Non-Interactive Classical Verification of Quantum Depth: A Fine-Grained Characterization

Nai-Hui Chia[1] and Shih-Han Hung[2]

[1]Department of Computer Science, Rice University
[2]Department of Computer Science, The University of Texas at Austin

Abstract

We introduce protocols for classical verification of quantum depth (CVQD). These protocols enable a classical verifier to differentiate between devices of varying quantum circuit depths, even in the presence of classical computation. The goal is to demonstrate that a classical verifier can reject a device with a quantum circuit depth of no more than d, even if the prover employs additional polynomial-time classical computation to deceive. Conversely, the verifier accepts a device with a quantum circuit depth of $d' > d$.

Previous results for separating hybrid quantum-classical computers with various quantum depths require either quantum access to oracles or interactions between the classical verifier and the quantum prover. However, instantiating oracle separations can significantly increase the quantum depth in general, and interaction challenges the quantum device to keep the qubits coherent while waiting for the verifier's messages. These requirements pose barriers to implementing the protocols on near-term devices.

In this work, we present a two-message protocol under the quantum hardness of learning with errors and the random oracle heuristic. An honest prover only needs classical access to the random oracle, and therefore any instantiation of the oracle does not increase the quantum depth. To our knowledge, our protocol is the first non-interactive CVQD, the instantiation of which using concrete hash functions, e.g., SHA-3, does not require additional quantum depth.

Our second protocol seeks to explore the minimality of cryptographic assumptions and the tightness of the separations. To accomplish this, we introduce an untrusted quantum machine that shares entanglements with the target machine. Utilizing a robust self-test, our protocol certifies the depth of the target machine with information-theoretic security and nearly optimal separation.

Contents

1 Introduction

Quantum circuit depth is a crucial factor when assessing the capabilities of near-term quantum devices. While quantum computers with many qubits have been recently implemented by companies such as IBM, Google, IonQ, and Rigetti [26, 27, 11, 31], their limited quantum circuit depth poses a challenge to demonstrate quantum advantages due to the presence of noisy gates and short coherence times. Hence, finding ways to harness the power of these small-depth quantum devices is not only a practical challenge but also an intriguing problem in quantum complexity theory.

Indeed, Aaronson and Chen showed that small-depth quantum computers can demonstrate so-called "Quantum Supremacy" [1] on the random circuit sampling problem, which means that quantum computers can efficiently solve the problem that is intractable for classical machines. Google [5] reported the results of experiments on demonstrating quantum supremacy by using superconducting quantum computers.[1] In the near term, the coherence time seriously limits the usable lifespan of quantum states. Thus, information processing with a small and noisy quantum device has become a central topic in the field of quantum computing.

Hybrid quantum-classical computing, which combines classical computers with quantum devices, is a promising approach to leveraging the power of small-depth quantum circuits. This computational model has recently gained significant attention and has the potential to outperform classical machines on real-world problems such as molecular simulation [29] and optimization problems [21]. Notably, Cleve and Watrous [16] demonstrated that the quantum Fourier transform can be implemented with logarithmic quantum depth in this model. This result implies that quantum algorithms for Abelian hidden subgroup problems, including Shor's factoring algorithm, can also be implemented with logarithmic quantum circuit depth.

The results above suggest that quantum devices with circuit depths beyond certain thresholds can demonstrate quantum advantages. With the potential applications of small-depth quantum devices in mind, one may begin to wonder:

Can we verify whether a device has sufficient quantum depth to demonstrate quantum advantages?

An answer to the question is to find some problem, give an efficient algorithm that only requires small-depth quantum circuits, and prove that no algorithm using strictly smaller quantum depth achieves the same time complexity. For instance, the aforementioned results [1, 16] showed separations between small-depth quantum circuits and classical computers under plausible computational assumptions. In other words, assuming a problem is classically hard, a device that can solve it efficiently must have quantum power.

Another possible approach is designing cryptographic protocols that demonstrate the *quantumness* of a quantum device [8, 9, 25, 28]. In these protocols, the classical verifier sends the description of a cryptogrpahic hash function f and random coins to challenge the prover to answer information about f. It is guaranteed that only a prover which performs quantum computation will successfully answer these challenges with high probability. While these protocols seem to be satisfying proposals for demonstrating quantumness, there is a caveat: for a quantum prover to succeed, it is required to evaluate f coherently, and thus the implementation of f with quantum gates sets a lower bound on the resource requirement. To address the issue, in subsquent works, Hirahara and Le Gall [25] and Liu

[1]New classical algorithms are found for solving the problem in few days (by estimation) [30], which implies that random circuit sampling of the size in the experiment in [5] might not be classically intractable. However, even these new classical algorithms are slower than the quantum one (that solves the problem in 200 seconds); therefore, the experiments showed quantum advantages on the problem.

and Gheorghiu [28] independently showed that these protocols only requires a hybrid computation that only uses a constant-depth quantum circuit using different approaches.

It is important to note that the current state-of-the-art approaches focus on distinguishing quantum computers from classical computers and do not directly address separating quantum computers with different quantum resources. It remains an open question whether these protocols or problems cannot be efficiently solved by quantum devices with smaller quantum depth.

In this work, we provide a "fine-grained" solution to the question in the following scenario: Alice wants to verify if Bob's computer has a quantum circuit depth larger than d. However, Bob might cheat using additional classical machines. We aim to design protocols that allow Alice to detect such cheating, which we call Classical Verification of Quantum Depth (CVQD) protocols. Note that our approach focuses on distinguishing quantum computers with different quantum resources rather than simply distinguishing them from classical computers.

1.1 Main results

In this work, we give an affirmative answer to the question by presenting three CVQD protocols capable of distinguishing quantum circuits with different depths in the presence of polynomial-time classical computation. We start by giving the informal definition of CVQD.

Definition 1.1 ($\mathsf{CVQD}(d, d')$, informal)**.** *Let $d, d' \in \mathbb{N}$ and $d' > d$. Let P_A be a bounded-depth quantum circuit with classical polynomial-time computation. Let V be a classical verifier. A $\mathsf{CVQD}(d, d')$ protocol that separates quantum circuit depth d from d' satisfies the following properties:*

- **Completeness:** *There exists a prover P_A of quantum circuit depth at least d' such that $\langle V, P_A \rangle$ accepts with probability at least $2/3$.*
- **Soundness:** *For every prover P_A of quantum circuit depth at most d, $\langle V, P_A \rangle$ accepts with probability at most $1/3$.*

In Definition 1.1, the verifier accepts if P_A has a quantum circuit with depth at least d', and rejects any dishonest prover which might interleave its small-depth quantum circuit (depth at most d) with a polynomial-time classical algorithm.

Chia, Chung, and Lai [12] defined two hybrid approaches for interleaving a quantum machine with a classical one, namely the d-CQ and d-QC schemes. In the d-CQ scheme, a classical algorithm can query a d-depth quantum circuit a polynomial number of times. On the other hand, the d-QC scheme allows a d-depth quantum circuit to access polynomial-time classical algorithms after each layer of 1-depth circuit. Our goal is to design protocols that can effectively detect and prevent cheating provers from exploiting either scheme.

In this work, we present two CVQD protocols.

Theorem 1.2 (Informal)**.** *Let $d \in \mathbb{N}$.*

1. *For polynomially bounded function $d(\cdot)$, there exists a $d(n)$-round $\mathsf{CVQD}(d(n), d(n) + d_f)$ protocol under the QLWE assumption, where d_f is a fixed constant.*
2. *For any constants d and d_f, there exists a 2-message $\mathsf{CVQD}(d, d + d_f)$ protocol under the QLWE assumption and the random oracle heuristic. Especially, an honest prover only needs classical access to the random oracle, and one can instantiate the random oracle using the random oracle heuristic.*

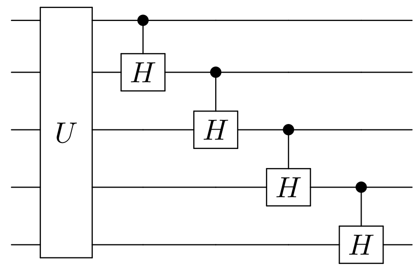

Figure 1: A quantum circuit of linear depth.

Here, the *QLWE* assumption assumes that the Learning With Error (LWE) problems are hard for any quantum polynomial-time algorithms.[2] The constant d_f is the quantum circuit depths for implementing particular functions that will be specified later.

What is quantum depth, and why do we care about it on near-term quantum devices? The depth of a quantum circuit is defined as the *length of the longest directed path from an input qubit to the output qubit*. Note that this is different from the number of gates operating on qubits. For example, consider a quantum circuit computing d controlled-Hadamard gates on adjacent qubits following a quantum circuit implementing U of depth d_f in Figure 1. The depth of the circuit is $d_f + d$ since the longest path in the circuit is $d_f + d$. One of the main reasons that near-term quantum devices can only implement small-depth quantum circuits is that the gates are noisy, and the noise will accumulate along the longest path. For instance, suppose that each gate in Figure 1 incurs a depolarizing noise of probability p. The probability that the last qubit is correct is roughly $(1-p)^{d_f+d}$.

What do CVQD protocols certify? The CVQD protocol aims to identify whether a "near-term quantum device" can implement a quantum circuit with a specific depth and ensure that a near-term quantum device with a strictly smaller quantum depth cannot cheat even in the presence of classical computation. Therefore, $\mathsf{CVQD}(d, d')$ focuses on being secure against $\mathsf{BPP}^{\mathsf{QNC_d}}$ and $\mathsf{QNC_d}^{\mathsf{BPP}}$ (see Section 2.1 for formal definitions). Basically, one can view the first model as a classical computer having access to a d-depth quantum circuit, and the second model is a d-depth quantum circuit allowing intermediate measurements and classical computation.

Arora, Coladangelo, Coudron, Gheorghiu, Singh, and Waldner considered a generalization $\mathsf{BPP}^{\mathsf{QNC}_d^{\mathsf{BPP}^{\mathsf{QNC}_d \cdots}}}$ of the multi-level hybrid schemes and claimed that CVQD should consider attacks from such powerful adversary [4, Section 1.3]. While we agree that considering class of multi-level hybrid schemes may be an interesting topic in computational complexity theory, we notice that this class includes extremely long computations. Thus, results about this class can be very detached from "practical" hybrid schemes. To see why, let us first consider the quantum circuit for computing the parity of n-bit strings. It is a well-known fact that computing the parity requires a quantum circuit of depth $\Theta(\log n)$ (for example, see Figure 2). Computing the parity can be computed using a $\mathsf{QNC}_2^{\mathsf{QNC}_2^{\mathsf{QNC}_2 \cdots}}$ scheme of $\log n$ layers:[3] For the base case, the parity of two bits can be computed

[2] In fact, it is sufficient to assume that QLWE is hard for a d-depth hybrid machine.

[3] Indeed, even an $\mathsf{NC}_2^{\mathsf{NC}_2^{\mathsf{NC}_2 \cdots}}$ circuit of $\log n$ levels can compute the parity of n-bit strings.

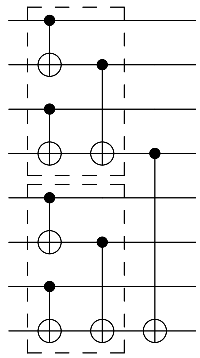

Figure 2: A quantum circuit for computing the parity, which may be viewed as a depth-2 circuit with access to an oracle (the dashed boxes) computing the parity of half-sized inputs.

using a CNOT gate. The parity of n-bit strings can be recursively computed by making two queries to a circuit for computing the parity of $(n/2)$-bit strings in parallel, followed by an application of a CNOT gate (see Figure 2). Since each query can be thought of as an application of a special gate simulated in the associated model, each inductive step only requires a depth-2 computation. By the same argument, an r-level $\mathsf{QNC}_d^{\mathsf{QNC}_d^{\mathsf{QNC}_d^{\cdots}}}$ computation can compute a recursively defined family of quantum circuits of depth $\Omega(d^r)$. More generally, an r-level $\mathsf{QNC}_d^{\mathsf{QNC}_d^{\mathsf{QNC}_d^{\cdots}}}$ computation is even allowed to keep qubits in coherence when making queries, and thus the depth of $\mathsf{QNC}_d^{\mathsf{QNC}_d^{\mathsf{QNC}_d^{\cdots}}}$ shall be viewed as d^r after instantiating the oracles instead of d. While we do not know if the circuit families can be compressed into short computations in general, it seems unlikely that the class defined by enumerating polynomially many levels is contained in BQP.

In terms of near-term quantum computation, the error propagation heavily depends on the position in the hierarchy. For example, in the rth level, suppose that a depolarizing noise of constant probability p is applied to each gate; the probability with no error is roughly $(1-p)^{d^r}$ in the worst case. More concretely, for $r > 1$ and $d = \Omega(\log n)$, the fraction of "good signal" is only negligibly small.

Instantiability of CVQD protocols. This paper aims to give instantiable protocols for quantum depth, i.e., when the oracles are implemented concretely, any scheme succeeding with sufficient quantum depth in the query model can also pass the verification in *any* instantiation. To our knowledge, prior to our results, we were not aware of any existing protocol that is also instantiable without blowups in depth. Previously, Chia, Chung, and Lai [12] and Coudron and Menda [18] first independently gave oracle separations between hybrid schemes with different quantum depths. Subsequently, Hasegawa and Le Gall [23] and Chia and Hung [14] independently gave shaper oracle separations based on the problem in [12] but in different query models. More recently, Arora, Coladangelo, Coudron, Gheorghiu, Singh, and Waldner gave a new separation in the quantum random oracle model [4]. All aforementioned results require the algorithm or the honest prover to have quantum access to the oracle. Therefore, replacing the oracle with an instantiation significantly increases the algorithm's quantum depth, resulting in a very loose separation (e.g., polynomial versus constant depth) of quantum depth between devices. In particular, suppose instantiating the oracle requires ℓ quantum depth, then the original d-depth quantum oracle circuit will become a $(d \cdot \ell)$-depth quantum circuit. This implies that an honest prover requires $d \cdot \ell$ quantum depth to convince the verifier. On the other hand, an honest prover in Theorem 1.2 only needs $d_f + d$ quantum depth, where d_f is a fixed constant to implement a specific function, While these results using quantum oracles justify that schemes with more quantum depth are more powerful from a complexity-theoretic perspective, whether they yield practical protocols for certifying quantum depth has remained unclear.

Construction of non-interactive CVQD protocols Our first protocol in Theorem 1.2 achieves $\mathsf{CVQD}(d(n), d(n) + d_f)$ for any polynomially bounded function $d(\cdot)$, where d_f is a fixed constant. However, one potential drawback is that the round complexity is linear in $d(n)$, which raises concerns about the prover's qubits potentially decohering during communication despite the communication being classical. To address this issue, we present a two-message protocol (the second result in Theorem 1.2) that significantly reduces the communication in the random oracle model such that an honest prover does not need to keep the qubits coherent during the communication. Additionally, an honest prover does not require quantum access to the random oracle, and thus, we can use the random oracle heuristic to instantiate the random oracle without increasing the quantum circuit depth of an honest prover. However, it is worth noting that the second protocol has a weaker separation compared to the first one and can only certify constant-depth quantum circuits.

The CVQD protocols presented in Theorem 1.2 require the QLWE assumption, and the second protocol also requires the random oracle heuristic. Furthermore, the tightness of the depth separation in both protocols is unknown. These observations motivate us to explore the following question:

Can we develop CVQD *protocols with optimal depth separation under weaker assumptions?*

Our findings suggest that it is possible to achieve nearly optimal depth separation unconditionally with the help of an additional untrusted prover. Specifically, we investigate protocols involving two provers who cannot communicate with each other but can share entanglements. In this setting, one prover (P_A) represents the target machine being tested, while the other prover (P_O) helps certify the quantum depth. However, neither of the provers can be trusted by the classical verifier.

Definition 1.3 ($\mathsf{CVQD}_2(d, d')$, informal). *Let $d, d' \in \mathbb{N}$ and $d' > d$. Let P_A be a bounded-depth quantum circuit with classical polynomial-time computation. Let P_O be an unbounded quantum prover and V be a classical verifier. A $\mathsf{CVQD}_2(d, d')$ protocol that separates quantum circuit depth d from d′ satisfies the following properties:*

- **Non-locality:** *P_O and P_A share arbitrarily many EPR pairs and are not allowed to communicate with each other once the protocol starts.*
- **Completeness:** *If P_A has quantum circuit depth at least d', then there exists P_O and P_A such that $\langle V, P_O, P_A \rangle$ accepts with probability at least $2/3$.*
- **Soundness:** *If P_A has quantum circuit depth at most d, then for any P_O and polynomial-time P_A, $\langle V, P_O, P_A \rangle$ accepts with probability at most $1/3$.*

We then prove the following theorem.

Theorem 1.4 (Informal). *Let $d \in \mathbb{N}$. There exists a two-prover $\mathsf{CVQD}_2(d, d+3)$ protocol $\langle V, P_A, P_O \rangle$ that is unconditionally secure with inefficient P_O and V. Moreover, an honest P_O and V can be efficient, assuming the existence of quantum-secure pseudorandom permutation (qPRP).*

To prove the result in Theorem 1.4, we provide a framework that transforms a quantum oracle separation into a two-prover protocol.

Theorem 1.5 (Informal)**.** *Let $\mathcal{C}$ and $\mathcal{C}'$ be two complexity classes. Let L^O be an oracle problem such that $L^O \in \mathcal{C}^O$ and $L^O \notin \mathcal{C}'^O$. Then, there exists a two-prover protocol $\langle V, P_A, P_O \rangle$ two real numbers $c, s \in [0,1]$ satisfying $c - s = 1/\operatorname{poly}(n)$ for size n of input such that the following conditions hold.*

- **Completeness:** *If P_A can solve problems in $\mathcal{C}$, then there exists P_O such that $\langle V, P_A, P_O \rangle$ accepts with probability at least c.*
- **Soundness:** *If P_A can only solve problems in $\mathcal{C}'$, then for any P_O, $\langle V, P_A, P_O \rangle$ accepts with probability at most s.*
- **Classical verification:** *V is classical, and the runtimes of V and the honest P_O depend on the number of queries for solving L^O and the complexity for implementing O.*

We start by transforming d-SSP, a quantum oracle problem introduced by Chia, Chung and Lai [12] for separating quantum depth, into a CVQD_2 protocol using the framework outlined in Theorem 1.5. However, this transformation alone does not yield the desired separation outlined in Theorem 1.4. To address this issue, we modify the original d-SSP to create the "in-place d-SSP," which achieves a separation of d versus $d+1$ and enables us to obtain the required separation in Theorem 1.4.

Theorem 1.6. *For any d, in-place d-$\mathsf{SSP}^* \in \mathsf{BPP}^{\mathsf{BQNC_{d+1}}} \bigcap {\mathsf{BQNC_{d+1}}}^{\mathsf{BPP}}$ but in-place d-SSP is not in $\mathsf{BPP}^{\mathsf{BQNC_d}} \bigcup {\mathsf{BQNC_d}}^{\mathsf{BPP}}$* [4]

1.2 Technical overview

1.2.1 d-round CVQD from QLWE

The second protocol relies on the assumption that the Learning-with-Errors (LWE) problem is hard for quantum computers (also called the QLWE assumption). In a breakthrough [8], Brakerski, Christiano, Mahadev, Vazirani and Vidick showed that the QLWE assumption implies the existence of a noisy trapdoor claw-free function (NTCF). A function f is trapdoor claw-free if it is 2-to-1, and given a pair (x, y) such that $f(x) = y$, it is computationally intractable to find the other preimage of y. Furthermore, the function f is also equipped with a strong property called the adaptive hardcore bit property. In a nutshell, the property states that no quantum adversary given access to a description of f can output (y, x, e) such that x is a preimage of y and $e \cdot (x_0 + x_1) = 0$ with probability non-negligibly better than $1/2$, where x_0, x_1 are the preimages of y. In contrast, there exist quantum processes which allow an efficient quantum device to output either (y, x) or (y, e).

This observation leads to a proof-of-quantumness protocol: the verifier on receiving y requests the prover to present a preimage x or an equation e. An efficient quantum prover can succeed with nearly perfect probability. For proving classical hardness, the idea is that one can rewind a classical prover which succeeds with probability non-negligibly more than $1/2$ to extract both x and e with non-negligibly probability: For every prover $\mathcal{A}$, let the state before receiving the challenge be a random variable σ_y. The adversary challenges $\mathcal{A}$ to use the same state σ_y to output both a preimage and an equation. Any prover $\mathcal{A}$ that wins the test with a non-negligible advantage would imply that the adversary breaks the property.

[4] Atsuya Hasegawa and Francois Le Gall also found an oracle problem with similar ideas to improve the separation. See related work.

In subsequent works, Hirahara and Le Gall [25] and Liu and Gheorghiu [28] showed that the same protocol only requires a quantum prover of constant depth. The ideas behind these constructions basically follow from presenting NTCFs that can be evaluated with constant quantum depth.

A proof-of-quantumness protocol can be viewed as a protocol which separates a prover of non-zero quantum depth from one of zero quantum depth (i.e., a classical device). It seems natural to rely on the same hardness assumption to separate a high-depth quantum device from a low-depth one with the following protocol:

1. The verifier samples the functions $f_1, \ldots, f_d$ and sends these functions to the prover.
2. The prover outputs $y_1, \ldots, y_d$.
3. For $i = 1 \ldots d$, the verifier sequentially samples a random bit c_i which indicates the request to send a preimage x_i or a equation e_i for y_i. The verifier rejects if, in any of the rounds, the prover fails.

In this protocol, the prover must increase its quantum depth by 1 in each round of Step 3, since the operation the prover performs depends on the challenge bit c_i, which depends on the previous message from the prover. It is straightforward to see a $(d_f + d)$-depth prover succeeds with nearly perfect probability, where d_f is the depth required for the evaluation of f. However, to show the hardness for any small-depth device, since the device is no longer purely classical, the same rewinding argument does not directly apply.

We formalize an observation that a $(d-1)$-depth prover cannot stay coherent throughout the protocol, and has to "reset" (i.e., to destroy all its coherence and to continue with a purely classical state) in an intermediate round j. Thus from round $i = j \ldots d$, the prover begins with an intermediate classical state σ, and responds with its quantum power. To break the adaptive hardcore bit property, the reduction simulates the protocol to compute the state σ, and rewinds on σ to compute both a preimage and an equation for f_d.

1.2.2 Two-message CVQD under QLWE and the random oracle heuristic

One potential way to obtain a two-message CVQD protocol is to use the quantum Fiat-Shamir transformation introduced by Don, Fehr, and Majenz [19]. This transformation can convert an $O(1)$-round public-coin interactive protocol into a non-interactive protocol in the random oracle model. Intuitively, we can apply this approach to the first CVQD protocol since it is a public-coin protocol.

There are a few challenges when it comes to applying the quantum Fiat-Shamir transformation to the current CVQD protocol. First, the protocol does not currently have negligible soundness, which is a requirement for the transformation to work. The standard approach to address this is to prove the parallel repetition theorem for the protocol, but it remains an open challenge to achieve negligible soundness for post-quantum multi-round public-coin protocols via parallel repetition (see, e.g., the discussions in [13, 2]).

Second, even if we can achieve negligible soundness, the quantum Fiat-Shamir transformation might not preserve quantum circuit depth. In particular, the reduction in [19] constructs a simulator that needs to implement a $q(n)$-wise independent hash to simulate the random oracle and an additional operation to reprogram the hash function. These operations have large quantum depths, and directly using the simulator in [19] would lead to a meaningless separation for quantum circuit depth. These two issues present a significant challenge to applying the quantum Fiat-Shamir transformation to our CVQD protocol.

Parallel repetition of CVQD. To address the first issue, we apply parallel repetition to the protocol in Section 1.2.1. More concretely, for super-logarithmic m, the verifier and the prover run an m-copy parallel repetition:

1. The verifier samples $d \cdot m$ functions $f_{1,1}, \ldots, f_{1,m}, \ldots, f_{d,1}, \ldots, f_{d,m}$, which are sent to the prover.

2. For each $i \in [d]$ and $j \in [m]$, the prover samples an image $y_{i,j}$ for $f_{i,j}$, and send all the images to the verifier.

3. For $i = 1, \ldots, d$, the verifier samples m coins $c_{i,1}, \ldots, c_{i,m} \in \{0, 1\}$, and rejects if the prover does not respond with a preimage or a valid equation.

While the protocol is simply a parallel repetition applied to the previous protocol, proving that the soundness error reduces to negligible for sufficiently large m seems to require new ideas. Specifically, parallel repetition for a quantum prover interactive argument is currently known to only work in special cases such as four-message protocols with a special structure [2, 13]. Even for a constant-round protocol with more than four message exchanges, currently we do not know if classical techniques, e.g., the random termination transformation [22, 24], can help amplify the hardness. Furthermore, the standard approach to prove a direct product theorem is to give a reduction, i.e., suppose that there is a prover which succeeds with non-negligible probability for sufficiently large m, then we can use it to construct a prover that breaks the soundness in a one-copy protocol. In our context, when it comes to a prover with limited quantum depth, it is unclear whether the reduction incurs no (or small) depth overhead.

We give new ideas to prove a new direct product theorem for a special type of quantum prover interactive arguments in the following steps: To start, we consider a restricted quantum prover in the proof-of-quantumness protocol, described as follows. Without loss of generality for any four-message quantum prover interactive arguments, we view the two moves of the prover in the four-message protocol as two quantum algorithms $\mathcal{P}_1, \mathcal{P}_2$. The prover is restricted in the sense that the side information from $\mathcal{P}_1$ to $\mathcal{P}_2$ is purely classical. We show that when the prover is restricted to only pass along a purely classical side information, it only wins the proof-of-quantumness protocol with success probability bounded away from 1 by a constant.

With the same restriction, the key step is to show parallel repetition reduces the success probability exponentially in the number of copies using an argument similar to Haitner's approach in purely classical settings [22]. For our purpose, it then suffices to show give a reduction that breaks the soundness of this four-message protocol: When the prover has limited quantum depth, in some intermediate step, it has to destroy all the coherence to obtain a purely classical intermediate state. Thus the reduction of the first move fast forwards the protocol to this intermediate state, and embeds the external verifier's challenge to to the last round.

Preserving quantum circuit depth in the quantum Fiat-Shamir transformation. In the last step, we apply multi-round Fiat-Shamir by Don, Fehr, and Majenz [20] to reduce the round complexity to two messages, thereby achieving a non-interactive CVQD, in the random oracle model. To prove the security, suppose that there is a low-depth adversary $\mathcal{A}$ which succeeds with non-negligible probability. Our goal is to use $\mathcal{A}$ to construct a reduction $S^{\mathcal{A}}$ which responds to the verifier in a d-round protocol and wins with non-negligible success probability. The key observation is that the simulator randomly chooses d queries and measures the query register, and therefore the reduction itself does not increase the quantum depth.

We note that on the completeness side, the prover evaluates the random oracle classically. This implies that when the oracle is instantiated using a concrete hash function. e.g., SHA-3, it does not require additional quantum depth from the prover. Previously, there have been a few constructions for separation of quantum depth [12, 18, 4] in query models. Unfortunately, none of them seems to be instantiable with a quantum circuit of the same depth as the query algorithm, since they all require coherent evaluations of the oracle. A coherent evaluation of the oracle takes quantum depth one, but any instantiation incurs a blow-up in quantum depth for any function that is not known to have a depth-one implementation. By contrast, the evaluation of the hash function in our protocol is purely classical, and takes no additional quantum depth using any instantiation.

1.2.3 A two-prover protocol with optimal depth separation

The problem d-SSP for separating quantum depth. Our protocol can be seen as a two-player instantiation of the algorithms for solving d-SSP in [12], an oracle problem that distinguishes d- from $(2d+1)$-depth quantum circuits.

The problem is a shuffled version of the Simon's problem. Recall that for the "plain" Simon's problem, a constant-depth algorithm is sufficient to output the hidden shift. To turn the problem into one that certifies large quantum query depth, Chia, Chung, and Lai proposed the d-Shuffling Simon's problem (d-SSP) [12]. The algorithm is given oracle access to $d+1$ functions $f_0, \dots, f_d$, where $f_1, \dots, f_{d-1}$ are random permutations on an exponentially larger set, and the last function f_d is a 2-to-1 function such that $f_d \circ \cdots \circ f_2 \circ f_1(x) = f(x)$ for a Simon's function f. We call the functions $f_0, \dots, f_d$ to be a d-shuffling of a Simon's function f. The task is to find the hidden shift.

It is obvious that d-SSP remains easy for a $(2d+1)$-depth quantum algorithm which simulates a query to f using two queries to each function in $\{f_0, \dots, f_{d-1}\}$ and one query to f_d: first query $f_0, \dots, f_d$ in sequence to get

$$|x, y\rangle \mapsto |x, f_0(x), f_1 \circ f_0(x), \dots, f_{d-1} \circ \cdots \circ f_0(x), f(x) \oplus y\rangle, \tag{1}$$

and then query the first d functions in the reverse order to reset the intermediate registers to back zero states. On the other hand, any polynomial-time algorithm with quantum depth at most d cannot solve d-SSP. This follows from the intuitions that one needs to make $(d+1)$-sequential quantum queries to $f_0, \dots, f_d$ in order for evaluating f on a uniform superposition, and only f_d in an exponentially small random subset of the domain has information about f. Thus, any polynomial-time algorithm without sufficient quantum depth cannot even evaluate f in superposition.

To turn the problem into a protocol that certifies quantum depth, an idea is to have the verifier V play the role of the oracle, and checks if prover P_A outputs the hidden shift. The resulting protocol is quite straightforward: the prover P_A is allowed to perform arbitrary quantum computation (subject to its quantum resources) between message exchanges with the verifier. In the intermediate rounds, V computes the quantum circuits of the oracles on the state given by P_A, and sends the resulting quantum state back. At the end, the verifier accepts if P_A outputs the hidden shift. The analysis of the protocol is also straightforward. As long as the verifier implements the quantum-accessible oracles $f_0, \dots, f_d$ reliably between the computation performed by the prover, the completeness and soundness directly follows from the result of Chia, Chung, and Lai [12].

However, this approach has two drawbacks: First, the verifier needs to reliably implement a large QRAM to support quantum access to the oracle. This requires a reliable large-scale quantum computer that can solve problems in quantum polynomial time. Moreover, it requires reliable quantum communication between the prover and the verifier. None of the requirements seems to be within the reach in the near future.

In this paper, we give constructions that allows a purely classical veriifer to certify quantum depth. Our first protocol is to rely on the technique of self-testing to certify the untrusted quantum servers sharing entanglements. In particular, we apply a sequence of transformations from the aforementioned straightforward approach into one that has a weak requirement on the verifier, i.e., it runs in probabilistic polynomial time. We briefly introduce the techniques as follows.

Delegating the oracle to another quantum prover. To achieve purely classical verification, we introduce another untrusted prover, denoted P_O, which may share entanglements with P_A but they are not allowed to communicate with each other once the protocol starts. The verifier delegates the oracle computations to P_O, and checks if P_A outputs the hidden shift in the end. To make "queries," P_A forwards a quantum state by quantum teleportation.

To ensure that P_O behaves honestly, we modify the EPR protocol by Broadbent [10] to verify the computation of P_O. To understand how this works, let us recall some idea of the protocol. The original Broadbent protocol allows a weakly quantum verifier to delegate a quantum computation to the prover. To show that the prover has to be honest, the computation is made indistinguishable from two tests (X-test and Z-test). These tests are used to check if the prover's attack is trivial on the single bit the verifier aims to learn from the prover.

However, to apply the protocol to our problem, there are two caveats that remain to solve. First, the Broadbent protocol is designed for verifying a BQP-complete language. An instance in the language is a classical description of a unitary U with the promise that sampling the first qubit of $U|0\rangle$ by performing a standard basis measurement yields a 0 with probability at least $2/3$, or at most $1/3$. In our setting, we do not have such a promise. Secondly, the protocol only guarantees that the output b is an encryption of the random variable close to sampling $U|0\rangle$ by performing a standard basis measurement on the first qubit, provided the prover passes the tests with high probability. For our purposes, we would need to show if P_O's output state ρ_i on each query $|\psi_i\rangle$ is close to $O|\psi_i\rangle$ for each query $i \in \{0, 1, \ldots, d\}$ in a reasonable metric.

We show that with a modification, our variant of the Broadbent protocol is rigid in the sense that every prover that is accepted in our variant with probability $1 - \epsilon$ must output a state ρ_i which is $O(\epsilon)$-close the ideal state $O|\psi_i\rangle$ in trace distance. The modification requires a quantum channel which allows a transmission of $\mathrm{poly}(n)$ qubits between P_O and V, but the requirement is not necessary when we turn the protocol into a purely classical verification.

Dequantizing the verification. We further dequantize the quantum verification and communication by applying the Verifier-on-a-Leash protocol (also called the Leash protocol) by Coladangelo, Grilo, Jeffery and Vidick [17]. In a high level, the idea is to add another prover to perform the measurements by the quantum verifier in the Broadbent protocol, and check if the added prover behaves honestly.

To transform an oracle separation into purely classical verification, one possible approach is to add a third prover P_V to help certify that P_O behaves as intended. More concretely, the classical verifier asks P_A to perform the computation between queries, and P_O to apply the quantum circuit of the oracle. To check P_O behaves as intended, a third player P_V is added to perform the measurements in the bases determined from the rules of the Broadbent protocol. In our settings, none of the provers are assumed to be trusted. Thus it is necessary to verify P_V performs the measurements in the correct bases. Thus the verifier challenges P_V and P_O to run either the protocol for verifying P_O or a rigidity test to certify P_V, and the two choices are made indistinguishable to P_V's viewpoint.

However, this approach does not work directly. More specifically, for the security to hold, it is crucial that P_O does not distinguish the computational round and the test rounds. In this approach, P_O interacts with P_A via quantum teleportation to implement the original query algorithms in the computation round, whereas to certify P_O's behavior, the classical verifier must ask P_O to interact with P_V in the test round. Hence, P_O can determine the round type and cheat. Moreover, another drawback with this approach is that it requires three provers.

We can fix the issues about the aforementioned two-prover protocol by *asking P_A to play the role of P_V simultaneously*. To explain how this works, we consider the following protocol for a single-query algorithm: Initially, the verifier chooses to run the computation, X-test, Z-test or rigidity test. The prover P_A prepares an (arbitrary) initial n-bit quantum state $|\psi_0\rangle$ and teleports three states $|\psi_0\rangle, |0^n\rangle, |+^n\rangle$ to disjoint random subsets of P_O's halves of EPR pairs (the other halves are held by P_A). Note that the subsets are chosen by the verifier, but it does not reveal the underlying states. If any of the first three tests is chosen, the prover P_O performs the computation O on one of the subsets specified by the verifier. Note that since these three states are encrypted with quantum one-time pad, P_O cannot distinguish them and thus the round type. To perform the computation O, a set S of EPR pairs shared by P_O and P_A is used to implement gadgets for computing O. To be more specific, the verifier asks P_A to perform measurements on S in random bases, and chooses a subset (of S) on which P_A is specified to perform measurements in desirable bases determined by the rules of the Broadbent protocol according to the round type. The verifier then tells P_O to use the subset in S to compute O. The random-basis measurement on S is to certify the behavior of P_A in the rigidity test. Roughly speaking, when the rigidity test is chosen, the verifier can check (with help of P_O) if P_A performs the measurements on the EPR pairs in the random bases chosen by the verifier. Since P_A's behavior for all four tests are measurements in random bases, whether a rigidity test is executed is unknown to P_A. Note that although P_O can learn if a rigidity test is executed, it does not affect the security since P_O has no chance to reveal this to P_A.

However, there are a couple of issues that remain to address when considering multiple rounds of interaction between P_O and P_A. First, some tests running on more than one query can potentially reveal the type of the test. More specifically, if the verifier chooses to run the rigidity test, then P_O would certainly learn an application of the oracle unitary O is not necessary for this round. The prover P_A can possibly detect the choice of the test by observing the input state and the resulting state using, say, a swap test. Furthermore, to reflect the actual performance of the query algorithm, it is crucial that with sufficiently large probability, no test has been applied throughout the protocol. This is because when the computation is not performed in this round, computation applied in the following rounds will not yield a useful result (e.g., outputting the hidden shift for d-SSP), even when the provers opt to follow the protocol honestly. If the tests are nevertheless executed with very low probability, the provers may deviate from the protocol seriously.

We show that it suffices that the verifier randomly selects to certify one random query and trusts all the other queries, and with probability $\Theta(1/q)$, no test is executed for a q-query protocol. For the selected query, the verifier either asks P_O to certify P_A's measurements, or P_A to certify P_O performs the oracle unitary O by running the test phases in our variant of the Broadbent protocol. If the provers pass the test, the verifier accepts and terminates the protocol. Since the knowledge of the round type for certifying a query can only lead to an attack on the following queries, verification on a random query can prevent these issues from breaking the soundness of the protocol.

Putting things together. We then combine our aforementioned tests to turn an oracle separation problem into a two-player protocol. In particular, we show that with a suitable choice of the weights of entering each test, the completeness-soundness gap shrinks by at most an inverse polynomial multiplicative factor in the number of queries. More formally, we prove the implication by reduction. Suppose that in the protocol, there are provers P_A, P_O such that P_A is subject to its quantum resources and they break the soundness. Then we construct a query algorithm which succeeds with sufficiently large probability to break the soundness guarantee in the associated relativized world. Since the oracle separation problem distinguishes the quantum complexity classes, the resulting protocol yields a completeness-soundness gap $1/\operatorname{poly}(q)$ for a q-query algorithm.

Given that d-SSP is a oracle separation problem between a hybrid d-depth and a hybrid $(2d+1)$-depth computation, we conclude that our transformation yields a construction of $\mathsf{CVQD}_2(d, 2d+1)$ with gap $1/\operatorname{poly}(d)$. We apply a sequential repetition to amplify the gap to constant. The repetition itself does not require an increase of the quantum depth of P_A since the same hybrid computation can be reused.

Efficient instantiation. We have shown that an oracle separation problem implies a two-player protocol that distinguishes hybrid quantum computation with different quantum depth. However, to succeed in the protocol honestly, V must sample an oracle from a distribution $\mathcal{D}$ which is not known to be efficiently samplable, and P_O must perform a quantum circuit that implements O. In the problem d-SSP, the oracles consist of random permutations. By a counting argument, most of the permutations does not have an efficient implementation.

To address the issue, we leverage oracle indistinguishability in the associated relativized world. More concretely, suppose that for the distribution $\mathcal{D}$ of random d-shuffling of a random Simon's function, there is an efficiently samplable distribution $\mathcal{D}'$ which is indistinguishable from $\mathcal{D}$. Then in the two-player protocol, when the efficient verifier samples the oracle according to $\mathcal{D}'$, the soundness error is increased negligibly. The idea for showing this directly follows from our proof for showing an oracle separation implies a two-player protocol. For every query algorithm $\mathcal{A}$ that has small quantum depth, it succeeds with probability at most p when the oracle is sampled from $\mathcal{D}$. Then replacing $\mathcal{D}$ with $\mathcal{D}'$, by the oracle indistinguishability, $\mathcal{A}$ succeeds with probability at most negligibly close to p. Applying the transformation with $\mathcal{D}'$ yields a sound two-player protocol with efficient P_O and V.

We show how to give a distribution $\mathcal{D}'$ using quantum-secure pseudorandom permutations (qPRP) against adversaries making queries to the permutation and its inverse. In particular, to sample a pseudorandom d-shuffling of a Simon's function, first sample d independent keys $k_0, \ldots, k_{d-1}$ from the key space of the pseudorandom permutation P and let $f_i = P(k_i, \cdot)$ for $i \in \{0, 1, \ldots, d-1\}$. For the last function, again by a counting argument, not every Simon's function has an efficient implementation. We observe that every Simon's function can be computed by composing a permutation and an efficiently computable function that is constant on every one dimensional affine subspace of the form $\{x, x \oplus s\}$. This implies that sampling a random Simon's function can be done by sampling a random shift and a random permutation. Replacing the random permutation with a pseudorandom permutation yields a pseudorandom Simon's function.

A nearly optimal separation. As mentioned previously, the problem d-SSP provides an oracle separation between d- and $(2d+1)$-depth quantum circuits in the presence of polynomial-time classical computation. Next we further improve the separation to distinguish d- from $(d+3)$-depth hybrid quantum computation. In particular, we modify the problem to allow a $(d+3)$-depth algorithm to succeed with high probability, while at the same time, it remains hard for a d-depth prover to learn the hidden shift.

First we recall that a $(2d+1)$-depth quantum algorithm is needed because to simulate a query to f, the algorithm queries $f_0, f_1, \ldots, f_d$ followed by queries to $f_{d-1}, \ldots, f_0$ to uncompute the intermediate values. To avoid the need of extra depth for uncomputation, our idea is to replace the standard access to $f_0, \ldots, f_d$ with "in-place oracles." In this model, the algorithm is given access to $|x\rangle \xmapsto{f_i} |f_i(x)\rangle$, and thus the intermediate queries have been erased automatically. While in-place oracle access to an arbitrary is not a unitary in general, in our case, perhaps fortunately, the functions $f_0, \ldots, f_d$ are either permutations or 2-to-1 functions. It is clear that the in-place oracle access for permutations is a unitary. Furthermore, we modify the last function f_d such that f_d is bijective, but a depth $(d+1)$-depth algorithm can simulate a query to the underlying Simon's function f with constant probability. We call the same problem with in-place oracle access the *in-place* d-Shuffling Simons Problem (in-place d-SSP, see Definition 7.10). Finally, we show that that the in-place d-SSP cannot be solved by any hybrid quantum-classical computers with quantum circuit depth at most d.

Related work. To our knowledge, all the existing works for separation between hybrid schemes are given relative to classical oracles. First, Chia, Chung, and Lai [12] and Coudron and Menda [18] independently gave separation between hybrid schemes of different quantum depth, relative to different classical oracles. In the former, Chia, Chung, and Lai uses the Simon's problem with a shuffling of the Simon's function. Coudron and Menda uses the Welded Tree Problem, defined by Childs, Cleve, Deotto, Farhi, Gutmann, and Spielman [15].

More recently, in an independent work [23], Atsuya Hasegawa and François Le Gall defined the d-Bijective Shuffling Simon's Problem that improves the quantum depth separation in [12] to d versus $d+1$ using the similar idea as in-place d-SSP (Definition 7.10). For in-place d-SSP, the gap is d versus $d+1$ if we consider the same models as in Definition 3.8 and Definition 3.10 in [12]. However, the models in [12] count the depth of quantum queries to the oracle. In this work, we also count the two layers of Hadamard transforms at the beginning and the end of Simon's algorithm. This results in the gap d versus $d+3$ in our first result in Theorem 1.2 (see Theorem 7.11 and Corollary 7.12 for formal statements).

In a subsequent work, Arora, Coladangelo, Coudron, Gheorghiu, Singh, and Waldner [4] took ideas from Chia, Chung, and Lai [12] and the previous version of this work to show an oracle separation as in [12] in the quantum random oracle model and can directly convert it to a non-interactive CVQD protocol with the corresponding depth separation (d versus $2d+3$) relative to a quantum random oracle. However, their result cannot be instantiated under the random oracle heuristic, which follows from the fact that the instantiation does not preserve quantum depth. In particular, an honest prover in their protocol requires quantum access to the oracle, and instantiating the oracle using known heuristics (such as SHA-3) requires a large circuit depth; hence, the separation will be d versus $2d \times \mathrm{poly}(n)$, where $\mathrm{poly}(n)$ is used to implement the random oracle heuristic. This makes the protocol hard to be implemented. On the other hand, our two-message protocol only needs classical access to the random oracle and thus can keep the depth separation when instantiating the random oracle. Also, our other CVQD and CVQD_2 protocols have a tighter separation under different cryptographic assumptions.

1.3 Organization

The rest of the paper is organized as follows. Section 2 includes the required technical background knowledge for this paper, and our modifications of the previous protocols which will be useful for our contributions. In Section 3, we present a new single-prover protocol from QLWE. In Section 4, we reduce the round complexity of the protocol in Section 3 via the multi-round Fiat-Shamir transform.

The second part of this work describes two-player protocols for certifying quantum depth. Section 5 defines a transformation from a quantum oracle separation to a two-prover protocol that preserves completeness and soundness. Section 6 presents a framework that transforms a quantum oracle separation to a two-prover protocol with a classical verifier. Section 7 shows a protocol for classical verification of quantum depth under the framework developed in Section 5 and Section 6.

Acknowledgement

We thank Scott Aaronson, Kai-Min Chung, and anonymous reviewers for helpful suggestions on an earlier version of this paper. We also thank Atsuya Hasegawa and François Le Gall for sharing their results [23] with us. SHH acknowledges the support from Simons Investigator in Computer Science award, award number 510817. NHC was supported by NSF Award FET-2243659, Google Scholar Award, and DOE Quantum Testbed Finder Award DE-SC0024301.

2 Preliminaries

For finite set $\mathcal{X}$, we denote $x \leftarrow_R \mathcal{X}$ the process of sampling a random variable x uniformly from $\mathcal{X}$. For distribution $\mathcal{D}$ over a finite set $\mathcal{X}$, we denote $x \leftarrow_R \mathcal{D}$ the process of sampling a random variable $x \in \mathcal{X}$ according to $\mathcal{D}$. For a classical or quantum process $\mathcal{A}$, we denote $y \leftarrow \mathcal{A}(x)$ to specify that $\mathcal{A}$ on input x outputs y. A function $f : \mathbb{N} \to \mathbb{R}$ is negligible, denoted $f(n) = \mathrm{negl}(n)$, if there exists an integer n_0 such that for $n \geq n_0$, $f(n) \leq n^{-c}$ for every constant c. In other words, f if negligible if $f(n) = n^{-\omega(1)}$. We use the notation 1_P to denote 1 if P is true and 0 if P is false.

2.1 Oracle separation for quantum depth

We first introduce the two models for interleaving d-depth quantum circuits and classical polynomial-time computation.

Definition 2.1 (d-CQ scheme [12])**.** *Let $k = \mathrm{poly}(n)$. Let $\mathcal{A}_c^1, \ldots, \mathcal{A}_c^k$ be a sequence of classical polynomial-time algorithms and $\mathcal{A}_q^1, \ldots, \mathcal{A}_q^k$ be a sequence of d-depth quantum circuits. A d-CQ scheme can be represented as following:*

$$\mathcal{A}_c^k \circ (\Pi_{0/1} \circ \mathcal{A}_q^k) \circ \cdots \circ \mathcal{A}_c^2 \circ (\Pi_{0/1} \circ \mathcal{A}_q^2) \circ \mathcal{A}_c^1 \circ (\Pi_{0/1} \circ \mathcal{A}_q^1),$$

where, $\Pi_{0/1}$ is a measurement on all qubits in the computational basis.

Definition 2.2 (d-QC scheme [12])**.** *Let $k = \mathrm{poly}(n)$. Let $\mathcal{A}_c^0, \mathcal{A}_c^1 \ldots, \mathcal{A}_c^d$ be a sequence of classical polynomial-time algorithms and $\mathcal{A}_q^1, \ldots, \mathcal{A}_q^d$ be a sequence of 1-depth quantum circuits. A d-CQ scheme can be represented as following:*

$$\mathcal{A}_c^d \circ (\Pi_{0/1} \otimes I) \circ \mathcal{A}_q^d \circ \cdots \circ \mathcal{A}_c^2 \circ (\Pi_{0/1} \otimes I) \circ \mathcal{A}_q^2 \circ \mathcal{A}_c^1 \circ (\Pi_{0/1} \otimes I) \circ \mathcal{A}_q^1 \circ \mathcal{A}_c^0,$$

where, $\Pi_{0/1} \otimes I$ is a computational basis measurement that only operates on part of the qubits. The input of $\mathcal{A}_q^i$ includes the output quantum state of $\mathcal{A}_q^{i-1}$ for $i = 2, \ldots, d$ and the classical information from $\mathcal{A}_c^j$ and the measurement outcome of $\mathcal{A}_q^j$ for $j < i$. The input of $\mathcal{A}_c^i$ includes the measurement outcome of $\mathcal{A}_q^i$ and other classical information from $\mathcal{A}_c^j$ and $\mathcal{A}_q^j$ for $j < i$ for all $i \in [d]$.

Remark 2.3. *In this work, we generally choose the universal gateset to be all one- and two-qubit gates. In particular, the impossibility results in Theorem 1.2 showing all d-CQ and d-QC schemes fail the* CVQD *protocols hold for any universal gateset with bounded fan-in gates.*

Roughly speaking, d-CQ schemes allow a classical algorithm to access a d-depth quantum circuit polynomially many times; however, all the qubits of the quantum circuit need to be measured after each access (and thus no quantum state can be passed to following d-depth quantum circuits). On the other hand, d-QC schemes let a quantum circuit to access classical algorithms after each depth and pass quantum states to the rest of the circuits for at most d depths.

In [12], the class of problems that can be solved by d-CQ schemes is defined as $\mathsf{BPP}^{\mathsf{BQNC}_d}$, and the class of problems that can be solved by d-QC schemes is defined as $\mathsf{BQNC}_d^{\mathsf{BPP}}$.

Chia, Chung and Lai [12] presented an oracle problem that can separate schemes in Definition 2.1 and Definition 2.2 with different quantum circuit depths. We briefly introduce the oracle separation in the following.

Definition 2.4 (d-shuffling [12, Definition 4.1])**.** *Let $f : \{0,1\}^n \to \{0,1\}^n$ be any function. A d-shuffling of f is defined by $\mathcal{F} := (f_0, \ldots, f_d)$, where $f_0, \ldots, f_{d-1}$ are random permutations over $\{0,1\}^{(d+2)n}$. The last function f_d is a fixed function satisfying the following properties: let $S_d := \{f_{d-1} \circ \cdots \circ f_0(x') : x' \in \{0,1\}^n\}$.*

- *For $x \in S_d$, let $f_{d-1} \circ \cdots \circ f_0(x') = x$, and choose the function $f_d : S_d \to [0, 2^n - 1]$ such that $f_d \circ f_{d-1} \circ \cdots \circ f_0(x') = f(x')$.*
- *For $x \notin S_d$, $f_d(x) = \perp$.*

Then, we recall the definition of Simon's function.

Definition 2.5 (Simon's function)**.** *For a finite set S and $s \in \{0,1\}^n$ (also called the hidden shift), the Simon's function $f : \{0,1\}^n \to S$ satisfies that $f(x) = f(x')$ if and only if $x' = \{x, x \oplus s\}$.*

The Simon's problem is to compute the hidden shift s given oracle access to a Simon's function f. The quantum algorithm for Simon's problem uses one quantum query to sample a random vector y satisfying $y \cdot s = 0$. Making $O(n)$ queries suffices to find a generating set of the subspace $H = \{y : y \cdot s = 0\}$ with overwhelming probability, and thus the hidden shift is uniquely determined from the generators. It is worth noting that any classical algorithm that finds s with high probability requires $\Omega(\sqrt{2^n})$ queries even if the Simon's function is given uniformly randomly. A random Simon's function is defined as a function drawn uniformly from the set of Simon's functions from $\{0,1\}^n$ to S and we choose $S = \{0,1\}^n$.

We now define the d-Shuffling Simon's problem (d-SSP) that separates $\mathsf{BPP}^{\mathsf{BQNC}_{2d+3}} \cap \mathsf{BQNC}_{2d+3}^{\mathsf{BPP}}$ from $\mathsf{BPP}^{\mathsf{BQNC}_d} \cup \mathsf{BQNC}_d^{\mathsf{BPP}}$ relative to an oracle.

Problem 1 (d-shuffling Simon's problem (d-SSP) [12, Definition 4.9])**.** *Let $n \in \mathbb{N}$ and $f : \{0,1\}^n \to \{0,1\}^n$ be a random Simon's function. Given oracle access to the d-shuffling $\mathcal{F} := \{f_0, f_1, \ldots, f_d\}$ of f, the problem is to find the hidden shift s of f.*

Chia, Chung and Lai showed the following theorem [12].

Theorem 2.6 ([12])**.** *Let $d = \mathrm{poly}(n)$. The d-SSP problem can be solved by $(2d+3)$-CQ and $(2d+3)$-QC schemes with oracle access to the d-shuffling oracle of f. Furthermore, for any d'-CQ and d'-QC schemes $\mathcal{A}$ with with oracle access to the d-shuffling oracle of f and $d' \leq d$, the probability that $\mathcal{A}$ solves the problem is negligible.*

Remark 2.7. *In [12], it said that d-SSP can be solved by $(2d+1)$-CQ and -QC schemes because the models defined in Definition 3.8 and Definition 3.10 in [12] mainly considered the depth for querying the oracle. Here, for our purpose, we count the two Hadamard transforms at the beginning and the end of Simon's algorithm, which gives additional two depths.*

This means that when there is a quantum algorithm of $(2d+3)$ quantum circuit depth (including access to the oracle) succeeding with probability at least $2/3$ (in fact the success probability is $1 - \mathrm{negl}(n)$). The second part of Theorem 2.6 shows that every quantum algorithm of quantum circuit depth at most d outputs the hidden shift with negligible probability, even if it makes an arbitrary polynomial number of queries.

2.2 Quantum-secure pseudorandom permutations

For our task, we also want the oracle can be implemented efficiently. However, by a counting argument, a random permutation cannot be computed efficiently with overwhelming probability. We use pseudorandom permutations to address this issue. In a query model, we can replace a random permutation with a pseudorandom one without decreasing the performance of a query algorithm by non-negligible difference.

Definition 2.8 (Quantum-secure pseudorandom permutations (qPRP) [32])**.** *For security parameter λ and a polynomial $m = m(\lambda)$, a pseudorandom permutation P over $\{0,1\}^m$ is a keyed function $\mathcal{K} \times \{0,1\}^m \to \{0,1\}^m$ such that there exists a negligible function ϵ such that for every quantum adversary $\mathcal{A}$, it holds that*

$$\left| \Pr_{F \leftarrow_R \mathcal{P}}[\mathcal{A}^{O_F, O_{F^{-1}}} = 1] - \Pr_{k \leftarrow_R \mathcal{K}}[\mathcal{A}^{O_{P(k,\cdot)}, O_{P^{-1}(k,\cdot)}} = 1] \right| \leq \epsilon(\lambda), \tag{2}$$

where $\mathcal{P}$ is the set of permutations over $\{0,1\}^m$ and $O_Q : |x, y\rangle \mapsto |x, y \oplus Q(x)\rangle$ for permutation $Q : \{0,1\}^m \to \{0,1\}^m$ and $x, y \in \{0,1\}^m$.

2.3 Proof of quantumness

Our protocol in the plain model will be based on a construction of noisy trapdoor claw-free function (NTCF) function family [8], based on QLWE.

Definition 2.9 (NTCF family [8, Definition 3.1])**.** *Let λ be a security parameter and $\mathcal{X}, \mathcal{Y}$ be finite sets. Let $\mathcal{K}_{\mathcal{F}}$ be a finite set of keys. A family of functions*

$$\mathcal{F} = \{f_{k,b} : \mathcal{X} \to \mathcal{D}_{\mathcal{Y}}\}_{k \in \mathcal{K}_{\mathcal{F}}, b \in \{0,1\}} \tag{3}$$

is called a noisy trapdoor claw free (NTCF) family if the following conditions hold:

1. *Efficient function generation. There exists an efficient probabilistic algorithm* $\textsc{Gen}_{\mathcal{F}}$ *which generates a key $k \in \mathcal{K}_{\mathcal{F}}$ together with a trapdoor t: $(k, t) \leftarrow \textsc{Gen}_{\mathcal{F}}(1^\lambda)$.*

2. *Trapdoor injective pair. For all keys $k \in \mathcal{K}_{\mathcal{F}}$, the following conditions hold.*

   (a) *Trapdoor: There exists an efficient deterministic algorithm* $\textsc{Inv}_{\mathcal{F}}$ *such that for all $b \in \{0,1\}$, $x \in \mathcal{X}$ and $y \in \textsc{Supp}(f_{k,b}(x))$, $\textsc{Inv}_{\mathcal{F}}(t, b, y) = x$. Note that this implies that for all $b \in \{0,1\}$ and $x \neq x'$, $\textsc{Supp}(f_{k,b}(x)) \cap \textsc{Supp}(f_{k,b}(x')) = \emptyset$.*

   (b) *Injective pair: there exists a perfect matching $\mathcal{R}_k \subseteq \mathcal{X} \times \mathcal{X}$ such that $f_{k,0}(x_0) = f_{k,1}(x_1)$ if and only if $(x_0, x_1) \in \mathcal{R}_k$.*

3. *Efficient range superposition. For all key $k \in \mathcal{K}_{\mathcal{F}}$ and $b \in \{0,1\}$, there exists a function $f'_{k,b} : \mathcal{X} \to \mathcal{D}_{\mathcal{Y}}$ such that the following hold.*

   (a) *For all $(x_0, x_1) \in \mathcal{R}_k$ and $y \in \textsc{Supp}(f'_{k,b}(x_b))$, $\textsc{Inv}_{\mathcal{F}}(t, 0, y) = x_0$ and $\textsc{Inv}_{\mathcal{F}}(t, 1, y) = x_1$.*

   (b) *There exists an efficient deterministic procedure* $\textsc{Chk}_{\mathcal{F}}$ *that, on input $k, b \in \{0,1\}$, $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, returns 1 if $y \in \textsc{Supp}(f'_{k,b}(x))$ and 0 otherwise. Note that* $\textsc{Chk}_{\mathcal{F}}$ *is not provided the trapdoor t.*

(c) For every k and $b \in \{0,1\}$,

$$\mathop{\mathbb{E}}_{x \leftarrow_R \mathcal{X}}[H^2(f_{k,b}(x), f'_{k,b}(x))] \leq \mu(\lambda), \tag{4}$$

for some negligible function μ. Here H^2 is the Hellinger distance. Moreover, there exists an efficient procedure $\textsc{Samp}_{\mathcal{F}}$ *that on input k and $b \in \{0,1\}$, prepares the state*

$$|\mathcal{X}|^{-1/2} \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} \sqrt{f'_{k,b}(x)(y)} |x\rangle|y\rangle. \tag{5}$$

4. *Adaptive hardcore bit. For all keys $k \in \mathcal{K}_{\mathcal{F}}$, the following conditions hold for some integer w that is a polynomially bounded function of λ.*

   (a) For all $b \in \{0,1\}$ and $x \in \mathcal{X}$, there exists a set $G_{k,b,x} \subseteq \{0,1\}^w$ such that $\Pr_{d \leftarrow_R \{0,1\}^w}[d \notin G_{k,b,x}]$ is negligible, and moreover there exists an efficient algorithm that checks for membership in $G_{k,b,x}$ given k, b, x and the trapdoor t.

   (b) There is an efficiently computable injection $J : \mathcal{X} \to \{0,1\}^w$ such that J can be inverted efficiently on its range, and such that the following holds. If

$$\begin{aligned} H_k &= \{(b, x_b, d, d \cdot (J(x_0) \oplus J(x_1))) | b \in \{0,1\}, (x_0, x_1) \in \mathcal{R}_k, d \in G_{k,0,x_0} \cap G_{k,1,x_1}\}, \\ \bar{H}_k &= \{(b, x_b, d, c) | (d, x, d, c \oplus 1) \in H_k\}, \end{aligned}$$

   then for any quantum polynomial-time procedure $\mathcal{A}$, there exists a negligible function μ such that

$$\left| \Pr_{(k,t) \leftarrow \textsc{Gen}_{\mathcal{F}}(1^\lambda)}[\mathcal{A}(k) \in H_k] - \Pr_{(k,t) \leftarrow \textsc{Gen}_{\mathcal{F}}(1^\lambda)}[\mathcal{A}(k) \in \bar{H}_k] \right| \leq \mu(\lambda). \tag{6}$$

In a breakthrough, Brakerski, Christiano, Mahadev, Vazirani and Vidick give a proof-of-quantumness protocol (the BCMVV protocol) based on NTCFs [8]. The protocol proceeds in the following steps.

1. The verifier samples $(k,t) \leftarrow \textsc{Gen}(1^\lambda)$ and sends k to the prover.

2. The prover performs $\textsc{Samp}_{\mathcal{F}}$ on the input state $|+\rangle$ and measures the image register to yield an outcome y.

3. The verifier samples a random coin $c \leftarrow_R \{0,1\}$ and sends c to the prover.

4. If $c = 0$, the prover performs a standard basis measurement; otherwise the prover performs a Hadamard basis measurement. The outcome w is then sent to the verifier.

5. The verifier outputs $V(t,c,w)$, which is defined as

$$V(t,y,c,w) := \begin{cases} 1 & \text{if } c = 0 \text{ and } \textsc{Chk}_{\mathcal{F}}(k,y,b,x) = 1,\ w = (b,x) \\ 1 & \text{if } c = 1,\ d \in G_{k,y} \text{ and } d \cdot (J(x_0) \oplus J(x_1)) = u,\ w = (u,d) \\ 0 & \text{otherwise,} \end{cases} \tag{7}$$

   where $G_{k,y} := G_{k,0,x_0} \cap G_{k,1,x_1}$. Here we note that for $c = 0$, $\textsc{Chk}_{\mathcal{F}}$ does not require the trapdoor t, but the bit can be determined using the trapdoor.

The adaptive hardcore bit property (see (6)) implies that every adversary $\mathcal{A}$ given access to k, outputs y, w_0, w_1 such that $V(y, 0, w_0) = V(y, 1, w_1) = 1$ with probability at most $1/2 + \mathrm{negl}(\lambda)$. On the other hand, there exist efficient quantum processes to output valid (y, w_0) or (y, w_1) with probability $1 - \mathrm{negl}(\lambda)$.

Hirahara and Le Gall [25] and Liu and Gheorghiu [28] independently proposed two different methods of transforming the BCMVV protocol [8] into one that can be computed using only constant quantum depth (interleaving with classical computation). In our second protocol for certifying quantum depth based on LWE, we will be using the theorem which states that there exists a construction of NTCF family for which the function evaluation takes constant depth, based on randomized encoding, defined as follows.

Definition 2.10 (Randomized encoding [3, 28])**.** *Let $f : \{0,1\}^n \to \{0,1\}^\ell$ be a function and $r \leftarrow_R \{0,1\}^m$. A function $\hat{f} : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}^s$ is a δ-correct, ϵ-private randomized encoding of f if it satisfies the following properties:*

- *Efficient generation: there exists a deterministic polynomial-time algorithm that, given a description of the circuit implementing f, outputs a description of a circuit implementing $\hat{f}$.*
- *δ-correctness: there exists a deterministic polynomial-time algorithm* Dec*, called the decoder, such that for every input $x \in \{0,1\}^n$, $\Pr_r[\textsc{Dec}(\hat{f}(x,r)) \neq f(x)] \leq \delta$.*
- *ϵ-privacy: there exists a PPT algorithm S, called the simulator, such that for every $x \in \{0,1\}^n$, the total variation distance between $S(f(x))$ and $\hat{f}(x,r)$ is at most ϵ.*

Furthermore, a perfect randomized encoding is one for which $\epsilon = \delta = 0$.

Theorem 2.11 ([28, Section 3.1])**.** *There exists an efficient quantum process which uses gates of bounded fan-out in constant quantum depth and prepares the state*

$$\sum_{b,x} |b\rangle|x\rangle|\hat{f}_k(b,x)\rangle, \tag{8}$$

where $\hat{f}$ is a perfect randomized encoding of an NTCF f.

In particular, as shown in the same paper [28, Section 3.3], the proof-of-quantumness protocol takes total quantum depth 14 and three quantum-classical interleavings.

Theorem 2.12 ([28, Theorem 3.1])**.** *There exists a perfect randomized encoding of an NTCF, which satisfies the randomness reconstruction property and is an NTCF.*

In particular, Theorem 2.12 implies that with the new construction, the adaptive hardcore bit property holds.

2.4 The Broadbent protocol

In this section, we briefly introduce the Broadbent protocol for verifying quantum computation [10]. The protocol consists of two parties, the prover P which is untrusted but can perform arbitrary quantum computation, and the verifier V which is almost classical. In particular, V can perform measurements in certain bases. The prover and the verifier interact, and at the end of the protocol, the verifier outputs a bit which is either "accept" or "reject." The protocol can be used to verify a complete language in BQP (more precisely, PromiseBQP):

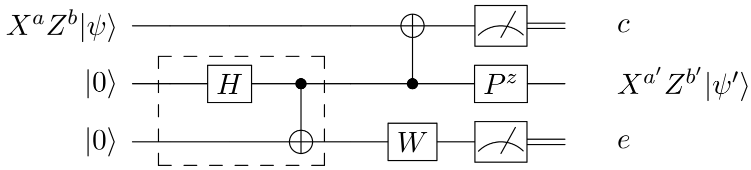

Figure 3: The quantum circuit of the T gadget. The dashed box prepares an EPR pair. The first two qubits are held by the prover and the third qubit is held by the verifier. The bit $z \leftarrow_R \{0,1\}$ is sampled independently for each T gadget by the verifier. For the choice of W, see Table 1.

Round type	Unitary W
Computation	$HP^{a'+c+z}T$
Even parity	$\mathbb{1}$
Odd parity	HP^z

Table 1: The choice of unitary W in a T gadget. The table is modified from [17, Table 3].

- Completeness: if the computation U satisfies $\|\Pi_0 U|0\rangle\|^2 \geq 2/3$, then there exists a quantum prover which makes V accept with probability at least c.
- Soundness: if the computation U satisfies $\|\Pi_0 U|0\rangle\|^2 \leq 1/3$, then for every prover, the verifier accepts with probability no more than s.

Here the projector $\Pi_0 = |0\rangle\langle 0| \otimes \mathbb{1}$ refers to the event that measuring the first qubit of the state $U|0\rangle$ in the standard basis yields an outcome 0. The parameters c, s are called the completeness and soundness respectively.

In the Broadbent protocol, the prover and the verifier share $(n+t)$ EPR pairs, where n is the number of qubits the computation U acts on, and t is the number of T gadgets in U. The quantum circuit for the T gadget is shown in Figure 3. The computation U is a quantum circuit consisting of gates in $\{CNOT, H, T\}$ (which is a universal gate set), performed on a quantum state encrypted by quantum one-time pad. The prover is designated to perform the following operations for each gate.

- For each $CNOT$ gate, the prover is designated to perform the gate on the associated qubits.
- For each T gate, the prover performs a T gadget using a half of an EPR pair as an ancilla qubit.
- Each Hadamard gate is compiled into the sequence $HTTHTTHTTH$ of single-qubit gates. For each T gate in the sequence, they run a T gadget. For each Hadamard gate in the sequence, the prover applies a Hadamard gate. We will also refer to the implementation as a Hadamard gadget (or an H gadget for short).

The verifier V chooses to run one of the following rounds, and performs the operations on the other halves of the $(n+t)$ EPR pairs depending on the choice of rounds. More specifically, V performs measurements on the EPR pairs to create the initial state and the states required for each T gadget, and the actions for each round type is specified as follows.

- Computation round (with constant probability p): the computation is delegated to the prover P. If P behaves honestly, the result can be recovered by V. In the beginning of the protocol, V measures the first n EPR pairs in the standard basis. The half held by the prover collapses

Gate/Gadget	Key update rule
T gadget, computation	$(a+c, b+e+a+c+(a+c)z)$
T gadget, even parity	$(e, 0)$
T gadget, odd parity	$(0, b+e+z)$
H gate	(b, a)
$CNOT$ gate	$(a, b+b', a+a', b')$

Table 2: Rules for updating the one-time-pad keys after applying each gate in the protocol. The bit z is chosen uniformly by the verifier, and the measurement outcomes e, c are obtained by the prover (see Figure 3). For the T gadget and the Hadamard gate, before updating the key is (a, b). For a CNOT gate, the keys are (a, b) on the control qubit and (a', b') on the target.

to a quantum one-time pad of the zero state. Then the (honest) prover performs the quantum computation on this state. For each T-gadget (used to apply a T gate or a Hadamard gate), the prover uses the a half of the EPR pair associated with the T-gadget as an ancilla qubit, and the verifier performs the associated measurement so the T-gadget implements a T gate, up to a key update (Table 2).

- Test round (with probability $1-p$): V tests if P behaves honestly, and rejects if any error is detected. A test round has two types, each of which is executed with probability $1/2$, outlined as follows.
  - An X-test round is used to detect if there is a bit flip error. In an X-test round, the verifier measures the first n EPR half in the standard basis to create a quantum one-time pad of the zero state $|0\rangle^{\otimes n}$. For each T-gadget, the verifier performs the measurement in the basis such that the T-gadget acts trivially (as the identity operation) up to a key update (Table 2). In the end, the verifier applies the key update rule to compute the key, and decrypts the first qubit. The verifier accepts if the bit is 0 and rejects otherwise.
  - A Z-test round is used to detect if there is a phase flip error. The operations are the same as the X-test round except that they are performed in the Hadamard basis. The verifier measures the first n EPR pairs in the Hadamard basis to create a quantum one-time pad of the plus state $|+\rangle^{\otimes n}$. For each T gadget, the verifier performs the measurement in some basis such that the T gadget acts trivially up to a key update (Table 2). In the end, the verifier applies the key update rule to compute the key, and decrypts the first qubit. The verifier disregards the result and rejects only if any error was detected throughout the computation.

The protocol performs quantum computation on encrypted data by quantum one-time pad. Depending on the round type, different key update rules are adopted. For a Hadamard gadget, six T-gadgets are performed. Recall that in an X-test round, a T-gadget acts trivially on the zero state. Passing an odd number of Hadamard gates yields an encrypted plus state, and thus the T-gadget used in this case will be the same as a T gate application in a Z-test round. Thus for convenience, we may define the parity of a T-gadget as follows. A T-gadget is of even parity if it is not part of an Hadamard gadget, or an even (resp. odd) number of Hadamard gates has been applied before in an Hadamard gadget in an X-test (resp. a Z-test) round; otherwise it is of odd parity. The key update rules are formally defined in Table 2.

The original Broadbent protocol is used to verify BQP languages (more precisely $\mathsf{PromiseBQP}$). Now we consider the following modification to show a rigidity result: the last message from the prover to the verifier is an n-qubit quantum state ρ. The verifier computes the key according to the key update rule (Table 2), and applies one-time pad decryption on ρ, with the following modification after the last message from the prover is sent.

- In an X-test round, the verifier accepts if measuring the state in the standard basis yields 0^n.
- In a Z-test round, the verifier accepts if measuring the state in the Hadamard basis yields 0^n.

The differences are instead of checking only the first qubit, the verifier determines, in a test round, if an attack has been applied on any of the qubits. The new verification procedure requires a larger quantum channel for the prover to send the entire state to the verifier. As we will see, the extra cost is not necessary when the protocol is made purely classical with two provers.

Our rigidity statement is formally stated as follows: if the prover is accepted with probability $1-\epsilon$ in test rounds, the prover in computation run implements a quantum channel $\mathcal{E}_C$ that is $O(\epsilon)$-close to the honest computation. First recall the following fact about Pauli twirls.

Lemma 2.13 (Pauli twirls). *Let $P_a := X^{a_1}Z^{a_2}$ denote a Pauli operator for $a = (a_1, a_2)$ where $a_1, a_2 \in \{0,1\}^n$. For any quantum state ρ and quantum channel Φ, it holds that*

$$\sum_{a\in\{0,1\}^{2n}} P_a^\dagger \Phi(P_a \rho P_a^\dagger) P_a = \sum_a r_a P_a \rho P_a^\dagger, \tag{9}$$

for some distribution r over Pauli matrices.

Broadbent shows that without loss of generality, any attack performed by the prover can be written as a honest execution C followed by a quantum channel Φ. The prover first performs C and yields a quantum state $\rho = C|\psi\rangle\langle\psi|C^\dagger$, one-time padded with key Q. The prover then applies any quantum channel Φ, followed by decryption performed by the verifier. By Lemma 2.13, the prover's attack can be written as a probabilistic mixture of Pauli unitaries. We prove the following theorem.

Theorem 2.14. *For $\epsilon \in [0, 1/2]$, any prover who succeeds with probability $1-\epsilon$ in the test rounds if and only if it implements a quantum channel $\mathcal{E}_C$ satisfying $\|\mathcal{E}_C - \mathcal{C}\|_\diamond \le 4\epsilon$ where $\mathcal{C}(\rho) := C\rho C^\dagger$ in the computational round.*

Proof. Without loss of generality, we may express the action of any prover given access to circuit description C and the verifier's decryption as a quantum channel

$$\mathcal{E}_{C,i} = \mathop{\mathbb{E}}_k(\mathcal{O}_k \circ \Phi \circ \mathcal{O}_k) \circ \mathcal{C}_i = \mathcal{P} \circ \mathcal{C}_i, \tag{10}$$

for some Pauli channel $\mathcal{P}$ with Kraus form $\{r_a^{1/2} P_a : a \in \{0,1\}^{2n}\}$ by Lemma 2.13 and round type i. Also recall that here we denote $P_a = X^{a_1}Z^{a_2}$ for $a = (a_1, a_2)$ and $a_1, a_2 \in \{0,1\}^n$. Since these rounds look completely identical to the prover, the quantum channel Φ must be identical among the choices of round type.

Let the success probability of the prover in an X- and a Z-test round be $1-\epsilon_X$ and $1-\epsilon_Z$ respectively. If the prover succeeds with probability $1-\epsilon$ conditioned on the event that a test round is chosen, it must hold that $\epsilon_X, \epsilon_Z \le 2\epsilon$ since otherwise $\frac{1}{2}(1-\epsilon_X) + \frac{1}{2}(1-\epsilon_Z) = 1 - \frac{\epsilon_X+\epsilon_Z}{2} < 1-\epsilon$.

In an X-test round, $\mathcal{C}_X = \mathcal{I}$, the identity channel, and

$$\begin{aligned} 1-\epsilon_X &= \langle 0^n|\mathcal{P}(|0^n\rangle\langle 0^n|)|0^n\rangle \\ &= \sum_a r_a |\langle 0^n|P_a|0^n\rangle|^2 \\ &= \sum_{a:a_1=0} r_a \ge 1-2\epsilon. \end{aligned} \tag{11}$$

Similarly, in a Z-test round, $\sum_{a:a_2=0} r_a \geq 1-2\epsilon$. Combining the inequalities, we conclude that $r_0 \geq 1-4\epsilon$. Therefore, for any prover who succeeds with probability $1-\epsilon$, it holds that in the computation round

$$\begin{aligned}\|\mathcal{P}\circ\mathcal{C}-\mathcal{C}\|_\diamond &\leq \|\mathcal{P}-\mathcal{I}\|_\diamond \\ &= \max_{\rho:\mathrm{tr}\rho=1} \mathrm{tr}\left|\sum_a r_a(P_a\rho P_a^\dagger-\rho)\right| \\ &\leq \sum_{a\neq 0} r_a\|\mathcal{P}_a-\mathcal{I}\|_\diamond \\ &\leq 4\epsilon, \end{aligned} \tag{12}$$

where $\mathcal{P}_a(\rho)=P_a\rho P_a^\dagger$. The above reasoning shows the "only if" direction.

If the prover implements $\mathcal{E}_C=\mathcal{P}\circ\mathcal{C}$ which is δ-close to $\mathcal{C}$ in the computation round. Then in the X-test round, we have $\|\mathcal{P}(|0^n\rangle\langle 0^n|)-|0^n\rangle\langle 0^n|\|_{\mathrm{tr}} \leq \|\mathcal{E}_C-\mathcal{C}\|_\diamond \leq \delta$. Similarly, in the Z-test round, we have $\|\mathcal{P}(|+^n\rangle\langle +^n|)-|+^n\rangle\langle +^n|\|_{\mathrm{tr}} \leq \|\mathcal{E}_C-\mathcal{C}\|_\diamond \leq \delta$. Then we conclude that the success probability is at $1-\delta$ when a X-test or a Z-test round is chosen. This implies that the success probability in a test round is at least $1-\delta$. □

2.5 The Verifier-on-a-Leash protocol

The Broadbent protocol can be used to verify arbitrary quantum computation, but it requires the verifier to have the capability to perform measurements in the bases listed in Table 1. To achieve purely classical verification, Coladangelo, Grilo, Jeffery and Vidick presented a new protocol called the Verifier-on-a-Leash protocol (or the Leash protocol). In the Leash protocol, two provers called PP and PV share entanglement, but are not allowed to communicate with each other after the protocol starts. PP plays the role of the prover in the Broadbent protocol, and PV performs the measurements by the verifier in the Broadbent protocol. To certify PV plays honestly, a new rigidity test is introduced to verify Clifford measurements, in particular, the observables in $\Sigma=\{X,Y,Z,F,G\}$ where X,Y,Z are Pauli matrices, $F=\frac{1}{\sqrt{2}}(-X+Y)$ and $G=\frac{1}{\sqrt{2}}(X+Y)$. The idea is to use a non-local game for certifying the standard basis measurement (i.e., the observable X) and the Hadamard basis measurement (the observable Z). We will modify a test called $\mathsf{RIGID}'(\Sigma,m)$ which certifies Clifford measurements by Coladangelo, Grilo, Jeffery and Vidick [17] to only act on a subset of qubits. In particular, the following theorem holds with the test.

Theorem 2.15 ([17, Theorem 4]). *There exists a test $\mathsf{RIGID}'(\Sigma,m)$ such that the following holds: suppose a strategy for the players succeeds in test $\mathsf{RIGID}'(\Sigma,m)$ with probability at least $1-\epsilon$. Then for $D\in\{A,B\}$, there exists an isometry V_D such that*

$$\|(V_A\otimes V_B)|\psi\rangle_{AB}-|EPR\rangle_{A'B'}^{\otimes m}|AUX\rangle_{\hat{A}\hat{B}}\|^2 \leq O(\sqrt{\epsilon}), \tag{13}$$

and

$$\mathop{\mathbb{E}}_{W\in\Sigma^m}\sum_{u\in\{\pm\}^m}\left\|V_A\mathrm{tr}_B((\mathbb{1}_A\otimes W_B^u)|\psi\rangle\langle\psi|_{AB}(\mathbb{1}_A\otimes W_B^u))V_A^\dagger-\sum_{\lambda\in\{\pm\}}\left(\bigotimes_{i=1}^m\frac{\sigma_{W_i,\lambda}^{u_i}}{2}\otimes\tau_\lambda\right)\right\|_1=O(\mathrm{poly}(\epsilon)). \tag{14}$$

Moreover, players employing the honest strategy succeed with probability $1-e^{-\Omega(m)}$ in the test.

Theorem 2.15 shows that a strategy that succeeds in $\mathsf{RIGID}'(\Sigma, m)$ with probability at least $1-\epsilon$ must satisfy the following conditions: The players' joint state is $O(\sqrt{\epsilon})$-close to a tensor product of m EPR pairs together with an arbitrary ancilla register. Moreover, on average over uniformly chosen basis $W \leftarrow_R \Sigma^m$, the provers' measurement is poly(ϵ)-close to a probabilistic mixture of ideal measurements and their conjugates. More specifically, the probabilistic mixture can be realized as follows: performing a measurement on the ancilla register yields a post-measurement state τ_λ and a single bit outcome $\lambda \in \{\pm\}$ that specifies whether the ideal measurement associated with the observable $\sigma_{W_i,+}$ or that with its conjugate $\sigma_{W_i,-}$ is performed.[5] Then applying $\sigma_{W_i,\lambda}$ yields a post-measurement state $\sigma^{u_i}_{W_i,\lambda}$ and an outcome $u_i \in \{\pm\}$ for each index $i \in [m]$. We can only hope to certify that the strategy is close a probabilisitc mixture of the ideal strategy and its conjugate because a protocol with classical communication does not distinguish a strategy from its conjugate (more concretely, replacing $i = \sqrt{-1}$ with $-i$ for any strategy by the prover does not change the score).

In Section 6, we will modify $\mathsf{RIGID}'(\Sigma, m)$ to only apply on a random subset of indices in $[m]$, chosen by the verifier. It is clear that for the subset, Theorem 2.15 holds.

3 Certifying quantum depth from learning with errors

In this section, we describe a protocol for certifying quantum depth using NTCFs. First we define a single-prover protocol for certifying quantum depth.

Definition 3.1. *Let $d, d' \in \mathbb{N}$ and $d' > d$. A single-prover protocol $\mathsf{CVQD}(d, d')$ that separates quantum circuit depth d from d' consists of a classical verifier V and a prover P such that the following properties hold:*

- **Completeness:** *There exist an integer $\hat{d} \geq d$ and a $\hat{d}$-QC or $\hat{d}$-CQ scheme P such that* $\Pr[\langle V, P\rangle] \geq 2/3$.
- **Soundness:** *For integer $\hat{d} \leq d$ and any P that are $\hat{d}$-CQ or $\hat{d}$-QC schemes,* $\Pr[\langle V, P\rangle = accept] \leq 1/3$.

We give a prototocol based on a randomized encoding of NTCFs, and include the honest behavior of the prover in the description of the protocol (but the prover does not necessarily follow the instructions).

Protocol 1. $[\mathsf{CVQD}(d, d+d_0)]$

1. The verifier samples $\{(k_i, t_i) \leftarrow \textsc{Gen}(1^\lambda) : i \in [d+1]\}$ and sends $(k_1, \ldots, k_{d+1})$.
2. (Honest prover's behavior) The prover performs the quantum process in Theorem 2.11 and prepares the state

$$\frac{1}{(2|\mathcal{X}|)^{(d+1)/2}} \bigotimes_{i=1}^{d+1} \left(\sum_{b_i \in \{0,1\}, x_i \in \mathcal{X}} |b_i\rangle_{B_i} |x_i\rangle_{X_i} |\hat{f}_{k_i}(b_i, x_i)\rangle_{Y_i} \right), \tag{15}$$

and performs a standard basis measurement on the registers $Y_1, \ldots, Y_{d+1}$ to yield $y_1, \ldots, y_{d+1}$, which is sent to the verifier.

[5] The conjugate of a matrix A is obtained by taking the complex conjugate of each element of A.

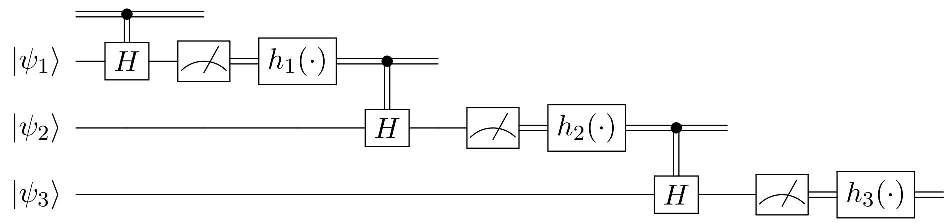

Figure 4: The quantum circuit for $d = 2$. In each layer $i \in [3]$, the verifier's action can be viewed as a classical computation h_i, which takes the measurement outcome as input. If the measurement outcome is accepted, then it outputs a random string $c_i \leftarrow_R \{0,1\}^{\ell(\lambda)}$; otherwise it outputs $\perp$ indicating rejection. Each $|\psi_i\rangle$ is the post-measurement state of (15) after a standard basis measurement on Y_1, Y_2, Y_3 is performed. By Theorem 2.11, the state $|\psi_1\rangle \otimes |\psi_2\rangle \otimes |\psi_3\rangle$ can be prepared in constant depth.

3. For $i = 1 \ldots d+1$, the verifier and the prover proceed as follows.
   (a) The verifier samples a random bit $c_i \in \{0,1\}$ and sends c_i to the verifier.
   (b) (Honest prover's behavior) If $c_i = 0$, the prover performs a standard basis measurement on B_iX_i; otherwise the prover performs a Hadamard basis measurement on B_iX_i. The prover then sends the outcome w_i to the verifier.
   (c) The verifier computes $a = V(t_i, y_i, c_i, w_i)$, where V is defined in (7). If $a = 0$, then the verifier rejects and aborts.

   If the verifier does not reject for each $i \in [d]$, then it accepts.

Theorem 3.2 (Completeness)**.** *There exists a constant d_0 such that for security parameter λ and polynomially bounded function d, a negligible function ϵ, there is a prover which is a $(d_0 + d(\lambda))$-QC scheme and succeeds with probability $1 - \epsilon(\lambda)$.*

Proof. By Theorem 2.11, the preparation of the state in (15) can be done in constant depth d'. Let $d_0 = d' + 1$. The prover performs standard basis measurement on the registers $Y_1, \ldots, Y_{d+1}$ to sample $y_1, \ldots, y_{d+1}$. For every $i \in [d+1]$, the prover measures the state in the i^{th} coordinate in the standard basis if $c_i = 0$ and in the Hadamard basis otherwise. There exists a negligible function μ such that if $c_i = 0$, with probability at least $1 - \mathrm{negl}(\lambda)$, performing a standard basis on B_iX_i yields a preimage; if $c_i = 1$, with probability at least $1 - \mu(\lambda)$, performing a Hadamard basis measurement on B_iX_i yields an outcome that passes the equation test $V(t_i, y_i, 1, \cdot)$. By the union bound, the prover succeeds with probability

$$\begin{aligned} \Pr[\text{success}] &\geq 1 - \sum_i \Pr[\text{Prover fails the } i^{th} \text{ round}] \\ &\geq 1 - d(\lambda) \cdot \mu(\lambda) = 1 - \mathrm{negl}(\lambda) \end{aligned} \tag{16}$$

for polynomially bounded function d. $\square$

As an example, we present the quantum circuit for $d = 2$ in Figure 4.

Next we show a lower bound on the quantum depth. In each round i, let the prover's action on receiving challenge $c_i = c$ be an isometry $U_{i,c}$ acting on the quantum state $|\psi_{i,T}\rangle$, which depends on the previous transcript T, followed by a standard basis measurement. We show that if there exists i

such that the quantum depth does not increase by 1 for round i, then there is a quantum adversary which breaks the adaptive hardcore bit property.

Theorem 3.3 (Soundness)**.** *There exists a negligible function μ such that for sufficiently large λ, for any prover that is either a d-CQ or d-QC scheme succeeds with probability at most $\frac{3}{4} + \mu(\lambda)$.*

Proof. Suppose toward contradiction that there exists a prover P which succeeds with probability $3/4 + \epsilon$ for non-negligible ϵ. First we show that the operation in each round must have non-zero quantum depth. If this is not the case for some round $i \in [d+1]$, then without loss of generality, P's operation consists of

1. a standard basis measurement on some intermediate quantum state ρ_i to yield an outcome v_i, followed by
2. a classical algorithm $\mathcal{A}_i$ which on input v_i and the challenge $c_i = c$, outputs the response $w_i \leftarrow \mathcal{A}_i(c_i, v_i, T)$ for previous transcript T.

Then there is a reduction $\mathcal{A}$ which uses P to break the adaptive hardcore bit property. Since the probability of passing d rounds is at least $3/4 + \epsilon(\lambda)$, the probability of winning the first i^{th} round is at least $3/4 + \epsilon$. Thus $\mathcal{A}$, on input challenge key k, samples $k_1, \ldots, k_{i-1}, k_{i+1}, \ldots, k_{d+1}$ and sets $k_i = k$ and computes v_i. Using v_i, $\mathcal{A}$ runs $\mathcal{A}_i$ on $w_b \leftarrow \mathcal{A}_i(b, v_i, T)$ for each $b \in \{0,1\}$. Let the probability that w_b is a valid response be p_b. By the assumption, $p_0 + p_1 \geq 3/2 + 2\epsilon$. This implies that with probability $1 - (1 - p_0) - (1 - p_1) = p_0 + p_1 - 1 \geq 1/2 + 2\epsilon$ *both* w_0 and w_1 are valid. Thus with probability at least $1/2 + 2\epsilon$, w_0 is a valid preimage and w_i is a valid equation, and thus the adaptive hardcore bit is broken.

Since P must have non-zero quantum depth in each round and it has total quantum depth d, there must exist a round $j \in [d+1]$ such that P must destroy all its coherence after receiving c_j, and continue answering the remaining rounds with an intermediate classical information σ_j. Now the reduction $\mathcal{A}'$ samples $c_1, \ldots, c_{d+1} \leftarrow_R \{0,1\}$ and simulates the verifier in the protocol, and runs the following steps to break the adaptive hardcore bit property:

1. $\mathcal{A}'$, on receiving k, samples $k_1, \ldots, k_d$ and sets $k_{d+1} = k$, and runs P to get $y_1, \ldots, y_{d+1}$ and some quantum information ρ.
2. $\mathcal{A}'$ continues running P on input ρ and $c_1, \ldots, c_j$ to compute σ_j.
3. $\mathcal{A}'$ continues running P on input $\sigma_j, c_j, \ldots, c_d, c_{d+1} = b$, to yield a response w'_b in round d for each $b \in \{0,1\}$.
4. $\mathcal{A}'$ outputs (y_{d+1}, w'_0, w'_1) as the response.

To analyze the performance of $\mathcal{A}'$, we apply the same idea as that for calculating the performance of $\mathcal{A}$. Let the probability that w'_b be a valid response for round d be p'_d. By the assumption, $p'_0 + p'_1 \geq 3/2 + 2\epsilon$. Thus both w'_0 and w'_1 are valid with probability at least $1 - (1 - p'_0) - (1 - p'_1) \geq 1/2 + 2\epsilon$. This implies that the adaptive hardcore bit property is broken. ☐

Theorem 3.2 and Theorem 3.3 immediately imply the following theorem.

Theorem 3.4. *Assuming that LWE is hard for any d-CQ and d-QC schemes, Protocol 1 satisfies the following conditions.*

- **Completeness:** *There exists a prover which is a $(d_0 + d)$-QC scheme and succeeds with probability at least* $1 - \text{negl}(\lambda)$.
- **Soundness:** *Every prover that are d-CQ and d-QC schemes succeed with probability at most* $\frac{3}{4} + \text{negl}(\lambda)$.

By sequential repetition, the completeness-soundness gap can be amplified to $1 - \text{negl}(\lambda)$. We note that we do not know if the function evaluation can be done by using $(d_0 + d)$-CQ schemes and leave it as an open question.

4 Non-interactive classical verification of quantum depth

In this section, we prove that there is a non-interactive classical verification of quantum depth, in the quantum-accessible random oracle model. Recall that in Theorem 3.3, we have shown that there exists a verification protocol with constant soundness error for the same purpose. By parallel repetition, in Section 4.1, we prove that the soundness error can be amplified to negligible. To justify our claim, we follow the same proof idea as in the previous section: First we show that in a four-message protocol, if the side information is restricted to classical, any quantum adversary can only succeed in a parallel repetition of the proof-of-quantumness protocol with negligible probability. Then, in a multi-round protocol, if the quantum device does not have sufficient quantum depth, then it must destroy its coherence to yield a purely classical intermediate state. If it further succeeds in the protocol with non-negligible probability, we can construct a reduction that violates the negligible soundness error in a four-message protocol. This implies the soundness of a multi-round verification, with round complexity dependent on the depth threshold.

The multi-round protocol has nice structure, allowing us to achieve round reduction by Fiat-Shamir: Recall that Don, Fehr, and Majenz showed that in the quantum random oracle model, constant-round Σ-protocol (i.e., computationally sound public-coin protocols) can compressed to a single-message one by Fiat-Shamir tranform [20]. In our protocol, the verifier's moves are tosses of public coins except that in the first round, it generates a key pair and keeps the private key secret. This protocol can be characterized as a generalized Σ-protocol [2]. Applying the extension to generalized Σ-protocols shown independently by Alagic, Childs, Grilo, and Hung [2] and Chia, Chung, and Lai [13], in the random oracle model, the interaction can be reduced to two messages. In Section 4.2, we summarize our proof ideas and prove the soundness of the non-interactive protocol.

4.1 Interactive verification with negligible soundness error

We start with a four message protocol in the following characterization: Let the prover in the first move and the second denote $\mathcal{P}_1$ and $\mathcal{P}_2$, respectively. Initially, $\mathcal{P}_1$, on receiving the message from the verifier, outputs its first message y to the verifier, and a *classical* side information σ to $\mathcal{P}_2$. Then $\mathcal{P}_2$, on receiving the challenge coin c from the verifier and the side information σ from $\mathcal{P}_1$, outputs the second message z. Finally the verifier outputs a bit indicating acceptance or rejection on input (k, t, y, c, z). Now we show that when the side information σ is classical, by the adaptive hardcore bit property, every prover must succeed with probability $3/4 + \text{negl}(\lambda)$.

Theorem 4.1. *Every efficient $\mathcal{P}_1$ and $\mathcal{P}_2$ succeed with probability* $3/4 + \text{negl}(\lambda)$.

Proof. The proof is similar to the soundness proof in the previous section. Let the success probability of the prover on $c \in \{0, 1\}$ be p_c. If $p_0 + p_1 \geq 3/2 + \epsilon(\lambda)$ for some non-negligible function ϵ, we construct a reduction that breaks the adaptive hardcore bit property: The reduction $\mathcal{B}$ first runs $\mathcal{P}_1$ to yield y and a classical side information σ. Then using the same side information, the prover runs $\mathcal{P}_2(\sigma, c = 0)$ and $\mathcal{P}_2(\sigma, c = 1)$ to yield two messages z_0 and z_1 respectively. By the assumption we make,

$$\Pr_{k,t,\sigma}[V_0 \text{ accepts } \mathcal{P}_2(\sigma, 0) \wedge V_1 \text{ accepts } \mathcal{P}_2(\sigma, 1)] \geq p_0 + p_1 - 1 \geq 1/2 + \epsilon(\lambda). \tag{17}$$

This implies that the reduction breaks the adaptive hardcore bit property. □

The above reasoning only uses the fact that classical information is clonable. Thus the same argument would also apply if σ is a clonable quantum state.

The protocol has a soundness error $3/4+\text{negl}(\lambda)$. To reduce the error, we apply parallel repetition to get negligible soundness error. Our direct product theorem is closely related to the approaches by a series of works by Håstad, Pass, Wikström, and Pietrzak [24], by Haitner [22], and by Berman, Haitner, and Tsfadia [7]. The main caveat we will address here is that the prover's algorithms in each move is a quantum algorithm (as opposed to deterministic ones).

Theorem 4.2. *For $m = \omega(\log \lambda)$, let $\mathcal{P}^m = (\mathcal{P}_1^m, \mathcal{P}_2^m)$ be any quantum prover in an m-fold parallel repetition of the protocol such that the side information from $\mathcal{P}_1^m$ to $\mathcal{P}_2^m$ is classical. Then $\mathcal{P}^m$ wins the protocol with probability at most* $\text{negl}(\lambda)$.

We consider the following reduction $\mathcal{P}^*$ that works in a single-copy protocol: $\mathcal{P}^*$ samples a random coordinate $i \in [m]$, and embeds the external verifier as the ith verifier in the simulated m-fold parallel repetition protocol. Furthermore, $\mathcal{P}^*$ simulates the other $m-1$ verifiers. To generate the first move of the prover, $\mathcal{P}^*$ first generates the simulated verifiers' key pairs for $[-i] := \{j \in [m] : j \neq i\}$ using the following strategy: First $\mathcal{P}^*(k_i)$ sets $c_i = 0$ and repeat the following steps for at most w times, where w is to be determined later, or $b = 1$:

1. Sample (k_{-i}, t_{-i}) and run $\mathcal{P}_1^m(k)$, which outputs (y, σ).
2. Sample c_{-i}, run $z \leftarrow \mathcal{P}_2^m(y, c, \sigma)$, and compute $b \leftarrow \mathcal{V}^m(k, t, y, c, z)$.

Since for $c_i = 0$, the verifier's final verdict is simulable, and the decision can be computed. The rejection sampling above is used to sample good key pairs (k_{-i}, t_{-i}) and y. Upon success, $\mathcal{P}^*$ sends y_i to the external verifier.

In the second move, $\mathcal{P}^*$ on receiving c_i from the external verifier, runs the following rejection sampling process for at most w times or $b = 1$:

1. Sample c_{-i} and run $z \leftarrow \mathcal{P}_2^m(y, \sigma, c)$.
2. $\mathcal{P}^*$ checks if z_{-i} are accepted by the simulated verifier by running $b_{-i} \leftarrow \mathcal{V}^{m-1}(k_{-i}, t_{-i}, y_{-i}, c_{-i}, z_{-i})$. If so, the verifier terminates and sends z_i to the external verifier.

For strings with lowercase labels, let their uppercase labels denote the random variable (as opposed to samples). Our goal is to show that

$$(K, T, Y, \Sigma, C, Z)_{real} \approx (K, T, Y, \Sigma, C, Z)_{ideal}, \tag{18}$$

where the real distribution can be sampled from the above process, and the ideal case is obtained by simulating all m verifiers and rejection sampling all the transcripts for infinitely many times until success. To this end, we rely on the lemmas provided in Haitner's paper. Let W denote the event of success, i.e., $b = 1$, and $\alpha := \sqrt{\frac{\log(1/\Pr[W])}{m}}$.

Lemma 4.3. *Let $Y, X_1, \ldots, X_m$ be independent random variables over some probability spaces and W be an event in the same subspace. Then*

1. $\Pr_{i\sim[m],x_i\leftarrow X_i}[\Pr[W|X_i = x_i] \notin (1 \pm \epsilon \Pr[W])] \leq 2\alpha/\epsilon$ *for* $\epsilon > 0$,
2. $\mathbb{E}_{i\leftarrow[m]}[SD((i,(\bar{X}|W)),(i,(X|W,X_i)))] \leq \alpha$ *where* $\bar{X} = (X_1, \ldots, X_m)$, *and*
3. $\mathbb{E}_{i\leftarrow[m]}[SD((i,(Y,X_i|W)),(i,(Y|W),X_i))] \leq \alpha$.

We will be using the following lemma.

Lemma 4.4. *Let X_0, X_1 be two random variables over the same finite set $\mathcal{X}$ and W be an event. If $\Pr[X_0 \in W] = \delta$, then*

$$SD(X_0|W, X_1|W) \leq \frac{2 \cdot SD(X_1, X_2)}{\Pr[X_0 \in W]}. \tag{19}$$

We include a proof for completeness.

Proof. Let p_b be the distribution of X_b for $b \in \{0,1\}$ and $p_{b|W}$ denote the conditional distribution. By definition,

$$SD(X_0|W, X_1|W) = \frac{1}{2}\sum_{x\in W}|p_{0|W}(x) - p_{1|W}(x)| = \frac{1}{2}\cdot \mathop{\mathbb{E}}_{x\sim p_{0|W}}[|1 - F(X)|] \tag{20}$$

where $F(x) := \frac{p_{1|W}(x)}{p_{0|W}(x)} \cdot \mathbb{1}[p_{0|W}(x) > 0]$ denote a random variable for X distributed according to $p_{0|W}$. By definition, $\mathbb{E}_{x\sim p_{0|W}}[F(x)] = 1$, and thus the expectation

$$\mathop{\mathbb{E}}_{x\sim p_{0|W}}[|1 - F(x)|] \leq 2 \cdot \mathbb{E}\left[\left|1 - \frac{\Pr[X_1 \in W]}{\Pr[X_0 \in W]} \cdot F(x)\right|\right] \tag{21}$$

$$\leq \frac{2}{\Pr[X_0 \in W]}\sum_{x\in W}|p_0(x) - p_1(x)| \tag{22}$$

$$\leq \frac{4 \cdot SD(X_0, X_1)}{\Pr[X_0 \in W]}. \tag{23}$$

The first inequality comes from the fact that for non-negative random variable Z satisfying $E[Z] = 1$, then $\mathbb{E}[1 - Z] \leq \mathbb{E}[|1 - sZ|]$ for every $s > 0$ (see Lemma 5 of [24] for a detailed proof of this fact). ☐

Let's consider a variant of real, denoted *real′*, in which the rejection sampling processes can repeat infinite times, i.e., w is set to infinity. In the following analysis, we first show the closeness between *real′* and *ideal*, and we will handle the difference between *real* and *real′* later.

4.1.1 Closeness between *real′* and *ideal*

In the first step, our goal is to show that

$$I, (K, T, Y, \Sigma)|(W, K_I, T_I, C_I = 1) \approx_{O(\alpha)} I, (K, T, Y, \Sigma)|W. \tag{24}$$

That is, sampling the internal key pairs and the output of $\mathcal{P}_1^m$ conditioned on the event of acceptance, the external key pair embedded into the Ith coordinate for uniform I and the random challenge $C_I = 1$ is δ-close to simulating all key pairs and the output of $\mathcal{P}_1^m$.

Lemma 4.5. *For $\delta \in [0,1]$, let $C_1, \ldots, C_m$ be identical and independent binary random variables such that $\Pr[C_i = 1] = \delta$. For $\alpha = \frac{1}{\delta}\sqrt{\frac{\log(1/\Pr[W])}{m}}$, $(I, (K, T, Y, \Sigma)|(W, K_I, T_I, C_I = 1))$ and $I, (K, T, Y, \Sigma)|W$ are $O(\alpha)$-close in total variation distance.*

Proof. We consider a hybrid in which the reduction gets to choose the Ith key pair. In this case, the marginal distribution of the Ith key pair is no longer uniform. Our first step is to show that

$$I, (K, T, Y, \Sigma)|(W, C_I = 1) \approx_{2\alpha} I, (K, T, Y, \Sigma)|W \tag{25}$$

Let the distance be D_1. Since in each world, the random variables (Y, Σ) are output by an efficient quantum algorithm (prior to the conditioning on W), we can approximate the distribution to exponential precision using a randomized algorithm given access to exponentially many random bits and running in exponential time. Thus let $(Y, \Sigma) = F(K, R)$ for internal randomness R and F be the function describing the approximation. By the data-processing inequality,

$$\begin{aligned} D_1 &\leq SD((I, (K, T, R, Y, \Sigma)|(W, C_I = 1)), (I, (K, T, R, Y, \Sigma)|W)) \\ &= SD((I, (K, T, R)|(W, C_I = 1)), (I, (K, T, R)|W)). \end{aligned} \tag{26}$$

Now we notice that by the third inequality of Lemma 4.3,

$$SD((I, (K, T, R, C_I)|W), (I, (K, T, R)|W, C_I)) \leq \sqrt{\frac{\log(1/\Pr[W])}{m}}. \tag{27}$$

Conditioning both sides on the event that $C_I = 1$, the left side becomes $(I, (K, T, R)|(W, C_I = 1))$ and the right side becomes $(I, (K, T, R)|W)$, and by Lemma 4.4 and (26), we have

$$D_1 \leq SD((I, (K, T, R)|(W, C_I = 1)), (I, (K, T, R)|W)) \leq \frac{2}{\delta}\sqrt{\frac{\log(1/\Pr[W])}{m}}. \tag{28}$$

In the second step, we aim to bound the distance

$$D_2 := SD((I, (K, T, Y, \Sigma)|(W, K_I, T_I, C_I = 1)), (I, (K, T, Y, \Sigma)|(W, C_I = 1))). \tag{29}$$

Applying the same idea to remove the dependence of Y, Σ, we have

$$D_2 \leq SD((I, (K, T, R)|(W, K_I, T_I, C_I = 1)), (I, (K, T, R)|(W, C_I = 1))) \tag{30}$$

$$\leq SD((I, (K, T, R, C)|(W, K_I, T_I, C_I = 1)), (I, (K, T, R, C)|(W, C_I = 1))). \tag{31}$$

When on each side, C_I is sampled according to the original distribution, it holds that

$$SD((I, (K, T, R, C)|(W, K_I, T_I, C_I)), (I, (K, T, R, C)|(W, C_I))) \leq \sqrt{\frac{\log(1/\Pr[W])}{m}}. \tag{32}$$

Then conditioning both sides on $C_I = 1$, we have $D_2 \leq 2\alpha$. Finally we conclude the proof by triangle inequality of trace distance and $D_1 + D_2 \leq 4\alpha$. ☐

In summary for the first step, we have explained that the distance between *ideal* and *real'* is $O(\alpha)$-close in total variation distance.

4.1.2 Closeness between *real* and *real'*

Now we show that when $w = \frac{c_0}{\epsilon\alpha}$ for a sufficiently large constant c_0 to be determined later, *real* and *real'* are $O(\alpha)$-close in total variation distance. The idea follows from [22].

We consider the hybrid game, denoted *hybrid*, as follows. In the first step, the reduction samples the first message via rejection sampling for infinite times, and in the second step, the rejection sampling has at most w iterations. To clarify, let (w_1, w_2) denote the game in which the first step rejection samples w_1 times and the second step rejection sample w_2 times. By the definition of each game introduced above, we have $real' = (\infty, \infty)$, $hybrid = (\infty, w)$ and $real = (w, w)$.

To see why $(\infty, \infty) \approx_{O(\alpha)} (\infty, w)$ for sufficiently large w, note the marginal distributions of (K, T, Y, Σ) in both games are identical. For a fixed tuple $\pi_1 = (K = k, T = t, Y = y, \Sigma = \sigma)$, let the success probability of $\mathcal{P}_2^m$ be $\epsilon(\pi_1)$. Consider the following sampling process over two random variables (Π_1, Π_2) where $\Pi_1 := (K, T, Y, \Sigma)$ and $\Pi_2 := (C, Z)$. We have the following lemma.

Lemma 4.6 ([22, Proposition 2.5])**.** $\mathbb{E}_{\pi_1 \sim \Pi_1 | W}\left[\frac{1}{\epsilon(\pi_1)}\right] = \frac{1}{\epsilon}$, *where π_1 is obtained by infinite rejection sampling.*

Lemma 4.7. *For every reduction that succeeds in real' with probability p, in hybrid it succeeds with probability at least $p - 2\alpha$.*

Proof. The idea is as follows: By Lemma 4.6 and Markov's inequality, $\frac{c_0}{\epsilon(\pi_1)} \leq \frac{c_0}{\epsilon\alpha}$ with probability at most α. For each π_1 such that $\frac{c_0}{\epsilon(\pi_1)} \leq w = \frac{c_0}{\epsilon\alpha}$, w iterations are sufficient for *real'* to terminate except with probability at most $(1 - \epsilon(\pi_1))^w$. Thus overall $'$ does not terminate with probability at most $\mathbb{E}_{\pi_1}[(1 - \epsilon(\pi_1))^w] \leq \alpha + (1 - \alpha)e^{-c_0/\alpha} \leq 2\alpha$ for sufficiently large constant c_0. Then for every adversary which succeeds with probability p in *real'*, its variant in *hybrid* succeeds with probability at least $p - 2\alpha$. $\square$

The following lemma holds by applying the same proof idea for Lemma 4.7 and setting π_1 to empty.

Lemma 4.8. *For every reduction that succeeds in hybrid with probability p, in real it succeeds with probability at least $p - 2\alpha$.*

By Lemma 4.7 and Lemma 4.8, we conclude this section with the following corollary.

Corollary 4.9. *For every reduction that succeeds in real' with probability p, in real it succeeds with probability probability at least $p - 4\alpha$.*

4.1.3 The parallel repetition theorem

In this section, we complete the proof of Theorem 4.2.

Proof of Theorem 4.2. In Lemma 4.5, we have shown that *ideal* and *real'* are $O(\alpha)$-close in total variation distance. Since in *ideal*, the reduction succeeds with probability 1, the reduction in *real'* succeeds with probability at least $1 - O(\alpha)$. Then by Corollary 4.9, the reduction in *real* succeeds with probability $1 - O(\alpha)$. Now let c_0 be a constant such that the reduction in *real* succeeds with probability at least $1 - c_0\alpha$. By Theorem 4.1, $1 - c_0\alpha \leq 3/4 + \text{negl}(\lambda) \leq 7/8$. This implies that $\Pr[W] \leq \exp\left(-\frac{1}{64c_0^2}m\right) = 2^{-\Omega(m)} = \text{negl}(\lambda)$ for $m = \omega(\log\lambda)$. $\square$

In the proof, we apply the argument that any quantum polynomial-time algorithm with classical inputs and outputs can be simulated to exponential precision in total variation distance by a classical algorithm running in exponential time. To justify the validity, abstractly, we can view our bound on the total variation distance as two classical algorithms $\mathcal{A}, \mathcal{B}$ given access to a quantum algorithm $\mathcal{O}$ which takes classical input and output classical strings. We bound the distance $D = SD(\mathcal{A}^{\mathcal{O}}, \mathcal{B}^{\mathcal{O}})$ by considering a classical exponential-time simulator $\mathcal{S}$ which simulates $\mathcal{O}$ to exponential precision in total variation distance. Then we show that $SD(\mathcal{A}^{\mathcal{S}}, \mathcal{B}^{\mathcal{S}})$ has a desired upper bound, which in turn implies that D has the same upper bound up to an exponentially small additive loss.

We emphasize that this argument has its limitations. In particular, the argument does not extend to prove a direct product theorem when the side information from $\mathcal{P}_1$ to $\mathcal{P}_2$ is a (unclonable) quantum state. We note that the constraint on the prover is sufficient for the problem we study here (i.e., the soundness of a CVQD protocol).

4.1.4 An interactive protocol with negligible soundness error

We formally describe our protocol with negligible completeness and soundness errors in Protocol 2.

Protocol 2. [$\mathsf{CVQD}(d, d+d_0)$]

1. The verifier samples $\{(k_{ij}, t_{ij}) \leftarrow \textsc{Gen}(1^\lambda) : i \in [d+1], j \in [m]\}$, and sends $k = (k_{ij} : i \in [d+1], j \in [m])$ to the prover.
2. (Honest prover's behavior) The prover performs the quantum process in Theorem 2.11 and prepares the state

$$\frac{1}{(2|\mathcal{X}|)^{(d+1)m/2}} \bigotimes_{i=1}^{d+1} \left(\sum_{b_i \in \{0,1\}^m, x_i \in \mathcal{X}^m} |b_i\rangle_{B_i} |x_i\rangle_{X_i} |\hat{f}_{k_i}(b_i, x_i)\rangle_{Y_i} \right), \tag{33}$$

   and performs a standard basis measurement on the registers $Y_1, \ldots, Y_{d+1}$ to yield $y_1, \ldots, y_{d+1} \in \mathcal{Y}^m$, which is sent to the verifier. Here $\hat{f}_{k_i}(b_i, x_i) := (\hat{f}_{k_{ij}}(b_{ij}, x_{ij}) : j \in [m])$ is the function obtained by querying $\hat{f}_{k_{ij}}$ for $j \in [m]$ in parallel. For each i, $B_i := \bigotimes_{j=1}^m B_{ij}$, and X_i and Y_i are defined similarly.
3. For $i = 1 \ldots d+1$, the verifier and the prover proceed as follows.
   (a) The verifier samples a random bit $c_i \in \{0,1\}^m$ and sends c_i to the verifier.
   (b) (Honest prover's behavior) For each $j \in [m]$, if $c_{ij} = 0$, the honest prover performs a standard basis measurement on $B_{ij}X_{ij}$; otherwise the prover performs a Hadamard basis measurement on $B_{ij}X_{ij}$. The prover then sends the outcome $w_i = (w_{ij} : j \in [m])$ to the verifier.
   (c) The verifier computes $a := \prod_{j=1}^m V(t_{ij}, y_{ij}, c_{ij}, w_{ij})$, where V is defined in (7). If $a = 0$, then the verifier rejects and aborts.

   If the verifier does not reject for each $i \in [d]$, then it accepts.

By the same reasoning as in the proof of Theorem 3.2, the protocol has negligible completeness error. We state the theorem and omit the proof.

Theorem 4.10 (Completeness)**.** *There exists a constant d_0 such that for security parameter λ and polynomially bounded function d, a negligible function ϵ, there is a prover which is a $(d_0 + d(\lambda))$-QC scheme and succeeds with probability $1 - \epsilon(\lambda)$.*

This section is devoted to proving that the protocol has negligible soundness error.

Theorem 4.11 (Soundness)**.** *Assuming that LWE is hard for d-CQ and d-QC schemes, every depth-$(d + d_f - 1)$ prover succeeds in Protocol 2 with probability* $\text{negl}(\lambda)$*,*

Proof. Suppose toward contradiction that there exists a prover P which succeeds with non-negligible probability ϵ. Since the prover has depth $d + d_f - 1$, there exists an index $i \in [d+1]$ such that P's operation consists of

1. a quantum operation depending on ith challenge string $c_i \in \{0,1\}^m$ followed by a standard basis measurement to yield a classical state σ_{c_i},
2. a quantum algorithm that takes σ_{c_i} and the previous transcript as input to output a quantum state, forwarded as in input to the prover's algorithm in $(i+1)$st round.

Without loss of generality, let σ denote the joint classical state (c_i, σ_{c_i}) for uniform c_i.

We now describe the reduction. In a high level, the reduction simulates the verifier's behavior for the first d rounds, and embed the external's challenge string in the $(d+1)$st round. More specifically, the reduction, on input the public key k, samples the key pairs (k_i, t_i) for $i \in [d]$ and sets $k_{d+1} = k$. It then simulates $P(k_1, \ldots, k_{d+1})$ to yield $y = (y_{ij} : i \in [d+1], j \in [m])$ and forwards y_{d+1}. Then it "fast forwards" P to the ith round to yield a classical side information σ. Now upon finishing the first move, the reduction is left with a classical side information σ.

In the second move, the reduction takes as input the challenge coin c and the classical state σ. The reduction continues running P to the $(i+1)$st round, and forwards c to P. By the assumption, the prover yields a valid transcript with non-negligible probability ϵ and violates Theorem 4.2. □

4.2 Non-interactive classical verification of quantum depth

Applying multi-round Fiat-Shamir transform, we prove the following theorem.

Theorem 4.12. *Assuming LWE is hard, in the random oracle model, every depth-$(d+d_f-1)$ prover succeeds in the 2-message protocol with probability* $\text{negl}(\lambda)$*.*

Proof. Let m be large enough such that the m-parallel repetition of the d-round CVQD protocol has negligible soundness as shown in Theorem 4.11. For simplicity, we use V_i and P_i to denote the algorithms that the verifier and prover apply in each round i and V_{final} to denote the verifier's algorithm when receiving the last message from the prover. Consider a hash function $H : [d+1] \times \mathcal{X} \times \mathcal{W} \to \{0,1\}^m$ idealized as a random oracle, where $\mathcal{W}$ is the space of the message sent from the prover and $\mathcal{X}$ is the space of the message sent from the verifier in Protocol 2. We describe the two-round protocol as follows:

Protocol 3. $[\mathsf{CVQD}(d, d + d_0)]$

1. The verifier applies $\textsc{Gen}(1^\lambda)$ to generate $k = \{k_{ij} : i \in [d+1], j \in [m]\}$ and $t = \{t_{ij} : i \in [d+1], j \in [m]\}$ and sends k to the prover.
2. The prover applies P_0 on input k to generate $y = \{y_{ij} : i \in [d+1], j \in [m]\}$ and some internal side information σ_0. Then, the prover sets $c_1 = H(1, k, y)$ and runs P_1 on (c_1, y, k, σ)

to generate $w_1 = \{w_{11}, \ldots, w_{1m}\}$. For $i = 2, \ldots, d+1$, the prover sets $c_i = H(i, c_{i-1}, w_{i-1})$ and applies P_i on $(c_i, y, k, \sigma_{i-1})$ to generate w_i where $w_i := (w_{i1}, w_{i2}, \ldots, w_{im})$. Finally, the prover sends (y, w) to the verifier, where $w := (w_1, \ldots, w_{d+1})$.

3. The verifier computes $c := c_1, \ldots, c_{d+1}$ from (y, w) and H, runs V_{final} on (k, t, y, c, w), and output whatever V_{final} outputs.

Completeness. The completeness of the protocol directly follows from the completeness of Protocol 2. One caveat is that an honest prover can use the classical machine to query H to generate the random challenge c, which does not increase the quantum depth.

Soundness. To prove the soundness, we reduce the task of breaking the soundness of Protocol 2 to breaking the non-interactive protocol via quantum Fiat-Shamir transformation. We use the measure-and-reprogram lemma with enforced extraction order in [19].

Lemma 4.13 (Adapted from [19, Theorem 7])**.** *Let m be a positive integer, and let $\mathcal{X}_0$, $\mathcal{X}$ and $\mathcal{Y}$ be finite non-empty sets. There exists a black-box polynomial-time $(m+1)$-stage algorithm S, satisfying the following property: Let $\mathcal{A}$ be an arbitrary oracle quantum algorithm making q queries to a uniformly random $H : (\mathcal{X}_0 \cup \mathcal{Y}) \times \mathcal{X} \to \mathcal{Y}$ and that outputs a tuple $x = \{x_0, \ldots, x_d\} \in \{\mathcal{X}_0 \times \mathcal{X}^d\}$ and z such that for any x^o without duplicate entries and for any predicate V,*

$$\begin{aligned}&\Pr_c[x = x^o \wedge V(x, c, z) : (x, z) \leftarrow (S^{\mathcal{A}}, c)]\\&\geq \left(\frac{d!}{2q+d+1}\right)^{2d} \Pr_H\left[x = x^o \wedge V(x, \mathbf{h}^{H,x}, z) : x, z \leftarrow \mathcal{A}^H\right] - \epsilon_{x^o}.\end{aligned}$$

Here, ϵ_{x^o} is equal to $\frac{m!}{|\mathcal{Y}|}$ when summed over all x^o and $(x, z) \leftarrow (S^{\mathcal{A}}, c)$ means that $S^{\mathcal{A}}$ outputs the message (x, z) on inputs c. $\mathbf{h}^{H,x} := (h_1^{H,x}, \ldots, h_d^{H,x})$ is a hash chain with respect to H and x, where $h_1^{H,x} = H(x_0, x_1)$ and $h_i^{H,x} = H(h_{i-1}^{H,x}, x_i)$ for $i = 2, \ldots, d$.

To break the soundness of Protocol 2, we suppose that there is a d-depth quantum polynomial-time algorithm $\mathcal{A}$ that breaks the two-round protocol. Then, by an averaging argument over the key pairs, we apply Lemma 4.13 to construct a simulator $S^{\mathcal{A}}$ with respect to the verifier such that

$$\Pr_{c,k,t}\left[V_{final}(k, t, y, c, w) : (y, w) \leftarrow (S^{\mathcal{A}}, c)\right] \tag{34}$$

$$\geq \left(\frac{(d+1)!}{2q+d+2}\right)^{2(d+1)} \Pr_{H,k,t}\left[V_{final}(k, t, y, c_H, w) : y, w \leftarrow \mathcal{A}^H\right] - \frac{(d+1)!}{2^m}, \tag{35}$$

where $c_H = (c_1, \ldots, c_{d+1})$ with $c_1 = H(1, k, y)$ and $c_i = H(i, c_{i-1}, w_{i-1})$ for $i = 2, \ldots, d+1$.

Note that the inputs to H must have no duplicated entries as required by Lemma 4.13 since the message index i is included in the input of the H. Furthermore, $S^{\mathcal{A}}$ has the same quantum depth as $\mathcal{A}$ since $S^{\mathcal{A}}$ follows $\mathcal{A}$ except that it randomly chooses $[d+1]$ queries of $\mathcal{A}$ and "measure-and-reprogram" the oracle H after these queries, which does not increase the quantum circuit depth. Therefore, when $\mathcal{A}$ wins with noticeable probability, $d = O(1)$, and $q = \mathrm{poly}(n)$, $\Pr_{c,k,t}[V_{final}(k, t, y, c, w) : (y, w) \leftarrow (S^{\mathcal{A}}, c)]$ is also noticeable following (35).

Then, we can construct an adversary $\mathcal{A}'$ for Protocol 2 as follows: a) Given k, $\mathcal{A}'$ runs the first part of $S^{\mathcal{A}}$ to obtain y and send y to the verifier. b) For each round $i = 1, \ldots, d+1$, on input c_i and the side information of previous round, $\mathcal{A}'$ applies $S^{\mathcal{A}}$ to obtain w_i and sends w_i to the verifier.

Obviously, the probability that $\mathcal{A}'$ wins is noticeable following from the fact that $\Pr_c[V_{final}(k, t, y, c, w) : (y, w) \leftarrow (S^{\mathcal{A}}, c)]$ is noticeable. This contradicts the soundness of Protocol 2. Hence, there does not exist a d-CQ or d-QC algorithm $\mathcal{A}$ that succeeds in the two-round protocol with noticeable probability in the random oracle model. $\square$

Finally, we can obtain a $\mathsf{CVQD}(d, d + d_0)$ in the pain model under QLWE assumption and the random oracle heuristic.

Corollary 4.14. *Assuming LWE is hard and the random oracle heuristic, for constants d and d_f, every depth-$(d + d_f - 1)$ prover succeeds in the 2-message protocol with probability* $\mathrm{negl}(\lambda)$.

Proof. This corollary mainly follows from Theorem 4.12. It remains to show that an honest prover can win with $1 - \mathrm{negl}(\lambda)$ probability without increasing the quantum circuit depth when initiating the random oracle by hash functions. Since the two-message protocol only requires an honest prover to use the random oracle to generate the classical challenge c, the prover only needs to use a classical machine to query the oracle, which does not increase any quantum depth. This also holds when we instantiate the random oracle by hash functions. $\square$

While our protocol is private-coin, Lemma 4.13 can be applied since for any fixed key pair (k, t), we can view the verifier and the prover running a protocol in which each move of the verifier is a public coin toss. The protocol is not a public-coin protocol since the final verdict V_{final} cannot be computed by the prover without access to t. But this is not a problem for applying Lemma 4.13 since running the simulator $S^{\mathcal{A}}$ does not require information about t. Instead, we only use the fact that each verifier's move is a coin toss. This observation has been made independently by Alagic, Childs, Grilo, and Hung [2] and by Chia, Chung, and Yamakawa [13] for the round reduction of classical verification of quantum computation. Here we apply the same idea to our multi-round protocol.

5 Certifying a query algorithm

A q-query algorithm $\mathcal{A}$ given access to O, denoted $\mathcal{A}^O$, without loss of generality, consists of a sequence of unitary maps $U_q O \ldots U_1 O U_0$ acting on the zero state, followed by a standard basis measurement to extract the information about the oracle. In the real world, information is transmitted in a non-black-box way, and therefore one must instantiate the oracle with an efficient quantum process. However, any instantiation in the plain model would in general leak the information about the oracle, even based on computational assumptions [6]. Thus, the analysis or the conclusions obtained in a query model does not apply when the oracle is replaced with an instantiation.

On the other hand, the impossibility result does not apply when the oracle is implemented by an external device to which the algorithm has limited access. In this setting, another device, denoted P_O, is added to implement the oracle O, and the secret encoded in the oracle may be transparent to P_O. An equivalent two-player protocol that mimics the query computation can be described as the following protocol:

1. A player P_A performs U_0 on the zero state, and teleports the state to the other player P_O.
2. P_O performs O on the quantum state and teleport the output state back.
3. $\mathcal{A}$ performs U_1 on the input state and teleports the state.

4. Repeat the last two steps $(q-1)$ times with unitaries $U_2, \ldots, U_q$ and outputs w.

5. The verifier accepts if the oracle O and w satisfies some relation R and rejects otherwise.

When the players behave honestly, i.e., P_O performs O in each iteration, and P_A performs $U_0, \ldots, U_q$ in order, the verifier accepts with probability exactly the same as the query algorithm $\mathcal{A}$ solving R. When neither of the players is trusted, we ask the question: given a query algorithm $\mathcal{A}$, can we construct an equivalent protocol against untrusted players? To answer the question, we specify the classes of problems which we consider in this work, and the equivalence between a query algorithm and a two-player protocol.

Oracle separation problems. For quantum complexity classes $\mathcal{C}_{yes}, \mathcal{C}_{no}$, an oracle separation problem R is a relation between the oracle O (drawn from a public distribution) and a binary string w such that

- there is a quantum query algorithm $\mathcal{A}$ that is given access to O and runs in $\mathcal{C}_{yes}$, outputs w such that $R(O, w) = 1$ with probability at least p, and
- no quantum algorithm that is given access to O and runs in $\mathcal{C}_{no}$, outputs w such that $R(O, w) = 1$ with probability at most p'.

Here, the probabilities p and p' are defined by $\mathcal{C}_{yes}$ and $\mathcal{C}_{no}$, e.g., if $\mathcal{C}_{yes} = \mathsf{PP}$ and $\mathcal{C}_{no} = \mathsf{BPP}$, $p = 1/2$ and $p' = 1/3$.

Two-player protocol for separation. A two-player protocol distinguishes $\mathcal{C}_{yes}$[6] from $\mathcal{C}_{no}$ if there exists a classical verifier V interacting with two provers P_O and P_A such that the following conditions hold.

- **Non-locality.** P_O and P_A share an arbitrary quantum state and there is no quantum and classical channel between once the protocol starts.
- **Completeness.** There exists P_O and P_A that runs in $\mathcal{C}_{yes}$ such that $\Pr[\langle V, P_O, P_A \rangle = \mathit{accept}] \geq c$.
- **Soundness.** For every P_A that runs in $\mathcal{C}_{no}$ and any P_O, $\Pr[\langle V, P_O, P_A \rangle = \mathit{accept}] \leq s$. Note that the protocol is sound for unbounded P_O.

Transforming oracle separation into two-player protocols. Our goal is to transform an oracle separation problem R into a two-player protocol that distinguishes $\mathcal{C}_{yes}$ from $\mathcal{C}_{no}$. In this work, we focus on the cases where the oracle O is quantum accessible, and the predicate R can be computed deterministically. For real numbers $s \geq 0$ and $c \geq s + n^{-O(1)}$, we say a two-player protocol is (c, s)-equivalent to a problem R if there exists a classical verifier V such that the following conditions hold.

- **Completeness.** For every $\mathcal{A}^O$ that runs in $\mathcal{C}_{yes}^O$ and solves R with probability $2/3$, there exist P_A that runs in $\mathcal{C}_{yes}$ and a quantum prover P_O, such that V accepts with probability at least c.

[6] In this work, we only focus on the machines that can at least teleport quantum states.

- **Soundness.** For every P_A that runs in $\mathcal{C}_{no}$, if V accepts with probability more than s, there exists an oracle algorithm $\mathcal{A}^O$ that runs in $\mathcal{C}_{no}^O$ and solves R with probability more than $1/3$.

The completness statemenet together with the contrapositive of the soundness statement implies that if there is an oracle separation problem between $\mathcal{C}_{yes}$ and $\mathcal{C}_{no}$, then there is a two-player protocol that separates $\mathcal{C}_{yes}$ and $\mathcal{C}_{no}$. We note the thresholds $2/3$ and $1/3$ are unimportant for our analysis; they can be replaced for giving a contradiction.

6 A two-player non-local game for oracle separation

In this section, we show how to classically verify a query algorithm. More specifically, we show how to turn a query algorithm into a two-player protocol that preserves the completeness and soundness, up to an inverse polynomial multiplicative factor. Our protocol consists of two quantum provers P_A and P_O sharing entanglements and interacting with a purely classical verifier V. The prover P_O is designated to perform quantum computation O, and the prover P_A performs arbitrary quantum computation to learn the information about O.

The protocol consists of two phases: in a query phase, P_O receives a teleported quantum state from P_A. An honest P_O performs the unitary map of the oracle O and teleports the resulting state to P_A. Next, in the computation phase, P_A may perform any quantum computation, subject to its resource constraints.

To classically verify whether P_A, P_O instantiates a query algorithm in the two-player model, it is crucial that the verifier V can classically verify the P_O behaves as intended. To this end, we rely on the Leash protocol by Coladangelo, Grilo, Jeffery and Vidick [17], as introduced in Section 2.5. In this protocol, the classical verifier interacts with two entangled provers PP and PV. PP is designated to perform a quantum computation U on encrypted classical input $|x\rangle$ (using quantum one-time pad). The Leash protocol relies on the Broadbent protocol to verify that PP performs U, conditioned on PV performing the correct single-qubit measurement for each T-gadget. To classically verify PV's measurements, the verifier chooses either to run either a rigidity test to verify PV, or to execute the Broadbent protocol. It is crucial that a rigidity test looks indistinguishable to PV from the real execution of the Broadbent protocol, while at the same time PP needs to play differently for these two tests. On the other hand, when running the Broadbent protocol, PP and PV are required to perform one of three indistinguishable tests introduced in the previous sections. This allows the verifier to make sure PP behaves as intended.

To verify a query algorithm, a naive approach is to run the Leash protocol to check in every query, P_O honestly run the unitary O, and P_A performs the correct measurements (for the T-gadgets). However, note that in the Leash protocol, the tests hide the round type from the provers, and only one "query" to PP is made. Between the queries, P_A performs any computation on the plaintext (which can be done by applying the Pauli correction on the teleported state from P_O). When integrating the Leash protocol for multiple queries, the round type could possibly be leaked, at least to P_O: imagine that P_A teleports its quantum state $|\psi\rangle$ to P_O. When the verifier chooses to run a rigidity test, P_O actually performs corresponding measurements on fresh EPR pairs, and leaves the state $|\psi\rangle$ unchanged. Thus P_O on receiving the teleported quantum state, can run a swap test to check if the returned state has been changed.

To overcome the issue, our approach is to run the tests in the Leash protocol only once: in the beginning of the protocol, the verifier tosses a biased coin γ such that $\Pr[\gamma = 0] = O(1/q)$. If $\gamma = 1$, the verifier chooses a uniformly random index $\ell \in \{1, \ldots, q\}$, and performs a random test. In particular, when $\gamma = 1$, the verifier chooses a random iteration $i \in [q]$, and runs the Leash protocol for the i^{th} query, i.e., running a X-test, a Z-test, or a rigidity test with probability determined later. If the provers pass the test, then the verifier simply accepts and terminates (i.e., ignores the rest of the queries). If $\gamma = 0$, the verifier only performs the same check as in the original query algorithm, i.e., it checks whether P_A's final output w satisfies $R(O, w) = 1$.

We describe the protocol in the following steps. Note that though the provers do not necessarily follow the steps, we include the honest behavior for concreteness and clarity.

Protocol 4. [$\mathsf{QUERY}(q, \mathcal{O}, V)$] Let $p = 1/3$ and $\alpha = \Theta(1/q)$.

1. The verifier V samples a oracle $O \leftarrow_R \mathcal{O}$ for oracle ensemble $\mathcal{O}$ and a bit γ with $\Pr[\gamma = 0] = \alpha$. If $\gamma = 1$, then V samples a random index $\ell \leftarrow_R [q]$.
2. If $\gamma = 1$, for $i \in \{1, \dots, \ell - 1\}$, run $\mathsf{COMP}(\Sigma, O, m)$. For $i = \ell$,
   (a) V chooses a random protocol from $\{\mathsf{X\text{-}TEST}(\Sigma, O, m), \mathsf{Z\text{-}TEST}(\Sigma, O, m), \mathsf{RIGID}(\Sigma, m)\}$ with probability $p_X = p, p_Z = p, p_R = 1 - 2p$ respectively.
   (b) If the test succeeds then the verifier accepts and terminates; otherwise it rejects.

   Otherwise, $\gamma = 0$, for $i \in \{1, \dots, q\}$, run $\mathsf{COMP}(\Sigma, O, m)$. Finally P_A sends an answer w, and the verifier outputs 1 if $V(O, w) = 1$ and 0 otherwise.

The protocol QUERY (Protocol 4) describes V's behavior to initiate the protocol. More specifically, V randomly decides to do computation or tests first. If V decides to do tests, it randomly picks one iteration ℓ from $[q]$ and randomly selects a test (X, Z, or rigidity) with probability p_X, p_Z, p_R respectively. Then V applies corresponding protocols $\mathsf{X\text{-}TEST}$ (Protocol 6), $\mathsf{Z\text{-}TEST}$ (Protocol 7), and RIGID (Protocol 8) at iteration ℓ and always applies COMP (Protocol 5) to other rounds.

Steps 1 to 4 in these four protocols are the same. In particular, the verifier V divides the EPR pairs shared between P_O and P_A into subsets N_C, N_X, and N_Z for running COMP (Protocol 5), $\mathsf{X\text{-}TEST}$ (Protocol 6), and $\mathsf{Z\text{-}TEST}$ (Protocol 7) with corresponding measurements (chosen from the set Σ for each EPR pair). The registers $B_1, \dots, B_d$ are used for applying the T gadgets. Then, the verifier asks P_A to measure its halves of the EPR pairs according to W, to teleport his queries to P_O, and to send the corresponding Pauli correction to the V. The verifier V sets N according to the round type and reveals N only to P_O, but the underlying state remains unknown to P_O since it is encrypted by a quantum one-time pad. On the other hand, P_A cannot distinguish COMP (Protocol 5), $\mathsf{X\text{-}TEST}$ (Protocol 6) or $\mathsf{Z\text{-}TEST}$ (Protocol 7) since it does not know N. Moreover, from P_A's view, RIGID (Protocol 8) looks the same as the other tests. This implies that P_A cannot know if V is chooses to run RIGID (Protocol 8) to check its behavior.

Protocol 5. [$\mathsf{COMP}(\Sigma, O, m)$]

1. Setup:
   (a) V samples a random subset $N_C \subseteq [m]$ of size n, where n is the number of qubits O acts on.
   (b) For each $i \in \bar{N}_C$, V samples $W_i \leftarrow_R \Sigma$.
   (c) Then V samples two random subsets $N_X \subseteq \{i : W_i = X\}$ and $N_Z \subseteq \{i : W_i = Z\}$, both of size n.
   (d) V sets $N = N_C$.

(e) V partitions $[m]\backslash(N_C \cup N_X \cup N_Z)$ into subsets of equal size $B_1, \ldots, B_d$, where d is the number of T layers in O.

2. $(V \to P_A)$ V sends $N_C, W_{\bar{N}_C}$ to P_A sequentially.

3. $(P_A \to V)$ P_A teleports its quantum state $|\psi\rangle$ using EPR pairs with indices in N, and sends the Pauli correction (a_{N_C}, b_{N_C}). For each $i \in \bar{N}_C$, perform measurements on the i^{th} EPR pair in basis W_i. Finally P_A sends the outcomes $e_{\bar{N}_C}$.

4. $(V \to P_O)$: V sends N to P_O.

5. For each $\ell = 1, \ldots, d$,

   (a) $(V \to P_O)$ V chooses a random subset $T_\ell \subseteq \{i \in B_\ell : W_i \in \{G, F\}\}$, and sends T_ℓ to P_O. For each Clifford gate in the ℓ-th layer, perform the appropriate key update.

   (b) $(P_O \to V)$: P performs the Clifford operations in the ℓ-th layer. For each T-gadget in the ℓ-th layer, P_O runs the T-gadget on $(i, j) \in T_\ell \times N$, and sends the measurement outcome c_{T_ℓ} to V.

   (c) $(V \to P_O)$ For each $i \in T_\ell$, set $z_i = a_j + 1_{W_i = F} + c_i$. V sends z_{T_ℓ} to P_O.

6. $(P_O \to V)$ Let $|\phi\rangle$ be the resulting state after the evaluation of O is done. P_O teleports $|\phi\rangle$ to P_A and sends the Pauli correction a', b' to V.

In the following steps in Protocol 5, V basically guides P_O to apply the oracle on $|\psi\rangle$ using corresponding gadgets and updates the keys according to the measurement outcomes of P_O and P_A.

Protocol 6. [X-TEST(Σ, O, m)]

1. Setup:

   (a) V samples a random subset $N_C \subseteq [m]$ of size n.

   (b) For each $i \in \bar{N}_C$, V samples $W_i \leftarrow_R \Sigma$.

   (c) Then V samples two random subsets $N_X \subseteq \{i : W_i = X\}$ and $N_Z \subseteq \{i : W_i = Z\}$, both of size n.

   (d) V sets $N = N_X$.

   (e) V partitions $[m]\backslash(N_C \cup N_X \cup N_Z)$ into subsets of equal size $B_1, \ldots, B_d$, where d is the number of T layers in O.

2. $(V \to P_A)$ V sends $N_C, W_{\bar{N}_C}$ to P_A sequentially.

3. $(P_A \to V)$ P_A teleports its quantum state $|\psi\rangle$ using EPR pairs with indices in N, and sends the Pauli correction (a_{N_C}, b_{N_C}). For each $i \in \bar{N}_C$, perform measurements on the i^{th} EPR pair in basis W_i. Finally P_A sends the outcomes $e_{\bar{N}_C}$.

4. $(V \to P_O)$: V sends N to P_O.

5. For each $\ell = 1, \ldots, d$,

   (a) $(V \to P_O)$ V chooses a random subset $T_\ell = T_\ell^0 \cup T_\ell^1$ such that T_ℓ^0 is a random subset of $\{i : W_i = Z\}$, and T_ℓ^1 is a random subset of $\{i : W_i \in \{X, Y\}\}$. V sends T_ℓ to P_O. For each Clifford gate in the ℓ-th layer, perform the appropriate key update.

(b) $(P_O \to V)$: P_O performs the Clifford operations in the ℓ-th layer. For each T-gadget in the ℓ-th layer, P_O runs the T-gadget on $(i,j) \in T_\ell \times N$, and sends the measurement outcome c_{T_ℓ} to V.

(c) $(V \to P_O)$ For each $i \in T_\ell^0$, set $z_i \leftarrow_R \{0,1\}$; if $i \in T_\ell^1$, $z_i = 1_{W_i=Y}$. V sends z_{T_ℓ} to P_O. P_O teleports $|\phi\rangle$ to P_A and sends the Pauli correction a', b' to V.

6. $(P_O \to V)$ Let $|\phi\rangle$ be the resulting state after the evaluation of O is done. P_O teleports $|\phi\rangle$ to P_A and sends the Pauli correction a', b' to V.

7. $(V \to P_A)$ V requests P_A to perform a standard basis measurement on every qubit of the teleported state.

8. $(P_A \to V)$ P_A performs a standard basis measurement on every qubit of the teleported state and sends the outcome d. V accepts if and only if $d \oplus a' \oplus a'' = 0$, where a'' is calculated by the key update rule.

Protocol 7. [Z-TEST(Σ, O, m)]

1. Setup:

   (a) V samples a random subset $N_C \subseteq [m]$ of size n.

   (b) For each $i \in \bar{N}_C$, V samples $W_i \leftarrow_R \Sigma$.

   (c) Then V samples two random subsets $N_X \subseteq \{i : W_i = X\}$ and $N_Z \subseteq \{i : W_i = Z\}$, both of size n.

   (d) V sets $N = N_Z$.

   (e) V partitions $[m] \backslash (N_C \cup N_X \cup N_Z)$ into subsets of equal size $B_1, \ldots, B_d$, where d is the number of T layers in O.

2. $(V \to P_A)$ V sends $N_C, W_{\bar{N}_C}$ to P_A sequentially.

3. $(P_A \to V)$ P_A teleports its quantum state $|\psi\rangle$ using EPR pairs with indices in N, and sends the Pauli correction (a_{N_C}, b_{N_C}). For each $i \in \bar{N}_C$, perform measurements on the i^{th} EPR pair in basis W_i. Finally P_A sends the outcomes $e_{\bar{N}_C}$.

4. $(V \to P_O)$: V sends N to P_O.

5. For each $\ell = 1, \ldots, d$,

   (a) $(V \to P_O)$ V chooses a random subset $T_\ell = T_\ell^0 \cup T_\ell^1$ such that T_ℓ^0 is a random subset of $\{i : W_i = \{X, Y\}\}$, and T_ℓ^1 is a random subset of $\{i : W_i \in Z\}$. V sends T_ℓ to P_O. For each Clifford gate in the ℓ-th layer, perform the appropriate key update.

   (b) $(P_O \to V)$: P_O performs the Clifford operations in the ℓ-th layer. For each T-gadget in the ℓ-th layer, P_O runs the T-gadget on $(i,j) \in T_\ell \times N$, and sends the measurement outcome c_{T_ℓ} to V.

   (c) $(V \to P_O)$ For each $i \in T_\ell^1$, set $z_i \leftarrow_R \{0,1\}$; if $i \in T_\ell^0$, $z_i = 1_{W_i=Y}$. V sends z_{T_ℓ} to P_O. P_O teleports $|\phi\rangle$ to P_A and sends the Pauli correction a', b' to V.

6. $(P_O \to V)$ Let $|\phi\rangle$ be the resulting state after the evaluation of O is done. P_O teleports $|\phi\rangle$ to P_A and sends the Pauli correction a', b' to V.

7. $(V \to P_A)$ V requests P_A to perform a Hadamard basis measurement on every qubit of the teleported state.

8. $(P_A \to V)$ P_A performs a Hadamard basis measurement on every qubit of the teleported state and sends the outcome d. V accepts if and only if $d \oplus b' \oplus b'' = 0$, where b'' is calculated by the key update rule.

Protocol 6 and Protocol 7 originate from the Broadbent protocol to check if P_O is consistent with O. The crucial idea, as introduced in Section 2.4, is that P_O acts as applying identity on an all-zero or an all-plus state up to key update. Therefore, the verifier can detect if there is an attack applied to the computation: First V asks P_A to perform a standard basis or a Hadamard basis measurement on the state received from P_O. Then it applies the key update rules to compute the decryption key, and check if the measurement outcomes from P_A decrypts to zero.

Since the above steps relies on reliable measurements performed by P_A, it is essential to enforce P_A to perform the measurements correctly. We include our modification $\mathsf{RIGID}(\Sigma, m)$ of the rigidity test in Section 2.5. The test is the same as $\mathsf{RIGID}'(\Sigma, |\bar{N}|)$ on a random subset $\bar{N}$ of $[m]$. The purpose of RIGID (Protocol 8) is to check if P_A measures its EPR pairs in bases W'. From the collection of measurement outcomes for questions W to P_A and W' to P_O, V checks if the outcomes follows the relation specified in Protocol 8. By Theorem 2.15, passing with probability $1 - \epsilon$ ensures P_A's output is $\text{poly}(\epsilon)$-close in total variation distance to a measurement performed on EPR pairs in the correct bases W. Note that P_O knows that V chooses to execute RIGID (Protocol 8) after receiving W from V. However, P_A does not know this since it only receives random partitions and W, indistinguishable from the messages he obtained in other protocols.

Protocol 8. [$\mathsf{RIGID}(\Sigma, m)$]

1. Setup:

   (a) V samples a random subset $N_C \subseteq [m]$ of size n.

   (b) For each $i \in \bar{N}_C$, V samples $W_i \leftarrow_R \Sigma$.

   (c) Then V samples two random subsets $N_X \subset \{i : W_i = X\}$ and $N_Z \subseteq \{i : W_i = Z\}$, both of size n.

   (d) V sets $N = N_C$.

   (e) V partitions $[m] \backslash (N_C \cup N_X \cup N_Z)$ into subsets of equal size $B_1, \ldots, B_d$, where d is the number of T layers in O.

2. Execute $\mathsf{RIGID}'(\Sigma, |\bar{N}|)$ on the subset $\bar{N}$ and output the outcome.

6.1 Completeness

The completeness immediately follows: for algorithm $\mathcal{A}$, the provers P_O performs O and P_A performs U_i in each iteration i. Then the provers with probability 1 if $\mathsf{X\text{-}TEST}$ or $\mathsf{Z\text{-}TEST}$ is chosen. By Theorem 2.15, when RIGID is chosen, they succeeds with probability $1 - \exp(-\Omega(n + t))$. We give a proof as follows.

Theorem 6.1 (Completeness)**.** *For every q-query algorithm $\mathcal{A}^O$ which outputs w satisfying $R(O, w) = 1$ with probability at least $2/3$, there exist provers P_O, P_A which succeed with probability at least $1 - \frac{\alpha}{3} - \exp(-\Omega(n + t))$.*

Proof. For every algorithm $\mathcal{A}$ that succeeds with probability at least $c' \geq 2/3$, in each iteration i, P_O runs O and P_A runs U_i. The provers passes X-TEST and Z-TEST, when they are chosen, with probability 1. By Theorem 2.15, when the verifier chooses to execute RIGID (with probability $(1-\alpha)/3$), honest provers succeed with probability $1 - \exp(-\Omega(n+t))$ since CHSH games are run in sequential repetition. Thus the success probability of the provers is

$$\alpha \cdot c' + \frac{2(1-\alpha)}{3} + \frac{1-\alpha}{3}(1 - \exp(-\Omega(n+t))) = 1 - \alpha(1-c') - \frac{1-\alpha}{3}\exp(-\Omega(n+t))$$
$$\geq 1 - \frac{\alpha}{3} - \exp(-\Omega(n+t)). \tag{36}$$

□

6.2 Soundness

To show the soundness, recall that it suffices to show that if the provers succeed with probability more than s, then there exists a query algorithm which is accepted with probability more than $1/3$. First we show a simpler case in which P_A behaves honestly.

Lemma 6.2 (Soundness with honest P_A)**.** *There exists $\alpha = \Theta(1/q)$ such that, for every provers P_O that succeeds with success probability $s > 1 - \frac{2\alpha}{3}$, there exists a query algorithm that is given access to O and outputs w satisfying $R(O, w) = 1$ with probability more than $1/3$.*

Proof. Let the success probability be $1 - \epsilon_i$ conditioned on $\gamma = 1$ and the chosen index is i. The the probability of failing X-TEST or Z-TEST is at most $p^{-1}\epsilon_i$. Then the quantum channel $\mathcal{E}_i$ that P_A implements at iteration i satisfies $\|\mathcal{E}_i - \mathcal{O}\|_\diamond \leq 2p^{-1}\epsilon_i$.

Then let $\mathcal{A}$ be $\mathcal{U}_q \circ \mathcal{O} \circ \mathcal{U}_{q-1} \circ \cdots \circ \mathcal{U}_1 \circ \mathcal{O} \circ \mathcal{U}_0$. By union bound,

$$\|\mathcal{A} - \mathcal{U}_q \circ \mathcal{E}_q \circ \mathcal{U}_{q-1} \circ \cdots \circ \mathcal{U}_1 \circ \mathcal{E}_1 \circ \mathcal{U}_0\|_\diamond \leq \frac{2}{p}\sum_{i=1}^{t} \epsilon_i = 2qp^{-1}\epsilon, \tag{37}$$

where $\epsilon := \frac{1}{q}\sum_i \epsilon_i \in [0, 1]$. Then the success probability of P_O is

$$s = (1-\alpha)(1-\epsilon) + \alpha\delta \leq (1-\alpha)(1-\epsilon) + \alpha(p_{\mathcal{A}} + 2q\epsilon/p), \tag{38}$$

where δ is the probability that the provers output w such that such that $R(O, w) = 1$ conditioned on $\gamma = 0$ (i.e., the second quantum channel in (37)), and $p_{\mathcal{A}}$ is the success probability of $\mathcal{A}$. This implies that

$$p_{\mathcal{A}} \geq \frac{s}{\alpha} - \Big(\frac{1}{\alpha} - 1\Big)(1-\epsilon) - 2q\epsilon/p = 1 - \frac{1-s}{\alpha} + \epsilon\Big(\frac{1}{\alpha} - 1 - \frac{2q}{p}\Big). \tag{39}$$

Setting $\alpha = \frac{1}{1+2q\cdot c/p}$ for any constant $c > 0$, we conclude that $p_{\mathcal{A}} \geq 1 - \frac{1-s}{\alpha}$. If $s > 1 - \frac{2\alpha}{3}$, $p_{\mathcal{A}} > 1/3$. □

Now we consider the effect of a cheating P_A. The crucial idea is if P_A chooses to deviate non-trivially from the protocol by ϵ in total variation distance, then the probability it is accepted when RIGID (Protocol 8) is chosen is then at most $1-\epsilon$. As argued previously, since P_A does not learn whether RIGID is selected, the same strategy must have been applied for other tests. This implies that in the delegation game (where X-TEST, Z-TEST or COMP is chosen), the score can be at most increased by at most $\operatorname{poly}(\epsilon)$. This is because as shown in Theorem 2.15, the rigidity test guarantees that for every pair of provers succeeds with probability $1-\epsilon$, the output transcript must be $\operatorname{poly}(\epsilon)$-close to the that from an honest strategy in total variance distance. More formally, for every pair of provers P_O and P_A such that they succeed in the rigidity test with probability $1-\epsilon$ and in the delegation game with probability q, there exist P'_O and P'_A such that P'_A plays honestly (i.e., performs a correct measurement on a half of every EPR pairs) and they succeed in the delegation game with probability $q - \operatorname{poly}(\epsilon)$. We use the result to prove the following theorem.

Theorem 6.3 (Soundness)**.** *For constant p, there exists $\alpha = 1/\operatorname{poly}(q)$, such that for every pair of provers that succeeds with probability $s > 1 - \frac{2\alpha}{3}$, there exists a query algorithm that is given access to O and outputs w satisfying $R(O, w) = 1$ with probability more than $1/3$.*

Proof. Let the success probability be $1-\epsilon_i$ conditioned on $\gamma = 1$ and the chosen index being $i \in [q]$. Thus the failure probabilities are at most $\frac{\epsilon_i}{p}$, $\frac{\epsilon_i}{p}$ and $\frac{\epsilon_i}{1-2p}$ respectively conditioned on the events that an X-TEST, an Z-TEST and an rigidity test RIGID is chosen. Also note that when an X-TEST or a Z-TEST is chosen, the provers do not distinguish the test from COMP until V asks a measurement from P_A. When RIGID is chosen, P_A does not distinguish it from COMP, X-TEST, Z-TEST until V accepts or rejects.

By Theorem 2.15, there exist P'_A, P'_O such that P'_A plays honestly, and P'_O successfully passes X-TEST and Z-TEST with probability at least $1 - \delta_i = 1 - \frac{\epsilon_i}{p} - \operatorname{poly}\left(\frac{\epsilon_i}{1-2p}\right)$. Thus by Theorem 2.14, P'_O implements a quantum channel $\mathcal{E}_i$ such that $\|\mathcal{E}_i - \mathcal{O}\|_\diamond \le 1 - 2\delta_i$.

Conditioned on $\gamma = 0$, let the process of P'_A on receiving a teleported state $\rho_{in}^{(i)}$, produces the output state $\rho_{out}^{(i)}$ be $\mathcal{U}_i : \rho_{in}^{(i)} \mapsto \rho_{out}^{(i)}$. Now let the algorithm $\mathcal{A} := \mathcal{U}_q \circ \mathcal{O}_q \circ \cdots \circ \mathcal{U}_1 \circ \mathcal{O}_1 \circ \mathcal{U}_0$. By union bound,

$$\|\mathcal{A} - \mathcal{U}_q \circ \mathcal{E}_q \circ \mathcal{U}_{q-1} \circ \cdots \circ \mathcal{U}_1 \circ \mathcal{E}_1 \circ \mathcal{U}_0\|_\diamond \le 2 \sum_{i=1}^{q} \delta_i, = 2q\delta, \tag{40}$$

where $\delta = \frac{1}{q}\sum_{i=1}^{q} \delta_i = \epsilon/p + \operatorname{poly}(\frac{\epsilon}{1-2p}) \le g(\epsilon) = \operatorname{poly}(\epsilon)$ for some monotonically increasing g in $[0, \infty)$ (e.g., $c \cdot \epsilon^a$ for constants $a \le 1$ and c). Since g is monotonically increasing for $\epsilon \ge 0$, we note that the inverse g^{-1} exists. The success probability of the provers is upper bounded by

$$s \le (1-\alpha)(1-\epsilon) + \alpha \cdot \max\{(p_{\mathcal{A}} + 2q \cdot g(\epsilon)), 1\} \tag{41}$$

where $\epsilon = \frac{1}{q}\sum_i \epsilon_i$ and $p_{\mathcal{A}}$ is the probability that measuring the associated qubits on $\mathcal{A}$'s output state yields an outcome w satisfying $R(O, w) = 1$. Since $g(\epsilon)$ is monotonically increasing, there exists $\epsilon^* \ge 0$ such that $2q\delta(\epsilon^*) = 1 - p_{\mathcal{A}} \le 2q \cdot g(\epsilon^*)$. This implies that

$$s \le (1-\alpha)(1-\epsilon^*) + \alpha = 1 - (1-\alpha)\epsilon^* \le 1 - (1-\alpha) \cdot g^{-1}\Big(\frac{1-p_{\mathcal{A}}}{2q}\Big), \tag{42}$$

and

$$p_{\mathcal{A}} \ge 1 - 2q \cdot g\Big(\frac{1-s}{1-\alpha}\Big). \tag{43}$$

By (43), if $s > 1-(1-\alpha)\cdot g^{-1}(\frac{1}{3q})$, $p_{\mathcal{A}} > 1/3$. For $\alpha > \frac{g^{-1}(\frac{1}{3q})}{\frac{2}{3}+g^{-1}(\frac{1}{3q})}$, we have $1-(1-\alpha)\cdot g^{-1}(\frac{1}{3q}) > 1-\frac{2\alpha}{3}$. It suffices to choose $\alpha = \frac{2\cdot g^{-1}(\frac{1}{3q})}{\frac{2}{3}+g^{-1}(\frac{1}{3q})} = 1/\operatorname{poly}(q)$. □

Setting $p = 1/3$, we conclude with the following corollary, a direct consequence of Theorem 6.1 and Theorem 6.3.

Corollary 6.4. *Let $\mathcal{C}_{yes}, \mathcal{C}_{no}$ be two complexity classes. If there exists an oracle $\mathcal{O}$ and a relation R such that R is solvable in $\mathcal{C}_{yes}^{\mathcal{O}}$ using q queries but not in $\mathcal{C}_{no}^{\mathcal{O}}$. Then, there exists $\alpha = 1/\operatorname{poly}(q)$ and a protocol $\langle V, P_O, P_A\rangle$ such that the following statements hold.*

- *There exist P_O that runs in $O(q \cdot CC(\mathcal{O})) + \operatorname{poly}(n)$ and P_A runs in $\mathcal{C}_{yes}$ such that the verifier accepts with proability at least $1 - \frac{\alpha}{3} - e^{-\Omega(n)}$.*
- *For any P_A that runs in $\mathcal{C}_{no}$ and any unbounded P_O, the verifier accepts with probability at most $1 - \frac{2\alpha}{3}$.*

Here, $CC(\mathcal{O})$ is the quantum circuit complexity for implementing $\mathcal{O}$.

7 Classical verification of quantum depth from oracle separation

In this section, we will prove the existence of $\mathsf{CVQD}_2(d, d')$ for integers $d' > d$. First we give the formal definition as follows.

Definition 7.1 ($\mathsf{CVQD}_2(d, d')$)**.** *Let $d, d' \in \mathbb{N}$ and $d' > d$. A two-prover protocol $\mathsf{CVQD}_2(d, d')$ that separates quantum circuit depth d from d' consists of a classical verifier V and two provers P_O, P_A such that the following properties hold:*

- **Non-locality:** *P_O and P_A share an arbitrary quantum state, and there is no quantum and classical channel between them once the protocol starts.*
- **Completeness:** *There exist an integer $\hat{d} \geq d'$, a quantum prover P_O and a $\hat{d}$-QC or $\hat{d}$-CQ scheme P_A such that $\Pr[\langle V, P_O, P_A\rangle = accept] \geq 2/3$.*
- **Soundness:** *For integer $\hat{d} \leq d$ and any $\hat{d}$-QC or $\hat{d}$-CQ scheme P_A and any P_O, $\Pr[\langle V, P_O, P_A\rangle = accept] \leq 1/3$.*

We prove the following theorem by Corollary 6.4 and the oracle separation in [12]. Note that Definition 7.1 does not specify the power of V and honest P_O except that V is a classical algorithm and P_O is a quantum machine that can store EPR pairs. we first show the existence of an information-theoretically secure $\mathsf{CVQD}_2(d, 2d+3)$ for any $d = \operatorname{poly}(n)$ with inefficient verification, i.e., V and honest P_O need to run in exponential time. Then, we show a $\mathsf{CVQD}_2(d, 2d+3)$ for any $d = \operatorname{poly}(n)$ with efficient verification under the assumption that qPRP (Definition 2.8) exists.

Theorem 7.2. *Let $d = \operatorname{poly}(n)$.*

1. *There exist $\alpha = 1/\operatorname{poly}(d)$ and an unconditionally secure $\mathsf{CVQD}_2(d, 2d+3)$ (Definition 7.1), in which the verifier V runs in probabilistic $O(2^n)$ time such that the following holds.*
   - **Completeness:** *There exist P_A which has quantum depth at least $2d+3$ and P_O which runs in quantum $O(2^n)$ time such that $\Pr[\langle V, P_O, P_A\rangle = accept] \geq 1 - \frac{\alpha}{3}$.*

- **Soundness:** *For any unbounded P_O and P_A that are d-CQ or d-QC schemes,* $\Pr[\langle V, P_O, P_A\rangle = accept] \leq 1 - \frac{2\alpha}{3}$.

2. *Assume that there exist quantum-secure pseudorandom permutations (qPRP) as defined in Definition 2.8. There exist $\alpha = 1/\operatorname{poly}(d)$ and $\mathsf{CVQD}_2(d, 2d+3)$ (Definition 7.1), in which V runs in probabilistic polynomial time such that the following holds.*

   - **Completeness:** *There exist P_A that has quantum depth at least $2d+3$ and P_O that runs in quantum polynomial time such that* $\Pr[\langle V, P_O, P_A\rangle = accept] \geq 1 - \frac{\alpha}{3}$.
   - **Soundness:** *For any unbounded P_O and P_A that are d-CQ or d-QC schemes,* $\Pr[\langle V, P_O, P_A\rangle = accept] \leq 1 - \frac{2\alpha}{3}$.

For efficient instantiations, it is required that the functions $f_0, \dots, f_d$ are efficiently samplable and computable. Any construction of qPRP satisfying Definition 2.8 (e.g., [32]) can be used to construct a pseudorandom d-shuffling of a pseudorandom Simon's function. We now give constructions.

In the problem d-SSP, the functions $f_0, \dots, f_{d-1}$ are random permutations. These functions can be replaced with pseudorandom permutations. For the last function f_d, we note that f_d can be written as $f \circ f_0^{-1} \circ \cdots \circ f_{d-1}^{-1}$, where f is a random Simon's function, when the domain is restricted to a hidden subset. It then suffices to show a construction of a pseudorandom Simon's function.

Definition 7.3 (Pseudorandom Simon's function)**.** *For finite set $\mathcal{Y}$, let $\mathcal{S}$ be the set of Simon's function from $\{0,1\}^n$ to $\mathcal{Y}$, i.e., $f \in \mathcal{S}$ if there exists $s \in \{0,1\}^n$ such that $f(x) = f(x')$ if and only if $x = x' \oplus s$. For key space $\mathcal{K}$, a pseudorandom Simon's function is a keyed function $F : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^m$ such that for every quantum adversary $\mathcal{A}$, it holds that*

$$\left| \Pr_{F \leftarrow_R \mathcal{S}}[\mathcal{A}^F() = 1] - \Pr_{k \leftarrow_R \mathcal{K}}[\mathcal{A}^{F_k}() = 1] \right| \leq \operatorname{negl}(n). \tag{44}$$

We note that by the definition of Simon's function, it must be the case that $m \geq n-1$. Next we prove that there exists a pseudorandom Simon's function from qPRP.

Claim 7.4. *Assume that qPRP exists as defined in Definition 2.8. Then there exists a pseudorandom Simon's function as defined in Definition 7.3.*

Proof. We first show that a random Simon's function can be constructed from a random permutation, and thus replacing a random permutation with a qPRP, we obtain a pseudorandom Simon's function.

Let $H := \{x \in \{0,1\}^n : x < x \oplus s\}$ for total ordering $<$ over $\{0,1\}^n$ defined as follows: For $x, y \in \{0,1\}^n$, $x < y$ if the smallest index $i \in [n]$ where $x_i \neq y_i$ satisfies $x_i = 0$ and $y_i = 1$.

The subset H forms a subgroup of $\{0,1\}^n$ for group operation $\oplus$: It is clear that $0 \in H$ since 0 is smaller than any string in $\{0,1\}^n$. Let $i \in [n]$ be the smallest index such that $s_i = 1$. For $x, y \in H$, $x_i = y_i = 0$, and thus $(x \oplus y)_i = 0$. This implies that $x \oplus y \in H$. Since H is a subgroup, the cosets $\{H, s \oplus H\}$ forms a partition of $\{0,1\}^n$.

Now we show that for codomain $\mathcal{Y} = \{0,1\}^m$ where $m \geq n-1$, every Simon's function $f : \{0,1\}^n \to \mathcal{Y}$ can be constructed from a permutation and a hidden shift s. We then define the following function $T_s : \{0,1\}^n \to H$, $T_s(x) = x$ for $x \in H$, and $T_s(x) = x \oplus s$ for $x \in s \oplus H$. Let the mapping $W_s : H \to \{0,1\}^m$,

$$W_s(x_1, \dots, x_n) = (x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n, 0, \dots, 0), \tag{45}$$

where i is the smallest index such that $s_i = 1$. The padding has $m - n + 1$ zeros. For every Simon's function f with shift s, we know that $f = f \circ T_s$. When the domain is restricted to H, f is injective, and thus there exist $(2^m - |H|)!$ permutations $P : \{0,1\}^m \to \{0,1\}^m$ such that $f = P \circ W_s \circ T_s$ (we use the convention that $0! = 1$). On the other hand, by definition, for every P, $P \circ W_s \circ T_s$ is a Simon's function. These facts imply that $(P, s) \mapsto P \circ W_s \circ T_s$ is a well-defined mapping from a pair of permutation and shift to a Simon's function and is $(2^m - |H|)!$-to-1.

Thus a random Simon's function can be sampled (inefficiently) as follows: First sample a uniform shift $s \leftarrow_R \{0,1\}^n$ and a random permutation $P : \{0,1\}^m \to \{0,1\}^m$ and output $P \circ W_s \circ T_s$. A pseudorandom Simon's function can be sampled efficiently using a qPRP $F : \mathcal{K} \times \{0,1\}^m \to \{0,1\}^m$: Sample a random shift $s \leftarrow_R \{0,1\}^n$ and $k \leftarrow_R \mathcal{K}$ and output $g_{k,s} := F_k \circ W_s \circ T_s$.

To show that $g_{k,s}$ indeed yields a pseudorandom Simon's function, suppose toward contradiction there exists an adversary $\mathcal{A}$ which distinguishes $g_{k,s}$ for uniform k, s from a random Simon's function with non-negligible probability ϵ. Then the reduction $\mathcal{B}$ given oracle access to a permutation Q, samples $s \leftarrow_R \{0,1\}^n$ and outputs $b \leftarrow \mathcal{A}^{Q \circ W_s \circ T_s}$. If Q is random, $Q \circ W_s \circ T_s$ is a random Simon's function; otherwise let the key be k, and the Simon's function is $g_{k,s}$. By the assumption, $\mathcal{B}$ distinguishes a random permutation from a qPRP with non-negligible advantage. □

We then define the pseudorandom d-shuffling of a function f, which can be implemented in quantum polynomial time.

Definition 7.5 (Pseudorandom d-shuffling (cf. Definition 2.4))**.** *For $f : \{0,1\}^n \to \{0,1\}^n$, pseudorandom d-shuffling of f is a tuple of functions $(f_0, \ldots, f_d)$, where $f_0, \ldots, f_{d-1}$ are pseudorandom permutations over $\{0,1\}^{(d+2)n}$, and the last function f_d is a fixed function satisfying the following properties: Let $S_d := \{f_{d-1} \circ \cdots \circ f_0(x') : x' \in \{0,1\}^n\}$.*

- *For $x \in S_d$, let $f_{d-1} \circ \cdots \circ f_0(x') = x$, and choose the function $f_d : S_d \to [0, 2^n - 1]$ such that $f_d \circ f_{d-1} \circ \cdots f_0(x') = f(x')$.*
- *For $x \notin S_d$, $f_d(x) = \perp$.*

Now we are ready to define a pseudorandom d-shuffling Simon's problem.

Problem 2 (Pseudorandom d-SSP (cf. Problem 1))**.** *Given oracle access to a pseudorandom d-shuffling (Definition 7.5) of a pesudorandom Simon's function (Definition 7.3), the problem is to find the hidden shift.*

By a simple hybrid argument, pseudorandom d-SSP separates a depth-$(2d+3)$ quantum computation from a depth-d one.

Corollary 7.6. *Let $d = \mathrm{poly}(n)$. Pseudorandom d-SSP (Problem 2) can be solved by a QNC_{2d+3} circuit with classical post-processing. Furthermore, for any $\hat{d}$-CQ and $\hat{d}d$-QC schemes $\mathcal{A}$ with $\hat{d} \le d$, the probability that $\mathcal{A}$ solves the problem is negligible.*

Now, we can prove Theorem 7.2.

Proof of Theorem 7.2. By Theorem 2.6, d-SSP separates the complexity classes $\mathsf{BPP}^{\mathsf{BQNC}_d} \cup \mathsf{BQNC}_d^{\mathsf{BPP}}$ and $\mathsf{BPP}^{\mathsf{BQNC}_{2d+3}} \cap \mathsf{BQNC}_{2d+3}^{\mathsf{BPP}}$ relative to the d-shuffling oracle of f. Furthermore, teleporting quantum states only takes one circuit depth by choosing the gateset properly or considering the gateset including all one- and two-qubit gates as in Remark 2.3. Therefore, by setting $q = 2d + 1$, $\mathcal{O}$ to be the shuffling oracle, and R to be d-SSP, Protocol 4 separates depth-$(2d+3)$ quantum computation from depth-d and the relation $R(O, w) = 1$ if and only if O is the shuffling oracle and w is the hidden shift according (see Corollary 6.4). Here, V and P_O are inefficient since describing and implementing the shuffling oracle are inefficient.

Following the same proof, we can also show that Protocol 4 separates depth-$(2d+3)$ quantum computation from depth-d by replacing d-SSP by pseudorandom d-SSP. This follows from the fact that pseudorandom d-SSP also separates depth-$(2d+3)$ quantum computation from depth-d by Corollary 7.6. Then, we follow the same proof for d-SSP by using Corollary 6.4 except that we set $\mathcal{O}$ to be a pseudorandom-shuffling oracle that can be described and implemented efficiently, and thus both V and P_O are efficient. $\square$

Note that the algorithm is allowed to make parallel queries which do not increase the query depth. It is straightforward to adapt Protocol 4 to allow parallel queries: let t denote the largest number of parallel queries. For any query algorithm $\mathcal{A}$ of depth q, there is a query algorithm $\mathcal{A}'$ that is given access to $O^{\otimes t}$ and achieves the same performance as $\mathcal{A}$. The equivalent two-player protocol to $\mathcal{A}'$ is $\mathsf{QUERY}(2d+3, O', R)$, where sampling O' can be performed by sampling $O \leftarrow_R \mathcal{O}$ and outputting $O' = O^{\otimes t}$.

Furthermore, we emphasize that while the protocol only has a small completeness-soundness gap $\alpha/3 = 1/\operatorname{poly}(d)$, by sequential repetition for $O(\alpha^{-2} \cdot \log^2 \lambda)$ times it suffices to amplify the gap to $1 - \operatorname{negl}(\lambda)$.

7.1 A nearly optimal separation

In this section, we modify the original d-SSP to give an oracle separation that reduces the gap from d versus $2d+3$ to d versus $d+3$.

First, instead of considering the standard quantum query model, we define d-shuffling in the "in-place" quantum oracle model.

Definition 7.7 (In-place d-shuffling)**.** *Let $f : \{0,1\}^n \to \{0,1\}^n$ be a Simon's function with shift s. Let $\mathcal{F} := \{f_0, \ldots, f_d\}$ be a d-shuffling of f. We define the in-place (d, f)-shuffling $U := \{U_{f_0}, \ldots, U_{f_d}\}$ as follows:*

1. *For $i = 0$, let U_{f_0} be a unitary such that for all $x \in \{0,1\}^{(d+2)n}$, $U_{f_0}|x\rangle|0\rangle = |x\rangle|f_0(x)\rangle$.*
2. *For $i = 1, \ldots, d-1$, let U_{f_i} be a unitary in $\mathbb{C}^{2^{(d+2)n} \times 2^{(d+2)n}}$ such that for all $x \in \{0,1\}^{(d+2)n}$, $U_{f_i}|x\rangle = |f_i(x)\rangle$.*
3. *Let U_{f_d} be a unitary in $\mathbb{C}^{2^{(d+2)n+1} \times 2^{(d+2)n+1}}$ such that for all $x \in \{0,1\}^{(d+2)n}$ and $b \in \{0,1\}$, $U_{f_d}|x, b\rangle = |f_d(x)\rangle|b \oplus b'\rangle$, where $b' = 1$ if $x \in H$ (see the definition of H in the proof of Claim 7.4).*

We note that an in-place pseudorandom permutation exists if there exists a qPRP defined in Definition 2.8. First we give the definition of in-place qPRPs.

Definition 7.8 (In-place qPRP (cf. Definition 2.8))**.** *For security parameter λ and polynomial $m = m(\lambda)$, a pseudorandom permutation P over $\{0,1\}^m$ is a keyed function $\mathcal{K} \times \{0,1\}^m \to \{0,1\}^m$ such that there exists a negligible function such that for every quantum adversary $\mathcal{A}$, it holds that*

$$\left| \Pr_{F \leftarrow_R \mathcal{P}}[\mathcal{A}^{I_F, I_{F^{-1}}} = 1] - \Pr_{k \leftarrow_R \mathcal{K}}[\mathcal{A}^{I_{P(k,\cdot)}, I_{P^{-1}(k,\cdot)}} = 1] \right| \leq \operatorname{negl}(\lambda), \tag{46}$$

where $I_Q : |x\rangle \mapsto |Q(x)\rangle$ for $x \in \{0,1\}^m$ and permutation $Q : \{0,1\}^m \to \{0,1\}^m$.

Theorem 7.9. *If there exists a qPRP as defined in Definition 2.8, then an in-place qPRP defined in Definition 7.8 exists.*

Proof. It suffices to show that an in-place oracle of a permutation P can be implemented using two queries to the standard oracles $O_P : |x,y\rangle \mapsto |x, y \oplus P(x)\rangle$ and $O_{P^{-1}}$. A query to the in-place oracle of P can be computed in the following steps:

$$|x,0\rangle \xmapsto{O_P} |x,P(x)\rangle \xmapsto{O_{P^{-1}}} |x \oplus P^{-1}(P(x)), P(x)\rangle = |0,P(x)\rangle. \tag{47}$$

Swapping the registers and removing the ancilla yields a mapping $|x\rangle \mapsto |P(x)\rangle$.

Now we show that if $F : \mathcal{K} \times \{0,1\}^n \to \{0,1\}^n$ is a qPRP as defined in Definition 2.8, then F is an in-place qPRP. Suppose toward contradiction that a quantum adversary $\mathcal{A}$ which distinguishes $|x\rangle \mapsto |F_k(x)\rangle$ for $k \leftarrow_R \mathcal{K}$ and $|x\rangle \mapsto |P(x)\rangle$ for uniform permutation P with advantage ϵ. Then we show there exists an adversary $\mathcal{B}$ which distinguishes $O_{F_k}, O_{F_k^{-1}}$ from $O_P, O_{P^{-1}}$. The adversary $\mathcal{B}$ simulates the mapping in (47) using two queries to either $O_{F_k}, O_{F_k^{-1}}$ or $O_P, O_{P^{-1}}$ and runs $\mathcal{A}$. By the assumption, $\mathcal{B}$ distinguishes the oracles with advantage ϵ. □

We define *in-place* d-SSP as follows:

Definition 7.10 (In-place d-SSP)**.** *Let $n \in \mathbb{N}$ and $f : \{0,1\}^n \to \{0,1\}^n$ be a random Simon's function. Given access to the in-place d-shuffling oracle of f, the problem is to find the hidden shift of f.*

Theorem 7.11. *Let $d = \mathrm{poly}(n)$. In-place d-SSP can be solved using $(d+3)$-CQ and $(d+3)$-QC schemes with access to the in-place d-shuffling oracle of f. Furthermore, for any d'-CQ or d'-QC schemes $\mathcal{A}$ with access to the in-place d-shuffling oracle of f and $d' \le d$, the probability that $\mathcal{A}$ solves the problem is negligible.*

Note that if we consider the models defined in Definition 3.8 and Definition 3.10 in [12], the quantum depth separation will be d versus $d+1$. See Remark 2.7 for the detailed discussion.

We present the proof of Theorem 7.11 in Appendix A. By Corollary 6.4 and Theorem 7.11, we have a construction of $\mathsf{CVQD}_2(d, d+3)$.

Corollary 7.12. *For $d = \mathrm{poly}(n)$, there exists an unconditionally secure $\mathsf{CVQD}_2(d, d+3)$ which is sound as in Theorem 7.2, 1 and complete with P_O and V running in $O(2^n)$ time. Moreover, if qPRP (Definition 2.8) exists, then there exists $\mathsf{CVQD}_2(d, d+3)$ which is sound as in Theorem 7.2, 2 and complete with P_O and V running in polynomial time.*

Proof. Following the same idea, we set $q = d+1$, and we choose $\mathcal{O}$ to be the in-place shuffling oracle and R to be in-place d-SSP. Then, Protocol 4 separates $d+3$-depth quantum circuit from d-depth quantum circuit in the presence of polynomial-time classical computation by Corollary 6.4 and Theorem 7.11.

Again, V and P_O are not efficient since implementing random permutations is expensive. Following the same idea for proving the second result of Theorem 7.2, we can make both V and P_O efficient using qPRP in the in-place oracle model. □

References

[1] Scott Aaronson and Lijie Chen. Complexity-theoretic foundations of quantum supremacy experiments. In *Proceedings of the 32nd Computational Complexity Conference*, CCC '17, Dagstuhl, DEU, 2017. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[2] Gorjan Alagic, Andrew M. Childs, Alex B. Grilo, and Shih-Han Hung. Non-interactive classical verification of quantum computation. In Rafael Pass and Krzysztof Pietrzak, editors, *Theory of Cryptography - 18th International Conference, TCC 2020, Durham, NC, USA, November 16-19, 2020, Proceedings, Part III*, volume 12552 of *Lecture Notes in Computer Science*, pages 153–180. Springer, 2020.

[3] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. Cryptography in nc^0. *SIAM Journal on Computing*, 36(4):845–888, 2006.

[4] Atul Singh Arora, Andrea Coladangelo, Matthew Coudron, Alexandru Gheorghiu, Uttam Singh, and Hendrik Waldner. Quantum depth in the random oracle model, 2022.

[5] Frank Arute, Kunal Arya, Ryan Babbush, Dave Bacon, Joseph C. Bardin, Rami Barends, Rupak Biswas, Sergio Boixo, Fernando G. S. L. Brandao, David A. Buell, Brian Burkett, Yu Chen, Zijun Chen, Ben Chiaro, Roberto Collins, William Courtney, Andrew Dunsworth, Edward Farhi, Brooks Foxen, Austin Fowler, Craig Gidney, Marissa Giustina, Rob Graff, Keith Guerin, Steve Habegger, Matthew P. Harrigan, Michael J. Hartmann, Alan Ho, Markus Hoffmann, Trent Huang, Travis S. Humble, Sergei V. Isakov, Evan Jeffrey, Zhang Jiang, Dvir Kafri, Kostyantyn Kechedzhi, Julian Kelly, Paul V. Klimov, Sergey Knysh, Alexander Korotkov, Fedor Kostritsa, David Landhuis, Mike Lindmark, Erik Lucero, Dmitry Lyakh, Salvatore Mandrà, Jarrod R. McClean, Matthew McEwen, Anthony Megrant, Xiao Mi, Kristel Michielsen, Masoud Mohseni, Josh Mutus, Ofer Naaman, Matthew Neeley, Charles Neill, Murphy Yuezhen Niu, Eric Ostby, Andre Petukhov, John C. Platt, Chris Quintana, Eleanor G. Rieffel, Pedram Roushan, Nicholas C. Rubin, Daniel Sank, Kevin J. Satzinger, Vadim Smelyanskiy, Kevin J. Sung, Matthew D. Trevithick, Amit Vainsencher, Benjamin Villalonga, Theodore White, Z. Jamie Yao, Ping Yeh, Adam Zalcman, Hartmut Neven, and John M. Martinis. Quantum supremacy using a programmable superconducting processor. *Nature*, 574(7779):505–510, 2019.

[6] Boaz Barak, Oded Goldreich, Rusell Impagliazzo, Steven Rudich, Amit Sahai, Salil Vadhan, and Ke Yang. On the (im)possibility of obfuscating programs. In *Annual international cryptology conference*, pages 1–18. Springer, 2001.

[7] Itay Berman, Iftach Haitner, and Eliad Tsfadia. A tight parallel repetition theorem for partially simulatable interactive arguments via smooth kl-divergence. In *Annual International Cryptology Conference*, pages 544–573. Springer, 2020.

[8] Zvika Brakerski, Paul Christiano, Urmila Mahadev, Umesh Vazirani, and Thomas Vidick. A cryptographic test of quantumness and certifiable randomness from a single quantum device. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 320–331, 2018.

[9] Zvika Brakerski, Venkata Koppula, Umesh Vazirani, and Thomas Vidick. Simpler proofs of quantumness. In *15th Conference on the Theory of Quantum Computation, Communication and Cryptography*, 2020.

[10] Anne Broadbent. How to verify a quantum computation. *Theory of Computing*, 14(1):1–37, 2018.

[11] Peter Chapman. Introducing the world's most powerful quantum computer. https://ionq.com/posts/october-01-2020-introducing-most-powerful-quantum-computer, 2021.

[12] Nai-Hui Chia, Kai-Min Chung, and Ching-Yi Lai. On the need for large quantum depth. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, page 902–915, 2020.

[13] Nai-Hui Chia, Kai-Min Chung, and Takashi Yamakawa. Classical verification of quantum computations with efficient verifier. In Rafael Pass and Krzysztof Pietrzak, editors, *Theory of Cryptography*, pages 181–206, Cham, 2020. Springer International Publishing.

[14] Nai-Hui Chia and Shih-Han Hung. Classical verification of quantum depth. *arXiv preprint arXiv:2205.04656*, 2022.

[15] Andrew M Childs, Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann, and Daniel A Spielman. Exponential algorithmic speedup by a quantum walk. In *Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 59–68, 2003.

[16] Richard Cleve and John Watrous. Fast parallel circuits for the quantum Fourier transform. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 526–536, 2000.

[17] Andrea Coladangelo, Alex B. Grilo, Stacey Jeffery, and Thomas Vidick. Verifier-on-a-leash: new schemes for verifiable delegated quantum computation, with quasilinear resources. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 247–277. Springer, 2019.

[18] Matthew Coudron and Sanketh Menda. Computations with greater quantum depth are strictly more powerful (relative to an oracle). In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 889–901, 2020.

[19] Jelle Don, Serge Fehr, and Christian Majenz. The measure-and-reprogram technique 2.0: Multi-round fiat-shamir and more. In *Advances in Cryptology – CRYPTO 2020*, pages 602–631. Springer International Publishing, 2020.

[20] Jelle Don, Serge Fehr, and Christian Majenz. The measure-and-reprogram technique 2.0: Multi-round fiat-shamir and more. In *Annual International Cryptology Conference*, pages 602–631. Springer, 2020.

[21] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. A quantum approximate optimization algorithm, 2014.

[22] Iftach Haitner. A parallel repetition theorem for any interactive argument. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 241–250. IEEE, 2009.

[23] Atsuya Hasegawa and Franccois Le Gall. An optimal oracle separation of classical and quantum hybrid schemes. In *International Symposium on Algorithms and Computation*, 2022.

[24] Johan Håstad, Rafael Pass, Douglas Wikström, and Krzysztof Pietrzak. An efficient parallel repetition theorem. In *Theory of Cryptography: 7th Theory of Cryptography Conference, TCC 2010, Zurich, Switzerland, February 9-11, 2010. Proceedings 7*, pages 1–18. Springer, 2010.

[25] Shuichi Hirahara and François Le Gall. Test of quantumness with small-depth quantum circuits. *arXiv preprint arXiv:2105.05500*, 2021.

[26] IBM announces advances to IBM quantum systems & ecosystem. https://www-03.ibm.com/press/us/en/pressrelease/53374.wss, 2017.

[27] Julian Kelly. A preview of bristlecone, Google's new quantum processor. https://ai.googleblog.com/2018/03/a-preview-of-bristlecone-googles-new.html, 2018.

[28] Zhenning Liu and Alexandru Gheorghiu. Depth-efficient proofs of quantumness. *arXiv preprint arXiv:2107.02163*, 2021.

[29] Jarrod R. McClean, Jonathan Romero, Ryan Babbush, and Alán Aspuru-Guzik. The theory of variational hybrid quantum-classical algorithms. *New Journal of Physics*, 18(2):023023, Feb 2016.

[30] Edwin Pednault, John Gunnels, Dmitri Maslov, and Jay Gambetta. On "quantum supremacy". https://www.ibm.com/blogs/research/2019/10/on-quantum-supremacy/, 2019.

[31] Rigetti Aspen-8 on AWS. https://medium.com/rigetti/rigetti-aspen-8-on-aws-236d9dc11613, 2021.

[32] Mark Zhandry. A note on quantum-secure PRPs. *arXiv preprint arXiv:1611.05564*, 2016.

A Proof of Theorem 7.11

Lemma A.1. *In-place d-SSP can be solved by a $(d+3)$-depth quantum circuit with classical post-processing.*

Proof. The algorithm is as follows:

$$
\begin{aligned}
\sum_{x\in\mathbb{Z}_2^n}|x\rangle|0\rangle|0\rangle &\xrightarrow{U_{f_0}} \sum_{x\in\mathbb{Z}_2^n}|x\rangle|f_0(x)\rangle|0\rangle \\
&\xrightarrow{U_{f_1}} \sum_{x\in\mathbb{Z}_2^n}|x\rangle|f_1(x)\rangle|0\rangle \\
&\xrightarrow{U_{f_d}} \sum_{x\in\mathbb{Z}_2^n}|x\rangle|f(x)\rangle|b(x)\rangle \\
&\xrightarrow{\text{measure the second register}} \frac{1}{\sqrt{2}}(|x\rangle|b(x)\rangle+|x\oplus s\rangle|b(x\oplus s)\rangle) \\
&\xrightarrow{\text{Apply H on the last qubit and measure}} \frac{1}{\sqrt{2}}(|x\rangle+|x\oplus s\rangle) \text{ with probability } 1/2. \\
&\xrightarrow{H^{\otimes n}} \frac{1}{\sqrt{2^n}}\sum_{j\in\mathbb{Z}_2^n}((-1)^{x\cdot j}+(-1)^{(x+s)j})|j\rangle
\end{aligned}
$$

Then, the rest of the algorithm follows from Simon's algorithm. The additional two depths come from the Hadamard gates an the beginning and at the end of the above algorithm. ☐

We sketch a proof for the lower bound in the following.

Lemma A.2. *For any d-CQ or d-QC schemes $\mathcal{A}$, the probability that $\mathcal{A}$ solves in-place d-SSP is negligible.*

To prove that in-place d-SSP is hard for d-depth quantum circuit, we need to prove the oneway-to-hiding lemma (Lemma 5.7 in [12]) for in-place shuffling oracle.

Claim A.3 (in-place oracle version of Lemma 5.7 in [12])**.** *Let $\mathcal{F}$ be a d-shuffling of a random Simon's function f and $\mathcal{U} := \{U_{f_0}, \ldots, U_{f_d}\}$ be the corresponding in-place d-shuffling. Let $\mathbf{S} = \{\bar{S}^{(0)}, \ldots, \bar{S}^{(d)}\}$ be a sequence of hidden sets as defined in Definition 5.2 in [12]. Then, for all $k = 0, \ldots, d$, there exists a shadow $\mathcal{G}$ of $\mathcal{F}$ in $\bar{S}^{(k)}$ such that for any single-depth quantum circuit U_c, initial state ρ, and any binary string t,*

$$
\begin{aligned}
|\Pr[\Pi_{0/1} \circ \mathcal{U}_{\mathcal{F}} U_c(\rho) = t] - \Pr[\Pi_{0/1} \circ \mathcal{U}_{\mathcal{G}} U_c(\rho) = t]| &\leq B(\mathcal{U}_{\mathcal{F}} U_c(\rho), \mathcal{U}_{\mathcal{G}} U_c(\rho)) \\
&\leq \sqrt{2 \Pr[\textit{find } \bar{S}^{(k)} : U^{\mathcal{F} \setminus \bar{S}^{(k)}}, \rho]}.
\end{aligned}
$$

Here, $B(\cdot, \cdot)$ is the Bures distance between quantum states, and $U^{\mathcal{F} \setminus \bar{S}^{(k)}}$ is defined in Definition 5.6 in [12]. $\mathcal{U}_{\mathcal{G}}$ is the in-place oracle for $\mathcal{G}$.

Proof. We first define the shadow $\mathcal{G}$ of $\mathcal{F}$ in $\bar{S}^{(k)}$. The definition follows the same spirit of the shadow in [12]. In the original definition, the shadow $\mathcal{G}$ maps $x \in \bar{S}^{(k)}$ to a special symbol $\perp$ and is consistent with $\mathcal{F}$ for $x \notin \bar{S}^{(k)}$. This definition of shadow does not work in the in-place oracle setting since the corresponding oracle is not a unitary.

So, here, we define $\mathcal{G}$ as a random function satisfying the following: If $x \notin \bar{S}^{(k)}$, we let $\mathcal{G}(x) = \mathcal{F}(x)$; else if $x \in \bar{S}^{(k)}$, we let $\mathcal{G}(x)$ to be independent of $\mathcal{F}(x)$, and the in-place oracle of $\mathcal{G}$, $U_{\mathcal{G}}$, is still a unitary. In particular, for shadows corresponding to $f_1, \ldots, f_{d-1}$ in $\bar{S}^{(k)}$, we pick another random permutation that is independent of $f_1, \ldots, f_{d-1}$ in $\bar{S}^{(k)}$ and is consistent with $f_1, \ldots, f_{d-1}$ for $x \notin \bar{S}^{(k)}$; for f_d, we can pick another 2-to-1 function that results in no hidden shift or a different hidden shift.

Then, the rest of the proof directly follows from the proof for Lemma 5.7 in [12].

□

The rest of the analysis to show that in-place d-SSP is hard for d-depth quantum circuit in the presence of classical computation follows the proof in [12] by using the new shadow we construct in the proof for Claim A.3.