z,b}}|^2$.

Recall the well-known fact that $\Theta(\frac{n+\log(1/\delta)}{\varepsilon^2})$ samples are necessary and sufficient to learn a distribution to total variation distance at most ε with probability at least $1 - \delta$ (cf. [18, Theorem 1]). Hence, the $O(n^4)$ non-collapsing measurements in Step 5 suffice to learn the distribution over (z, b) pairs in the support of $|\phi\rangle$ to total variation distance at most

$O(1/n)$ with probability at least $1-\exp(-\Omega(n))$. In particular, for Steps 4 through 6 to pass, $|\phi\rangle$ must be (approximately) uniformly supported on pairs (z, b_z) for each $z \in B \setminus A$. (The verifier will immediately reject if any $z \in B \setminus A$ is paired with more than one $b \in \mathbb{F}_q$ value.) Because this must hold for any affine plane A and affine cube B, we can deduce that Merlin is forced to send a state that is (approximately) uniform over pairs (z, b_z) for every $z \in \mathbb{F}_q^n$.

Assuming these steps pass, Step 7 passes only if the observed pairs $(z, b_z)_{z \in B \setminus A}$ fit a degree-n trivariate polynomial u_B. As discussed in the honest case, by selecting a $w'' \in \mathbb{F}_q^n$ uniformly at random, we guarantee that the affine cube B contains a random line ℓ, independent of the w and w' (this is precisely why we need a 3-dimensional affine subspace). Therefore, the interpolation succeeding implies that the function encoded by Merlin matches a degree-n polynomial on a randomly chosen line. By Lemma 1, we can conclude that Merlin indeed encoded a truth table for a degree-n polynomial.

If all of these steps pass, then the verifier can learn $\pi(w)$ and $\pi(w')$ as desired by evaluating $u_B(0,0,0)$ and $u_B(0,1,0)$. By plugging the values $\pi(w)$ and $\pi(w')$ into the PCP verifier, the PDQMA verifier accepts with probability at most s. ◀

4 QMA and Hidden Variables

We introduce and characterize the complexity class DQMA, a variant of QMA where the verifier can perform DQP computations. Informally, DQP is like BQP with the ability to inspect the entire history of a hidden variable. For completeness, we begin this section by giving a formal definition of DQMA. Then, as with PDQMA, we prove that DQMA = NEXP. This result is more "quantum" than PDQMA = NEXP (Theorem 5) because the DQP verifier will use quantum circuits (as opposed to merely computational basis measurements).

4.1 The Complexity Class DQMA

We give a formal definition of DQMA. To do so, we must recall a few definitions related to the class DQP (see [4] for more detail about this class).

We begin by defining hidden-variable theories. To aid intuition, one can think of hidden-variable theories like standard quantum mechanics where there is a state vector that evolves unitarily. However, there is also a deeper "hidden variable" in some definite state (i.e., not in superposition) that evolves stochastically in a manner determined by the state vector and the state vector's unitary evolution. In the series of definitions that follow, we are building up to defining a model of computation where one evolves a state vector unitarily, and then (at the end of the computation) can inspect which states the hidden variable was in at each step of the computation.

▶ **Definition 6** (Hidden-variable theory). *A hidden-variable theory is a family of functions $\{S_d\}_{d\in\mathbb{N}}$, where each S_d maps a d-dimensional mixed state ρ and a $d \times d$ unitary matrix U onto a singly stochastic matrix $S_d(\rho, U)$.*

That is, we take a hidden-variable theory to be a function that maps the unitary evolution of a state to a stochastic matrix that evolves one probability distribution to another. Conditioned on a hidden variable being in a state $|j\rangle$, $(S)_{ij}$ is the probability that a hidden variable transitions to the $|j\rangle$.

Aaronson [4] defined a number of axioms a hidden-variable theory could satisfy. We require three of these axioms to define DQP: the marginalization axiom, the indifference axiom, and the robustness axiom. In the following definitions, S denotes the hidden-variable theory, ρ a d-dimensional quantum state, and U a $d \times d$ unitary matrix. Each axiom must hold for all $d \in \mathbb{N}$.

▶ **Definition 7** (Marginalization axiom). *The marginalization axiom says that for all $j \in \{1, \dots, d\}$,*

$$\sum_i (S)_{ij}(\rho)_{ii} = (U\rho U^\dagger)_{jj}.$$

In words, the marginalization axiom says that the hidden-variable theory should make predictions that are consistent with quantum mechanics.

▶ **Definition 8** (Indifference axiom). *For a matrix $M \in \mathbb{C}^{d\times d}$, let a block be a subset $B \subseteq \{1, \dots, d\}$ such that $(M)_{ij} = 0$ for all (i, j) such that $i \in B$, $j \notin B$ and $i \notin B$, $j \in B$. The indifference axiom says that the stochastic matrix $S(\rho, U)$ must have the same blocks as U.*

Physically, the indifference axiom says the following. Given any quantum state ρ in a tensor product space $\mathcal{H}_A \otimes \mathcal{H}_B$ and any unitary U acting nontrivially only on $\mathcal{H}_A$, the stochastic matrix $S(\rho, U)$ acts nontrivially only on $\mathcal{H}_A$ as well.

Finally, we state the robustness axiom, for which we need the following notation. Let $P(\rho, U)$ be the matrix of joint probabilities whose (i, j) entry is $(P)_{ij} \coloneqq (S)_{ij}(\rho)_{ii}$.

▶ **Definition 9** (Robustness axiom). *Let $\widetilde{\rho}$ and $\widetilde{U}$ be perturbations of ρ and U, respectively, and, for a matrix M, let $\|M\|_\infty \coloneqq \max_{i,j} |(M)_{ij}|$. The robustness axiom says that, for all polynomials p, there should exist a polynomial q such that,*

$$\|P(\widetilde{\rho}, \widetilde{U}) - P(\rho, U)\|_\infty \le \frac{1}{p(d)},$$

whenever $\|\widetilde{\rho} - \rho\|_\infty \le 1/q(d)$ and $\|\widetilde{U} - U\| \le 1/q(d)$.

The robustness axiom is necessary to prove that the class DQP is not sensitive to the choice of gate set defining the class.

Next, we define the history of a hidden variable.

▶ **Definition 10** (Hidden variable history). *Let $|\psi_{\text{init}}\rangle$ be an n-qubit quantum state, and let $U \coloneqq U_T \cdots U_1$ be an n-qubit, depth-T quantum circuit, where $U_1, \dots, U_T$ denote each layer of the quantum circuit U. The history of a hidden variable is a sequence $H = (v_0, \dots, v_T)$ of basis states, where v_t is the state of the hidden variable immediately after the layer U_t of the circuit is applied. Given a hidden-variable theory $\mathcal{T} \coloneqq \{S_d\}_{d\in\mathbb{N}}$, we obtain a probability distribution over hidden variable histories $\Omega(\mathcal{T}, U, |\psi_{\text{init}}\rangle)$ via the stochastic matrices*

$$S(|\psi_{\text{init}}\rangle, U_1), S(U_1|\psi_{\text{init}}\rangle, U_2), \dots, S(U_{T-1} \cdots U_1|\psi_{\text{init}}\rangle, U_T).$$

We can now define the complexity class DQMA.

▶ **Definition 11.** DQMA(c, s) *is the class of languages $L \subseteq \{0,1\}^*$ for which there exists a* DQP *verifier V (to be defined shortly) such that, for all $x \in \{0,1\}^*$:*

- ***Completeness:*** *If $x \in L$ then there exists a witness state $|\phi\rangle$, on* $\mathrm{poly}(n)$ *qubits, such that $V(x, |\phi\rangle)$ accepts with probability at least c.*
- ***Soundness:*** *If $x \notin L$ then $V(x, |\phi\rangle)$ accepts with probability at most s for all witness states $|\phi\rangle$.*

A DQP *verifier is defined as follows. Let $\mathcal{T}$ be a hidden-variable theory satisfying the marginalization, indifference, and robustness axioms (Definitions 7–9), and let C_x (depending on the input x) be a* P*-uniform quantum circuit comprised of gates from any finite gate set that is universal for* BQP*. A* DQP *verifier is a deterministic classical Turing machine that is allowed to draw one sample from the distribution $\Omega\left(\mathcal{T}, C_x, |\phi\rangle|0^{p(n)}\rangle\right)$ (Definition 10), where $|\phi\rangle$ is the witness and p is a polynomial.*

4.2 DQMA = NEXP and DQP/qpoly = ALL

We conclude this section by proving DQMA = NEXP and DQP/qpoly = ALL. Recall that in the proof of PDQMA = NEXP (Theorem 5), the verifier uses non-collapsing measurements to sample $O(n \log n)$ values of a multivariate polynomial along an affine line of one's choice and then does interpolation. To prove DQMA = NEXP, we use the same verification procedure, except we use the history of a hidden variable to collect the samples in place of non-collapsing measurements. Hence, to prove DQMA = NEXP it suffices to explain how we collect these samples with the history of a hidden variable. We achieve this by generalizing the "juggle subroutine" due to Aaronson [4, Section VII].

▶ **Lemma 12** (Juggle subroutine [4, Section VII]). *Suppose we have an ℓ-qubit state*

$$\frac{|a\rangle \pm |b\rangle}{\sqrt{2}},$$

where $|a\rangle$ and $|b\rangle$ are unknown basis states. Given a single copy of the state, the juggle subroutine (a DQP algorithm) can efficiently learn both a and b with success probability at least $1 - e^{-\ell}$.

We generalize this algorithm to work on states that are an equal superposition over polynomially many strings by reducing to the case of an equal superposition over two strings. Before explaining the generalization, we first must define (and give a simple fact about) pairwise independent families of hash functions, which are used in our reduction.

▶ **Definition 13** (Pairwise independent family of hash functions). *A family of hash functions $\mathcal{H} = \{h : \{0,1\}^\ell \to R\}$ is called pairwise independent if $\forall\ x \neq y \in \{0,1\}^\ell$ and $\forall a_1, a_2 \in R$, we have*

$$\Pr[h(x) = a_1 \wedge h(y) = a_2] = \frac{1}{|R|^2}.$$

▷ Claim 14. Let $S \subseteq \{0,1\}^\ell$ be a subset, and let $\mathcal{H} := \{h : \{0,1\}^\ell \to R\}$ be a family of pairwise independent hash functions such that $2|S| - 4 \leq |R| \leq 3|S| - 3$. Then, for any fixed $x \in S$, the probability that x collides with exactly one other $y \in S$ is at least $\frac{1}{6}$.

Proof. Let $h \in \mathcal{H}$ be chosen uniformly at random, and let $x \in S$ be some fixed element. The probability that exactly one other element collides with x is

$$\frac{|S|-1}{|R|} \cdot \left(1 - \frac{1}{|R|}\right)^{|S|-2}.$$

We lower bound this quantity.

$$\begin{aligned}\frac{|S|-1}{|R|} \cdot \left(1 - \frac{1}{|R|}\right)^{|S|-2} &\geq \frac{|S|-1}{|R|} \cdot \left(1 - \frac{|S|-2}{|R|}\right)\\ &\geq \frac{1}{3} \cdot \left(1 - \frac{|S|-2}{|R|}\right)\\ &\geq \frac{1}{6}.\end{aligned}$$

The first inequality follows from Bernoulli's inequality. The second and third inequalities use the fact that $2|S| - 4 \leq |R| \leq 3|S| - 3$. ◁

We now give the generalized juggle subroutine.

▶ **Lemma 15** (Generalized juggle subroutine). *Suppose we have an ℓ-qubit state*

$$\frac{1}{\sqrt{|S|}}\sum_{x\in S}|x\rangle,$$

where $S \subseteq \{0,1\}^\ell$ is an unknown subset of $|S| \leq \mathrm{poly}(\ell)$ basis states. Given a single copy of the state, the generalized juggle subroutine (a DQP *algorithm) can efficiently learn S with success probability at least $1-e^{-\ell}$.*

Proof. We begin with the state of the form

$$|\psi\rangle = \frac{1}{\sqrt{|S|}}\sum_{x\in S}|x\rangle.$$

We perform a procedure that involves applying transformations to $|\psi\rangle$ and then inverting them to get back to $|\psi\rangle$, in an attempt to "dislodge" the hidden variable from whichever basis state it's currently sitting in, and get it into a different, uniformly random one. Each time we do this, we have a $1/\mathrm{poly}(\ell)$ probability of success. Importantly, since there's no penalty for failure, we can repeat this procedure $\mathrm{poly}(\ell)$ times, and then with overwhelming probability, the hidden variable will have visited every basis state $|x\rangle$ for $x \in S$. Therefore, one will learn S upon observing the hidden-variable history.

The procedure works as follows. First, choose a random hash function h (with a range satisfying the conditions in Claim 14) from some pairwise-independent family, and then map each $|x\rangle$ to $|x\rangle|h(x)\rangle$. Let y be the current state of the hidden variable. By Claim 14, with probability at least $\frac{1}{6}$, there's *exactly* one other basis state $z \neq y$ such that $h(z) = h(y)$, and this z is uniformly random. If that happens, then because of the indifference axiom (which says that, if we don't touch the h-register, then the hidden variable will never move between h-values), we've reduced to the problem handled by the original juggle subroutine. In particular, we can now run the original juggle subroutine on the first register (where the hidden variable is). Lemma 12 tells us that, with probability at least $1-e^{-\ell}$, the hidden variable moves from y to z. Finally, we uncompute h, leaving us with our original state $|\psi\rangle$. Overall, the inner loop moves the hidden variable from y to a uniformly random z with probability at least $\frac{1}{6} - \frac{e^{-\ell}}{6} \geq 1/\mathrm{poly}(\ell)$.[3] Since there is no penalty for failure, we can repeat this procedure $2\ell^2$ times to ensure that with probability at least $1-e^{-\ell}$, the hidden variable was successfully moved to a uniformly random basis state. Finally, since we visit a uniformly random state with high probability, we can visit every state with high probability by repeating this entire procedure a polynomial number of times. ◀

We are now ready to prove the main theorem of this section. Namely, that giving Arthur access to hidden-variable histories blows up the power of QMA to NEXP.

▶ **Theorem 16.** DQMA = NEXP.

Proof. It is clear that DQMA $\subseteq$ NEXP. By Theorem 4, we can complete the proof by showing MIP $\subseteq$ DQMA.

The verification procedure and the honest witness sent by Merlin are the same as in the proof of Theorem 5. Suppose we have a yes-instance, and Merlin sends the honest witness with the form:

$$\frac{1}{q^n}\sum_{z\in\mathbb{F}_q^n}|z\rangle|p(z)\rangle.$$

[3] We note that there is some chance that the hidden variable stays put or moves to somewhere other than z, but that's OK too, since our ultimate goal is for the hidden variable to visit every possible state in S. In any case, we keep repeating.

We must explain how to use the history of a hidden variable in lieu of non-collapsing measurements. After the verifier makes *collapsing* measurements, they must learn the support of the post-measurement state. To do this, the verifier runs the generalized juggle subroutine (Lemma 15). After that, the verifier can do polynomial interpolation with just classical computation (or reject if there is insufficient data to do the interpolation).

The only difference in the verification procedure is that the data for polynomial interpolation is collected via the generalized juggle subroutine instead of non-collapsing measurements. Therefore, the completeness and soundness of this procedure follow in the same way as for Theorem 5. In particular, if Merlin deviates from the honest witness state, then he can only hurt his success probability by causing the polynomial interpolation step to fail. ◀

Finally, we remark that our generalized juggle subroutine can be used to prove DQP/qpoly = ALL, complementing Aaronson's result that PDQP/qpoly = ALL [6] and Raz's result that QIP(2)/qpoly = ALL [24]. Similar to Theorem 16, this result is more "quantum" than Aaronson's or Raz's, because the generalized juggle subroutine requires quantum computation.

▶ **Theorem 17.** DQP/qpoly = ALL.

Proof. The advice state and verification procedure are the same as in [6, Theorem 2], except we replace the non-collapsing measurements with the generalized juggle subroutine in the same manner described in the proof of Theorem 16. ◀

5 Open Problems

Is PDQP $\subseteq$ DQP? Aaronson et al. [8] give intuition for why this containment ought to be true, but it remains an open problem. Proving PDQP $\subseteq$ DQP (combined with Theorem 5) immediately implies DQMA = NEXP, simplifying our proof in Theorem 16. We also remark that improving the upper bound DQP $\subseteq$ EXP remains an interesting open problem.

We now know that modifying QMA by giving the verifier access to non-collapsing measurements, hidden-variable histories, or non-negative witnesses will cause QMA to "explode" in power to NEXP. Is it possible to replace the verifier with some variant that does *not* lead to NEXP? Having access to such variants may find applications in proving better bounds on QMA(2).

References

1 S. Aaronson. Quantum Lower Bound for the Collision Problem. In *Proc. ACM STOC*, pages 635–642, 2002. quant-ph/0111102.

2 S. Aaronson. Is Quantum Mechanics An Island In Theoryspace? In A. Khrennikov, editor, *Proceedings of the Växjö Conference "Quantum Theory: Reconsideration of Foundations"*, 2004. quant-ph/0401062.

3 S. Aaronson. Limitations of Quantum Advice and One-Way Communication. *Theory of Computing*, 1:1–28, 2005. Earlier version in CCC'2004. quant-ph/0402095. doi:10.4086/TOC.2005.V001A001.

4 S. Aaronson. Quantum Computing and Hidden Variables. *Phys. Rev. A*, 71(032325), 2005. quant-ph/0408035 and quant-ph/0408119.

5 S. Aaronson. Quantum Computing, Postselection, and Probabilistic Polynomial-Time. *Proc. Roy. Soc. London*, A461(2063):3473–3482, 2005. quant-ph/0412187.

6 S. Aaronson. PDQP/qpoly = ALL. arXiv:1805.08577, 2018. arXiv:1805.08577.

7 S. Aaronson, S. Beigi, A. Drucker, B. Fefferman, and P. Shor. The power of unentanglement. In *Proc. Conference on Computational Complexity*, pages 223–236, 2008. arXiv:0804.0802.

8 S. Aaronson, A. Bouland, J. Fitzsimons, and M. Lee. The space "just above" BQP. In *Proc. Innovations in Theoretical Computer Science (ITCS)*, pages 271–280, 2016. arXiv:1412.6507.

9 S. Aaronson and A. Drucker. A Full Characterization of Quantum Advice. *SIAM J. Comput.*, 43(3):1131–1183, 2014. Earlier version in STOC'2010. arXiv:1004.0377.

10 S. Aaronson and J. Watrous. Closed Timelike Curves Make Quantum and Classical Computing Equivalent. *Proc. Roy. Soc. London*, A465:631–647, 2009. arXiv:0808.2669.

11 D. S. Abrams and S. Lloyd. Nonlinear quantum mechanics implies polynomial-time solution for NP-complete and #P problems. *Phys. Rev. Lett.*, 81:3992–3995, 1998. quant-ph/9801041.

12 D. Aharonov and O. Regev. A Lattice Problem in Quantum NP. In *Proc. IEEE FOCS*, pages 210–219, 2003. quant-ph/0307220.

13 S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof Verification and the Hardness of Approximation Problems. *J. of the ACM*, 45(3):501–555, 1998. doi:10.1145/278298.278306.

14 S. Arora and S. Safra. Probabilistic Checking of Proofs: A New Characterization of NP. *Journal of the ACM (JACM)*, 45(1):70–122, 1998. doi:10.1145/273865.273901.

15 J. Barrett, N. de Beaudrap, M. J. Hoban, and C. M. Lee. The computational landscape of general physical theories. *npj Quantum Information*, 5(1):41, 2019. doi:10.1038/s41534-019-0156-9.

16 R. Bassirian, B. Fefferman, and K. Marwaha. Quantum merlin-arthur and proofs without relative phase, 2023. arXiv:2306.13247.

17 R. Bassirian and K. Marwaha. Superposition detection and QMA with non-collapsing measurements, 2024. arXiv:2403.02532.

18 C. L. Canonne. A short note on learning discrete distributions, 2020. arXiv:2002.11457.

19 D. Dieks. Modal interpretation of quantum mechanics, measurements, and macroscopic behaviour. *Phys. Rev. A*, 49:2290–2300, 1994.

20 K. Friedl and M. Sudan. Some improvements to total degree tests. In *Proceedings Third Israel Symposium on the Theory of Computing and Systems*, pages 190–198. IEEE, 1995. doi:10.1109/ISTCS.1995.377032.

21 P. Harsha, M. Kumar, R. Saptharishi, and M. Sudan. An improved line-point low-degree test, 2023. arXiv:2311.12752.

22 F. G. Jeronimo and P. Wu. The Power of Unentangled Quantum Proofs with Non-Negative Amplitudes. In *Proc. ACM STOC*, pages 1629–1642, 2023. doi:10.1145/3564246.3585248.

23 C. M. Lee and J. Barrett. Computation in generalised probabilisitic theories. *New Journal of Physics*, 17(8):083001, 2015. doi:10.1088/1367-2630/17/8/083001.

24 R. Raz. Quantum Information and the PCP Theorem. *Algorithmica*, 55(3):462–489, 2009. Earlier version in FOCS'2005. quant-ph/0504075. doi:10.1007/S00453-007-9033-6.

25 T. Rudolph and L. Grover. A 2 rebit gate universal for quantum computing, 2002. arXiv:quant-ph/0210187.

26 E. Schrödinger. Über die Umkehrung der Naturgesetze. *Sitzungsberichte der Preussischen Akademie der Wissenschaften, Physikalisch-Mathematische Klasse*, 8(9):144–153, 1931. English title: On the Reversal of the Laws of Nature.