Quantum Distinguishing Complexity, Zero-Error Algorithms, and Statistical Zero Knowledge

Shalev Ben-David
University of Waterloo, Waterloo, ON, Canada
shalev.b@uwaterloo.ca

Robin Kothari
Quantum Architectures and Computation (QuArC) group, Microsoft Research, Redmond, WA, USA
robin.kothari@microsoft.com

Abstract

We define a new query measure we call quantum distinguishing complexity, denoted $\mathrm{QD}(f)$ for a Boolean function f. Unlike a quantum query algorithm, which must output a state close to $|0\rangle$ on a 0-input and a state close to $|1\rangle$ on a 1-input, a "quantum distinguishing algorithm" can output any state, as long as the output states for any 0-input and 1-input are distinguishable.

Using this measure, we establish a new relationship in query complexity: For all total functions f, $\mathrm{Q}_0(f) = \widetilde{O}(\mathrm{Q}(f)^5)$, where $\mathrm{Q}_0(f)$ and $\mathrm{Q}(f)$ denote the zero-error and bounded-error quantum query complexity of f respectively, improving on the previously known sixth power relationship.

We also define a query measure based on quantum statistical zero-knowledge proofs, $\mathrm{QSZK}(f)$, which is at most $\mathrm{Q}(f)$. We show that $\mathrm{QD}(f)$ in fact lower bounds $\mathrm{QSZK}(f)$ and not just $\mathrm{Q}(f)$. $\mathrm{QD}(f)$ also upper bounds the (positive-weights) adversary bound, which yields the following relationships for all f: $\mathrm{Q}(f) \geq \mathrm{QSZK}(f) \geq \mathrm{QD}(f) = \Omega(\mathrm{Adv}(f))$. This sheds some light on why the adversary bound proves suboptimal bounds for problems like Collision and Set Equality, which have low QSZK complexity.

Lastly, we show implications for lifting theorems in communication complexity. We show that a general lifting theorem for either zero-error quantum query complexity or for QSZK would imply a general lifting theorem for bounded-error quantum query complexity.

2012 ACM Subject Classification Theory of computation → Quantum complexity theory

Keywords and phrases Quantum query complexity, quantum algorithms

Digital Object Identifier 10.4230/LIPIcs.TQC.2019.2

Related Version A full version of the paper is available at `https://arxiv.org/abs/1902.03660`.

Acknowledgements We thank Scott Aaronson, Mika Göös, John Watrous, and Ronald de Wolf for helpful conversations about this work. Most of this work was performed while the first author was at the Massachusetts Institute of Technology and the University of Maryland and the second author was at the Massachusetts Institute of Technology. This work was partially supported by NSF grant CCF-1629809.

1 Introduction

In the model of query complexity, we wish to compute some known Boolean function $f : \{0,1\}^n \to \{0,1\}$ on an unknown input $x \in \{0,1\}^n$ that we can access through an oracle that knows x. In the classical setting, the oracle responds with x_i when queried with an index $i \in [n]$. For quantum models, we use essentially the same oracle, but slightly modified to make it unitary. The bounded-error quantum query complexity of a function f, denoted $\mathrm{Q}(f)$, is the minimum number of queries to the oracle needed to compute the function f with probability greater than $2/3$ on any input x. In other words, the quantum query algorithm outputs a quantum state that is close to $|f(x)\rangle$.

In this paper we study "quantum distinguishing complexity," a query measure obtained by relaxing the output requirement of quantum query algorithms. Essentially, a quantum distinguishing algorithm for f doesn't need to compute $f(x)$, but merely needs to *behave differently on input x and input y if $f(x) \neq f(y)$*. We claim that this weaker notion of computation helps shed light on quantum query complexity and various lower bound techniques for it. We use quantum distinguishing complexity to prove a new query complexity relationship for total functions: $\mathrm{Q}_0(f) = O(\mathrm{Q}(f)^5 \log \mathrm{Q}(f))$. We also use it to explain why the non-negative adversary bound fails for some problems, to provide lower bound techniques for the query version of the complexity class QSZK, and to prove some reductions between lifting theorems in communication complexity.

1.1 Quantum distinguishing complexity

The *quantum distinguishing complexity* of a function $f : D \to \{0,1\}$ (where $D \subseteq \{0,1\}^n$), denoted $\mathrm{QD}(f)$, is the minimum number of queries needed to the input $x \in D$ to produce an output state $|\psi_x\rangle$, such that the output states corresponding to 0-inputs and 1-inputs are nearly orthogonal (or far apart in trace distance). Note that the usual bounded-error quantum query complexity of a function f, denoted $\mathrm{Q}(f)$, is defined similarly with the additional requirement that there should exist a 2-outcome measurement that (with high probability) accepts states corresponding to 1-inputs and rejects states corresponding to 0-inputs. Since measurements can only distinguish nearly orthogonal states, every quantum algorithm for computing f satisfies the definition of quantum distinguishing complexity. Hence for all functions f, we have $\mathrm{QD}(f) \leq \mathrm{Q}(f)$. We formally define quantum distinguishing complexity and establish some basic properties in Section 3.

This is a natural relaxation of bounded-error quantum query complexity and has been mentioned in passing in several prior works. Indeed, Barnum, Saks, and Szegedy call this measure $\mathrm{DQA}(f)$ in an early technical report [7, Remark 1]. This measure often comes up in discussions about the (positive-weights) adversary bound,[1] a lower bound for quantum query complexity introduced by Ambainis [4]. The (positive-weights) adversary bound, which we denote by $\mathrm{Adv}(f)$, has several variants [4, 5, 8, 23, 37], which are all essentially the same [32]. It was noted in several works [8, 21] that the proof that the adversary bound lower bounds quantum query complexity only uses the fact that the outputs corresponding to 0-inputs and 1-inputs are nearly orthogonal, and hence for all functions $\mathrm{QD}(f) = \Omega(\mathrm{Adv}(f))$. However, it is not the case that $\mathrm{QD}(f) = \Theta(\mathrm{Adv}(f))$ for all f, and we exhibit functions separating these measures.

Lastly, we show in Section 3 that this measure is the quantum analogue of a lower bound method for randomized query complexity called randomized sabotage complexity [11]. Hence this measure could also be called "quantum sabotage complexity."

1.2 Fifth power query relation

Our first result establishes a new relation between query measures for total functions. A total function is a function of the form $f : \{0,1\}^n \to \{0,1\}$, as opposed to a partial function, which is a function of the form $f : D \to \{0,1\}$, where $D \subseteq \{0,1\}^n$. We show a new upper bound on the zero-error quantum query complexity of f, denoted $\mathrm{Q}_0(f)$, in terms of its quantum

[1] The positive-weights adversary bound should not be confused with the stronger negative-weights adversary bound (also known as the general adversary bound), which essentially equals quantum query complexity [21, 24].

distinguishing complexity, and hence its quantum query complexity. The zero-error quantum query complexity of f is the minimum number of queries needed by a quantum algorithm that either outputs the correct answer $f(x)$ on input x, or outputs ? indicating that it does not know, but does this with probability at most $1/2$ on any input x. In Section 4 we prove the following.

▶ **Theorem 1.** *For all total functions $f : \{0,1\}^n \to \{0,1\}$, we have*

$$\mathrm{Q}_0(f) = O(\mathrm{QD}(f)^5 \log \mathrm{QD}(f)) = O(\mathrm{Q}(f)^5 \log \mathrm{Q}(f)). \tag{1}$$

Additionally, the algorithm also outputs a certificate for $f(x)$ when it outputs $f(x)$.

This is an improvement over the previous best relationship between zero-error and bounded-error quantum query complexity, $\mathrm{Q}_0(f) = O(\mathrm{Q}(f)^6)$ [9], which follows from $\mathrm{D}(f) = O(\mathrm{Q}(f)^6)$, where $\mathrm{D}(f)$ is deterministic query complexity. In fact, our result is the first upper bound on zero-error quantum query complexity that does not follow from an upper bound on zero-error randomized query complexity. Our proof borrows ideas from the classical result $\mathrm{R}_0(f) = O(\mathrm{R}(f)^2 \log \mathrm{R}(f))$ [27, 22], which is essentially optimal due to a nearly matching separation by Ambainis et al. [6].

1.3 Quantum statistical zero knowledge

Next we show that, surprisingly, quantum distinguishing complexity lower bounds a more powerful model of computation than quantum query complexity: the query complexity of computing a function using a quantum statistical zero-knowledge (QSZK) proof system. A QSZK proof system is an interactive protocol between a quantum verifier and a computationally unbounded, but untrusted prover in which the verifier learns the value of $f(x)$ but learns essentially no more. QSZK can also be characterized in terms of its complete problem Quantum State Distinguishability [34, 35].

In Section 5, we discuss the history of quantum statistical zero-knowledge proofs and define an associated query measure $\mathrm{QSZK}(f)$ based on the complete problem Quantum State Distinguishability. We establish some basic properties of our definition, such as $\mathrm{QSZK}(f) \leq \mathrm{Q}(f)$, which corresponds to the complexity class containment $\mathsf{BQP} \subseteq \mathsf{QSZK}$. We then show that quantum distinguishing complexity lower bounds QSZK complexity.

▶ **Theorem 2.** *For all (partial) Boolean functions f, $\mathrm{QD}(f) \leq \mathrm{QSZK}(f)$.*

As a corollary of Theorem 2 and $\mathrm{QD}(f) = \Omega(\mathrm{Adv}(f))$, we have for all (partial) functions f,

$$\mathrm{Q}(f) \geq \mathrm{QSZK}(f) \geq \mathrm{QD}(f) = \Omega(\mathrm{Adv}(f)). \tag{2}$$

This sheds some light on why the adversary bound sometimes proves poor lower bounds: it lower bounds a more powerful model of computation! For example, it is well known that the adversary bound cannot prove a super-constant lower bound for the collision problem [3]. It is also easy to see that the collision problem has a constant-query QSZK (and even classical SZK) protocol.

On the bright side, this gives us a new way to prove lower bounds on QSZK query complexity and prove oracle separations against the complexity class QSZK. For example, since we know the OR function on n bits has $\mathrm{Adv}(\mathrm{OR}) = \Omega(\sqrt{n})$, this yields an oracle A such that $\mathsf{NP}^A \not\subseteq \mathsf{QSZK}^A$, since the OR function has small certificates. A similar strategy was used recently by Menda and Watrous to show oracle separations against QSZK [26].

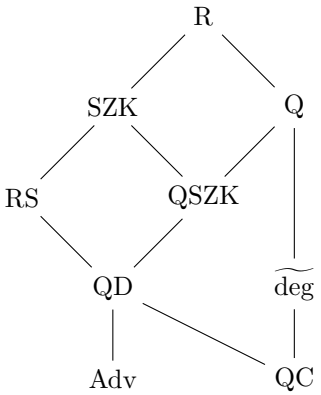

Figure 1 Relationships between measures. An upward line indicates that a measure is asymptotically upper bounded by the other measure. E.g., for all (partial) functions f, $\mathrm{Q}(f) = O(\mathrm{R}(f))$.

1.4 Comparison with other lower bounds

We compare quantum distinguishing complexity to the two main lower bound techniques for quantum query complexity: the (positive-weights) adversary bound and the polynomial method. Recall that the negative-weights adversary or general adversary completely characterizes quantum query complexity, so we do not compare quantum distinguishing complexity with it.

As noted earlier, the adversary bound is weaker than quantum distinguishing complexity since for all (partial) functions f, $\mathrm{QD}(f) = \Omega(\mathrm{Adv}(f))$. This implies that $\mathrm{QD}(f)$ coincides with $\mathrm{Q}(f)$ for most functions studied in the literature, since most quantum lower bounds are proved using the adversary method. Moreover, not only is quantum distinguishing complexity always larger than the adversary bound, it can be exponentially larger for partial functions and quadratically larger for total functions as we show in Theorem 3.

Another popular lower bound technique is the polynomial method [9], which uses the fact that the approximate degree of a function lower bounds $\mathrm{Q}(f)$. The approximate degree of a Boolean function f, denoted $\widetilde{\deg}(f)$, is the minimum degree of a real polynomial $p(x)$ over the input variables such that for all inputs x we have $|f(x) - p(x)| \leq 1/3$.

We do not know an exponential separation between quantum distinguishing complexity and approximate degree (for a partial function), since it is not even known if quantum query complexity can be exponentially larger than approximate degree for a partial function. We do, however, show in Theorem 3 that quantum distinguishing complexity can be polynomially larger than approximate degree for total functions.

▶ **Theorem 3.** *There exist total functions f and g with* $\mathrm{QD}(f) = \widetilde{\Omega}(\mathrm{Adv}(f)^2)$ and $\mathrm{QD}(g) \geq \widetilde{\deg}(g)^{4-o(1)}$.
There also exists an n-bit partial function h with $\mathrm{QD}(h) = \widetilde{\Omega}(n^{1/3})$ and $\mathrm{Adv}(h) = O(\log n)$.

This theorem is proved in Section 6. Figure 1 shows the known relationships between all the measures discussed in this paper. The measures RS and QC are introduced later, and refer to randomized sabotage complexity and quantum certificate complexity, respectively.

1.5 Lifting theorems

Most measures in query complexity have an analogous measure in communication complexity, which we denote with the superscript cc, such as $\mathrm{Q}^{\mathrm{cc}}(F)$ and $\mathrm{QSZK}^{\mathrm{cc}}(F)$. A lifting theorem is a result that transfers a lower bound on a query function f to a lower bound in communication

complexity for a lifted version of the function f, obtained by composing the function f with a hard communication problem G. For example, a lifting theorem is known for deterministic protocols, which means there exists a communication problem G such that for all functions f, $\mathrm{D}^{\mathrm{cc}}(f \circ G) = \Omega(\mathrm{D}(f))$ [29, 19].

Lifting theorems have been shown for some measures, such as nondeterministic query complexity [18] and (zero-error or bounded-error) randomized query complexity [20], and remain open for measures like zero-error and bounded-error quantum query complexity. Our next result, proved in Section 7, shows that if we could prove a lifting theorem for zero-error quantum query complexity or for QSZK query complexity, then we would get a lifting theorem for bounded-error quantum query complexity.

▶ **Theorem 4** (informal). *If a general lifting theorem holds using some gadget G for either zero-error quantum query complexity, i.e., $\mathrm{Q}_0^{\mathrm{cc}}(f \circ G) = \widetilde{\Omega}(\mathrm{Q}_0(f))$, or for quantum statistical zero-knowledge protocols, i.e., $\mathrm{QSZK}^{\mathrm{cc}}(f \circ G) = \widetilde{\Omega}(\mathrm{QSZK}(f))$, then we obtain a general lifting theorem for bounded-error quantum query complexity (up to logarithmic factors) with the same gadget G.*

In fact, the same conclusion follows from a weaker assumption. We can assume that the lifting theorem proves a lower bound on bounded-error quantum communication complexity assuming a lower bound on quantum distinguishing complexity. In other words, we can assume a lifting theorem of the form $\mathrm{Q}^{\mathrm{cc}}(f \circ G) = \widetilde{\Omega}(\mathrm{QD}(f))$, which is weaker than a QSZK lifting theorem since it assumes a stronger lower bound and proves a weaker one.

2 Preliminaries

We assume the reader is generally familiar with quantum computation [28] and query complexity (for more details, see [15]). We do not assume the reader is familiar with statistical zero-knowledge protocols.

For any positive integer n, let $[n] = \{1, \ldots, n\}$. We use $f(n) = \widetilde{O}(g(n))$ to mean there exists a constant k such that $f(n) = O(g(n) \log^k g(n))$ and similarly $f(n) = \widetilde{\Omega}(g(n))$ means $f(n) = \Omega(g(n)/\log^k g(n))$ for some constant k.

2.1 Distance measures

For any matrix A, we define the spectral norm of A, denoted $\|A\|$ as the largest singular value of A. The 1-norm of A, denoted $\|A\|_1$, is defined as $\mathrm{Tr}\left(\sqrt{A^\dagger A}\right)$, which is also equal to the sum of the singular values of A.

We define the trace distance between two quantum states ρ and σ as $\|\rho - \sigma\|_{\mathrm{tr}} = \frac{1}{2}\|\rho - \sigma\|_1$. The factor of $1/2$ makes this distance measure lie between 0 and 1 for density matrices. Trace distance is a useful distance measure since it exactly captures distinguishability of states and is non-increasing under quantum operations [28, Th. 9.2]. For pure states $|\psi\rangle$ and $|\phi\rangle$, trace distance is related to their inner product as follows [36, eq. 1.186].

$$\||\psi\rangle\langle\psi| - |\phi\rangle\langle\phi|\|_{\mathrm{tr}} = \sqrt{1 - |\langle\psi|\phi\rangle|^2}. \tag{3}$$

2.2 Quantum query complexity

In query complexity, we wish to compute a Boolean function f on an input x given query access to the bits of x. In this paper, we will mostly deal with functions with Boolean input and output. An n-bit function $f : \{0,1\}^n \to \{0,1\}$ is called a *total function*. An n-bit

function $f : D \to \{0,1\}$, where $D \subseteq \{0,1\}^n$, is called a *partial function* since it is defined on a subset of $\{0,1\}^n$. We will also refer to this subset D as the domain of f, or $\mathrm{Dom}(f)$. The goal in query complexity is to compute $f(x)$ while making the fewest queries to the oracle for the bits of x.

Classical algorithms have access to an oracle that given an index $i \in [n]$ outputs x_i, the i^{th} bit of x. A quantum algorithm is allowed access to a unitary map that implements this oracle, and is usually taken to be the unitary O_x which acts as follows on inputs $i \in [n]$ and $b \in \{0,1\}$: $O_x|i,b\rangle = |i, b \oplus x_i\rangle$. A quantum algorithm that uses the gate O_x in its circuit k times is said to have made k queries to the oracle.

Since we do not count the complexity of any other gates used in the algorithm, we can assume a k-query quantum algorithm always starts with the all-zeros state $|0^m\rangle$ and applies an oracle-independent unitary U_0 followed by the oracle O_x and so on. Thus a k-query quantum algorithm is specified by $k+1$ oracle-independent unitaries $U_0, \ldots, U_k$, which act on m output qubits. The state output by the quantum algorithm is $|\psi_x\rangle = U_k O_x U_{k-1} O_x \cdots O_x U_1 O_x U_0 |0^m\rangle$, where O_x is implicitly $(O_x \otimes \mathbb{1})$ if U_i acts on more qubits than O_x. If the quantum algorithm outputs a mixed state, then we assume it traces out some subset S of the m qubits, and hence outputs $\mathrm{Tr}_S(|\psi_x\rangle\langle\psi_x|)$. If the quantum algorithm outputs a bit, then we assume it measures the first qubit in the standard basis and outputs the result of that measurement.

We can now define the various complexity measures associated with quantum query complexity. We say the *bounded-error quantum query complexity* of computing a Boolean function f, $\mathrm{Q}(f)$, is the minimum k such that there exists a k-query quantum algorithm that on every $x \in \mathrm{Dom}(f)$ outputs $f(x)$ with probability greater than or equal to $2/3$. As usual, the constant $2/3$ is unimportant as long as it is a constant strictly greater than half, due to standard error reduction.

A *zero-error quantum algorithm* (or a Las Vegas quantum algorithm) never outputs an incorrect answer on an input $x \in \mathrm{Dom}(f)$, but is allowed to claim ignorance and answer `?` with probability at most $1/2$. The *zero-error quantum query complexity* of f, $\mathrm{Q}_0(f)$ is the minimum number of queries needed for a zero-error quantum algorithm to compute f. Note that $\mathrm{Q}(f) \leq \mathrm{Q}_0(f)$, since a zero-error algorithm can be turned into a bounded-error algorithm by simply outputting a random bit when the zero-error algorithm outputs `?`.

For zero-error quantum algorithms, there is a subtlety to do with whether or not the algorithm also produces a classical certificate for the input x. A certificate for x is a subset of bits of x, such that the value of $f(x)$ is completely determined by reading these bits alone. A classical zero-error algorithm can always be assumed to output such a certificate without loss of generality. However, this is not known to be true for zero-error quantum algorithms, and zero-error quantum algorithms that also output a certificate when they output a non-`?` answer are called *self-certifying* algorithms [14]. All the zero-error quantum algorithms in this paper are self-certifying, which makes our results stronger since we only prove upper bounds on zero-error quantum algorithms.

3 Quantum distinguishing complexity

3.1 Definition

We now define quantum distinguishing complexity more formally. As explained in the introduction, instead of requiring that the quantum algorithm output the value of the function $f(x)$, as in standard quantum query complexity, we only want the quantum algorithm's outputs to be distinguishable (or nearly orthogonal) for 0-inputs and 1-inputs.

As an example of how these definitions differ, consider the collision problem. In this problem, we are given an input $x \in [n]^n$ and we are promised that if we view x as a function from $[n] \to [n]$, the function is either 1-to-1 or 2-to-1. The goal is to distinguish these two cases under the assumption that the input satisfies this promise. In this problem, since every 0-input and 1-input differ in exactly half the positions $i \in [n]$, our quantum algorithm can simply create the state $|\psi_x\rangle = \frac{1}{\sqrt{n}} \sum_i |i, x_i\rangle$ and the states corresponding to 0-inputs and 1-inputs will have trace distance $\Omega(1)$. Thus this problem has quantum distinguishing complexity $O(1)$, but its quantum query complexity is $\Theta(n^{1/3})$ [3].

▶ **Definition 5** (Quantum Distinguishing complexity). *Let $f : D \to \{0,1\}$, where $D \subseteq \{0,1\}^n$, be an n-bit partial function. $\mathrm{QD}(f)$ is defined as the smallest integer k such that there exists a k-query quantum algorithm that on input $x \in D$ outputs a quantum state ρ_x such that*

$$\forall x, y \in D \text{ with } f(x) \neq f(y), \quad \|\rho_x - \rho_y\|_{\mathrm{tr}} \geq 1/6. \tag{4}$$

Note that the definition is robust to minor changes. First, we allow outputting mixed states, although this does not offer any additional power over only outputting pure states. The reason is that we can always assume that the quantum algorithm is pure until the final step where some subset of qubits is traced out. But if two states are far apart in trace distance after a partial trace, then they were far apart to begin with since trace distance is non-increasing under partial trace.

The constant $1/6$ in Definition 5 is also arbitrary and any constant in $(0, 1)$ would not change the measure by more than a multiplicative constant. This is because we can increase the trace distance between the states by outputting multiple copies of the states. We choose the constant $1/6$ purely for aesthetic reasons: This choice ensures that the result in Theorem 2 has no constant factors.

3.2 Properties

We can now establish some basic properties of quantum distinguishing complexity. First, let us formally show that quantum distinguishing complexity lower bounds quantum query complexity.

▶ **Proposition 6.** *For all (partial) Boolean functions f, $\mathrm{QD}(f) \leq \mathrm{Q}(f)$.*

Proof. Let $\mathrm{Q}(f) = k$ and consider the k-query algorithm that witnesses this fact. Let p_x be the probability that this k-query algorithm, when run on input x, outputs 1 upon measuring the first qubit. Since the algorithm computes f with bounded error, we know that for all 1-inputs x, $p_x \geq 2/3$, and for all 0-inputs y, $p_y \leq 1/3$.

Now consider the single-qubit state ρ_x, which is obtained by taking the final state of this algorithm, tracing out all the qubits except the first one, and then applying a completely dephasing channel to it. This state is $\rho_x = \left(\begin{smallmatrix} 1-p_x & 0 \\ 0 & p_x \end{smallmatrix}\right)$. Thus for all x, y with $f(x) \neq f(x)$, $\|\rho_x - \rho_y\|_{\mathrm{tr}} = |p_x - p_y| \geq 1/3$. ◀

As noted in the introduction, quantum distinguishing complexity is also lower bounded by the adversary bound, i.e., $\mathrm{QD}(f) = \Omega(\mathrm{Adv}(f))$.

We do not prove this since this follows from the arguments that establish that the adversary bound is a lower bound on quantum query complexity [4, 5, 8, 23, 37, 32], since all these proofs only use the fact that the states output on 0-inputs and 1-inputs are nearly orthogonal.

Quantum distinguishing complexity is also superior to quantum certificate complexity $\mathrm{QC}(f)$, as we show in Proposition 8. Quantum certificate complexity is a lower bound on quantum query complexity defined by Aaronson [1]. It was later shown that quantum certificate complexity also lower bounds approximate polynomial degree [22].

Before proving Proposition 8, we first define certificate complexity, randomized certificate complexity, and quantum certificate complexity.

▶ **Definition 7** (Certificate complexity). *For any (partial) function f and input $x \in \mathrm{Dom}(f)$, consider the partial function f^x defined on the domain $\{x\} \cup \{y \in \mathrm{Dom}(f) : f(y) \neq f(x)\}$ that satisfies $f^x(x) = 1$ and $f^x(y) = 0$ for all $y \in \mathrm{Dom}(f)$ with $f(y) \neq f(x)$.*

We define the certificate complexity of f, denoted $\mathrm{C}(f)$, the randomized certificate complexity of f, denoted $\mathrm{RC}(f)$, and the quantum certificate complexity of f, denoted $\mathrm{QC}(f)$, as follows:

$$\mathrm{C}(f) = \max_{x \in \mathrm{Dom}(f)} \mathrm{D}(f^x), \quad \mathrm{RC}(f) = \max_{x \in \mathrm{Dom}(f)} \mathrm{R}(f^x), \quad \text{and} \quad \mathrm{QC}(f) = \max_{x \in \mathrm{Dom}(f)} \mathrm{Q}(f^x). \tag{5}$$

The problem f^x is clearly no harder than computing f itself in any model of computation, and hence these are lower bounds on their respective measures, i.e., $\mathrm{C}(f) \leq \mathrm{D}(f)$, $\mathrm{RC}(f) \leq \mathrm{R}(f)$, and $\mathrm{QC}(f) \leq \mathrm{Q}(f)$. We can now prove that $\mathrm{QD}(f)$ is a better lower bound on $\mathrm{Q}(f)$ than $\mathrm{QC}(f)$.

▶ **Proposition 8.** *For all (partial) Boolean functions f, $\mathrm{QD}(f) = \Omega(\mathrm{QC}(f))$.*

Proof. Let $\mathrm{QD}(f) = k$ and consider the k-query quantum algorithm that witnesses this fact. We can use this algorithm to solve f^x for any $x \in \mathrm{Dom}(f)$. Consider the output of the algorithm on input x before the partial trace operation and call this $|\psi_x\rangle$. The trace distance between $|\psi_x\rangle$ and $|\psi_y\rangle$ for $y \in \mathrm{Dom}(f)$ with $f(y) \neq f(x)$ is at least $1/6$ since trace distance is non-increasing under partial trace [28, Th. 9.2].

Now we construct an algorithm for f^x from this algorithm to show that $\mathrm{Q}(f^x) = O(\mathrm{QD}(f))$. To do so, we run the supposed algorithm and measure whether the output state is $|\psi_x\rangle$ or not and accept only when the measurement accepts. This yields an algorithm that outputs 1 on x with probability 1 and accepts inputs y with $f(x) \neq f(x)$ with some constant probability strictly less than 1. More precisely, the acceptance probability is $|\langle\psi_x|\psi_y\rangle|^2 \leq 1 - (1/6)^2$ due to the relationship between inner product and trace distance for pure states. Repeating this algorithm a constant number of times yields a bounded-error quantum algorithm for f^x. ◀

3.3 Relation with randomized sabotage complexity

We start by reviewing the definition of randomized sabotage complexity, as presented in [11]. Fix a (partial) Boolean function $f : D \to \{0,1\}$ with $D \in \{0,1\}^n$. For any pair $x, y \in \mathrm{Dom}(f)$ such that $f(x) \neq f(y)$, let $p \in \{0,1,*\}^n$ be the partial assignment of all bits where x and y agree (with the symbol $*$ used for the bits where x and y disagree). We call p a "sabotaged input", imagining that a saboteur replaced bits of x with $*$ symbols until it was no longer possible to determine $f(x)$.

Let $S_* \subseteq \{0,1,*\}^n$ be the set of all sabotaged inputs to f, that is, the set of all partial assignments that are consistent with both a 0-input and a 1-input to f. Let $S_\dagger \in \{0,1,\dagger\}^n$ be the same as S_*, except that the $\dagger$ symbol is used instead of the $*$ symbol. Finally, let $f_{\mathrm{sab}} : S_* \cup S_\dagger \to \{0,1\}$ be the function that takes a sabotaged input and identifies whether it has $*$ symbols or $\dagger$ symbols, promised that it contains only one type of symbol. Intuitively, f_{sab} is a decision problem that forces an algorithm computing it to find a $*$ or $\dagger$. We then define $\mathrm{RS}(f) := \mathrm{R}_0(f_{\mathrm{sab}})$, the expected running time of a zero-error randomized algorithm computing f_{sab}.

To show that $\mathrm{RS}(f)$ is larger than $\mathrm{QD}(f)$ for all f, we will define a classical measure analogous to $\mathrm{QD}(f)$. We will then show this measure is equivalent to $\mathrm{RS}(f)$.

▶ **Definition 9** (Randomized distinguishing complexity)**.** *Let $f : D \to \{0,1\}$, where $D \subseteq \{0,1\}^n$, be an n-bit partial function.* $\mathrm{RD}(f)$ *is defined as the smallest integer k such that there exists a k-query randomized algorithm that on input $x \in D$ outputs a sample from a probability distribution d_x such that*

$$\forall x, y \in D \text{ with } f(x) \neq f(y), \quad D_{\mathrm{TV}}(d_x, d_y) \geq 1/6, \tag{6}$$

where $D_{\mathrm{TV}}(\cdot,\cdot)$ stands for the total variation distance between probability distributions.

Since quantum algorithms can simulate classical algorithms, we immediately get that $\mathrm{QD}(f) \leq \mathrm{RD}(f)$. Next, we will show that $\mathrm{RD}(f) = \Theta(\mathrm{RS}(f))$, completing the argument that $\mathrm{QD}(f) = O(\mathrm{RS}(f))$.

▶ **Theorem 10.** *Let f be a partial Boolean function. Then* $\mathrm{RS}(f)/12 \leq \mathrm{RD}(f) \leq (12/11)\,\mathrm{RS}(f)$.

Proof. First, we show that $\mathrm{RS}(f) \leq 12\,\mathrm{RD}(f)$. Let A be an optimal randomized algorithm for $\mathrm{RD}(f)$, that on input x outputs a sample from the distribution d_x. Let $z \in \mathrm{Dom}(f_{\mathrm{sab}})$ be a sabotaged input, and consider running A on z. Since z is sabotaged, there are inputs x and y with $f(x) \neq f(y)$ that are both consistent with the non-$*$, non-† bits of z. The variation distance between d_x and d_y is at least $1/6$.

A randomized algorithm can be viewed as a probability distribution over deterministic algorithms. Split the support of the distribution for A into two parts: a set S consisting of deterministic algorithms that, when run on z, query a $*$ or †, and a set T consisting of deterministic algorithms that don't query a $*$ or † when run on z. Note that algorithms in T behave the same on x and y. If A samples an algorithm from T with probability p, the total variation distance between the run of A on x and the run of A on y must therefore be at most $2(1-p)$. Since this is at least $1/6$, we have $p \leq 11/12$. Hence when A is run on z, it queries a $*$ or † with probability at least $1/12$.

If we repeat A whenever it does not query a $*$ or †, we get an algorithm that always finds such an entry and uses at most $12\,\mathrm{RD}(f)$ queries on expectation. This is a zero-error randomized algorithm for f_{sab}, so $\mathrm{RS}(f) \leq 12\,\mathrm{RD}(f)$.

We now handle the other direction, showing $\mathrm{RD}(f) \leq (12/11)\,\mathrm{RS}(f)$. Let A be an optimal zero-error randomized algorithm for f_{sab}. It makes $\mathrm{RS}(f)$ queries on expectation, and always finds a $*$ or † in any sabotaged input. Consider the algorithm B that, on input $x \in \mathrm{Dom}(f)$, runs A for at most $2\,\mathrm{RS}(f)$ queries and outputs the partial assignment it queried (that is, it outputs all the pairs (i, x_i) that were queried by the algorithm A).

Let x and y be inputs to f with $f(x) \neq f(y)$. Let z be the sabotaged input defined by x and y, that is, $z_i = *$ if $x_i \neq y_i$ and $z_i = x_i = y_i$ otherwise. By Markov's inequality, after $(12/11)\,\mathrm{RS}(f)$ queries, A finds a $*$ with probability at least $1/12$ when it is run on z. This means that when A is run on x, it queries an index i for which $x_i \neq y_i$ with probability at least $1/12$. When this happens, the output of $B(x)$ is not in the support of d_y. This means d_x puts weight at least $1/12$ on symbols not in the support of d_y. Conversely, d_y puts weight at least $1/12$ on symbols not in the support of d_x. The total variation distance between the two distributions is therefore at least $1/6$, meaning B is a valid $\mathrm{RD}(f)$ algorithm. We conclude that $\mathrm{RD}(f) \leq (12/11)\,\mathrm{RS}(f)$. ◀

Combined with $\mathrm{QD}(f) \leq \mathrm{RD}(f)$, this theorem gives us the following corollary.

▶ **Corollary 11.** *For all (partial) Boolean functions f,* $\mathrm{QD}(f) = O(\mathrm{RS}(f))$.

4 Fifth power query relation

In this section we prove a new relationship between zero-error quantum query complexity and quantum distinguishing complexity and bounded-error quantum query complexity, restated below.

▶ **Theorem 1.** *For all total functions $f : \{0,1\}^n \to \{0,1\}$, we have*

$$\mathrm{Q}_0(f) = O(\mathrm{QD}(f)^5 \log \mathrm{QD}(f)) = O(\mathrm{Q}(f)^5 \log \mathrm{Q}(f)). \tag{1}$$

Additionally, the algorithm also outputs a certificate for $f(x)$ when it outputs $f(x)$.

Our proof uses ideas from an analogous classical result [27, 22] and the main quantum ingredient used is the hybrid argument of Bennett, Bernstein, Brassard, and Vazirani [12]. We now describe and prove a version of the hybrid argument that we use.

4.1 Hybrid argument

We start by defining the concept of a *sensitive block*. For a string $x \in \{0,1\}^n$ and a subset of input bits $B \subseteq [n]$, which we call a block, we use x^B to denote the input with all bits in B flipped. In other words, x^B agrees with x on all positions outside B and disagrees on B. For a function f and an input $x \in \mathrm{Dom}(f)$, we say a block B is a sensitive block if $f(x) \neq f(x^B)$.

Now any algorithm that computes f must also be able to distinguish x from x^B, where B is a sensitive block. Any algorithm that can distinguish x from x^B must "look at" the bits in B in some informal sense. For classical algorithms, this simply means the algorithm has to query a bit from B with high probability. The analogous statement for quantum algorithms is not so clear, since quantum algorithms can query all input bits in superposition. Nevertheless, the hybrid argument still allows us to formalize this intuition in the quantum setting. The hybrid argument asserts that the total weight of queries within the sensitive block (i.e., the total sum of probabilities of querying within the sensitive block over the course of the algorithm) cannot be too small [12]:

▶ **Lemma 12** (Hybrid Argument)**.** *Let $x \in \{0,1\}^n$ be an input, and let $B \subseteq [n]$ be a block. Let Q be a T-query quantum algorithm that accepts x and rejects x^B with high probability, or more generally produces output states that are a constant distance apart in trace distance for x and x^B. Let m_i^t be the probability that, when Q is run on x for t queries and then subsequently measured, it is found to be querying position i of x (i.e., the query register collapses to $|i\rangle$). Then*

$$\sum_{t=1}^{T} \sum_{i \in B} m_i^t = \Omega\left(\frac{1}{T}\right). \tag{7}$$

Note that for a randomized algorithm, we would have $\Omega(1)$ on the right-hand side instead of $\Omega(1/T)$, since a randomized algorithm must look within B (with high probability) at some point during its execution. This lemma was implicitly proven in [12]. We reproduce the proof in Appendix A for the reader's convenience.

4.2 New upper bound

To prove our result we also need to upper bound the number of minimal sensitive blocks of a function. It is not too hard to show that any minimal sensitive block has size at most the sensitivity of f, $s(f)$, which is the maximum number of sensitive blocks of size 1 over all

inputs x. Since there are at most $\binom{n}{\mathrm{s}(f)} = O(n^{\mathrm{s}(f)})$ different subsets of n positions of size $\mathrm{s}(f)$, we know that the number of minimal sensitive blocks is at most this quantity. Kulkarni and Tal [22] improve this simple upper bound replacing n with randomized certificate complexity $\mathrm{RC}(f)$ (Definition 7).

▶ **Lemma 13.** *For any total function $f : \{0,1\}^n \to \{0,1\}$ and any input $x \in \{0,1\}^n$, the number of minimal sensitive blocks of x with respect to f is at most $O(\mathrm{RC}(f)^{\mathrm{s}(f)})$.*

We are now ready to prove Theorem 1.

Proof of Theorem 1. Let Q be the optimal quantum distinguishing algorithm for f, that uses $T = \mathrm{QD}(f)$ queries. Consider running the following quantum algorithm P on oracle input $x \in \{0,1\}^n$:

1. Pick $t \in [T]$ uniformly at random.
2. Run Q on x for t queries and measure the query register.
3. Write down (on a classical tape) the position i where Q is found to be querying, as well as the query output x_i.

The algorithm P uses $t \leq T$ quantum queries. Now that the probability P wrote down the index i is $(1/T)\sum_{t=1}^{T} m_i^t$. For any block $B \subseteq [n]$, the probability that P wrote down some index in B is

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{i\in B} m_i^t. \tag{8}$$

If B is a sensitive block for the input x, then the hybrid argument (Lemma 12) implies the probability that our new algorithm P outputs an index in B is $\Omega(1/T^2)$.

Next, we repeat the algorithm P several times. We claim that after $O(T^2\,\mathrm{s}(f)\log \mathrm{RC}(f))$ repetitions, the outputs of P constitute a certificate for x with constant probability.

To see this, note that for any minimal sensitive block B of the input x, the probability that some run of P (out of the $O(T^2\,\mathrm{s}(f)\log \mathrm{RC}(f))$ many runs) queries in the block B is $1 - O(\mathrm{RC}(f)^{-\mathrm{s}(f)})$. This is because T^2 repetitions boost the probability of querying in a minimal sensitive block from $\Omega(1/T^2)$ to $\Omega(1)$, and then $\mathrm{s}(f)\log \mathrm{RC}(f)$ repetitions of this boosted algorithm further boost the probability to the claimed bound. Hence, by Lemma 13 and the union bound, there is a constant probability that these runs of P query a bit in every minimal sensitive block of the input x. But a set of bits that intersects every sensitive block of x is a certificate for x. Thus these runs of P output a certificate for the input x with constant probability.

Any algorithm that finds a certificate with constant probability can be turned into a zero-error algorithm by repeating whenever a certificate is not found. We therefore get a zero-error algorithm that works simply by repeating P a sufficient number of times. Note that P uses $O(T)$ quantum queries and must be repeated $O(T^2\,\mathrm{s}(f)\log \mathrm{RC}(f))$ times. Recalling that $T = \mathrm{QD}(f)$, we get

$$\mathrm{Q}_0(f) = O(\mathrm{QD}(f)^3\,\mathrm{s}(f)\log \mathrm{RC}(f)). \tag{9}$$

We can simplify this to $\mathrm{Q}_0(f) = O(\mathrm{QD}(f)^5 \log \mathrm{QD}(f))$, since $\mathrm{s}(f) = O(\mathrm{RC}(f)) = O(\mathrm{QC}(f)^2)$ [1] and $\mathrm{QC}(f) = O(\mathrm{QD}(f))$ (Proposition 8). ◀

5 Quantum statistical zero knowledge

5.1 History

The subject of statistical zero-knowledge proof systems has a rich history in the classical setting, and the interested reader is referred to the paper of Sahai and Vadhan [30]. Informally, the complexity class SZK contains problems that can be solved by a probabilistic polynomial-time verifier interacting with a computationally unbounded prover (like the class IP) with the additional restriction that the verifier not learn anything from the prover (statistically) other than the answer to the problem. From this it is clear that BPP $\subseteq$ SZK, since the verifier can simply not interact with the prover, and SZK $\subseteq$ IP, since IP is simply SZK without the zero-knowledge constraint.

More surprisingly, it is also known that SZK = coSZK, and that we can assume without loss of generality that the interaction is only one round and uses public randomness, which means SZK $\subseteq$ AM $\cap$ coAM. Another interesting subtlety is that SZK can be defined assuming an honest verifier, one who does not deviate from the protocol to learn more, or a cheating verifier, who may deviate from the protocol. It turns out that these definitions lead to the same complexity class [17]. The class SZK also has a much simpler characterization in terms of a complete problem called *statistical difference*, as shown by Sahai and Vadhan [30], which yields easier proofs of some of these facts. Informally, in the statistical difference problem we are given two circuits that sample from probability distributions, and the task is determine whether the distributions are far or close in total variation distance.

On the quantum side, (honest-verifier) QSZK was first defined by Watrous [34], and like the classical case, it satisfies BQP $\subseteq$ QSZK $\subseteq$ QIP. The same paper strengthened these obvious containments by showing that QSZK is closed under complement (i.e., QSZK = coQSZK) and that the protocol can be assumed to be one round, which gives QSZK $\subseteq$ QIP(2). Watrous also showed that QSZK has a complete problem, called *quantum state distinguishability*, which is a quantum generalization of the statistical difference problem of Sahai and Vadhan. In this problem, we are given two quantum circuits outputting mixed states and have to decide if the states are far apart or close in trace distance. Later, Watrous [35] also showed that honest-verifier QSZK and cheating-verifier QSZK are the same, as in the classical case.

5.2 Definition

We now define a query analogue of quantum statistical zero-knowledge. Instead of defining QSZK(f) in terms of an interactive zero-knowledge protocol for f, we use the complete problem characterization by Watrous. This yields a considerably simpler definition of QSZK in the query setting.[2]

▶ **Definition 14** (QSZK). *Let $f : D \to \{0,1\}$, where $D \subseteq \{0,1\}^n$, be an n-bit partial function.* QSZK(f) *is defined as the smallest integer k such that there exists two quantum query algorithms making k queries in total that on input $x \in D$ output states ρ_x and σ_x of the same size such that*

- $\forall x \in D$ *with* $f(x) = 1$, $\|\rho_x - \sigma_x\|_{\mathrm{tr}} \geq 2/3$,
- $\forall x \in D$ *with* $f(x) = 0$, $\|\rho_x - \sigma_x\|_{\mathrm{tr}} \leq 1/3$.

[2] The complete problem is often used to define SZK (and its variants, like NISZK) in query complexity and communication complexity (for example, see [13, 33]). It is not obvious whether the definition via an interactive proof and the definition via the complete problem coincide exactly as the problem is complete under polynomial-time reductions, which may add polynomial overhead.

This definition is also robust to some changes. In particular, the constants $2/3$ and $1/3$ can be replaced by any constants $\alpha \in [0,1]$ and $\beta \in [0,1]$ as long as $\alpha^2 > \beta$. Hence an alternate definition with 0.99 instead of $2/3$ and 0.01 instead of $1/3$ leads to the same complexity measure up to multiplicative constants. This follows from the analogous property of the complexity class QSZK, which was shown by Watrous [34] (see Theorem 1 in the conference version or Theorem 5 in the full version for more details).

5.3 Properties

As a sanity check, let us prove the query analog of the obvious containment BQP $\subseteq$ QSZK.

▶ **Proposition 15.** *For all (partial) Boolean functions f, $\mathrm{QSZK}(f) \leq \mathrm{Q}(f)$.*

Proof. Let $\mathrm{Q}(f) = k$ and consider the k-query algorithm that witnesses this fact. Let p_x be the probability that this k-query algorithm when run on input x outputs 1 upon measuring the first qubit. Since the algorithm computes f with bounded error, we know that $p_x \geq 2/3$ for 1-inputs and $p_x \leq 1/3$ for 0-inputs.

Now consider the single-qubit state ρ_x, which is obtained by taking the final state of this algorithm, tracing out all the qubits except the first one, and then applying a completely dephasing channel to it. This is equivalent to measuring the first qubit in the standard basis and outputting $|b\rangle$ when the result is b. This state is $\rho_x = \left(\begin{smallmatrix} 1-p_x & 0 \\ 0 & p_x \end{smallmatrix}\right)$. Let us also define σ_x as $\left(\begin{smallmatrix} 1 & 0 \\ 0 & 0 \end{smallmatrix}\right)$ for all x.

Now let us check that the conditions of Definition 14 are satisfied by these states. For all inputs x, we have $\|\rho_x - \sigma_x\|_{\mathrm{tr}} = |p_x|$. And we know that $p_x \geq 2/3$ for 1-inputs and $0 \leq p_x \leq 1/3$ for 0-inputs, which completes the proof. ◀

The measure $\mathrm{QSZK}(f)$ also satisfies another useful property, that $\mathrm{QSZK}(f) = \Theta(\mathrm{QSZK}(\neg f))$. This is the analogue of the result that QSZK = coQSZK [34]. Since we do not use this property, we only provide a sketch of the proof in Appendix B.

5.4 Relation with adversary bound

We have already showed that $\mathrm{QD}(f) \leq \mathrm{Q}(f)$ (Proposition 6) and $\mathrm{QSZK}(f) \leq \mathrm{Q}(f)$ (Proposition 15). We now show that $\mathrm{QD}(f)$ is actually smaller than $\mathrm{QSZK}(f)$.

▶ **Theorem 2.** *For all (partial) Boolean functions f, $\mathrm{QD}(f) \leq \mathrm{QSZK}(f)$.*

Proof. Let $\mathrm{QSZK}(f) = k$ and consider the quantum algorithms that witnesses this fact. We claim that the tensor product of outputs of these algorithms already satisfies the conditions in Definition 5 and hence proves $\mathrm{QD}(f) \leq k$.

To see this, observe that the algorithm outputs the state $\rho_x \otimes \sigma_x$ on input x, which satisfies the conditions of Definition 14. More precisely, this means for any x and y such that $f(x) = 1$ and $f(y) = 0$, we know that $\|\rho_x - \sigma_x\|_{\mathrm{tr}} \geq 2/3$ and $\|\rho_y - \sigma_y\|_{\mathrm{tr}} \leq 1/3$. We want to show that

$$\|\rho_x \otimes \sigma_x - \rho_y \otimes \sigma_y\|_{\mathrm{tr}} \geq 1/6. \tag{10}$$

Since trace distance is non-increasing under partial trace, we have $\|\rho_x \otimes \sigma_x - \rho_y \otimes \sigma_y\|_{\mathrm{tr}} \geq \|\rho_x - \rho_y\|_{\mathrm{tr}}$ and $\|\rho_x \otimes \sigma_x - \rho_y \otimes \sigma_y\|_{\mathrm{tr}} \geq \|\sigma_x - \sigma_y\|_{\mathrm{tr}}$, which imply

$$\|\rho_x \otimes \sigma_x - \rho_y \otimes \sigma_y\|_{\mathrm{tr}} \geq \max\left\{\|\rho_x - \rho_y\|_{\mathrm{tr}}, \|\sigma_x - \sigma_y\|_{\mathrm{tr}}\right\}.$$

Now if we can show the right-hand side is at least $1/6$, then we are done. To show this, toward a contradiction assume that $\max\left\{\|\rho_x - \rho_y\|_{\mathrm{tr}}, \|\sigma_x - \sigma_y\|_{\mathrm{tr}}\right\} < 1/6$. Then we have

$$\begin{aligned}\|\rho_x - \sigma_x\|_{\mathrm{tr}} &= \|\rho_x - \rho_y + \rho_y - \sigma_y + \sigma_y - \sigma_x\|_{\mathrm{tr}} \\ &\leq \|\rho_x - \rho_y\|_{\mathrm{tr}} + \|\rho_y - \sigma_y\|_{\mathrm{tr}} + \|\sigma_y - \sigma_x\|_{\mathrm{tr}} \\ &< 1/6 + 1/3 + 1/6 = 2/3,\end{aligned}$$

which contradicts $\|\rho_x - \sigma_x\|_{\mathrm{tr}} \geq 2/3$. ◀

As noted, as a corollary of this theorem and $\mathrm{QD}(f) = \Omega(\mathrm{Adv}(f))$, we have for all (partial) functions f, $\mathrm{QSZK}(f) = \Omega(\mathrm{Adv}(f))$.

This can be used to prove lower bounds on QSZK protocols for functions. For example, consider the OR function and let us try to compute it with an interactive protocol without the zero-knowledge requirement. It is easy to see that when $\mathrm{OR}(x) = 1$, a computationally unbounded prover can simply send over the location of a bit i such that $x_i = 1$, which can be checked using only 1 query. Of course, this protocol leaks information and in particular lets the verifier know the location of a 1. But is it necessary that an efficient protocol for OR must leak information? Our lower bound says this must be the case, because $\mathrm{Adv}(\mathrm{OR}) = \Omega(\sqrt{n})$ and hence any zero-knowledge protocol for the function must make $\Omega(\sqrt{n})$ queries.

6 Comparison with other lower bounds

In this section, we establish the separations between quantum distinguishing complexity and the adversary bound and the polynomial method claimed in Theorem 3.

To prove this, we will compose known functions with the index function and establish the behavior of quantum distinguishing complexity under composition with the index function. This kind of composition was also studied by Chen [16], who used it to show an oracle separation between $\mathsf{P}^{\mathsf{SZK}}$ and QSZK.

6.1 Index functions

Let $\textsc{Ind}_k : \{0,1\}^{k+2^k} \to \{0,1\}$ denote the index function, the function that on input (x, y) with $x \in \{0,1\}^k$ and $y \in \{0,1\}^{2^k}$, outputs the bit of y indexed by the string x. We wish to study the composition of the index function with an arbitrary Boolean function f, but composed only on the first k bits of the index function. We'll denote this composition by $\textsc{Ind}_k \circ_k f$. More precisely, if f is an n-bit function, $\textsc{Ind}_k \circ_k f$ is a function on $nk + 2^k$ bits that evaluates f on the first k n-bit strings to obtain a binary string x of length k, and then uses x to index into the next 2^k bits of the input and outputs the bit indexed by x.

In addition to the index function, which is total, we will also study a function we call the "unambiguous index function," $\mathrm{U}\textsc{Ind}_k$. This is a partial function defined similarly to the index function, except that the location of the array y pointed to by the first part of the input is "marked," and we are promised that no other bits of the array are "marked." More explicitly, the function is defined on $k + 2 \cdot 2^k$ bits, with the first k bits indexing a pair of adjacent bits in the remainder of the input. So if the first part of the input represents the integer x, that means it points to the cells $2x$ and $2x + 1$ in the second part of the input. The output of $\mathrm{U}\textsc{Ind}_k$ is the first bit of the pair pointed to, i.e., it will be the bit stored at array location $2x$. Moreover, we are promised that the second bit of this pair (the bit at array location $2x + 1$) will always be 1, and also that the second bit in every *other* pair (i.e., other than the pair $2x$, $2x + 1$) will always be 0.

Intuitively, there is only one strategy to solve $\textsc{Ind}_k$, which is to read the first k bits and find the cell pointed to. But to solve $\textsc{UInd}_k$, there are two good strategies: either read the first k bits (and determine x), or search the remainder of the input for the unique position where the second bit of a pair is 1, which marks the cell pointed to by x.

6.2 Index function composition

We now examine the behavior of quantum distinguishing complexity under composition with the Index and Unambiguous Index functions. To prove our result, we need the following strong direct product theorem for quantum query complexity due to Lee and Roland [25]:

▶ **Theorem 16** (Strong direct product). *Let f be a partial Boolean function with $\mathrm{Dom}(f) \subseteq \{0,1\}^n$, and let $f^{(k)} : \mathrm{Dom}(f)^k \to \{0,1\}^k$ be the task of solving k independent inputs to f simultaneously. Then any quantum algorithm that solves $f^{(k)}$ with success probability at least $(5/6)^k$ uses $\Omega(k\,\mathrm{Q}(f))$ queries.*

We prove the following composition theorem in Appendix C.

▶ **Theorem 17.** *There is a $c > 0$ such that for any partial function f, if $k \geq c \log \mathrm{Q}(f)$, then*

$$\mathrm{QD}(\textsc{Ind}_k \circ_k f) = \Theta(\mathrm{Q}(\textsc{Ind}_k \circ_k f)) = \Theta(k\,\mathrm{Q}(f)) \tag{11}$$

$$\mathrm{QD}(\textsc{UInd}_k \circ_k f) = \Theta(\mathrm{Q}(\textsc{UInd}_k \circ_k f)) = \Theta(k\,\mathrm{Q}(f)). \tag{12}$$

In other words, composing a function with a large index gadget makes QD *and* Q *coincide.*

6.3 Separations

Using this theorem we can now establish Theorem 3, restated for convenience:

▶ **Theorem 3.** *There exist total functions f and g with $\mathrm{QD}(f) = \widetilde{\Omega}(\mathrm{Adv}(f)^2)$ and $\mathrm{QD}(g) \geq \widetilde{\deg}(g)^{4-o(1)}$.*
There also exists an n-bit partial function h with $\mathrm{QD}(h) = \widetilde{\Omega}(n^{1/3})$ and $\mathrm{Adv}(h) = O(\log n)$.

Proof. There exists an n-bit total function f' with a quadratic separation between quantum query complexity and the adversary bound, i.e., $\mathrm{Q}(f') = \widetilde{\Omega}(\mathrm{Adv}(f')^2)$. The function is k-sum with $k \approx \log n$ (see [10, 2] for more details). Now consider the function $f = \textsc{Ind}_k \circ f'$, where $k = \Omega(\log \mathrm{Q}(f))$. By Theorem 17, the QD of these functions increases to Q. However, since the adversary bound satisfies a composition theorem [21], its value only increases by a factor of k. Thus $\mathrm{QD}(f) = \widetilde{\Omega}(\mathrm{Adv}(f)^2)$.

Similarly, if we start with the collision problem which has $\mathrm{Q}(h') = \Theta(n^{1/3})$ [3], but $\mathrm{Adv}(h') = O(1)$, and define $h = \textsc{Ind}_k \circ h'$ for $k = \Theta(\log n)$, then $\mathrm{QD}(h) = \widetilde{\Omega}(n^{1/3})$ but $\mathrm{Adv}(f) = O(\log n)$.

There also exist total functions with $\mathrm{Q}(g') \geq \widetilde{\deg}(g')^{4-o(1)}$ [2]. Composing this function with $\textsc{Ind}_k$ on the first k bits with $k = \Omega(\log \mathrm{Q}(f))$ yields a function g with the desired separation, since approximate polynomial degree also composes in the upper bound direction [31]. ◀

7 Lifting theorems

7.1 Background

Lifting theorems are results that relate communication complexity measures to query complexity measures. For a fixed query measure, such as $\mathrm{D}(f)$, and a communication complexity measure that intuitively corresponds to it, such as deterministic communication complexity

$\mathrm{D}^{\mathrm{cc}}(F)$, we may hope to be able to prove a theorem of the form: there exists some communication gadget G such that $\mathrm{D}^{\mathrm{cc}}(f \circ G) = \widetilde{\Theta}(\mathrm{D}(f))$. In fact, when the size of the gadget G is allowed to depend on the input size of f and the $\widetilde{\Theta}$ is allowed to hide $\operatorname{polylog} n$ factors, such a result is known [19].

We remark that the upper bound direction – showing the communication measure of $f \circ G$ is at most the corresponding query measure of f – is usually easy. We can simulate the query algorithm in the communication complexity world, losing only a multiplicative factor that depends on the difficulty of computing G. The lower bound direction, which lower bounds a communication complexity measure by a query complexity measure, is usually much harder, and is what we will usually refer to when we use the term "lifting theorem."

The result of [19] gives a lifting theorem for deterministic protocols, which we will denote by $\mathrm{D} \to \mathrm{D}^{\mathrm{cc}}$ to mean it transfers a lower bound on the first measure to a lower bound on the second. Recently, lifting theorems have been shown for R and R_0 (with the corresponding communication complexity measures being the obvious ones: randomized communication with bounded error and randomized communication with zero error, denoted R^{cc} and $\mathrm{R}_0^{\mathrm{cc}}$) [20]. We do not know how to lift Q or Q_0 to their analogous communication measures; this is likely to be significantly harder.

7.2 Lifting theorem reductions

In this section, we prove several lifting theorem reductions, showing that a lifting theorem for one measure (such as Q_0) implies a lifting theorem for another measure (such as Q). Our work (including prior work [11]) is the first instance we know of where such reductions are shown; it is perhaps surprising that these reductions can be proven without proving the lifting theorems themselves.

▶ **Theorem 18.** *If there is a lifting theorem for* Q_0 *with gadget* G*, then there is also a lifting theorem for* Q *with the same gadget* G*.*

Proof. Fix a partial function f. We wish to show that $\mathrm{Q}^{\mathrm{cc}}(f \circ G) = \widetilde{\Omega}(\mathrm{Q}(f))$ using a lifting theorem for Q_0.

Let $g = \textsc{UInd}_k \circ_k f$, with $k = \Theta(\mathrm{Q}(f))$. By Theorem 17, we have $\mathrm{Q}(g) = \widetilde{\Omega}(\mathrm{Q}(f))$. Next, apply the lifting theorem to g to get

$$\mathrm{Q}_0^{\mathrm{cc}}(g \circ G) = \widetilde{\Omega}(\mathrm{Q}_0(g)) = \widetilde{\Omega}(\mathrm{Q}(g)) = \widetilde{\Omega}(\mathrm{Q}(f)). \tag{13}$$

To complete the argument, it remains to show that $\mathrm{Q}_0^{\mathrm{cc}}(g \circ G)) = \widetilde{O}(\mathrm{Q}^{\mathrm{cc}}(f \circ G))$. Note that $g \circ G = \textsc{UInd}_k \circ_k f \circ G$. If we have a communication protocol for $f \circ G$, we can simulate it k times (and use error reduction) to obtain the correct index with constant error. We can then use the promise of UInd to check if the index is correct, by verifying that the second bit of the pair at that index is 1. This turns the algorithm into a zero-error algorithm. Since $k = O(\log \mathrm{Q}(f))$, our algorithm uses only $\widetilde{O}(\mathrm{Q}^{\mathrm{cc}}(f \circ G))$ communication. Thus $\mathrm{Q}^{\mathrm{cc}}(f \circ G) = \widetilde{\Omega}(\mathrm{Q}(f))$, as desired. ◀

▶ **Theorem 19.** *If there is a lifting theorem for* QSZK *with gadget* G*, then there is also a lifting theorem for* Q *with the same gadget* G*.*

By a lifting theorem for QSZK, we mean a theorem that lifts it to some communication complexity analogue $\mathrm{QSZK}^{\mathrm{cc}}$. The only property we use of $\mathrm{QSZK}^{\mathrm{cc}}$ is that it lower bounds Q^{cc}.

Proof. Let f be a partial function. Let $g = \textsc{Ind}_k \circ_k f$, where $k = \Theta(\log \mathrm{Q}(f))$. By Theorem 17, $\mathrm{QD}(g) = \widetilde{\Omega}(\mathrm{Q}(f))$. By Theorem 2, $\mathrm{QSZK}(g) = \Omega(\mathrm{QD}(g)) = \widetilde{\Omega}(\mathrm{Q}(f))$. Then

$$\mathrm{Q}^{\mathrm{cc}}(g \circ G) = \Omega(\mathrm{QSZK}^{\mathrm{cc}}(g \circ G)) = \widetilde{\Omega}(\mathrm{QSZK}(g)) = \widetilde{\Omega}(\mathrm{Q}(f)). \tag{14}$$

Also, note that if we had a quantum communication protocol for $f \circ G$ we could easily convert it to a communication protocol for $g \circ G = \textsc{Ind}_k \circ_k f \circ G$. Thus $\mathrm{Q}^{\mathrm{cc}}(f \circ G) = \widetilde{\Omega}(\mathrm{Q}^{\mathrm{cc}}(g \circ G) = \widetilde{\Omega}(\mathrm{Q}(f))$, as desired. ◀

▶ **Theorem 20.** *If there is a lifting theorem that lifts* $\mathrm{QD} \to \mathrm{Q}^{\mathrm{cc}}$ *with gadget* G*, then there is also a lifting theorem for* Q *with the same gadget* G*.*

By a lifting theorem for $\mathrm{QD} \to \mathrm{Q}^{\mathrm{cc}}$, we mean a theorem that shows $\mathrm{Q}^{\mathrm{cc}}(f \circ G) = \widetilde{\Omega}(\mathrm{QD}(f))$ for all partial functions f. This is formally easier to prove than a $\widetilde{\Omega}(\mathrm{Q}(f))$ lower bound, but we show it is actually equivalent.

Proof. Let f be a partial function. Let $g = \textsc{Ind}_k \circ_k f$, where $k = \Theta(\log \mathrm{Q}(f))$. By Theorem 17, $\mathrm{QD}(g) = \widetilde{\Omega}(\mathrm{Q}(f))$. Then

$$\mathrm{Q}^{\mathrm{cc}}(g \circ G) = \widetilde{\Omega}(\mathrm{QD}(g)) = \widetilde{\Omega}(\mathrm{Q}(f)). \tag{15}$$

Also, note that if we had a quantum communication protocol for $f \circ G$ we could easily convert it to a communication protocol for $g \circ G = \textsc{Ind}_k \circ_k f \circ G$. Thus $\mathrm{Q}^{\mathrm{cc}}(f \circ G) = \widetilde{\Omega}(\mathrm{Q}^{\mathrm{cc}}(g \circ G) = \widetilde{\Omega}(\mathrm{Q}(f))$, as desired. ◀

In summary, what we have shown is that a lifting theorem for Q is implied by a lifting theorem for either Q_0, QSZK, or a $\mathrm{QD} \to \mathrm{Q}^{\mathrm{cc}}$ lifting theorem. In fact, each of these statements also has a classical analogue which remains true. Proving a lifting theorem for R_0, SZK, or $\mathrm{RS} \to \mathrm{R}^{\mathrm{cc}}$ would imply a lifting theorem for R. This can be proved analogously; the only property we need is that $\mathrm{RS}(\textsc{UInd}_k \circ_k f) = \widetilde{\Omega}(\mathrm{R}(f))$ when k is at least polylogarithmic in $\mathrm{R}(f)$. An equivalent statement to this was proven in [11]. However, since lifting theorems for R and R_0 are already known (with an index gadget [20]), this reduction is less interesting in the classical case, though it might still be relevant for proving lifting theorems with other gadgets.

References

1. Scott Aaronson. Quantum certificate complexity. *Journal of Computer and System Sciences*, 74(3):313–322, 2008. doi:10.1016/j.jcss.2007.06.020.
2. Scott Aaronson, Shalev Ben-David, and Robin Kothari. Separations in query complexity using cheat sheets. In *Proceedings of the 48th Symposium on Theory of Computing (STOC 2016)*, pages 863–876, 2016. doi:10.1145/2897518.2897644.
3. Scott Aaronson and Yaoyun Shi. Quantum Lower Bounds for the Collision and the Element Distinctness Problems. *Journal of the ACM*, 51(4):595–605, 2004. doi:10.1145/1008731.1008735.
4. Andris Ambainis. Quantum Lower Bounds by Quantum Arguments. *Journal of Computer and System Sciences*, 64(4):750–767, June 2002. doi:10.1006/jcss.2002.1826.
5. Andris Ambainis. Polynomial Degree vs. Quantum Query Complexity. In *Proceedings of the 54th Symposium on Foundations of Computer Science (FOCS 2003)*, page 230, 2003. doi:10.1109/SFCS.2003.1238197.
6. Andris Ambainis, Kaspars Balodis, Aleksandrs Belovs, Troy Lee, Miklos Santha, and Juris Smotrovs. Separations in Query Complexity Based on Pointer Functions. In *Proceedings of the 48th Symposium on Theory of Computing*, STOC '16, pages 800–813, 2016. doi:10.1145/2897518.2897524.

7 Howard Barnum, Michael Saks, and Mario Szegedy. Quantum Decision Trees and Semidefinite Programming. Technical report, Los Alamos National Laboratory, 2001. URL: http://permalink.lanl.gov/object/view?what=info:lanl-repo/lareport/LA-UR-01-6417.

8 Howard Barnum, Michael Saks, and Mario Szegedy. Quantum query complexity and semidefinite programming. In *18th Conference on Computational Complexity (CCC 2003)*, pages 179–193, 2003. doi:10.1109/CCC.2003.1214419.

9 Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. *Journal of the ACM*, 48(4):778–797, 2001. doi:10.1145/502090.502097.

10 Aleksandrs Belovs and Robert Spalek. Adversary lower bound for the k-sum problem. In *Proceedings of the 4th conference on Innovations in Theoretical Computer Science*, pages 323–328, 2013. doi:10.1145/2422436.2422474.

11 Shalev Ben-David and Robin Kothari. Randomized Query Complexity of Sabotaged and Composed Functions. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, volume 55 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 60:1–60:14, 2016. doi:10.4230/LIPIcs.ICALP.2016.60.

12 Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and Weaknesses of Quantum Computing. *SIAM Journal on Computing (special issue on quantum computing)*, 26:1510–1523, 1997. doi:10.1137/S0097539796300933.

13 Adam Bouland, Lijie Chen, Dhiraj Holden, Justin Thaler, and Prashant Nalini Vasudevan. On the Power of Statistical Zero Knowledge. In *58th Annual Symposium on Foundations of Computer Science (FOCS 2017)*, pages 708–719, October 2017. doi:10.1109/FOCS.2017.71.

14 Harry Buhrman, Richard Cleve, Ronald de Wolf, and Christof Zalka. Bounds for Small-Error and Zero-Error Quantum Algorithms. In *Proceedings of the 40th Symposium on Foundations of Computer Science*, FOCS '99, pages 358–368, 1999. doi:10.1109/SFFCS.1999.814607.

15 Harry Buhrman and Ronald de Wolf. Complexity measures and decision tree complexity: a survey. *Theoretical Computer Science*, 288(1):21–43, 2002. doi:10.1016/S0304-3975(01)00144-X.

16 Lijie Chen. A note on oracle separations for BQP. *arXiv preprint*, 2016. arXiv:1605.00619.

17 Oded Goldreich, Amit Sahai, and Salil Vadhan. Honest-verifier Statistical Zero-knowledge Equals General Statistical Zero-knowledge. In *Proceedings of the 30th Symposium on Theory of Computing*, STOC '98, pages 399–408, 1998. doi:10.1145/276698.276852.

18 Mika Göös, Shachar Lovett, Raghu Meka, Thomas Watson, and David Zuckerman. Rectangles Are Nonnegative Juntas. *SIAM Journal on Computing*, 45(5):1835–1869, 2016. doi:10.1137/15M103145X.

19 Mika Göös, Toniann Pitassi, and Thomas Watson. Deterministic Communication vs. Partition Number. In *Proceedings of the 56th Symposium on Foundations of Computer Science (FOCS 2015)*, pages 1077–1088, 2015. doi:10.1109/FOCS.2015.70.

20 Mika Göös, Toniann Pitassi, and Thomas Watson. Query-to-communication lifting for BPP. In *58th Annual Symposium on Foundations of Computer Science (FOCS 2017)*, pages 132–143, October 2017. doi:10.1109/FOCS.2017.21.

21 Peter Høyer, Troy Lee, and Robert Špalek. Negative weights make adversaries stronger. In *Proceedings of the 39th Symposium on Theory of Computing (STOC 2007)*, pages 526–535, 2007. doi:10.1145/1250790.1250867.

22 Raghav Kulkarni and Avishay Tal. On Fractional Block Sensitivity. *Chicago Journal of Theoretical Computer Science*, 2016(8), July 2016. doi:10.4086/cjtcs.2016.008.

23 Sophie Laplante and Frédéric Magniez. Lower bounds for randomized and quantum query complexity using Kolmogorov arguments. In *Proceedings of the 19th Conference on Computational Complexity*, pages 294–304, June 2004. doi:10.1109/CCC.2004.1313852.

24 Troy Lee, Rajat Mittal, Ben W. Reichardt, Robert Špalek, and Mario Szegedy. Quantum query complexity of state conversion. In *Proceedings of the 52nd Symposium on Foundations of Computer Science (FOCS 2011)*, pages 344–353, 2011. doi:10.1109/FOCS.2011.75.

25 Troy Lee and Jérémie Roland. A strong direct product theorem for quantum query complexity. *Computational Complexity*, 22(2):429–462, 2013. doi:10.1007/s00037-013-0066-8.

26 Sanketh Menda and John Watrous. Oracle Separations for Quantum Statistical Zero-Knowledge. *arXiv preprint*, 2018. arXiv:1801.08967.

27 Gatis Midrijanis. On randomized and quantum query complexities. *arXiv preprint*, 2005. arXiv:quant-ph/0501142.

28 Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.

29 Ran Raz and Pierre McKenzie. Separation of the Monotone NC Hierarchy. *Combinatorica*, 19(3):403–435, 1999. doi:10.1007/s004930050062.

30 Amit Sahai and Salil Vadhan. A Complete Problem for Statistical Zero Knowledge. *Journal of the ACM*, 50(2):196–249, March 2003. doi:10.1145/636865.636868.

31 Alexander A. Sherstov. Making Polynomials Robust to Noise. In *Proceedings of the 44th Symposium on Theory of Computing*, STOC '12, pages 747–758, 2012. doi:10.1145/2213977.2214044.

32 Robert Špalek and Mario Szegedy. All Quantum Adversary Methods are Equivalent. *Theory of Computing*, 2(1):1–18, 2006. doi:10.4086/toc.2006.v002a001.

33 List of Open Problems in Sublinear Algorithms. Problem 77: Frontiers in Structural Communication Complexity. https://sublinear.info/index.php?title=Open_Problems:77, 2017.

34 John Watrous. Limits on the Power of Quantum Statistical Zero-Knowledge. In *Proceedings of the 43rd Symposium on Foundations of Computer Science*, FOCS '02, pages 459–468, 2002. Full version available at https://cs.uwaterloo.ca/~watrous/Papers/. doi:10.1109/SFCS.2002.1181970.

35 John Watrous. Zero-Knowledge against Quantum Attacks. *SIAM Journal on Computing*, 39(1):25–58, 2009. doi:10.1137/060670997.

36 John Watrous. *The Theory of Quantum Information*. Cambridge University Press, 2018. Available at https://cs.uwaterloo.ca/~watrous/TQI/.

37 Shengyu Zhang. On the power of Ambainis lower bounds. *Theoretical Computer Science*, 339(2):241–256, 2005. doi:10.1016/j.tcs.2005.01.019.

A Proof of the hybrid argument

In this section, we prove Lemma 12, restated below for convenience.

▶ **Lemma 12** (Hybrid Argument). *Let $x \in \{0,1\}^n$ be an input, and let $B \subseteq [n]$ be a block. Let Q be a T-query quantum algorithm that accepts x and rejects x^B with high probability, or more generally produces output states that are a constant distance apart in trace distance for x and x^B. Let m_i^t be the probability that, when Q is run on x for t queries and then subsequently measured, it is found to be querying position i of x (i.e., the query register collapses to $|i\rangle$). Then*

$$\sum_{t=1}^{T} \sum_{i \in B} m_i^t = \Omega\left(\frac{1}{T}\right). \tag{7}$$

Proof. We start by fixing some notation. Let the quantum query algorithm Q act on m qubits, initialized in the all-zeros state $|0^m\rangle$. A T-query algorithm is specified by $T+1$ unitaries $U_0, U_1, \ldots, U_T$ acting on m qubits. For any input $x \in \{0,1\}^n$, the oracle O_x acts as $O_x|i,b\rangle = |i, b \oplus x_i\rangle$ for all $i \in [n]$ and $b \in \{0,1\}$. The output state produced by this quantum algorithm (before measurement) on input x is

$$|\psi_x\rangle = U_T O_x U_{T-1} O_x \cdots O_x U_1 O_x U_0 |0^m\rangle, \tag{16}$$

where O_x is implicitly $O_x \otimes \mathbb{1}$ if O_x acts on fewer than m qubits. Within the m qubits, we further group the qubits into three registers, the first register holds an index $|i\rangle$, for $i \in [n]$, the second holds a qubit $|b\rangle$, for $b \in \{0,1\}$, and the third register contains all the remaining qubits.

For a quantum algorithm outputting a Boolean function, we assume that the first qubit of $|\psi_x\rangle$ is measured at the end to determine the output. A quantum distinguishing algorithm may trace out some qubits of $|\psi_x\rangle$ before producing an output or it may simply output the state $|\psi_x\rangle$ without loss of generality, since tracing out qubits cannot increase the distance between a pair of states.

In our case we have an algorithm Q that accepts x and rejects x^B with high probability. To be more concrete, let us assume Q has error probability ϵ. As we saw in Proposition 6, such an algorithm can be made to output a mixed state ρ_x such that $\|\rho_x - \rho_{x^B}\|_{\mathrm{tr}} \geq 1 - 2\epsilon$. Since trace distance is non-increasing under partial trace [28, Th. 9.2], we get that the pure output states must also be far, and hence $\||\psi_x\rangle\langle\psi_x| - |\psi_{x^B}\rangle\langle\psi_{x^B}|\|_{\mathrm{tr}} \geq 1 - 2\epsilon$. This is all we need to assume about the output of the algorithm on these inputs.

We now consider the intermediate states produced by this quantum algorithm after t queries to input x. Let

$$|\psi_x^0\rangle := U_0|0^m\rangle \quad \text{and} \quad |\psi_x^t\rangle := U_t O_x |\psi_x^{t-1}\rangle. \tag{17}$$

for $t \in [T]$. The final state of the algorithm is $|\psi_x^T\rangle = |\psi_x\rangle$, and hence we have

$$\left\||\psi_x^T\rangle\langle\psi_x^T| - |\psi_{x^B}^T\rangle\langle\psi_{x^B}^T|\right\|_{\mathrm{tr}} \geq 1 - 2\epsilon. \tag{18}$$

We know that the states are far apart in trace distance, but we also want to bound their closeness in ℓ_2 distance. By (3), we have

$$|\langle\psi_x^T|\psi_{x^B}^T\rangle| \leq \sqrt{1-(1-2\epsilon)^2} = 2\sqrt{\epsilon(1-\epsilon)} \leq 1 - (1/2)(1-2\epsilon)^2. \tag{19}$$

Then we have

$$\begin{aligned} \||\psi_x^T\rangle - |\psi_{x^B}^T\rangle\|^2 &= 2 - \langle\psi_{x^B}^T|\psi_x^T\rangle - \langle\psi_x^T|\psi_{x^B}^T\rangle \\ &= 2 - 2\,\mathrm{Re}(\langle\psi_{x^B}^T|\psi_x^T\rangle) \geq 2 - 2|\langle\psi_{x^B}^T|\psi_x^T\rangle| \geq (1-2\epsilon)^2, \end{aligned} \tag{20}$$

and so $\||\psi_x^T\rangle - |\psi_{x^B}^T\rangle\| \geq 1 - 2\epsilon$.

Hence the final states of the algorithm are far in apart in ℓ_2 distance on inputs x and x^B. We also know that the initial states $|\psi_x^0\rangle$ and $|\psi_{x^B}^0\rangle$ are identical. We keep track of how much this distance $d_t := \||\psi_x^t\rangle - |\psi_{x^B}^t\rangle\|$ changes for $t \in \{0,1,\ldots,T\}$. For each t, we have

$$d_{t+1} = \||\psi_x^{t+1}\rangle - |\psi_{x^B}^{t+1}\rangle\| = \|U_{t+1}O_x|\psi_x^t\rangle - U_{t+1}O_{x^B}|\psi_{x^B}^t\rangle\| = \|O_x|\psi_x^t\rangle - O_{x^B}|\psi_{x^B}^t\rangle\|, \tag{21}$$

since U_{t+1} is a unitary and preserves norms. This equals

$$\|O_{x^B}|\psi_x^t\rangle - O_{x^B}|\psi_{x^B}^t\rangle + (O_x - O_{x^B})|\psi_x^t\rangle\| \leq \|O_{x^B}|\psi_x^t\rangle - O_{x^B}|\psi_{x^B}^t\rangle\| + \|(O_x - O_{x^B})|\psi_x^t\rangle\| \tag{22}$$

$$= d_t + \|(O_x - O_{x^B})|\psi_x^t\rangle\|. \tag{23}$$

Next, decompose $|\psi_x^t\rangle$ by the value of the query register. On basis vectors when the query register is not in B, the unitaries O_x and O_{x^B} behave the same; such vectors therefore get mapped to zero. If $|\psi_t^{x,B}\rangle$ denotes the component of $|\psi_x^t\rangle$ whose query register is in B, we get

$$\|(O_x - O_{x^B})|\psi_x^t\rangle\| = \|(O_x - O_{x^B})|\psi_t^{x,B}\rangle\| \leq \|O_x|\psi_t^{x,B}\rangle\| + \|O_{x^B}|\psi_t^{x,B}\rangle\| = 2\||\psi_t^{x,B}\rangle\| \tag{24}$$

$$= 2 \cdot \sqrt{\sum_{i \in B} m_i^{t+1}}, \tag{25}$$

where the last equality follows from the definition of m_i^{t+1}, which is defined to be the probability that the algorithm is found to be querying position i right before making query $t+1$. The increase from d_t to d_{t+1} is therefore upper bounded by $2\sqrt{\sum_{i \in B} m_i^{t+1}}$, so we have

$$2 \sum_{t=1}^{T} \sqrt{\sum_{i \in B} m_i^t} \geq d_T - d_0 \geq 1 - 2\epsilon. \tag{26}$$

Using the Cauchy–Schwarz inequality on the outer sum gives

$$2\sqrt{T} \sqrt{\sum_{t=1}^{T} \sum_{i \in B} m_i^t} \geq 1 - 2\epsilon, \tag{27}$$

or

$$\sum_{t=1}^{T} \sum_{i \in B} m_i^t \geq \frac{(1-2\epsilon)^2}{4T} = \Omega\left(\frac{1}{T}\right), \tag{28}$$

when ϵ is a constant.[3] ◀

B QSZK closed under complement

Sketch of proof of $\mathrm{QSZK}(f) = \Theta(\mathrm{QSZK}(\neg f))$. To prove this, we would like to reduce the complete problem to its complement. In other words, we are given two circuits that query an oracle preparing ρ_x and σ_x that are either far apart in trace distance (when $f(x) = 1$) or close in trace distance (when $f(x) = 0$). From these circuits, we want to define two new states ρ'_x and σ'_x, such that these states are far when ρ_x and σ_x were close, and close when ρ_x and σ_x were far. Before starting the transformation, we first boost the parameters $2/3$ and $1/3$ to be extremely close to 1 and 0 respectively. For this sketch we will assume the parameters are exactly 1 and 0, which means when the states are far, they are perfectly distinguishable (i.e., $\|\rho_x - \sigma_x\|_{\mathrm{tr}} = 1$), and when they are close, they are equal (i.e., $\rho_x = \sigma_x$).

To perform this transformation, consider the pure states output by the circuits before tracing out any qubits. Let $|R_x\rangle_{BC}$ and $|S_x\rangle_{BC}$ be the pure state on registers B and C, which yields ρ_x and σ_x, respectively when register B is traced out. More formally, we have

$$\rho_x = \mathrm{Tr}_B(|R_x\rangle\langle R_x|_{BC}) \text{ and } \sigma_x = \mathrm{Tr}_B(|S_x\rangle\langle S_x|_{BC}). \tag{29}$$

From the pure states $|R_x\rangle_{BC}$ and $|S_x\rangle_{BC}$, we define two new pure states on registers A, B, C, and D, as follows:

$$|R'_x\rangle = \frac{1}{\sqrt{2}}\Big(|0\rangle_A |R_x\rangle_{BC} |0\rangle_D + |1\rangle_A |S_x\rangle_{BC} |0\rangle_D\Big) \text{ and} \tag{30}$$

$$|S'_x\rangle = \frac{1}{\sqrt{2}}\Big(|0\rangle_A |R_x\rangle_{BC} |0\rangle_D + |1\rangle_A |S_x\rangle_{BC} |1\rangle_D\Big). \tag{31}$$

[3] This can be slightly improved to $(1 - 2\sqrt{\epsilon(1-\epsilon)})/2T$ by not using the approximation in (19).

Note that the only difference between these states is in register D. If we have circuits preparing states $|R_x\rangle_{BC}$ and $|S_x\rangle_{BC}$, it is easy to see that we can construct circuits preparing $|R'_x\rangle_{ABCD}$ and $|S'_x\rangle_{ABCD}$. We now define the states ρ'_x and σ'_x from these states by tracing out registers C and D:

$$\rho'_x = \mathrm{Tr}_{CD}(|R'_x\rangle\langle R'_x|_{ABCD}) \text{ and } \sigma'_x = \mathrm{Tr}_{CD}(|S'_x\rangle\langle S'_x|_{ABCD}). \tag{32}$$

We claim that these states satisfy the conditions we require. When $f(x) = 1$, we have that $\|\rho_x - \sigma_x\|_{\mathrm{tr}} = 1$, i.e., the residual state on register C for states $|R'_x\rangle$ and $|S'_x\rangle$ is completely distinguishable. In this case, before we trace out registers C and D, we could implement a unitary on these registers which reads register C and writes onto register D whether the state in C is ρ_x or σ_x. This operation maps the state $|R'_x\rangle$ to the state $|S'_x\rangle$ and only acts on the traced out qubits, which does not affect the qubits that are not traced out, and we have $\rho'_x = \sigma'_x$.

When $f(x) = 0$, we have that $\rho_x = \sigma_x$. In this case we want to show that ρ'_x and σ'_x are distinguishable. We will show that after applying a specific unitary to these states are tracing out register B, in the first case we are left with the state $|+\rangle\langle+|_A$, but in the second case we have $\frac{1}{2}\mathbb{1}_A$, which can be distinguished.

Since $\rho_x = \sigma_x$, there is a unitary U_B such that $(U_B \otimes \mathbb{1}_C)|R_x\rangle_{BC} = |S_x\rangle_{BC}$. Controlled on the qubit in register A, let us apply the unitary U_B to register B of $|R'_x\rangle$ and $|S'_x\rangle$ before we trace out registers C and D, which is equivalent to applying it after tracing out the registers. This makes registers BC unentangled with the rest of the state, and equal to $|S_x\rangle_{BC}$. In the first case we are left with the state $|+\rangle_A|0\rangle_D$ on registers A and D, while in the second case we have $\frac{1}{2}(|00\rangle_{AD} + |11\rangle_{AD})$. Tracing out register D leaves us with the $|+\rangle$ state in the first case and the maximally mixed state in the second case, as claimed. ◁

C Proof of Theorem 17

Proof. Recall that quantum query complexity composes perfectly [24], so $\mathrm{Q}(\textsc{Ind}_k \circ f) = \Theta(\mathrm{Q}(\textsc{Ind}_k)\,\mathrm{Q}(f)) = O(k\,\mathrm{Q}(f))$. We argue that $\mathrm{Q}(\textsc{Ind}_k \circ_k f)$ is smaller than $\mathrm{Q}(\textsc{Ind}_k \circ f)$. This is because we can convert any algorithm for $\mathrm{Q}(\textsc{Ind}_k \circ f)$ into an algorithm for $\mathrm{Q}(\textsc{Ind}_k \circ_k f)$: fix a 0-input x^0 and a 1-input x^1 for f; then, given an input to $\mathrm{Q}(\textsc{Ind}_k \circ_k f)$, pretend that each 0 bit in the second half of the input is actually x^0, and that each 1 bit is actually x^1 (the algorithm can do this by applying the appropriate unitary). This converts the input into an input for $\mathrm{Q}(\textsc{Ind}_k \circ f)$, completing the reduction.

Thus $\mathrm{Q}(\textsc{Ind}_k \circ_k f) = O(k\,\mathrm{Q}(f))$. Similarly, $\mathrm{Q}(\textsc{UInd}_k \circ_k f) = O(k\,\mathrm{Q}(f))$. Since QD is smaller than Q, it remains only to show that $\mathrm{QD}(\textsc{Ind}_k \circ_k f) = \Omega(k\,\mathrm{Q}(f))$ and $\mathrm{QD}(\textsc{UInd}_k \circ_k f) = \Omega(k\,\mathrm{Q}(f))$. We complete the argument for UIND; the argument for IND is similar.

Let Q be an optimal quantum distinguishing algorithm for $\textsc{UInd}_k \circ_k f$. We turn Q into a quantum algorithm Q' that uses the same number of queries, and solves all k copies of f with non-negligible probability; we then apply the direct product theorem (Theorem 16) to lower bound the number of queries required by Q', and hence by Q.

Given k inputs to f, the first thing the algorithm Q' does is append an all-0 array to turn it into an input to $\textsc{UInd}_k \circ_k f$. (Since the array is all zeros, the new input does not satisfy the promise of $\textsc{UInd}_k \circ_k f$, but we will still be able to run Q on it.) Then Q' picks a random number t between 1 and T uniformly, where $T = \mathrm{QD}(\textsc{UInd}_k \circ_k f)$ is the number of queries used by Q, and simulates Q for t queries. The algorithm Q' then measures the state of Q to determine the position at which Q was going to query. If this position is in the array part of the input and is inside a pair that has index $i \in \{0,1\}^k$, the algorithm Q' will then output the string i.

Consider the correct pair in the array (the one really pointed to by the k copies of f). Flipping the pair from 00 to 01 causes the input to satisfy the promise of $\text{UIND}_k \circ_k f$, and causes the output to become a 0-input. On the other hand, flipping the pair from 00 to 11 causes the input to become a 1-input. Let $|\psi\rangle$ be the final state of Q when run on the original, illegal input. Let $|\psi_0\rangle$ be the final state of Q when run on the flipped 0-input, and let $|\psi_1\rangle$ be the final state of Q when run on the 1-input. We know that $|\psi_0\rangle$ and $|\psi_1\rangle$ are far in trace distance. Hence $|\psi\rangle$ must be a far in trace distance from at least one on them.

Thus by Lemma 12, the probability that Q' finds Q querying inside the correct pair of the array is $\Omega(1/T^2)$. This means that Q' outputs the correct string of answers to the k inputs to f is with probability at least $\Omega(1/T^2)$. Since Q' uses only T queries, by Theorem 16 we must have either $T = \Omega(k\,\mathrm{Q}(f))$ or $1/T^2 = O((5/6)^k)$. The latter implies $T = \Omega((6/5)^{k/2}) = \Omega((6/5)^{k/4} \cdot (6/5)^{k/4}) = 2^{\Omega(k)} \cdot 2^{\Omega(k)}$. When $k \geq c \log \mathrm{Q}(f)$ for a large enough constant c, this gives $T \geq 2^{\Omega(k)}\,\mathrm{Q}(f) = \Omega(k\,\mathrm{Q}(f))$. Recalling that $T = \mathrm{QD}(\text{UIND}_k \circ_k f)$, we get $\mathrm{QD}(\text{IND}_k \circ_k f) = \Omega(k\,\mathrm{Q}(f))$, as desired. ◀