Oracle Separation of QMA and QCMA with Bounded Adaptivity

Shalev Ben-David
Institute for Quantum Computing, University of Waterloo, Canada

Srijita Kundu
Institute for Quantum Computing, University of Waterloo, Canada

Abstract

We give an oracle separation between QMA and QCMA for quantum algorithms that have bounded adaptivity in their oracle queries; that is, the number of rounds of oracle calls is small, though each round may involve polynomially many queries in parallel. Our oracle construction is a simplified version of the construction used recently by Li, Liu, Pelecanos, and Yamakawa (2023), who showed an oracle separation between QMA and QCMA when the quantum algorithms are only allowed to access the oracle classically. To prove our results, we introduce a property of relations called *slipperiness*, which may be useful for getting a fully general classical oracle separation between QMA and QCMA.

2012 ACM Subject Classification Theory of computation → Quantum complexity theory

Keywords and phrases Quantum computing, computational complexity

Digital Object Identifier 10.4230/LIPIcs.ICALP.2024.21

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: https://arxiv.org/abs/2402.00298

Funding *Shalev Ben-David*: Supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), DGECR-2019-00027 and RGPIN-2019-04804[1].
Srijita Kundu: Supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grants Program, and Fujitsu Labs America.

1 Introduction

It is a long-standing open problem in quantum complexity theory whether the two possible quantum analogs of the complexity class NP are equivalent. QMA is defined as the class of decision problems that are solvable by a polynomial-time quantum algorithm that has access to a polynomial-sized *quantum* witness, whereas QCMA is the class of decision problems that are solvable by a polynomial-time quantum algorithm that only has access to the polynomial-sized *classical* witness. In other words, the question asks: are quantum proofs more powerful than classical proofs?

While the inclusion QCMA $\subseteq$ QMA is easy to see, the question of whether these two classes are actually equal, which was first posed by Aharonov and Naveh [3], remains unanswered. Indeed, an unconditional separation between these classes is beyond currently known techniques.

An easier, but still unsolved, problem is to show an oracle separation between QMA and QCMA. This is because oracle separations in the Turing machine model can be shown by means of separations in the much simpler model of *query complexity*, where similar

[1] Cette recherche a été financée par le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG), DGECR-2019-00027 et RGPIN-2019-04804.

separations between complexity classes are routinely shown (for example, a recent oracle separation between BQP and PH was provided in [12]). The problem of finding an oracle separation between QMA and QCMA has been a longstanding focus of the quantum computing community; it boils down to asking whether quantum proofs are more powerful than classical proofs in the query model.

1.1 Previous work

The first progress on the question of an oracle separation of QMA and QCMA was made by Aaronson and Kuperberg [2], who showed that there is a quantum oracle, i.e., a blackbox unitary, relative to which QMA $\neq$ QCMA. Later, Fefferman and Kimmel [7] showed that the separation also holds under what they called an "in-place permutation oracle", which is still inherently quantum. Ideally, we would like to get these separations in the standard model of classical oracles: classical functions that a quantum algorithm may query in superposition. [4] showed separations between QMA and QCMA in other non-standard oracle models.

Very recently, there has been some progress on this question, with two different variations of the standard classical oracle model. Natarajan and Nirkhe [11] showed an oracle separation relative to a "distributional oracle". This essentially means that the classical oracle is drawn from a distribution, which the prover knows, but the specific instance drawn is not known to the prover. Therefore, the witness only depends on the distribution over the oracles, which makes it easier to show QCMA lower bounds. Following this, [9] showed a separation with a classical oracle that is fully known to the prover, but assuming the verifier is only allowed to access this classical oracle classically, i.e., the verifier is not allowed to make superposition queries (this makes the class similar to MA in terms of its query power and witness type). This model is also simpler to analyze because whatever information the verifier gets from the oracle by classically querying it, could also have been provided as the classical QCMA witness. [9] also gave an alternate construction of a distributional oracle separation, with a simpler proof than [11]. Their constructions are based on the relational problem used by Yamakawa and Zhandry [14], in their result on quantum advantage without structure.

Closely related to the QMA vs QCMA question is the $\mathsf{BQP}_{/\mathrm{qpoly}}$ vs $\mathsf{BQP}_{/\mathrm{poly}}$ question. $\mathsf{BQP}_{/\mathrm{qpoly}}$ is the class of decision problems that are solvable by a polynomial-time quantum algorithm with access to polynomial-sized quantum *advice*, which depends non-uniformly on the length of inputs, but nothing else. $\mathsf{BQP}_{/\mathrm{qpoly}}$ is the class of decision problems solvable by a polynomial-time quantum algorithm with access to polynomial-sized classical advice. Most works which have found oracle separations for QMA vs QCMA in various oracle models, such as [2, 11, 9], have also found oracle separations between $\mathsf{BQP}_{/\mathrm{qpoly}}$ and $\mathsf{BQP}_{/\mathrm{poly}}$ with related constructions in the same oracle models.

The question of the relative power of classical vs quantum advice was recently resolved unconditionally (without oracles) for relational problems by Aaronson, Buhrman and Kretschmer [1], who showed an unconditional separation between $\mathsf{FBQP}_{/\mathrm{qpoly}}$ and $\mathsf{FBQP}_{/\mathrm{poly}}$. $\mathsf{FBQP}_{/\mathrm{qpoly}}$ and $\mathsf{FBQP}_{/\mathrm{poly}}$ are the classes of relational problems analogous to $\mathsf{BQP}_{/\mathrm{qpoly}}$ and $\mathsf{BQP}_{/\mathrm{poly}}$ respectively. Their result was based on observing that separations between quantum and classical one-way communication complexity can be used to show separations between classical and quantum advice. The reason their result only works for the relation classes is that a separation in one-way communication complexity which satisfies the necessary conditions can only hold for relational problems. The specific relational problem used in [1] is known as the Hidden Matching problem. But as was observed in [9], the Yamakawa-Zhandry problem [14] also achieves the required communication separation, and could have been used instead. In light of this, the constructions in [14] can viewed as a way to convert relational

separations in one-way communication complexity, which correspond to relational separations for quantum vs classical advice, to separations for decision QMA vs QCMA, and $\mathsf{BQP}_{/\mathrm{qpoly}}$ vs $\mathsf{BQP}_{/\mathrm{poly}}$, relative to classically accessible oracles. The construction is not blackbox – it does not work if the Hidden Matching Problem is used instead of the Yamakawa-Zhandry problem, though it plausibly might work with a parallel repetition of the former.

1.2 Our results

Unlike previous work, we prove an oracle separation between QMA and QCMA relative to a *bona fide* regular oracle with regular (quantum) queries. Our catch is, instead, that we only allow the algorithms *bounded adaptivity*.

Bounded adaptivity means that the number of rounds of queries made by the algorithms is small, although there can be polynomially many queries in each round. Although our result is not formally stronger than those of [11] and [9], we feel our result is intuitively closer to a full QMA-QCMA separation, as it allows the full power of classical proofs and some of the power of quantum queries. Our main result is formally stated below.

▶ **Theorem 1.** *There is an oracle* $\mathcal{O}\colon \{0,1\}^* \to \{0,1\}$ *such that* $\mathsf{QCMA}^{\mathcal{O},r} \neq \mathsf{QMA}^{\mathcal{O},r}$*, for* $r = o(\log n/\log\log n)$.

In the above statement, $\mathsf{QMA}^{\mathcal{O},r}$ is the class of decision problems solvable by QMA algorithms that have oracle access to $\mathcal{O}$, and make at most r batches of parallel queries to $\mathcal{O}$; $\mathsf{QCMA}^{\mathcal{O},r}$ is defined analogously. The parameter n is the efficiency parameter (so the number of queries is $\mathrm{poly}(n)$).

▶ **Theorem 2.** *There is a function family* $F = \{F_N\}_{N\in I}$ *which is efficiently computable in 1-round query* QMA*, but with the property that the growth rate of* $\mathrm{QCMA}^r(F_N)$ *for* $r = o(\log\log N/\log\log\log N)$ *as* $N \to \infty$ *is not in* $O(\mathrm{polylog}(N))$.

We shall formally define the query versions of QMA and QCMA, and the r-round QCMA query complexity QCMA^r, which are used in this theorem statement, in Section 2.1.

Our construction for the query complexity separation is a somewhat simplified version of the construction in [9], which is based on the Yamakawa-Zhandry problem. [14] and [10] showed that there exists a relational problem R_f, indexed by functions $f : [n] \times \{0,1\}^m \to \{0,1\}$, for $m = \Theta(n)$, such that given oracle access to a quantum advice $|z_f\rangle$, a quantum algorithm on any input $x \in \{0,1\}^n$, and on average over f, can find a u such that $(x,u) \in R_f$[2]. On the other hand, no quantum algorithm can find such an u for most x, when given only a classical advice z_f, and classical query access to f. Using this relation R_f, for a subset $E \subseteq \{0,1\}^n$, we construct the following oracle:

$$O[f,E](x,u) = \begin{cases} 1 & \text{if } (x,u) \in R_f \wedge x \notin E \\ 0 & \text{otherwise.} \end{cases}$$

The 1-instances of the problem F_N that will separate QMA and QCMA in the query complexity model will be $O[f,\emptyset]$, and the 0-instances will be $O[f,E]$ for $|E| \geq \frac{2}{3}\cdot 2^n$, for a large subset of all functions f. This is essentially the same construction that is used in [9], except they also use an additional oracle G for a random function from $\{0,1\}^n$ to $\{0,1\}^n$, which O also depends on.

[2] The Yamakawa-Zhandry relation is a TFNP relation, which means that the u-s are of $\mathrm{poly}(n)$ length, and a u such that $(x,u) \in R_f$ exists for every x.

Note that the query complexity lower bound we obtain for QCMA is of a different nature than the one obtained in [9]: we need to lower bound (bounded-round) quantum query algorithms instead of only classical query algorithms, and we focus on the worst-case rather than average-case setting. In order to get an oracle separation for Turing machines from a separation in query complexity, one needs to use a diagonalization argument; because our result is set up a bit differently than in previous work, we reprove the diagonalization argument for our setting in. This can be found in Appendix A of the full version of this paper on arXiv.

Finally, we emphasize that the bounded adaptivity limitation of our result is because we allow the full power of classical proofs and also quantum queries. If one were to drop the power of classical proofs (resulting in the class BQP), or if one were to drop the power of quantum queries (resulting in, essentially, MA), it would follow from [9] that close variants of F_N cannot be solved even without the bounded-round restriction. We conjecture their lower bounds apply to F_N as well.

1.3 Our techniques

We briefly describe the techniques used to obtain the query complexity result. We start by observing that the oracle $O[f, \emptyset]$ is essentially just a verification oracle for the Yamakawa-Zhandry relation. Therefore, there is a quantum witness and a quantum algorithm that can distinguish $O[f, \emptyset]$ and $O[f, E]$ by using this witness, with only one query, with probability $1 - 2^{-\Omega(n)}$ over f. The witness for the yes instance $O[f, \emptyset]$ is simply the quantum advice for the Yamakawa-Zhandry problem, which finds a u for any x with probability $1 - 2^{-\Omega(n)}$ over f. The quantum algorithm finds a u for a random x using the witness, and queries the oracle. Since the no instances return 0 on any (x, u) for most x, this algorithm can distinguish $O[f, \emptyset]$ and $O[f, E]$ for $1 - 2^{-\Omega(n)}$ fraction of the f-s.

We now consider the uniform distribution over these good f-s (for which we can distinguish $O[f, \emptyset]$ and $O[f, E]$ with quantum advice), which has $2^{\Omega(n)}$ min-entropy. If there was a classical witness function depending on f, of size k, that made a quantum algorithm accept $O[f, \emptyset]$ for these f-s, then there would exist a fixed witness string w that would make $O[f, \emptyset]$ accept for 2^{-k} fraction of f-s. The quantum algorithm depends on the witness, but if we fix the witness string w, the algorithm is fixed, and we can then ignore the dependence of the algorithm on the witness.

We now attempt to remove rounds of the quantum query algorithm, starting with the first round, while keeping the behavior of the algorithm the same on as many oracles as possible. Every time we remove a round, we restrict our attention to a smaller set of oracles, all of which are consistent with a growing partial assignment we assume is given to us. At the end, the quantum algorithm will have no rounds left, and hence will make no queries; we want the set of oracles $O[f, \emptyset]$ on which the behavior is preserved to be non-empty, because then we can conclude that the algorithm cannot distinguish $O[f, \emptyset]$ and $O[f, E]$ for some large erased set E (since it now makes no queries).

To remove the first round of the query algorithm, we start by considering the the uniform distribution over the 2^{-k} fraction of good f-s such that $O[f, \emptyset]$ is accepted by w. This distribution has $2^{\Omega(n)} - k$ min-entropy, and therefore, by a result of [8, 6], it can be written as a convex combination of finitely many dense distributions. Dense distributions are a concept that was first introduced in the context of communication complexity: in a dense distribution, some coordinates are fixed, and the rest of the coordinates have high min-entropy in every subset. In fact we will not need the full convex combination of dense distributions – we restrict our attention to one such distribution in the convex combination, and try to preserve the behavior of the quantum algorithm only within a subset of its support.

Since some coordinates are fixed in our dense distribution, the probability over this distribution of the event $(x,u) \in R_f$ non-negligible, for some (x,u) pairs (this probability is exactly 2^{-n} over uniform f). The quantum algorithm can potentially learn a lot about f by querying the oracle $O[f,\emptyset]$ for these pairs. Therefore, we shall fix the coordinates of f that are fixed by (x,u) being in R_f. Here is where we use the fact that the Yamakawa-Zhandry relation is what we shall call *slippery*. This essentially means that given a small number of fixed coordinates for f, the number (x,u) pairs that have non-negligible probability is not too high, and the number of extra coordinates fixed by these (x,u) pairs being in R_f is also not too high. The Yamakawa-Zhandry relation being slippery essentially follows from it using a code that has good list recoverability properties. (The Hidden Matching relation, or its parallel repetition, are not slippery by this definition, and so our construction does not work with these.)

Using the slippery property, we can increase the size of the partial assignment by not too much, and via a hybrid-like argument [5], we can ensure that the first round of the quantum algorithm does not learn much from queries outside this partial assignment. We then restrict our attention to oracles consistent with this partial assignment; on those, we can simulate the first round of the algorithm without making real queries (we simply use the known partial assignment and guess "0" on the rest of the oracle positions, which are highly unlikely to be 1). This way, we get a quantum algorithm with one fewer round, which mimics the original algorithm on a small (but not too small) set of oracles.

Continuing this way, we eliminate all rounds of the algorithm while still maintaining a non-empty set of oracles on which the behavior is preserved. Each such oracle can be "erased", turning a 1-input into a 0-input, so we only need the final 0-round algorithm to preserve the behavior of the original algorithm on at least one input. Using this technique, we can handle up to $o(\log n/\log\log n)$ rounds of $O(\operatorname{poly} n)$ non-adaptive quantum queries each.

1.4 Discussion and further work

We expect our techniques for the QMA vs QCMA separation may also work for a $\mathsf{BQP}_{/\mathrm{qpoly}}$ vs $\mathsf{BQP}_{/\mathrm{poly}}$ separation with boundedly adaptive oracle queries, using the same problem that is described in [9]. Their oracle in the query complexity setting is given by a random function G, which the BQP algorithm has to compute given oracle access to

$$O[f,G](x,u) = \begin{cases} G(x) & \text{if } (x,u) \in R'_f \\ \perp & \text{otherwise,} \end{cases}$$

and a quantum or classical advice. Here R'_f is a modified 1-out-of-n version of the Yamakawa-Zhandry problem, which has better completeness properties, but is similar to the original problem otherwise. Clearly this problem can be solved in $\mathsf{BQP}_{/\mathrm{qpoly}}$ by using the quantum advice for the Yamakawa-Zhandry problem. It cannot be solved on input x with any classical advice and with access to an oracle that outputs $\perp$ for every (x,u). In order to show a $\mathsf{BQP}_{/\mathrm{poly}}$ lower bound for this problem, one needs that there exist many x-s such that a quantum algorithm with classical advice cannot distinguish $O[f,G]$ from a version of $O[f,G]$ that is erased on those x-s. Since $O[f,G]$ essentially serves as a verification oracle for the relation just as $O[f,\emptyset]$ does in the QMA vs QCMA construction, we expect that when the quantum algorithm has bounded rounds, a proof very similar to our QCMA lower bound will work.[3]

[3] R'_f, being a 1-out-of-n version of R_f, has worse slipperiness properties than R_f, which gets in the way of applying our techniques. But instead of using R'_f for better completeness, we can focus on the (large enough) subset of x, f for which the BQP algorithm with quantum advice works for R_f with high probability, and have $O[f,G]$ give $G(x)$ for free outside of this set. This would make the analysis very similar to the QMA vs QCMA case.

The final goal is, of course, to be able to show both these results without a bound on the number of rounds of oracle queries the quantum algorithm makes. As mentioned earlier, we fail to do this because the slipperiness parameters of the relation we picked are not good enough, and our methods would work to separate QMA and QCMA with an analogous problem definition where the Yamakawa-Zhandry relation is replaced by a different relation R_f that has the appropriate slipperiness property.

We now expand more on the required strong slipperiness property. Let R_f be a family of TFNP relations on $\{0,1\}^n \times \{0,1\}^m$ indexed by $f \in \{0,1\}^N$, where $m = \text{poly}(n)$ and $N = \Omega(2^n)$. We further assume R_f satisfies the property that if $(x,u) \in R_f$, then there is a polynomial-sized partial assignment p for f which certifies this, i.e., $(x,u) \in R_f \ \forall f \supseteq p$. Let $\mathcal{P} \subseteq \{0,1,*\}^N$ denote the set of polynomial-sized partial assignments for f. We define the extended version $\widetilde{R}$ of the family of relations R_f as follows:

$$\widetilde{R} = \{(p,x,u) : p \text{ is the minimal partial assignment s.t. } (x,u) \in R_f \, \forall f \supseteq p\}.$$

Since p is polynomial-sized, if we consider the uniform distribution over $\{0,1\}^N$, $\Pr[p \subseteq f]$ is exponentially small. Now consider a partial assignment q for f with size at most $s(n)$; we fix the bits in q and generate the other bits of f uniformly at random, which can make the probability of some other partial assignments p non-negligible. The slipperiness property is concerned with the total additional support (outside of q) of all partial assignments p such that $\Pr[p \subseteq f | q \subseteq f]$ is non-negligible, and $(p,x,u) \in \widetilde{R}$. We say $\widetilde{R}$ is $(\eta, s(n), t(n))$-slippery if for all $s(n)$-sized q, the total additional support of all p-s such that $\Pr[p \subseteq f | q \subseteq f] \geq \eta$ and $(p,x,u) \in \widetilde{R}$ is at most $t(n)$. See Definition 13 for a more formal definition.

Our techniques show that the following conjecture implies an oracle separation between QMA and QCMA.

▶ **Conjecture 3.** *There exists a family of* TFNP *relations R_f such that*

1. *There exists a polynomial-time algorithm $\mathcal{A}$, and for each f, a* poly(n)*-sized quantum state $|z_f\rangle$ such that, given access to x and $|z_f\rangle$, $\mathcal{A}$ can find u such that $(x,u) \in R_f$, with probability at least $1 - 2^{-\Omega(n)}$ over a product distribution $\mu_X \mu_F$ on x, f. Moreover, μ_X and μ_F are required to respectively have min-entropy $2^{\Omega(n)}$ and $\Omega(n)$.*
2. *There exists a function $s(n) = 2^{o(n)}$ such that for all polynomial functions $p(n)$, the extended relation $\widetilde{R}$ is $(1/p(n), s(n), t(n))$-slippery for some $t(n)$ such that $\log(t(n)) = o(\log(s(n)))$.*

Assuming Conjecture 3 is true, the oracle function separating QMA and QCMA would be distinguishing $O[f, \emptyset]$ and $O[f, E]$, for $|E| \geq \frac{2}{3} \cdot 2^n$, which we have defined earlier, using a relation R_f that satisfies the conjecture. (We can only prove the Yamakawa-Zhandry relation is $(\eta, s(n), t(n))$-slippery, with $t(n)$ bigger than $s(n)$, though it is possible that is satisfies the conjecture under finer analysis.)

We further note that any family of relations R_f that satisfies Conjecture 3 must give an exponential separation between quantum and randomized one-way communication complexity, with the communication setting being that Alice gets input f, Bob gets input x, and Bob has to output u such that $(x,u) \in R_f$.[4] This is because, if there was a polynomial-sized classical message w_f that Alice could send to Bob in the communication setting, then w_f could also serve as a QCMA proof. Therefore, it seems that the slipperiness condition could also be used for lower-bounding one-way randomized communication complexity (although weaker slipperiness parameters than in the conjecture would also suffice for this).

[4] Strictly speaking, condition 1 of the conjecture only implies that there exists a one-way communication protocol, in which Alice sends the state $|z_f\rangle$, which works on average over x and f, whereas we usually require worst-case success in communication complexity. However, we can restrict to the set of x and f for which the algorithm $\mathcal{A}$ works, in order to get the communication problem.

2 Preliminaries

2.1 QMA and QCMA in query complexity

In this section, we review the formal definitions of QMA, QCMA, computationally-efficient QMA, and bounded-round QCMA in the context of query complexity.

▶ **Definition 4** (Bounded-round quantum query algorithm). *For $r, T, n \in \mathbb{N}$, give the following definition of a quantum query algorithm Q acting on n bits, using r rounds, with T queries in each round. The algorithm will be a tuple of $r+1$ unitary matrices, $Q = (U_0, U_1, \ldots, U_r)$. These unitary matrices will each act on T "query-input" registers of dimension n, T "query-output" registers of dimension 2, an "output" register of dimension 2, and a work register of arbitrary dimension.*

For each $x \in \{0,1\}^n$, let U^x be the oracle unitary, which acts on the query-input and query-output registers by mapping

$$|i_1\rangle |b_1\rangle |i_2\rangle |b_2\rangle \ldots |i_T\rangle |b_T\rangle \to |i_1\rangle |b_1 \oplus x_{i_1}\rangle |i_2\rangle |b_2 \oplus x_{i_2}\rangle \ldots |i_T\rangle |b_T \oplus x_{i_T}\rangle$$

for all $i_1, \ldots, i_T \in [n]$ and all $b_1, \ldots, b_T \in \{0,1\}$. We extend U^x to other registers via a Kronecker product with identity, so that U^x ignores the other registers.

The action of the algorithm Q on input $x \in \{0,1\}^n$, denoted by the Bernoulli random variable $Q(x)$, will be the result of measuring the output register of the state

$$U_r U^x U_{r-1} U^x \ldots U^x U_1 U^x U_0 |\psi_{init}\rangle ,$$

where $|\psi_{init}\rangle$ is a fixed initial state.

We will use the term "T-query quantum algorithm" without referring to the number of rounds to indicate T rounds with 1 query in each.

▶ **Definition 5** (Query algorithm with witness). *Let Q be a r-query quantum algorithm on n bits with T queries in each round. For any quantum state $|\phi\rangle$ and any $x \in \{0,1\}^n$, let $Q(x, |\phi\rangle)$ be the random variable corresponding to the measured output register after the algorithm terminates, assuming the initial state contained $|\phi\rangle$ in the work register (with ancilla padding) instead of being $|\psi_{init}\rangle$. That is, $Q(x, |\phi\rangle)$ is a Bernoulli random variable corresponding to the measurement outcome of the output register of the final state*

$$U_r U^x U_{r-1} U^x \ldots U_1 U^x U_0 |\phi\rangle |pad\rangle ,$$

where $|pad\rangle$ is the ancilla padding.

▶ **Definition 6** (Query QMA and QCMA). *Let f be a possibly partial Boolean function on n bits, and let Q be a quantum query algorithm on n bits with T total queries. We say that Q is a QMA algorithm for f with witness size k if the following holds:*

1. (Soundness). *For every $x \in f^{-1}(0)$ and every k-qubit state $|\phi\rangle$, we have $\Pr[Q(x, |\phi\rangle) = 1] \leq \epsilon$.*
2. (Completeness). *For every $x \in f^{-1}(1)$, there exists a k-qubit state $|\phi\rangle$ such that $\Pr[Q(x, |\phi\rangle) = 1] \geq 1 - \delta$.*

Here, ϵ and δ govern the soundness and completeness of Q; by default, we take them both to be $1/3$. We denote the QMA query complexity of f by $\mathrm{QMA}_{\epsilon,\delta}(f)$, which is the minimum possible value of $T + k$ over any QMA algorithm for f with the specified soundness and completeness.

We say that Q is a QCMA algorithm for f if the same conditions hold, except with the witness state $|\phi\rangle$ quantifying over only classical k-bit strings in both the soundness and completeness conditions. We define $\mathrm{QCMA}_{\epsilon,\delta}(f)$ *analogously to* $\mathrm{QMA}_{\epsilon,\delta}(f)$*, and we omit the subscripts when they are both* $1/3$*.*

▶ **Definition 7** (Bounded round query QMA and QCMA)**.** *We define r-round QMA and QCMA in exactly the same way as the above definition, except the query algorithms are required to have at most r rounds. We use* $\mathrm{QMA}^r_{\varepsilon,\delta}(f)$ *and* $\mathrm{QCMA}^r_{\varepsilon,\delta}(f)$ *to denote the r-round QMA and QCMA query complexities of f respectively.*

▶ **Definition 8** (Function family)**.** *A* function family *is an indexed set $F = \{f_n\}_{n\in I}$ where $I \subseteq \mathbb{N}$ is an infinite set and where each f_n is a partial Boolean function $f_n\colon \mathrm{Dom}(f_n) \to \{0,1\}$ with $\mathrm{Dom}(f_n) \subseteq \{0,1\}^n$.*

▶ **Definition 9** (Efficiently computable QMA)**.** *Let $F = \{f_n\}_{n\in I}$ be a function family. We say that F is in* efficiently computable query QMA *if there is a polynomial-time Turing machine which takes in the binary encoding $\langle n\rangle$ of a number $n \in I$ and outputs a QMA verifier Q by explicitly writing out the unitaries of Q as quantum circuits (with a fixed universal gate set). The verifier Q must be sound and complete for f_n. Efficiently computable bounded-round QMA is defined analogously.*

In other words, $\mathrm{QMA}(f_n)$ *must be* $O(\mathrm{polylog}(n))$*, and moreover, the different algorithms for f_n must be uniformly generated by a single polynomial-time Turing machine.*

With these definitions, we show in the full version that Theorem 2 implies Theorem 1.

2.2 Error-correcting codes

A Reed-Solomon error-correcting code $\mathrm{RS}_{q,\gamma,k}$ over $\mathbb{F}_q$, with degree parameter $0 < k < q-1$ and generator $\gamma \in \mathbb{F}_q^*$, is defined as

$$\mathrm{RS}_{q,\gamma,k} = \{(f(\gamma), \dots f(\gamma^q)) : f \in \mathbb{F}_q[x]_{\deg\le k}\},$$

where $\mathbb{F}_q[x]_{\deg\le k}$ is the set of polynomials over $\mathbb{F}_q$ of degree at most k.

Let $q-1 = mn$, for some integers m and n. The m-folded version $\mathrm{RS}^{(m)}_{q,\gamma,k}$ of $\mathrm{RS}_{q,\gamma,k}$ is a mapping of the code to the larger alphabet $\mathbb{F}_q^m$ as follows:

$$\mathrm{RS}^{(m)}_{q,\gamma,k} = \{((x_1,\dots,x_m),\dots,(x_{q-m},\dots,x_q)) : (x_1,\dots,x_q) \in \mathrm{RS}_{q,\gamma,k}\}.$$

Note that the alphabet of $\mathrm{RS}^{(m)}_{q,\gamma,k}$ is $\mathbb{F}_q^m$.

▶ **Definition 10.** *We say that a code $C \subseteq \Sigma^n$ is combinatorially (ζ,ℓ,L)-list recoverable if for any subsets $S_i \subseteq \Sigma$ such that $|S_i| \le \ell$, we have,*

$$|\{(x_1,\dots,x_n) \in C : |\{i : x_i \in S_i\}| \ge (1-\zeta)n\}| \le L.$$

▶ **Lemma 11** ([13, 14])**.** *For a prime power q such that $mn = q-1$, any generator $\gamma \in \mathbb{F}_q^*$, and degree $k < q-1$, $\mathrm{RS}^{(m)}_{q,\gamma,k}$ is (ζ,ℓ,q^s)-list recoverable for some $s \le m$ if there exists an integer r such that the following inequalities hold:*

$$(1-\zeta)n(m-s+1) \ge \left(1+\frac{s}{r}\right)(mn\ell k^s)^{1/(s+1)} \tag{1}$$

$$(r+s)\left(\frac{mn\ell}{k}\right)^{1/(s+1)} < q. \tag{2}$$

▶ **Corollary 12.** *Let m be $\Theta(n)$ integer such that $nm+1=q$ is a prime power. Let $k=\frac{5}{6}mn$ and let c,d be constants. Then $\mathrm{RS}^{(m)}_{q,\gamma,k}$ is $(c\log n/n, 2^{(\log n)^d}, 2^{(\log n)^{d+2}})$-list recoverable.*

This corollary is proved simply by checking that the equations (1)–(2) are satisfied with this choice of parameters. The choice of parameters is in fact the same as those as [14]. Therefore, the above code satisfies the other conditions required for the [14] quantum algorithm to succeed in evaluating the relation $R_{C,f}$ defined in the next section.

3 The Yamakawa-Zhandry problem

For a function $f:[n]\times\{0,1\}^m\to\{0,1\}$ and a linear code $C\subseteq\{0,1\}^{nm}$, define the relation $R_{C,f}\subseteq\{0,1\}^n\times\{0,1\}^{nm}$

$$R_{C,f}=\{(x,u)=(x_1\dots x_n,u_1\dots u_n):(u_1\dots u_n\in C)\wedge(\forall i\, f(i,u_i)=x_i)\}.$$

We shall typically work with $m=\Theta(n)$. We shall usually work with a fixed code C, in which case we shall omit the subscript C from $R_{C,f}$.

Let $\mathcal{P}\subseteq\{0,1,*\}^{n2^m}$ denote the set of polynomial-sized partial assignments for functions $f:[n]\times\{0,1\}^m\to\{0,1\}$. We define the extended version $\widetilde{R}_C$ of $\{R_{C,f}\}_f$ over $\mathcal{P}\times\{0,1\}^n\times\{0,1\}^{nm}$ as follows:

$$\widetilde{R}_C=\{(p,x,u):p\text{ is the minimal partial assignment s.t. }(x,u)\in R_{C,f}\,\forall f\supseteq p\}.$$

In particular, (p,x,u) is in $\widetilde{R}_C$ when p is the partial assignment $(f(i,u_i)=x_i)_i$, which is n bits.

▶ **Definition 13.** *Let $\widetilde{R}_n$ be a sequence of relations on $\mathcal{P}_n\times\{0,1\}^n\times\{0,1\}^{\mathrm{poly}(n)}$, where $\mathcal{P}_n$ consists of fixed polynomial-sized partial assignments for $N=2^{\Omega(n)}$-bit strings, and $\mathrm{poly}(n)$ is some fixed polynomial. We say $\widetilde{R}_n$ is $(\eta,s(n),t(n))$-slippery w.r.t. distribution μ on f if for any partial assignment q on N bits with size at most $s(n)$, if we fix the bits of q in f and generate the other bits of f according to μ (conditioned on q), we will have*

$$\left|\bigcup_{\substack{(p,x,u)\in\widetilde{R}_n,\\ \Pr_{f\sim\mu}[p\subseteq f|q\subseteq f]\geq\eta}}\mathrm{supp}(p)\setminus\mathrm{supp}(q)\right|\leq t(n).$$

We omit mentioning the distribution μ explicitly if it is the uniform distribution.

▶ **Lemma 14.** *When C is a code with parameters from Corollary 12, then for $c=\mathrm{polylog}(n)$ and $d=o(\log n/\log\log n)$, $\widetilde{R}_C$ is $(\frac{1}{n^c},2^{(\log n)^d},c\log n\cdot 2^{(\log n)^{d+2}})$-slippery.*

Proof. Let q be a partial assignment of size $2^{(\log n)^d}$. For each $i\in[n]$, let $S_i=\{v:(i,v)\text{ is fixed in }q\}$. Clearly for each i, $|S_i|\leq 2^{(\log n)^d}$. By Corollary 12,

$$C_q=|\{u_1\dots u_n\in C:|\{i:u_i\in S_i\}|\geq n-c\log n\}|\leq 2^{(\log n)^{d+2}}.$$

A tuple (p,x,u) can satisfy $(p,x,u)\in\widetilde{R}_C$ and $\Pr_{f\sim U}[p\subseteq f|q\subseteq f]$ only if $u\in C_q$, so we only need to compute $|\bigcup\mathrm{supp}(p)|$ for such tuples. In fact we only need to worry about the number of (x,u) pairs that could be in $R_{C,f}$, since p is completely fixed by x and u. Each u has at most $c\log n$ many locations that are not fixed by q, and x can take any value in those $c\log n$ locations. The x-s taking different values in these locations have overlapping

p-s (i.e., the same bits are fixed to different values for the different x-s), and since we only care about $|\bigcup \mathrm{supp}(p)|$, we need not count these x-s separately. Therefore, the number of unique indices fixed by such p-s is determined only by the number of u-s in C_q.

Since the total support of each p is outside of q is $c \log n$, we have,

$$\left| \bigcup_{\substack{(p,x,u)\in \widetilde{R}_n, \\ \Pr_f[p\subseteq f | q \subseteq f] \geq \frac{1}{n^c}}} \mathrm{supp}(p) \setminus \mathrm{supp}(q) \right| \leq 2^{(\log n)^{d+2}} \cdot c \log n. \quad \blacktriangleleft$$

▶ **Corollary 15.** *If μ is a distribution such that for all partial assignments p with $|p| = n$, we have $\mu|_q[p] \leq k \cdot u|_q[p]$ (where $\mu|_q[p]$ is the probability mass of strings consistent with p under μ conditioned on q, and $u|_q[p]$ is the same with the uniform distribution), then $\widetilde{R}_C$ from Lemma 14 is also $(\frac{k}{n^c}, 2^{(\log n)^d}, c \log n \cdot 2^{(\log n)^{d+2}})$-slippery w.r.t. μ.*

Proof. Since $\mu[p] \leq k \cdot u|_q[p]$ for all p, partial assignments that have probability at least $\frac{k}{n^c}$ against μ conditioned on q have probability at least $\frac{1}{n^c}$ against the uniform distribution conditioned on q. Now we can apply Lemma 14. $\blacktriangleleft$

▶ **Theorem 16.** *There exists a code C such that*

1. *$\widetilde{R}_C$ is $(\frac{1}{n^c}, 2^{(\log n)^d}, c \log n \cdot 2^{(\log n)^{d+2}})$-slippery for $c = \mathrm{polylog}(n)$ and $d = o(\log n / \log\log n)$.*
2. *There exists a quantum advice $|z_f\rangle$ with polynomially many qubits, and a polynomial-time quantum algorithm $\mathcal{A}$ that has access to $|z_f\rangle, x$, and makes no queries to any oracle, such that for any $x \in \{0,1\}^n$,*

$$\Pr_{f\sim U}[(u \leftarrow \mathcal{A}(|z_f\rangle, x)) \wedge ((x,u) \in R_{C,f})] \geq 1 - 2^{-\Omega(n)},$$

where the probability is over uniformly random functions $f : [n] \times \{0,1\}^m \to \{0,1\}$, and the internal randomness of $\mathcal{A}$.

Proof. Item 1 is due to Lemma 14. As stated before, the problem $\widetilde{R}_C$, and the choice of parameters for the code C in Lemma 14, is the same as [14]. Therefore, item 2 is due to [14, 10].[5] $\blacktriangleleft$

4 Techniques for bounded-round quantum query algorithms

In this section, we prove some results about bounded-round quantum query algorithms that will be useful in proving our QCMA lower bound.

Recall that a non-adaptive quantum algorithm works on T query-input registers and T query-output registers plus an additional work register W, so that its basis states look like

$$|i_1\rangle |b_1\rangle |i_2\rangle |b_2\rangle \dots |i_T\rangle |b_T\rangle |W\rangle .$$

To clear up notational clutter, we will use $\vec{i} \in [N]^T$ to represent a tuple of T indices in $[N]$. Moreover, for a string $x \in \{0,1\}^N$ and for $\vec{i} \in [N]^T$, we will define $x_{\vec{i}} := (x_{\vec{i}_1}, x_{\vec{i}_2}, \dots, x_{\vec{i}_T})$.

[5] The quantum algorithm in [14] makes some non-adaptive quantum queries (not depending on x), and does not take an advice state. The modified algorithm, which instead takes an advice state (which is essentially the state of the algorithm in [14] after its non-adaptive queries) and makes no queries, was described in [10].

The basis states can then be written $|\vec{i}\rangle\,|\vec{b}\rangle\,|W\rangle$, and the action of the query unitary U^x to the string x is to map $|\vec{i}\rangle\,|\vec{b}\rangle\,|W\rangle \to |\vec{i}\rangle\,|\vec{b} \oplus x_{\vec{i}}\rangle\,|W\rangle$, extended linearly to the rest of the space. (Here $\oplus$ denotes the bitwise XOR of the two strings of length T.)

Define $\Pi_{\vec{i}} := |\vec{i}\rangle\langle\vec{i}| \otimes I_{\vec{b},W}$ to be the projection onto basis states with $\vec{i}$ in the query-input registers. For $i \in [N]$, define $\Pi_i := \sum_{\vec{i} \ni i} \Pi_{\vec{i}}$ to be the projection onto basis states with i occurring in one of the query-input registers. The projections $\Pi_{\vec{i}}$ are onto orthogonal spaces, though the projections Π_i are not. Observe that $\sum_{\vec{i}} \Pi_{\vec{i}} = I$, and that $\sum_i \Pi_i = \sum_i \sum_{\vec{i} \ni i} \Pi_{\vec{i}} = \sum_{\vec{i}} \sum_{i \in \vec{i}} \Pi_{\vec{i}} = T \cdot I$. Moreover, since the oracle unitary U^x does not change the query-input registers, U^x commutes with both $\Pi_{\vec{i}}$ and Π_i. Another convenient property is that if $x_{\vec{i}} = y_{\vec{i}}$ for two strings $x, y \in \{0,1\}^N$, then $\Pi_{\vec{i}}(U^x - U^y) = 0$; this holds because both U^x and U^y map $|\vec{i}\rangle\,|\vec{b}\rangle\,|W\rangle$ to the same vector when $x_{\vec{i}} = y_{\vec{i}}$. Using these properties, we have the following lemma.

▶ **Lemma 17** (Hybrid argument for nonadaptive queries). *For any strings $x, y \in \{0,1\}^N$ and any quantum state $|\psi\rangle = \sum_{\vec{i},\vec{b},W} \alpha_{\vec{i},\vec{b},W}\,|\vec{i}\rangle\,|\vec{b}\rangle\,|W\rangle$, we have*

$$\| U^x\,|\psi\rangle - U^y\,|\psi\rangle \,\|_2^2 \le 4 \sum_{i : x_i \neq y_i} \| \Pi_i\,|\psi\rangle \,\|_2^2.$$

Proof. We write the following, with justification afterwards.

$$\begin{aligned}
\| U^x\,|\psi\rangle - U^y\,|\psi\rangle \,\|_2^2 &= \left\| \sum_{\vec{i}} \Pi_{\vec{i}}(U^x - U^y)\,|\psi\rangle \right\|_2^2 \\
&= \sum_{\vec{i}} \| \Pi_{\vec{i}}(U^x - U^y)\,|\psi\rangle \,\|_2^2 \\
&= \sum_{\vec{i} : x_{\vec{i}} \neq y_{\vec{i}}} \| \Pi_{\vec{i}}(U^x - U^y)\,|\psi\rangle \,\|_2^2 \\
&\le \sum_{\vec{i}} \sum_{i \in \vec{i} : x_i \neq y_i} \| \Pi_{\vec{i}}(U^x - U^y)\,|\psi\rangle \,\|_2^2 \\
&= \sum_{i : x_i \neq y_i} \sum_{\vec{i} \ni i} \| \Pi_{\vec{i}}(U^x - U^y)\,|\psi\rangle \,\|_2^2 \\
&= \sum_{i : x_i \neq y_i} \left\| \sum_{\vec{i} \ni i} \Pi_{\vec{i}}(U^x - U^y)\,|\psi\rangle \right\|_2^2 \\
&= \sum_{i : x_i \neq y_i} \| \Pi_i(U^x - U^y)\,|\psi\rangle \,\|_2^2 \\
&= \sum_{i : x_i \neq y_i} \| (U^x - U^y)\Pi_i\,|\psi\rangle \,\|_2^2 \\
&\le 4 \sum_{i : x_i \neq y_i} \| \Pi_i\,|\psi\rangle \,\|_2^2.
\end{aligned}$$

In the first line, we used $\sum_{\vec{i}} \Pi_{\vec{i}} = I$. In the second, we used the orthogonality of the images of the projections $\Pi_{\vec{i}}$. In the third, we used $\Pi_{\vec{i}}(U^x - U^y) = 0$ when $x_{\vec{i}} = y_{\vec{i}}$.

In the fourth line, we replaced the sum over $\vec{i}$ containing at least one i with $x_i \neq y_i$ with a weighted sum, where the weight of $\vec{i}$ is the number of $i \in \vec{i}$ such that $x_i \neq y_i$; this weight is 0 when $x_{\vec{i}} = y_{\vec{i}}$ and at least 1 when $x_{\vec{i}} \neq y_{\vec{i}}$. This weight can be represented as a sum over $i \in \vec{i}$ with $x_i \neq y_i$, since we are counting $\vec{i}$ once for each such i in the tuple.

The fifth line flips the order of the sums, and the sixth uses orthogonality of the images of $\Pi_{\vec{i}}$ to put the sum back inside the squared norm. The seventh line is the definition of Π_i, and the eighth holds since Π_i commutes with U^x and U^y. Finally, the last line follows either from the triangle inequality, or from the fact that the spectral norm of $(U^x - U^y)$ is at most 2 (since U^x and U^y are unitary). ◀

For an oracle $x \in \{0,1\}^n$ and a block $B \subseteq [N]$, use $x[B]$ to denote the string x with queries in B erased; that is, $x[B]_i = x_i$ if $i \notin B$, and $x[B]_i = 0$ for $i \in B$. Next, we use this hybrid argument in combination with a Markov inequality to show that if a distribution μ over $\{0,1\}^n$ has a set of queries $B \in [N]$ that nearly always return zero for oracles sampled from μ, then for any non-adaptive quantum algorithm, there exists a large set of oracles (measured against μ) such that the algorithm does not detect whether any subset of B is erased.

▶ **Lemma 18** (Nonadaptive algorithms don't detect oracle erasures)**.** *Fix $|\psi\rangle$ representing the state of a quantum algorithm before a batch of non-adaptive queries. Let μ be a distribution over $\{0,1\}^N$, and let $\epsilon > 0$. Let $B = \{i \in [N] : \Pr_{x\sim\mu}[x_i = 1] \leq \epsilon\}$. Then there exists a set $S \subseteq \{0,1\}^N$ such that $\mu[S] \geq 1/2$ and for all $x \in S$ and all subsets $B_1, B_2 \subseteq B$, we have*

$$\| U^{x[B_1]} |\psi\rangle - U^{x[B_2]} |\psi\rangle \|_2 \leq \sqrt{8\epsilon T}.$$

Proof. We write the following, with justification afterwards.

$$\begin{aligned}
\mathop{\mathbb{E}}_{x\sim\mu}\left[\sum_{i:x_i\neq x[B]_i} \|\Pi_i |\psi\rangle\|_2^2\right] &= \mathop{\mathbb{E}}_{x\sim\mu}\left[\sum_{i\in B} x_i \|\Pi_i |\psi\rangle\|_2^2\right] \\
&= \sum_{i\in B} \|\Pi_i |\psi\rangle\|_2^2 \mathop{\mathbb{E}}_{x\sim\mu}[x_i] \\
&\leq \epsilon \sum_{i\in B} \|\Pi_i |\psi\rangle\|_2^2 \\
&\leq \epsilon \sum_{i\in[N]} \|\Pi_i |\psi\rangle\|_2^2 \\
&= \epsilon \sum_{i\in[N]} \sum_{\vec{i}\ni i} \|\Pi_{\vec{i}} |\psi\rangle\|_2^2 \\
&= \epsilon T \sum_{\vec{i}} \|\Pi_{\vec{i}} |\psi\rangle\|_2^2 \\
&= \epsilon T.
\end{aligned}$$

The first line follows by noting that $x_i \neq x[B]_i$ can only happen if both $i \in B$ and $x_i = 1$; we replace the sum over $i : x_i \neq x[B]_i$ with the sum over $i \in B$, and multiply the summand by the indicator for $x_i = 1$, which is x_i itself.

The second line is the result of pushing the expectation inside the sum, and observing that the norm does not depend on x and can be factored out of the expectation. The third line follows from the definition of B: we know that for all $i \in B$, the probability of $x_i = 1$ is at most ϵ. The fourth replaces the sum over B with that over $[N]$. The fifth uses the definition of Π_i, and exchanges the sum over $\vec{i}$ with the squared norm using orthogonality. The sixth line follows by noting that each $\vec{i}$ appears exactly T times in this double sum. Finally, the last line follows by pushing the sum inside the squared norm (using orthogonality), and recalling that $\sum_{\vec{i}} \Pi_{\vec{i}} = I$, together with the fact that $|\psi\rangle$ is a unit vector.

Given this bound on the expectation, we can apply Markov's inequality to conclude that at least half the strings x (weighted by μ) must satisfy $\sum_{i:x_i \neq x[B]_i} \|\Pi_i |\psi\rangle\|_2^2 \leq 2\epsilon T$. Let S be the set of such strings x; then $\mu[S] \geq 1/2$. Observe that for any $x \in S$ and any $B_1, B_2 \subseteq B$, the set $\{i : x[B_1]_i \neq x[B_2]_i\}$ is a subset of $\{i : x_i \neq x[B]_i\}$. We now apply Lemma 17 to get

$$\|U^{x[B_1]} |\psi\rangle - U^{x[B_2]} |\psi\rangle\|_2^2 \leq 4 \sum_{i:x[B_1]_i \neq x[B_2]_i} \|\Pi_i |\psi\rangle\|_2^2 \leq 4 \sum_{i:x_i \neq x[B]_i} \|\Pi_i |\psi\rangle\|_2^2 \leq 8\epsilon T.$$

The desired result follows by taking square roots. ◀

5 QMA vs QCMA

In this section, we prove Theorem 2. Theorem 19 will define the function F_N and show that it is in QMA, and Theorem 21 will show that it is not in QCMA.

5.1 Construction and QMA protocol

Fix a code C for which Theorem 16 holds, with $c = \log n$. We shall henceforth refer to $R_{C,f}$ as only R_f for this C. For a subset $E \subseteq \{0,1\}^n$, define the oracle $O[f, E]\colon \{0,1\}^n \times \{0,1\}^{nm} \to \{0,1\}$ as

$$O[f,E](x,u) = \begin{cases} 1 & \text{if } (x,u) \in R_f \wedge x \notin E \\ 0 & \text{otherwise.} \end{cases}$$

▶ **Theorem 19.** *There exists an efficient uniform collection of query QMA protocols (generated uniformly by a polynomial time Turing machine) which uses 1 query and polynomial witness size, and which outputs 0 on all oracles $O[f, E]$ with $|E| \geq (2/3) \cdot 2^n$, and outputs 1 on $O[f, \emptyset]$ for $1 - 2^{-\Omega(n)}$ fraction of f-s.*

Proof. The quantum witness for the algorithm will be quantum advice state for R_f from Theorem 16. The quantum algorithm works as follows: it samples a uniformly random $x \in \{0,1\}^n$, and runs the procedure from Theorem 16 to find a u such that $(x,u) \in R_f$. Note that this requires no queries to the oracle. Then it queries the oracle at (x,u) and returns the query output. If the oracle is $O[f,\emptyset]$ and the actual state $|z_f\rangle$ from Theorem 16 is provided as witness, then due to Theorem 16 we have,

$$\Pr_{f \sim U}[\mathcal{A}^{O[f,\emptyset]}(|z_f\rangle) = 1] \geq 1 - 2^{-\Omega(n)}.$$

On the other hand, if the oracle is $O[f,E]$ for $|E| \geq \frac{2}{3} \cdot 2^n$, no matter what witness is provided, and what u is obtained from this witness, the oracle outputs 0 on (x,u) for $\frac{2}{3}$ of the x-s. Since the algorithm samples a uniformly random x and queries it with some u for every f, we have for every f,

$$\Pr[\mathcal{A}^{O[f,E]}(|z_f\rangle) = 1] \leq \frac{1}{3}.$$ ◀

Defining the function F_N. We now define the following partial query function with input size $2^n \times 2^{mn}$: its 1-inputs are all the oracles $O[f,\emptyset]$ for which the algorithm from Theorem 19 accepts with probability at least $2/3$, and its 0-inputs are $O[f,E]$ for which $O[f,\emptyset]$ is a 1-input and $|E| \geq (2/3) \cdot 2^n$. Note that these oracles correspond to the inputs "x" of the query problem. This defines a family F_N of query tasks with $N = 2^n \times 2^{mn}$, and Theorem 19 showed that this family is in efficiently-computable QMA.

5.2 Densification of probability distributions

To prove our QCMA lower bound, we will need some properties of distributions on $\{0,1\}^N$. For such a distribution μ, let $\mathrm{RU}(\mu) \coloneqq \max_{x\in\{0,1\}^N} \log_2(2^N\mu[x])$ be the max relative entropy of μ relative to the uniform distribution. We will generally be interested in distributions μ such that $\mathrm{RU}(\mu)$ is small (say, $\mathrm{polylog}\, N$), which means that no input $x \in \{0,1\}^N$ has probability $\mu[x]$ much larger than 2^{-N}.

For a partial assignment p, let $\mu[p]$ be the probability mass of strings in $\{0,1\}^N$ which are consistent with p. Let $|p|$ be the size of p (the number of revealed bits in p). We define the density of μ to be $\mathrm{density}(\mu) \coloneqq 1 - \max_p \frac{\log_2(2^{|p|}\mu[p])}{|p|}$, with the maximum taken over partial assignments p. The density of the uniform distribution is 1.

For a partial assignment p, we let $\mu|_p$ denote the distribution μ conditioned on the sampled input being consistent with p. Items 1 and 3 of the following lemma essentially follow from results in [8, 6]. We produce a proof here because the version of the lemma we need is simpler than what was shown in [8, 6].

▶ **Lemma 20** (Densification). *Let μ be a distribution over $\{0,1\}^N$, and let $\delta \in (0,1)$. Then there exists a partial assignment p such that*

1. $|p| \le \mathrm{RU}(\mu)/\delta$
2. $\mathrm{RU}(\mu|_p) \le \mathrm{RU}(\mu)/\delta$
3. $\mathrm{density}(\mu|_p) > 1-\delta$, *where the density is measured on the bits not fixed by p.*

Proof. Let p be the largest partial assignment (we can pick the lexicographically first one according to some ordering, if there is a tie) for which $\mu[p] \ge 2^{-(1-\delta)|p|}$. Then

$$2^{-(1-\delta)|p|} \le \mu[p] = \sum_{x\supseteq p} \mu[x] \le 2^{N-|p|}\cdot 2^{-(N-\mathrm{RU}(\mu))} = 2^{\mathrm{RU}(\mu)-|p|},$$

so $\delta|p| \le \mathrm{RU}(\mu)$, from which the first item follows. Next,

$$\mathrm{RU}(\mu|_p) = \max_x \log_2(2^N\mu|_p[x]) = \max_{x\supseteq p} \log_2(2^N\mu[x]/\mu[p]) \le \mathrm{RU}(\mu) + \log_2(1/\mu[p])$$

$$\le \mathrm{RU}(\mu) + \log_2(2^{(1-\delta)|p|}) = \mathrm{RU}(\mu) + (1-\delta)|p| \le \mathrm{RU}(\mu) + (1-\delta)\,\mathrm{RU}(\mu)/\delta = \mathrm{RU}(\mu)/\delta,$$

which gives the second item. Finally, to upper bound the density of $\mu|_p$, let q be a partial assignment on a set of indices disjoint from that of p. By the maximality of p, we must have $\mu[p\cup q] < 2^{-(1-\delta)(|p|+|q|)}$. Now,

$$\log_2(2^{|q|}\mu|_p[q]) = \log_2(2^{|q|}\mu[q\cup p]/\mu[p]) < \log_2(2^{|q|}2^{-(1-\delta)(|p|+|q|)}/2^{-(1-\delta)|p|}) = \delta|q|.$$

From this it follows that $\mathrm{density}(\mu|_p) > 1-\delta$, as desired. ◀

5.3 QCMA lower bound

▶ **Theorem 21.** *There is no bounded-round, polynomial-cost QCMA protocol for the family F_N defined in Section 5.1. More formally, consider any family of QCMA protocols for the query problems F_N. If the number of rounds for these QCMA protocols grows slower than $o(\log\log N/\log\log\log N)$, then either the number of queries or the witness size must grow like $\log^{\omega(1)} N$.*

We will prove this theorem by a sequence of claims. The idea of the proof will be to remove the rounds of the algorithm one by one. We start by moving from QCMA to BQP via the following claim.

▷ Claim 22. If there is a QCMA protocol for F_N with witness size $k = k(N)$, then there is a quantum algorithm Q and a large set of functions S such that Q accepts all oracles $O[f, \emptyset]$ for $f \in S$ and rejects all oracles $O[f, E]$ for $f \in S$ and $|E| \geq (2/3)2^n$. The set S is large enough that the uniform distribution μ over S has $\mathrm{RU}(\mu) \leq 2k$. The algorithm Q makes the same number of rounds and number of queries as the QCMA protocol. By "accepting" and "rejecting", we mean with probability at least 2/3.

Proof. The idea is just to take a witness w that works for as many 1-inputs as possible, and hard-code this witness into the quantum algorithm. S will correspond to the set of 1-inputs on which this witness works.

More explicitly, since the witness is a classical string, there are only 2^k witnesses over which we quantify. Since each 1-input $O[f, \emptyset]$ has some witness accepting it, we conclude that at least one witness w of size k is a valid witness for at least a 2^{-k} fraction of the 1-inputs, and hence also for at least a $2^{-k}(1 - 2^{-\Omega(n)})$ fraction of all oracles $O[f, \emptyset]$ (including those not in the domain of F_N). This is because the fraction of f-s for which the quantum algorithm does not succeed with probability at least 2/3 is at most $2^{-\Omega(n)}$. We can assume $2^{-k}(1 - 2^{-\Omega(n)}) \geq 2^{-2k}$.

Let S be the set of f such that $O[f, \emptyset]$ is accepted by the algorithm given witness w. Let μ be the uniform distribution over S, and observe that $\mathrm{RU}(\mu) \leq 2k$. Let Q be the quantum algorithm which hard-codes the witness w into the verifier; then Q accepts all oracles $O[f, \emptyset]$ for $f \in \mathrm{supp}(\mu)$ and rejects all oracles $O[f, E]$ if $|E| \geq (2/3)2^n$. ◁

Defining the round reduction. Given a pair (Q, μ) of a quantum algorithm and a distribution over functions, we wish to define a pair $(\tilde{Q}, \tilde{\mu})$ such that $\tilde{Q}$ has one less round than Q, $\mathrm{supp}(\tilde{\mu})$ is a subset of $\mathrm{supp}(\mu)$ but "not by much" (i.e. $\mathrm{RU}(\tilde{\mu})$ is not much larger than $\mathrm{RU}(\mu)$), and the two algorithms behave similarly on $\tilde{\mu}$.

To define $(\tilde{Q}, \tilde{\mu})$ given (Q, μ), we proceed in several steps.

1. First, use Lemma 20 with $\delta = 1/n$ to find a partial assignment q with $|q| \leq n\,\mathrm{RU}(\mu)$, $\mathrm{RU}(\mu|_q) \leq n\,\mathrm{RU}(\mu)$, and with $\mu|_q$ being $(1 - \delta)$-dense on the bits not used by q.
2. Second, use Lemma 18 with $\epsilon = 1/3200r^2T$ on the distributions of oracles $O[f, \emptyset]$ when f is sampled from $\mu|_q$. The state $|\psi\rangle$ in the lemma will be the state of the algorithm Q just before the first batch of T queries. The lemma gives a set $S \subseteq \mathrm{supp}(\mu|_q)$ with $\mu|_q[S] \geq 1/2$. It has the property that for all $f \in S$ and all sets B_1, B_2 containing pairs (x, u) with $\Pr_{f \sim \mu|_q}[O[f, \emptyset](x, u) = 1] \leq \epsilon$, we have $\| U^{O[f,B_1]} |\psi\rangle - U^{O[f,B_2]} |\psi\rangle \|_2 \leq 1/20r$. Condition $\mu|_q$ on the set S to get a distribution μ'.
   Note that $O[f, B_1]$ is an abuse of notation, since normally we erase inputs x to f from the oracle, yet B_1 is a set of pairs (x, u). We will use this abuse of notation throughout; if we write $O[f, B]$ where B is a set of pairs, we mean to erase those pairs from the oracle, while if B is a subset of $\mathrm{Dom}(f)$, we mean to erase the pairs (x, u) for $x \in B$ and all u from the oracle.
3. Third, use the slippery property from Corollary 15 on q to conclude that the number of bits used by partial assignments p for which $(p, x, u) \in \tilde{R}_C$ and $\Pr_{f \sim \mu'}[p \subseteq f | q \subseteq f] \geq \epsilon/4$ is small. Recall that $(p, x, u) \in \tilde{R}_C$ means that the condition $O[f, \emptyset](x, u) = 1$ is equivalent to $p \subseteq f$ for all f; such certifying p have $|p| = n$. Corollary 15 can be applied because $\epsilon/4$ is larger than $1/n^c$ for $c = \log n$, since we are choosing $r = o(\log n / \log\log n)$ and $T \leq O(2^{\log^2 n} / \log n)$. Now, since $\mu|_q$ is $(1 - \delta)$-dense outside of q, the probability of a partial assignment p against $\mu|_q$ is at most $2^{\delta|p|}$ times the probability against the uniform distribution conditioned on q. Here $|p| = n$ and $\delta = 1/n$, so the probability against

$\mu|_q$ is at most twice that against the uniform distribution conditioned on q. Moving from $\mu|_q$ to μ' conditions on a set S of probability at least $1/2$, so it can increase the probability of p by at most a factor of 2. Hence the probability of p against μ' is overall at most 4 times its probability against the uniform distribution conditioned on q. By Corollary 15, we conclude the total number of bits used by partial assignments p for which $\Pr_{f\sim\mu'}[O[f,\emptyset](x,u)=1] \geq \epsilon$ is small. Let Z be the set of all such bits.

Our final modification to μ' will be to fix the bits in Z to the highest-probability partial assignment (measured against μ'), and let $\tilde{\mu}$ be μ' conditioned on this partial assignment.

4. Set $\tilde{Q}$ to be the quantum algorithm which is the same as Q, except that the first batch of queries is made to a fake oracle instead of a real one. The fake oracle is defined as follows: on queries (x,u) for which $O[f,\emptyset](x,u)$ is fixed for all $f \in \mathrm{supp}(\tilde{\mu})$, return this value $O[f,\emptyset](x,u)$; on queries (x,u) for which this value is not fixed for $f \in \mathrm{supp}(\tilde{\mu})$, return 0. Note that the fake oracle does not depend on the true input oracle $O[f,\emptyset]$, so queries to it can be implemented by $\tilde{Q}$ without making queries to the real oracle. This replaces the first round of Q, so $\tilde{Q}$ has one less round.

▷ Claim 23. Let Q and μ be as in Claim 22 (with μ the uniform distribution over S). Let $(Q_0,\mu_0)=(Q,\mu)$, and iteratively define $(Q_\ell,\mu_\ell)=(\tilde{Q}_{\ell-1},\tilde{\mu}_{\ell-1})$ for $\ell=1,2,\ldots,r$, where r is the number of rounds of Q.

Then Q_r makes no queries and μ_r "has large support": $\log \mathrm{RU}(\mu_r) \leq (2\log n)^{2r}\log 2k$ (assuming n is sufficiently large).

Proof. That Q_r makes no queries is clear, since each Q_ℓ in the chain makes one less round of queries than $Q_{\ell-1}$, and since the first algorithm $Q_0=Q$ makes r rounds.

To bound $\mathrm{RU}(\mu_r)$, we need to show that $\log \mathrm{RU}(\mu_{\ell+1})$ is at most a factor of $2\log^2 n$ more than $\log \mathrm{RU}(\mu_\ell)$.

Recall the construction of $\mu_{\ell+1}$ from μ_ℓ. The first step moved from μ_ℓ to $\mu_\ell|_q$ with $\mathrm{RU}(\mu_\ell|_q) \leq n\,\mathrm{RU}(\mu_\ell)$. The second step conditioned the latter distribution on a set S of probability mass at least $1/2$, which can only increase $\mathrm{RU}(\cdot)$ by 1, so $\mathrm{RU}(\mu'_\ell) \leq n\,\mathrm{RU}(\mu_\ell)+1$.

The third step found the set of all bits fixed in partial assignments p which certify some (x,u) as evaluating to 1, and picked the highest-probability partial assignment on those bits. The maximum increase in $\mathrm{RU}(\cdot)$ is the number of bits that were fixed in this way. This number comes from Theorem 16, and depends on the number of bits fixed in q; when $|q| = 2^{(\log n)^d}$, the number we are looking for is $c\log n \cdot 2^{(\log n)^{d+2}}$, so we can express this as $c\log n \cdot 2^{(\log^2 n)(\log|q|)}$. We had $|q| \leq n\,\mathrm{RU}(\mu_\ell)$ and $c=\log n$. It is not hard to see that this additive increase dominates $n\,\mathrm{RU}(\mu_\ell)+1$; assuming everything is large enough (e.g. $\log n$ is sufficiently large, and $\mathrm{RU}(\mu_\ell)$ is at least n^2, which is without loss of generality by restricting the original μ_0 to a smaller set if necessary), we can get the upper bound $\mathrm{RU}(\mu_{\ell+1}) \leq 2^{2\log^2 n \log \mathrm{RU}(\mu_\ell)}$, as desired. ◁

▷ Claim 24. Assume the witness size k is $O(\mathrm{poly}(n))$ and the number of rounds r is $o(\log n/\log\log n)$, and let n be large enough. With notation as in Claim 23, there exists $\hat{f} \in \mathrm{supp}(\mu_r)$ and a large set E (with $|E| \geq (2/3)2^n$) of inputs x such that for every $x \in E$ and every u, the pair (x,u) is "not fixed to 1 by μ_r" (that is, there exists $f \in \mathrm{supp}(\mu_r)$ such that $O[f,\emptyset](x,u)=0$).

Proof. We essentially apply another round-reduction iteration (without the second step) to μ_r. Using Lemma 20, we find a partial assignment q' such that $\mu_r|_{q'}$ is $(1-\delta)$-dense outside of q', with $\delta = 1/n$. We then apply Theorem 16 to conclude there are few pairs (x,u) with $\Pr_f[O[f,\emptyset](x,u)=1] \geq 1/2$, and hence few pairs (x,u) with $\Pr_f[O[f,\emptyset](x,u)=1] = 1$

when f is sampled from $\mu_r|_{q'}$; the number of such pairs is at most $2^{(2\log n)^{2r+2}\log k}$. Using $k = O(\text{poly}(n))$ and $r = o(\log n/\log\log n)$, this means that there are at most $2^{o(n)}$ pairs (x,u) that are fixed to 1 for all the oracles $O[f,\emptyset]$ for $f \in \text{supp}(\mu_r|_{q'})$. Therefore, there are $2^{n-o(n)}$ many inputs x such that for all u, the pair (x,u) is not fixed to 1 by $\text{supp}(\mu_r|_{q'})$. Let E be the set of such x; then $|E| \geq (2/3)2^n$. Let $\hat{f} \in \text{supp}(\mu_r|_{q'})$ be arbitrary, and the desired result follows. ◁

Proof of Theorem 21. Start with a QCMA protocol for f_N, and use Claim 22 to get a Q and μ; to get a contradiction, we just need to find $f \in \text{supp}(\mu)$ and a large set E of inputs x such that Q fails to distinguish the oracle $O[f,\emptyset]$ from the oracle $O[f,E]$.

Let (Q_ℓ, μ_ℓ) be as in Claim 23, and let $\hat{f}$ and E be as in Claim 24. To complete the proof, we just need show that $Q = Q_0$ fails to distinguish $O[\hat{f},\emptyset]$ and $O[\hat{f},E]$.

Let $B = \{(x,u) : x \in E, O[\hat{f},\emptyset](x,u) = 1\}$. Moreover, let B_ℓ be the set of pairs (x,u) which had $\Pr_{f\sim\mu_{\ell-1}|_q}[O[f,\emptyset](x,u) = 1] \leq \epsilon$ in iteration ℓ (where q is the partial assignment from step 1 of iteration ℓ). Note that the pairs not in B_ℓ are all fixed in all the oracles in the support of μ_ℓ, because we choose values for the bits used by their proving partial assignments p. This means that $B \subseteq B_\ell$ for all ℓ. Also, let O_ℓ be the oracle used by Q_ℓ to simulate the first query batch of $Q_{\ell-1}$. Recall that $O_\ell(x,u)$ returns 0 unless (x,u) is fixed to 1 in all $O[f,\emptyset]$ for $f \in \text{supp}(\mu_\ell)$. Since the support of μ_ℓ decreases as a subset in each iteration, the bits fixed in μ_ℓ are also fixed in μ_r, and hence also agree with $\hat{f}$. This means that O_ℓ can be written as an erased oracle $O[\hat{f},A_\ell]$ for some set A_ℓ of pairs (x,u) that were not fixed in μ_ℓ; in other words, $A_\ell \subseteq B_\ell$.

We now note the oracle $O[\hat{f},E]$ is the same as $O[\hat{f},B]$. Additionally, since $B, A_\ell \subseteq B_\ell$, we have by Lemma 18,

$$\|U^{O[\hat{f},B]}|\psi\rangle - U^{O[\hat{f},A_\ell]}|\psi\rangle\|_2 \leq 1/20r$$

where $|\psi\rangle$ is the state right before the first query of the algorithm $Q_{\ell-1}$. This can also be written

$$\|U^{O[\hat{f},E]}|\psi\rangle - U^{O_\ell}|\psi\rangle\|_2 \leq 1/20r.$$

Now, applying additional unitary matrices does not change the 2-norm, and Q_ℓ replaces only the first query of $Q_{\ell-1}$ with O_ℓ and applies the same unitaries as $Q_{\ell-1}$ in all other rounds. If we use $Q_\ell(O)$ to denote the final state of Q_ℓ on the oracle O, we therefore get

$$\|Q_\ell(O[\hat{f},E]) - Q_{\ell-1}(O[\hat{f},E])\|_2 \leq 1/20r.$$

By triangle inequality, we then get

$$\|Q(O[\hat{f},E]) - Q_r(O[\hat{f},E])\|_2 \leq 1/20.$$

Since $\emptyset \subseteq B_\ell$ for all ℓ, the same argument also works to show that

$$\|Q(O[\hat{f},\emptyset]) - Q_r(O[\hat{f},\emptyset])\|_2 \leq 1/20,$$

and of course we also have $Q_r(O[\hat{f},\emptyset]) = Q_r(O[\hat{f},E])$ since Q_r makes no queries. A final application of the triangle inequality gives us

$$\|Q(O[\hat{f},E]) - Q(O[\hat{f},\emptyset])\|_2 \leq 1/10.$$

This gives the desired contradiction, as Q failed to sufficiently distinguish these two oracles (it must accept one with probability at least 2/3 and the other with probability at most 1/3; converting this to a lower bound on the 2-norm distance is a straightforward exercise). ◀

References

1 Scott Aaronson, Harry Buhrman, and William Kretschmer. A Qubit, a Coin, and an Advice String Walk into a Relational Problem. In *15th Innovations in Theoretical Computer Science Conference (ITCS 2024)*, 2024. doi:10.4230/LIPIcs.ITCS.2024.1.

2 Scott Aaronson and Greg Kuperberg. Quantum versus classical proofs and advice. In *Twenty-Second Annual IEEE Conference on Computational Complexity (CCC'07)*, pages 115–128, 2007. doi:10.1109/CCC.2007.27.

3 Dorit Aharonov and Tomer Naveh. Quantum NP - A Survey, 2002. doi:10.48550/arXiv.quant-ph/0210077.

4 Roozbeh Bassirian, Bill Fefferman, and Kunal Marwaha. On the Power of Nonstandard Quantum Oracles. In *18th Conference on the Theory of Quantum Computation, Communication and Cryptography (TQC 2023)*, 2023. doi:10.4230/LIPIcs.TQC.2023.11.

5 Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and weaknesses of quantum computing. *SIAM Journal on Computing*, 26:1510–1523, 1997. doi:10.1137/S0097539796300933.

6 Sandro Coretti, Yevgeniy Dodis, Siyao Guo, and John Steinberger. Random oracles and non-uniformity. In *Advances in Cryptology – EUROCRYPT 2018*, 2018.

7 Bill Fefferman and Shelby Kimmel. Quantum vs. Classical Proofs and Subset Verification. In *43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018)*, 2018. doi:10.4230/LIPIcs.MFCS.2018.22.

8 Mika Göös, Toniann Pitassi, and Thomas Watson. Query-to-communication lifting for BPP. In *Proceedings of the 58th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2017. doi:10.1109/FOCS.2017.21.

9 Xingjian Li, Qipeng Liu, Angelos Pelecanos, and Takashi Yamakawa. Classical vs Quantum Advice and Proofs Under Classically-Accessible Oracle. In *15th Innovations in Theoretical Computer Science Conference (ITCS 2024)*, 2024. doi:10.4230/LIPIcs.ITCS.2024.72.

10 Qipeng Liu. Non-uniformity and quantum advice in the quantum random oracle model. In Carmit Hazay and Martijn Stam, editors, *Advances in Cryptology – EUROCRYPT 2023*, 2023.

11 Anand Natarajan and Chinmay Nirkhe. A Distribution Testing Oracle Separating QMA and QCMA. In *38th Computational Complexity Conference (CCC 2023)*, 2023. doi:10.4230/LIPIcs.CCC.2023.22.

12 Ran Raz and Avishay Tal. Oracle separation of bqp and ph. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, 2019. doi:10.1145/3313276.3316315.

13 Atri Rudra. *List Decoding and Property Testing of Error Correcting Codes*. PhD thesis, University of Washington, 2007. URL: https://cse.buffalo.edu/faculty/atri/papers/coding/thesis.html.

14 Takashi Yamakawa and Mark Zhandry. Verifiable quantum advantage without structure. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 69–74, 2022. doi:10.1109/FOCS54457.2022.00014.