comput. complex. (2022) 31:13
© The Author(s), under exclusive licence to Springer Nature Switzerland AG, 2022, corrected publication 2022
1016-3328/22/020001-52
published online September 20, 2022
https://doi.org/10.1007/s00037-022-00231-8

computational complexity

QUANTUM GENERALIZATIONS OF THE POLYNOMIAL HIERARCHY WITH APPLICATIONS TO QMA(2)

Sevag Gharibian, Miklos Santha, Jamie Sikora, Aarthi Sundaram, and Justin Yirka

Abstract. The polynomial-time hierarchy (PH) has proven to be a powerful tool for providing separations in computational complexity theory (modulo standard conjectures such as PH does not collapse). Here, we study whether two quantum generalizations of PH can similarly prove separations in the quantum setting. The first generalization, QCPH, uses classical proofs, and the second, QPH, uses quantum proofs. For the former, we show quantum variants of the Karp–Lipton theorem and Toda's theorem. For the latter, we place its third level, $\mathrm{Q}\Sigma_3$, into NEXP using the ellipsoid method for efficiently solving semidefinite programs. These results yield two implications for QMA(2), the variant of Quantum Merlin-Arthur (QMA) with two unentangled proofs, a complexity class whose characterization has proven difficult. First, if QCPH = QPH (i.e., alternating quantifiers are sufficiently powerful so as to make classical and quantum proofs "equivalent"), then QMA(2) is in the counting hierarchy (specifically, in $\mathrm{P}^{\mathrm{PP}^{\mathrm{PP}}}$). Second, because $\mathrm{QMA}(2) \subseteq \mathrm{Q}\Sigma_3$, QMA(2) is strictly contained in NEXP unless $\mathrm{QMA}(2) = \mathrm{Q}\Sigma_3$ (i.e., alternating quantifiers do not help in the presence of "unentanglement").

Keywords. Complexity theory, Quantum computing, Polynomial hierarchy, QMA(2), Semidefinite programming, Toda's theorem

Subject classification. 68Q12 Quantum algorithms and complexity in the theory of computing, 68Q15 Complexity classes, 90C22 Semidefinite programming

1. Introduction

The polynomial-time hierarchy (PH) (Meyer & Stockmeyer 1972) is a staple of computational complexity theory, and generalizes P, NP and coNP with the use of alternating existential ($\exists$) and universal ($\forall$) operators. Roughly, a language $L \subseteq \{0,1\}^*$ is in Σ_i, the i-th level of PH, if there exists a polynomial-time deterministic Turing machine M that acts as a verifier and accepts i proofs $y_1, \dots, y_i$, each polynomially bounded in the length of the input x, such that:

$$x \in L \Rightarrow \exists y_1 \forall y_2 \exists y_3 \cdots Q_i y_i \text{ such that } M \text{ accepts } (x, y_1, \dots, y_i),$$
$$x \notin L \Rightarrow \forall y_1 \exists y_2 \forall y_3 \cdots \overline{Q}_i y_i \text{ such that } M \text{ rejects } (x, y_1, \dots, y_i),$$

where $Q_i = \exists$ if i is odd and $Q_i = \forall$ if i is even, and $\overline{Q}$ denotes the complement of Q. Then, PH is defined as the union over all Σ_i for all $i \in \mathbb{N}$. The study of PH has proven remarkably fruitful in the classical setting, from celebrated results such as Toda's theorem (1991), which shows that PH is contained in $\mathrm{P}^{\#\mathrm{P}}$, to the Karp–Lipton theorem (1980), which says that unless PH collapses to its second level, NP does not have polynomial-size non-uniform circuits.

As PH has played a role in separating complexity classes (assuming standard conjectures like "PH does not collapse"), it is natural to ask whether quantum generalizations of PH can be used to separate *quantum* complexity classes. Here, there is some flexibility in defining "quantum PH," as there is more than one well-defined notion of "quantum NP": The first, Quantum-Classical Merlin-Arthur (QCMA) (Aharonov & Naveh 2002), is a quantum analogue of Merlin-Arthur (MA) with a classical proof but quantum verifier. The second, Quantum Merlin-Arthur (QMA) (Kitaev *et al.* 2002), is QCMA except with a quantum proof. Generalizing each of these definitions leads to (at least) two possible definitions for "quantum PH," the first using classical proofs (denoted QCPH), and the second using quantum proofs (denoted QPH) (formal definitions in Section 2).

With these definitions in hand, our aim is to separate quantum classes whose complexity characterization has generally been difficult to pin down. A prime example is QMA(2), the variant of QMA with two "unentangled" quantum provers. While the classical analogue of QMA(2) (i.e., an MA proof system with two provers) equals MA, in the quantum regime multiple unentangled provers are conjectured to yield a more powerful proof system (e.g., there exist problems in QMA(2) not known to be in QMA), see Aaronson *et al.* (2009); Beigi (2010); Blier & Tapp (2009); Liu *et al.* (2007). For this reason, QMA(2) has received much attention, despite which the strongest bounds known on QMA(2) remain the trivial ones:

$$\mathrm{QMA} \subseteq \mathrm{QMA}(2) \subseteq \mathrm{NEXP}.$$

(Note: QMA ⊆ PP, see Gharibian & Yirka 2019; Kitaev & Watrous 2000; Marriott & Watrous 2005; Vyalyi 2003.) In this work, we show that, indeed, results about the structure of QCPH or QPH yield implications about the power of QMA(2).

1.1. Results, techniques, and discussion. We begin by informally defining the two quantum generalizations of PH to be studied (formal definitions in Section 2).

How to define a "quantum PH"? The first definition, QCPH, has its i-th level $\mathrm{QC}\Sigma_i$ defined analogously to Σ_i, except we replace the Turing machine M with a polynomial-size uniformly generated quantum circuit V such that:

$$\begin{aligned} &(1.1) \quad x \in A_{\text{yes}} \Rightarrow \\ &\qquad\qquad \exists y_1 \forall y_2 \exists y_3 \cdots Q_i y_i \text{ such that} \\ &\qquad\qquad V \text{ accepts } (x, y_1, \ldots, y_i) \text{ with probability } \geq 2/3, \\ &(1.2) \quad x \in A_{\text{no}} \Rightarrow \\ &\qquad\qquad \forall y_1 \exists y_2 \forall y_3 \cdots \overline{Q}_i y_i \text{ such that} \\ &\qquad\qquad V \text{ accepts } (x, y_1, \ldots, y_i) \text{ with probability } \leq 1/3 \end{aligned}$$

where the use of a language L has been replaced with a *promise problem*[1] $A = (A_{\text{yes}}, A_{\text{no}})$ (since $\mathrm{QC}\Sigma_i$ uses a bounded error ver-

[1] Recall that unlike a decision problem, for a promise problem $A = (A_{\text{yes}}, A_{\text{no}})$, it is not necessarily true that for all inputs $x \in \Sigma^*$, either $x \in A_{\text{yes}}$ or $x \in A_{\text{no}}$. In the case of proof systems such as QCPH, when $x \notin A_{\text{yes}} \cup A_{\text{no}}$, V can output an arbitrary answer. Additionally, $A_{\text{yes}} \cap A_{\text{no}} = \emptyset$.

ifier). The values $(2/3, 1/3)$ are *completeness* and *soundness* parameters for A and the interval $(1/3, 2/3)$ where no acceptance probabilities are present is termed the *promise gap* for A. Notice that QCPH, defined as $\bigcup_{i\in\mathbb{N}} \mathrm{QC}\Sigma_i$, is a generalization of QCMA in that $\mathrm{QC}\Sigma_1 = \mathrm{QCMA}$.

We next define QPH using *quantum* proofs. Here, however, there are various possible definitions one might consider. Can the quantum proofs be entangled between alternating quantifiers? If not, we are enforcing "unentanglement" as in QMA(2). Allowing entanglement, on the other hand, might weaken the class, just as $\mathrm{QMA} \subseteq \mathrm{QMA}(2)$. Assuming proofs are unentangled, should the proofs be pure or mixed quantum states? (Mixed states are the more general and physical definition. For QMA and QMA(2), standard convexity arguments show both classes of proofs are equivalent, but such arguments fail when *alternating* quantifiers are allowed. For example, consider the predicate "$\exists x \forall y$ the Swap Test on (x, y) accepts with probability p." When x and y are pure states, this predicate is false for any choice of $p > \frac{1}{2}$. When x and y are mixed, the predicate is true for $p = \frac{1}{2} + 2^{-n-1}$ by using the fully mixed state $x = I/2^n$.)

Here, we define QPH to have its i-th level, $\mathrm{Q}\Sigma_i$, defined similarly to $\mathrm{QC}\Sigma_i$, except each classical proof y_j is replaced with a mixed quantum state ρ_j on polynomially many qubits (for clarity, each ρ_j acts on a disjoint set of qubits, making the ρ_j unentangled). We say a promise problem $A = (A_{\mathrm{yes}}, A_{\mathrm{no}})$ is in $\mathrm{Q}\Sigma_i$ if it satisfies the following conditions:

$$\begin{aligned}
&x \in A_{\mathrm{yes}} \Rightarrow \\
&\qquad \exists \rho_1 \forall \rho_2 \exists \rho_3 \cdots Q_i \rho_i \text{ such that} \\
&\qquad V \text{ accepts } (x, \rho_1, \ldots, \rho_i) \text{ with probability } \geq 2/3, \\
&x \in A_{\mathrm{no}} \Rightarrow \\
&\qquad \forall \rho_1 \exists \rho_2 \forall \rho_3 \cdots \overline{Q}_i \rho_i \text{ such that} \\
&\qquad V \text{ accepts } (x, \rho_1, \ldots, \rho_i) \text{ with probability } \leq 1/3.
\end{aligned}$$

Note that $\mathrm{QMA} = \mathrm{Q}\Sigma_1$ and $\mathrm{QMA}(2) \subseteq \mathrm{Q}\Sigma_3$ (simply ignore the second proof).

Our results are now stated as follows under three headings.

1. An analogue of Toda's theorem for QCPH. As previously mentioned, PH is one way to generalize NP using alternations. Another approach is to count the number of solutions for an NP-complete problem such as SAT, as captured by #P. Surprisingly, these two notions are related, as shown by the following celebrated theorem of Toda.

Theorem 1.3 (Toda 1991). $\mathrm{PH} \subseteq \mathrm{P}^{\#\mathrm{P}}$.

In the quantum setting, for QCPH, it can be shown using standard arguments involving enumeration over classical proofs that $\mathrm{QCPH} \subseteq \mathrm{PSPACE}$. However, here we show a stronger result.

Theorem 1.4 (A quantum-classical analogue of Toda's theorem). $\mathrm{QCPH} \subseteq \mathrm{P}^{\mathrm{PP}^{\mathrm{PP}}}$.

Thus, we almost recover the original bound of Toda's theorem[2], except we require an oracle for the *second* level of the counting hierarchy (CH). CHcan be defined with its first level as $\mathbf{C}_1 = \mathrm{PP}$ and its kth level for $k \geq 2$ as $\mathbf{C}_k = \mathrm{PP}^{\mathbf{C}_{k-1}}$.

Why did we move up to the next level of CH? There are two difficulties in dealing with QCPH (see Section 3 for a detailed discussion). The first can be sketched as follows. Classically, many results involving PH, from basic ones implying the collapse of PH to more advanced statements such as Toda's theorem, use the following recursive idea (demonstrated with Σ_2 for simplicity): By fixing the existentially quantified proof of Σ_2 the remnant reduces to a coNP problem, i.e., we can recurse to a lower level of PH. In the quantum setting, however, this does not hold—fixing the existentially quantified proof for $\mathrm{QC}\Sigma_2$ does *not* necessarily yield a coQCMA problem as some acceptance probabilities may fall in the $(1/3, 2/3)$ promise gap which cannot happen for a problem in coQCMA. (This is due to the same phenomenon that has been an obstacle to resolving whether $\exists \cdot \mathrm{BPP}$ equals MA (see Section 1.2 and Remark 3.12).) Thus, we cannot directly generalize recursive arguments from the classical setting to the quantum setting. The

[2] PP captures all problems for which a majority of all possible answers is correct and it is known that $\mathrm{P}^{\mathrm{PP}} = \mathrm{P}^{\#\mathrm{P}}$.

second difficulty is trickier to explain briefly (see Section 3.2 for details). Roughly, Toda's proof that $\mathrm{PH} \subseteq \mathrm{P}^{\mathrm{PP}}$ crucially relies on the Valiant–Vazirani (VV) theorem (1986), which has one-sided error (i.e., VV may map YES instances of SAT to NO instances of UNIQUE-SAT, but NO instances of SAT are always mapped to NO instances of UNIQUE-SAT). The VV theorem for QCMA (Aharonov *et al.* 2022) also has this property, but in addition it can output instances which are "invalid." Roughly, an "invalid" instance of a promise problem $\mathcal{P}$ is an instance violating the promise of $\mathcal{P}$. Such instances pose a problem, because feeding an oracle an invalid instance results in an arbitrary output; coupled with the two-sided error which arises in QCPH due to the presence of alternating quantifiers, it is unclear how to extend the parity arguments used in Toda's proof to the QCPH setting.

To circumvent these difficulties, we exploit a high-level idea from Gharibian & Yirka (2019), where an oracle for SPECTRAL GAP[3] was used to detect "invalid" QMA instances[4]. In our setting, the "correct" choice of oracle turns out to be a Precise-BQP oracle[5], where Precise-BQP is roughly BQP with an inverse exponentially small promise gap. Using this, we are able to essentially "remove" the promise gap of QCPH altogether, thus recovering a "decision problem" which does not pose the difficulties above. Specifically, this mapping is achieved by Lemma 3.13 (Cleaning Lemma), which shows that $\forall i \in \mathbb{N}$, we have $\mathrm{QC}\Sigma_i \subseteq \exists \cdot \forall \cdot \cdots \cdot Q_i \cdot \mathrm{P}^{\mathrm{PP}}$. The latter

[3] This problem determines whether the spectral gap of a given local Hamiltonian is "small" or "large."

[4] This was used, in turn, to show in conjunction with Ambainis (2014) that SPECTRAL GAP is $\mathrm{P}^{\text{Unique-QMA[log]}}$-hard.

[5] For the purposes of our Cleaning Lemma, we may instead use a PQP oracle, where recall PQP is BQP except in the YES case, the verifier accepts with probability $> 1/2$, and in the NO case accepts with probability $\leq 1/2$. Note that in contrast to Precise-BQP, PQP is defined without completeness/soundness parameters (Watrous 2009a); that one may impose an inverse exponential promise gap on PQP is a non-trivial consequence of the fact that one may choose an "appropriately nice" gate set for PQP to ensure acceptance probabilities are rational. For this reason, and since whether Precise-BQP equals PQP or not depends strongly on the choice of completeness/soundness parameters, we have opted to treat Precise-BQP as a generally distinct entity from PQP; see Section 3.1 for details.

expression applies the existential ($\exists$) and universal ($\forall$) operators to a complexity class $\mathcal{C}$. Informally, $\exists \cdot \mathcal{C}$ is the class of languages such that an input x is in the language if and only if there is a polynomial-size witness y such that $\langle x, y \rangle$ is in a language in $\mathcal{C}$. Correspondingly, the $\forall \cdot \mathcal{C}$ class is defined when for every witness y, $\langle x, y \rangle$ is in some language in $\mathcal{C}$. (See Definition 3.11 for formal definitions of $\exists$ and $\forall$.)

Notice that although we use a Precise-BQP oracle above, the Cleaning Lemma shows containment using a PP oracle. This is because, as shown in Lemma 3.3 and Corollary 3.7, Precise-BQP $\subseteq$ PP. One may ask whether our proof technique also works with an oracle *weaker* than PP. We show in Theorem 3.27 that this is unlikely, since the problem of detecting proofs in promise gaps of quantum verifiers is PP-complete.

Finally, an immediate corollary of Theorem 1.4 and the fact that QMA(2) $\subseteq$ QPH is:

Corollary 1.5. *If* QCPH = QPH, *then* QMA(2) $\subseteq \mathrm{P}^{\mathrm{PP}^{\mathrm{PP}}}$.

In other words, if alternating quantifiers are so powerful so as to make classical and quantum proofs equivalent in power, then QMA(2) is contained in CH(and thus in PSPACE). For comparison, QMA $\subseteq \mathrm{P}^{\mathrm{QMA[log]}} \subseteq$ PP (Gharibian & Yirka 2019; Kitaev & Watrous 2000; Marriott & Watrous 2005; Vyalyi 2003).

2. QPH versus NEXP. We next turn to the study of *quantum* proofs, i.e., QPH. As mentioned above, the best known upper bound on QMA(2) is NEXP–a non-deterministic verifier can simply guess an exponential-size description of the proof. When alternating quantifiers are present, however, this strategy seemingly no longer works. In other words, it is not even clear that QPH $\subseteq$ NEXP! This is in stark contrast to the explicit $\mathrm{P}^{\#\mathrm{P}}$ upper bound for PH (Toda 1991). In this part, our goal is to use semidefinite programming to give bounds on some levels of QPH. As we will see, this will yield the existence of a complexity class lying “between” QMA(2) and NEXP.

Theorem 1.6 (Informal statement). *It holds that* $\mathrm{Q}\Sigma_2 \subseteq \mathrm{EXP}$ *and* $\mathrm{Q}\Pi_2 \subseteq \mathrm{EXP}$*, even when the completeness-soundness gap is inverse doubly-exponentially small.*

(See Corollary 1.12 for a tighter bound given a larger completeness-soundness gap.) The proof idea is to map alternating quantifiers to an optimization problem with alternating minimizations and maximizations. Namely, to decide if $x \in A_{\text{yes}}$ or $x \in A_{\text{no}}$ for a $\mathrm{Q}\Sigma_i$ promise problem $A = (A_{\text{yes}}, A_{\text{no}})$, where i is even, we can solve for α defined as the optimal value of the optimization problem:

$$\alpha := \max_{\rho_1} \min_{\rho_2} \max_{\rho_3} \cdots \min_{\rho_i} \langle C, \rho_1 \otimes \rho_2 \otimes \cdots \otimes \rho_i \rangle \tag{1.7}$$

where C is the POVM operator[6] corresponding to the ACCEPT state of the verifier. (Note that each optimization attains an optimal solution by a simple compactness/continuity argument, hence the use of "max" instead of "sup" and "min" instead of "inf.") This is a non-convex problem, and as such is (likely) hard to solve in general. Our approach is to cast the case of $i = 2$ as a semidefinite program (SDP), allowing us to efficiently approximate α.

The next natural question is whether a similar SDP reformulation might be used to show whether $\mathrm{Q}\Sigma_3$ or $\mathrm{Q}\Pi_3$ is also contained in EXP. Unfortunately, this is likely to be difficult—indeed, if there exists a "nice" SDP for the optimal success probability of $\mathrm{Q}\Sigma_3$ protocols, then it would imply $\mathrm{QMA}(2) \subseteq \mathrm{EXP}$, resolving the longstanding open problem of separating QMA(2) from NEXP (recall $\mathrm{QMA}(2) \subseteq \mathrm{Q}\Sigma_3$). Likewise, a "nice" SDP for $\mathrm{Q}\Pi_3$ would place $\mathrm{coQMA}(2) \subseteq \mathrm{EXP}$.

To overcome this, we resort to non-determinism by stepping up to NEXP. Namely, one can non-deterministically guess the first proof of a $\mathrm{Q}\Sigma_3$ protocol, then approximately solve the SDP for the resulting $\mathrm{Q}\Pi_2$-flavored computation. Hence, we have:

Theorem 1.8 (Informal statement). *It holds true* $\mathrm{QMA}(2) \subseteq \mathrm{Q}\Sigma_3 \subseteq \mathrm{NEXP}$ *and* $\mathrm{coQMA}(2) \subseteq \mathrm{Q}\Pi_3 \subseteq \mathrm{coNEXP}$*. All contain-*

[6] A POVM is a set of Hermitian positive semidefinite operators that sum to the identity. In this case, the POVM has two operators—corresponding to the ACCEPT and REJECT states of the verifier.

ments hold with equality in the inverse exponentially and doubly-exponentially small completeness-soundness gap setting as then QMA(2) = NEXP (Pereszlényi 2012).

Three remarks are in order. First, note that our results are independent of the gate set. Second, in principle, it remains plausible that the *fourth* level of QPH already exceeds NEXP in power. Finally, we have the following implication for QMA(2). Assuming PH does not collapse, alternating quantifiers strictly add power to NP proof systems. If alternating quantifiers similarly add power in the quantum setting, then it would separate QMA(2) from NEXP via the following immediate corollary.

Corollary 1.9. *If* $\mathrm{QMA}(2) \subsetneq \mathrm{Q}\Sigma_3$*, i.e., if the second universally quantified proof of* $\mathrm{Q}\Sigma_3$ *adds proving power, then it follows that* $\mathrm{QMA}(2) \subsetneq \mathrm{NEXP}$*. Similarly, if* $\mathrm{coQMA}(2) \subsetneq \mathrm{Q}\Pi_3$*, then* $\mathrm{coQMA}(2) \subsetneq \mathrm{coNEXP}$.

Note added: Since the original release of this article, new observations relevant to the discussion above have been made. Since these observations are closely related to the open questions of this article, they have been placed in Section 1.3 (Recent observations and open questions).

3. A quantum generalization of the Karp–Lipton Theorem. Finally, our last result studies a topic which is unrelated to QMA(2)—the well-known Karp–Lipton theorem (1980). The latter shows that if NP-complete problems can be solved by polynomial-size non-uniform Boolean circuit families, then $\Sigma_2 = \Pi_2$ (formal definitions in Section 2), which in turn implies that PH collapses to its second level. Here, a "non-uniform" circuit family implies that for each input length, there exists a circuit which decides the problem, with no restriction on the computational difficulty of producing the circuit. The class of decision problems solved by such polynomial-size circuit families is $\mathrm{P}_{/\mathrm{poly}}$. An equivalent description for $\mathrm{P}_{/\mathrm{poly}}$ is the class languages decidable by polynomial-time Turing machines that receive, in addition to the input instance, a polynomial-size "advice string" y such that (1) y depends only on the input size n, and (2) there is no computational restriction on y.

THEOREM 1.10 (Karp & Lipton 1980). *If* $\mathrm{NP} \subseteq \mathrm{P}_{/\mathrm{poly}}$ *then* $\Pi_2 = \Sigma_2$.

Denote the bounded-error analogue of $\mathrm{P}_{/\mathrm{poly}}$ with polynomial-size non-uniform quantum circuits as $\mathrm{BQP}_{/\mathrm{mpoly}}$. In this work, we ask: Does $\mathrm{QCMA} \subseteq \mathrm{BQP}_{/\mathrm{mpoly}}$ imply $\mathrm{QC}\Pi_2 = \mathrm{QC}\Sigma_2$? Unfortunately, generalizing the proof of the Karp–Lipton theorem is problematic for the same "$\exists \cdot$ BPP versus MA phenomenon" encountered in extending Toda's result. Namely, the proof of Karp–Lipton proceeds by fixing the outer, universally quantified, proof of a Π_2 machine, and applying the $\mathrm{NP} \subseteq \mathrm{P}_{/\mathrm{poly}}$ hypothesis to the resulting NP computation. However, for $\mathrm{QC}\Pi_2$, it is not clear that fixing the outer, universally quantified, proof yields a QCMA computation; thus, it is not obvious how to use the hypothesis $\mathrm{QCMA} \subseteq \mathrm{BQP}_{/\mathrm{mpoly}}$.

To sidestep this, our approach is to strengthen the hypothesis. Specifically, using the results of Jordan *et al.* (2012) on perfect completeness for QCMA, fixing the outer proof of a $\mathrm{QC}\Pi_2$ computation can be seen to yield a Precise-QCMA "decision problem," where by "decision problem," we mean no proofs for the Precise-QCMA verifier are accepted within the promise gap. Here, Precise-QCMA is QCMA with inverse exponentially small promise gap. We hence obtain the following.

THEOREM 1.11 (A quantum-classical Karp–Lipton theorem). *If* $\text{Precise-QCMA} \subseteq \mathrm{BQP}_{/\mathrm{mpoly}}$, *then* $\mathrm{QC}\Pi_2 = \mathrm{QC}\Sigma_2$.

To give this result context, we also show that $\text{Precise-QCMA} = \mathrm{NP}^{\mathrm{PP}}$ (Lemma 5.11). However, whether $\mathrm{QC}\Pi_2 = \mathrm{QC}\Sigma_2$ collapses QCPH remains open due to the same "$\exists \cdot$ BPP versus MA phenomenon."

1.2. Related work. As far as we are aware, Yamakami (2002) was the first to consider a quantum version of PH. His version differs from our setting in that it considers quantum Turing machines (we use quantum circuits) and quantum inputs (we use classical inputs, like QMA). The next work, by Gharibian & Kempe (2012), introduced and studied cq-Σ_2, defined as our $\mathrm{QC}\Sigma_2$ except with a quantum universally quantified proof. Gharibian & Kempe showed completeness and hardness of approximation results for cq-Σ_2 for (roughly) the following problem: What is the smallest number of terms required in a given local Hamiltonian for it to have a frustrated ground space? More recently, Lockhart & González-Guillén (2017) considered a hierarchy (denoted QCPH′ here) which *a priori* appears identical to our QCPH, but is apparently not so due to the "$\exists \cdot$ BPP versus MA phenomenon," which we now discuss briefly (see also Remark 3.12).

In this work, the "$\exists \cdot$ BPP versus MA phenomenon," refers to the following discrepancy: Unlike with MA, *all* proofs in an $\exists \cdot$ BPP system must be accepted with probability at least $2/3$ or at most $1/3$ (i.e., no proof is accepted with probability in the gap $(1/3, 2/3)$). The quantum analogue of this phenomenon yields the open question: Is $\exists \cdot$ BQP (which equals $\mathrm{NP}^{\mathrm{BQP}}$) equal to QCMA? For this reason, it is not clear whether QCPH equals QCPH′, where QCPH′ is defined recursively as $\mathrm{QC}\Sigma'_1 = \exists \cdot \mathrm{BQP}$, $\mathrm{QC}\Pi'_1 = \forall \cdot \mathrm{BQP}$, and

$$\forall i \geq 1, \mathrm{QC}\Sigma'_i = \exists \cdot \mathrm{QC}\Pi'_{i-1}; \qquad \mathrm{QC}\Pi'_i = \forall \cdot \mathrm{QC}\Sigma'_{i-1}.$$

Thus, in our work $\mathrm{QC}\Sigma_1 = \mathrm{QCMA}$, but in Lockhart & González-Guillén, $\mathrm{QC}\Sigma'_1 = \exists \cdot \mathrm{BQP}$.[7] The advantage of the latter definition is that one avoids the recursion problems discussed earlier—e.g., fixing the first existential proof in $\mathrm{QC}\Sigma'_2$ *does* reduce the problem to a $\mathrm{QC}\Pi'_1$ computation, unlike the case with $\mathrm{QC}\Sigma_2$. Hence, recursive arguments from the context of PH can be extended to show that, for instance, QCPH′ collapses to $\mathrm{QC}\Sigma'_2$ when $\mathrm{QC}\Sigma'_2 = \mathrm{QC}\Pi'_2$. On the other hand, the advantage of our definition of QCPH is that it generalizes the natural quantum complexity class QCMA.

Let us also remark on Toda's theorem in the context of QCPH′ (for clarity, Toda's theorem is not studied in Lockhart & González-Guillén). The recursive definition of QCPH′ allows one to obtain

[7] While the "$\exists \cdot$ BPP versus MA phenomenon" is not strictly quantum, there is no issue in defining a classical probabilistic PH due to the ability of the standard PH to simulate randomness. Specifically, both $\exists \cdot$ BPP and MA are in PH, and this generalizes to show any straightforward definitions of a bounded-error PH are equivalent to deterministic PH.

Toda's P^{PP} upper bound for QCPH' with a simple argument:

$$\begin{aligned} & \forall i,\ \mathrm{QC}\Sigma_i' = \mathrm{NP}^{\mathrm{NP}^{\cdot^{\cdot^{\cdot^{\mathrm{BQP}}}}}} && = \Sigma_i^{\mathrm{BQP}} \\ \implies & \forall i,\ \mathrm{QC}\Sigma_i' \subseteq \left(\mathrm{P}^{\mathrm{PP}}\right)^{\mathrm{BQP}} && = \mathrm{P}^{\mathrm{PP}}, \end{aligned}$$

where the first equality holds due to the recursive definition of $\mathrm{QC}\Sigma_i'$ (but is not known to hold for our $\mathrm{QC}\Sigma_i$), the implication arises by relativizing Toda's theorem, and the last equality holds as BQP is low for PP (Fortnow & Rogers 1999). In contrast, our Theorem 1.4 yields $\mathrm{QCPH} \subseteq \mathrm{P}^{\mathrm{PP}^{\mathrm{PP}}}$, raising the question: is $\mathrm{QCPH}' = \mathrm{QCPH}$? A positive answer may help shed light on whether $\exists \cdot \mathrm{BQP}$ equals QCMA; we leave this for future work.

Finally, a quantum version of the Karp–Lipton theorem was covered by Aaronson & Drucker (2014) and further improved by Aaronson, Cojocaru, Gheorghiu & Kashefi (2019), where the consequences of NP-complete problems being solved by small quantum circuits with polynomial-size quantum advice were considered. Their results differ from ours in that different hierarchies are studied, and in their use of quantum advice as opposed to our use of classical advice. Other Karp–Lipton style results for PH involving classes beyond NP show a collapse of PH to MA (usually) if either PP (Lund *et al.* 1992; Vinodchandran 2005), $\mathrm{P}^{\#\mathrm{P}}$ or PSPACE (Karp & Lipton 1980) has $\mathrm{P}_{/\mathrm{poly}}$ circuits.

1.3. Recent observations and open questions.

1.3.1. Recent observations. Upon release of the current article, Sanketh Menda, Harumichi Nishimura, and John Watrous (whom we thank) made the observation that $\mathrm{Q}\Sigma_2 = \mathrm{QRG}(1)$, where QRG(1) captures one-round zero-sum quantum games (Jain & Watrous 2009). Briefly, this equivalence follows immediately since QRG(1) from Jain & Watrous can be defined as in Equation (1.7), but restricted to just the first two proofs, ρ_1 and ρ_2.[8] This insight has led to some remarkable immediate corollaries regarding $\mathrm{Q}\Sigma_2$ and $\mathrm{Q}\Pi_2$, which we now discuss.

[8] The use of *mixed* state proofs in our definition of $\mathrm{Q}\Sigma_2$ and in Equation (1.7) is crucial for this equivalence.

The starting point for the discussion is that, as done for QRG(1) in Jain & Watrous, one can apply an extension of von Neumann's min-max theorem (1928) to conclude in Equation (1.7) that

$$\max_{\rho_1}\min_{\rho_2}\langle C, \rho_1 \otimes \rho_2\rangle = \min_{\rho_2}\max_{\rho_1}\langle C, \rho_1 \otimes \rho_2\rangle.$$

In other words, $Q\Sigma_2 = Q\Pi_2$. In addition, Jain & Watrous shows $\mathrm{QRG}(1) \subseteq \mathrm{PSPACE}$. We thus immediately have the following.

Corollary 1.12. $Q\Sigma_2 = Q\Pi_2 = \mathrm{QRG}(1) \subseteq \mathrm{PSPACE}$.

1.3.2. Relation to current work. For the standard completeness-soundness gap regime ($c - s \in \Omega(1/\mathrm{poly}(n))$), Corollary 1.12 improves upon our result of Theorem 1.6 (which recall showed $Q\Sigma_2$, $Q\Pi_2 \subseteq \mathrm{EXP}$). However, Theorem 1.6 and its proof, are still useful for the results in this paper: First, Theorem 1.6 works in the very small completeness-soundness gap regime. Second, the proof technique of Theorem 1.6 allows us to prove Theorem 1.8 (e.g., $Q\Sigma_3 \subseteq \mathrm{NEXP}$), which also holds in the very small completeness-soundness gap regime.

1.3.3. Further important implications of Corollary 1.12.

1. (Showing a "collapse theorem" for QPH will be "hard") One of the most frequently used results about PH is that if $\Sigma_2 = \Pi_2$, then PH collapses to Σ_2. Does a quantum analogue of this statement hold for QPH? Corollary 1.12 yields the following.

   Corollary 1.13. *If* $Q\Sigma_2 = Q\Pi_2$ *implies* $\mathrm{QPH} = Q\Sigma_2$, *then* $\mathrm{QMA}(2) \subseteq \mathrm{PSPACE}$.

   This follows immediately since recall $\mathrm{QMA}(2) \subseteq Q\Sigma_3 \subseteq \mathrm{QPH}$. Thus, proving such a "collapse theorem" for QPH would require a breakthrough regarding the complexity characterization of QMA(2), which is believed to be challenging.

2. (Separation between the second levels of PH and QPH) Corollary 1.12 also yields a separation of PH and QPH in the following sense, assuming the standard conjecture that PH is infinite.

Corollary 1.14. *If* $\mathrm{Q}\Sigma_2 = \Sigma_2$ *(or, equivalently,* $\mathrm{Q}\Pi_2 = \Pi_2$*), then* PH *collapses to* Σ_2*.*

This follows immediately since, as mentioned above, if $\Sigma_2 = \Pi_2$, then PH collapses to Σ_2. We are also using the fact that if A = B for language/promise classes A and B, then co-A = co-B. Thus, it is highly likely that $\mathrm{Q}\Sigma_2 \neq \Sigma_2$ and $\mathrm{Q}\Pi_2 \neq \Pi_2$

1.3.4. Open questions. As far as general upper bounds on QPH go, the currently best upper bound remains the naive one: The exponential-time analogue of PH, by which we mean constant-height towers of form $\mathrm{NEXP}^{\mathrm{NEXP}^{\mathrm{NEXP}^{\cdots}}}$ (i.e., use each copy of NEXP to "guess" the next exponential size quantum proof, roughly speaking, just as in the proof that PH equals constant-height towers of NP oracles). An open question is to find a better upper bound on QPH; we believe the naive bound to be loose.

One can also ask about the relationship between our QCPH and QPH classes and constant-height towers of the form $\mathrm{QCMA}^{\mathrm{QCMA}^{\cdots}}$ (a "QCMA-hierarchy") and $\mathrm{QMA}^{\mathrm{QMA}^{\cdots}}$ (a "QMA-hierarchy"), respectively. In this work, we have not studied the QCMA- and QMA-hierarchies, as they involve *quantum* machines making oracle queries, and this in itself would presumably need to be correctly defined. Recently, Vinkhuijzen (2018) defined a QMA-hierarchy, denoted BQPH, in which the QMA machines make queries in superposition. Vinkhuijzen proved BQPH is in the counting hierarchy (CH). However, it remains unclear how BQPH compares to QCPH and QPH. An alternative definition might use in-place queries. We believe this is an interesting avenue for future work. What we do observe here is that, if QPH were equal to a QMA-hierarchy (such as in Vinkhuijzen), then it would put QMA(2) in CH (and hence in PSPACE)[9], which would again require a breakthrough in our understanding of unentangled quantum proofs (i.e., QMA(2)).

Finally, determining where in the complexity zoo QMA(2) belongs remains an important open question. Assuming alternating

[9] Here we are using QMA ⊆ PP (Gharibian & Yirka 2019; Kitaev & Watrous 2000; Marriott & Watrous 2005; Vyalyi 2003).

quantifiers *do* add proving power to quantum proofs (the analogous assumption for classical proofs is widely believed), our work shows QMA(2) is strictly contained in NEXP. Can this statement be strengthened?

Organization: We begin in Section 2 by formally introducing relevant complexity classes. In Section 3, we show a quantum-classical analogue of Toda's theorem. Section 4 gives upper bounds on levels of QPH, and Section 5 shows a Karp–Lipton-type theorem.

2. Definitions, preliminaries, and basic properties

We begin by recalling the definition of uniformly-generated families of quantum circuits.

Definition 2.1. *(Polynomial-time uniform family of quantum circuits). A family of quantum circuits $\{V_n\}_{n\in\mathbb{N}}$ is said to be uniformly generated in polynomial time if there exists a polynomially bounded function $t:\mathbb{N}\rightarrow\mathbb{N}$ and a deterministic Turing machine M acting as follows. For every n-bit input x, $M(1^n)$ outputs in time $t(n)$ a description of a quantum circuit V_n (consisting of 1- and 2-qubit gates) that takes $|x\rangle$ as input and the all-zeros state as ancilla and outputs a single qubit. We say $V_n(x)$* accepts *if measuring its output qubit in the computational basis yields 1. When the distinction is clear from context, we may refer to $V_n(x)$ by just V_n, implicitly fixing the input x into V_n.*

Throughout this paper, we study *promise problems.* A promise problem is a pair $A=(A_{\text{yes}},A_{\text{no}})$ such that $A_{\text{yes}},A_{\text{no}}\subseteq\{0,1\}^*$, $A_{\text{yes}}\cup A_{\text{no}}\subset\{0,1\}^*$ and $A_{\text{yes}}\cap A_{\text{no}}=\emptyset$. We now formally define each level of our quantum-classical polynomial hierarchy below.

Definition 2.2 ($\mathrm{QC}\Sigma_i$). *Let $A=(A_{\text{yes}},A_{\text{no}})$ be a promise problem. We say that A is in $\mathrm{QC}\Sigma_i(c,s)$ for polynomial-time computable functions $c,s:\mathbb{N}\rightarrow[0,1]$ if there exists a polynomially bounded function $p:\mathbb{N}\rightarrow\mathbb{N}$ and a polynomial-time uniform family* *of quantum circuits* $\{V_n\}_{n\in\mathbb{N}}$ *such that for every* n*-bit input* x*,* V_n *takes in* x *and classical proofs* $y_1 \in \{0,1\}^{p(n)}, \ldots, y_i \in \{0,1\}^{p(n)}$ *and outputs a single qubit, such that:*

- *Completeness:*

$$x \in A_{\text{yes}} \Rightarrow \exists y_1 \forall y_2 \ldots Q_i y_i \text{ such that } \Pr\text{ob}[V_n \text{ accepts } (x, y_1, \ldots, y_i)] \geq c.$$

- *Soundness:*

$$x \in A_{\text{no}} \Rightarrow \forall y_1 \exists y_2 \ldots \overline{Q}_i y_i \text{ such that } \text{Prob}[V_n \text{ accepts } (x, y_1, \ldots, y_i)] \leq s.$$

Here, Q_i *equals* $\exists$ *when* m *is odd and equals* $\forall$ *otherwise and* $\overline{Q}_i$ *is the complementary quantifier to* Q_i*.*

$$\text{(2.3)} \qquad \textit{Define } \mathrm{QC\Sigma}_i := \bigcup_{c-s\in\Omega(1/\mathrm{poly}(n))} \mathrm{QC\Sigma}_i(c,s).$$

Note that the first level of this hierarchy corresponds to QCMA. The complement of the i-th level of the hierarchy, $\mathrm{QC\Sigma}_i$, is the class $\mathrm{QC\Pi}_i$ defined below.

Definition 2.4 ($\mathrm{QC\Pi}_i$). *Let* $A = (A_{\text{yes}}, A_{\text{no}})$ *be a promise problem. We say that* $A \in \mathrm{QC\Pi}_i(c,s)$ *for polynomial-time computable functions* $c, s : \mathbb{N} \to [0,1]$ *if there exists a polynomially bounded function* $p : \mathbb{N} \to \mathbb{N}$ *and a polynomial-time uniform family of quantum circuits* $\{V_n\}_{n\in\mathbb{N}}$ *such that for every* n*-bit input* x*,* V_n *takes in* x *and classical proofs* $y_1 \in \{0,1\}^{p(n)}, \ldots, y_i \in \{0,1\}^{p(n)}$ *and outputs a single qubit, such that:*

- *Completeness:*

$$x \in A_{\text{yes}} \Rightarrow \forall y_1 \exists y_2 \ldots Q_i y_i \text{ such that } \text{Prob}[V_n \text{ accepts } (x, y_1, \ldots, y_i)] \geq c.$$

- *Soundness:*

$$x \in A_{\text{no}} \Rightarrow \exists y_1 \forall y_2 \ldots \overline{Q}_i y_i \text{ such that } \text{Prob}[V_n \text{ accepts } (x, y_1, \ldots, y_i)] \leq s.$$

Here, Q_i equals $\forall$ when m is odd and equals $\exists$ otherwise and $\overline{Q}_i$ is the complementary quantifier to Q_i.

$$\text{(2.5)} \qquad \textit{Define } \mathrm{QC\Pi}_i := \bigcup_{c-s\in\Omega(1/\mathrm{poly}(n))} \mathrm{QC\Pi}_i(c,s).$$

Now the corresponding quantum-classical polynomial hierarchy is defined as below.

Definition 2.6 (Quantum-Classical Polynomial Hierarchy).

$$\mathrm{QCPH} = \bigcup_{m\in\mathbb{N}} \mathrm{QC\Sigma}_i = \bigcup_{m\in\mathbb{N}} \mathrm{QC\Pi}_i.$$

A few remarks are in order. First, by encoding a polynomial-time predicate into a quantum verification circuit, one can see that (where Σ_i and Π_i refer to the i-th level of the corresponding classical polynomial hierarchy)

$$\forall i,\ \Sigma_i \subseteq \mathrm{QC\Sigma}_i, \quad \Pi_i \subseteq \mathrm{QC\Pi}_i \quad \text{and} \quad \mathrm{PH} \subseteq \mathrm{QCPH}.$$

Second, a natural question is to what extent the completeness and soundness parameters of $\mathrm{QC\Sigma}_i$ and $\mathrm{QC\Pi}_i$ can be improved. Toward achieving one-sided error, we apply known techniques to prove that “every other level” (see Theorem 2.10 for a formal statement) has *perfect completeness* (i.e., we can improve the completeness parameter to $c = 1$), in addition to every level having inverse exponentially small soundness. This is shown using techniques from the proof of the following theorem.

Theorem 2.7 (Jordan *et al.* 2012). QCMA *has perfect completeness, i.e.,*

$$\text{(2.8)} \qquad \mathrm{QCMA} = \mathrm{QCMA}(1, 1 - 1/\mathrm{poly}(n)).$$

The proof of the above result starts by choosing a suitable gate-set for the QCMA verifier, i.e., Hadamard, Toffoli and CNOT gates (Aharonov 2003; Shi 2003). This ensures that the acceptance probability for any proof y can be expressed as $k/2^{\ell(|x|)}$ for an integer $k \in \{0,\ldots,2^{\ell(|x|)}\}$ and a polynomially bounded integer function

$\ell(|x|)$. The verifier then asks the prover to send k (expressed as a polynomial-size bit string) along with the classical proof. When k is above a certain threshold, the verifier chooses one of two tests with equal probability: (a) run the original verification circuit or (b) trivially accept with probability $> k/2^{\ell(|x|)}$. This allows for the completeness to be reduced to exactly 1/2 while the soundness is strictly bounded below 1/2. Then, by using the quantum rewinding technique (Watrous 2009c), c can be boosted to exactly 1. The ideas in this proof have been adapted to several similar scenarios (see, e.g., Grilo *et al.* 2016; Kobayashi *et al.* 2015).

For easy reference later, we state a generalized version of the technique used by Jordan *et al.* in the following observation.

Observation 2.9 (Rational acceptance probabilities). *By fixing an appropriate universal gate set (e.g., Hadamard and Toffoli Aharonov 2003; Jordan* et al.*), we assume henceforth, without loss of generality, that the acceptance probabilities of all quantum circuits V taking only classical (standard basis) states as input are rational numbers that can each be represented using at most* poly(n) *bits.*

We state our result below.

Theorem 2.10. *For polynomially bounded functions $r, q : \mathbb{N} \rightarrow \mathbb{N}$ and polynomial-time computable functions $c, s : \mathbb{N} \rightarrow [0, 1]$ such that for any n-bit input $c(n) - s(n) \geq 1/q(n)$, we have:*

$$\begin{array}{rrcl} \text{For } i \text{ even:} & \mathrm{QC\Sigma}_i(c,s) & = & \mathrm{QC\Sigma}_i(1-2^{-r}, 2^{-r}), \\ & \mathrm{QC\Pi}_i(c,s) & = & \mathrm{QC\Pi}_i(1, 2^{-r}), \\ \text{for } i \text{ odd:} & \mathrm{QC\Sigma}_i(c,s) & = & \mathrm{QC\Sigma}_i(1, 2^{-r}), \\ & \mathrm{QC\Pi}_i(c,s) & = & \mathrm{QC\Pi}_i(1-2^{-r}, 2^{-r}). \end{array}$$

Proof (Proof Sketch). To achieve perfect completeness (i.e., $c = 1$), the idea is to append to the register of the last proof (which must be an existential quantifier for this to work) a classical register containing the acceptance probability of the verification circuit C. Specifically, for level i, for any input x and any set of $i-1$ proofs $y_1, \ldots, y_{i-1}$, the final (existential) proof y_i is augmented with k, such that $\Pr[C(x, y_1, \ldots, y_i) = 1] = k/2^{\ell(|x|)}$ (that this probability is rational is due solely to the use of an appropriate universal gate set, as done for Theorem 2.7, and is independent of how each y_i for $i \in \{1, \ldots, i-1\}$ is quantified). Then, the proof of Theorem 2.7 in Jordan *et al.* proves the result. The error reduction follows from standard arguments. □

Notice that by explicitly emulating the technique from Jordan *et al.* we are using it as a *white-box* and not a black box reduction. Hence, the issues discussed in Section 1 that arise from "fixing proofs" does not apply here. We leave as an open problem the question of obtaining perfect completeness for the remaining levels of the hierarchy. This seems like a considerably harder problem, with current proof techniques requiring the last quantifier to be existential.

Now, we move on to defining the fully quantum hierarchy.

Definition 2.11 ($\mathrm{Q}\Sigma_i$). *A promise problem $A = (A_{\mathrm{yes}}, A_{\mathrm{no}})$ is in $\mathrm{Q}\Sigma_i(c, s)$ for polynomial-time computable functions $c, s : \mathbb{N} \to [0, 1]$ if there exists a polynomially bounded function $p : \mathbb{N} \to \mathbb{N}$ and a polynomial-time uniform family of quantum circuits $\{V_n\}_{n\in\mathbb{N}}$ such that for every n-bit input x, V_n takes x as input and $p(n)$-qubit density operators $\rho_1, \ldots, \rho_i$ as quantum proofs and outputs a single qubit, then:*

- ○ *Completeness: If $x \in A_{\mathrm{yes}}$, then $\exists\rho_1\forall\rho_2 \ldots Q_i\rho_i$ such that V_n accepts $(x, \rho_1 \otimes \rho_2 \otimes \cdots \otimes \rho_i)$ with probability $\geq c$.*
- ○ *Soundness: If $x \in A_{\mathrm{no}}$, then $\forall\rho_1\exists\rho_2 \ldots \overline{Q}_i\rho_i$ such that V_n accepts $(x, \rho_1 \otimes \rho_2 \otimes \cdots \otimes \rho_i)$ with probability $\leq s$.*

Here, Q_i equals $\forall$ when m is even and equals $\exists$ otherwise, and $\overline{Q}_i$ is the complementary quantifier to Q_i.

$$\text{(2.12)} \qquad \textit{Define } \mathrm{Q}\Sigma_i = \bigcup_{c-s\in\Omega(1/\mathrm{poly}(n))} \mathrm{Q}\Sigma_i(c, s).$$

A few comments are in order: (1) In contrast to the standard quantum circuit model, here we allow *mixed* states as inputs to V_n; this can be formally modeled via the mixed state framework of Aharonov *et al.* (1998). (2) Clearly, $\mathrm{Q}\Sigma_1 = \mathrm{QMA}$. (3) We recover the definition of $\mathrm{QMA}(k)$ by ignoring the ρ_i proofs, for i even, in the definition of $\mathrm{Q}\Sigma_{2k}$.

Definition 2.13 ($\mathrm{Q\Pi}_i$). *A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in $\mathrm{Q\Pi}_i(c, s)$ for polynomial-time computable functions $c, s : \mathbb{N} \to [0, 1]$ if there exists a polynomially bounded function $p : \mathbb{N} \to \mathbb{N}$ and a polynomial-time uniform family of quantum circuits $\{V_n\}_{n\in\mathbb{N}}$ such that for every n-bit input x, V_n takes x as input and $p(n)$-qubit density operators $\rho_1, \dots, \rho_i$ as quantum proofs and outputs a single qubit, then:*

- *Completeness: If $x \in A_{\text{yes}}$, then $\forall\rho_1\exists\rho_2 \dots Q_i\rho_i$ such that V_n accepts $(x, \rho_1 \otimes \rho_2 \otimes \cdots \otimes \rho_i)$ with probability $\geq c$.*
- *Soundness: If $x \in A_{\text{no}}$, then $\exists\rho_1\forall\rho_2 \dots \overline{Q}_i\rho_i$ such that V_n accepts $(x, \rho_1 \otimes \rho_2 \otimes \cdots \otimes \rho_i)$ with probability $\leq s$.*

Here, Q_i equals $\exists$ when m is even and equals $\forall$ otherwise, and $\overline{Q}_i$ is the complementary quantifier to Q_i.

$$\textit{Define } \mathrm{Q\Pi}_i = \bigcup_{c-s\in\Omega(1/\mathrm{poly}(n))} \mathrm{Q\Pi}_i(c, s). \tag{2.14}$$

The fully quantum polynomial hierarchy can now be defined as follows.

Definition 2.15 (Quantum Polynomial Hierarchy).

$$\mathrm{QPH} = \bigcup_{m\in\mathbb{N}} \mathrm{Q\Sigma}_i = \bigcup_{m\in\mathbb{N}} \mathrm{Q\Pi}_i.$$

We recall the definition of PP as it is repeatedly used in various results throughout this work.

Definition 2.16 (PP). *A language $\mathcal{L}$ is in* PP *if there exists a probabilistic polynomial-time Turing machine M such that $x \in \mathcal{L} \iff \Pr[M(x) = 1] > 1/2$.*

Note that the above defines PP as a class of decision problems. Another option, used for example in Watrous (2009a), is to define PP as a class of promise problems. In fact, for PP, the two definitions are equivalent. The forward direction Decision-PP $\subseteq$ Promise-PP is trivial, and the reverse containment follows because all inputs are polynomial in length, so all probabilities are at least inverse-exponential[10], so in the promise problem definition, we may reduce any problem to one in which no inputs violate the promise, which reduces to Decision-PP. Therefore, we will refer to PP both in terms of decision and promise problems.

Relevant to arguments involving the equivalent oracular definition of classical PH, we generally adopt the conventional definition that a complexity class with oracles denoted $\left(\mathrm{A}^{\mathrm{B}}\right)^{\mathrm{C}}$ allows both A *and* B to make calls to C.

Finally, we note that for matrices A and B, $\|A\|_F := \sqrt{\mathrm{Tr}(AA^\dagger)}$ denotes the Frobenius norm and $\langle A, B\rangle := \mathrm{Tr}\left(AB^\dagger\right)$ is the standard inner product of A and B.

3. A quantum-classical analogue of Toda's theorem

In this section, we show an analogue of Toda's theorem to bound the power of QCPH (Theorem 1.4, Section 3.2), and give evidence that the bound of Theorem 1.4 is likely the best possible using our specific proof approach (Section 3.3, Theorem 3.27).

3.1. Precise-BQP. Our proof of a "quantum-classical Toda's theorem" requires us to define the Precise-BQP class, which we do now.

Definition 3.1 (Precise-BQP(c, s)). *A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is contained in* Precise-BQP(c, s) *for polynomial-time computable functions $c, s : \mathbb{N} \to [0, 1]$ if there exists a polynomially bounded function $p : \mathbb{N} \to \mathbb{N}$ such that $\forall \ell \in \mathbb{N},\ c(\ell) - s(\ell) \geq 2^{-p(\ell)}$, and a polynomial-time uniform family of quantum circuits $\{V_n\}_{n\in\mathbb{N}}$ whose input is x with the all-zeroes state as ancilla and output is a single qubit. Furthermore, for an n-bit input x:*

- *Completeness: If $x \in A_{\text{yes}}$, then V_n accepts with probability at least $c(n)$.*

[10] Specifically, in the proof verification definition of PP, the verifier's Turing machine has at most exponentially many paths. Thus, the difference in acceptance probabilities for a YES versus NO case when picking a random path/proof is at least inverse exponential.

- *Soundness: If $x \in A_{\text{no}}$, then V_n accepts with probability at most $s(n)$.*

$$\textit{Define } \text{Precise-BQP} = \bigcup_{c-s\in\Omega(1/\exp(n))} \text{Precise-BQP}(c, s). \tag{3.2}$$

In contrast, BQP is defined such that the completeness and soundness parameters are $2/3$ and $1/3$, respectively (alternatively, the gap is at least an inverse polynomial in n). The following lemmas help to characterize the complexity of Precise-BQP.

Lemma 3.3. *For any polynomial p, if $c - s \geq 1/2^{p(n)}$, then*

$$\text{Precise-BQP}(c, s) \subseteq \text{PP} \tag{3.4}$$

when c and s are computable in polynomial-time in the size of the input, n.

Proof (Proof sketch). Recall that the complexity class PQP is defined as PP except with a uniform quantum circuit family $\{ Q_n \}$ in place of a probabilistic Turing machine, i.e., for YES (NO) instances Q_n accepts with probability $> 1/2$ ($\leq 1/2$). Consider any Precise-BQP(c, s) circuit V_n as in Definition 3.1. Then, by flipping a coin with appropriately chosen bias $\gamma \in \mathbb{Q}$ and choosing to either accept/reject with probability γ and run Q_n with probability $1 - \gamma$, one may map c, s to polynomial-time computable functions c', s' such that

$$c' > 1/2, \; s' \leq 1/2, \; \text{ and } \; c' - s' \in \Theta(c - s) \tag{3.5}$$

(roughly, one loses about a factor of at most approximately $1/2$ in the gap). Thus,

$$\text{Precise-BQP}(c, s) \subseteq \text{PQP} = \text{PP}, \tag{3.6}$$

where the last equality is shown in Watrous (2009a). □

As an aside, we remark the following.

Corollary 3.7. *Let $\mathbb{P}$ denote the set of all polynomials $p : \mathbb{N} \to \mathbb{N}$. Then,*

$$\bigcup_{p\in\mathbb{P}} \text{Precise-BQP}\left(\frac{1}{2} + \frac{1}{2^{p(n)}}, \frac{1}{2}\right) = \text{PP}.$$

To prove Corollary 3.7, we need the classical counterpart of Precise-BQP, denoted Precise-BPP. Accordingly, we define Precise-BPP analogous to Definition 3.1 except by replacing the quantum circuit family $\{ V_n \}_{n\in\mathbb{N}}$ with a deterministic polynomial-time Turing machine which takes in polynomially many bits of randomness.

Proof. The direction $\subseteq$ is given by Lemma 3.3. For the reverse containment, note that

$$\text{PP} = \bigcup_{p\in\mathbb{P}} \text{Precise-BPP}\left(\frac{1}{2} + \frac{1}{2^{p(n)}}, \frac{1}{2}\right), \tag{3.8}$$

since PP can be defined as the set of decision problems of the form: Given as input a polynomial-time non-deterministic Turing machine N and string x, do more than half of $N(x)$'s computational paths accept? The claim now follows, since for all c, s as in Definition 3.1, clearly Precise-BPP$(c, s) \subseteq$ Precise-BQP(c, s). □

Note that this proof does not go through as is (assuming PP $\neq$ coNP) when we fix (say) completeness $c = 1$ and soundness $s = 1 - 2^{-p(n)}$, for some polynomial p. This is because

$$\bigcup_{p\in\mathbb{P}} \text{Precise-BPP}\left(1, 1 - \frac{1}{2^{p(n)}}\right) = \text{coNP}. \tag{3.9}$$

Similarly, setting $s = 0$ and $c = 2^{-p(n)}$ yields NP.

Finally, we define the promise problem QCIRCUIT(c, s), which is trivially Precise-BQP(c, s)-complete when $c - s$ is an inverse exponential.

Definition 3.10 (QCIRCUIT(c, s)). *Parameters $c, s : \mathbb{N} \to [0, 1]$ are polynomial-time computable functions such that $c > s$.*

- *(Input) A classical description of quantum circuit V_n (acting on n qubits, consisting of* $\mathrm{poly}(n)$ *1- and 2-qubit gates), taking in a fixed string x and the all-zeroes state, and outputting a single qubit.*

- *(Output) Decide if* $\Pr[V_n(x)$ *accepts*$] \geq c$ *or* $\leq s$*, assuming one of the two is the case.*

3.2. Bounding the power of QCPH. We begin by briefly outlining a proof of Toda's theorem (1991), that $\mathrm{PH} \subseteq \mathrm{P}^{\#\mathrm{P}}$. The proof can be split into two major parts: (1) show $\mathrm{PH} \subseteq \mathrm{BPP}^{\oplus\mathrm{P}}$, and (2) show $\mathrm{BPP}^{\oplus\mathrm{P}} \subseteq \mathrm{P}^{\#\mathrm{P}}$. The second step is accomplished using a clever counting scheme combined with lifting. The first step is more involved and can be broken down further: (a) use the Valiant–Vazirani theorem (1986) to filter the proof y_i so that at most one proof is accepted (for Σ_1, this gives a randomized reduction that $\mathrm{NP} \subseteq_R$ Unique-NP); (b) negate the proof y_{i-1}, causing the i-th and $(i-1)$-st quantifiers to match and collapse; and (c) recurse.

Classically, PH can be defined in terms of the existential ($\exists$) and universal ($\forall$) operators (distinct from quantifiers), while it is not clear that one can also define QCPH using these operators, they nevertheless prove useful in bounding the power of QCPH.

Definition 3.11. (Existential and universal operators: Allender & Wagner 1993; Wrathall 1976) *For $\mathcal{C}$ a class of languages, $\exists \cdot \mathcal{C}$ is defined as the set of languages L such that there is a polynomial p and set $A \in \mathcal{C}$ such that for input x,*

$$x \in L \Leftrightarrow [\exists y\, (|y| \leq p(|x|)) \text{ and } \langle x, y\rangle \in A]\,.$$

The set $\forall \cdot \mathcal{C}$ is defined similarly with $\exists$ replaced with $\forall$.

Remark 3.12 (Languages versus promise problems). *Directly extending Definition 3.11 to promise problems, gives rise to subtle issues. To demonstrate, recall that* $\exists \cdot \mathrm{P} = \mathrm{NP}$. *Then, let (L, A) for* $L \in \exists{\cdot}\mathrm{P} = \mathrm{NP}$ *and* $A \in \mathrm{P}$ *be as in Definition 3.11, such that T_A is a polynomial-time Turing machine deciding A. If $x \in L$, there exists* *a bounded length witness y^* such that T_A accepts $\langle x, y^* \rangle$ and, for all $y' \neq y^*$, T_A by definition either accepts or rejects $\langle x, y' \rangle$. Now consider instead $\exists \cdot$* BPP*, which* a priori *seems equal to Merlin-Arthur (*MA*). Applying the same definition of $\exists$, we should obtain a* BPP *machine T_A such that if $x \in L$, then for all $y' \neq y^*$, T_A* either accepts or rejects *$\langle x, y' \rangle$. But this means, by definition of* BPP*, that $\langle x, y' \rangle$ is either accepted or rejected with probability at least $2/3$, respectively. (Equivalently, for any fixed y, the machine $T_{A,y}$ must be a* BPP *machine.) Unfortunately, the definition of* MA makes no such promise—*any $y' \neq y^*$ can be accepted with arbitrary probability when x is a YES instance. Indeed, whether $\exists \cdot$* BPP = MA *remains an open question (*Fenner et al. *2003).*

The following lemma is the main contribution of this section. To set context, adapting the ideas from Toda's proof of PH $\subseteq$ P$^{\text{PP}}$ to QCPH is problematic for at least two reasons:

1. The "Quantum Valiant–Vazirani (QVV)" theorem for QCMA (and MA) (Aharonov *et al.* 2022) is not a many-one reduction, but a *Turing* reduction. Specifically, it produces a set of quantum circuits $\{ Q_i \}$ (with fixed input, accepting a corresponding proof), *at least one of which* is guaranteed to be a YES instance of some Unique-QCMA promise problem $\mathcal{P}$ if the input Γ to the reduction was a YES instance. Unfortunately, some of the Q_i may violate the promise gap of $\mathcal{P}$, which implies that when such Q_i are substituted into the Unique-QCMA oracle O, O returns an arbitrary answer. This does not pose a problem in Aharonov *et al.*, as one-sided error suffices for that reduction—so long as O accepts at least one Q_i, one safely concludes Γ was a YES instance. In the setting of Toda's theorem, however, the use of *alternating* quantifiers turns this one-sided error into two-sided error. This renders the output of O useless, as one can no longer determine whether Γ was a YES or NO instance.

2. Remark 3.12 says that it is not necessarily true that by fixing a proof y to an MA (resp. QCMA) machine, the resulting machine is a BPP (resp. BQP) machine. This prevents the direct extension of recursive arguments, say from Toda

(1991), to this regime (so, even if we could overcome the previous and fix the first proof, it remains unclear how to repeat the process).

To sidestep these issues, we adapt a high-level idea from Gharibian & Yirka (2019): With the help of an appropriate *oracle*, one can sometimes detect "invalid proofs" (i.e., proofs in promise gaps of bounded error verifiers) and "remove" them. Indeed, we show that using a PP oracle, one can eliminate the promise-gap of QCPH altogether, thus overcoming the limitations given above. This is accomplished by the following "Cleaning Lemma." We also show subsequently that it is highly unlikely for an oracle weaker than PP to suffice for our particular proof technique (see Remark 3.16 and Section 3.3).

Lemma 3.13 (Cleaning Lemma). *For all* $i \geq 0$,

$$\mathrm{QC}\Sigma_i \subseteq \exists \cdot \forall \cdot \cdots \cdot Q_i \cdot \mathrm{P}^{\text{Precise-BQP}} \subseteq \exists \cdot \forall \cdot \cdots \cdot Q_i \cdot \mathrm{P}^{\mathrm{PP}},$$

where $Q_i = \exists$ *(*$Q_i = \forall$*) if* i *is odd (even). An analogous statement holds for* $\mathrm{QC}\Pi_i$.

Proof. Let C be a $\mathrm{QC}\Sigma_i$ verification circuit for a promise problem $\mathcal{P}$; we implicitly fix the input x corresponding to $\mathcal{P}$ into the circuit. Let $C_{y_1^*,\ldots,y_i^*}$ denote the quantum circuit obtained from C by fixing values $y_1^*,\ldots,y_i^*$ of the i classical proofs. In general, nothing can be said about the acceptance probability $p_{y_1^*,\ldots,y_i^*}$ of $C_{y_1^*,\ldots,y_i^*}$, except that, by Observation 2.9, $p_{y_1^*,\ldots,y_i^*}$ is a rational number representable using $p(n)$ bits for some fixed polynomial p. Let S denote the set of all rational numbers in $[0,1]$ representable using $p(n)$ bits of precision and order the elements $s_1 < s_2 < \cdots$. (Note $|S| \in \Theta(2^{p(n)})$.) Then, for any pair $a = s_{j+1}, b = s_j$ in S, $C_{y_1^*,\ldots,y_i^*}$ is a valid $\mathrm{QCIRCUIT}(a,b)$ instance, in that $C_{y_1^*,\ldots,y_i^*}$ accepts with probability at least a or at most b for $a > b$. It follows that using binary search (by varying the values $a, b \in S$ with $a > b$) in conjunction with $\mathrm{poly}(n)$ calls to a $\bigcup_{a>b} \mathrm{QCIRCUIT}(a,b)$ oracle, we may exactly and deterministically compute $p_{y_1^*,\ldots,y_i^*}$. Moreover, since for all such a and b, $\mathrm{QCIRCUIT}(a,b) \in \text{Precise-BQP}(a,b)$, Lemma 3.3 implies the oracle calls can be simulated with a PP oracle. Denote the binary search subroutine using the PP oracle as B.

Using C and B, we now construct an oracle Turing machine C' as follows. Given any proofs $y_1^*, \dots, y_i^*$ as input (with x still implicitly fixed into the circuits), C' uses B to compute $p_{y_1^*,\dots,y_i^*}$ for $C_{y_1^*,\dots,y_i^*}$. If $p_{y_1^*,\dots,y_i^*} \geq c$, C' accepts with certainty, and if $p_{y_1^*,\dots,y_i^*} < s$, C' rejects with certainty. Suppose that the circuits C and C' return 1 when they accept and 0 when they reject. Two observations: (1) Since by construction, for any fixed $y_1^*, \dots, y_i^*$, B makes only makes "valid" QCIRCUIT(a, b) queries (i.e., satisfying the promise of QCIRCUIT(a, b)), C' is a P^{PP} machine (cf. Observation 3.17). (2) Since $C'_{y_1^*,\dots,y_i^*}$ accepts if $C_{y_1^*,\dots,y_i^*}$ accepts with probability at least c, and since $C'_{y_1^*,\dots,y_i^*}$ rejects if $C_{y_1^*,\dots,y_i^*}$ accepts with probability at most s, we conclude that

$$\exists y_1 \forall y_2 \cdots Q_i y_i : \text{Prob}[C(y_1, \dots, y_i) = 1] \geq c \quad \Leftrightarrow \exists y_1 \forall y_2 \cdots Q_i y_i \; C'(y_1, \dots, y_i) = 1 \tag{3.14}$$

$$\forall y_1 \exists y_2 \cdots \overline{Q}_i y_i : \text{Prob}[C(y_1, \dots, y_i) = 1] \leq s, \quad \Leftrightarrow \forall y_1 \exists y_2 \cdots \overline{Q}_i y_i \; C'(y_1, \dots, y_i) = 0. \tag{3.15}$$

Equations (3.14) and (3.15) imply that we can reduce $\mathcal{P}$ to a $\exists \cdot \forall \cdot \cdots \cdot Q_i \cdot \text{P}^{\text{PP}}$ computation. The proof for $\text{QC}\Pi_i$ is analogous. □

Remark 3.16 (Possibility of a stronger containment). *A key question is whether one may replace the* Precise-BQP *oracle in the proof of Lemma 3.13 with a weaker* BQP *oracle. For example, consider the following alternative definition for oracle Turing machine C': Given proofs $y_1^*, \dots, y_i^*$, C' plugs $C_{y_1^*,\dots,y_i^*}$ into a BQP oracle and returns the oracle's answers. It is easy to see that in this case, Equations* (3.14) *and* (3.15) *hold. However, C' is* not *necessarily a* P^{BQP} *machine, since for some settings of $y_1^*, \dots, y_i^*$, its input to the* BQP *oracle may violate the BQP promise, hence making the output of C' ill-defined. To further illustrate this subtle point, consider Observation 3.17. Moreover, in Section 3.3 we show that the task the* Precise-BQP *oracle is used for in Lemma 3.13 is in fact PP-complete; thus, it is highly unlikely that one can substitute a weaker oracle into the proof above.*

Observation 3.17. *(When a* P *machine querying a* BQP *oracle is not a* $\mathrm{P}^{\mathrm{BQP}}$ *machine). The proof of the Cleaning Lemma uses a* $\mathrm{P}^{\text{Precise-BQP}}$ *machine. Let us highlight a subtle reason why using a weaker* BQP *oracle instead might be difficult (indeed, in Section 3.3 we show that the task we use the* Precise-BQP *oracle for is* PP*-complete). Let M denote the trivially* BQP*-complete problem of determining whether a given polynomial-size quantum circuit Q and input x accepts with probability at least $2/3$, or accepts with probability at most $1/3$, with the promise that one of the two is the case. Now consider the following polynomial-time computation Γ which uses an oracle for M: Γ sends a circuit taking $|0\rangle$ as input and consisting of a Hadamard gate and measurement and then Γ outputs the oracle's answer. Γ is a* P *machine querying a* BQP *oracle, but does it hold that $\Gamma \in \mathrm{P}^{\mathrm{BQP}}$? No. The circuit which Γ sends to the oracle violates the promise of* BQP *and of M since applying Hadamard and measuring yields 0 or 1 with equal probability. Therefore, the oracle can answer arbitrarily (Goldreich 2006), and since Γ outputs the same answer as the oracle, the output of Γ is also arbitrary. Γ is not deterministic, so it is not in* $\mathrm{P}^{\mathrm{BQP}}$.

Using standard techniques, we next show the following.

Lemma 3.18. *For all $i \geq 0$, the following holds true:*

$$\begin{aligned} \exists \cdot \forall \cdot \dots \cdot Q_i \cdot \mathrm{P}^{\mathrm{PP}} &\subseteq \Sigma_i^{\mathrm{PP}}, \\ \forall \cdot \exists \cdot \dots \cdot Q_i \cdot \mathrm{P}^{\mathrm{PP}} &\subseteq \Pi_i^{\mathrm{PP}}, \end{aligned}$$

where $Q_i = \exists$ (resp. $Q_i = \forall$) when i is odd (resp. even) in the first containment and vice versa for the second containment.

Proof. We show the first statement with containment in Σ_i^{PP}; the second containment follows using an analogous proof. Let NP_i be defined recursively as $\mathrm{NP}_i := \mathrm{NP}^{\mathrm{NP}_{i-1}}$ with $\mathrm{NP}_1 := \mathrm{NP}$. We show that $\exists \cdot \forall \cdot \dots \cdot Q_i \cdot \mathrm{P}^{\mathrm{PP}} \subseteq \mathrm{NP}_i^{\mathrm{PP}}$, and then use the (relativizing) fact that $\mathrm{NP}_i = \Sigma_i$ using the oracular definition for Σ_i. Recall that MAJSAT is a PP-complete language, where, given a Boolean formula ϕ, one must decide if more than half of the possible assignments x satisfy $\phi(x) = 1$. For brevity, let A_i denote the following (trivially) $\exists \cdot \forall \cdot \cdots \cdot Q_i \cdot \mathrm{P}^{\mathrm{PP}}$-complete language (under polynomial-time many-one reductions): given as input a string x and a polynomial-time oracle Turing machine T with access to a MAJSAT oracle, decide whether

$$\begin{array}{ll} \exists y_1 \forall y_2 \cdots Q_i y_i & \text{such that } T \text{ accepts } \langle x, y_1, \ldots, y_i \rangle, \text{ or} \\ \forall y_1 \exists y_2 \cdots \overline{Q}_i y_i & \text{such that } T \text{ rejects } \langle x, y_1, \ldots, y_i \rangle. \end{array}$$

Let B_i denote the analogous trivially complete problem for $\forall \cdot \exists \cdot \cdots \cdot Q_i \cdot \mathrm{P}^{\mathrm{PP}}$. We proceed by induction. The base case $i = 0$ holds trivially since $\Sigma_0 = \mathrm{P}$ by definition. For the inductive step $i \geq 1$, let L be a language in $\exists \cdot \forall \cdot \cdots \cdot Q_i \cdot \mathrm{P}^{\mathrm{PP}}$. Then there exists a polynomial-time oracle Turing machine T with access to a MAJSAT oracle such that $x \in L$ if and only if $\exists y_1$ such that

$$\forall y_2 \exists y_3 \cdots Q_i y_i \quad \text{such that } T \text{ accepts } \langle x, y_1, y_2, \cdots, y_i \rangle.$$

By non-deterministically guessing y_1, it follows that $L \in \mathrm{NP}^{B_{i-1}} = \mathrm{NP}^{A_{i-1}}$. This equality holds since for all $i \geq 1$, $\mathrm{NP}^{B_i} = \mathrm{NP}^{A_i}$, as one can run the oracle for A_i instead of B_i and negate its answer. Since A_{i-1} is an oracle for $\exists \cdot \forall \cdot \cdots \cdot Q_{i-1} \cdot \mathrm{P}^{\mathrm{PP}}$, the induction hypothesis now implies that

$$\exists \cdot \forall \cdot \cdots \cdot Q_i \cdot \mathrm{P}^{\mathrm{PP}} \subseteq \mathrm{NP}^{\mathrm{NP}_{i-1}{}^{\mathrm{PP}}} = \mathrm{NP}_i^{\mathrm{PP}} = \Sigma_i^{\mathrm{PP}}. \qquad \square$$

We can now show the main theorem of this section.

Theorem 1.4. QCPH $\subseteq \mathrm{P}^{\mathrm{PP}^{\mathrm{PP}}}$.

Proof. The claim follows by combining the Cleaning Lemma (Lemma 3.13), Lemma 3.18, and Toda's theorem (PH $\subseteq \mathrm{P}^{\mathrm{PP}}$), whose proof relativizes (see, e.g., page 4 of Fortnow 1994). □

3.3. Detecting non-empty promise gaps is PP-complete.

The technique behind the Cleaning Lemma (Lemma 3.13) can essentially be viewed as using a PP oracle to determine whether a given quantum circuit accepts some input with probability within the promise gap (s, c), where $c - s$ is an inverse polynomial. One can ask whether this rather powerful PP oracle can be replaced with a weaker oracle (Remark 3.16)? We show that unless one deviates from our specific proof approach, the answer is negative. Specifically, we show that the problem of detecting non-empty promise gaps is PP-complete, even if the gap is *constant* in size. Let us begin by formalizing this problem.

Definition 3.19 (NON-EMPTY GAP(c, s)). *Let V_n, with fixed input x, be an input for* QCIRCUIT(c, s). *Then, output* YES *if* Prob$[V_n(x)$ *accepts* $] \in (s, c)$, *and* NO *otherwise.*

We now show that NON-EMPTY GAP is PP-complete.

Lemma 3.20. *For all c, s with the $c - s$ gap at least an inverse exponential in input size,*

$$\text{NON-EMPTY GAP}(c, s) \in \text{PP}.$$

Proof. Our approach to show containment in PP is to give a polynomial-time many-one reduction of NON-EMPTY GAP(c, s) with $c - s$ at least an inverse exponential to QCIRCUIT(P, Q) with $P - Q$ an inverse exponential. (Note that even if $c - s \in \Omega(1)$, we will still have $P - Q$ an inverse exponential.) Let V_n be an input to NON-EMPTY GAP(c, s) (all circuits in this proof have the same implicitly fixed input x). We construct an instance V''_n of QCIRCUIT(P, Q) as follows.

The first step is to adjust the completeness and soundness parameters for NON-EMPTY GAP so that they "straddle" the midpoint 1/2. Formally, map $c > s$ to $c' > s'$, respectively, so that $c'(n) - 1/2 = 1/2 - s'(n)$. For this, construct the following circuit V'_n, whose completeness and soundness parameters we denote by c' and s', respectively.

If $c(n) + s(n) > 1$, then with probability

$$\alpha := \frac{c(n) + s(n) - 1}{c(n) + s(n)}, \tag{3.21}$$

reject, and with probability $1 - \alpha$, run V_n and output its answer.

The case of $c(n) + s(n) < 1$ is analogous, except with

$$\alpha := \frac{1 - c(n) - s(n)}{2 - c(n) - s(n)}. \tag{3.22}$$

Finally, if $c(n) + s(n) = 1$, set $c' = c$ and $s' = s$. (Here, we use Observation 2.9, which allows us to assume $c, s \in \mathbb{Q}$ with $\mathrm{poly}(n)$ bits of precision.)

Next, map V_n' to V_n'' as follows: Given an input $y \in \{0,1\}^m$, (1) run two copies of V_n' in parallel on y, (2) negate the output of the second copy of V_n' via a Pauli X gate, (3) apply an AND gate to both output qubits, and (4) measure in the standard basis. Let p_y denote the probability that V_n' accepts y. Then, V_n'' accepts y with probability $p_y(1 - p_y)$.

Correctness. Intuitively, since the function $f(x) = x(1 - x)$ is maximized over $x \in [0, 1]$ when $x = 1/2$, the acceptance probability of V_n'' is maximized when y falls into the promise gap of V_n', i.e., $p_y \approx 1/2$. Formally, let $c'(n) = 1/2 + \gamma$ and $s'(n) = 1/2 - \gamma$ for $\gamma \in (0, 1/2]$, and express $p_y = 1/2 + \delta$ for bias $\delta \in [-1/2, 1/2]$. Then, V_n'' accepts y with probability

$$p_y(1 - p_y) = 1/4 - \delta^2. \tag{3.23}$$

It follows that if $p_y \geq c'(n)$ or $p_y \leq s'(n)$, then V_n'' accepts y with probability $Q \leq 1/4 - \gamma^2$, and if $s'(n) < p_y < c'(n)$, then V_n'' accepts y with probability $P > 1/4 - \gamma^2$. By Observation 2.9, we may assume

$$P - Q \in \Omega(1/\exp(n)), \tag{3.24}$$

thus yielding that a YES instance of NON-EMPTY GAP(c, s) with at least inverse exponential $c - s$ is mapped to a NO instance of QCIRCUIT(P, Q) and vice versa with inverse exponential $P - Q$. The claim now follows by Lemma 3.3, which says QCIRCUIT$(P, Q) \in$ PP. □

Lemma 3.25. *There exist* $c, s \in \Theta(1)$ *such that* NON-EMPTY GAP(c, s) *is* PP*-hard.*

Proof. Let $\phi : \{0,1\}^n \to \{0,1\}$ be an instance of the PP-complete problem MAJSAT (see proof of Lemma 3.18). We construct an instance V_n with fixed input x of NON-EMPTY GAP(c,s) with $c - s \in \Theta(1)$ as follows. Let V'_n be a polynomial-size quantum circuit which prepares the state

$$2^{-n/2} \sum_{x \in \{0,1\}^n} |x\rangle_A |\phi(x)\rangle_B \in (\mathbb{C}^2)^{\otimes n+1}, \tag{3.26}$$

then measures register B in the standard basis, and accepts if and only if it obtains result 1. If ϕ is a YES instance, then V'_n accepts with probability in range $[1/2 + 1/2^n, 1]$, and if ϕ is a NO instance, V'_n accepts with probability in range $[0, 1/2]$. Thus, setting $c = 3/4$, $s = 1/4$, and constructing circuit V_n which with probability 1/2 rejects, and with probability 1/2 runs V'_n and outputs its answer, yields the claim. □

Lemmas 3.20 and 3.25 immediately yield the following.

Theorem 3.27. *There exist* $c, s \in \Theta(1)$ *such that* NON-EMPTY GAP(c,s) *is* PP-*complete.*

4. Bounding the complexity of $\mathrm{Q}\Sigma_2$ and $\mathrm{Q}\Sigma_3$

In this section, we upper bound the complexity of the second and third levels of our fully quantum hierarchy. For brevity, we sometimes use shorthand $\mathrm{Q}\Sigma_2$ and $\mathrm{Q}\Pi_2$ to refer to $\mathrm{Q}\Sigma_2(c,s)$ and $\mathrm{Q}\Pi_2(c,s)$, respectively, for completeness and soundness parameters c and s, respectively. We begin by restating Theorem 1.6 as follows.

Theorem 4.1. *For any polynomial* r *and input size* n*, if* $c - s \geq 1/2^{2^{r(n)}}$*, then* $\mathrm{Q}\Sigma_2(c,s) \subseteq \mathrm{EXP}$ *and*

$\mathrm{Q}\Pi_2(c,s) \subseteq \mathrm{EXP}$*, when* c *and* s *are computable in exponential time in* n*.*

Note for classes with small completeness-soundness gaps such as these, a gate set must be fixed[11]. However, this result is independent of a fixed gate set.

[11] The Solovay–Kitaev algorithm (see, e.g., Dawson & Nielsen 2006) allows one to convert between gate sets in time scaling polylogarithmically in $1/\epsilon$ per

Proof. It suffices to show the first containment for $Q\Sigma_2$, the second containment holds by taking complements and noting that coEXP = EXP.

Given a $Q\Sigma_2$ instance, let its two proofs be denoted ρ_1 and ρ_2, with the former existentially quantified and the latter universally quantified. Let α be the maximum acceptance probability of a $Q\Sigma_2(c, s)$ protocol, i.e., the special case of Equation (1.7) such that

$$\alpha := \max_{\rho_1} \min_{\rho_2} \langle C, \rho_1 \otimes \rho_2 \rangle \tag{4.2}$$

for accepting POVM operator C (the input x to the $Q\Sigma_2$ instance is implicitly fixed into C). We wish to decide in exponential time whether $\alpha \geq c$ or $\alpha \leq s$. Since the promise gap satisfies $c - s \geq 1/2^{2^{r(n)}}$, it suffices to approximate α within additive error (say) $\frac{1}{4}(c - s)$. Hence, we show how to compute $\gamma \in \mathbb{R}$ such that $|\gamma - \alpha| \leq 1/(4 \cdot 2^{2^{r(n)}})$ in exponential time.

Beginning with Equation (4.2), note that we can write C as

$$\begin{aligned} C = \operatorname{Tr}_{\text{anc}} \big[\; & (I \otimes |0 \cdots 0\rangle\langle 0 \cdots 0|_{\text{anc}}) V_n^\dagger (|1\rangle\langle 1|_{\text{out}} \otimes I) \\ & V_n (I \otimes |0 \cdots 0\rangle\langle 0 \cdots 0|_{\text{anc}}) \; \big] \end{aligned} \tag{4.3}$$

for verification circuit V_n. By definition, V_n is generated by a polynomial-time Turing machine, which we assume specifies V_n via a sequence of gates from a universal gate set G (e.g., {CNOT, H, T}). Since we wish to proceed via numerical optimization techniques, we begin by computing a numerical approximation C' to C. Specifically, in exponential time, we can approximate each entry[12]

Footnote 11 continued
gate, where ϵ is the desired approximation precision per gate. Thus, the setting of doubly exponentially small precision takes superpolynomial-overhead to convert between gate sets, which is problematic for promise classes involving *polynomial*-time uniform circuit families (such as $Q\Sigma_2$ and $Q\Pi_2$). For this reason, in the small gap regime, one can "circumvent" the problem by fixing a gate set when defining the class.

[12] This can be accomplished in exponential time as follows. Replace gate set G with G' by approximating each entry of each gate in G using $2^{s(n)}$ bits of precision, for some sufficiently large, fixed polynomial s. Define C' as C, except each use of a gate $U \in G$ is replaced with its approximation $U' \in G'$. Then, via

of C using $2^{q(n)}$ bits of precision, for some polynomial q. Therefore, we have

$$\begin{aligned} |\langle C - C', \sigma_1 \otimes \sigma_2 \rangle| &\leq \|C - C'\|_F \|\sigma_1 \otimes \sigma_2\|_F \\ &\leq \|C - C'\|_F = O\left(2^{2p(n)-2^{q(n)}}\right) \end{aligned} \tag{4.4}$$

for any density matrices σ_1, σ_2. (Recall $p(n)$ is the size of each proof, for some polynomial p.) Therefore, for sufficiently large polynomial q, we have that

$$\alpha' := \max_{\rho_1} \min_{\rho_2} \{\langle C', \rho_1 \otimes \rho_2 \rangle : \mathrm{Tr}(\rho_1) = \mathrm{Tr}(\rho_2) = 1,\ \rho_1, \rho_2 \succeq 0\} \tag{4.5}$$

satisfies $|\alpha - \alpha'| \leq \frac{1}{8} \cdot 2^{-2^{r(n)}} \leq \frac{1}{8}(c - s)$.

We now use SDP duality (in a manner reminiscent of LP solutions for the Chebyshev approximation problem, p. 293 of Boyd & Vandenberghe 2004) to rephrase Equation (4.5) as an SDP. Suppose we fix a feasible ρ_1 and solve the inner optimization problem in (4.5). Then:

$$\alpha'(\rho_1) := \min_{\rho_2} \{\langle C', \rho_1 \otimes \rho_2 \rangle : \mathrm{Tr}(\rho_2) = 1,\ \rho_2 \succeq 0\}.$$

We can rewrite $\langle C', \rho_1 \otimes \rho_2 \rangle$ as $\langle \mathrm{Tr}_1[(\rho_1 \otimes I)C'], \rho_2 \rangle$ where Tr_1 is the partial trace over the register that ρ_1 acts on. As the partial trace is cyclic over the target subsystem, $\mathrm{Tr}_1[(\rho_1 \otimes I)C'] = \mathrm{Tr}_1[(\rho_1^{1/2} \otimes I)C'(\rho_1^{1/2} \otimes I)]$. Because the approximation[12] of C preserves Hermiticity and the partial trace is completely positive and preserves Hermiticity, this term is Hermitian and positive semidefinite. This implies that the best choice for ρ_2 is a rank-1 projector onto the eigenspace corresponding to the least eigenvalue. In other words, $\alpha'(\rho_1) = \lambda_{\min}(\mathrm{Tr}_1[(\rho_1 \otimes I)C'])$ where $\lambda_{\min}(X)$ denotes the

Footnote 12 continued
the well-known bound $\|U_m \cdots U_1 - V_m \cdots V_1\|_\infty \leq \sum_{i=1}^m \|U_i - V_i\|_\infty$ (for unitary U_i, V_i), it follows that $\|C' - C\|_\infty \in O(\mathrm{poly}(n)/(2^{2^{s(n)}}))$, since V_n contains $\mathrm{poly}(n)$ gates. Here, $\|A\|_\infty = \max_{|\psi\rangle} \|A\,|\psi\rangle\|_2$ for unit vectors $|\psi\rangle$ denotes the spectral or operator norm. Finally, apply the fact that $\max_{i,j} |A(i,j)| \leq \|A\|_\infty$ (p. 314 of Horn & Johnson 1990).

least eigenvalue of the operator X. For fixed ρ_1, this minimum eigenvalue calculation can be rephrased via the dual optimization program for $\alpha'(\rho_1)$,

$$\alpha'(\rho_1) = \max_{t} \{t : tI \preceq \mathrm{Tr}_1[(\rho_1 \otimes I)C']\}.$$

Re-introducing the maximization over ρ_1, we hence obtain

$$\alpha' = \max_{\rho_1,t} \{t : tI \preceq \mathrm{Tr}_1[(\rho_1 \otimes I)C'],\ \mathrm{Tr}(\rho_1) = 1,\ \rho_1 \succeq 0\},$$

which is a semidefinite program.

With an SDP in hand, we now apply the ellipsoid method to obtain an estimate, γ, for α'. Note that not all SDPs can be solved in polynomial time, as the runtime of the ellipsoid method depends in part on two parameters, R and ϵ, where R is the radius of a ball (with respect to the Euclidean norm) containing the feasible region, and ϵ is the radius of a ball contained in the feasible region (see Grötschel *et al.* 1993 for details). For this reason, we give an equivalent SDP which allows us to bound R and ϵ as follows. First, relax the constraint $\mathrm{Tr}(\rho) = 1$ to $\mathrm{Tr}(\rho) \leq 1$. Second, replace t with $t_1 - t_2$ where $t_1, t_2 \geq 0$. From context, we know t is a probability, and so we have the implicit constraint $t \in [0, 1]$. Therefore, we add redundant constraints $t_1, t_2 \leq 100$ without changing α'. Thus, we have the following reformulation of α'.

$$\alpha' = \max_{\rho_1,t_1,t_2} \{t_1 - t_2 : (t_1 - t_2)I \preceq \mathrm{Tr}_1[(\rho_1 \otimes I)C'], \\ \mathrm{Tr}(\rho_1) \leq 1,\ \rho_1 \succeq 0,\ t_1, t_2 \in [0, 100]\}. \tag{4.6}$$

We can now use the ellipsoid method to approximately solve this SDP in time that is exponential in n. We follow a similar analysis to Watrous (2009b) and find a γ such that $|\gamma - \alpha'| \leq \epsilon$ in time

$$\mathrm{poly}(\log(R), \log(1/\epsilon), n', m, J), \tag{4.7}$$

for parameters R, ϵ, n', m, and J defined as:

- R: This is equal to the maximum of $\|\rho_1 \oplus t_1 \oplus t_2\|_F$ over all feasible (ρ_1, t_1, t_2). Since we have

  $$\|\rho_1 \oplus t_1 \oplus t_2\|_F \leq \|\rho_1 \oplus t_1 \oplus t_2\|_1 \\ = \mathrm{Tr}(\rho_1 \oplus t_1 \oplus t_2) = \mathrm{Tr}(\rho_1) + t_1 + t_2 \leq 201,$$

  for feasible (ρ_1, t_1, t_2), we can set $R = 201$.

- ϵ: This is the radius of a small ball contained in the feasible region. Specifically, ϵ is defined so that there exists feasible (ρ_1, t_1, t_2) such that $(\rho_1, t_1, t_2) + (\sigma, \tau_1, \tau_2)$ is feasible for all (σ, τ_1, τ_2) with $\|\sigma \oplus \tau_1 \oplus \tau_2\|_F \leq \epsilon$. Since we have

  $$\epsilon \geq \|\sigma \oplus \tau_1 \oplus \tau_2\|_F = \|\sigma\|_F + |\tau_1| + |\tau_2| \geq \max\{\|\sigma\|_F, |\tau_1|, |\tau_2|\},$$

  we will use the more convenient bound $\max\{\|\sigma\|_F, \tau_1, \tau_2\} \leq \epsilon$ for the analysis. We choose the interior point

  $$\rho_1 = \frac{1}{\dim(\rho_1)^2} I, \quad t_1 = 10, \quad t_2 = 20,$$

  and

  $$\epsilon = \frac{1}{8} \cdot 2^{(-2^{r(n)})} \leq \frac{1}{8}(c - s).$$

  Note that this has the sufficiently small accuracy we require.

  We now prove that $(\rho_1, t_1, t_2) + (\sigma, \tau_1, \tau_2)$ is feasible so long as $\max\{\|\sigma\|_F, \tau_1, \tau_2\} \leq \epsilon$. One can check that for these values, we have

  1. $\rho_1 + \sigma \succeq \rho_1 - \|\sigma\|_\infty I \succeq \rho_1 - \epsilon I \succeq 0$, (where we used $\|\sigma\|_F \leq \epsilon$ implies $\|\sigma\|_\infty \leq \epsilon$),
  2. $\mathrm{Tr}(\rho_1 + \sigma) = \mathrm{Tr}(\rho_1) + \mathrm{Tr}(\sigma) \leq \mathrm{Tr}(\rho_1) + \|\sigma\|_F \|I\|_F \leq \frac{1}{\dim(\rho_1)} + \epsilon\sqrt{\dim(\rho_1)} \leq 1$, (using the Cauchy-Schwarz inequality),
  3. $t_1 + \tau_1 \in [0, 100]$ and $t_2 + \tau_2 \in [0, 100]$,
  4. $((t_1 + \tau_1) - (t_2 + \tau_2))I \prec 0 \preceq \mathrm{Tr}_1[((\rho_1 + \sigma) \otimes I)C']$, (since $(t_1 + \tau_1) - (t_2 + \tau_2) < 0$ and $\rho_1 + \sigma \succeq 0$ as shown above).

- n': The dimension of $\rho_1 \oplus t_1 \oplus t_2$, which is equal to the sum of the dimensions, i.e., $O(2^{p(n)})$.

- m: The dimension of the operators appearing in the constraints. Note, from Equation (4.6), that constraint $(t_1 - t_2)I \preceq \mathrm{Tr}_1[(\rho_1 \otimes I)C]$ involves operators acting on a space of dimension $O(2^{p(n)})$. Moreover, there are only 3 other inequality constraints: $\mathrm{Tr}(\rho) \leq 1$ and $t_1, t_2 \leq 100$. Thus, $m = O(2^{p(n)})$.

- ○ J: The maximum bit-length of the entries in C', which is $2^{q(n)}$, by definition.

We conclude that the EXP protocol approximates α' via γ, which is correct up to an additive error of $\frac{1}{8}(c-s)$. Finally, if $\gamma \geq (c+s)/2$, we output YES (i.e., $x \in A_{\text{yes}}$). Otherwise, we output NO (i.e., $x \in A_{\text{no}}$). □

Using the power of non-determinism, we can also bound the complexity of $\mathrm{Q}\Sigma_3$ and $\mathrm{Q}\Pi_3$.

THEOREM 4.8. *For any polynomial r and input size n, if $c - s \geq 1/r(n)$, then*

$$\mathrm{QMA}(2) \subseteq \mathrm{Q}\Sigma_3 \subseteq \mathrm{NEXP} \quad \textit{and} \quad \mathrm{coQMA}(2) \subseteq \mathrm{Q}\Pi_3 \subseteq \mathrm{coNEXP},$$

where all classes have completeness and soundness c and s, respectively. Moreover, if we allow smaller gaps (in principle, gaps which are at most inverse singly exponential in n suffice), such as $c - s \geq 1/2^{2^{r(n)}}$, then

$$\begin{aligned} &\mathrm{QMA}(2)(c,s) = \mathrm{Q}\Sigma_3(c,s) = \mathrm{NEXP} \\ \textit{and} \quad &\mathrm{coQMA}(2) = \mathrm{Q}\Pi_3(c,s) = \mathrm{coNEXP}, \end{aligned}$$

where we assume c and s are computable in exponential time in n.

Note for classes with doubly exponentially small completeness-soundness gaps, a gate set must be fixed. However, this result is independent of the choice of a fixed gate set.

PROOF. Again, we prove the statements involving $\mathrm{Q}\Sigma_3$ and the analogous statements involving $\mathrm{Q}\Pi_3$ follow by taking complements.

Consider the maximum acceptance probability of a $\mathrm{Q}\Sigma_3$ protocol,

$$\begin{aligned} \beta := \max_{\rho_1} \min_{\rho_2} \max_{\rho_3} \{ &\langle C, \rho_1 \otimes \rho_2 \otimes \rho_3 \rangle : \\ &\mathrm{Tr}(\rho_1) = \mathrm{Tr}(\rho_2) = \mathrm{Tr}(\rho_3) = 1,\ \rho_1, \rho_2, \rho_3 \succeq 0 \} \end{aligned}$$

where C is the POVM element corresponding to the verifier accepting (again, the input x to the $\mathrm{Q}\Sigma_3$ protocol is implicitly fixed into C).

As in the proof of Theorem 4.1, define C' to be equal to C where each entry is correct up to $2^{q(n)}$ bits of precision. Consider now the optimization problem

$$\beta' := \max_{\rho_1} \min_{\rho_2} \max_{\rho_3} \{\langle C', \rho_1 \otimes \rho_2 \otimes \rho_3 \rangle : \\ \mathrm{Tr}(\rho_1) = \mathrm{Tr}(\rho_2) = \mathrm{Tr}(\rho_3) = 1,\ \rho_1, \rho_2, \rho_3 \succeq 0\}$$

and note that $|\beta' - \beta| = O(2^{3p(n)-2^{q(n)}})$ by an argument similar to Equation (4.4).

Now, suppose we non-deterministically guess a value for the optimal $\rho_1^\star$ (which, recall, exists by compactness) by a matrix, each of whose entries is specified up to $2^{t(n)}$ bits of precision for some polynomial t. Call this approximation ρ_1'. We now consider the optimization problem

$$\begin{aligned} \beta'' &:= \min_{\rho_2} \max_{\rho_3} \{\langle C', \rho_1' \otimes \rho_2 \otimes \rho_3 \rangle : \mathrm{Tr}(\rho_2) = \mathrm{Tr}(\rho_3) = 1, \rho_2, \rho_3 \succeq 0\} \\ &= \min_{\rho_2} \max_{\rho_3} \{\langle C'', \rho_2 \otimes \rho_3 \rangle : \mathrm{Tr}(\rho_2) = \mathrm{Tr}(\rho_3) = 1,\ \rho_2, \rho_3 \succeq 0\}, \end{aligned}$$

where $C'' := \mathrm{Tr}_1[(\rho_1' \otimes I \otimes I)C']$ is the matrix which hardcodes ρ_1' into C'. We now bound $|\beta' - \beta''|$. For any density operators σ_2 and σ_3, we have

$$\begin{aligned} |\langle C', (\rho_1^\star - \rho_1') \otimes \sigma_2 \otimes \sigma_3 \rangle| &\le \|C'\|_F \|\rho_1^\star - \rho_1'\|_F, \\ \|C'\|_F \le \|C' - C\|_F + \|C\|_F &\le 2\|I\|_F = O(2^{\frac{3p(n)}{2}}), \\ \|\rho_1^\star - \rho_1'\|_F &= O(2^{p(n)-2^{t(n)}}). \end{aligned}$$

Thus, we have $|\beta' - \beta''| = O(2^{\frac{5p(n)}{2}-2^{t(n)}})$ and therefore we can choose q and t such that

$$|\beta - \beta''| \le |\beta - \beta'| + |\beta' - \beta''| \le \frac{1}{8} \cdot 2^{-2^{r(n)}} \le \frac{1}{8}(c - s)$$

as before.

Since C'' still has exponential bit-length, is positive semidefinite, and can be computed in non-deterministic exponential time, we can repeat the arguments from Theorem 4.1 to find γ in exponential time such that $|\beta'' - \gamma| \le \frac{1}{8}(c - s)$. Thus, if $\gamma \ge \frac{1}{2}(c + s)$

we output YES (i.e., $x \in A_{\text{yes}}$). Otherwise, we output NO (i.e., $x \in A_{\text{no}}$). This yields

$$\mathrm{Q}\Sigma_3 \subseteq \mathrm{NEXP}, \tag{4.9}$$

even when $\mathrm{Q}\Sigma_3$ has small gap. The rest of the theorem holds by a result of Pereszlényi (2012), who proved that the equality $\mathrm{QMA}(2)(c,s) = \mathrm{NEXP}$ holds when $c - s \geq 1/2^{2^r}$ for polynomial r. Combining this with the fact that $\mathrm{QMA}(2) \subseteq \mathrm{Q}\Sigma_3$ and Equation (4.9) finishes the proof. □

5. A Karp–Lipton-type theorem

The Karp–Lipton (1980) theorem showed that if $\mathrm{NP} \subseteq \mathrm{P}_{/\mathrm{poly}}$ (i.e., if NP can be solved by polynomial-size non-uniform circuits), then $\Sigma_2 = \Pi_2$ (which in turn collapses PH to its second level). Then, building on the conjecture that the polynomial hierarchy is infinite, this result implies that $\mathrm{NP} \not\subseteq \mathrm{P}_{/\mathrm{poly}}$ (a stronger claim than $\mathrm{P} \neq \mathrm{NP}$ as $\mathrm{P} \subseteq \mathrm{P}_{/\mathrm{poly}}$). Some attempts to separate NP from P use this as a basis to try and prove the stronger claim instead. For instance, this has led to the approach of proving super-polynomial circuit lower bounds for circuits of NP-complete problems. Here, we show that the proof technique used by Karp and Lipton carries over directly to the quantum setting, *provided* one uses the stronger hypothesis

$$\text{Precise-QCMA} \subseteq \mathrm{BQP}_{/\mathrm{mpoly}} \tag{5.1}$$

(as opposed to $\mathrm{QCMA} \subseteq \mathrm{BQP}_{/\mathrm{mpoly}}$). Whether this causes QCPH to collapse to its second level, however, remains open (see Remark 5.10 below). We begin by formally defining $\mathrm{BQP}_{/\mathrm{mpoly}}$ and Precise-QCMA.

Definition 5.2 ($\mathrm{BQP}_{/\mathrm{mpoly}}$). *A promise problem $A = (A_{\text{yes}}, A_{\text{no}})$ is in $\mathrm{BQP}_{/\mathrm{mpoly}}$ if there exists a polynomial-time uniform family of quantum circuits $\{C_n\}_{n\in\mathbb{N}}$ and a collection of binary advice strings $\{a_n\}_{n\in\mathbb{N}}$ with $|a_n| = \mathrm{poly}(n)$, such that for all n and all strings x where $|x| = n$, $\Pr[C_n(|x\rangle, |a_n\rangle) = 1] \geq 2/3$ if $x \in A_{\text{yes}}$ and $\Pr[C_n(|x\rangle, |a_n\rangle) = 1] \leq 1/3$ if $x \in A_{\text{no}}$.*

Equivalently, $\mathrm{BQP}_{/\mathrm{mpoly}}$ is the set of promise problems solvable by a *non-uniform* family of polynomial-size bounded error quantum circuits. It is used as a quantum analogue for $\mathrm{P}_{/\mathrm{poly}}$ in this scenario.[13] Here, we remark on the use of mpoly instead of poly in Definition 5.2. Note that $\mathrm{BQP}_{/\mathrm{poly}}$ accepts Karp–Lipton style advice, i.e., it is a BQP circuit that accepts a polynomial-size advice string to provide *some answer* with probability at least 2/3 even if the "advice is bad." On the other hand, $\mathrm{BQP}_{/\mathrm{mpoly}}$ accepts Merlin style advice, i.e., it is a BQP circuit accepting polynomial-size classical advice such that the output is correct with probability at least 2/3 if the "advice is good." Note $\mathrm{BQP}_{/\mathrm{poly}}$ versus $\mathrm{BQP}_{/\mathrm{mpoly}}$ is analogous to the "$\exists \cdot$ BPP versus MA" phenomenon. Moreover, as we are concerned with variations of QCMA, and not $\exists \cdot$ BQP, $\mathrm{BQP}_{/\mathrm{mpoly}}$ is the right candidate for us.

Definition 5.3 (Precise-QCMA). *A promise problem $A = (A_{\mathrm{yes}}, A_{\mathrm{no}})$ is said to be in* Precise-QCMA(c, s) *for polynomial-time computable functions $c, s : \mathbb{N} \to [0, 1]$ if there exists polynomially bounded functions $p, q : \mathbb{N} \to \mathbb{N}$ such that $\forall \ell \in \mathbb{N},\ c(\ell) - s(\ell) \geq 2^{-q(\ell)}$, and there exists a polynomial-time uniform family of quantum circuits $\{V_n\}_{n\in\mathbb{N}}$ that takes input x and a classical proof $y \in \{0, 1\}^{p(n)}$ and outputs a single qubit. Moreover, for an n-bit input x:*

- *Completeness: If $x \in A_{\mathrm{yes}}$, then $\exists\, y$ such that $V_n(x, y)$ accepts with probability at least $c(n)$.*
- *Soundness: If $x \in A_{\mathrm{no}}$, then $\forall\ y$, $V_n(x, y)$ accepts with probability at most $s(n)$.*

Define
$$\text{Precise-QCMA} = \bigcup_{c-s\in\Omega(1/\exp(n))} \text{Precise-QCMA}(c, s).$$

[13] In contrast to $\mathrm{P}_{/\mathrm{poly}}$, we use polynomial-size uniform quantum circuit families accepting advice in Definition 5.2 instead of polynomial-time bounded error quantum Turing machines. The two models are computationally equivalent (Molina & Watrous 2019).

Observation 5.4. *The proof of Theorem 2.7 and footnote 2 in Jordan* et al. *(2012) show that by choosing an appropriate universal gate set (e.g., Hadamard, Toffoli, NOT), one has that*

$$\text{Precise-QCMA} = \text{Precise-QCMA}(1, 1 - 1/\exp(n)). \tag{5.5}$$

As an aside, note QCMA is defined with $c - s \in \Omega(1/\text{poly}(n))$. Recall from the discussion in Section 1.1 that the main obstacle to the recursive arguments that work well for NP in Karp & Lipton (1980) is the "promise problem" nature of $\text{QC}\Pi_2$ and QCMA. However, exploiting the perfect completeness of Precise-QCMA and the fact that $\forall s \le s' < c$, $\text{Precise-QCMA}(c, s) \subseteq \text{Precise-QCMA}(c, s')$, we "recover" the notion of a decision problem in a rigorous sense by working with Precise-QCMA as demonstrated below.

Claim 5.6. *For every promise problem $A = (A_{\text{yes}}, A_{\text{no}}) \in$ Precise-QCMA(c, s) with verifier V, where an input x is hardcoded into V, there exists a verifier V' (a polynomial-time uniform quantum circuit family), a polynomial q and a language $A' = (A_{\text{yes}}, \{0, 1\}^* \setminus A_{\text{yes}})$ such that $A' \in$ Precise-QCMA$(1, 1 - 2^{-q(n)})$ with verifier V'. Moreover, for all proofs y, V' accepts y with probability either 1 or at most $1 - 2^{-q(n)}$.*

Proof. By Observation 5.4, we may assume

$$\text{Precise-QCMA}(c, s) = \text{Precise-QCMA}(1, 1 - 2^{-p(n)})$$

for some polynomial p. Let $A = (A_{\text{yes}}, A_{\text{no}}) \in \text{Precise-QCMA}(1, 1 - 2^{-p(n)})$ be a promise problem with verifier V, where an input x has already been hardcoded into the verifier. The concern is that for $x \in A_{\text{yes}}$, there may exist a proof y accepted by V with probability in $(1 - 2^{-p(n)}, 1)$. By Observation 2.9, we may we modify V to create V' such that the acceptance probabilities of V' are integer multiples of $2^{-g(n)}$ for some polynomial g. Since p and g are polynomials, there exists $n_0 \ge 0$ such that $\forall n \ge n_0$, either $p(n) \ge g(n)$ or vice versa. Thus, setting q equal to p in the former case or equal to g in the latter case ensures that for sufficiently large n, no proofs are accepted by V' with probability in $(1 - 2^{-q(n)}, 1)$. This yields the second claim of the observation. The first claim also follows, since all inputs in A_{yes} have proofs which accept with probability 1, and if no proofs are accepted in the promise gap, then certainly the optimal proofs for inputs in $\{0,1\}^* \setminus A_{\mathrm{yes}}$ are not accepted with probability in the gap. □

Note that the same process fails to map a promise problem $\mathcal{P} \in \mathrm{QCMA}(1, s)$ to some corresponding decision problem $\mathcal{P}' \in \mathrm{QCMA}(1, s')$ where $s < s'$. As shown above, s' could very well be exponentially close to 1, which would violate the requirement for QCMA that the promise gap should be an inverse polynomial function in the input size.

Building on this "decision problem" flavor of Precise-QCMA, we first show the following.

LEMMA 5.7. *Suppose* Precise-QCMA $\subseteq \mathrm{BQP}_{/\mathrm{mpoly}}$. *Then, for every promise problem* $A = (A_{\mathrm{yes}}, A_{\mathrm{no}})$ *in* Precise-QCMA *and every n-bit input x, there exists a polynomially bounded function $p : \mathbb{N} \to \mathbb{N}$ and a bounded error polynomial-time non-uniform quantum circuit family $\{C_n\}_{n\in\mathbb{N}}$ such that:*

- *if $x \in A_{\mathrm{yes}}$, then $C_n(x)$ outputs valid proof $y \in \{0,1\}^{p(n)}$ with probability exponentially close to* 1 *such that (x, y) is accepted by the corresponding* Precise-QCMA *verifier with probability* 1*;*
- *if $x \in A_{\mathrm{no}}$, then $C_n(x)$ outputs a symbol $\perp$ with probability exponentially close to* 1 *signifying that there is no $y \in \{0,1\}^{p(n)}$, such that (x, y) is accepted by the corresponding* Precise-QCMA *verifier with probability* 1.

PROOF. To begin, recall from Claim 5.6 that we may assume that a given promise problem A in Precise-QCMA has (a) completeness/soundness parameters $(1, 1 - 2^{-q(n)})$ for a polynomial q and (b) a verifier V_n which on an n-bit input x accepts no proofs with probability in the promise gap. Since Precise-QCMA $\subseteq \mathrm{BQP}_{/\mathrm{mpoly}}$, by assumption, there exists a non-uniform polynomial-size quantum circuit family $\{C'_n\}_{n\in\mathbb{N}}$ that receives x as input such that for any $x \in A_{\mathrm{yes}}$, C'_n accepts with probability at least 2/3 and rejects with probability 2/3 otherwise. By using standard parallel repetition, we may assume without loss of generality that $C'_n(x)$ accepts or rejects the corresponding cases with probability at least $1 - 2^{-p(n)}$ for some polynomial p. Now, we would like to construct a $\mathrm{BQP}_{/\mathrm{mpoly}}$ circuit C_n that uses the input x and a description of C'_n as polynomial-size advice and outputs a valid proof y such that V_n accepts (x, y) with probability 1.

The construction of $\{ C_n \}$ is now as follows. C_n first runs C'_n using x to check if $x \in A_{\mathrm{yes}}$; if not, it rejects and outputs $\perp$. To find a proof y, we now use standard self-reducibility ideas from SAT, coupled with the crucial Claim 5.6. Specifically, fix $y_1 = 0$ (i.e., the first bit of y) to obtain a new circuit $C'_{n,1}$, run $C'_{n,1}$ on (x, y_1) and record its answer $z_1 \in \{ 0, 1 \}$. Since no proofs are accepted in the gap as per Claim 5.6, $C'_{n,1}$ is a valid Precise-QCMA verifier (i.e., satisfying the promise of the completeness/soundness parameters). Thus, with high probability, if $z_1 = 1$ there is an accepting proof for x whose first bit is 0 and if $z_1 = 0$, there is a proof with the first bit set to 1. Hence, we can fix y_1's accordingly. Iterating this process successively for all remaining bits of y yields the claim. □

We next give a quantum-classical analogue of the Karp–Lipton theorem.

Theorem 1.11 (A Quantum-Classical Karp–Lipton Theorem). *If* Precise-QCMA $\subseteq \mathrm{BQP}_{/\mathrm{mpoly}}$ *then* $\mathrm{QC}\Pi_2 = \mathrm{QC}\Sigma_2$.

Proof. We essentially follow the proof of the original Karp–Lipton theorem, coupled with careful use of Observation 2.9. To show $\mathrm{QC}\Pi_2 = \mathrm{QC}\Sigma_2$, it suffices to show that $\mathrm{QC}\Pi_2 \subseteq \mathrm{QC}\Sigma_2$. To see this, consider promise problem $A \in \mathrm{QC}\Sigma_2$. Now, $\bar{A}$ (the complement of A) is in $\mathrm{QC}\Pi_2$ by definition. However, if $\mathrm{QC}\Pi_2 \subseteq \mathrm{QC}\Sigma_2$, then $\bar{A} \in \mathrm{QC}\Sigma_2$, which in turn implies by definition that $A \in \mathrm{QC}\Pi_2$, as desired.

To show $\mathrm{QC}\Pi_2 \subseteq \mathrm{QC}\Sigma_2$, let $A = (A_{\mathrm{yes}}, A_{\mathrm{no}})$ be a $\mathrm{QC}\Pi_2$ problem. As $\mathrm{QC}\Pi_2$ has perfect completeness from Result 2.10, there exist polynomials p, r and a polynomial-time uniform family of quantum circuits $\{V_i\}_{i\in\mathbb{N}}$ that take as input a string $x \in \{0,1\}^n$ for some $n \in \mathbb{N}$, two classical proofs $u, v \in \{0,1\}^{p(n)}$, and outputs a single qubit such that:

$$x \in A_{\text{yes}} \quad \Rightarrow \quad \forall u\, \exists v\;\; \Pr[V_n(x,u,v)=1]=1, \tag{5.8}$$

$$x \in A_{\text{no}} \quad \Rightarrow \quad \exists u\, \forall v\;\; \Pr[V_n(x,u,v)=1] \leq \frac{1}{2^{r(n)}}. \tag{5.9}$$

Let us now highlight the difficulty in proving the claim for QCMA instead of Precise-QCMA. Specifically, if we fix the first proof u, what we would ideally require is that the resulting existentially quantified computation over v, denoted M_u[14], is in QCMA. Indeed, if $x \in A_{\text{yes}}$, then for any fixed u, there exists a v causing $M_u(x,v)$ to accept with certainty (henceforth x is implicitly fixed into M_u). The problem arises when $x \in A_{\text{no}}$, in which case, we require that for all v, M_u accepts with probability at most s for some soundness parameter s inverse polynomially gapped away from 1. Unfortunately, the definition of $\text{QC}\Pi_2$ only ensures this holds for *some* u, and not necessarily *all* u. To circumvent this, we use Observation 2.9, which implies we may assume M_u's acceptance probabilities are given by rational numbers with poly(n) bits of precision (assuming an appropriate universal gate set is used). It follows that if M_u does not accept some v with probability 1, then it must reject v with probability at least $1-2^{-q(n)}$ for some efficiently computable polynomial q. Thus, by definition M_u is a Precise-QCMA$(1, 1-2^{-q(n)})$ computation, to which we may now apply our hypothesis that Precise-QCMA $\subseteq \text{BQP}_{/\text{mpoly}}$. (Note: There is a subtle point here—the precise choice of q depends on the length of circuit M_u, which in turn depends on the Hamming weight of u, since we can simulate "fixing" u by adding appropriate Pauli X gates to our circuit. Nevertheless, it is trivial to choose a polynomial q which provides sufficient precision in our rational approximation in order to accommodate the fixing of proofs u of any Hamming weight.)

Since for any fixed u, M_u denotes a Precise-QCMA computation, our assumption says that there exists a non-uniform family of polynomial-size bounded-error quantum circuits $\{\, Q'_n \,\}_{n\in\mathbb{N}}$ (here, the $\text{BQP}_{/\text{mpoly}}$ advice strings are implicitly fixed into the circuits) that receive (x,u) and a description of M_u as input and outputs a

[14] Notice that M_u is the remnant of the verifier circuit obtained from V_n when the first proof register is loaded with u. In a slight abuse of notation, M_u will be referred to both as a computation and as a circuit.

bit such that:

- ∘ if there exists a proof v such that M_u accepts (x, v) with probability 1, then Q'_n accepts (x, u, M_u) with probability at least 2/3, and
- ∘ if for all proofs v, M_u accepts (x, v) with probability at most $1 - 2^{-q(n)}$, then Q'_n accepts (x, u, M_u) with probability at most 1/3.

Crucially, the set $\{ Q'_n \}$ is non-uniform, and thus Q'_n depends only on n, not the choice of x or u.

Continuing, from Lemma 5.7, we now conclude there exists a bounded-error polynomial-time non-uniform quantum circuit family $\{ Q_n \}_{n\in\mathbb{N}}$ which, whenever $x \in A_{\text{yes}}$, outputs a proof v which M_u accepts with certainty. For clarity, note that Q_n receives (x, u) and a description of M_u as input and outputs a string $v \in \{ 0, 1 \}^{p(n)}$. Suppose Q_n outputs the correct answer with probability at least $1 - 2^{-s(n)}$ for some polynomial s, as per Lemma 5.7. Using the existence and non-uniformity of $\{ Q_n \}$, as done in the proof of the classical Karp–Lipton theorem, we claim we may now swap the order of the quantifiers and write:

- ∘ If $x \in A_{\text{yes}}$, then $\exists Q_n \ \forall u \Pr[V_n(x, u, Q_n(x, u, M_u)) = 1] \geq 1 - 2^{-s(n)}$, and
- ∘ if $x \in A_{\text{no}}$, then $\forall Q_n \ \exists u \Pr[V_n(x, u, Q_n(x, u, M_u)) = 1] \leq \frac{1}{2^{r(n)}}$.

This would imply the desired claim that $\text{QC}\Pi_2 \subseteq \text{QC}\Sigma_2$.

To see that we may indeed swap quantifiers in this fashion, assume first that $x \in A_{\text{yes}}$. Then, choosing the non-uniform circuit family from Lemma 5.7 yields that for any fixed x and u, with probability at least $1 - 2^{-s(n)}$, Q_n outputs a proof v such that C_n accepts (x, u, v) with probability 1. Conversely, if $x \in A_{\text{no}}$, since for an appropriate choice of u, there are *no* proofs v such that C_n accepts (x, u, v) with probability more than $2^{-r(n)}$. Then, clearly no choice of Q_n is able to generate a proof $Q_n(x, u, M_u)$ such that C_n accepts $(x, u, Q_n(x, u, M_u))$ with probability more than $2^{-r(n)}$.

□

Remark 5.10 (Collapse of QCPH?). *An appeal of the classical Karp–Lipton theorem is that it implies that if* NP $\subseteq$ $\mathrm{P}_{/\mathrm{poly}}$*, then PH collapses to its second level; this is because if* $\Pi_2 = \Sigma_2$*, then* PH *collapses to* Σ_2*. Does an analogous statement hold for* QCPH *as a result of Theorem 1.11? Unfortunately, the answer is not clear. The problem is similar to that outlined in Remark 3.12. Namely, classically* $\Pi_2 = \Sigma_2$ *collapses* PH *since for any* Π_3 *decision problem, fixing the first (universally) quantified proof yields a* Σ_2 *computation. But this can be replaced with a* Π_2 *computation by assumption, yielding a computation with quantifiers* $\forall\forall\exists$*, which collapses to* $\forall\exists$*, i.e.,* $\Pi_3 \subseteq \Pi_2$*. In contrast, for (say)* $\mathrm{QC}\Pi_3$*, similar to the phenomenon in Remark 3.12, fixing the first (universally) quantified proof does* not *necessarily yield a* $\mathrm{QC}\Sigma_2$ *computation. Thus, a recursive application of the assumption* $\mathrm{QC}\Sigma_2 = \mathrm{QC}\Pi_2$ *cannot straightforwardly be applied.*

Since Precise-QCMA plays an important role in Theorem 1.11, we close with an equivalent characterization of Precise-QCMA.[15]

Lemma 5.11. Precise-QCMA = $\mathrm{NP}^{\mathrm{PP}}$.

Proof. Let V be a Precise-QCMA verifier. Using Claim 5.6, we may assume that for any proof y, V either accepts y with probability 1 or rejects with probability at most $1 - 2^{-q(n)}$. Thus, for any fixed y, the resulting computation V_y is a Precise-BQP = PP computation. This implies Precise-QCMA $\subseteq \exists \cdot$ PP (see also Remark 3.12). The fact $\exists \cdot \mathrm{PP} = \mathrm{NP}^{\mathrm{PP}}$ (Torán 1991) completes the proof of the forward direction.

To show the reverse direction, observe that Precise-QCMA $\supseteq$ $\exists \cdot$PP since the Precise-QCMA verifier can guess the classical proof and simulate the classical verification circuit with exponential precision. The previous reasoning then works in the reverse direction. □

[15] Thank you to an anonymous reviewer for suggesting to improve this from an upper bound to an equality. This potential improvement was also observed by Deshpande *et al.* (2022), which referenced an earlier version of this work to claim Precise-QCMA = $\mathrm{NP}^{\mathrm{PP}}$.

Acknowledgements

An extended abstract of an earlier version of this work has appeared as Gharibian *et al.* (2018).

SG and AS thank the Centre for Quantum Technologies at the National University of Singapore for their support and hospitality, where part of this research was carried out. SG acknowledges support from NSF grants CCF-1526189 and CCF-1617710. AS is supported by the Department of Defense. JY was supported by a Virginia Commonwealth University Presidential Scholarship. Research at the Centre for Quantum Technologies is partially funded by the Singapore Ministry of Education and the National Research Foundation under grant R-710-000-012-135. This research was supported in part by the QuantERA ERA-NET Cofund project QuantAlgo. This research was supported in part by Perimeter Institute for Theoretical Physics. Research at Perimeter Institute is supported by the Government of Canada through the Department of Innovation, Science and Economic Development Canada and by the Province of Ontario through the Ministry of Research, Innovation and Science.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

S. Aaronson, S. Beigi, A. Drucker, B. Fefferman & P. Shor (2009). The Power of Unentanglement. *Theory of Computing* **5**(1), 1–42.

S. Aaronson, A. Cojocaru, A. Gheorghiu & E. Kashefi (2019). Complexity-theoretic limitations on blind delegated quantum computation. In *46th International Colloquium on Automata, Languages,* *and Programming (ICALP 2019)*, C. Baier, I. Chatzigiannakis, P. Flocchini & S. Leonardi, editors, volume 132 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 6:1–6:13. Schloss Dagstuhl—Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany.

S. Aaronson & A. Drucker (2014). A full characterization of quantum advice. *SIAM Journal on Computing* **43**(3), 1131–1183.

D. Aharonov (2003). A simple proof that Toffoli and Hadamard are quantum universal. Available at arXiv:quant-ph/0301040v1.

D. Aharonov, M. Ben-Or, F. Brandão & O. Sattath (2022). The pursuit of uniqueness: Extending Valiant-Vazirani Theorem to the probabilistic and quantum Settings. *Quantum* **6**, 668.

D. Aharonov, A. Kitaev & N. Nisan (1998). Quantum circuits with mixed states. In *Proceedings of 13th ACM Symposium on Theory of Computing (STOC 1998)*, 20–30.

D. Aharonov & T. Naveh (2002). Quantum NP—A survey. Available at arXiv:quant-ph/0210077v1.

E. W. Allender & K. W. Wagner (1993). Counting hierarchies: Polynomial time and constant depth circuits. In *Current Trends in Theoretical Computer Science*, 469–483. World Scientific.

A. Ambainis (2014). On physical problems that are slightly more difficult than QMA. In *Proceedings of 29th IEEE Conference on Computational Complexity (CCC 2014)*, 32–43.

S. Beigi (2010). NP vs $\mathrm{QMA}_{\log}(2)$. *Quantum Information and Computation (QIC)* **10**(1–2), 141–151.

H. Blier & A. Tapp (2009). All languages in NP have very short quantum proofs. In *Proceedings of the 3rd International Conference on Quantum, Nano and Micro Technologies*, 34–37.

S. Boyd & L. Vandenberghe (2004). *Convex Optimization*. Cambridge University Press. ISBN 9780521833783.

C. M. Dawson & M. A. Nielsen (2006). The Solovay-Kitaev algorithm. *Quantum Information and Computation (QIC)* **6**(1).

A. Deshpande, A. V. Gorshkov & B. Fefferman (2022). The importance of the spectral gap in estimating ground-state energies. In *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*, M. Braverman, editor, volume 215 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 54:1–54:6. Schloss Dagstuhl—Leibniz-Zentrum für Informatik, Dagstuhl, Germany.

S. Fenner, L. Fortnow, S. A. Kurtz & L. Li (2003). An oracle builder's toolkit. *Information and Computation* **182**(2), 95–136.

L. Fortnow (1994). The Role of Relativization in Complexity Theory. *Bulletin of the European Association for Theoretical Computer Science* **52**, 52–229.

L. Fortnow & J. Rogers (1999). Complexity limitations on quantum computation. *Journal of Computer and System Sciences* **59**(2), 240–252.

S. Gharibian & J. Kempe (2012). Hardness of approximation for quantum problems. In *Automata, Languages, and Programming (ICALP 2012)*, A. Czumaj, K. Mehlhorn, A. Pitts & R. Wattenhofer, editors, 387–398. Springer, Berlin, Heidelberg.

S. Gharibian, M. Santha, J. Sikora, A. Sundaram & J. Yirka (2018). Quantum Generalizations of the Polynomial Hierarchy with Applications to QMA(2). In *43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018)*, I. Potapov, P. Spirakis & J. Worrell, editors, volume 117 of *Leibniz International Proceedings in Informatics (LIPIcs)*, 58:1–58:16. Schloss Dagstuhl—Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany.

S. Gharibian & J. Yirka (2019). The complexity of simulating local measurements on quantum systems. *Quantum* **3**, 189.

O. Goldreich (2006). On promise problems: A survey. In *Theoretical Computer Science: Essays in Memory of Shimon Even*, O. Goldreich, A.L. Rosenberg & A.L. Selman, editors, 254-290. Springer-Verlag, Berlin, Heidelberg.

A. B. Grilo, I. Kerenidis & J. Sikora (2016). QMA with subset state witnesses. *Chicago Journal of Theoretical Computer Science* **2016**(4).

M. Grötschel, L. Lovász & A. Schrijver (1993). *Geometric Algorithms and Combinatorial Optimization.* Springer-Verlag. ISBN 9783642782404.

R. A. Horn & C. H. Johnson (1990). *Matrix Analysis.* Cambridge University Press. ISBN 9780521548236.

R. Jain & J. Watrous (2009). Parallel approximation of non-interactive zero-sum quantum games. In *24th Annual IEEE Conference on Computational Complexity (CCC)*, 243–253.

S. P. Jordan, H. Kobayashi, D. Nagaj & H. Nishimura (2012). Achieving perfect completeness in classical-witness quantum Merlin-Arthur proof systems. *Quantum Information and Computation (QIC)* **12**(5–6), 461–471.

R. M. Karp & R. J. Lipton (1980). Some connections between nonuniform and uniform complexity classes. In *Proceedings of the Twelfth Annual ACM Symposium on Theory of Computing (STOC 1980)*, 302–309. ACM, New York, NY, USA.

A. Kitaev, A. Shen & M. Vyalyi (2002). *Classical and Quantum Computation.* American Mathematical Society. ISBN 082182161X.

A. Kitaev & J. Watrous (2000). Parallelization, amplification, and exponential time simulation of quantum interactive proof systems. In *Proceedings of the 32nd ACM Symposium on Theory of Computing (STOC 2000)*, 608–617.

H. Kobayashi, F. le Gall & H. Nishimura (2015). Stronger methods of making quantum interactive proofs perfectly complete. *SIAM Journal on Computing* **44**(2), 243–289.

Y.-K. Liu, M. Christandl & F. Verstraete (2007). Quantum Computational complexity of the N-representability problem: QMA complete. *Phys. Rev. Lett.* **98**, 110 503.

J. Lockhart & C. E. González-Guillén (2017). Quantum state isomorphism. Available at arXiv:1709.09622v1 [quant-ph].

C. Lund, L. Fortnow, H. Karloff & N. Nisan (1992). Algebraic methods for interactive proof systems. *Journal of the ACM* **39**(4), 859–868.

C. Marriott & J. Watrous (2005). Quantum Arthur-Merlin games. *Computational Complexity* **14**(2), 122–152.

A. Meyer & L. Stockmeyer (1972). The equivalence problem for regular expressions with squaring requires exponential space. In *13th Annual Symposium on Switching and Automata Theory*, 125–129.

A. Molina & J. Watrous (2019). Revisiting the simulation of quantum Turing machines by quantum circuits. *Proceedings of the Royal Society A* **475**(2226), 20180 767.

J. von Neumann (1928). Zur Theorie der Gesellschaftspiele. *Mathematische Annalen* **100**(1), 295–320.

A. Pereszlényi (2012). Multi-prover quantum Merlin-Arthur proof systems with small gap. Available at arXiv:1205.2761v1 [quant-ph].

Y. Shi (2003). Both Toffoli and controlled-NOT need little help to do universal quantum computing. *Quantum Information and Computation* **3**(1), 84–92.

S. Toda (1991). PP is as hard as the Polynomial-Time Hierarchy. *SIAM Journal on Computing* **20**, 865–877.

J. Torán (1991). Complexity classes defined by counting quantifiers. *Journal of the ACM* **38**(3), 752-773.

L. G. Valiant & V. V. Vazirani (1986). NP is as easy as detecting unique solutions. *Theoretical Computer Science* **47**, 85–93.

L. Vinkhuijzen (2018). *A quantum polynomial hierarchy and a simple proof of Vyalyi's Theorem*. Master's thesis, Leiden University.

N. V. Vinodchandran (2005). A note on the circuit complexity of PP. *Theoretical Computer Science* **347**(1-2), 415–418.

M. Vyalyi (2003). QMA=PP implies that PP contains PH. Technical Report TR03-021, Electronic Colloquium on Computational Complexity.

J. Watrous (2009a). Quantum computational complexity. In *Encyclopedia of Complexity and Systems Science*, R. A. Meyers, editor, 7174–7201. Springer, New York, NY.

J. Watrous (2009b). Semidefinite programs for completely bounded norms. *Theory of Computation* **5**, 217–238.

J. Watrous (2009c). Zero-knowledge against quantum attacks. *SIAM Journal of Computing* **39**(1), 25–58.

C. Wrathall (1976). Complete sets and the Polynomial-Time Hierarchy. *Theoretical Computer Science* **3**(1), 23 – 33.

T. Yamakami (2002). Quantum NP and a quantum hierarchy. In *Foundations of Information Technology in the Era of Network and Mobile Computing (TCS 2002)*, R. Baeza-Yates, U. Montanari & N. Santoro, editors, 323–336. Springer, Boston, MA.

Manuscript received 4 February 2021

Sevag Gharibian
Paderborn University
Paderborn, North Rhine-Westphalia, Germany

and
Virginia Commonwealth University
Richmond, VA, USA

Miklos Santha
CNRS, IRIF
Université de Paris
France

and
Centre for Quantum Technologies
National University of Singapore
Singapore, Singapore

Jamie Sikora
Virginia Polytechnic Institute and State University
Blacksburg, VA, USA

and
Perimeter Institute for Theoretical Physics
Waterloo, ON, Canada

Aarthi Sundaram
Microsoft Quantum
Redmond, WA, USA

Justin Yirka
The University of Texas at Austin
Austin, TX, USA
`yirka@utexas.edu`