Quantum Queries for Testing Distributions

Sourav Chakraborty* Eldar Fischer* Arie Matsliah† Ronald de Wolf†

Abstract

We consider probability distributions given in the form of an oracle $f : [n] \to [m]$ that we can query. Here the probability $\mathcal{P}_f(j)$ of an outcome $j \in [m]$ is the fraction of the domain that is mapped to j by f. We give quantum algorithms for testing whether two such distributions are identical or at least ϵ-far in L_1-norm. Recently, Bravyi, Hassidim, and Harrow showed that if $\mathcal{P}_f$ and $\mathcal{P}_g$ are both unknown (i.e., given by oracles f and g), then this testing can be done in roughly $\sqrt{m}$ quantum queries. We consider the case where the second distribution is known, and show that testing can be done with roughly $m^{1/3}$ quantum queries, which is essentially optimal. In contrast, it is known that classical testing algorithms need about $m^{2/3}$ queries in the unknown-unknown case and about $\sqrt{m}$ queries in the known-unknown case. Our results can also be used to reduce the query complexity of graph isomorphism testers with quantum oracle access.

1 Introduction

1.1 Distribution Testing

How many samples are needed to determine whether two distributions are identical or have L_1 distance more than ϵ? This is a fundamental problem in statistical hypothesis testing and also arises in other subjects like property testing and machine learning.

We use the notation $[n] = \{1, 2, 3, \ldots, n\}$. For a function $f : [n] \to [m]$, we denote by $\mathcal{P}_f$ the distribution over $[m]$ in which the weight $\mathcal{P}_f(j)$ of every $j \in [m]$ is proportional to the number of elements $i \in [n]$ that are mapped to j. Formally, for all $j \in [m]$

$$\mathcal{P}_f(j) \triangleq \Pr_{i \sim U}[f(i) = j] = \frac{|f^{-1}(j)|}{n},$$

where U denotes the uniform distribution on $[n]$. We use this form of representation for distributions in order to allow *queries*. Namely, we assume that the function $f : [n] \to [m]$ is accessible by an oracle of the form $|x\rangle|b\rangle \mapsto |x\rangle|b \oplus f(x)\rangle$, where x is a $\log n$-bit string, b and $f(x)$ are $\log m$-bit strings and $\oplus$ is bitwise addition modulo two. Note that a classical random sample according to a distribution $\mathcal{P}_f$ can be simply obtained by picking $i \in [n]$ uniformly at random and evaluating $f(i)$. See Section 2.3 for more on the relation between sampling a distribution and querying a function.

*Computer Science Faculty, Israel Institute of Technology (Technion). Email: {sourav,eldar}@cs.technion.ac.il. Partially supported by an ERC-2007-StG grant number 202405-2 and by an ISF grant number 1101/06.

†Centrum Wiskunde & Informatica, Amsterdam. Email: {ariem,rdewolf}@cwi.nl. RdW is partially supported by a Vidi grant from the Netherlands Organization for Scientific Research (NWO), and by the European Commission under the Integrated Project Qubit Applications (QAP) funded by the IST directorate as Contract Number 015848.

We say that the distribution $\mathcal{P}_f$ is *known* (or *explicit*) if the function f is given explicitly, and hence all probabilities $\mathcal{P}_f(j)$ can be computed. $\mathcal{P}_f$ is *unknown* (or *black-box*) if we only have oracle access to the function f, and no additional information about f is given. Two distributions $\mathcal{P}_f, \mathcal{P}_g$ defined by functions $f, g : [n] \to [m]$ are *ϵ-far* if the L_1 distance between them is at least ϵ, i.e., $\|\mathcal{P}_f - \mathcal{P}_g\|_1 = \sum_{j=1}^m |\mathcal{P}_f(j) - \mathcal{P}_g(j)| \geq \epsilon$. Note that $f = g$ implies $\mathcal{P}_f = \mathcal{P}_g$ but not vice versa (for instance, permuting f leaves $\mathcal{P}_f$ invariant).

Two problems of testing distributions can be formally stated as follows:

- **Unknown-unknown case.** Given n, m, ϵ and oracle access to $f, g : [n] \to [m]$, how many queries to f and g are required in order to determine whether the unknown distributions $\mathcal{P}_f$ and $\mathcal{P}_g$ are identical or ϵ-far?

- **Known-unknown case.** Given n, m, ϵ, oracle access to $f : [n] \to [m]$ and a known distribution $\mathcal{P}_g$ (defined by an explicitly given function $g : [n] \to [m]$), how many queries to f are required to determine whether $\mathcal{P}_f$ and $\mathcal{P}_g$ are identical or ϵ-far?

If only *classical* queries are allowed (where querying the distribution means asking for a random sample), the answers to these problems are well known. For the unknown-unknown case Batu, Fortnow, Rubinfeld, Smith, and White [6] proved an upper bound of $\widetilde{O}(m^{2/3})$ on the query complexity, and Valiant [21] proved a matching (up to polylogarithmic factors) lower bound. For the known-unknown case, Goldreich and Ron [15] showed a lower bound of $\Omega(\sqrt{m})$ queries and Batu, Fischer, Fortnow, Rubinfeld, Smith, and White [5] proved a nearly tight upper bound of $\widetilde{O}(\sqrt{m})$ queries.[1]

1.2 The Quantum Case

Since the mid-1990s, a number of *quantum* algorithms have been discovered that have much better query complexity than their best classical counterparts [12, 20, 16, 3, 13, 4]. Allowing quantum queries for accessing distributions, Bravyi, Hassidim, and Harrow [8] very recently showed that the L_1 distance between two unknown distributions can actually be estimated up to small error with only $O(\sqrt{m})$ queries. Their result implies an $O(\sqrt{m})$ upper bound on the quantum query complexity for the unknown-unknown testing problem defined above. As for lower bounds, known quantum query lower bounds for the collision problem [1, 2, 18] imply that in both cases at least $\Omega(m^{1/3})$ quantum queries are required (we provide details in Section 4.1).

In this paper we consider the known-unknown case, and prove a nearly tight upper bound on its quantum query complexity.

Theorem 1.1 *Given n, m, ϵ, oracle access to $f : [n] \to [m]$ and a known distribution $\mathcal{P}_g$ (defined by an explicitly given function $g : [n] \to [m]$), the quantum query complexity of determining whether $\mathcal{P}_f$ and $\mathcal{P}_g$ are identical or ϵ-far is $\widetilde{O}(m^{1/3})$.*

[1] These classical lower bounds are stated in terms of number of samples rather than number of queries, but it is not hard to see that they hold in both models. In fact, the $\sqrt{m}$ classical query lower bound for the known-unknown case follows by the same argument as the quantum lower bound in Section 4.1.

The current status of the problem of determining the classical and the quantum query complexities for distribution testing is summarized in the following table.

	upper bound	lower bound
classical queries, unknown-unknown	$\widetilde{O}(m^{2/3})$ [6]	$\Omega(m^{2/3})$ [21]
classical queries, known-unknown	$\widetilde{O}(\sqrt{m})$ [5]	$\Omega(\sqrt{m})$ [15]
quantum queries, unknown-unknown	$O(\sqrt{m})$ [8]	$\Omega(m^{1/3})$ [1, 2, 18]
quantum queries, known-unknown	$\widetilde{O}(m^{1/3})$ this work	$\Omega(m^{1/3})$ [1, 2, 18]

The main remaining open problem is to tighten the bounds on the quantum query complexity for the unknown-unknown case. It would be very interesting if this case could also be tested using roughly $m^{1/3}$ quantum queries. In Section 4.2 we show that the easiest way to do this (just reconstructing both unknown distributions up to small error) requires $\Omega(m/\log m)$ quantum queries.

1.3 Application to Graph Isomorphism Testing

Fischer and Matsliah [14] studied the problem of testing graph isomorphism in the dense-graph model, where the graphs are represented by their adjacency matrices, and querying the graph corresponds to reading a single entry from its adjacency matrix. The goal in isomorphism testing is to determine, with high probability, whether two graphs G and H are isomorphic or ϵ-far from being isomorphic, making as few queries as possible. (The graphs are ϵ-far from being isomorphic if at least an ϵ-fraction of the entries in their adjacency matrices need to be modified in order to make them isomorphic.)

In [14] two models were considered:

- **Unknown-unknown case.** Both G and H are unknown, and they can only be accessed by querying their adjacency matrices.
- **Known-unknown case.** The graph H is known (given in advance to the tester), and the graph G is unknown (can only be accessed by querying its adjacency matrix).

As usual, in both models the query complexity is the worst-case number of queries needed to test whether the graphs are isomorphic. [14] give nearly tight bounds of $\widetilde{\Theta}(\sqrt{|V|})$ on the (classical) query complexity in the known-unknown model. For the unknown-unknown model they prove an upper bound of $\widetilde{O}(|V|^{5/4})$ and a lower bound of $\Omega(|V|)$ on the query complexity.

A natural question to ask in our context is whether allowing quantum queries can reduce the query complexity of testing graph isomorphism. A quantum query to the adjacency matrix of a graph G can be of the form $|i,j\rangle|b\rangle \mapsto |i,j\rangle|b \oplus G(i,j)\rangle$, where $G(i,j)$ is the (i,j)-th entry of the adjacency matrix of G and $\oplus$ is addition modulo two.

In [14], the bottleneck (with respect to the query complexity) of the algorithm for testing graph isomorphism in the known-unknown case is the subroutine that tests closeness between two distributions over V (also in the known-unknown case). All other parts of the algorithm make only a polylogarithmic number of queries. Therefore, our main theorem implies that with quantum oracle access, graph isomorphism in the known-unknown model can be tested with $\widetilde{O}(|V|^{1/3})$ queries.

On the other hand, a general lower bound on the query complexity of testing distributions in the known-unknown case need not imply a lower bound for testing graph isomorphism. But still, in [14] it is proved that a lower on the query complexity for deciding whether the function $f : [n] \to [n]$ is one-to-one (that is injective) or is two-to-one (that is pre-image of any $j \in [n]$ is either empty or size 2) is sufficient for showing a matching lower bound for graph isomorphism. Since our quantum lower bound for the known-unknown testing case is derived from exactly that problem (see Section 4.1), we get a matching lower bound of $\Omega(|V|^{1/3})$ on the number of quantum queries necessary for testing graph isomorphism in the known-unknown case.

Theorem 1.2 *The quantum query complexity of testing graph isomorphism in the known-unknown case is* $\widetilde{\Theta}(|V|^{1/3})$.

For the unknown-unknown case, we can use our result to prove an upper bound of $\widetilde{O}(|V|^{7/6})$ on the number of quantum queries. For this proof, we have to slightly modify the algorithm from [14] (see details in the Appendix, Section 7). As for lower bounds for the quantum query complexity, nothing better than the lower bound mentioned above for the known-unknown case is known.

Theorem 1.3 *The quantum query complexity of testing graph isomorphism in the unknown-unknown case is between* $\Omega(|V|^{1/3})$ *and* $\widetilde{\Theta}(|V|^{7/6})$.

The current status of the problem of determining the classical and the quantum query complexities for graph isomorphism testing is summarized in the following table.

	upper bound	lower bound
classical queries, unknown-unknown	$\widetilde{O}(\|V\|^{5/4})$ [14]	$\Omega(\|V\|)$ [14]
classical queries, known-unknown	$\widetilde{O}(\sqrt{\|V\|})$ [14]	$\Omega(\sqrt{\|V\|})$ [14]
quantum queries, unknown-unknown	$\widetilde{O}(\|V\|^{7/6})$ this work	$\Omega(\|V\|^{1/3})$ [1, 2, 18]
quantum queries, known-unknown	$\widetilde{O}(\|V\|^{1/3})$ this work	$\Omega(\|V\|^{1/3})$ [1, 2, 18]

The main problem remaining open on this front is to tighten the bounds on both quantum and classical query complexities for the unknown-unknown case.

2 Preliminaries

2.1 Notation

For any distribution $\mathcal{P}$ on $[m]$ we denote by $\mathcal{P}(j)$ the probability mass of $j \in [m]$ and for any $M \subseteq [m]$ we denote by $\mathcal{P}(M)$ the sum $\sum_{j\in M} \mathcal{P}(j)$. For a function $f : [n] \to [m]$, we denote by $\mathcal{P}_f$ the distribution over $[m]$ in which the weight $\mathcal{P}_f(j)$ of every $j \in [m]$ is proportional to the number of elements $i \in [n]$ that are mapped to j. Formally, for all $j \in [m]$

$$\mathcal{P}_f(j) \triangleq \Pr_{i\sim U}[f(i) = j] = \frac{|f^{-1}(j)|}{n},$$

where U is the uniform distribution on $[n]$, that is $U(i) = 1/n$ for all $i \in [n]$. Whenever the domain is clear from context (and may be something other than $[n]$), we also use U to denote the uniform distribution on that domain.

Let $\|\cdot\|_1$ and $\|\cdot\|_\infty$ stand for L_1-norm and L_∞-norm respectively. Two distributions $\mathcal{P}_f, \mathcal{P}_g$ defined by functions $f, g : [n] \to [m]$ are *ϵ-far* if the L_1 distance between them is at least ϵ. Namely, $\mathcal{P}_f$ is ϵ-far from $\mathcal{P}_g$ if

$$\|\mathcal{P}_f - \mathcal{P}_g\|_1 = \sum_{j=1}^{m} |\mathcal{P}_f(j) - \mathcal{P}_g(j)| \geq \epsilon.$$

2.2 Bucketing

Bucketing is a general tool, introduced in [6, 5], that decomposes any explicitly given distribution into a collection of distributions that are almost uniform. In this section we recall the bucketing technique and the lemmas (from [6, 5]) that we will need for our proofs.

Definition 2.1 *Given a distribution $\mathcal{P}$ over $[m]$, and $M \subseteq [m]$ such that $\mathcal{P}(M) > 0$, the* restriction $\mathcal{P}_{|M}$ *is a distribution over M with $\mathcal{P}_{|M}(i) = \mathcal{P}(i)/\mathcal{P}(M)$.*

Given a partition $\mathcal{M} = \{M_0, M_1, \ldots, M_k\}$ of $[m]$, we denote by $\mathcal{P}_{\langle\mathcal{M}\rangle}$ the distribution over $\{0\} \cup [k]$ in which $\mathcal{P}_{\langle\mathcal{M}\rangle}(i) = \mathcal{P}(M_i)$.

Given an explicit distribution $\mathcal{P}$ over $[m]$, $Bucket(\mathcal{P}, [m], \epsilon)$ is a procedure that generates a partition $\{M_0, M_1, \ldots, M_k\}$ of the domain $[m]$, where $k = \frac{2\log m}{\log(1+\epsilon)}$. This partition satisfies the following conditions:

- $M_0 = \{j \in [m] \mid \mathcal{P}(j) < \frac{1}{m \log m}\}$;
- for all $i \in [k]$, $M_i = \left\{j \in [m] \mid \frac{(1+\epsilon)^{i-1}}{m\log m} \leq \mathcal{P}(j) < \frac{(1+\epsilon)^i}{m \log m}\right\}$.

Lemma 2.2 ([5]) *Let $\mathcal{P}$ be a distribution over $[m]$ and let $\{M_0, M_1, \ldots, M_k\} \leftarrow Bucket(\mathcal{P}, [m], \epsilon)$. Then (i) $\mathcal{P}(M_0) \leq 1/\log m$; (ii) for all $i \in [k]$, $\|\mathcal{P}_{|M_i} - U_{|M_i}\|_1 \leq \epsilon$.*

Lemma 2.3 ([5]) *Let $\mathcal{P}, \mathcal{P}'$ be two distributions over $[m]$ and let $\mathcal{M} = \{M_0, M_1, \ldots, M_k\}$ be a partition of $[m]$. If $\|\mathcal{P}_{|M_i} - \mathcal{P}'_{|M_i}\|_1 \leq \epsilon_1$ for every $i \in \{0\} \cup [k]$ and if in addition $\|\mathcal{P}_{\langle\mathcal{M}\rangle} - \mathcal{P}'_{\langle\mathcal{M}\rangle}\|_1 \leq \epsilon_2$, then $\|\mathcal{P} - \mathcal{P}'\|_1 \leq \epsilon_1 + \epsilon_2$.*

Corollary 2.4 *Let $\mathcal{P}, \mathcal{P}'$ be two distributions over $[m]$ and let $\mathcal{M} = \{M_0, M_1, \ldots, M_k\}$ be a partition of $[m]$. If $\|\mathcal{P}_{|M_i} - \mathcal{P}'_{|M_i}\|_1 \leq \epsilon_1$ for every $i \in \{0\} \cup [k]$ such that $\mathcal{P}(M_i) \geq \epsilon_3/k$, and if in addition $\|\mathcal{P}_{\langle\mathcal{M}\rangle} - \mathcal{P}'_{\langle\mathcal{M}\rangle}\|_1 \leq \epsilon_2$, then $\|\mathcal{P} - \mathcal{P}'\|_1 \leq 2(\epsilon_1 + \epsilon_2 + \epsilon_3)$.*

2.3 From Sampling Problems to Oracle Problems

A standard way to access a probability distribution $\mathcal{P}$ on $[m]$ is by *sampling* it: sampling once gives the outcome $y \in [m]$ with probability $\mathcal{P}(y)$. However, in this paper we usually assume that we can access the distribution by querying a function $f : [n] \to [m]$, where the probability of y is now interpreted as the fraction of the domain that is mapped to y. Below we describe the connection between these two approaches.

Suppose we sample $\mathcal{P}$ n times, and estimate each probability $\mathcal{P}(y)$ by the fraction $\widetilde{\mathcal{P}}(y)$ of times y occurs among the n outcomes. We will analyze how good an estimator this is for $\mathcal{P}(y)$. For all $j \in [n]$, let Y_j be the indicator random variable that is 1 if the jth sample is y, and 0 otherwise. This has expectation $\mathbb{E}[Y_j] = \mathcal{P}(y)$ and variance $\mathrm{Var}[Y_j] = \mathcal{P}(y)(1 - \mathcal{P}(y))$. Our estimator is $\widetilde{\mathcal{P}}(y) = \sum_{j\in[n]} Y_j/n$. This has expectation $\mathbb{E}[\widetilde{\mathcal{P}}(y)] = \mathcal{P}(y)$ and variance $\mathrm{Var}[\widetilde{\mathcal{P}}(y)] = \mathcal{P}(y)(1-\mathcal{P}(y))/n$, since the Y_j's are independent. Now we can bound the expected error of our estimator for $\mathcal{P}(y)$ by

$$\mathbb{E}\left[|\widetilde{\mathcal{P}}(y) - \mathcal{P}(y)|\right] \leq \sqrt{\mathbb{E}\left[|\widetilde{\mathcal{P}}(y) - \mathcal{P}(y)|^2\right]} = \sqrt{\mathrm{Var}\left[\widetilde{\mathcal{P}}(y)\right]} \leq \sqrt{\mathcal{P}(y)/n}.$$

And we can bound the expected L_1-distance between the original distribution $\mathcal{P}$ and its approximation $\widetilde{\mathcal{P}}$ by

$$\mathbb{E}\left[\|\widetilde{\mathcal{P}} - \mathcal{P}\|_1\right] = \sum_{y\in[m]} \mathbb{E}\left[|\widetilde{\mathcal{P}}(y) - \mathcal{P}(y)|\right] \leq \sum_{y\in[m]} \sqrt{\mathcal{P}(y)/n} \leq \sqrt{m/n},$$

where the last inequality used Cauchy-Schwarz and the fact that $\sum_y \mathcal{P}(y) = 1$. For instance, if $n = 10000m$ then $\mathbb{E}[\|\widetilde{\mathcal{P}} - \mathcal{P}\|_1] \leq 1/100$, and hence (by Markov's Inequality) $\|\widetilde{\mathcal{P}} - \mathcal{P}\|_1 \leq 1/10$ with probability at least 9/10. If we now define a function $f : [n] \to [m]$ by setting $f(j)$ to the jth value in the sample, we have obtained a representation which is a good approximation of the original distribution. Note that if $n = o(m)$ then we cannot hope to be able to approximately represent all possible m-element distributions by some $f : [n] \to [m]$, since all probabilities will be integer multiples of $1/n$. For instance if $\mathcal{P}$ is uniform and $n = o(m)$, then the total L_1 distance between $\mathcal{P}$ and a $\widetilde{\mathcal{P}}$ induced by any $f : [n] \to [m]$ is near-maximal. Accordingly, the typical case we are interested in is $n = \Theta(m)$.

2.4 Quantum Queries and Approximate Counting

We only use one specific quantum procedure as a black-box in otherwise classical algorithms. Hence we will not explain the model of quantum query algorithms in much detail (see [19, 10] for that). Suffice it to say that the function f is assumed to be accessible by the oracle unitary transformation O_f, which acts on a $(\log n + \log m)$-qubit space by sending the basis vector $|x\rangle|b\rangle$ to $|x\rangle|b \oplus f(x)\rangle$ where $\oplus$ is bitwise addition modulo two.

For any set $S \subseteq [m]$, let U_f^S denote the unitary transformation which maps $|x\rangle|b\rangle$ to $|x\rangle|b \oplus 1\rangle$ if $f(x) \in S$, and to $|x\rangle|b \oplus 0\rangle$ otherwise. This unitary transformation can be easily implemented using $\log m$ ancilla bits and two queries to O_f.[2] If $f_S : [n] \to \{0, 1\}$ is defined as $f_S(x) = 1$ if and only if $f(x) \in S$, then the unitary transformation U_f^S acts as an oracle to the function f_S. Brassard, Høyer, Mosca, and Tapp [7, Theorem 13] gave an algorithm to approximately count the size of certain sets.

Theorem 2.5 (BHMT) *For every positive integer q and $\ell > 1$, and given quantum oracle access to a Boolean function $h : [n] \to \{0, 1\}$, there is an algorithm that makes q queries to h and outputs an estimate t' to $t = |h^{-1}(1)|$ such that*

$$|t' - t| \leq 2\pi\ell\frac{\sqrt{t(n-t)}}{q} + \pi^2\ell^2\frac{n}{q^2}$$

with probability at least $1 - 1/2(\ell - 1)$.

Corollary 2.6 *For every $\delta \geq 0$, $r \geq 1$, $0 < \epsilon < 1/2$, and given oracle O_f for the function $f : [n] \to [m]$, and for every set $S \subseteq [m]$, there is a quantum algorithm $\mathrm{QEstimate}(\mathcal{P}_f, S, \delta, r, \epsilon)$ that makes $(8\pi/\sqrt{\delta^2 r})(2 + 1/\epsilon)$ queries to f (we ignore rounding to integers for simplicity) and with probability at least $1 - \epsilon$ outputs an estimate p' to $p = |f^{-1}(S)|/n$ such that*

[2] We need *two* queries to f instead of one, because the quantum algorithm has to "uncompute" the first query in order to clean up its workspace.

1. If $p < (1 + \frac{3}{2}\delta)r$ then $|p' - r| < 2\delta r$

2. If $p > (1 + 3\delta)r$ then $|p' - r| > 2\delta r$

Proof. Basically the algorithm is required to estimate $|f_S^{-1}(1)|$. Using two queries to the oracle O_f we can construct U_f^S that acts like an oracle to the function f_S. Now estimate $f_S^{-1}(1)$ using the algorithm in the Theorem 2.5 with parameters $q = (4\pi/\sqrt{\delta^2 r})(2 + 1/\epsilon)$ and $\ell = 1 + 1/2\epsilon$. With probability at least $1 - \epsilon$, the estimate p' to $|f_S^{-1}(1)|$ satisfies

$$|p' - p| < \frac{\sqrt{p(1-p)\delta^2 r}}{4} + \frac{\delta^2 r}{64}$$

Now if $p < (1 + \frac{3}{2}\delta)r$ then $|p' - r| < \frac{3}{2}\delta r + |p' - p| < 2\delta r$. And if $|p - r| = \gamma r > 3\delta r > 3\delta^2 r$ then $|p' - r| > \gamma r - |p' - p| > \gamma r - \left(\frac{\sqrt{\delta^2(1+\gamma)r^2}}{4} + \frac{\delta^2 r}{64}\right) > \frac{2\gamma}{3} r > 2\delta r$. ■

In other words, the algorithm QEstimate$(\mathcal{P}_f, S, \delta, r, \epsilon)$ uses the oracle O_f for the function $f : [n] \rightarrow [m]$, to determine with probability $(1-\epsilon)$ whether $p = |f^{-1}(S)|/N$ is close to a given value r. The algorithm outputs an estimate p' to p which is close to r if p is close to r (item 1) and which is far from r if p is far from r (item 2). The error parameter ϵ, the closeness parameter δ and the fixed value r against which p is compared are given as inputs to the algorithm. The number of queries made by the algorithm is dependent on ϵ, r and δ.

3 Proof of Theorem 1.1

We prove Theorem 1.1 in two parts. First, in Section 3.1 we prove that with $\widetilde{O}(m^{1/3})$ quantum queries it is possible to test whether a black-box distribution $\mathcal{P}_f$ (defined by some $f : [n] \rightarrow [m]$) is close to uniform. We actually prove that this can be even done *tolerantly* in a sense, meaning that a distribution that is close to uniform in both L_1 and L_∞ norms is accepted with high probability. Tolerance is essential for the application of the uniformity tester in the second part. There we use the bucketing technique of [6, 5] to reduce the task of testing closeness to a known distribution to testing uniformity. In order to incur only a polylogarithmic overhead in the query complexity, this reduction relies on the fact that the uniformity tester is tolerant. (The second part of the proof appears in the Appendix, Section 6.)

Combining the two parts, we deduce that the quantum query complexity for testing whether a black-box distribution is close to an explicitly given one is $\widetilde{O}(m^{1/3})$.

3.1 Testing Uniformity

Given $\epsilon > 0$ and oracle access to a function $f : [n] \rightarrow [m]$, our task is to distinguish the case $\|\mathcal{P}_f - U\|_1 \geq \epsilon$ from the case $\|\mathcal{P}_f - U\|_\infty \leq \epsilon/4m$. Notice that this is a stronger condition than the one required for the usual testing task, where the goal is to distinguish the case $\|\mathcal{P}_f - U\|_1 \geq \epsilon$ from the case $\|\mathcal{P}_f - U\|_1 = 0$. For simplicity we will assume here that $n = \widetilde{\Theta}(m)$, so we can state our upper bounds on the query complexity in terms of n.[3]

[3] If $n \ll m$ then $\mathcal{P}_f$ will always be far from the uniform distribution on $[m]$; and if $n \gg m$ then we can choose a random subset $S \subseteq [n]$ of size $O(m \log^2 m)$, and restrict to $f' : S \rightarrow [m]$. A simple probabilistic argument shows that, with high probability, we will have $|\mathcal{P}_f(j) - \mathcal{P}_{f'}(j)| = o(1/m)$ for all $j \in [m]$ simultaneously.

Theorem 3.1 *There is a quantum testing algorithm (Algorithm 1, below) that for any fixed $\epsilon > 0$, given oracle access to a function $f : [n] \to [m]$ makes $\widetilde{O}(n^{1/3})$ quantum queries and with probability at least $1 - 1/\log^2 n$ outputs REJECT if $\|\mathcal{P}_f - U\|_1 \geq \epsilon$, and ACCEPT if $\|\mathcal{P}_f - U\|_\infty \leq \epsilon/4m$.*

Algorithm 1 (Tests closeness to the uniform distribution.)

---

pick a set $T \subseteq [n]$ of $t = 4n^{1/3}\log n$ indices uniformly at random
query f on all indices in T
if $f(i) = f(j)$ for some $i, j \in T, i \neq j$ **then**
  REJECT
end if
$p' \leftarrow \text{QEstimate}(\mathcal{P}_f, f(T), \epsilon^2/16, t/m, 1/\log^3 n)$
if p' is in the interval $(1 \pm \frac{\epsilon^2}{8})\frac{t}{m}$ **then**
  ACCEPT
else
  REJECT
end if

Proof. Fix $\epsilon > 0$ and let $f : [n] \to [m]$ be any function. It is easy to see that Algorithm 1 makes $\widetilde{O}(n^{1/3})$ queries. It makes $t = \widetilde{O}(n^{1/3})$ classical queries initially, and the call to QEstimate requires $\widetilde{O}(\sqrt{2m/\epsilon^4 t}) = \widetilde{O}(n^{1/3})$ additional quantum queries.

Now we show that Algorithm 1 satisfies the correctness conditions in Theorem 3.1. Let $V \subseteq [m]$ denote the multi-set of values $\{f(x) \mid x \in T\}$. If $\|\mathcal{P}_f - U\|_\infty \leq \epsilon/4m$ then $\mathcal{P}_f(V) \leq (1 + \frac{\epsilon}{4})t/m$, and hence

$$p(t;m) \triangleq \Pr[\text{the elements in } V \text{ are distinct}] \geq \left(1 - \frac{(1+\frac{\epsilon}{4})t}{m}\right)^t \geq 1 - \frac{(1+\frac{\epsilon}{4})t^2}{m} > 1 - \frac{1}{\log^3 n}.$$

Thus if $\|\mathcal{P}_f - U\|_\infty \leq \epsilon/4m$ then with probability at least $1 - 1/\log^3 n$, the tester does not discover any collision. If, on the other hand, $\|\mathcal{P}_f - U\|_1 \geq \epsilon$ and a collision is discovered, then the tester outputs REJECT, as expected. Hence the following lemma suffices for completing the proof of Theorem 3.1.

Lemma 3.2 *Conditioned on the event that all elements in V are distinct, we have*

- *if* $\|\mathcal{P}_f - U\|_\infty \leq \epsilon/4m$ *then* $\Pr\left[|\mathcal{P}_f(V) - t/m| \leq \frac{3\epsilon^2 t}{32m}\right] \geq 1 - 1/\log^3 n$*;*
- *if* $\|\mathcal{P}_f - U\|_1 \geq \epsilon$ *then* $\Pr\left[|\mathcal{P}_f(V) - t/m| > \frac{3\epsilon^2 t}{16m}\right] \geq 1 - 1/\log^3 n$*.*

Assuming Lemma 3.2, let us first finish the proof of Theorem 3.1. If $\|\mathcal{P}_f - U\|_\infty \leq \epsilon/4m$ then with probability at least $1 - 2/\log^3 n$ the elements in V are distinct and also $|\mathcal{P}_f(V) - t/m| \leq \frac{3\epsilon^2 t}{32m}$. In this case, by Corollary 2.6 the probability that the estimate of QEstimate is outside the interval $(1 \pm \frac{\epsilon^2}{8})\frac{t}{m}$ is less than $1/\log^3 n$. Hence the overall probability that Algorithm 1 outputs REJECT is bounded by $3/\log^3 n < 1/\log^2 n$.

If $\|\mathcal{P}_f - U\|_1 \geq \epsilon$, then either Algorithm 1 discovers a collision and outputs REJECT, or otherwise, by Lemma 3.2, $\Pr\left[|\mathcal{P}_f(V) - t/m| > \frac{3\epsilon^2 t}{16m}\right] \geq 1 - 1/\log^3 n$. In the latter case, by Corollary 2.6 the probability that the estimate of QEstimate is inside the interval $(1 \pm \frac{\epsilon^2}{8})\frac{t}{m}$ is less than $1/\log^3 n$. Hence the overall probability that Algorithm 1 outputs ACCEPT is bounded by $1/\log^2 n$. ■

Proof. [of Lemma 3.2] Let $W_f(V) = \sum_{y \in V} \mathcal{P}_f(y)$. Assuming that all elements in V are distinct, $\mathcal{P}_f(V) = W_f(V)$. For the first item of the lemma, it suffices to prove that if $\|\mathcal{P}_f - U\|_\infty \le \epsilon/4m$ then

$$\Pr\left[|W_f(V) - \frac{t}{m}| > \frac{3\epsilon^2 t}{32m}\right] \le 1/\log^3 n$$

and for the second item of the lemma, it suffices to prove that if $\|\mathcal{P}_f - U\|_1 \ge \epsilon$ then

$$\Pr\left[W_f(V) > (1 + \frac{3\epsilon^2}{16})\frac{t}{m}\right] \ge 1 - 1/\log^3 n.$$

It is not clear how to prove the last inequality directly, because the probabilities of certain elements under $\mathcal{P}_f$ can be very high. To address this issue we define $\widetilde{\mathcal{P}}_f(y) \triangleq \min\{3/m, \mathcal{P}_f(y)\}$ and $\widetilde{W}_f(V) \triangleq \sum_{y \in V} \widetilde{\mathcal{P}}_f(y)$. Clearly $\widetilde{W}_f(V) \le W_f(V)$ for any V. Also, if $\|\mathcal{P}_f - U\|_\infty \le \epsilon/4m$ then $\widetilde{W}_f(V) = W_f(V)$.

Claim 3.3 *The following three statements hold:*

1. *if $\|\mathcal{P}_f - U\|_\infty \le \epsilon/4m$, then $\frac{t}{m} \le \mathbb{E}[\widetilde{W}_f(V)] < \left(1 + \frac{\epsilon^2}{16}\right)\frac{t}{m}$*
2. *if $\|\mathcal{P}_f - U\|_1 \ge \epsilon$, then $\mathbb{E}[\widetilde{W}_f(V)] > \left(1 + \frac{\epsilon^2}{4}\right)\frac{t}{m}$;*
3. $\Pr\left[\left|\widetilde{W}_f(V) - \mathbb{E}[\widetilde{W}_f(V)]\right| > \frac{\epsilon^2 t}{32m}\right] < 1/\log^3 n.$

Assuming the above claim (its full proof appears in the Appendix, Section 8) we have

- if $\|\mathcal{P}_f - U\|_\infty \le \epsilon/4m$ then

$$\Pr\left[|W_f(V) - \frac{t}{m}| > \frac{3\epsilon^2 t}{32m}\right] \le \Pr\left[\left|W_f(V) - \mathbb{E}[W_f(V)]\right| > \frac{\epsilon^2 t}{32m}\right] < 1/\log^3 n;$$

- if $\|\mathcal{P}_f - U\|_1 \ge \epsilon$ then

$$\Pr\left[W_f(V) < (1 + \frac{3\epsilon^2}{16})\frac{t}{m}\right] \le \Pr\left[\widetilde{W}_f(V) < (1 + \frac{3\epsilon^2}{16})\frac{t}{m}\right]$$
$$\le \Pr\left[\left|\widetilde{W}_f(V) - \mathbb{E}[\widetilde{W}_f(V)]\right| > \frac{\epsilon^2 t}{16m}\right] \le \Pr\left[\left|\widetilde{W}_f(V) - \mathbb{E}[\widetilde{W}_f(V)]\right| > \frac{\epsilon^2 t}{32m}\right] < 1/\log^3 n.$$

Hence the lemma follows. ■

4 Quantum Lower Bounds

In this section we prove two lower bounds on quantum query complexity.

4.1 Lower Bound for the Known-Unknown Case

Here we show that our quantum testing algorithm for the known-unknown case is close to optimal: even for testing an unknown distribution (given as $f : [n] \to [m]$) against the uniform one, we need $\Omega(m^{1/3})$ quantum queries. As Bravyi, Hassidim, and Harrow [8] also observed, such a lower bound can be derived from known lower bounds for the collision problem. However, one has to be careful to use the version of the lower bound that applies to functions $f : [m] \to [m]$, due to Ambainis [2] and Kutin [18], rather than the earlier lower bound of Aaronson and Shi [1] that had to assume a larger range-size.

Theorem 4.1 *Let $0 < \epsilon < 1/2$ be a fixed constant. Let A be a quantum algorithm that tests whether an unknown distribution is equal to uniform or at least ϵ-far from it, meaning that for every $f : [n] \to [m]$, with success probability at least $2/3$, it decides whether $\mathcal{P}_f = U$ or $\|\mathcal{P} - \mathcal{P}_f\|_1 \geq \epsilon$ (under the promise that one of these two cases holds). Then A makes $\Omega(m^{1/3})$ queries to f.*

Proof. Consider the following probability distribution on f: with probability 1/2, f is a random 1-1 function (equivalently, a random permutation on $[m]$), and with probability 1/2, f is a random 2-to-1 function. Note that in the first case we have $\mathcal{P}_f = U$, while in the second case $\mathcal{P}_f(j) \in \{0, 2/m\}$ for all $j \in [m]$ and hence $\|\mathcal{P}_f - U\|_1 = 1$. Thus a quantum testing algorithm like A can decide between these two cases with high success probability. But Ambainis [2] and Kutin [18] showed that this requires $\Omega(m^{1/3})$ queries. ■

4.2 Lower Bound for Reconstructing the Distribution

Previously we studied the problem of *deciding* whether an unknown distribution, given by $f : [n] \to [m]$, is close to or far from another distribution (which itself may be known or unknown). Of course, the easiest way to solve such a decision problem would be to *reconstruct* the unknown distribution, up to some small L_1-error. Efficiently solving the reconstruction problem, say in $m^{1/2}$ or even $m^{1/3}$ queries, would immediately allow us to solve the decision problem. However, below we prove that even quantum algorithms cannot solve the reconstruction problem efficiently; the proof is in the Appendix, Section 9:

Theorem 4.2 *Let $0 < \epsilon < 1/2$ be a fixed constant. Let A be a quantum algorithm that solves the reconstruction problem, meaning that for every $f : [n] \to [m]$, with probability at least $2/3$, it outputs a probability distribution $\mathcal{P} \in [0,1]^m$ such that $\|\mathcal{P} - \mathcal{P}_f\|_1 \leq \epsilon$. Then A makes $\Omega(m/\log m)$ queries to f.*

5 Conclusion

In this paper we studied the quantum query complexity of testing whether two probability distributions on a set $[m]$ are equal or ϵ-far. Our main result is a quantum tester for the case where one of the two distributions is known (i.e., given explicitly) while the other is unknown and represented by a function that can be queried. Our tester uses roughly $m^{1/3}$ queries to the function, which is close to optimal. It would be very interesting to extend this quantum upper bound to the case where both distributions are unknown. Such a quantum tester would show that the known-unknown and unknown-unknown cases have the same complexity in the quantum world. In contrast, they are known to have different complexities in the classical world: about $m^{1/2}$ queries for the known-unknown case and about $m^{2/3}$ queries for the unknown-unknown case.

Acknowledgments

We would like to thank Avinatan Hassidim, Harry Buhrman and Prahladh Harsha for useful discussions. We would also like to thank Avinatan Hassidim for informing us about the results in [8] and providing a copy of the writeup.

References

[1] S. Aaronson and Y. Shi. Quantum lower bounds for the collision and the element distinctness problems. *Journal of the ACM*, 51(4):595–605, 2004.

[2] A. Ambainis. Polynomial degree and lower bounds in quantum complexity: Collision and element distinctness with small range. *Theory of Computing*, 1(1):37–46, 2005. quant-ph/0305179.

[3] A. Ambainis. Quantum walk algorithm for element distinctness. *SIAM Journal on Computing*, 37(1):210–239, 2007. Earlier version in FOCS'04. quant-ph/0311001.

[4] A. Ambainis, A. Childs, B. Reichardt, R. Špalek, and S. Zhang. Any AND-OR formula of size n can be evaluated in time $N^{1/2+o(1)}$ on a quantum computer. In *Proceedings of 48th IEEE FOCS*, 2007.

[5] T. Batu, E. Fischer, L. Fortnow, R. Kumar, R. Rubinfeld, and P. White. Testing random variables for independence and identity. In *Proceedings of 42nd IEEE FOCS*, pages 442–451, 2001.

[6] T. Batu, L. Fortnow, R. Rubinfeld, W. D. Smith, and P. White. Testing that distributions are close. In *Proceedings of 41st IEEE FOCS*, pages 259–269, 2000.

[7] G. Brassard, P. Høyer, M. Mosca, and A. Tapp. Quantum amplitude amplification and estimation. In *Quantum Computation and Quantum Information: A Millennium Volume*, volume 305 of *AMS Contemporary Mathematics Series*, pages 53–74. 2002. quant-ph/0005055.

[8] S. Bravyi, A. Hassidim, and A. Harrow. Quantum algorithms for testing properties of distributions, 2009. In preparation.

[9] H. Buhrman, R. Cleve, and A. Wigderson. Quantum vs. classical communication and computation. In *Proceedings of 30th ACM STOC*, pages 63–68, 1998. quant-ph/9802040.

[10] H. Buhrman and R. d. Wolf. Complexity measures and decision tree complexity: A survey. *Theoretical Computer Science*, 288(1):21–43, 2002.

[11] R. Cleve, W. v. Dam, M. Nielsen, and A. Tapp. Quantum entanglement and the communication complexity of the inner product function. In *Proceedings of 1st NASA QCQC conference*, volume 1509 of *Lecture Notes in Computer Science*, pages 61–74. Springer, 1998. quant-ph/9708019.

[12] D. Deutsch and R. Jozsa. Rapid solution of problems by quantum computation. In *Proceedings of the Royal Society of London*, volume A439, pages 553–558, 1992.

[13] E. Farhi, J. Goldstone, and S. Gutmann. A quantum algorithm for the Hamiltonian NAND tree. *Theory of Computing*, 4(1):169–190, 2008. quant-ph/0702144.

[14] E. Fischer and A. Matsliah. Testing graph isomorphism. *SIAM Journal on Computing*, 38(1):207–225, 2008.

[15] O. Goldreich and D. Ron. On testing expansion in bounded-degree graphs. *Electronic Colloquium on Computational Complexity (ECCC)*, 7(20), 2000.

[16] L. K. Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of 28th ACM STOC*, pages 212–219, 1996. quant-ph/9605043.

[17] A. S. Holevo. Bounds for the quantity of information transmitted by a quantum communication channel. *Problemy Peredachi Informatsii*, 9(3):3–11, 1973. English translation in *Problems of Information Transmission*, 9:177–183, 1973.

[18] S. Kutin. Quantum lower bound for the collision problem with small range. *Theory of Computing*, 1(1):29–36, 2005. quant-ph/0304162.

[19] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

[20] D. Simon. On the power of quantum computation. *SIAM Journal on Computing*, 26(5):1474–1483, 1997. Earlier version in FOCS'94.

[21] P. Valiant. Testing symmetric properties of distributions. In *Proceedings of 40th ACM STOC*, pages 383–392, 2008.

Appendix

6 Using the Uniformity Test for Testing Closeness to a Known Distribution

In this section we prove Theorem 1.1 based on Theorem 3.1. Let $\mathcal{P}_f$ be an unknown distribution and let $\mathcal{P}_g$ be a known distribution, defined by $f, g : [n] \to [m]$ respectively. We claim that for any fixed $\epsilon > 0$, Algorithm 2 makes $\widetilde{O}(n^{1/3})$ queries and distinguishes the case $\|\mathcal{P}_f - \mathcal{P}_g\|_1 = 0$ from the case $\|\mathcal{P}_f - \mathcal{P}_g\|_1 > 5\epsilon$ with probability at least $2/3$, satisfying the requirements of Theorem 1.1.[4]

Algorithm 2 (Tests closeness to a known distribution.)

1: let $\mathcal{M} \triangleq \{M_0, \ldots, M_k\} \leftarrow Bucket(\mathcal{P}_g, [m], \frac{\epsilon}{4})$ for $k = \frac{2\log m}{\log(1+\epsilon/4)}$
2: **for** $i = 1$ to k **do**
3:    **if** $\mathcal{P}_g(M_i) \geq \epsilon/k$ **then**
4:       **if** $\|(\mathcal{P}_f)_{|M_i} - U_{|M_i}\|_1 \geq \epsilon$ (check with Algorithm 1) **then**
5:          REJECT
6:       **end if**
7:    **end if**
8: **end for**
9: **if** $\|(\mathcal{P}_f)_{\langle\mathcal{M}\rangle} - (\mathcal{P}_g)_{\langle\mathcal{M}\rangle}\|_1 > \epsilon/4$ (check with Algorithm 1) **then**
10:    REJECT
11: **end if**
12: ACCEPT

Observe that no queries are made by Algorithm 2 itself, and the total number of queries made by calls to Algorithm 1 is bounded by $k \cdot \widetilde{O}(m^{1/3}) + \widetilde{O}(k^{1/3}) = \widetilde{O}(m^{1/3})$. In addition, since the failure probability of Algorithm 1 is at most $1/\log^2 m \ll 1/k$, we can assume that with high probability none of its executions failed.

For any $i \in [k]$ and any $x \in M_i$, by the definition of the buckets $\frac{(1+\epsilon/4)^{i-1}}{m\log m} \leq \mathcal{P}_g(x) \leq \frac{(1+\epsilon/4)^{i}}{m\log m}$. Thus, for any $i \in [k]$ and $x \in M_i$, $(1-\frac{\epsilon}{4})/|M_i| < 1/(1+\frac{\epsilon}{4})|M_i| < (\mathcal{P}_g)_{|M_i}(x) < (1+\frac{\epsilon}{4})/|M_i|$, or equivalently for any $i \in [k]$ we have $\|(\mathcal{P}_g)_{|M_i} - U_{|M_i}\|_\infty \leq \frac{\epsilon}{4|M_i|}$. This means that if $\|\mathcal{P}_f - \mathcal{P}_g\|_1 = 0$ then

1. for any $i \in [k]$, $\|(\mathcal{P}_f)_{|M_i} - U_{|M_i}\|_\infty \leq \frac{\epsilon}{4|M_i|}$ and thus the tester never outputs REJECT in Line 5 (recall that we assume that Algorithm 1 did not err).

2. $\|(\mathcal{P}_f)_{\langle\mathcal{M}\rangle} - (\mathcal{P}_g)_{\langle\mathcal{M}\rangle}\|_1 = 0$, and hence the tester does not output REJECT in Line 10 either.

On the other hand, if $\|\mathcal{P}_f - \mathcal{P}_g\|_1 > 5\epsilon$ then by Lemma 2.4 we know that either $|(\mathcal{P}_f)_{\langle\mathcal{M}\rangle} - (\mathcal{P}_g)_{\langle\mathcal{M}\rangle}| > \epsilon/4$ or there is at least one $i \in [k]$ for which $\mathcal{P}_f(M_i) \geq \epsilon/k$ and $\|(\mathcal{P}_f)_{|M_i} - (\mathcal{P}_g)_{|M_i}\|_1 > 5\epsilon/4$ (otherwise $\|\mathcal{P}_f - \mathcal{P}_g\|_1$ must be smaller than $2(5\epsilon/4 + \epsilon/4 + \epsilon) = 5\epsilon$). In the first case the tester will reject in Line 10. In the second case the tester will reject in Line 5 as $\|(\mathcal{P}_f)_{|M_i} - (\mathcal{P}_g)_{|M_i}\|_1 > 5\epsilon/4$ implies (by the triangle inequality) $\|(\mathcal{P}_f)_{|M_i} - U_{|M_i}\|_1 > \epsilon$, since $\|(\mathcal{P}_g)_{|M_i} - U_{|M_i}\|_1 < \epsilon/4$ by Lemma 2.2.

[4]We use 5ϵ instead ϵ for better readability in the sequel.

7 Proof-sketch of Theorem 1.3

First we outline the ideas in the algorithm of [14] for testing isomorphism between two unknown graphs G and H, and then we describe the changes that are required in order to reduce its query complexity with a quantum oracle access.

Let G be a graph and $C_G \subseteq V(G)$. A C_G*-label* of a vertex $v \in V(G)$ is a binary vector of length $|C_G|$ that represents the neighbors of v in C_G. The distribution $\mathcal{P}_{C_G}$ over $\{0,1\}^{|C_G|}$ is defined according to the graph G, where for every $x \in \{0,1\}^{|C_G|}$ the probability $\mathcal{P}_{C_G}(x)$ is proportional to the number of vertices in G with C_G-label equal to x. Notice that the support of $\mathcal{P}_{C_G}$ is bounded by $|V(G)|$.

The algorithm of [14] is based on two main observations:

1. if there is an isomorphism σ between G and H, then for every $C_G \subseteq V(G)$ and the corresponding $C_H \triangleq \sigma(C_G)$, the distributions $\mathcal{P}_{C_G}$ and $\mathcal{P}_{C_H}$ are equivalent.

2. if G and H are far from being isomorphic, then for every equivalently sized (and not too small) $C_G \subseteq V(G)$ and $C_H \subseteq V(H)$, either the distributions $\mathcal{P}_{C_G}$ and $\mathcal{P}_{C_H}$ are far, or otherwise it is possible to "realize" with only a poly-logarithmic number of queries that there exists no isomorphism that maps C_G to C_H.

Once these observations are made, the high level idea in the algorithm of [14] is to go over a sequence of pairs of sets C_G, C_H (such that with high probability at least one of them satisfies $C_H \triangleq \sigma(C_G)$ if indeed an isomorphism σ exists), and to test closeness between the corresponding distributions $\mathcal{P}_{C_G}$ and $\mathcal{P}_{C_H}$.

This sequence of pairs is defined as follows: first we pick (at random) a set U_G of $|V|^{1/4}\log^3|V|$ vertices from G and a set U_H of $|V|^{3/4}\log^3|V|$ vertices from H. Then we make all $|V|^{5/4}\log^3|V|$ possible queries in $U_G \times V(G)$. After this, for any $C_G \subseteq U_G$ the distribution $\mathcal{P}_{C_G}$ is known exactly. Indeed, the sequence of sets C_G, C_H will consist of all pairs $C_G \subseteq U_G, C_H \subseteq U_H$, where both C_G and C_H are of size $\log^2|V|$. It is not hard to prove that if G and H have an isomorphism σ, then with probability $1 - o(1)$ the size of $U_G \cap \sigma(U_H)$ will exceed $\log^2|V|$, and hence one of the pairs will satisfy $C_H \triangleq \sigma(C_G)$.

Now, for each pair C_G, C_H we test if the distributions $\mathcal{P}_{C_G}$ and $\mathcal{P}_{C_H}$ are identical. Since we know the distributions $\mathcal{P}_{C_G}$ (for every $C_G \subseteq U_G$), we only need to sample the distributions $\mathcal{P}_{C_H}$. Sampling the distributions $\mathcal{P}_{C_H}$ is done by taking a set $S \subseteq V(H)$ of size $\widetilde{O}(\sqrt{|V|})$ and re-using it for all these tests. In total, the algorithm in [14] makes roughly $|U_G \times V(G)| + |U_H \times S| = \widetilde{O}(|V|^{5/4})$ queries.

To get the desired improvement, we follow the same path, but use our quantum distribution tester instead of the classical one. This allows us to reduce the size of the set S to $\widetilde{O}(|V|^{1/3})$. Consequently, in order to balance the amount of queries we make in both graphs, we will resize the sets U_G and U_H to $\widetilde{O}(|V|^{1/6})$ and $\widetilde{O}(|V|^{5/6})$ respectively, which still satisfies the "large-intersection" property and brings the total number of queries down to $|U_G \times V(G)| + |U_H \times S| = \widetilde{O}(|V|^{7/6})$.

8 Proof of Claim 3.3

We start by computing the expected value of $\widetilde{W}_f(V)$.

$$\mathbb{E}[\widetilde{W}_f(V)] = \sum_{y\in V}\sum_{z\in[m]} \mathcal{P}_f(z)\widetilde{\mathcal{P}}_f(z) = t\left(\sum_{z:\mathcal{P}_f(z)<3/m} \mathcal{P}_f(z)^2 + \sum_{z:\mathcal{P}_f(z)\geq 3/m} 3\mathcal{P}_f(z)/m\right)$$

$$= t\left(\sum_{z\in[m]} \mathcal{P}_f(z)^2 - \sum_{z:\mathcal{P}_f(z)\geq 3/m} \mathcal{P}_f(z)(\mathcal{P}_f(z)-3/m)\right).$$

Let $\delta(z) \triangleq \mathcal{P}_f(z) - 1/m$ and let $r \triangleq |\{z \mid \delta(z) < 2/m\}|$. Then

$$\mathbb{E}[\widetilde{W}_f(V)] = t\left(\sum_{z\in[m]} (1/m+\delta(z))^2 - \sum_{z:\delta(z)\geq 2/m} (1/m+\delta(z))(\delta(z)-2/m)\right)$$

and since $\sum_{z\in[m]} \delta(z) = 0$ we have

$$= t\left(1/m + \sum_{z:\delta(z)<2/m} \delta(z)^2 + 2(m-r)/m^2 + \sum_{z:\delta(z)\geq 2/m} \delta(z)/m\right)$$

For the first item of the claim, since $\delta(z) \leq \epsilon/4m$ we have $r = m$, and hence the equality $W_f(V) = \widetilde{W}_f(V)$ always holds as there are no z for which $\delta(z) \geq 2/m$. Therefore, from the above equation we have

$$\mathbb{E}[W_f(V)] = t\left(1/m + \sum_{z:\delta(z)<2/m} \delta(z)^2\right) \geq \frac{t}{m}$$

and

$$\mathbb{E}[W_f(V)] = t\left(1/m + \sum_{z:\delta(z)<2/m} \delta(z)^2\right) < t\left(1/m + \sum_{z:\delta(z)<2/m} (\epsilon/4m)^2\right) \leq \left(1+\frac{\epsilon^2}{16}\right)\frac{t}{m},$$

as required in the first item of the claim.

Now we move to the second item of the claim, where $\|\mathcal{P}_f - U\|_1 \geq \epsilon$. By Cauchy-Schwarz we have

$$\sum_{z:\delta(z)<2/m} \delta(z)^2 = \sum_{z:\delta(z)<2/m} |\delta(z)|^2 \geq \frac{1}{r}\Big(\sum_{z:\delta(z)<2/m} |\delta(z)|\Big)^2,$$

hence

$$\mathbb{E}[\widetilde{W}_f(V)] \geq t\left(1/m + \frac{1}{r}\Big(\sum_{z:\delta(z)<2/m} |\delta(z)|\Big)^2 + \frac{1}{m}\sum_{z:\delta(z)\geq 2/m} \delta(z)\right)$$

$$\geq \frac{t}{m}\left(1 + \Big(\sum_{z:\delta(z)<2/m} |\delta(z)|\Big)^2 + \sum_{z:\delta(z)\geq 2/m} \delta(z)\right).$$

Since $\sum_{z\in[m]} |\delta(z)| = \|\mathcal{P}_f - U\|_1 \geq \epsilon$, at least one of

$$\sum_{z:\delta(z)<2/m} |\delta(z)| > \epsilon/2$$

or

$$\sum_{z:\delta(z)\geq 2/m} |\delta(z)| = \sum_{z:\delta(z)\geq 2/m} \delta(z) \geq \epsilon/2$$

must hold. In both cases we have $\mathbb{E}[\widetilde{W}_f(V)] > \frac{t}{m}(1 + \frac{\epsilon^2}{4})$, as required.

Finally, we prove the third statement of the claim. By Hoeffding's Inequality we have

$$\Pr\left[\mathbb{E}[\widetilde{W}_f(V)] - \widetilde{W}_f(V) > \frac{\epsilon^2 t}{32m}\right] \leq \exp\left(-\frac{2\epsilon^4 t^2}{1024 m^2 \sum_{y \in V}(b_y - a_y)^2}\right),$$

where b_y and a_y are upper and lower bounds on $\widetilde{\mathcal{P}}(y)$. Since $b_y \leq 3/m$ and $a_y \geq 0$ for all $y \in [m]$, we get

$$\Pr\left[\mathbb{E}[\widetilde{W}_f(V)] - \widetilde{W}_f(V) > \frac{\epsilon^2 t}{32m}\right] \leq \exp(-\Omega(\epsilon^4 t)) < \frac{1}{\log^3 n}.$$

9 Proof of Theorem 4.2

The proof uses some basic quantum information theory, and is most easily stated in a communication setting. Suppose Alice has a uniformly distributed m-bit string x of weight $m/2$. This is a random variable with entropy $\log\binom{m}{m/2} = m - O(\log m)$ bits. Let q be the number of queries A makes. We will show below that Alice can give Bob $\Omega(m)$ bits of information (about x), by a process that (interactively) communicates $O(q \log m)$ qubits. By Holevo's Theorem [17] (see also [11, Theorem 2]), establishing k bits of mutual information requires communicating at least k qubits, hence $q = \Omega(m/\log m)$.

Given an $x \in \{0,1\}^m$ of weight $n = m/2$, let $f : [n] \to [m]$ be an injective function to $\{j \mid x_j = 1\}$, and let $\mathcal{P}_f$ be the corresponding probability distribution over m elements (which is $\mathcal{P}_f(j) = 2/m$ where $x_j = 1$, and $\mathcal{P}_f(j) = 0$ where $x_j = 0$). Let $\mathcal{P}$ be the distribution output by algorithm A on f. We have $\|\mathcal{P} - \mathcal{P}_f\|_1 \leq \epsilon$ with probability at least $2/3$. Define a string $\widetilde{x} \in \{0,1\}^m$ by $\widetilde{x}_j = 1$ iff $\mathcal{P}(j) \geq 1/m$. Note that at each position $j \in [m]$ where $x_j \neq \widetilde{x}_j$, we have $|\mathcal{P}(j) - \mathcal{P}_f(j)| \geq 1/m$. Hence $\|\mathcal{P} - \mathcal{P}_f\|_1 \geq d(x, \widetilde{x})/m$. Since $\|\mathcal{P} - \mathcal{P}_f\|_1 \leq \epsilon$ (with probability at least $2/3$), the algorithm's output allows us to produce (with probability at least $2/3$) a string $\widetilde{x} \in \{0,1\}^m$ at Hamming distance $d(x, \widetilde{x}) \leq \epsilon m$ from x. But then it is easy to calculate that the mutual information between x and $\widetilde{x}$ is $\Omega(m)$ bits.

Finally, to put this in the communication setting, note that Bob can run the algorithm A, implementing each query to f by sending the $O(\log n)$-qubit query-register to Alice, who plugs in the right answer and sends it back (this idea comes from [9]). The overall communication is $O(q \log m)$ qubits.