Quantum vs Classical Proofs and Subset Verification

Bill Fefferman[1] and Shelby Kimmel[1]

[1]Joint Center for Quantum Information and Computer Science (QuICS), University of Maryland

Abstract

We study the ability of efficient quantum verifiers to decide properties of exponentially large subsets given either a classical or quantum witness. We develop a general framework that can be used to prove that QCMA machines, with only classical witnesses, cannot verify certain properties of subsets given implicitly via an oracle. We use this framework to prove an oracle separation between QCMA and QMA using an "in-place" permutation oracle, making the first progress on this question since Aaronson and Kuperberg in 2007 [3]. We also use the framework to prove a particularly simple standard oracle separation between QCMA and AM.

1 Introduction

How much computational power does an efficient quantum verifier gain when given a polynomial sized quantum state to support the validity of a mathematical claim? In particular, is there a problem that can be solved in this model, that cannot be solved if the verifier were instead given a classical bitstring? This question, the so-called QMA vs QCMA problem, is fundamental in quantum complexity theory. To complexity theorists, the question can be motivated simply by trying to understand the power of quantum nondeterminism, where both QMA and QCMA can be seen as "quantum analogues" of NP. More physically, QMA is characterized by the k-local Hamiltonian problem, in which we must decide if the ground state energy of a local Hamiltonian is above or below a specified threshold [15, 5]. In this setting, the QMA vs QCMA question asks whether there exists a purely *classical* description of the ground state that allows us to make this decision. For instance, if the ground state of any local Hamiltonian can be prepared by an efficient quantum circuit, then QMA = QCMA, as the classical witness for the k-local Hamiltonian problem would be the classical description of this quantum circuit. It was this intuition that caused Aharonov and Naveh to conjecture that these classes are equal, in the paper that first defined QCMA [5].

It was recently established [11] that the witness to a QMA machine may always be replaced by a subset state, where a subset state on n qubits has the form $|S\rangle = 1/\sqrt{|S|}\sum_{i\in S}|i\rangle$ for some subset $S \subset [2^n]$. However, it seems difficult to create a classical witness on n bits that captures the information in a subset state $|S\rangle$. Therefore, problems involving subsets seem like ripe ground for understanding the QMA vs. QCMA problem. We investigate the following question: under what circumstances is it possible for a quantum machine to verify properties of a subset? This question is not answered by [11]; they study general properties of languages that are in QMA and QCMA, while we attempt to prove specific languages of interest (that are related to subsets) are either in or not in QMA or QCMA.

In the hopes of further exploring these questions, we exhibit a general framework that can be used to obtain oracle separations against QCMA for subset-based problems. We use this framework to prove the existence of an "in-place" permutation oracle $\mathcal{P}$ (a unitary that permutes standard basis states within a single register) [9, 2] for which $\mathsf{QMA}^{\mathcal{P}} \not\subseteq \mathsf{QCMA}^{\mathcal{P}}$, making the first progress on this problem since Aaronson & Kuperberg in 2007 [3]. In this problem, for the case of QMA, the in-place permutation oracle allows us to verify that the given witness is indeed the correct subset state. On the other hand, our framework allows us to prove the language is not in QCMA. Our framework is quite general, and we are also able to use it to establish a particularly simple example of a (conventional) oracle $\mathcal{O}$ so that $\mathsf{AM}^{\mathcal{O}} \not\subseteq \mathsf{QCMA}^{\mathcal{O}}$.[1]

1.1 Subset-Verifying Oracle Problems

We consider two oracle problems related to verifying properties of subsets. In *Subset Size Checking*, we are given a black box function $f : [N] \to \{0,1\}$, that marks elements with either a 0 or 1. We are promised that the number of marked items is either $\sqrt{N}$ or $.99\sqrt{N}$, and we would like to decide which. In this case, we want to verify the size of the subset marked by f.

In the other oracle problem, *Preimage Checking*, we are given a black box permutation on N^2 elements. We are promised that the preimage of the first N elements under the permutation is either mostly even or mostly odd, and we would like to decide which is the case. In this problem, we want to verify the parity of a subset of the preimage of the function.

Subset Size Checking is in AM [10], and we give a procedure that proves *Preimage Checking* is in QMA when the permutation is given as an in-place quantum oracle. An in-place permutation unitary $\mathcal{P}_\sigma$ acts as $\mathcal{P}_\sigma|i\rangle = |\sigma(i)\rangle$ for a permutation σ. For *Preimage Checking*, we are interested in the set $S_{\mathrm{pre}}(\sigma) = \{i : \sigma(i) \in [N]\}$. Given the subset state $|S_{\mathrm{pre}}(\sigma)\rangle$, it is easy to verify that the correct state was sent, because $\mathcal{P}_\sigma|S_{\mathrm{pre}}(\sigma)\rangle = |[N]\rangle$, which is easy to verify using a measurement in the Hadamard basis.

However, we don't expect subset-based oracle problems like *Subset Size Checking* and *Preimage Checking* to be in QCMA because the classical witness does not have enough information to identify the relevant subset. We make this intuition more precise by providing a general recipe for proving that subset-verifying oracle languages are not in QCMA. We apply this procedure to show that both *Preimage Checking* (with a randomized in-place oracle - see Section 3 for more details) and *Subset Size Checking* are not in QCMA. The procedure involves familiar tools, like the adversary bound [6] (although adapted to our in-place oracle when necessary), as well as a new tool, the *Fixing Procedure*, which finds subsets with nice structure within a large arbitrary set. We sketch the recipe, which is similar to the approach used to show the first oracle separation between QMA and QCMA [3]:

1. We show that for every QCMA machine, there are more valid oracles than possible classical witnesses, so by a counting argument, there must be one classical witness w^* that corresponds to a large number of potential oracles. We then restrict ourselves to considering oracles that correspond to w^*.

2. Because we are considering subset-verifying problems, if we have a collection of black box functions that corresponds to w^*, we immediately have some set of subsets that corresponds to w^*. At this point, we know nothing about this set of subsets except its size, thanks to the counting argument. We next show (using the *Fixing Procedure*) that if we have a set of subsets of a certain size, we can always find a subset of the original set that has nice structure.

[1]Note there was previously an example of an oracle separating AM from PP [19]. Since $\mathsf{QMA} \subseteq \mathsf{PP}$ [17], this is formally a stronger result. Nonetheless, our oracle is substantially different, and uses completely different ideas.

3. We apply the adversary bound to the subset with nice structure to show that the number of quantum queries needed to distinguish between YES and NO cases is exponential.

4. We finally put these pieces together in a standard diagonalization argument.

1.2 Technical Contributions

We make several technical contributions that may be of outside interest. Our adversary bound for in-place permutation oracles provides a query lower-bounding technique for unitary oracles when access is *not* given to the oracle's inverse. (While Belovs [8] created an adversary bound for arbitrary unitaries, his results assume access to an inverse). While we typically assume quantum oracles include access to an inverse or are self-inverting, in open quantum systems it is natural to not have an inverse.

When proving that *Preimage Checking* is not in QCMA, we actually use an oracle that is not unitary. The oracle is a completely positive trace preserving (CPTP) map that at each application applies one unitary chosen uniformly at random from among a set of unitaries. Standard lower bounding techniques fail for such an oracle. The closest result is from Regev and Schiff [18] who give a lower bound on solving Grover's problem with an oracle that produces errors. Regev and Schiff deal with the non-unitarity of the map by showing that the state of the system can be modeled using pure states. This strategy does not work in our case. Instead, we take advantage of the fact that every non-unitary CPTP map can be implemented as a unitary on a larger system. In our case, we can simulate our random oracle using a unitary black box oracle acting on subsystem A, followed by a fixed unitary that entangles subsystems A and B. The entangling operation that acts on the larger system can not be efficiently implemented, but as we are bounding query complexity, this is acceptable. This technique may be of use for similar problems; for example, we do not know the query complexity of solving Grover's problem with an oracle that produces a depolarizing error with each application. A depolarizing map is similar to our CPTP map in that both maps can be thought of as applying a unitary at random from among a set of unitaries, and so perhaps this approach will stimulate new approaches for the Grover problem.

1.3 Impact and Directions for Future Research

While Aaronson and Kuperberg have previously proved an oracle separation between QMA and QCMA [3], their oracle seems to be especially quantum, as it is defined by a Haar random quantum state. Our in-place oracle has more of a classical feel, in that it encodes a classical permutation function. However, it is still not a standard quantum oracle, as it is not self-inverting. Is there a standard (i.e. not in-place) oracle language that separates QMA and QCMA? Although we can only prove a separation when our in-place oracle also has randomness, we believe our techniques could be adapted to prove a similar result but without the randomness in the oracle. While we give a recipe for showing certain subset-based problems are not in QCMA, we believe some of these problems are also not in QMA; for example, is it possible to prove *Subset Size Checking* is not in QMA?

Our contribution to techniques for lower bounding query complexity for non-standard oracles raise several questions. Is there a general adversary bound [13, 16] for in-place permutation oracles? There are examples of problems for which in-place permutation oracles require exponentially fewer queries that standard permutation oracles e.g., [7]. We conjecture that the opposite is also true - that there are examples of problems for which standard permutation oracles require exponentially fewer queries than in-place oracles. In fact, we do not believe it is possible to obtain a Grover-type speed-up with an in-place oracle; we believe the problem of determining the inverse of an element of an in-place permutation oracle requires N queries for a permutation on N elements. However, in order to prove these results, we suspect one needs a more powerful tool, like a general adversary bound for in-place oracles.

1.4 Organization

The rest of the paper is organized as follows, in Section 2, we introduce notation that will be used throughout the rest of the paper, and define QMA and QCMA. In Section 3, we define and discuss standard, in-place, and randomized in-place quantum permutations, as well as state an adversary lower bound for in-place permutation unitaries. In Section 4, we describe the *Preimage Checking* Problem and prove it is in QMA. In Section 5, we lay out the general recipe for proving subset-based languages are not in QCMA. In Section 6, we apply this procedure to the *Subset Size Checking* problem, and use it to prove an oracle separation between AM and QCMA. Finally, in Section 7, we apply the procedure to *Preimage Checking* and show this problem is not in QCMA.

2 Definitions and Notation

We use the notation $[M] = \{1, 2, \ldots, M\}$. σ will refer to a permutation, and $\boldsymbol{\sigma}$ will refer to a set of permutations. Likewise, S will refer to a set of positive integers, and $\mathbf{S}$ will refer to a set of sets of positive integers. For S a set of positive integers, a subset state $|S\rangle$ is

$$|S\rangle = \frac{1}{\sqrt{|S|}} \sum_{i \in S} |i\rangle. \tag{1}$$

Throughout, we use $N = 2^n$. All logarithms are in base 2. We use $\boldsymbol{\sigma}^n$ to be the set of permutations acting on N^2 elements. That is if $\sigma \in \boldsymbol{\sigma}^n$, $\sigma : [N^2] \to [N^2]$. For positive integers $i > j$, let $\mathbf{C}(i, j)$ be the set of sets containing j elements of $[i]$:

$$\mathbf{C}(i, j) = \{S \subset [i] : |S| = j\}. \tag{2}$$

We use calligraphic font $\mathcal{P}, \mathcal{U}$ to denote unitary operations. We use elaborated calligraphic font $\mathscr{P}$, $\mathscr{U}$ to denote CPTP maps. For a unitary $\mathscr{U}$ acting on a density matrix ρ, we have

$$\mathscr{U}(\rho) = \mathcal{U} \rho \mathcal{U}^\dagger, \tag{3}$$

where $(\cdot)^\dagger$ denotes conjugate transpose. We denote the composition of two CPTP maps with $\circ$, so $\mathscr{E} \circ \mathscr{F}$ means apply $\mathscr{F}$ first, and then $\mathscr{E}$.

We include the following definition in order to be complete, although it is completely standard (e.g., see also [1, 3])

Definition 1. QMA *is the set of decision languages $L \subseteq \{0,1\}^*$ so that there exists an efficient quantum verifier V_L and a polynomial $p(\cdot)$:*

1. *Completeness: For any $x \in L$ there exists a $p(|x|)$-qubit pure quantum state $|\psi\rangle$ so that* $\Pr[V_L(x, |\psi\rangle) = 1] \geq 2/3$
2. *Soundness: For any $x \notin L$ and any pure quantum state $|\psi\rangle$,* $\Pr[V_L(x, |\psi\rangle) = 1] \leq 1/3$

QCMA *is the exact same class, with the polynomial-length witness $|\psi\rangle$ replaced with a polynomial length classical bitstring.*

3 Permutation Maps

3.1 Permutations as Oracles: In-Place Permutation vs. Standard Permutation

Black box permutation unitaries have been considered previously, most notably in the collision and element distinctness problems [2, 4]. However, the permutation unitaries considered in these works were standard oracles. A standard oracle that implements the permutation $\sigma \in \boldsymbol{\sigma}^n$ acts as

$$\mathcal{P}_\sigma^{\text{stand}}|i\rangle|b\rangle = |i\rangle|b \oplus \sigma(i)\rangle \tag{4}$$

for $i, b \in [N^2]$, where $|i\rangle$ for $i \in [N^2]$ are standard basis states and $\oplus$ denotes bitwise XOR. Note that $(\mathcal{P}_\sigma^{\text{stand}})^2 = \mathbb{I}_{N^4}$; that is, acting with the unitary twice produces the $N^4 \times N^4$ identity operation.

In this work, we consider in-place permutation unitaries, which implement the permutation $\sigma \in \boldsymbol{\sigma}^n$ as

$$\mathcal{P}_\sigma|i\rangle = |\sigma(i)\rangle. \tag{5}$$

Now, in general $(\mathcal{P}_\sigma)^2 \neq \mathbb{I}_{N^2}$. Crucially, given black box access to $\mathcal{P}_\sigma$, *we do not give black box access to the inverse of the oracle.* In fact, in Section 3.2, we present an adversary bound that we use to show that given only $\mathcal{P}_\sigma$, it is hard to invert its action. The use of non-self-invertible permutation oracles has been considered previously, for example in [9, 2].

While the standard and in-place permutation unitaries seem similar, we believe they are incomparable. That is, given one type of oracle that implements a permutation σ, you can not efficiently simulate the other type implementing the same permutation. As a simple example, consider a permutation $\sigma \in \boldsymbol{\sigma}^n$. If we have the state $\sum_{y\in S}|y\rangle|\sigma(y)\rangle$ (normalization omitted), we can create the state $\sum_{y\in S}|y\rangle|0\rangle$ with a single query to $\mathcal{P}_\sigma^{\text{stand}}$. However, if we only have access to the corresponding in-place permutation $\mathcal{P}_\sigma$ and no access to $\mathcal{P}_{(\sigma)^{-1}} = (\mathcal{P}_\sigma)^{-1}$, it seems difficult to create this state.

On the other hand, suppose we want to prepare the state $\sum_{y\in[N]}|\sigma(y)\rangle$ (normalization omitted). If we have access to the in-place permutation oracle $\mathcal{P}_\sigma$, we can create this state in one query by applying the oracle to the uniform superposition $\sum_{y\in[N]}|y\rangle$. In the standard model, this problem is called "index erasure," and requires an exponential number of queries in n to $\mathcal{P}_\sigma^{\text{stand}}$ [7].

3.2 An Adversary Bound for In-Place Permutation Oracles

In Appendix A, we prove a non-weighted adversary bound for in-place permutations oracles that is analogous to the bound for standard permutation oracles. In fact, the bound is not just analogous, but identical to what Ambainis proves in Theorem 6 in [6] for standard permutation oracles.

Lemma 1. *Let $\boldsymbol{\sigma} \subset [V] \to [V]$ be a subset of permutations acting on the elements $[V]$. Let $f : \boldsymbol{\sigma} \to \{0,1\}$ be a function of permutations. Let $\boldsymbol{\sigma}_X \subset \boldsymbol{\sigma}$ be a set such that if $\sigma \in \boldsymbol{\sigma}_X$, then $f(\sigma) = 1$. Let $\boldsymbol{\sigma}_Y \subset \boldsymbol{\sigma}$ be a set such that if $\sigma \in \boldsymbol{\sigma}_Y$ then $f(\sigma) = 0$. Let $R \subset \boldsymbol{\sigma}_X \times \boldsymbol{\sigma}_Y$ be such that*

- *For every $\sigma_x \in \boldsymbol{\sigma}_X$, there exists at least m different $\sigma_y \in \boldsymbol{\sigma}_Y$ such that $(\sigma_x, \sigma_y) \in R$.*
- *For every $\sigma_y \in \boldsymbol{\sigma}_Y$, there exists at least m' different $\sigma_x \in \boldsymbol{\sigma}_X$ such that $(\sigma_x, \sigma_y) \in R$.*
- *Let $l_{x,i}$ be the number of $\sigma_y \in \boldsymbol{\sigma}_Y$ such that $(\sigma_x, \sigma_y) \in R$ and $\sigma_x(i) \neq \sigma_y(i)$. Let $l_{y,i}$ be the number of $\sigma_x \in \boldsymbol{\sigma}_Y$ such that $(\sigma_x, \sigma_y) \in R$ and $\sigma_x(i) \neq \sigma_y(i)$. Then let $l_{max} = \max_{(\sigma_x,\sigma_y)\in R, i} l_{x,i} l_{y,i}$.*

Then given an in-place permutation oracle $\mathcal{P}_\sigma$ for $\sigma \in \boldsymbol{\sigma}$ that acts as

$$\mathcal{P}_\sigma |i\rangle = |\sigma(i)\rangle \tag{6}$$

any quantum algorithm that correctly evaluates $f(\sigma)$ with probability $1-\epsilon$ for every element of $\boldsymbol{\sigma}_X$ and $\boldsymbol{\sigma}_Y$ must use $\left(1 - 2\sqrt{\epsilon(1-\epsilon)}\right)\sqrt{\frac{mm'}{l_{max}}}$ queries to the oracle.

As a corollary of Lemma 1, (using exactly the same technique as Theorem 7 in [6]), we have that the query complexity of inverting an in-place permutation oracle that permutes the elements of $[V]$ is $\Omega(V^{1/2})$.

3.3 Permutations with Randomness

Additionally, we consider in-place permutation oracles with internal randomness. An oracle with internal randomness is a black box quantum operation that is a generalized CPTP (completely positive trace preserving) map, rather than a unitary. Oracles with internal randomness were shown to cause a complete loss of quantum speed-up in [18], while in [12], such oracles were shown to give an infinite quantum speed-up.

In this work, we consider oracles that apply an in-place permutation at random from among a set of possible permutations. Let $\boldsymbol{\sigma} \in \boldsymbol{\sigma}^n$ be a set of permutations with $|\boldsymbol{\sigma}|$ permutations in the set. Then the quantum CPTP map $\mathscr{P}_{\boldsymbol{\sigma}}$ acts as follows:

$$\mathscr{P}_{\boldsymbol{\sigma}}(\rho) = \frac{1}{|\boldsymbol{\sigma}|}\sum_{\sigma \in \boldsymbol{\sigma}} \mathcal{P}_\sigma \rho \mathcal{P}_\sigma^\dagger. \tag{7}$$

4 Pre-Image Checking

In this section, we define a property of oracle families which we call *randomized-preimage-correct*, and construct a decision language based on such oracles that is in QMA. Essentially, the problem is to decide whether the preimage of the first N elements of a permutation is mostly even or mostly odd. We first define randomized-preimage-correct, and then show that the related language is in QMA.

Given a permutation $\sigma \in \boldsymbol{\sigma}^n$, we associate a subset $S_{\text{pre}}(\sigma)$ to that permutation ("pre" is for "preimage"), where

$$S_{\text{pre}}(\sigma) = \{j : \sigma(j) \in [N]\}. \tag{8}$$

That is, $S_{\text{pre}}(\sigma)$ is the subset of elements in $[N^2]$ whose image under σ is in $[N]$. Additionally, to each subset $S \in [N^2]$ with $|S| = N$, we associate a set of permutations $\boldsymbol{\sigma}_{\text{pre}}(S)$:

$$\boldsymbol{\sigma}_{\text{pre}}(S) = \{\sigma : \sigma \in \boldsymbol{\sigma}^n, S_{\text{pre}}(\sigma) = S\}. \tag{9}$$

In particular, we care about certain sets:

$$\begin{aligned} \boldsymbol{S}^n_{\text{even}} &= \{S : S \subset [N^2], |S| = N, |S \cap \mathbb{Z}_{\text{even}}| = 2/3N\} \\ \boldsymbol{S}^n_{\text{odd}} &= \{S : S \subset [N^2], |S| = N, |S \cap \mathbb{Z}_{\text{odd}}| = 2/3N\} \end{aligned} \tag{10}$$

Definition 2 (randomized-preimage-correct oracles)**.** *Let $\boldsymbol{\mathcal{O}}$ be a countably infinite set of quantum operators (CPTP maps): $\boldsymbol{\mathcal{O}} = \{\mathscr{O}_1, \mathscr{O}_2, \dots\}$, where each $\mathscr{O}_n$ implements an operation on $(2n)$-qubits. We say that $\boldsymbol{\mathcal{O}}$ is randomized-preimage-correct if for every n, $\mathscr{O}_n = \mathscr{P}_{\boldsymbol{\sigma}_{\text{pre}}(S)}$, with $S \in \boldsymbol{S}^n_{even} \cup \boldsymbol{S}^n_{odd}$.*

Theorem 1. *For any randomized-preimage-correct $\boldsymbol{\mathcal{O}}$, the unary language $L_{\boldsymbol{\mathcal{O}}}$, which contains those unary strings 1^n such that $\mathscr{O}_n = \mathscr{P}_{\boldsymbol{\sigma}_{\mathrm{pre}}(S)}$ with $S \in \boldsymbol{S}^n_{even}$, is in* $\mathsf{QMA}^{\boldsymbol{\mathcal{O}}}$.

Proof. We first prove completeness. If $1^n \in L_{\boldsymbol{\mathcal{O}}}$, then $\mathscr{O}_n = \mathscr{P}_{\boldsymbol{\sigma}_{\mathrm{pre}}(S)}$ for some $S \in \boldsymbol{S}^n_{\mathrm{even}}$. We show that there exists a witness of $2n$ qubits and an efficient verifier using $\mathscr{P}_{\boldsymbol{\sigma}_{\mathrm{pre}}(S)}$, which outputs 1 with probability 5/6.

Consider using as a witness the subset state $|S\rangle$ when the oracle is $\mathscr{P}_{\boldsymbol{\sigma}_{\mathrm{pre}}\sigma(S)}$. We analyze the following verifier: with probability 1/2, do either

Test (i) Apply $\mathscr{P}_{\boldsymbol{\sigma}_{\mathrm{pre}}(S)}$ to $|S\rangle$, and then measure whether the resultant state is $|[N]\rangle$. Since $N = 2^n$ is a power of 2, this measurement can be done by applying $H^{\otimes n}$ to the first n qubits, and then measuring in the standard basis. If the outcome is 0, output 1; otherwise, output 0.

Test (ii) Measure $|S\rangle$ in the standard basis. Let i^* be the resulting standard basis state. If i^* is odd, output 0. Otherwise, apply $\mathscr{P}_{\boldsymbol{\sigma}_{\mathrm{pre}}(S)}$ to $|i^*\rangle$ and measure the resultant standard basis state. If the resultant state is not in $[N]$, output 0; otherwise, output 1.

If Test (i) is implemented, the verifier will output 1 with probability 1. Even though $\mathscr{P}_{\boldsymbol{\sigma}_{\mathrm{pre}}(S)}$ does not apply a fixed permutation, but rather chooses a permutation at random from among a set to apply, all the permutations that might be applied by $\mathscr{P}_{\boldsymbol{\sigma}_{\mathrm{pre}}(S)}$ will transform $|S\rangle$ into $|[N]\rangle$. If Test (ii) is implemented, the verifier will output 1 with probability 2/3. Averaging over both Tests, the verifier will output 1 with probability 5/6.

Now we show soundness. If $1^n \notin L_{\boldsymbol{\mathcal{O}}}$, then $\mathscr{O}_n = \mathscr{P}_{\boldsymbol{\sigma}_{\mathrm{pre}}(S)}$ for some $S \in \mathbf{S}^n_{\mathrm{odd}}$. We will prove that for any witness of $2n$ qubits, the probability that the verifier given above outputs 1 is less than 2/3.

Without loss of generality, we can assume that the witness has the form

$$|\psi(S)\rangle = \sum_{i=1}^{N^2} \beta_i |i\rangle. \tag{11}$$

Then the probability the verifier outputs 1 after performing Test (i) is

$$p_{(i)} = \frac{1}{N}\left|\sum_{i\in S}\beta_i\right|^2, \tag{12}$$

regardless of which permutation the map $\mathscr{P}_{\boldsymbol{\sigma}_{\mathrm{pre}}(S)}$ applies. The probability the verifier outputs 1 after performing Test (ii) is

$$p_{(ii)} = \sum_{i\in\mathbb{Z}_{\mathrm{even}}\cap S} |\beta_i|^2. \tag{13}$$

We will relate these two quantities in the following way:

$$\begin{aligned}1 =&\sum_{i\in S}|\beta_i|^2+\sum_{i\notin S}|\beta_i|^2\\
=&\frac{1}{\sqrt{2}}\sum_{i\in\mathbb{Z}_{\text{even}}\cap S}|\beta_i|^2+\sum_{i\in\mathbb{Z}_{\text{odd}}\cap S}|\beta_i|^2+\frac{\sqrt{2}-1}{\sqrt{2}}\sum_{i\in\mathbb{Z}_{\text{even}}\cap S}|\beta_i|^2+\sum_{i\notin S}|\beta_i|^2\\
\geq&\sqrt{\frac{3}{2N}}\sum_{i\in\mathbb{Z}_{\text{even}}\cap S}|\beta_i|+\sqrt{\frac{3}{2N}}\sum_{i\in\mathbb{Z}_{\text{odd}}\cap S}|\beta_i|+\frac{\sqrt{2}-1}{\sqrt{2}}\sum_{i\in\mathbb{Z}_{\text{even}}\cap S}|\beta_i|^2+\sum_{i\notin S}|\beta_i|^2\\
\geq&\sqrt{\frac{3}{2N}}\left|\sum_{i\in S}\beta_i\right|+\frac{\sqrt{2}-1}{\sqrt{2}}\sum_{i\in\mathbb{Z}_{\text{even}}\cap S}|\beta_i|^2\\
=&\sqrt{\frac{3}{2N}}\sqrt{Np_{(i)}}+\frac{\sqrt{2}-1}{\sqrt{2}}p_{(ii)}\end{aligned}\tag{14}$$

where in the third line, we have used Cauchy-Schwarz, and in the fourth line, the triangle inquality. Thus the total probability that the verifier outputs 1 is

$$\frac{1}{2}\left(p_{(i)}+p_{(ii)}\right)\leq\frac{1}{2}\left(\frac{2}{3}\left(1-\frac{\sqrt{2}-1}{\sqrt{2}}p_{(ii)}\right)^2+p_{(ii)}\right).\tag{15}$$

Taking the derivative of the right hand side, we see it is positive for $0\leq p_{(ii)}\leq 1$, so to maximize the right hand side we take $p_{(ii)}=1$. Doing this, we find the probability that the verifier outputs 1 is at most 2/3. □

We will show that the Preimage Checking problem is not in QCMA in Section 7.

We note that this verification technique does not take advantage of the randomness of the oracle. In fact, this verification works equally well on an in-place oracle without randomness. We use the randomness in our oracle in the proof that randomized-preimage-correct languages can not be decided in QCMA. We believe the separation holds even without randomness in the oracle, but we have been unable to prove it.

5 Strategy for Proving Subset-Based Oracle Languages are not in QCMA

In this section, we will describe a general strategy for showing that certain oracle languages are not in QCMA. In particular, we consider the case when the unitaries that the oracle implements are naturally related to sets of integers.

We require four conditions for our strategy to work. The first condition lays out the required relationship between the unitaries and subsets.

Condition 1. *The oracle must be of the form $\boldsymbol{\mathcal{O}}=\{\mathcal{O}_1,\mathcal{O}_2,\dots\}$ where each $\mathcal{O}_n$ implements a $p_1(n)$-qubit unitary $\mathcal{U}$, for $p_1(\cdot)$ a polynomial, and $\mathcal{U}\in\mathbf{U}^n=\mathbf{U}_X^n\cup\mathbf{U}_Y^n$, where $\mathbf{U}_X^n$ and $\mathbf{U}_Y^n$ are disjoint families of unitaries on $p_1(n)$ qubits. To each unitary $\mathcal{U}$ in $\mathbf{U}^n$ is associated a (not necessarily unique) subset $S_U(\mathcal{U})\in[2^{p_2(n)}]$ for $p_2(\cdot)$ a polynomial. We further require that $\mathbf{S}_X^n=\{S:S=S_U(\mathcal{U}),U\in\mathbf{U}_X^n\}$ is disjoint from $\mathbf{S}_Y^n=\{S:S=S_U(\mathcal{U}),U\in\mathbf{U}_Y^n\}$. Finally, the language $L_{\boldsymbol{\mathcal{O}}}$ must be such that $1^n\in L_{\boldsymbol{\mathcal{O}}}$ if and only if $\mathcal{O}_n\in\mathbf{U}_X^n$.*

Condition 2. *The optimal witness to a $\mathsf{QCMA}^{\boldsymbol{\mathcal{O}}}$ machine that decides a language $L_{\boldsymbol{\mathcal{O}}}$ as in Condition 1, on input 1^n must depend only on $S_U(\mathcal{U})$ where $\mathcal{O}_n=\mathcal{U}$.*

Sometimes Condition 2 is naturally true, as when $\mathcal{U}$ is defined by a subset. In other cases, we must artificially force the the optimal witness to depend only on the subset.

The next two conditions are related to the following definition:

Definition 3. *We say a subset $\mathbf{S}$ is $(\beta, \mathbf{S}_1)$-distributed if:*

(1) There exists a set S_{fixed} such that $S_{\text{fixed}} \subset S$ for all $S \in \mathbf{S}$.

(2) There exists a subset $\mathbf{S}' \in \mathbf{S}_1$ such that $S_{\text{fixed}} \in S$ for all $S \in \mathbf{S}'$.

(3) For every element $i \in \left(\bigcup_{S\in\mathbf{S}} S\right)/S_{\text{fixed}}$, i appears in at most a $2^{-\beta}$-fraction of $S \in \mathbf{S}$.

Condition 3. *Let $0 < \alpha < 1/2$ be a constant and $p(\cdot)$ be a polynomial function. Then there exists a positive integer $n^*(p, \alpha)$, such that for every $n > n^*(p, \alpha)$, and every subset $\mathbf{S} \subset \mathbf{S}_X^n$ such that $|\mathbf{S}| \geq |\mathbf{S}_X^n| 2^{-p(n)}$, there exists a subset $\mathbf{S}' \subset \mathbf{S}$ such that $\mathbf{S}'$ is $(\alpha, \mathbf{S}_Y^n)$-distributed.*

Condition 3 allows us to find a structured subset within every subset of a given size of $\mathbf{S}_X^n$. Then in Condition 4, we use this structure to bound the query complexity of distinguishing between elements of $\mathbf{U}_X^n$ and $\mathbf{U}_Y^n$.

Condition 4. *Suppose $\mathbf{S} \subset \mathbf{S}_X^n$ is $(\delta, \mathbf{S}_Y^n)$-distributed. Then for every quantum algorithm G, there exists a unitary $\mathcal{U}_x \in \mathbf{U}_X^n$ such that $S_U(\mathcal{U}_x) \in \mathbf{S}$, and a unitary $U_y \in \mathbf{U}_Y^n$, such that, given oracle access to $\mathcal{U}_x$ or $\mathcal{U}_y$, G can not distinguish them with probability ϵ without using $\left(1 - 2\sqrt{\epsilon(1-\epsilon)}\right) N^{\delta/2}$ queries to the oracle.*

When these four conditions are satisfied, we can show

Theorem 2. *Let $\mathcal{O}$ be an oracle and $L_{\mathcal{O}}$ be a language that satisfies Conditions 1, 2, 3, and 4. Then $L_{\mathcal{O}}$ is not in $\mathsf{QCMA}^{\mathcal{O}}$.*

Proof. Fix some enumeration over all poly(n)-size quantum verifiers $M_1, M_2, \ldots$, which we can do because the number of such machines is countably infinite (as a consequence of the Solovay-Kitaev theorem [14]). Notice that some of these verifiers may try to guess the language; for example, M_i could output 1 no matter what the input. We fix a language L such that no machine M_i guesses the language. We can always do this because there are more languages than poly(n)-sized machines. We will start with $\mathcal{O} = \emptyset$, $i = 1$, $n_0 = 0$, and repeat the following process for each positive integer n.

Consider the QCMA machine M_i. By Condition 2, for each $\mathcal{U} \in \mathbf{U}_X^n$, the optimal witness only depends on the subset $S_U(\mathcal{U})$. So the witness really only needs to convince us that $S_U(\mathcal{U}) \in \mathbf{S}_X^n$. Let $w_i(S)$ on $p_{M_i}(n)$ bits for $p_{M_i}(\cdot)$ some polynomial be the witness that gives the highest probability of success in convincing M_i that $S \in \mathbf{S}_X^n$. For this optimal assignment of the witness strings, let $\mathbf{S}_{i,\text{wit}}(w)$ be the set of subsets associated with a witness string w. That is

$$\mathbf{S}_{i,\text{wit}}(w) = \{S : w = w_i(S)\}. \tag{16}$$

Then for every n, there exists some string w^{*n} of $p_{M_i}(n)$ bits such that such that

$$|\mathbf{S}_{i,\text{wit}}(w^{*n})| \geq \frac{1}{2^{p_{M_i}(n)}} |\mathbf{S}_X^n|. \tag{17}$$

That is, there exists some witness such that a large number of subsets correspond to that witness.

We now use Condition 3 to prove that for large enough n_i (in particular, for $n_i > n^*(p_{M_i}, 1/4)$, and $n_i > n_{i-1}$), there is a subset $\mathbf{S}_X \subset \mathbf{S}(w^{*n_i})$ that is $(1/4, \mathbf{S}_Y^{n_i})$-distributed. Finally, using

Condition 4, there exists a pair of unitaries $\mathcal{U}_x \in \mathbf{U}_X^{n_i}$ and $\mathcal{U}_y \in \mathbf{U}_Y^{n_i}$ so that M_i can not distinguish which unitary is in $\mathbf{U}_X^{n_i}$ without using $\Omega(2^{n_i/8})$ queries to $\mathcal{O}_{n_i}$.

If $1^{n_i} \in L$, then set $\mathcal{O}_{n_i} = \mathcal{U}_x$, and otherwise, if $1^{n_i} \notin L$, then set $\mathcal{O}_{n_i} = \mathcal{U}_y$. We inductively repeat this process for all M_i. For n that does not equal n_i for some machine M_i, if $1^n \in L$, then add $\mathcal{O}_n = \mathcal{U}$ for any $\mathcal{U} \in \mathbf{U}_X^n$, while if $1^n \notin L$, then add $\mathcal{O}_n = \mathcal{U}$ for any $\mathcal{U} \in \mathbf{U}_Y^n$.

Recall a language L is decided by a $\mathsf{QCMA}^{\boldsymbol{\mathcal{O}}}$ machine M_i iff $\forall x$:

1. $x \in L \Rightarrow \Pr[M_i(x) = 1] \geq \kappa$
2. $x \notin L \Rightarrow \Pr[M_i(x) = 1] \leq 1 - \kappa$

for some constant $\kappa > 1/2$, where the probability is taken over the measurement of the verifier. Notice that we've constructed a language L and an oracle $\boldsymbol{\mathcal{O}}$ so that no $\mathsf{QCMA}^{\boldsymbol{\mathcal{O}}}$ machine M_i will be able to decide every length input with constant bias, and so $L \notin \mathsf{QCMA}^{\boldsymbol{\mathcal{O}}}$. $\square$

6 Subset Size Checking

In this section, we create a subset-based oracle language $L_{\boldsymbol{\mathcal{O}}}$, such that $L_{\boldsymbol{\mathcal{O}}} \in \mathsf{AM}^{\boldsymbol{\mathcal{O}}}$, but $L_{\boldsymbol{\mathcal{O}}} \notin \mathsf{QCMA}^{\boldsymbol{\mathcal{O}}}$. We use the strategy of Section 5 to prove $L_{\boldsymbol{\mathcal{O}}} \notin \mathsf{QCMA}^{\boldsymbol{\mathcal{O}}}$.

Let f_S be a function that marks a subset $S \in N^2$. That is $f_S : \{0,1\}^{2n} \to \{0,1\}$, such that $f_S(i) = 1$ if $i \in S$ and 0 otherwise. Let $\mathcal{F}_S$ be the unitary that implements this function:

$$\mathcal{F}_S|i\rangle = (-1)^{f_S(i)}|i\rangle. \tag{18}$$

Definition 4. *Let $\boldsymbol{\mathcal{O}}$ be a countably infinite set of unitaries (resp. boolean functions): $\boldsymbol{\mathcal{O}} = \{\mathcal{O}_1, \mathcal{O}_2, ...\}$, where $\mathcal{O}_n$ implements a $2n$-qubit (resp. bit) unitary (function). We say $\boldsymbol{\mathcal{O}}$ is subset-gapped if for every n, $\mathcal{O}_n = \mathcal{F}_S$ (resp. $\mathcal{O}_n = f_S$) for $|S| = N$ or $|S| = .99N$.*

Then the following two lemmas give the desired oracle separation between AM and QCMA:

Lemma 2. *For any subset-gapped $\boldsymbol{\mathcal{O}}$, the language $L_{\boldsymbol{\mathcal{O}}}$ that contains those strings 1^n such that $\mathcal{O}_n = f_S$ with $|S| = N$, is in $AM^{\boldsymbol{\mathcal{O}}}$.*

Lemma 2 is proven by Goldwasser and Sipser in [10].

Lemma 3. *For any subset-gapped $\boldsymbol{\mathcal{O}}$, the language $L_{\boldsymbol{\mathcal{O}}}$ that contains those strings 1^n such that $\mathcal{O}_n = \mathcal{F}_S$ with $|S| = N$, is not in $\mathsf{QCMA}^{\boldsymbol{\mathcal{O}}}$.*

To prove this lemma, we will use Theorem 2. To do this, we simply need to show that the Conditions of Theorem 2 hold for the language $L_{\boldsymbol{\mathcal{O}}}$ with $\boldsymbol{\mathcal{O}}$ subset-gapped.

For Condition 1, we need to associate each oracle with a subset. In our case, the oracle is defined by a subset, so we naturally choose the subset S to be associated with $\mathcal{F}_S$, and then all of the other elements of the condition are satisfied. For Condition 2, we need that the optimal witness for $\mathcal{F}_S$ only depend on S, but since since $\mathcal{F}_S$ is completely characterized by S, the optimal witness must be a function of only S. Condition 3 requires that every subset of $\mathbf{C}(N^2, N)$ of a certain size have a subset with certain structure - we prove this result in Lemma 4. Condition 4 provides that if a subset has certain structure, one can prove a query lower bound for distinguishing between different types of oracles. This condition is proven in Lemma 5 using an adversary bound.

Lemma 4. *Let $0 < \alpha < 1/2$ be a constant and $p(\cdot)$ be a polynomial function. Then there exists a positive integer $n^*(p, \alpha)$, such that for every positive integer $n > n^*(p, \alpha)$, and every subset $\mathbf{S} \subset \mathbf{C}(N^2, N)$ such that $|\mathbf{S}| \geq |\mathbf{C}(N^2, N)| 2^{-p(n)}$, there exists a subset $\mathbf{S}' \subset \mathbf{S}$ such that $\mathbf{S}'$ is $(\alpha, \mathbf{C}(N^2, .99N))$-distributed.*

Proof. We prove the existence of $\mathbf{S}'$ by construction. Let $\mathbf{S}$ be any subset of $\mathbf{C}(N^2, N)$ such that $|\mathbf{S}| \geq |\mathbf{C}(N^2, N)| 2^{-p(n)}$. We construct $\mathbf{S}'$ using the following procedure:

Fixing Procedure

1. Set $\mathbf{S}' = \mathbf{S}$, and set $S_{\text{fixed}} = \emptyset$.
2. (a) Let $\nu(i)$ be the number of subsets $S \in \mathbf{S}'$ such that $i \in S$.

   (b) If there exists some element i for which

   $$|\mathbf{S}'| > \nu(i) \geq |\mathbf{S}'| N^{-\alpha} \tag{19}$$

   set $\mathbf{S}' \leftarrow \{S : S \in \mathbf{S}' \text{ and } i \in S\}$, set $S_{\text{fixed}} \leftarrow S_{\text{fixed}} \cup i$, and return to step 2(a). Otherwise exit the Fixing Procedure.

The fixing procedure by construction will always return a set that satisfies points (1) and (3) of Definition 3. Now we need to prove that point (2) is satisfied. As long as fewer than $.99N$ items are fixed (put into the set S_{fixed}), then point (2) will be satisfied.

Let's suppose that at some point in the Fixing Procedure, for sets $\mathbf{S}'$ and S_{fixed}, we have $.5N$ items fixed. Suppose for contradiction there is some element $i^* \notin S_{\text{fixed}}$ that appears in greater than $N^{-\alpha}$ fraction of $S \in \mathbf{S}'$.

Let us look at the set $\mathbf{S}'' = \{S : S \in \mathbf{S}', i^* \in S\}$. Because $(S_{\text{fixed}} \cup i^*) \subset S$ for all $S \in \mathbf{S}''$, there are $.5N - 1$ elements in each $S \in \mathbf{S}''$ that can be chosen freely from the remaining $N^2 - .5N - 1$ un-fixed elements. Thus, we have

$$|\mathbf{S}''| \leq \binom{N^2 - .5N - 1}{.5N - 1}. \tag{20}$$

By assumption $|\mathbf{S}''| \geq |\mathbf{S}'| N^{-\alpha}$, so

$$\begin{aligned} |\mathbf{S}'| &\leq \binom{N^2 - .5N - 1}{.5N - 1} N^{\alpha} \\ &\leq \binom{N^2}{.5N} N^{\alpha} \\ &\leq (2Ne)^{N/2} N^{\alpha} \\ &= 2^{N/2(\log(2e) + \log N) + \log(N)\alpha} \\ &= 2^{O(N) + (N/2)\log N}. \end{aligned} \tag{21}$$

However, we can also bound the size of $\mathbf{S}'$ from the Fixing Procedure. Notice that at every step of the Fixing Procedure, the size of the set $\mathbf{S}'$ is reduced by at most a factor $N^{-\alpha}$. Since we are assuming $.5N$ elements are in S_{fixed}, the Fixing Procedure can reduce the original set $\mathbf{S}$ by at most a factor $N^{-\alpha N/2}$. Since $|\mathbf{S}| \geq |\mathbf{C}(N^2, N)|2^{-p(n)}$, we have that at this point in the Fixing Procedure

$$\begin{aligned} |\mathbf{S}'| &\geq |\mathbf{C}(N^2, N)|2^{-p(n)}N^{-\alpha N/2} \\ &= \binom{N^2}{N} 2^{-p(n)}N^{-\alpha N/2} \\ &\geq N^N 2^{-p(n)}N^{-\alpha N/2} \\ &= 2^{N\log N - p(n) - \log(N)\alpha N/2} \\ &= 2^{-O(N)+N\log N(1-\alpha/2)}. \end{aligned} \tag{22}$$

Notice that as long as $\alpha < 1$, for large enough N (in particular, for $N > 2^{n^*}$ for some positive integer n^*, where n^* depends on α and $p(\cdot)$), the bound of Eq. (22) will be larger than the bound of Eq. (21), giving a contradiction. Therefore, our assumption must have been false, and more than $N/2$ elements can not have been fixed during the Fixing Procedure. Therefore, the final set produced by the fixing procedure will satisfy point (2) of Definition 3. □

We now show Condition 4 holds:

Lemma 5. *Suppose $\mathbf{S}_X \subset \mathbf{C}(N^2, N)$ is $(\delta, \mathbf{C}(N^2, .99N))$-balanced. Then for every quantum algorithm G, there exists a set $S_x \in \mathbf{S}_X$, and a set $S_y \in \mathbf{C}(N^2, .99N)$, (that depend on G) such that, given oracle access to $\mathcal{F}_{S_x}$ or $\mathcal{F}_{S_y}$, G can not distinguish them with probability $\epsilon > .5$ without using $\left(1 - 2\sqrt{\epsilon(1-\epsilon)}\right) N^{\delta/2}$ queries.*

Proof. Since $\mathbf{S}_X$ is $(\delta, \mathbf{C}(N^2, .99N))$-balanced, there exists a set S_{fixed} such that $S_{\text{fixed}} \in S$ for all $S \in \mathbf{S}_X$. Also, the set $\mathbf{S}_Y$ is non-empty, where

$$\mathbf{S}_Y = \{S : S \in \mathbf{C}(N^2, .99N), S_{\text{fixed}}\}. \tag{23}$$

We will use Theorem 6 from [6]. This result is identical to our Lemma 1, except with standard oracles rather than permutation oracles. We define the relation $\mathbf{R}$ as:

$$\mathbf{R} = \{(S_x, S_y) : S_x \in \mathbf{S}_X, S_y \in \mathbf{S}_Y\}. \tag{24}$$

Notice that each $S_x \in \boldsymbol{S}_X$ is paired with every element of $\mathbf{S}_Y$. Thus $m = |\mathbf{S}_Y|$. Likewise $m' = |\mathbf{S}_X|$.

Now consider $(S_x, S_y) \in \mathbf{R}$. We first consider the case of some element j such that $j \in S_x$ but $j \notin S_y$. By our construction of $\mathbf{S}_Y$, $j \notin S_{\text{fixed}}$. We upper bound $l_{x,j}$, the number of $S_{y'}$ such that $(S_x, S_{y'}) \in \mathbf{R}$ and $j \notin S_{y'}$. We use the trivial upper bound $l_{x,j} \leq |\mathbf{S}_Y|$, which is sufficient for our purposes. Next we need to upper bound $l_{y,j}$, the number of $S_{x'}$ such that $(S_{x'}, S_y) \in \mathbf{R}$ and $j \in S_x$. Since S_y is paired with every element of $\mathbf{S}_X$ in $\mathbf{R}$, we just need to determine the number of sets in $\mathbf{S}_X$ that contain j. Because $\mathbf{S}_X$ is δ-balanced, there can be at most $N^{-\delta}|\mathbf{S}_X|$ elements of $\mathbf{S}_X$ that contain j. In this case we have

$$l_{x,j} l_{y,j} \leq |\mathbf{S}_X||\mathbf{S}_Y|N^{-\delta}. \tag{25}$$

We now consider the case that $j \in S_y$ but $j \notin S_x$. (Note this case only occurs when S_{fixed} contains less than $.99N$ elements.) We upper bound $l_{y,j}$, the number of $S_{x'}$ such that $(S_{x'}, S_y) \in \mathbf{R}$ and $j \notin S_{x'}$. Again, we use the trivial upper bound of $l_{y,j} \leq |\mathbf{S}_X|$, which is sufficient for our analysis. Next we upper bound $l_{x,j}$, the number of $S_{y'}$ such that $(S_x, S_{y'}) \in \mathbf{R}$ and $j \in S_{y'}$. In our choice of

$\mathbf{R}$, S_x is paired with every $S_y \in \mathbf{S}_Y$, so we need to count the number of $S \in \mathbf{S}_Y$ that contain j. We have

$$\begin{aligned} l_{x,j} =& \binom{N^2 - S_{\text{fixed}} - 1}{.99N - S_{\text{fixed}} - 1} \\ =& \frac{.99N - S_{\text{fixed}}}{N^2 - S_{\text{fixed}}} |\mathbf{S}_Y| \\ \leq& \frac{1}{N} |\mathbf{S}_Y|. \end{aligned} \tag{26}$$

Therefore in this case, we have

$$l_{x,j} l_{y,j} \leq |\mathbf{S}_X||\mathbf{S}_Y| N^{-1}. \tag{27}$$

Looking at Eq. (25) and Eq. (27), we see that because $\delta < 1$, the bound of Eq. (25) dominates, and so we have that

$$\sqrt{\frac{mm'}{l_{x,j} l_{y,j}}} \geq \sqrt{\frac{|\mathbf{S}_X||\mathbf{S}_Y|}{|\mathbf{S}_X||\mathbf{S}_Y| N^{-\delta}}} = N^{\delta/2}. \tag{28}$$

Using the contrapositive of Lemma 1, if an algorithm G makes less than q queries to an oracle $\mathcal{F}_S$ where S is promised to be in $\boldsymbol{S}_X$ or $\boldsymbol{S}_Y$, there exists at least one element of $\boldsymbol{S}_X$ and one element of $\boldsymbol{S}_Y$ such that the probability of distinguishing between the corresponding oracles less than is $1/2 + \epsilon$, where

$$\frac{1}{2}\sqrt{2N^{-\delta/2} q} > \epsilon. \tag{29}$$

Equivalently, there exists at least one element of $\boldsymbol{S}_X$ and one element of $\boldsymbol{S}_Y$ such that in order for $\mathcal{A}$ to distinguish them with constant bias, one requires $\Omega(N^{\delta/2})$ queries. □

7 Oracle Separation of QMA and QCMA

In this section, we prove an oracle separation between QMA and QCMA. In particular, we show

Theorem 3. *There exists a randomized-preimage-correct oracle $\boldsymbol{\mathcal{O}}$, and a language $L_{\boldsymbol{\mathcal{O}}}$ as in Theorem 1 such that $L_{\boldsymbol{\mathcal{O}}} \notin \mathsf{QCMA}^{\boldsymbol{\mathcal{O}}}$.*

Combined with Theorem 1, this gives the desired separation.

First, for convenience we define the complexity class $\mathsf{QCMA}_{\mathsf{exp,poly}}$ to be the analogue of QCMA, in which the quantum verifier is allowed exponential time and space, but still receives a polynomial length classical witness. Note that although trivially $\mathsf{EXP} = \mathsf{QCMA}_{\mathsf{exp,poly}}$, we will be considering the query complexity of an $\mathsf{QCMA}_{\mathsf{exp,poly}}$ machine, which in general is not the same as the query complexity of an EXP machine.

Our proof works as follows. We first show that if there is a QCMA machine that decides $L_{\boldsymbol{\mathcal{O}}}$, for all randomized-preimage-correct oracles $\boldsymbol{\mathcal{O}}$, then there will be a $\mathsf{QCMA}_{\mathsf{exp,poly}}$ machine that solves a related oracle problem using polynomial number of queries. Then using Theorem 2, we show that there is no such $\mathsf{QCMA}_{\mathsf{exp,poly}}$ machine that solves the related problem in a query-efficient way. This implies that there is no QCMA machine that solves the original problem.

Definition 5 (preimage-correct oracles)**.** *Let $\boldsymbol{\mathcal{O}}$ be a countably infinite set of unitaries: $\boldsymbol{\mathcal{O}} = \{\mathcal{O}_1, \mathcal{O}_2, \dots\}$, where each $\mathcal{O}_n$ implements an $(2n)$-qubit unitary. We say that $\boldsymbol{\mathcal{O}}$ is preimage-correct if for every n, $\mathcal{O}_n = \mathcal{P}_\sigma$, for some σ such that $S_{pre}(\sigma) \in \mathbf{S}^n_{even} \cup \mathbf{S}^n_{odd}$.*

The definition of preimage-correct oracles is very similar to that of randomized-preimage-correct oracles in Definition 2, except there is no randomness in preimage-correct oracles - they are unitaries.

We first prove the reduction to the related oracle problem:

Lemma 6. *Given a randomized-preimage-correct oracle $\boldsymbol{\mathcal{O}}$, let $1^n \in L_{\boldsymbol{\mathcal{O}}}$ if $\mathscr{O}_n = \mathscr{P}_{\boldsymbol{\sigma}_{pre}(S)}$ with $S \in \boldsymbol{S}^n_{even}$. Given a preimage-correct oracle $\tilde{\boldsymbol{\mathcal{O}}}$ let $1^n \in L_{\boldsymbol{\mathcal{O}}}$ if $\mathcal{O}_n = \mathcal{P}_\sigma$ with $S_{pre}(\sigma) \in \mathbf{S}^n_{even}$. Then if there is a* QCMA *machine M that decides $L_{\boldsymbol{\mathcal{O}}}$ for every randomized-preimage-correct $\boldsymbol{\mathcal{O}}$, then there is an* $\mathsf{QCMA}_{\mathsf{exp,poly}}$ *machine $\tilde{M}$ that decides $L_{\tilde{\boldsymbol{\mathcal{O}}}}$ for every preimage-correct $\tilde{\boldsymbol{\mathcal{O}}}$ such that $\tilde{M}$ uses at most a polynomial number of queries to $\tilde{\boldsymbol{\mathcal{O}}}$, and on input 1^n takes as input a classical witness w that depends only on $S_{pre}(\sigma)$.*

Proof. For each input 1^n, M applies an algorithm that takes as input a standard basis state. Because S completely characterizes $\mathscr{P}_{\boldsymbol{\sigma}_{\text{pre}}(S)}$, the optimal witness will depend only on S.

Suppose on input 1^n to M, the algorithm is the following:

$$\mathscr{L}_{AB} \circ (\mathscr{O})_A \circ (\mathscr{U}_t)_{AB} \circ \cdots \circ (\mathscr{U}_2)_{AB} \circ (\mathscr{O})_A \circ (\mathscr{U}_1)_{AB}(|w\rangle\langle w| \otimes |\psi_0\rangle\langle\psi_0|)_{AB} \tag{30}$$

where $|w\rangle\langle w|$ is the witness state (that depends only on S) in the standard basis and $\mathscr{U}_i$ are fixed unitaries and $\mathscr{L}$. The two subspaces A and B refer to the subset where the oracle acts (A) and the rest of the workspace (B). The two subspaces do *not* refer to the tensor product structure of the initial state.

For $i \in [N!(N^2-N)!]$ let $\boldsymbol{\tau}^n = \{\tau_i\}$ be the set of permutations on the elements of $[N^2]$ that do not mix the first N elements with the last N^2-N elements. Then let $\mathcal{P}^{\text{C}}_n$ be the following control-permutation:

$$\mathcal{P}^{\text{C}}_n|i\rangle|j\rangle = \begin{cases} |i\rangle|\tau_i(j)\rangle \text{ for } i \in [N!(N^2-N)!] \\ |i\rangle|j\rangle \text{ otherwise.} \end{cases} \tag{31}$$

$\mathscr{P}^{\text{C}}_n$ is the respective CPTP map.

$\mathcal{P}^{\text{C}}_n$ is a completely known unitary that is independent of the oracle, however, we do not know how to implement this unitary in polynomial time. This unitary is the reason we consider the class $\mathsf{QCMA}_{\mathsf{exp,poly}}$ in this proof rather than the more standard QCMA. Ultimately, we care about query complexity - not the complexity of the unitaries that occur between the oracle applications.

Let

$$|\chi_n\rangle = \frac{1}{\sqrt{N!(N^2-N)!}} \sum_{i=1}^{N!(N^2-N)!} |i\rangle \tag{32}$$

Then on input 1^n we have $\tilde{M}$ implement the algorithm

$$\begin{aligned}\mathscr{L}_{AB} \circ (\mathscr{P}^{\text{C}}_n)_{C_tA} \circ (\mathscr{O})_A \circ (\mathscr{U}_t)_{AB} \circ \cdots \\ \circ (\mathscr{U}_2)_{AB} \circ (\mathscr{P}^{\text{C}}_n)_{C_1A} \circ (\mathscr{O})_A \circ (\mathscr{U}_1)_{AB} \left(|\chi_n\rangle\langle\chi_n|^t_C \otimes (|w\rangle\langle w| \otimes |\psi_0\rangle\langle\psi_0|)_{AB}\right)\end{aligned} \tag{33}$$

where $(\mathscr{P}^{\text{C}}_\tau)_{C_iA}$ means the C_ith register controls the Ath register, and initially, the C_ith register is the ith copy of $|\chi_n\rangle$, and $\mathscr{O}$ is the CPTP version of the oracle $\mathcal{O}$.

Let $\rho_i(\mathscr{O})$ (resp. $\tilde{\rho}_i(\mathcal{O})$) be the state of the system during the algorithm M (resp. $\tilde{M}$) after the ith use of the oracle. Let $\rho_0(\mathscr{O})$ (resp. $\tilde{\rho}_0(\mathcal{O})$) be the initial state of the respective algorithms. Then we will show that

$$\rho_i(\mathscr{P}_{\boldsymbol{\sigma}_{\text{pre}}(S_{\text{pre}}(\sigma))}) = \text{tr}_{C_1,\ldots,C_t}\left(\tilde{\rho}_i(\mathcal{P}_\sigma)\right). \tag{34}$$

As a consequence of this, the probability distribution of measurement outcome of the two algorithms will be identical.

We prove this by induction. For the initial step, we have

$$\rho_0(\mathscr{P}_{\boldsymbol{\sigma}_{\mathrm{pre}}(S_{\mathrm{pre}}(\sigma))}) = |w\rangle\langle w| \otimes |\psi_0\rangle\langle\psi_0| \tag{35}$$

while

$$\begin{aligned}\mathrm{tr}_C(\tilde{\rho}_0(\mathcal{P}_\sigma)) =& \mathrm{tr}_C\left(|\chi_n\rangle\langle\chi_n|_C^t \otimes (|w\rangle\langle w| \otimes |\psi_0\rangle\langle\psi_0|)_{AB}\right)\\ =& |w\rangle\langle w| \otimes |\psi_0\rangle\langle\psi_0|.\end{aligned} \tag{36}$$

For the induction step, we need to show

$$\rho_k(\mathscr{P}_{\boldsymbol{\sigma}_{\mathrm{pre}}(S_{\mathrm{pre}}(\sigma))}) = \mathrm{tr}_{C_1,\ldots,C_t}\left(\tilde{\rho}_k(\mathcal{P}_\sigma)\right). \tag{37}$$

Because $\mathscr{P}_{\boldsymbol{\sigma}_{\mathrm{pre}}(S_{\mathrm{pre}}(\sigma))}$ has an equal probability of applying $\mathcal{P}_\sigma$ for each σ such that $S(\sigma) = S$, we have

$$\begin{aligned}\rho_k(\mathscr{P}_{\boldsymbol{\sigma}_{\mathrm{pre}}(S_{\mathrm{pre}}(\sigma))}) =& \mathscr{P}_{\boldsymbol{\sigma}_{\mathrm{pre}}(S_{\mathrm{pre}}(\sigma))}\left(\mathcal{U}_k\rho_{k-1}(\mathscr{P}_{\boldsymbol{\sigma}_{\mathrm{pre}}(S_{\mathrm{pre}}(\sigma))})\mathcal{U}_k^\dagger\right)\\ =& \frac{1}{N!(N^2-N)!}\sum_{i=1}^{N!(N^2-N)!} \mathcal{P}_{\tau_i}\mathcal{P}_\sigma\mathcal{U}_k\rho_{k-1}(\mathscr{P}_{\boldsymbol{\sigma}_{\mathrm{pre}}(S_{\mathrm{pre}}(\sigma))})\mathcal{U}_k^\dagger\mathcal{P}_\sigma^\dagger\mathcal{P}_{\tau_i}^\dagger.\end{aligned} \tag{38}$$

On the other hand

$$\begin{aligned}\mathrm{tr}_C\left(\tilde{\rho}_k(\mathcal{P}_\sigma)\right) =& \mathrm{tr}_C\left((\mathcal{P}_\tau^{\mathrm{C}})_{C_kA}\mathcal{P}_\sigma\mathcal{U}_k(\tilde{\rho}_{k-1}(\mathcal{P}_\sigma))\mathcal{U}_k^\dagger\mathcal{P}_\sigma^\dagger(\mathcal{P}_\tau^{\mathrm{C}})^\dagger_{C_kA}\right)\\ =& \frac{1}{N!(N^2-N)!}\sum_{i=1}^{N!(N^2-N)!} \mathcal{P}_{\tau_i}\mathcal{P}_\sigma\mathcal{U}_k\,\mathrm{tr}_C\left(\tilde{\rho}_{k-1}(\mathcal{P}_\sigma)\right)\mathcal{U}_k^\dagger\mathcal{P}_\sigma^\dagger\mathcal{P}_{\tau_i}^\dagger\end{aligned} \tag{39}$$

Now we need to show $\tilde{M}$ decides $L_{\boldsymbol{\mathcal{O}}}$ for a preimage-correct oracle $\boldsymbol{\mathcal{O}}$. Let's consider an input 1^n. Suppose $\mathcal{O}_n = \mathcal{P}_\sigma$, where $S_{\mathrm{pre}}(\sigma) \in \mathbf{S}^n_{\mathrm{even}}$. Then because M decides $L_{\boldsymbol{\mathcal{O}}}$ for any randomized-preimage-correct, there exists a witness w that depends only on $S_{\mathrm{pre}}(\sigma)$ such that when the oracle is $\mathscr{P}_{\boldsymbol{\sigma}_{\mathrm{pre}}(S_{\mathrm{pre}}(\sigma))}$ the output of M is 1 with probability at least $2/3$. Using the same witness w, $\tilde{M}$ will therefore produce output 1 with probability at least $2/3$.

Now consider an input 1^n such that $\mathcal{O}_n = \mathcal{P}_\sigma$ where $S_{\mathrm{pre}}(\sigma) \in \mathbf{S}^n_{\mathrm{odd}}$. Because M decides $L_{\boldsymbol{\mathcal{O}}}$ for a randomized-preimage-correct oracle $\boldsymbol{\mathcal{O}}$, when M is run with the oracle $\mathscr{P}_{\boldsymbol{\sigma}_{\mathrm{pre}}(S_{\mathrm{pre}}(\sigma))}$, for any witness w, M will output 1 with probability at most $1/3$. But because $\tilde{M}$ will have the same probability distribution of outcomes, this means that for any witness w to $\tilde{M}$, with oracle $\mathcal{P}_\sigma$, $\tilde{M}$ will output 1 with probability at most $1/3$. $\square$

Lemma 7. *There exists a preimage-correct $\boldsymbol{\mathcal{O}}$ such that there is no $\mathsf{QCMA}_{\mathsf{exp,poly}}^{\boldsymbol{\mathcal{O}}}$ machine M that decides $L_{\boldsymbol{\mathcal{O}}}$ using a polynomial number of queries, where the classical witness on input 1^n depends only on $S_{pre}(\sigma)$, when $\mathcal{O}_n = \mathcal{P}_\sigma$.*

Note that Lemma 7, combined with the contrapositive of Lemma 6, proves Theorem 3.

To prove Lemma 7, we will use Theorem 2. While the statement of Theorem 2 refers to the class QCMA, because Condition 4 is a statement about query complexity rather than time or space complexity, the result applies equally well to characterizing the query complexity of the class $\mathsf{QCMA}_{\mathsf{exp,poly}}$. Thus we need to prove the conditions of Theorem 2 apply to this problem.

For Condition 1, we need to associate each oracle with a subset. We will associate the oracle $\mathcal{P}_\sigma$ with the subset $S_{\text{pre}}(\sigma)$. Then the rest of Condition 1 is immediately satisfied for preimage-correct oracle languages.

From Lemma 6, we only need to consider witnesses that only depend on $S_{\text{pre}}(\sigma)$, so Condition 2 is satisfied.

We prove Conditions 3 and 4 hold in Lemmas 8 and 9. These proofs are quite similar to the proofs of Lemmas 4 and 5, so we move the proofs to Appendix B, and provide brief proof sketches in the main body of the paper.

Lemma 8. *Let $0 < \alpha < 1/2$ be a constant and $p(\cdot)$ be a polynomial function. Then there exists a positive integer $n^*(p, \alpha)$, such that for every $n > n^*(p, \alpha)$, and every subset $\mathbf{S} \subset \mathbf{S}^n_{\text{even}}$ such that $|\mathbf{S}| \geq |\mathbf{S}^n_{\text{even}}|2^{-p(n)}$, there exists a subset $\mathbf{S}' \subset \mathbf{S}$ such that $\mathbf{S}'$ is $(\alpha, \mathbf{S}^n_{\text{odd}})$-distributed.*

The proof strategy of Lemma 8 is identical to Lemma 4, and involves the same Fixing Procedure. However, the details are slightly more complex because we must deal with fixing even and odd elements.

Lemma 9. *Suppose $\mathbf{S}_X \subset \mathbf{S}^n_{even}$ is $(\delta, \mathbf{S}^n_{\text{odd}})$-distributed. Then for every algorithm G, there exists a permutation σ_x with $S_{pre}(\sigma_x) \in \mathbf{S}_X$, and a permutation σ_y with $S_{pre}(\sigma_y) \in \mathbf{S}^n_{odd}$, such that, given oracle access to $\mathcal{P}_{\sigma_x}$ or $\mathcal{P}_{\sigma_y}$, G can not distinguish them with probability ϵ without using $\left(1 - 2\sqrt{\epsilon(1-\epsilon)}\right) N^{\delta/2}$ queries.*

Again, the proof strategy of Lemma 9 is nearly identical to Lemma 5. The main difference is a more complex relation $\mathbf{R}$ to which we apply the adversary bound. The challenge here is that we can have two subsets S_x and S_y that are similar to each other, but there exists permutations σ_x and σ_y that are extremely dissimilar such that $S_{\text{pre}}(\sigma_x) = S_x$ and $S_{\text{pre}}(\sigma_y) = S_y$. We would like to create a relationship $\mathbf{R}$ that connects similar permutations, while only having information about the structure of the related subsets. To address this problem, we notice that for any two subsets S_x and S_y, we can create a one to one matching between the elements of $\boldsymbol{\sigma}_{\text{pre}}(S_x)$ and the elements of $\boldsymbol{\sigma}_{\text{pre}}(S_y)$ such that each permutation is matched with a similar permutation. Using this one-to-one matching, we can create a relationship $\mathbf{R}$ between permutations that inherits the properties of the related subsets.

As an immediate corollary of Theorem 1 and Theorem 3, there exists a randomized-preimage-correct oracle $\boldsymbol{\mathcal{O}}$ and language $L_{\boldsymbol{\mathcal{O}}}$ such that $L \notin \mathsf{QCMA}^{\boldsymbol{\mathcal{O}}}$ but $L \in \mathsf{QMA}^{\boldsymbol{\mathcal{O}}}$, and so $\mathsf{QMA}^{\boldsymbol{\mathcal{O}}} \not\subseteq \mathsf{QCMA}^{\boldsymbol{\mathcal{O}}}$ as desired.

8 Acknowledgments

We are grateful for multiple discussions with Stephen Jordan regarding permutation oracle verification strategies. We appreciate the many people who discussed this project with us, including Scott Aaronson, David Gosset, Gus Gutoski, Yi-Kai Liu, Ronald de Wolf, Robin Kothari, Dvir Kafri, and Chris Umans.

References

[1] Complexity zoo. https://complexityzoo.uwaterloo.ca/Complexity_Zoo.

[2] Scott Aaronson. Quantum lower bound for the collision problem. In John H. Reif, editor, *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*, pages 635–642. ACM, 2002.

[3] Scott Aaronson and Greg Kuperberg. Quantum versus classical proofs and advice. In *Computational Complexity, 2007. CCC'07. Twenty-Second Annual IEEE Conference on*, pages 115–128. IEEE, 2007.

[4] Scott Aaronson and Yaoyun Shi. Quantum lower bounds for the collision and the element distinctness problems. *J. ACM*, 51(4):595–605, July 2004.

[5] Dorit Aharonov and Tomer Naveh. Quantum np-a survey. *arXiv preprint quant-ph/0210077*, 2002.

[6] Andris Ambainis. Quantum lower bounds by quantum arguments. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 636–643. ACM, 2000.

[7] Andris Ambainis, Loïck Magnin, Martin Roetteler, and Jérémie Roland. Symmetry-assisted adversaries for quantum state generation. In *Computational Complexity (CCC), 2011 IEEE 26th Annual Conference on*, pages 167–177. IEEE, 2011.

[8] Aleksandrs Belovs. Variations on quantum adversary. *arXiv preprint 1504.06943*, 2015.

[9] J Niel De Beaudrap, Richard Cleve, John Watrous, et al. Sharp quantum versus classical query complexity separations. *Algorithmica*, 34(4):449–461, 2002.

[10] Shafi Goldwasser and Michael Sipser. Private coins versus public coins in interactive proof systems. In Juris Hartmanis, editor, *Proceedings of the 18th Annual ACM Symposium on Theory of Computing, May 28-30, 1986, Berkeley, California, USA*, pages 59–68. ACM, 1986.

[11] Alex B Grilo, Iordanis Kerenidis, and Jamie Sikora. Qma with subset state witnesses. *arXiv preprint arXiv:1410.2882*, 2014.

[12] Aram W Harrow and David J Rosenbaum. Uselessness for an oracle model with internal randomness. *Quantum Information & Computation*, 14(7&8):608–624, 2014.

[13] Peter Hoyer, Troy Lee, and Robert Spalek. Negative weights make adversaries stronger. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 526–535. ACM, 2007.

[14] Alexei Kitaev. Quantum computation: Algorithms and error correction. *Russian Math. Surveys*, 52(6):1191–1249, 1997.

[15] Alexei Yu Kitaev, Alexander Shen, and Mikhail N Vyalyi. *Classical and quantum computation.* Number 47. American Mathematical Soc., 2002.

[16] Troy Lee, Rajat Mittal, Ben W Reichardt, Robert Spalek, and Mario Szegedy. Quantum query complexity of state conversion. In *Foundations of Computer Science (FOCS), 2011 IEEE 52nd Annual Symposium on*, pages 344–353. IEEE, 2011.

[17] Chris Marriott and John Watrous. Quantum arthur—merlin games. *Comput. Complex.*, 14(2):122–152, June 2005.

[18] Oded Regev and Liron Schiff. Impossibility of a quantum speed-up with a faulty oracle. In *Automata, Languages and Programming*, pages 773–781. Springer, 2008.

[19] Nikolay K. Vereshchagin. Oracle separation of complexity classes and lower bounds for perceptrons solving separation problems. *Izvestiya: Mathematics*, 59:1103–1122, December 1995.

A An Adversary Bound for Permutation Oracles

We will prove Lemma: 1

Lemma 1. *Let $\boldsymbol{\sigma} \subset [V] \to [V]$ be a subset of permutations acting on the elements $[V]$. Let $f : \boldsymbol{\sigma} \to \{0,1\}$ be a function of permutations. Let $\boldsymbol{\sigma}_X \subset \boldsymbol{\sigma}$ be a set such that if $\sigma \in \boldsymbol{\sigma}_X$, then $f(\sigma) = 1$. Let $\boldsymbol{\sigma}_Y \subset \boldsymbol{\sigma}$ be a set such that if $\sigma \in \boldsymbol{\sigma}_Y$ then $f(\sigma) = 0$. Let $R \subset \boldsymbol{\sigma}_X \times \boldsymbol{\sigma}_Y$ be such that*

- *For every $\sigma_x \in \boldsymbol{\sigma}_X$, there exists at least m different $\sigma_y \in \boldsymbol{\sigma}_Y$ such that $(\sigma_x, \sigma_y) \in R$.*
- *For every $\sigma_y \in \boldsymbol{\sigma}_Y$, there exists at least m' different $\sigma_x \in \boldsymbol{\sigma}_X$ such that $(\sigma_x, \sigma_y) \in R$.*
- *Let $l_{x,i}$ be the number of $\sigma_y \in \boldsymbol{\sigma}_Y$ such that $(\sigma_x, \sigma_y) \in R$ and $\sigma_x(i) \neq \sigma_y(i)$. Let $l_{y,i}$ be the number of $\sigma_x \in \boldsymbol{\sigma}_Y$ such that $(\sigma_x, \sigma_y) \in R$ and $\sigma_x(i) \neq \sigma_y(i)$. Then let $l_{max} = \max_{(\sigma_x,\sigma_y)\in R, i} l_{x,i} l_{y,i}$.*

Then given an in-place permutation oracle $\mathcal{P}_\sigma$ for $\sigma \in \boldsymbol{\sigma}$ that acts as

$$\mathcal{P}_\sigma |i\rangle = |\sigma(i)\rangle \tag{6}$$

any quantum algorithm that correctly evaluates $f(\sigma)$ with probability $1-\epsilon$ for every element of $\boldsymbol{\sigma}_X$ and $\boldsymbol{\sigma}_Y$ must use $\left(1 - 2\sqrt{\epsilon(1-\epsilon)}\right)\sqrt{\frac{mm'}{l_{max}}}$ queries to the oracle.

We note that this is identical to Ambainis' adversary bound for permutations (see Theorem 6 in [6]).

Proof. We assume that we have a control permutation oracle, that acts as

$$\mathcal{P}|x\rangle_C |i\rangle_A |z\rangle_Q = |x\rangle|\sigma_x(i)\rangle|z\rangle \tag{40}$$

where the Hilbert space $\mathcal{H}_C$ has dimension $|\boldsymbol{\sigma}|$,the Hilbert space $\mathcal{H}_A$ has dimension V and is where the permutation is carried out, and $\mathcal{H}_Q$ is a set of ancilla qubits.

Let $|\psi^t\rangle$ be the state of the system immediately after t uses of the control oracle. Let $|\varphi^t\rangle$ be the state of the system immediately before the t^{th} use of the control oracle. Let ρ^t be the reduced state of the system immediately after t uses of the control oracle, where systems A and Q have been traced out. That is $\rho^t = \operatorname{tr}_{AQ}(|\psi^t\rangle\langle\psi^t|)$. Let $\left(\rho^t\right)_{xy}$ be the $(x,y)^{\text{th}}$ element of the density matrix. Then we will track the progress of the following measure:

$$W^t = \sum_{(\sigma_x,\sigma_y)\in R} \left|(\rho^t)_{xy}\right|. \tag{41}$$

Notice that unitaries that only act on the subsystems Q and A do not affect W^t.

If the state before the first use of the oracle is

$$|\psi^0\rangle = \left(\frac{1}{\sqrt{2|\boldsymbol{\sigma}_X|}} \sum_{\sigma_x\in\boldsymbol{\sigma}_X} |x\rangle + \frac{1}{\sqrt{2|\boldsymbol{\sigma}_Y|}} \sum_{\sigma_y\in\boldsymbol{\sigma}_Y} |y\rangle\right) \otimes |\phi\rangle_{AQ}, \tag{42}$$

then following Ambainis (e.g. Theorem 2 [6]), we have that for an algorithm to succeed with probability at least $1-\epsilon$ after T uses of the oracle, we must have

$$W^0 - W^T > \left(1 - 2\sqrt{\epsilon(1-\epsilon)}\right)\sqrt{mm'} \tag{43}$$

Now we calculate how much W^t can change between uses of the oracle. Suppose without loss of generality that

$$|\varphi^t\rangle = \frac{1}{\sqrt{2|\boldsymbol{\sigma}_X|}} \sum_{\sigma_x \in \boldsymbol{\sigma}_X} \sum_{i,z} \alpha_{x,i,z} |x,i,z\rangle_{CAQ} + \frac{1}{\sqrt{2|\boldsymbol{\sigma}_Y|}} \sum_{\sigma_y \in \boldsymbol{\sigma}_Y} \sum_{i,z} \alpha_{y,i,z} |y,i,z\rangle_{CAQ}. \tag{44}$$

Then we have

$$\begin{aligned} |\psi^t\rangle &= \frac{1}{\sqrt{2|\boldsymbol{\sigma}_X|}} \sum_{\sigma_x \in \boldsymbol{\sigma}_X} \sum_{i,z} \alpha_{x,i,z} |x,\sigma_x(i),z\rangle_{CAQ} + \frac{1}{\sqrt{2|\boldsymbol{\sigma}_Y|}} \sum_{\sigma_y \in \boldsymbol{\sigma}_Y} \sum_{i,z} \alpha_{y,i,z} |y,\sigma_y(i),z\rangle_{CAQ} \\ &= \frac{1}{\sqrt{2|\boldsymbol{\sigma}_X|}} \sum_{\sigma_x \in \boldsymbol{\sigma}_X} \sum_{i,z} \alpha_{x,\sigma_x^{-1}(i),z} |x,i,z\rangle_{CAQ} + \frac{1}{\sqrt{2|\boldsymbol{\sigma}_Y|}} \sum_{\sigma_y \in \boldsymbol{\sigma}_Y} \sum_{i,z} \alpha_{y,\sigma_y^{-1}(i),z} |y,i,z\rangle_{CAQ}. \end{aligned} \tag{45}$$

Hence for $(\sigma_x, \sigma_y) \in R$, we have

$$\begin{aligned} (\rho^t)_{xy} &= \frac{1}{2\sqrt{|\boldsymbol{\sigma}_X||\boldsymbol{\sigma}_Y|}} \sum_{i,z} \alpha_{x,\sigma_x^{-1}(i),z} \alpha^*_{y,\sigma_y^{-1}(i),z} \\ (\rho^{t-1})_{xy} &= \frac{1}{2\sqrt{|\boldsymbol{\sigma}_X||\boldsymbol{\sigma}_Y|}} \sum_{i,z} \alpha_{x,i,z} \alpha^*_{y,i,z}, \end{aligned} \tag{46}$$

where $(\cdot)^*$ signifies the complex conjugate. Now we can calculate $W^t - W^{t-1}$:

$$\begin{aligned} W^{t-1} - W^t &= \sum_{(\sigma_x,\sigma_y)\in R} |(\rho^{t-1})_{xy}| - |(\rho^t)_{xy}| \\ &\le \sum_{(\sigma_x,\sigma_y)\in R} |(\rho^{t-1})_{xy} - (\rho^t)_{xy}|. \end{aligned} \tag{47}$$

From Eq. (46), we see that whenever $\sigma_x^{-1}(i) = \sigma_y^{-1}(i)$, we have a cancellation between the corresponding elements of $(\rho^t)_{xy}$ and $(\rho^{t-1})_{xy}$. However, when $\sigma_x^{-1}(i) \neq \sigma_y^{-1}(i)$, terms do not cancel. To see this more explicitly, we rewrite Eq. (47) as

$$\begin{aligned} W^{t-1} - W^t \le & \frac{1}{2\sqrt{|\boldsymbol{\sigma}_X||\boldsymbol{\sigma}_Y|}} \sum_{(\sigma_x,\sigma_y)\in R} \left| \sum_z \left[\sum_{i:\sigma_x(i)=\sigma_y(i)} \alpha_{x,i,z}\alpha^*_{y,i,z} + \sum_{i:\sigma_x(i)\neq\sigma_y(i)} \alpha_{x,i,z}\alpha^*_{y,i,z} \right.\right. \\ & \left.\left. - \sum_{i:\sigma_x^{-1}(i)=\sigma_y^{-1}(i)} \alpha_{x,\sigma_x^{-1}(i),z}\alpha^*_{y,\sigma_y^{-1}(i),z} - \sum_{i:\sigma_x^{-1}(i)\neq\sigma_y^{-1}(i)} \alpha_{x,\sigma_x^{-1}(i),z}\alpha^*_{y,\sigma_y^{-1}(i),z} \right] \right|. \end{aligned} \tag{48}$$

Consider the sets $T_{x,y} = \{i : \sigma_x(i) = \sigma_y(i)\}$ and $U_{x,y} = \{\sigma_x^{-1}(i) : \sigma_x^{-1}(i) = \sigma_y^{-1}(i)\}$. We will show $U_{x,y} = T_{x,y}$. Suppose $i \in T_{x,y}$. Then $\sigma_x(i) = \sigma_y(i) = i'$, for some i'. But that implies $\sigma_x^{-1}(i') = \sigma_y^{-1}(i') = i$, so $\sigma_x^{-1}(i') = i \in U_{x,y}$, and thus $T_{x,y} \subset U_{x,y}$. The opposite direction is shown similarly. Therefore, those two sums in Eq. (48) cancel, and, moving the summation over z and i outside the absolute values by the triangle inequality, we are left with

$$W^{t-1} - W^t \le \frac{1}{2\sqrt{|\boldsymbol{\sigma}_X||\boldsymbol{\sigma}_Y|}} \sum_{z,(\sigma_x,\sigma_y)\in R} \left(\sum_{i:\sigma_x(i)\neq\sigma_y(i)} \left|\alpha_{x,i,z}\alpha^*_{y,i,z}\right| + \sum_{i:\sigma_x^{-1}(i)\neq\sigma_y^{-1}(i)} \left|\alpha_{x,\sigma_x^{-1}(i),z}\alpha^*_{y,\sigma_y^{-1}(i),z}\right| \right). \tag{49}$$

Now we use the AM-GM to bound the terms in the absolute values:

$$\begin{aligned}W^{t-1} - W^t \leq& \frac{1}{2}\sum_{z,(\sigma_x,\sigma_y)\in R}\left(\sum_{i:\sigma_x(i)\neq\sigma_y(i)}\left(\sqrt{\frac{l_{y,i}}{l_{x,i}}}\frac{|\alpha_{x,i,z}|^2}{2|\boldsymbol{\sigma}_X|} + \sqrt{\frac{l_{x,i}}{l_{y,i}}}\frac{|\alpha^*_{y,i,z}|^2}{2|\boldsymbol{\sigma}_Y|}\right)\right)\\ &+ \frac{1}{2}\sum_{z,(\sigma_x,\sigma_y)\in R}\left(\sum_{i:\sigma_x^{-1}(i)\neq\sigma_y^{-1}(i)}\left(\sqrt{\frac{l_{y,i}}{l_{x,i}}}\frac{\left|\alpha_{x,\sigma_x^{-1}(i),z}\right|^2}{2|\boldsymbol{\sigma}_X|} + \sqrt{\frac{l_{x,i}}{l_{y,i}}}\frac{\left|\alpha^*_{y,\sigma_y^{-1}(i),z}\right|^2}{2|\boldsymbol{\sigma}_Y|}\right)\right)\end{aligned} \tag{50}$$

We now show that for $(\sigma_x, \sigma_y) \in R$,

$$\begin{aligned}\sum_{i:\sigma_x^{-1}(i)\neq\sigma_y^{-1}(i)} |\alpha_{x,\sigma_x^{-1}(i),z}|^2 &= \sum_{i:\sigma_x(i)\neq\sigma_y(i)} |\alpha_{x,i,z}|^2,\\ \sum_{i:\sigma_x^{-1}(i)\neq\sigma_y^{-1}(i)} |\alpha_{y,\sigma_y^{-1}(i),z}|^2 &= \sum_{i:\sigma_x(i)\neq\sigma_y(i)} |\alpha_{y,i,z}|^2.\end{aligned} \tag{51}$$

We prove the first equality, and the second is proven similarly. We define

$$\begin{aligned}T'_{x,y} &= [V]/T_{x,y},\\ U'_{x,y} &= [V]/U_{x,y}.\end{aligned} \tag{52}$$

Looking at the definition of $T_{x,y}$ and $U_{x,y}$, we see that

$$\begin{aligned}T'_{x,y} &= \{i : \sigma_x(i) \neq \sigma_y(i)\}\\ U'_{x,y} &= \{\sigma_x^{-1}(i) : \sigma_x^{-1}(i) \neq \sigma_y^{-1}(i)\}.\end{aligned} \tag{53}$$

We previously showed $T_{x,y} = U_{x,y}$, so we have $T'_{x,y} = U'_{x,y}$. Therefore

$$\sum_{i:\sigma_x^{-1}(i)\neq\sigma_y^{-1}(i)} |\alpha_{x,\sigma_x^{-1}(i),z}|^2 = \sum_{i:U'_{x,y}} |\alpha_{x,j,z}|^2 = \sum_{i:T'_{x,y}} |\alpha_{x,i,z}|^2 \sum_{i:\sigma_x(i)\neq\sigma_y(i)} |\alpha_{x,i,z}|^2. \tag{54}$$

Thus, Eq. (50) becomes

$$W^{t-1} - W^t \leq \sum_{z,(\sigma_x,\sigma_y)\in R}\left(\sum_{i:\sigma_x(i)\neq\sigma_y(i)}\left(\sqrt{\frac{l_{y,i}}{l_{x,i}}}\frac{|\alpha_{x,i,z}|^2}{2|\boldsymbol{\sigma}_X|} + \sqrt{\frac{l_{x,i}}{l_{y,i}}}\frac{|\alpha^*_{y,i,z}|^2}{2|\boldsymbol{\sigma}_Y|}\right)\right). \tag{55}$$

Now we switch the order of summation and then use the definition of $l_{x,i}$ and $l_{y,i}$ to get

$$\begin{aligned}W^{t-1} - W^t &\leq \sum_{i\in[V],z}\left(\sum_{(\sigma_x,\sigma_y)\in R:\sigma_x(i)\neq\sigma_y(i)}\left(\sqrt{\frac{l_{y,i}}{l_{x,i}}}\frac{|\alpha_{x,i,z}|^2}{2|\boldsymbol{\sigma}_X|} + \sqrt{\frac{l_{x,i}}{l_{y,i}}}\frac{|\alpha^*_{y,i,z}|^2}{2|\boldsymbol{\sigma}_Y|}\right)\right)\\ &\leq \sum_{i\in[V],z}\left(\sum_{\sigma_x\in\boldsymbol{\sigma}_X}\sqrt{l_{x,i}\max_{\sigma_y:(\sigma_x,\sigma_y)\in R} l_{y,i}}\frac{|\alpha_{x,i,z}|^2}{2|\boldsymbol{\sigma}_X|} + \sum_{\sigma_y\in\boldsymbol{\sigma}_Y}\sqrt{l_{y,i}\max_{\sigma_x:(\sigma_x,\sigma_y)\in R} l_{x,i}}\frac{|\alpha^*_{y,i,z}|^2}{2|\boldsymbol{\sigma}_Y|}\right)\end{aligned} \tag{56}$$

Finally, using the definition of l_{max} we have

$$\begin{aligned}W^{t-1} - W^t \leq& \sqrt{l_{max}}\sum_{i\in[V],z}\left(\sum_{x\in\boldsymbol{\sigma}_X}\frac{|\alpha_{x,i,z}|^2}{2|\boldsymbol{\sigma}_X|} + \sum_{\sigma_y\in\boldsymbol{\sigma}_Y}\frac{|\alpha^*_{y,i,z}|^2}{2|\boldsymbol{\sigma}_Y|}\right)\\ \leq& \sqrt{l_{max}},\end{aligned} \tag{57}$$

where we have used that Eq. (44) is a normalized state. □

B Proofs of Lemmas 8 and 9

Lemma 8. *Let $0 < \alpha < 1/2$ be a constant and $p(\cdot)$ be a polynomial function. Then there exists a positive integer $n^*(p, \alpha)$, such that for every $n > n^*(p, \alpha)$, and every subset $\mathbf{S} \subset \mathbf{S}^n_{\text{even}}$ such that $|\mathbf{S}| \geq |\mathbf{S}^n_{\text{even}}|2^{-p(n)}$, there exists a subset $\mathbf{S}' \subset \mathbf{S}$ such that $\mathbf{S}'$ is $(\alpha, \mathbf{S}^n_{\text{odd}})$-distributed.*

Proof. We prove the existence of $\mathbf{S}'$ by construction. Let $\mathbf{S}$ be any subset of $\mathbf{S}^n_{\text{even}}$ such that $|\mathbf{S}| \geq |\mathbf{S}^n_{\text{even}}|2^{-p(n)}$. We construct $\mathbf{S}'$ using the following procedure, which we call the fixing procedure.

Fixing Procedure

1. Set $\mathbf{S}' = \mathbf{S}$, and set $S_{\text{fixed}} = \emptyset$.
2. (a) Let $\nu(i)$ be the number of subsets $S \in \mathbf{S}'$ such that $i \in S$.

   (b) If there exists some element i for which
$$|\mathbf{S}'| > \nu(i) \geq |\mathbf{S}'|N^{-\alpha} \tag{58}$$
   set $\mathbf{S}' \leftarrow \{S : S \in \mathbf{S}' \text{ and } i \in S\}$, set $S_{\text{fixed}} \leftarrow S_{\text{fixed}} \cup i$, and return to step 2(a). Otherwise exit the Fixing Procedure.

By construction, $\mathbf{S}'$ will satisfy conditions (1) and (3) of Definition 3. Now we need to check condition (2) of Definition 3. Notice that since $\mathbf{S} \subset \mathbf{S}^n_{\text{even}}$, the maximum number of odd elements in S_{fixed} is $N/3$. Now we need to check that the Fixing Procedure must stop before fixing more than $N/3$ even items, so that S_{fixed} can be a subset of elements of $\mathbf{S}^n_{\text{odd}}$.

Let's suppose that at some point in the Fixing Procedure, for sets $\mathbf{S}'$ and S_{fixed}, we have $N/3$ even items fixed. Suppose for contradiction, that at this point, there is some even element i^* such that i^* appears in greater than $N^{-\alpha}$ fraction of $S \in \mathbf{S}'$. Let's also assume without loss of generality that $|S_{\text{fixed}} \cap \mathbb{Z}_{\text{odd}}| = k_{\text{odd}} \leq N/3$.

Let us look at the set
$$\mathbf{S}'' = \{S : S \in \mathbf{S}', i^* \in S\}. \tag{59}$$
Because $(S_{\text{fixed}} \cup i^*) \subset S$ for all $S \in \mathbf{S}''$, there are $N/3 - 1$ even elements that can be freely chosen for $S \in \mathbf{S}''$ and $N/3 - k_{\text{odd}}$ odd elements that can be freely chosen. Thus, we have
$$|\mathbf{S}''| \leq \binom{N^2/2 - N/3 - 1}{N/3 - 1}\binom{N^2/2 - k_{\text{odd}}}{N/3 - k_{\text{odd}}}. \tag{60}$$
By assumption
$$|\mathbf{S}''| \geq |\mathbf{S}'|N^{-\alpha}, \tag{61}$$

so

$$\begin{aligned}|\mathbf{S}'| &\leq \binom{N^2/2-N/3-1}{N/3-1}\binom{N^2/2-k_{\text{odd}}}{N/3-k_{\text{odd}}}N^\alpha \\ &\leq \binom{N^2/2}{N/3}\binom{N^2/2}{N/3}N^\alpha \\ &\leq (3Ne/2)^{2N/3}N^\alpha \\ &= 2^{2N/3(\log(3e/2)+\log N)+\log(N)\alpha} \\ &= 2^{O(N)+(2N/3)\log N}. \end{aligned} \tag{62}$$

However, we can also bound the size of $\mathbf{S}'$ from the Fixing Procedure. Notice that at every step of the Fixing Procedure, the size of the set $\mathbf{S}'$ is reduced by at most a factor $N^{-\alpha}$. Since we are assuming $N/3$ even elements are in S_{fixed} and $k_{\text{odd}} \leq N/3$ odd elements are in S_{fixed}, the Fixing Procedure can reduce the original set $\mathbf{S}$ by at most a factor $N^{-\alpha(2N/3)}$. Since $|\mathbf{S}| \geq |\mathbf{S}^n_{\text{even}}|2^{-p(n)}$, we have that at this point in the Fixing Procedure

$$\begin{aligned}|\mathbf{S}'| &\geq |\mathbf{S}^n_{\text{even}}|2^{-p(n)}N^{-\alpha(2N/3)} \\ &= \binom{N^2/2}{2N/3}\binom{N^2/2}{N/3}2^{-p(n)}N^{-\alpha(2N/3)} \\ &\leq (3N/4)^{2N/3}(3N/4)^{N/3}2^{-p(n)}N^{-\alpha(2N/3)} \\ &= 2^{2N/3(\log(3/4)+\log N)+N/3(\log(3/4)+\log N)-p(n)-\log(N)\alpha 2N/3} \\ &= 2^{N\log N(1-2\alpha/3)+N\log(3/4)-p(\log(N))} \\ &= 2^{-O(N)+N\log N(1-2\alpha/3)}. \end{aligned} \tag{63}$$

Notice that as long as $\alpha < 1/2$, for large enough N (in particular, for $N > 2^{n^*}$ for some positive integer n^*, where n^* depends on α and $p(\cdot)$), the bound of Eq. (63) will be larger than the bound of Eq. (62), giving a contradiction. Therefore, our assumption, must have been false, and at this point in the Fixing Procedure, all even elements $i \in [N^2]/S_{\text{fixed}}$ will appear in at most a fraction $N^{-\alpha}$ of $S \in \mathbf{S}'$. Thus at the next step of the Fixing Procedure, an even element will not be added to S_{fixed}, and the number of even elements in S_{fixed} will stay bounded by $N/3$. The same logic can be reapplied at future steps of the Fixing Procedure, even if additional odd items are added. □

Lemma 9. *Suppose $\mathbf{S}_X \subset \mathbf{S}^n_{even}$ is $(\delta, \mathbf{S}^n_{\text{odd}})$-distributed. Then for every algorithm G, there exists a permutation σ_x with $S_{pre}(\sigma_x) \in \mathbf{S}_X$, and a permutation σ_y with $S_{pre}(\sigma_y) \in \mathbf{S}^n_{odd}$, such that, given oracle access to $\mathcal{P}_{\sigma_x}$ or $\mathcal{P}_{\sigma_y}$, G can not distinguish them with probability ϵ without using $\left(1-2\sqrt{\epsilon(1-\epsilon)}\right)N^{\delta/2}$ queries.*

Proof. Since $\mathbf{S}_X$ is $(\delta, \mathbf{S}^n_{odd})$-distributed, there exists a set S_{fixed} of elements such that $S_{\text{fixed}} \subset S$ for all $S \in \mathbf{S}_X$, where S_{fixed} contains at most $N/3$ odd elements and at most $N/3$ even elements. (Otherwise Condition (2) of Definition 3 will not be satisfied.) We choose

$$\begin{aligned}\mathbf{S}_Y &= \{S : S \in \mathbf{S}^n_{\text{odd}}, S_{\text{fixed}} \subset S\}, \\ \boldsymbol{\sigma}_Y &= \{\sigma : S_{\text{pre}}(\sigma) \in \mathbf{S}_Y\}, \\ \boldsymbol{\sigma}_X &= \{\sigma : S_{\text{pre}}(\sigma) \in \mathbf{S}_X\}. \end{aligned} \tag{64}$$

We now define the relation $\mathbf{R}$ needed to apply our adversary bound. For each $(S_x, S_y) \in \mathbf{S}_X \times \mathbf{S}_Y$, we will create a one-to-one matching in $\mathbf{R}$ between the elements of $\boldsymbol{\sigma}_{\text{pre}}(S_x)$ and $\boldsymbol{\sigma}_{\text{pre}}(S_y)$. We first choose any element $\sigma^*_x \in \boldsymbol{\sigma}_{\text{pre}}(S_x)$. Then we choose a permutation $\sigma^*_y \in \boldsymbol{\sigma}_{\text{pre}}(S_y)$ such that

- $\forall j \in (S_x \cap S_y), \sigma_x^*(j) = \sigma_y^*(j)$,
- $\forall j \in [N^2]/(S_x \cup S_y), \sigma_x^*(j) = \sigma_y^*(j)$,
- $\forall j \in S_x/(S_x \cap S_y), \exists i \in S_y/(S_x \cap S_y)$ such that $\sigma_x^*(j) = \sigma_y^*(i)$ and $\sigma_x^*(i) = \sigma_y^*(j)$.

Since every permutation corresponding to S_y is in $\boldsymbol{\sigma}_{\text{pre}}(S_y)$, there will always be such a σ_y^* that satisfies the above criterion. We choose $(\sigma_x^*, \sigma_y^*) \in \mathbf{R}$.

For $i \in [N!(N^2-N)!]$ let $\boldsymbol{\tau}^n = \{\tau_i\}$ be the set of permutations on the elements of $[N^2]$ that do not mix the first N elements with the last $N^2 - N$ elements. By $\sigma_a \circ \sigma_b$, we mean apply first permutation σ_b, and then permutation σ_a. Notice that

$$\begin{aligned}\boldsymbol{\sigma}_{\text{pre}}(S_x) =& \{\tau \circ \sigma_x^* : \tau \in \boldsymbol{\tau}^n\} \\ \boldsymbol{\sigma}_{\text{pre}}(S_y) =& \{\tau \circ \sigma_y^* : \tau \in \boldsymbol{\tau}^n\}. \end{aligned} \tag{65}$$

Furthermore given $\tau \in \boldsymbol{\tau}^n$, we have

- $\forall j \in (S_x \cap S_y), \tau \circ \sigma_x^*(j) = \tau \circ \sigma_y^*(j)$,
- $\forall j \in [N^2]/(S_x \cup S_y), \tau \circ \sigma_x^*(j) = \tau \circ \sigma_y^*(j)$,
- $\forall j \in S_x/(S_x \cap S_y), \exists i \in S_y/(S_x \cap S_y)$ such that $\tau \circ \sigma_x^*(j) = \tau \circ \sigma_y^*(i)$ and $\sigma \circ \sigma_x^*(i) = \sigma \circ \sigma_y^*(j)$.

For every $\tau \in \boldsymbol{\tau}^n$, we set $(\tau \circ \sigma_x^*, \tau \circ \sigma_y^*) \in \mathbf{R}$. In doing so, we create a one-to-one correspondance in $\mathbf{R}$ between the elements of $\boldsymbol{\sigma}_{\text{pre}}(S_x)$ and $\boldsymbol{\sigma}_{\text{pre}}(S_y)$. We then repeat this process for all pairs $(S_x, S_y) \in \mathbf{S}_X \times \mathbf{S}_Y$. The end result is the $\mathbf{R}$ that we will use.

Now we need to analyze the properties of this $\mathbf{R}$. Notice that each $\sigma_x \in \boldsymbol{\sigma}_X$ is paired to exactly one element of $\boldsymbol{\sigma}_{\text{pre}}(S_y)$ for each $S_y \in \mathbf{S}_Y$. Thus $m = |\mathbf{S}_Y|$. Likewise $m' = |\mathbf{S}_X|$.

Now consider $(\sigma_x, \sigma_y) \in \mathbf{R}$. We consider some element j such that $\sigma_x(j) \neq \sigma_y(j)$. We first consider the case that $j \in S_x$. We upper bound $l_{x,j}$, the number of $\sigma_{y'}$ such that $(\sigma_x, \sigma_{y'}) \in \mathbf{R}$ and $\sigma_{y'}(j) \neq \sigma_x(j)$. To simplify analysis, we use the simple upper bound $l_{x,j} \leq |\mathbf{S}_Y|$, which is sufficient for our purposes. Next we need to upper bound $l_{y,j}$, the number of $\sigma_{x'}$ such that $(\sigma_{x'}, \sigma_y) \in \mathbf{R}$ and $\sigma_{x'}(j) \neq \sigma_y(j)$. By our construction of $\mathbf{R}$, we have $j \notin S_y$. Also, by construction, if $j \notin S_y$, $\sigma_{x'}(j) \neq \sigma_y(j)$ if and only if $j \in S_{x'}$. Since σ_y is paired to only one element σ_x for each set S_x, $l_{y,j}$ is bounded by the number of sets $S_x \in \mathbf{S}_X$ such that $j \in S_x$. Because $\mathbf{S}_X$ is $(\delta, \mathbf{S}^n_{odd})$-distributed, at most a fraction $N^{-\delta}$ of the sets of $\mathbf{S}_X$ can contain j, so $l_{y,j} \leq |\mathbf{S}_X| N^{-\delta}$. In this case we have

$$l_{x,j} l_{y,j} \leq |\mathbf{S}_X||\mathbf{S}_Y| N^{-\delta}. \tag{66}$$

We now consider the case that $j \in S_y$. We upper bound $l_{y,j}$, the number of $\sigma_{x'}$ such that $(\sigma_{x'}, \sigma_y) \in \mathbf{R}$ and $\sigma_{x'}(j) \neq \sigma_y(j)$. To simplify analysis, we use the upper bound of $l_{y,j} \leq |\mathbf{S}_X|$, which is sufficient for our analysis. Next we need to upper bound $l_{x,j}$, the number of $\sigma_{y'}$ such that $(\sigma_x, \sigma_{y'}) \in \mathbf{R}$ and $\sigma_{y'}(j) \neq \sigma_x(j)$. By our construction of $\mathbf{R}$, we have $j \notin S_x$. Also, by construction, if $j \notin S_x$, $\sigma_{y'}(j) \neq \sigma_x(j)$ if and only if $j \in S_{y'}$. Since σ_x is paired to only one element σ_x for each set S_x, $l_{x,j}$ is bounded by the number of sets $S_y \in \mathbf{S}_Y$ such that $j \in S_y$. Suppose S_{fixed} contains k_{even} even elements and k_{odd} odd elements. If j is odd, we have

$$\begin{aligned} l_{x,j} =& \binom{N^2/2 - k_{\text{odd}} - 1}{2N/3 - k_{\text{odd}} - 1}\binom{N^2/2 - k_{\text{even}}}{N/3 - k_{\text{even}}} \\ \leq& \frac{2N/3}{N^2/2 - N/3}|\mathbf{S}_Y|, \end{aligned} \tag{67}$$

while if j is even (in that case, we must have $k_{\text{even}} < N/3$), we have

$$l_{x,j} = \binom{N^2/2 - k_{\text{odd}}}{2N/3 - k_{\text{odd}}} \binom{N^2/2 - k_{\text{even}} - 1}{N/3 - k_{\text{even}} - 1} \\ \leq \frac{N/3}{N^2/2 - N/3} |\mathbf{S}_Y|, \tag{68}$$

where we've used that

$$|\mathbf{S}_Y| = \binom{N^2/2 - k_{\text{odd}}}{2N/3 - k_{\text{odd}}} \binom{N^2/2 - k_{\text{even}}}{N/3 - k_{\text{even}}}. \tag{69}$$

Therefore in this case, we have

$$l_{x,j} l_{y,j} = |\mathbf{S}_X||\mathbf{S}_Y| O(N^{-1}). \tag{70}$$

Looking at Eq. (66) and Eq. (70), we see that because $\delta < 1$, the bound of Eq. (66) dominates, and so we have that

$$\sqrt{\frac{mm'}{l_{x,j} l_{y,j}}} \geq \sqrt{\frac{|\mathbf{S}_X||\mathbf{S}_Y|}{|\mathbf{S}_X||\mathbf{S}_Y| N^{-\delta}}} = N^{\delta/2}. \tag{71}$$

Using the contrapositive of Lemma 1, if an algorithm G makes less than q queries to an oracle $\mathcal{O}_{\sigma_x}$ where σ_x is promised to be in $\boldsymbol{\sigma}_X$ or $\boldsymbol{\sigma}_Y$, there exists at least one element of $\boldsymbol{\sigma}_X$ and one element of $\boldsymbol{\sigma}_Y$ such that the probability of distinguishing between the corresponding oracles less than is $1/2 + \epsilon$, where

$$\frac{1}{2}\sqrt{2N^{-\delta/2} q} > \epsilon. \tag{72}$$

Equivalently, there exists at least one element of $\boldsymbol{\sigma}_X$ and one element of $\boldsymbol{\sigma}_Y$ such that in order for $\mathcal{A}$ to distinguish them with constant bias, one requires $\Omega(N^{\delta/2})$ queries. □