Quantum Depth in the Random Oracle Model

Atul Singh Arora
IQIM and CMS
California Institute of Technology
USA
atul.singh.arora@gmail.com

Andrea Coladangelo
University of Washington
USA
andrea.coladangelo@gmail.com

Matthew Coudron
QuICS, NIST
University of Maryland
USA
mcoudron@umd.edu

Alexandru Gheorghiu
Chalmers University of Technology
Sweden
ETH Zürich
Switzerland
alexandru.gheorghiu@chalmers.se

Uttam Singh
Polish Academy of Sciences
Poland
IIIT Hyderabad
India
uttam@iiit.ac.in

Hendrik Waldner
University of Maryland
USA
MPI-SP
Germany
hwaldner@umd.edu

ABSTRACT

We give a comprehensive characterisation of the computational power of shallow quantum circuits combined with classical computation. Specifically, for classes of *search problems*, we show that the following statements hold, relative to a *random oracle*:

(a) $\mathsf{BPP}^{\mathsf{QNC}^{\mathsf{BPP}}} \neq \mathsf{BQP}$. This refutes Jozsa's conjecturein the random oracle model. As a result, this gives the first *instantiatable* separation between the classes by replacing the oracle with a cryptographic hash function, yielding a resolution to one of Aaronson's ten semi-grand challenges in quantum computing.

(b) $\mathsf{BPP}^{\mathsf{QNC}} \not\subseteq \mathsf{QNC}^{\mathsf{BPP}}$ and $\mathsf{QNC}^{\mathsf{BPP}} \not\subseteq \mathsf{BPP}^{\mathsf{QNC}}$. This shows that there is a subtle interplay between classical computation and shallow quantum computation. In fact, for the second separation, we establish that, for some problems, the ability to perform adaptive measurements in a *single* shallow quantum circuit, is more useful than the ability to perform *polynomially many* shallow quantum circuits without adaptive measurements. We also show that $\mathsf{BPP}^{\mathsf{QNC}}$ and $\mathsf{QNC}^{\mathsf{BPP}}$ are both *strictly* contained in $\mathsf{BPP}^{\mathsf{QNC}^{\mathsf{BPP}}}$.

(c) There exists a 2-message *proof of quantum depth* protocol. Such a protocol allows a classical verifier to efficiently certify that a prover must be performing a computation of some minimum quantum depth. Our proof of quantum depth can be instantiated using the recent proof of quantumness by Yamakawa and Zhandry.

CCS CONCEPTS

• **Theory of computation** → **Quantum complexity theory**; **Circuit complexity**; **Complexity classes**; **Cryptographic protocols**.

KEYWORDS

Hybrid classical-quantum models of computation, proof of quantum depth, random oracle model

STOC '23, June 20–23, 2023, Orlando, FL, USA

ACM ISBN 978-1-4503-9913-5/23/06…$15.00
https://doi.org/10.1145/3564246.3585153

ACM Reference Format:
Atul Singh Arora, Andrea Coladangelo, Matthew Coudron, Alexandru Gheorghiu, Uttam Singh, and Hendrik Waldner. 2023. Quantum Depth in the Random Oracle Model. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing (STOC '23), June 20–23, 2023, Orlando, FL, USA.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3564246.3585153

1 INTRODUCTION

High depth circuits are believed to be strictly more powerful than low depth circuits, in the sense that having deeper circuits allows one to solve a larger set of problems. Indeed, this is a well established fact for both classical and quantum circuits of depth sub-logarithmic in the size of the input [5, 6, 30, 32, 41]. However, for circuits of (poly)logarithmic depth and general polynomial depth, proving any sort of *unconditional* separation is challenging [39]. In fact, there is not even an unconditional proof that the set of problems that can be solved by polylog-depth classical circuits, NC, is a *strict subset* of the set of problems solvable by poly-depth classical circuits, P (or BPP when allowing for randomness). The same is believed to be the case for the quantum analogues of these classes, QNC and BQP, respectively. Nevertheless, the strict containments $\mathsf{NC} \subsetneq \mathsf{P}$ and $\mathsf{QNC} \subsetneq \mathsf{BQP}$ are known to hold in the oracle setting and, in particular, relative to a *random oracle* [35].[1] This is a strong indication that there are problems in P (BQP) which cannot be parallelised so as to be solvable in NC (QNC). Under the *random oracle heuristic*, by replacing the random oracle with a cryptographic hash function, one can even provide concrete instantiations of such problems. A further indication of the separation between low and high depth computations is provided by certain inherently sequential cryptographic constructions such as time-lock puzzles and verifiable delay functions [16, 40].

The study of circuit depth can also yield insights into the subtle relationship between quantum and classical computation by considering *hybrid circuit models* that combine quantum and classical computation [11, 21, 28, 31]. In this setting, one can ask the question: how powerful are poly-depth classical circuits, when augmented with polylog-depth quantum circuits? Could it be the case that interspersing BPP with QNC computations captures the full power

[1] Technically [35] only shows the strict containment $\mathsf{NC} \subsetneq \mathsf{P}$, relative to a random oracle. However, the quantum version $\mathsf{QNC} \subsetneq \mathsf{BQP}$ can also be shown as a straightforward extension of that result. That containment also follows from [24].

of BQP computations? Jozsa famously conjectured that the answer is yes [33]. Indeed, there is some evidence to support this conjecture, as the quantum Fourier transform, a central building block for many quantum algorithms, was shown to be implementable with log-depth quantum circuits [25]. This also implies that Shor's algorithm can be performed by a $\mathsf{BPP}^{\mathsf{QNC}}$ machine, a polynomial-time classical computer having the ability to invoke a (poly)log depth quantum computer.[2] Moreover, in the oracle setting, a number of problems yielding exponential separations between quantum and classical computation require only constant quantum-depth to solve, providing further support for Jozsa's conjecture [1, 3, 42].

Despite the evidence in support of Jozsa's conjecture, it was recently shown that, in the oracle setting, the conjecture is false [21, 28]. Specifically, the results of [21] (hereafter referred to as CCL) and [28] (hereafter referred to as CM) considered two ways of interspersing poly-depth classical computation with d-depth quantum computation. The first is $\mathsf{BPP}^{\mathsf{QNC}_d}$, denoting problems solvable by a BPP machine that can invoke d-depth quantum circuits (whose outputs are measured in the computational basis). The second, $\mathsf{QNC}_d{}^{\mathsf{BPP}}$, denotes problems solvable by a d-depth quantum circuit that can invoke a BPP machine at each layer in the computation.[3] Later, borrowing terminology from [11, 21], we will refer to the former circuit model as CQ_d and the latter as QC_d. However, for the purposes of this introduction, we will stick to the more familiar notation using complexity classes. Intuitively, $\mathsf{BPP}^{\mathsf{QNC}_d}$ captures the setting of a classical computer that can invoke a d-depth quantum computer several times. Examples of this include quantum machine learning algorithms such as VQE or QAOA [29, 37], though as mentioned, Shor's algorithm is also of this type. On the other hand, $\mathsf{QNC}_d{}^{\mathsf{BPP}}$ captures a d-depth *measurement-based quantum computation* [18, 38], where intermediate measurements are performed after each layer in the quantum computation. The outcomes of those measurements are processed by a poly-depth classical computation and the results are "fed" into the next quantum layer. CCL and CM showed that there exists an oracle relative to which $\mathsf{BPP}^{\mathsf{QNC}_d} \cup \mathsf{QNC}_d{}^{\mathsf{BPP}} \subsetneq \mathsf{BQP}$, for any $d = \mathrm{polylog}(n)$, with n denoting the size of the input. Notably, each work considered a different oracle for showing the separation. For CM, the oracle is the same one as for Childs' glued trees problem [23]. For CCL, the oracle is a modified version of the oracle used for Simon's problem [42], where the modification involves performing a sequence of permutations, allowing them to enforce high quantum depth.

CCL and CM were the first results to provide a convincing counterpoint to Jozsa's conjecture. However, the main drawback of the CCL and CM results is that they are relative to oracles that are highly structured and it is unclear if they can be explicitly instantiated based on some cryptographic assumptions. Indeed, in his "Ten Semi-Grand Challenges for Quantum Computing Theory", Aaronson emphasizes this important distinction, and asks whether there is some *instantiatable* function that separates the hybrid models from BQP. In this work, we resolve Aaronson's question in the affirmative for the *search* variants of these classes.

[2]Note that here and throughout the paper, the QNC oracle can output a string, unlike a decision oracle which outputs a bit.

[3]Note that the BPP oracle is not invoked coherently. Instead, it is invoked on outcomes resulting from intermediate measurements performed in the layers of the QNC_d circuit.

In contrast to separations between different models of computation running in polynomial time, such as P and NP or BPP and BQP, where several plausible candidates exist for separating the classes, the case for depth separations is much more subtle. As was already observed in [14], no standard cryptographic assumption is known to yield a separation between NC and P. The best candidates for such a separation are sequential compositions of hash functions (under the random oracle heuristic) as shown in [35] and the iterated exponentiation scheme of Rivest, Shamir and Wagner [40]. Thus, informally, the best we could hope for in terms of an instantiatable separation between the hybrid models and BQP is a separation in the random oracle model which could then be instantiated using cryptographic hash functions. We note that while there are known counterexamples to the random oracle heuristic [20], these are generally contrived and do not apply to protocols where the random oracle heuristic has been used in practice (and, in particular, are not known to apply to the setting we consider here). Indeed, it has been shown that certain protocols proven secure with respect to a random oracle can also be concretely instantiated using *correlation intractable hash functions* [19, 34]. We also note that separating the hybrid models from BPP, rather than BQP, in the random oracle model already follows from Aaronson's Fourier Fishing problem [1]. That problem, of sampling from the Fourier spectrum of the random oracle, is solvable in[4] QNC but not in BPP. Implicitly, this means that the hybrid search classes $\mathsf{BPP}^{\mathsf{QNC}}$ and $\mathsf{QNC}^{\mathsf{BPP}}$ are strictly larger than BPP relative to a random oracle.

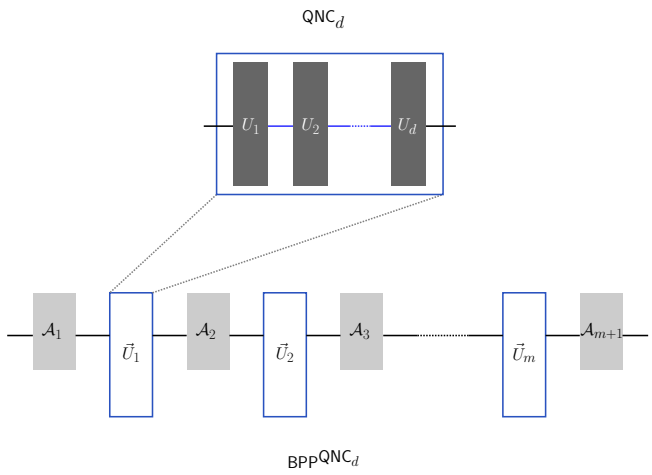
Our work is concerned not only with separations between the hybrid models and BQP in the random oracle model, but also with giving a comprehensive characterization of quantum depth in that model. To that end, we first re-examine Jozsa's conjecture and argue that the natural class associated to "d-depth quantum computation combined with polynomial-time classical computation" is not $\mathsf{BPP}^{\mathsf{QNC}_d} \cup \mathsf{QNC}_d{}^{\mathsf{BPP}}$, but $\mathsf{BPP}^{\mathsf{QNC}_d{}^{\mathsf{BPP}}}$. This is because, if one has the ability to perform $\mathsf{QNC}_d{}^{\mathsf{BPP}}$ computations, certainly it should also be possible to repeat this polynomially-many times as well as perform classical processing in between the runs. Note that $\mathsf{BPP}^{\mathsf{QNC}_d} \cup \mathsf{QNC}_d{}^{\mathsf{BPP}} \subseteq \mathsf{BPP}^{\mathsf{QNC}_d{}^{\mathsf{BPP}}}$ (in fact, we show strict containment). The separation we then obtain, relative to a random oracle, is $\mathsf{BPP}^{\mathsf{QNC}_d{}^{\mathsf{BPP}}} \subsetneq \mathsf{BQP}$, for any fixed $d \leq \mathrm{poly}(n)$. Going beyond this separation, we also show that the hybrid models $\mathsf{BPP}^{\mathsf{QNC}_d}$ and $\mathsf{QNC}_d{}^{\mathsf{BPP}}$ are separate from each other in both directions, relative to a random oracle (in fact, we show that $\mathsf{BPP}^{\mathsf{QNC}_{\mathcal{O}(1)}} \not\subseteq \mathsf{QNC}_d{}^{\mathsf{BPP}}$ and $\mathsf{QNC}^{\mathsf{BPP}}_{\mathcal{O}(1)} \not\subseteq \mathsf{BPP}^{\mathsf{QNC}_d}$), illustrating the subtle interplay between short-depth quantum computation and classical computation. Lastly, by combining the techniques that we develop with previous results on *proof of quantumness* protocols, we obtain *proof of quantum depth* protocols—protocols in which a BPP verifier, exchanging 2 messages[5] with an untrusted quantum prover, can certify that the prover has the ability to perform quantum computations of a minimum depth.

1.1 Main Results

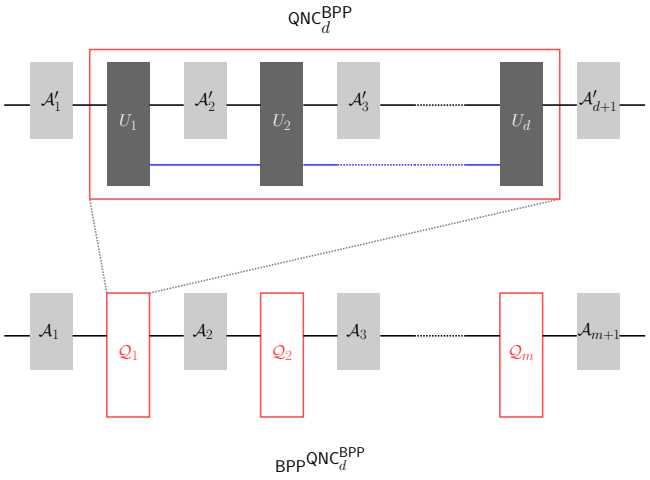
We now state our results more formally and provide some intuition about the proofs. We abuse the notation slightly and use the

[4]In fact in QNC^0, since queries to the oracle are assumed to have depth 1.

[5]2 messages in total or a 1 round protocol.



(a) Illustration of QNC_d and $\text{QNC}_d^{\text{BPP}}$ circuits.



(b) Illustration of $\text{QNC}_d^{\text{BPP}}$ and $\text{BPP}^{\text{QNC}_d^{\text{BPP}}}$ circuits.

Figure 1: The four hybrid quantum depth classes we consider. Blue wires carry qubits, black wires carry bits. Measurements are implicit and performed in the standard basis. U_is denote depth 1 unitaries, $\mathcal{A}_i$ and $\mathcal{A}'_i$ denote poly time classical algorithms.

standard *decision* complexity class names to refer to their *search* variants.

1.1.1 Lower Bounds on Quantum Depth. We first show the following separation.

Theorem 1 (informal). *Fix any function $d \leq \text{poly}(n)$. Then, relative to a random oracle,[6] it holds that $\text{BPP}^{\text{QNC}_d^{\text{BPP}}} \subsetneq \text{BQP}$.*

As motivated earlier, we take the class $\text{BPP}^{\text{QNC}_d^{\text{BPP}}}$ to capture computations performed by a combination of d-depth quantum computation and polynomial-depth classical computation. The interpretation of our result is that $\text{BPP}^{\text{QNC}_d^{\text{BPP}}}$ can be separated from BQP *using the least structured oracle possible*, a random oracle. Together with the (quantum) random oracle heuristic, by instantiating the oracle with a cryptographic hash function like SHA-2 or SHA-3, this yields the first plausible instantiation of a problem solvable in BQP but not in $\text{BPP}^{\text{QNC}_d^{\text{BPP}}}$. This provides a resolution to Aaronson's challenge. The main technical innovation that allows us to achieve the separation is a general lifting lemma that takes any problem separating BPP from BQP in the random oracle model, which additionally satisfies a property that we call *classical query soundness*, and constructs a problem separating $\text{BPP}^{\text{QNC}_d^{\text{BPP}}}$ and BQP. We show that several known problems satisfy this property. Our lifting lemma is inspired by [21], and crucially extends their analysis beyond highly structured oracles. We describe this lifting lemma more precisely in Subsection 1.2.1.

1.1.2 Proofs of Quantum Depth. It is natural to wonder whether Theorem 1 yields an efficient test to certify quantum depth, i.e. a *proof of quantum depth*. A proof of quantum depth is a more fine-grained version of a proof of quantumness: rather than distinguishing between quantum and classical computation, a proof of quantum depth protocol can distinguish between provers having large or small quantum depth. We show that instantiating our lifting lemma with a problem whose solution is *efficiently verifiable* immediately yields a proof of quantum depth. One such problem[7] is due to Yamakawa and Zhandry [43]. More precisely, we have the following.

Theorem 2 (informal). *Let n be the security parameter and fix any function $d \leq \text{poly}(n)$. In the random oracle model, there exists a two-message protocol between a poly-time classical verifier and a quantum prover such that,*

- *Completeness: There is a* BQP *prover which makes the verifier accept with probability at least $1 - \text{negl}(n)$*
- *Soundness: No malicious $\text{BPP}^{\text{QNC}_d^{\text{BPP}}}$ prover can make the verifier accept with probability greater than $\text{negl}(n)$.*

We emphasise that considering protocols with more than two messages leads to difficulties in formalising the notion of quantum depth. For instance, one can construct protocols where the prover is forced to hold r single qubit states and subsequently measures them. Information about the basis in which to measure each of these qubits is sent one at a time by the verifier over r messages (the verifier waits for the response to each measurement, before sending the next basis). The measurement results are used by the verifier to ensure soundness (each qubit is measured in its preparation basis and so the outcomes are completely determined). It is not hard to show that if the prover measures these qubits without knowing the measurement basis, it cannot succeed except with negligible probability. If one attempts to model the prover as a $\text{BPP}^{\text{QNC}_d}$ or $\text{QNC}_d^{\text{BPP}}$ circuit, then, because of the delay between messages, it appears that $d \geq r$ is necessary. However, this can be seen as an artefact of the modelling choice: in practice, the prover only needs

[6] Here, as well as in all subsequent results, the statements hold with probability 1 over the choice of the random oracle. In addition, queries to the oracle are viewed as having depth 1 (discussed later).

[7] We remark that, if one is only concerned with the complexity-theoretic separation of Theorem 1, and not with efficient verification, then a much simpler problem (CollisionHashing described later) suffices.

d single qubit quantum computers with quantum depth 1 where the last gate can be delayed until the appropriate message is received in order to pass the test. Essentially, this approach only tests the prover's ability to maintain the coherence of the qubits it received, without actually testing the depth of the circuit it has to perform. In Subsection 1.3, we discuss a possible resolution that captures quantum depth in the interactive setting.

1.1.3 Tighter Bounds. While Theorem 1 establishes that $\mathsf{BPP}^{\mathsf{QNC}_d^{\mathsf{BPP}}}$ does not capture the computational power of BQP for any fixed $d \leq \mathrm{poly}(n)$, it is not a priori clear if, for instance, $\mathsf{BPP}^{\mathsf{QNC}_{2d+\mathcal{O}(1)}^{\mathsf{BPP}}}$ is strictly larger than $\mathsf{BPP}^{\mathsf{QNC}_d^{\mathsf{BPP}}}$. Indeed, we show that the answer is affirmative.

Theorem 3 (informal). *Fix any function $d \leq \mathrm{poly}(n)$. Relative to a random oracle, it holds that*[8] $\mathsf{BPP}^{\mathsf{QNC}_d^{\mathsf{BPP}}} \subsetneq \mathsf{BPP}^{\mathsf{QNC}_{2d+\mathcal{O}(1)}^{\mathsf{BPP}}}$.

Formally, the theorems treat a call to the quantum random oracle as a depth-1 quantum gate. In practice, if instead the gate requires depth ℓ, then d can be replaced by $d\ell$. We remark that there exist hash functions that are thought to be quantum-secure which require only logarithmic depth to evaluate [7, 36]. Further, there is reason to believe that such hash functions could also be constructed in $\ell = \mathcal{O}(1)$ depth. In particular, if one is only concerned with specific cryptographic properties (such as collision resistance), then generic constructions are known which convert log-depth hash functions into ones that require only constant depth [9].

1.1.4 Separations between Hybrid Quantum Depth Classes. While both $\mathsf{BPP}^{\mathsf{QNC}}$ and $\mathsf{QNC}^{\mathsf{BPP}}$ capture some notion of a hybrid between efficient classical computation and shallow quantum computation, the relationship between the two is not immediately clear. To get a slightly better intuition about the two models, one can think of $\mathsf{BPP}^{\mathsf{QNC}}$ as capturing an efficient computation that contains *polynomially many* shallow quantum circuits (separated by measurements and classical computation). On the other hand, one can think of $\mathsf{QNC}^{\mathsf{BPP}}$ as a *single* shallow quantum circuit, where one is allowed to make partial measurements of some of the wires, and choose the next gates *adaptively*. While it may not be surprising that there exist problems that can be solved in $\mathsf{BPP}^{\mathsf{QNC}}$ but not in $\mathsf{QNC}^{\mathsf{BPP}}$, it turns out that the two classes are in fact incomparable—each class contains problems that the other does not, relative to a random oracle.

Theorem 4 (informal). *Fix any function $d \leq \mathrm{poly}(n)$. Relative to a random oracle, it holds that* $\mathsf{BPP}^{\mathsf{QNC}_{\mathcal{O}(1)}} \not\subseteq \mathsf{QNC}_d^{\mathsf{BPP}}$ *and* $\mathsf{QNC}_{\mathcal{O}(1)}^{\mathsf{BPP}} \not\subseteq \mathsf{BPP}^{\mathsf{QNC}_d}$.

The second separation is arguably more surprising. It says that, relative to a random oracle, there are problems that can be solved by a *single* shallow (in fact, constant-depth) quantum circuit *with* adaptive measurements but cannot be solved by circuits with *polynomially many* shallow quantum circuits *without* adaptive measurements. The problem that shows $\mathsf{QNC}_{\mathcal{O}(1)}^{\mathsf{BPP}} \not\subseteq \mathsf{BPP}^{\mathsf{QNC}_d}$ is a variant of the proof of quantumness from [17]. The key technical innovation to achieve this separation is a theorem that characterises the structure of strategies that succeed in the protocol of [17] (this is discussed further in Section 1.2.2). This "structure theorem" crucially strengthens a similar theorem from [26], and may be of independent interest.

Finally, we examine the relationship between $\mathsf{BPP}^{\mathsf{QNC}_d} \cup \mathsf{QNC}_d^{\mathsf{BPP}}$ and $\mathsf{BPP}^{\mathsf{QNC}_d^{\mathsf{BPP}}}$. By definition, it is manifest that $\mathsf{BPP}^{\mathsf{QNC}_d} \cup \mathsf{QNC}_d^{\mathsf{BPP}} \subseteq \mathsf{BPP}^{\mathsf{QNC}_d^{\mathsf{BPP}}}$. Even though $\mathsf{QNC}_d^{\mathsf{BPP}}$ and $\mathsf{BPP}^{\mathsf{QNC}_d}$ are incomparable, it is conceivable that their union captures any reasonable notion of quantum depth d. We show that this is not the case.

Theorem 5 (informal). *Fix any function $d \leq \mathrm{poly}(n)$. Relative to a random oracle, it holds that* $\mathsf{BPP}^{\mathsf{QNC}_{\mathcal{O}(1)}^{\mathsf{BPP}}} \not\subseteq \mathsf{BPP}^{\mathsf{QNC}_d} \cup \mathsf{QNC}_d^{\mathsf{BPP}}$.

In words, the latter theorem asserts that a computation consisting of polynomially many layers of *constant*-depth quantum circuits with adaptive control cannot be simulated by quantum circuits with d depth which are either adaptive (but consisting of a single d-depth quantum circuit) or consisting of many d-depth quantum circuits (but without adaptive control).

1.1.5 Summary . Table 1 lists our lower bounds on quantum depth, and Table 2 lists the separations among the hybrid classes.

1.2 Main Technical Contributions

1.2.1 Lifting Lemmas. One of the main technical contributions of our work is to prove *two* general lifting lemmas. These lemmas take problems, defined relative to a random oracle, that are classically hard (in a stronger sense, defined next) and create new problems which are, in addition, hard for specific hybrid quantum depth classes. We describe these lifting lemmas a bit more precisely.

We say that a problem (defined with respect to the random oracle) is *classical query sound* if the following holds: any (potentially unbounded time) algorithm which makes only polynomially many *classical* queries to the random oracle (i.e. no superposition queries), succeeds at solving the problem with at most negligible probability. It turns out that the problem introduced by YZ satisfies this property. Another problem which satisfies this property is inspired by the proof of quantumness protocol defined by Brakerski et al. [17] (hereafter referred to as BKVV).[9] For such problems, the following holds.

Lemma 6 (informal, simplified). *There is a procedure*[10] *that takes a classical query sound problem $\mathcal{P} \in \mathsf{BQP}$ and creates a new problem $\mathcal{P}' := d\text{-Rec}[\mathcal{P}]$, such that $\mathcal{P}' \notin \mathsf{BPP}^{\mathsf{QNC}_d^{\mathsf{BPP}}}$ and $\mathcal{P}' \in \mathsf{BQP}$.*

Observe that this lemma makes the problem hard for *the most general notion of quantum depth* we have considered. To give some intuition about how it is derived, suppose we have a problem $\mathcal{P}$ which is classical query sound and denote the random oracle as H. Then $\mathcal{P}' = d\text{-Rec}[\mathcal{P}]$ is the same problem, defined with respect to *a sequential composition of $d+1$ random oracles*, $\tilde{H} = H_d \circ \cdots \circ H_0$. In essence, we have substituted H with $\tilde{H}$. This new problem will retain classical query soundness, as $\tilde{H}$ behaves like a random oracle. But in addition, we have now made it so that querying $\tilde{H}$ effectively requires depth $d+1$. As QNC_d has depth d, only the BPP parts

[8] and more generally, that $\mathsf{QNC}_{2d+\mathcal{O}(1)} \not\subseteq \mathsf{BPP}^{\mathsf{QNC}_d^{\mathsf{BPP}}}$.

[9] Which we refer to as CollisionHashing later.

[10] d-Rec$[\cdot]$ is meant to be short for d-Recursive.

Table 1: (Simplified) Bounds on quantum depth. Separations are with respect to the random oracle and $d \leq \mathrm{poly}(n)$ is any fixed function of the input size.

Result	Remarks
$\mathsf{BPP}^{\mathsf{QNC}^{\mathsf{BPP}}} \subsetneq \mathsf{BQP}$	Refutes Jozsa's conjecture in the random oracle model
$\mathsf{BPP}^{\mathsf{QNC}^{\mathsf{BPP}}_{d}} \subsetneq \mathsf{BPP}^{\mathsf{QNC}^{\mathsf{BPP}}_{2d+\mathcal{O}(1)}}$	Fine grained advantage of quantum depth

Table 2: (Simplified) Separations of hybrid quantum depth with respect to the random oracle. The results hold, not only for log **but for any fixed polynomially-bounded function.**

Result	Physical Interpretation
$\mathsf{BPP}^{\mathsf{QNC}_{\mathcal{O}(1)}} \not\subseteq \mathsf{QNC}^{\mathsf{BPP}}$	Running *poly many constant* depth quantum circuits (with *no* adaptive measurements) cannot be simulated by running a *single* log depth quantum circuit *with* adaptive measurements.
$\mathsf{QNC}^{\mathsf{BPP}}_{\mathcal{O}(1)} \not\subseteq \mathsf{BPP}^{\mathsf{QNC}}$	Running a *single constant* depth quantum circuit *with* adaptive measurements cannot be simulated by running *poly many* log depth quantum circuits (with *no* adaptive measurements).
$\mathsf{BPP}^{\mathsf{QNC}^{\mathsf{BPP}}_{\mathcal{O}(1)}} \not\subseteq \mathsf{BPP}^{\mathsf{QNC}} \cup \mathsf{QNC}^{\mathsf{BPP}}$	Evidence that it is not enough to consider $\mathsf{BPP}^{\mathsf{QNC}}$ and $\mathsf{QNC}^{\mathsf{BPP}}$ when studying quantum depth. Running *poly many constant* depth quantum circuits *with* adaptive measurements cannot be simulated using either (a) *poly many* log depth quantum circuits with *no* adaptive measurements, or by (b) a *single* log *depth* quantum circuit *with* adaptive measurements.

of $\mathsf{BPP}^{\mathsf{QNC}^{\mathsf{BPP}}_{d}}$ will be able to query $\tilde{H}$. We can therefore simulate the $\mathsf{BPP}^{\mathsf{QNC}^{\mathsf{BPP}}_{d}}$ algorithm with an exponential time algorithm that is limited to polynomially many queries to $\tilde{H}$. By classical query soundness, such an algorithm cannot solve $\mathcal{P}'$, which yields the desired result.

This was a simplified description of our result. In fact, we show a more refined statement that relates the depth required to solve $\mathcal{P}'$ to the depth required to solve $\mathcal{P}$. In addition, arguing that $\tilde{H}$ behaves like a random oracle and that QNC_d cannot query $\tilde{H}$ requires a careful and more involved analysis. We use Lemma 6 to establish Theorem 3.

Our second lifting lemma produces a problem that is hard for $\mathsf{QNC}^{\mathsf{BPP}}_{d}$, starting from a problem that satisfies what we call *offline soundness*. Consider a two phase algorithm consisting of: *an online phase* which is a poly-time classical algorithm *with access* to the random oracle followed by *an offline phase* which is an unbounded(-time) algorithm with *no access* to the random oracle. Then, *offline soundness* requires that no such two phase algorithm succeeds at solving the problem with non-negligible probability. It turns out, again, that both YZ and BKVV satisfy this property.

Lemma 7 (informal)**.** *There is a procedure*[11] *which takes a problem* $\mathcal{P} \in \mathsf{QNC}_{\mathcal{O}(1)}$ *with offline soundness and creates a new problem* $\mathcal{P}' := d\text{-}\mathsf{Ser}[\mathcal{P}]$ *such that* $\mathcal{P}' \notin \mathsf{QNC}^{\mathsf{BPP}}_{d}$ *and* $\mathcal{P}' \in \mathsf{BPP}^{\mathsf{QNC}_{\mathcal{O}(1)}}$.

Again, we actually show a slightly more general upper bound which depends on the depth required to solve $\mathcal{P}$. We use Lemma 7 to establish $\mathsf{BPP}^{\mathsf{QNC}_{\mathcal{O}(1)}} \not\subseteq \mathsf{QNC}^{\mathsf{BPP}}_{d}$ (first separation of Theorem 4). Establishing the other direction ($\mathsf{QNC}^{\mathsf{BPP}}_{\mathcal{O}(1)} \not\subseteq \mathsf{BPP}^{\mathsf{QNC}_d}$) is quite involved and relies heavily on the structure of the problem we consider (explained below). Consequently, it is unclear whether there exists a general lifting lemma that yields hardness for $\mathsf{BPP}^{\mathsf{QNC}_d}$.

We remark that, by using Lemma 7 to lift the problem that yields $\mathsf{QNC}^{\mathsf{BPP}}_{\mathcal{O}(1)} \not\subseteq \mathsf{BPP}^{\mathsf{QNC}_d}$, we also obtain Theorem 5, i.e. $\mathsf{BPP}^{\mathsf{QNC}^{\mathsf{BPP}}_{1}} \not\subseteq \mathsf{BPP}^{\mathsf{QNC}_d} \cup \mathsf{QNC}^{\mathsf{BPP}}_{d}$.

1.2.2 A Structure Theorem for [17]. Another technical contribution of this work, which may be of independent interest, is to prove a theorem characterizing the structure of strategies that are successful at the proof of quantumness from [17]. This theorem is a crucial strengthening of a theorem from [26]. We employ this theorem as an intermediate step to establish the hybrid separation, $\mathsf{QNC}^{\mathsf{BPP}}_{\mathcal{O}(1)} \not\subseteq \mathsf{BPP}^{\mathsf{QNC}_d}$.

Recall, informally, that the proof of quantumness from [17] requires the prover to succeed at the following task: given access to a 2-to-1 function g, and to a random oracle H with a one-bit output, find a pair (y, r) such that

$$r \cdot (x_0 \oplus x_1) \oplus H(x_0) \oplus H(x_1) = 0\,,$$

where $\{x_0, x_1\} = g^{-1}(y)$. This can be solved in $\mathsf{QNC}_{\mathcal{O}(1)}$ as follows:

(i) Evaluate g on a uniform superposition of inputs, yielding $\sum_x |x\rangle\,|g(x)\rangle$,
(ii) Measure the image register obtaining some outcome y and a state $(|x_0\rangle + |x_1\rangle)\,|y\rangle$,

[11] d-Ser$[\cdot]$ is meant to be short for d-Serial.

(iii) Query a phase oracle for H to obtain $((-1)^{H(x_0)}|x_0\rangle + (-1)^{H(x_1)}|x_1\rangle)|y\rangle$,
(iv) Make a Hadamard basis measurement of the first register, obtaining outcome r.

Informally, our structure theorem establishes that querying at a superposition of pre-images is essentially *the only way to succeed* (provided finding a collision for g is hard—this is the case when g is a trapdoor claw-free function, as in [17], but more generally our theorem also holds e.g. when g is a uniformly random 2-to-1 function). Denote by n the bit-length of strings in the domain of g.

Theorem 8 (informal). *Let P be any* BQP *prover that succeeds with $1-\mathrm{negl}(n)$ probability at the proof of quantumness protocol from [17], by making q queries to the oracle H. Then, with $1-\mathrm{negl}(n)$ probability over pairs (H,y), the following holds. Let $p_{y|H}$ be the probability that P^H outputs y, and let x_0,x_1 be the pre-images of y. Then, for all $b\in\{0,1\}$, there exists $i\in[q]$ such that the state of the query register of P^H right before the i-th query has weight $\frac{1}{2}p_{y|H}\cdot(1-\mathrm{negl}(n))$ on x_b.*

Note that a version of the above theorem that applies to provers who win with probability non-negligibly greater than $\frac{1}{2}$ also holds (but we stated the close-to-ideal version for simplicity). We provide a sketch of how this theorem is used in the proof of $\mathsf{QNC}^{\mathsf{BPP}}_{\mathcal{O}(1)} \not\subseteq \mathsf{BPP}^{\mathsf{QNC}_d}$ in Subsection 2.2.2.

1.3 Discussion and Open Problems

Separations of decision classes and going beyond the random oracle model. Our separations are with respect to *search* classes. What about decision classes? The Aaronson-Ambainis conjecture [2] states that one cannot separate the decision versions of BPP and BQP in the random oracle model. Assuming the conjecture is true, this also implies that there cannot be a separation between the decision versions of $\mathsf{BPP}^{\mathsf{QNC}^{\mathsf{BPP}}_d}$ and BQP in the random oracle model. Thus, for the case of decision classes, it would be interesting to see whether there exists some *structured* oracle separation for which the oracle can be instantiated based on a suitable computational assumption. Both [21] and [28] suggested the possibility of using either *virtual black-box (VBB) obfuscation* or *indistinguishability obfuscation (iO)* to instantiate their oracles. For the former, one difficulty is that it is known that one cannot VBB obfuscate general functions [12, 13]. Irrespective of this fact, however, there is a more general obstacle that applies to either type of obfuscation. As we mentioned in the introduction, it was noted in [14] that no standard cryptographic assumption (not even the existence of iO) is known to imply a depth separation (even between P and NC). It therefore seems that one either has to use non-standard assumptions to prove the separations or make a significant advancement either in refuting the Aaronson-Ambainis conjecture or in proving $\mathsf{P}\neq\mathsf{NC}$ from a standard cryptographic assumption.

The limitations of using cryptographic assumptions to prove depth separations also apply to the search classes. In some sense, separations with respect to a random oracle are the best we can hope for, given the current techniques in computational complexity theory. This is peculiar, since one would imagine that using more structured non-oracular problems would allow one to prove stronger separations. The random oracle is the least structured type of oracle, but the fact that it is an oracle helps in establishing provable lower bounds.

Further questions in the random oracle model. Recall that our lifting lemma, turning a classical query sound problem into a problem which can separate $\mathsf{BPP}^{\mathsf{QNC}^{\mathsf{BPP}}_d}$ and BQP in the random oracle model, could be instantiated with the proof of quantumness from [43]. The resulting problem inherits the property that solutions can be publicly verified. We thus obtain a proof of quantum depth that is publicly verifiable. Unlike a proof of quantumness, the proof of quantum depth is sound against a family of *quantum* provers (in this case $\mathsf{BPP}^{\mathsf{QNC}^{\mathsf{BPP}}_d}$ provers). Can we further push this quantum soundness to obtain verification of BQP with a BPP verifier relative to a random oracle?

We have also seen that making use of a problem inspired by the Brakerski et al. [17] proof of quantumness allows us to prove more fine grained separations between hybrid classes. It is then natural to ask, whether these separations also yield *finer grained proofs of quantum depth* (which are sound against $\mathsf{BPP}^{\mathsf{QNC}^{\mathsf{BPP}}_d}$ provers and complete for a $\mathsf{BPP}^{\mathsf{QNC}^{\mathsf{BPP}}_{2d+\mathcal{O}(1)}}$ prover). This does not immediately follow from our results, as the problem we construct from BKVV is not efficiently verifiable, and our current techniques do not directly extend to the computationally-bounded setting. We therefore leave this as an open problem.

Generalizing beyond $\mathsf{BPP}^{\mathsf{QNC}^{\mathsf{BPP}}_d}$. We have argued that $\mathsf{BPP}^{\mathsf{QNC}^{\mathsf{BPP}}_d}$ is the most natural class capturing the notion of d-depth quantum computation, combined with polynomial-depth classical computation. However, for the purpose of *certifying* quantum depth, as we have mentioned earlier (and as we discuss in more detail in Example 12), the situation becomes more subtle when the certification protocol involves *interaction*. We therefore propose that any protocol which establishes quantum depth d and uses r rounds of interaction should be sound against at least an r level generalization of $\mathsf{BPP}^{\mathsf{QNC}^{\mathsf{BPP}}_d}$ (e.g. a 2 level generalization with quantum depth d would be $\mathsf{BPP}^{\mathsf{QNC}_d{}^{\mathsf{BPP}^{\mathsf{QNC}^{\mathsf{BPP}}_d}}}$ — here 2 counts the number of times QNC_d appears in the tower of complexity classes, so that an r level generalisation would have r appearances of QNC_d). In our case, since the proof of depth protocols are single-round, we show the necessary soundness against a $\mathsf{BPP}^{\mathsf{QNC}^{\mathsf{BPP}}_d}$ prover.

Of course, there are other possible ways to define hybrid d-depth quantum-classical computation. For instance, one can define the class QDepth_d of problems solved by polynomial sized circuits with quantum and classical gates where the key constraint is that *the longest path connecting quantum gates (with quantum wires) is at most d*. We expect our separating problems (and d-Rec$[\mathscr{P}]$ in general, for classical query sound $\mathscr{P}$) to not be contained in QDepth_d. We also expect that $\mathsf{Q}_d\mathsf{H}\subsetneq\mathsf{QDepth}_d$, but we leave the proof as future work.

1.4 Previous Work

We compare our results to the previous works [21], [28], [11], and [22].

Comparison to [21], [28] and [11]. Compared to previous work on the topic, our work gives a comprehensive treatment of the complexity of hybrid quantum-classical computation.

As mentioned earlier, the primary difference compared to [21] and [28] is that all of our separations are with respect to a random oracle, rather than with respect to highly structured oracles. However, one caveat is that our separations are for search problems. Our contribution is also conceptual. We propose $\mathsf{BPP}^{\mathsf{QNC}_d^{\mathsf{BPP}}}$ as the appropriate model to capture "d-depth quantum computation combined with polynomial-time classical computation". While [21] and [28] showed that $\mathsf{BPP}^{\mathsf{QNC}_d} \cup \mathsf{QNC}_d^{\mathsf{BPP}} \not\subseteq \mathsf{BQP}$, we show the stronger result that $\mathsf{BPP}^{\mathsf{QNC}_d^{\mathsf{BPP}}} \not\subseteq \mathsf{BQP}$.

Our work also shows separations between different hybrid models. Such separations were considered in [11], where they are again proven only with respect to highly structured oracles.

In terms of techniques, we take inspiration and ideas from both [21] and [11]. In particular we build on two key ideas—the sampling argument and domain hiding. One of the main contributions of our analysis is to abstract and generalise these techniques beyond their original scope which was tailored to specific promise problems. While most of our results build on these techniques, we also point out that to prove the separation between the hybrid models $\mathsf{QNC}_{\mathcal{O}(1)}^{\mathsf{BPP}} \not\subseteq \mathsf{BPP}^{\mathsf{QNC}_d}$ we use entirely different ideas. In particular, as an intermediate step, we establish a theorem that characterizes the structure of strategies that succeed in the proof of quantumness of BKVV, which may be of independent interest.

Comparison to [22]. The work of [22] was the first to consider proofs of quantum depth. However, the notion of soundness that they propose, and their corresponding protocol (in the single prover setting), suffers from the issues that we discussed after Theorem 2 (and in Example 12 below).

In particular, their protocol can be spoofed by a d level tower of $\mathsf{BPP}^{\mathsf{QNC}_{\mathcal{O}(1)}^{\mathsf{BPP}}}$ (as described in Subsection 1.3). In practical terms, this means that it can be spoofed by running several constant depth quantum computers in parallel, provided the "idle coherence time" of each quantum computer is longer than the time that elapses between messages in the protocol. In contrast, our proof of depth protocol does not suffer from this issue and can be used to certify that the prover is able to perform computations "beyond" $\mathsf{BPP}^{\mathsf{QNC}_d^{\mathsf{BPP}}}$.

2 TECHNICAL OVERVIEW

Here we give a high level technical overview of the paper.

2.1 Bounds on Quantum Depth

In this subsection, we describe the proof of Theorem 1. As mentioned previously, our main technical contribution is a general lifting lemma that takes any problem separating BPP from BQP in the random oracle model, which additionally satisfies a property that we call *classical query soundness*, and constructs a problem separating $\mathsf{BPP}^{\mathsf{QNC}_d^{\mathsf{BPP}}}$ and BQP. We first explain the key idea behind this construction. To be concrete, after describing the key idea, we restrict to an NP search problem due to Yamakawa and Zhandry [43], which satisfies classical query soundness (this problem is particularly appealing because it is in NP, and thus solutions can be publicly verified, however we emphasize that other known search problems that are not in NP can also be used for the separation). We then build towards a proof that this problem is not in $\mathsf{BPP}^{\mathsf{QNC}_d^{\mathsf{BPP}}}$ by considering hardness for the three special cases QNC_d, $\mathsf{QNC}_d^{\mathsf{BPP}}$ and $\mathsf{BPP}^{\mathsf{QNC}_d}$. The desired result is obtained by combining the ideas in these three cases.

Let $\mathcal{P}$ be a (search) problem, defined relative to a random oracle H, that separates BPP from BQP. Suppose that $\mathcal{P}$ is such that it requires *quantum* access to H in order to be solved with polynomially many queries (*classical query soundness* will eventually require a bit more than this). As mentioned in Subsection 1.2.1, the first natural idea to lift this to a separation between low quantum depth and polynomial quantum depth is to *replace the evaluation of H with a sequential evaluation of random oracles*. For example, suppose that originally $H : \Sigma \to \{0,1\}^n$. Then, let $H_0, \ldots, H_{d-1} : \Sigma \to \Sigma$, and $H_d : \Sigma \to \{0,1\}^n$ be random oracles. Define $\tilde{H} = H_d \circ \cdots \circ H_0$. Now, let $\mathcal{P}'$ be the problem that is identical to $\mathcal{P}$ except that it is relative to $\tilde{H}$. Then, it is natural to imagine that $\mathcal{P}'$ requires quantum depth at least $d+1$ to solve. This idea does not quite work right away, since $\tilde{H}$, as defined, is not actually a uniformly random oracle any more. This is because with every H_i that is added, the number of collisions in $\tilde{H}$ increases (on average). To remedy this, one could assume that $H_0, \ldots, H_{d-1}$ are random *permutations* (although note that random permutations cannot be generically constructed from random oracles). A similar idea works in a different setting, for arguing about the post-quantum security of "proofs of sequential work" [15]. However, in our case, the analysis is complicated by the fact that we consider hybrid models. CCL were the first to consider a variant of sequential hashing (sequential permutations), in the context of hybrid models. However, their analysis only works for certain structured oracles. In this work, we adapt their ideas to the random oracle setting and overcome these difficulties.

Lifting $\mathcal{P} \notin \mathsf{BPP}$ to $\tilde{\mathcal{P}} \notin \mathsf{BPP}^{\mathsf{QNC}_d^{\mathsf{BPP}}}$. Given a problem $\mathcal{P}$ with respect to H, we define the problem $\tilde{\mathcal{P}} = d\text{-Rec}[\mathcal{P}]$ to be $\mathcal{P}$ with respect to $\tilde{H} = H_d \circ \cdots \circ H_0$ where $H_0, \ldots, H_d$ are independent random oracles with the following domains and co-domains: $H_0 : \Sigma \to \Sigma^{d'}$, $H_i : \Sigma^{d'} \to \Sigma^{d'}$ for $i \in \{1 \ldots d-1\}$, and $H_d : \Sigma^{d'} \to \{0,1\}^n$ with $d' = 2d+5$.

Notice that H_0 is not surjective, as its codomain is much larger than its image.[12] In fact, this is also true for $H_i \circ \cdots \circ H_0$, for all $i < d$. This and the fact that the H_i functions are random, have two important consequences. First, it means that with high probability $H_{d-1} \circ \cdots \circ H_0$ is injective and so $\tilde{H}$ behaves like a random oracle. Consequently, $\mathcal{P}'$ inherits the soundness and completeness of $\mathcal{P}$. Second, it means that one can apply a "domain hiding" technique, which, at a high level, works as follows. One way of evaluating $\tilde{H}$ at $x \in \Sigma$ is to sequentially compose $H_0, H_1, \ldots, H_d$ which would require depth $d+1$. Intuitively, it seems unlikely that there is a more depth efficient way of evaluating $\tilde{H}$ because the domain on which the H_i's need to be evaluated (which is $H_{i-1} \circ \cdots \circ H_0(\Sigma)$) is getting shuffled and lost in an exponentially larger domain (which is $\Sigma^{d'}$). Therefore, even though one has access to all $\mathcal{L} = (H_0, H_1, \ldots, H_d)$ oracles at the first layer of depth, one only knows that H_0 needs to be queried at Σ but the algorithm has no information about where

[12] We sometimes refer to this fact by saying that the function is "expanding".

the relevant domains of $H_1 \dots H_d$ are. At the second depth layer, the algorithm can learn $H_0(\Sigma)$ and so learns where to query H_1 but, and this needs to be shown, it still does not know where the relevant domains of $H_2, \dots H_d$ are. By starting with a sufficiently large expansion, i.e. a sufficiently large $d' > d$, this argument can be repeated until depth d where the relevant domain of H_d still remains hidden. Thus, even though $\mathcal{P}'$ can potentially be solved with $d+1$ depth, it cannot be solved with depth d. This is the basic idea behind why the problem is not in QNC_d. Instead of working with $\mathcal{P}$ and d-Rec$[\mathcal{P}]$ abstractly, we consider the following concrete problem.

2.1.1 d-CodeHashing / The problem. We refer to the problem introduced by Yamakawa and Zhandry [43] as CodeHashing in this work. The problem is stated in terms of a family of error-correcting codes called *suitable codes*. For our purposes, it suffices to think of suitable codes as a family of sets $\{C_\lambda\}_\lambda$ where each C_λ is a set of codewords $\{(\mathbf{x}_1, \dots \mathbf{x}_n)\}$ with each coordinate $\mathbf{x}_i$ belonging to some alphabet Σ. The size of this alphabet, $|\Sigma| = 2^{\lambda^{\Theta(1)}}$ is exponential in λ, and the number of components $n = \Theta(\lambda)$ essentially equals λ. CodeHashing is defined as follows.

Definition 9 (CodeHashing; informal)**.** Let $\{C_\lambda\}_\lambda$ be a suitable code and let $H : \{0,1\}^{\log n} \times \Sigma \to \{0,1\}$ be a random oracle. Given a description of the suitable code (e.g. as parity check matrices) and oracle access to H, on input 1^λ, the problem is to find a codeword $\mathbf{x} = (\mathbf{x}_1 \dots \mathbf{x}_n) \in C_\lambda$ such that[13] $H(i||\mathbf{x}_i) = 1$ for all $i \in \{1 \dots n\}$.

Note that CodeHashing is an NP search problem, since from, e.g. the parity check matrix of the code, it is easy to verify that $\mathbf{x}$ is indeed a codeword and with a single parallel query (n queries in total) to H, one can check that it hashes correctly.

YZ shows that CodeHashing satisfies the following two properties.

Lemma 10 (Paraphrased from YZ)**.** *The following hold.*

- Completeness: *There is a QPT machine which solves* CodeHashing *with probability* $1 - \mathrm{negl}(\lambda)$ *and makes only one parallel query to* H.
- Soundness: *Every (potentially unbounded time) classical circuit which makes at most* 2^{λ^c} *queries to* H*, with* $c < 1$*, solves* CodeHashing *with probability at most* $2^{-\Omega(\lambda)}$.

The fact that soundness holds against *unbounded time* classical circuits which make only poly-many queries to the random oracle is essential in proving that $\mathsf{BPP}^{\mathsf{QNC}^{\mathsf{BPP}}} \subsetneq \mathsf{BQP}$. Applying our lifting map, d-Rec$[\mathcal{P}]$ on CodeHashing we obtain the following.[14]

Definition 11 (d-CodeHashing; informal)**.** Let $\{C_\lambda\}_\lambda$ be a suitable code, and $\tilde{H} := H_d \circ \dots \circ H_1 \circ H_0$, where $H_0, \dots, H_d$ are as in Section 2.1. Given a description of the suitable code, access to random oracles $\mathcal{L} = (H_0 \dots H_d)$, on input 1^λ, find a codeword for all $i \in \{1 \dots n\}$.

To convey the key ideas behind the proof that d-CodeHashing $\notin \mathsf{BPP}^{\mathsf{QNC}_d^{\mathsf{BPP}}}$, we first consider the QNC_d case in some more detail, and extend the analysis to $\mathsf{QNC}_d^{\mathsf{BPP}}$. We then analyse the $\mathsf{BPP}^{\mathsf{QNC}_d}$ case, which uses a technique called the "sampling argument" due to [27]. These ideas were first considered in the structured oracle setting by [21] and [11]. We adapt them to show d-CodeHashing $\notin \mathsf{BPP}^{\mathsf{QNC}_d^{\mathsf{BPP}}}$ relative to a random oracle.

2.1.2 d-CodeHashing $\notin \mathsf{QNC}_d$.

Base sets. We started our discussion in Subsection 2.1 by observing that the analysis is simplified by taking $H_0 \dots H_{d-1}$ to be injective functions. However, for a large enough d', it is not hard to see that this is indeed the case on an appropriately restricted domain. The sets which describe this restricted domain are chosen randomly. We call them *base sets* and denote them by $S_{01}, \dots S_{0d}$ (corresponding to $H_1, \dots H_d$ respectively). Observe that H_0 maps Σ to $\Sigma^{d'}$ (which is exponentially larger than Σ; recall that $|\Sigma| = 2^{\lambda^{\Theta(1)}}$) and, since H_0 is a random function, the probability that this mapping is injective is $1 - \mathrm{negl}(\lambda)$. Pick any set $S_{01} \subseteq \Sigma^{d'}$ uniformly at random in the domain of H_1 subject to two constraints: (1) it includes $H_0(\Sigma)$, i.e. the domain of H_1 on which the value of $\tilde{H}$ depends, and (2) its size is $|S_{01}| = |\Sigma|^{d+2}$. The first constraint ensures that the domain we care about is included in the base sets and the second ensures that: (a) $|S_{01}|$ is exponentially smaller than $|\Sigma|^{d'}$ and (b) $|S_{01}|$ is large enough for applying "domain hiding" as mentioned above. Define $S_{0i} := H_{i-1}(\dots H_1(S_{01}) \dots)$ to be the image of S_{01} through the first 1 to $(i-1)$'th oracles for $i \in \{2 \dots d\}$. Let E denote the event that H_0 is injective and $H_1 \dots H_{d-1}$ are injective on the base sets. We show that E (given our choice for d'), occurs with overwhelming probability. In the subsequent discussion, we assume that base sets have been selected and that E occurs.

Proof idea. We describe the proof that d-CodeHashing $\notin \mathsf{QNC}_d$ in some more detail, which implements the previously described "domain hiding" idea and proceeds via a hybrid argument. Denote a QNC_d circuit that makes d parallel calls to the oracle $\mathcal{L} = (H_0, \dots H_d)$ by $U_{d+1} \circ \mathcal{L} \circ U_d \dots U_2 \circ \mathcal{L} \circ U_1 \circ \rho_0$. Here, ρ_0 is some initial state, U_i are single layered unitaries, and the composition is meant to act as conjugation, i.e. $U_1 \circ \rho_0 = U_1 \rho_0 U_1^\dagger$. We show that the behaviour of such a circuit, i.e. its probability of outputting a valid answer, is negligibly close to the behaviour of another circuit $U_{d+1} \circ \mathcal{M}_d \circ U_d \dots U_2 \circ \mathcal{M}_1 \circ U_1 \circ \rho_0$ where $\mathcal{M}_1, \dots \mathcal{M}_d$ are "shadow oracles" corresponding to $\mathcal{L}$ that contain no information about the values taken by $\tilde{H}$ on Σ. Clearly then, this circuit cannot be solving d-CodeHashing because it never queries $\tilde{H}$. This in turn means that the original circuit also cannot solve d-CodeHashing, which implies d-CodeHashing $\notin \mathsf{QNC}_d$. It remains to define $\mathcal{M}_1 \dots \mathcal{M}_d$ and to argue that the two circuits have essentially the same behaviour. Using a hybrid argument, one can establish the latter by showing that the following are close in trace distance: (1) $\mathcal{L} \circ U_1 \circ \rho_0$ and $\mathcal{M}_1 \circ U_1 \circ \rho_0$, (2) $\mathcal{L} \circ U_2 \circ \mathcal{M}_1 \circ U_1 \circ \rho_0$ and $\mathcal{M}_2 \circ U_2 \circ \mathcal{M}_1 \circ U_1 \circ \rho_0$, and so on. To convey intuition, we sketch these steps one at a time, and we define $\mathcal{M}_1 \dots \mathcal{M}_d$ as we proceed. We restrict to base sets $S_{01} \dots S_{0d}$ as described above.

Hybrid 1. $\mathcal{L} \circ U_1 \circ \rho_0 \approx \mathcal{M}_1 \circ U_1 \circ \rho_0$.

Let $S_{11} \subseteq S_{01}$ be a random subset of S_{01}, subject to the constraints that (a) it includes $S_1 := H_0(\Sigma)$ and (b) $|S_{11}|/|S_{01}| = 1/|\Sigma| = \mathrm{negl}(\lambda)$. Let $S_{1j} := H_{j-1}(S_{1,j-1})$ be the propagation of S_{11} through H_1 to H_{j-1}. Here, we are trying to define a sequence of sets $(S_{11}, \dots S_{1d})$ on which we require that $\mathcal{M}_1$ outputs $\perp$ and outside of these sets,

[13] We use $a||b$ to mean concatenation of a and b.

[14] We used $\mathrm{bit}_1[\tilde{H}(\cdot)] = 1$ instead of $\tilde{H}(i||\cdot) = 1$ for notational convenience later.

we require that $\mathcal{M}_1$ behaves just like $\mathcal{L}$, i.e. if one denotes $\mathcal{M}_1 = (H_0, M_{11}, \dots M_{1d})$, then we require that M_{1i} behaves as H_i outside S_{1i} and outputs $\perp$ inside S_{1i}. To be concise, we will say that $\mathcal{M}_1$ is a shadow oracle of $\mathcal{L}$ with respect to $(S_{11} \dots S_{1d})$. Why do we want this behaviour? For $S_i := H_{i-1}(\dots H_0(\Sigma)\dots)$, $\mathcal{M}_1$ clearly contains no information about $\tilde{H}$ on Σ, since $S_j \subseteq S_{1j}$. But why couldn't we just have chosen $(S_1 \dots S_d)$ instead of $(S_{11} \dots S_{1d})$ to define $\mathcal{M}_1$? Briefly, this is because choosing to hide an exponentially larger set (note that $|S_{11}| = |\Sigma|^{d+1}$ while $|S_1| = |\Sigma|$) allows us to easily apply similar arguments in the subsequent hybrids. This will become evident shortly. Recalling our goal, we want to establish that $\mathcal{L} \circ U_1 \circ \rho_0$ and $\mathcal{M}_1 \circ U_1 \circ \rho_0$ are close in trace distance. To do this, we use the so-called one-way to hiding (O2H) lemma [8]. Informally, the lemma, as applied to our situation, says that if (a) the input state ρ_0 contains no information about the set where $\mathcal{L}$ and $\mathcal{M}_1$ behave differently, and (b) the probability of finding any element inside this set is negligible, then the trace distance between the two states of interest is negligible. The lemma clearly applies in our case because (a) initially the algorithm contains no information about $\mathcal{L}$ (it has not yet made any queries) and (b) the probability of finding any element in the set S_{1i} where $\mathcal{L}$ and $\mathcal{M}_1$ behave differently, without knowing anything about $\mathcal{L}$, is at most $|S_{1i}|/|S_{0i}| = \text{negl}(\lambda)$, for each $i \in \{1 \dots d\}$, and thus still negligible by a union bound.

Hybrid 2. $\mathcal{L} \circ U_2 \circ \rho_1 \approx \mathcal{M}_2 \circ U_2 \circ \rho_1$ where $\rho_1 = \mathcal{M}_1 \circ U_1 \circ \rho_0$. In this step, we will see the advantage of having chosen a sequence of sufficiently large sets $(S_{11}, \dots S_{1d})$ where $\mathcal{M}_1$ outputs $\perp$. Let us begin with examining the information contained in ρ_1 about $\mathcal{L}$. In the previous case, ρ_0 contained no information about $\mathcal{L}$. Since ρ_1 only learns about $\mathcal{L}$ by querying $\mathcal{M}_1$, it suffices to examine the information contained in $\mathcal{M}_1$. Since $\mathcal{M}_1$ does not hide any information about H_0, ρ_1 could have learnt $S_1 = H_0(\Sigma)$. Recall also that $S_1 \subseteq S_{11}$. This means that if one were to take $\mathcal{M}_2$ equal to $\mathcal{M}_1$, then one cannot expect $\mathcal{L} \circ U_2 \circ \rho_1$ to be close to $\mathcal{M}_2 \circ U_2 \circ \rho_1$ in general because U_2 could query the oracle at S_1 and the outputs of the two circuits would be different with probability one—$\mathcal{M}_1$ outputs $\perp$ while $\mathcal{L}$ does not. Consequently, when constructing $\mathcal{M}_2$, we do not hide anything about H_1. As for $H_2 \dots H_d$, note that, $\mathcal{M}_1$ contains no information about the behaviour of $\mathcal{L}$ inside $S_{12}, S_{13} \dots S_{1d}$. We can therefore, treat $S_{12} \dots S_{1d}$ as the new "base sets" and proceed analogously. Let $S_{22} \subseteq S_{12}$ be a random subset of S_{12}, subject to the constraint (as before) that (a) it includes $S_2 = H_1(H_0(\Sigma))$ and (b) $|S_{22}|/|S_{12}| = 1/|\Sigma| = \text{negl}(\lambda)$. Defining $\mathcal{M}_2$ to be the shadow oracle of $\mathcal{L}$ with respect to $(\emptyset, S_{22}, \dots S_{2d})$, one can again apply the O2H lemma to conclude that $\mathcal{L} \circ U_2 \circ \rho_1$ and $\mathcal{M}_2 \circ U_2 \circ \rho_1$ are close in trace distance. Note that it is crucial that $|S_{12}|$ is sufficiently large such that condition (b) above is satisfied.

Generalising the argument above, one sees that the sets S_{ij} constitute a triangular matrix (where the i-th row corresponds to sets on which $\mathcal{M}_i$ outputs $\perp$)

$$\begin{bmatrix} S_{11} & H_1(S_{11}) & H_2(H_1(S_{11}) & \dots & H_d(\dots H_1(S_{11})\dots) \\ \emptyset & S_{22} & H_2(S_{22}) & \dots & H_d(\dots H_2(S_{22})\dots) \\ \emptyset & \emptyset & S_{33} & \dots & H_d(\dots H_3(S_{33})\dots) \\ & & & \ddots & \\ \emptyset & \emptyset & \emptyset & & S_{dd} \end{bmatrix}$$

which clarifies why the argument can only be applied for d steps (as we expect). To see this, note that at the dth step, all oracles except the last have been completely revealed (last row). Crucially, the last oracle is blocked at $S_d \subseteq S_{dd}$ and therefore reveals no information about $\tilde{H}(\Sigma)$. If one proceeds with the $(d+1)$-th step, all oracles are revealed and one can no longer argue that the algorithm does not access $\tilde{H}(\Sigma)$.

Observe that so far, we have not used the fact that CodeHashing is classically hard, only that without access to the oracle, the problem cannot be solved. The classical hardness comes into play once BPP computations are allowed.

2.1.3 d-CodeHashing $\notin \text{QNC}_d^{\text{BPP}}$. We now sketch how one goes from arguing d-CodeHashing $\notin \text{QNC}_d$ to arguing d-CodeHashing $\notin \text{QNC}_d^{\text{BPP}}$. Denote circuits corresponding to $\text{QNC}_d^{\text{BPP}}$ by $\mathcal{A}_{d+1} \circ \mathcal{B}_d^{\mathcal{L}} \circ \dots \circ \mathcal{B}_1^{\mathcal{L}} \circ \rho_0$ where $\mathcal{B}_i^{\mathcal{L}} := \Pi_i \circ \mathcal{L} \circ U_i \circ \mathcal{A}_i^{\mathcal{L}}$, $\mathcal{A}_i^{\mathcal{L}}$ denotes a classical algorithm, and Π_i denotes a (possibly partial) measurement. The analogous circuit with shadow oracles is denoted by $\mathcal{A}_{d+1} \circ \mathcal{B}_d^{\mathcal{M}_d} \circ \dots \mathcal{B}_1^{\mathcal{M}_1} \circ \rho_0$ where $\mathcal{B}_i^{\mathcal{M}_i} := \Pi_i \circ \mathcal{M}_i \circ U_i \circ \mathcal{A}_i^{\mathcal{L}}$. The idea, again, is to establish, via a hybrid argument, that the two circuits are close in trace distance. In the QNC_d case, thanks to the depth of the circuit being d, we were able to argue that any QNC_d algorithm behaves equivalently if we take away its access to $\tilde{H}$. When trying to argue that a $\text{QNC}_d^{\text{BPP}}$ algorithm cannot solve the problem, we have to be more careful because the BPP part has sufficient depth to make queries to $\tilde{H}$. In our argument, this will affect how the shadow oracles $\mathcal{M}_i$ are defined.

In some more detail, we allow the classical algorithm to make "path queries"—which intuitively just means that if H_i is queried at x_i, the algorithm also learns $(x_0, x_1 \dots x_d)$ such that[15] $x_{j+1} = H_j(x_j)$ for all j. This of course can only help the algorithm.

The key idea is that we account for the "paths" that have been queried classically until depth i and define $\mathcal{M}_i$ to be consistent with those (i.e. it never outputs $\perp$ on these paths). As before, we can replace queries to $\mathcal{L}$ with queries to $\mathcal{M}_i$ that contain no information about $\tilde{H}$ except for the paths which were classically queried. Appealing to the soundness of CodeHashing, such an algorithm cannot succeed. This is because CodeHashing has the property that even an unbounded classical algorithm cannot succeed if it only makes polynomially many queries to the oracle.

2.1.4 d-CodeHashing $\notin \text{BPP}^{\text{QNC}_d}$. Observe that a poly depth quantum circuit can access $\tilde{H}$ and since a $\text{BPP}^{\text{QNC}_d}$ circuit has poly many QNC_d circuits, it is not a priori clear that $\text{BPP}^{\text{QNC}_d}$ cannot also access $\tilde{H}$. This is why the approach we used to prove that d-CodeHashing $\notin \text{QNC}_d$ cannot be applied directly. Crucially, to argue that the problem is not in $\text{BPP}^{\text{QNC}_d}$, one must use the fact that the contents of each QNC_d circuit are measured entirely, and that each QNC_d circuit takes only classical inputs. In order to handle the classical information that each QNC_d circuit receives as input, we use a technique called the "sampling argument". In essence, this says that if $\mathcal{L}$ has high entropy (which is to say that the oracles being queried are sufficiently random), then conditioned on any string s correlated with it, the resulting $\mathcal{L}|s$ behaves as a

[15] Two caveats: (1) $H_0 : \Sigma \to \Sigma^{d'}$ therefore some of the paths will not have well defined first components and (2) we only care about queries made inside the base sets where conditioned on $E, H_1 \dots H_{d-1}$ behave as permutations.

"convex combination" of high entropy distributions with a small fraction of their values completely fixed. This allows us to reduce the analysis to that of a particular set of paths being exposed, which we can handle by proceeding as in the $\mathsf{QNC}_d^{\mathsf{BPP}}$ case.

A similar argument was used by CCL to establish that a problem is not in $\mathsf{BPP}^{\mathsf{QNC}_d}$ with respect to a (structured) oracle. Their analysis used a sequence of permutation oracles and was simplified by viewing the oracles, equivalently, as distributions over paths (as opposed to a sequence of functions assigning values to individual points). The paths viewpoint was particularly helpful when considering the "sampling argument" (the version we use is derived from [27]). [11] showed that such a sampling argument can be obtained for almost any oracle which can be viewed as a distribution over paths. In our setting, since the oracles are random, paths can collide. Thus, one needs to define a suitable notion of "paths" in this setting. We provide more details in the next three paragraphs. However, since these are relatively more technical, one may wish to skip directly to Subsection 2.1.5 on a first read.

Sampling argument for Permutations. Suppose t is a permutation over N elements labelled $\{0, \dots, N-1\}$. This permutation t is ordinarily viewed as a function, $t(x)$ specifying how x is mapped. However, one could equivalently view t as a collection of pairs (or tuples later) (x, y) such that $t(x) = y$. We call such a pair a "path".

Now consider distributions over permutations. Let's begin with a uniform distribution $\mathbb{F}$ over all permutations u. One may characterise $\mathbb{F}$ as follows: for any $u \sim \mathbb{F}$, i.e. any u sampled from $\mathbb{F}$, it holds that $\Pr[u(x) = y] = \Pr[(x, y) \in \text{paths}(u)]$.

We first state a basic version of the sampling argument. To this end, we define a (p, δ) *non-uniform distribution*, $\mathbb{F}^{(p,\delta)}$, which is closely related to the uniform distribution $\mathbb{F}$. At a high level, $\mathbb{F}^{(p,\delta)}$ is "δ close to" $\mathbb{F}$ with at most p many paths fixed. What does "δ closeness" mean? Let $\Pr[S \subseteq \text{paths}(u)]$ denote the probability that a collection S of (non-colliding) paths is in u. Then, for any distribution $\mathbb{G}$ (over permutations), a distribution $\mathbb{G}^{\delta}$ is δ close to it if the following holds: when $t' \sim \mathbb{G}^{\delta}$ and $t \sim \mathbb{G}$, one has $\Pr[S \subseteq \text{paths}(t')] \leq 2^{\delta|S|} \Pr[S \subseteq \text{paths}(t)]$ for all S.

We are almost ready to state the basic sampling argument. We need the notion of a "convex combination" of random variables. We say a random variable (such as our permutation) t is a convex combination of random variables t_i, denoted by $t \equiv \sum_i \alpha_i t_i$ (where $\sum_i \alpha_i = 1$ and $\alpha_i \geq 0$), if the following holds for all t': $\Pr[t = t'] = \sum_i \alpha_i \Pr[t_i = t']$.

Informally, the basic sampling argument is a statement about a uniform permutation $u \sim \mathbb{F}$ and how the distribution $\mathbb{F}$ changes if we are given some "advice" about this permutation which is simply a function $g(u)$. Roughly speaking, given that $g(u)$ evaluates to r with probability at least 2^{-m}, the distribution $\mathbb{F}$ conditioned on r is a convex combination[16] of $\mathbb{F}^{(p,\delta)}$ distributions where the number of paths fixed is at most $p = 2m/\delta$. Here δ is a free parameter. We slightly abuse the notation and write this basic sampling argument as

$$\mathbb{F}|r \equiv \text{conv}(\mathbb{F}^{(p,\delta)}).$$

[16] In the convex combination, there is a small component, of weight at most 2^{-m}, of some arbitrary distribution.

If we view $g(u)$ as the output of the first quantum part of the circuit for $\mathsf{BPP}^{\mathsf{QNC}_d}$, and u as the oracle of interest (details are in the next section), it is suggestive that $u|g(u)$ will be the oracle for the second quantum part of the circuit. We can use the sampling argument above and re-use our analysis because $\mathbb{F}$ and $\mathbb{F}^{(p,\delta)}$ have very similar statistical properties. However, it is unclear how to use the sampling argument thereafter as the basic sampling argument seems to only apply to $\mathbb{F}$ (and not to $\mathbb{F}^{(p,\delta)}$). It turns out that one can extend the sampling argument to obtain

$$\mathbb{F}^{(p',\delta')}|r \equiv \text{conv}(\mathbb{F}^{(p+p',\delta'+\delta)}).$$

Consequently, if the procedure is successively applied $\tilde{n} \leq \text{poly}(n)$ times (starting with $\mathbb{F}$), the convex combination would be over distributions of the form $\mathbb{F}^{(\tilde{n}p,\tilde{n}\delta)}$. The parameters can be appropriately chosen to ensure that at most polynomially many paths are exposed but we omit the details in this overview.

Sampling argument for Injective Shufflers. The proofs of the previously mentioned statements do not rely on any special property of the distribution $\mathbb{F}$ nor do they depend on the fact that we were considering permutations. Any object for which we can describe a "reasonable" notion of "paths" admits such a sampling argument. Therefore, as we did for permutations, to describe the sampling argument, we change our viewpoint and consider "paths" in $\mathscr{L} = (H_0, \dots H_d)$ instead of individual values taken by the H_i's. Recall that a "path" was a tuple of the form $(x_0, x_1 \dots)$ such that $x_i = H_{i-1}(x_{i-1})$ for all i.

This viewpoint is inadequate for capturing the probabilistic behaviour of $\mathscr{L}$ due to two reasons (which are not hard to rectify). *First*, since $H_0 : \Sigma \to \Sigma^{d'}$, it is clear that at least $\left|\Sigma^{d'-1}\right|$ many points will never be contained in any "path" as described above. Therefore the behaviour of most points in H_i (for $i \in \{1 \dots d\}$) will not be captured by the "paths" viewpoint. *Second*, even though H_i maps $\Sigma^{d'} \to \Sigma^{d'}$ for $i \in \{1, \dots d-1\}$, H_i may not be injective and therefore the paths might collide, which again would mean the behaviour of many points would not be captured by the "paths" viewpoint.

To rectify the *second* issue, we can select base sets $(S_{01}, \dots S_{0d}) =: \bar{S}_0$ and condition on the event E. Since in our proofs, we only care about the behaviour of $\mathscr{L}$ on $\bar{S}_0$, it suffices to restrict our attention to $\bar{S}_0$. Recall that $\mathscr{L}|E$ behaves as a permutation on $\bar{S}_0$. Therefore no "path" inside $\bar{S}_0$ collides. To rectify the *first* issue, we consider two kinds of paths—Type 0 paths and Type 1 paths.[17] A *Type 0 path* is what we described earlier: a tuple of the form $(x_0, x_1 \dots)$ such that $x_i = H_{i-1}(x_{i-1})$ for all i. A *Type 1 path* is a tuple of the form $(\llcorner\lrcorner, x_1, x_2 \dots)$ such that $x_1 \notin H_0(\Sigma)$ (i.e. $\nexists x_0$ st $H_0(x_0) = x_1$) and $x_i = H_{i-1}(x_{i-1})$ for all $i \in \{2, 3 \dots\}$.

Observe that, restricted to $\bar{S}_0$ and conditioned[18] on E, we have the following equivalence: given $\Pr[H_i(x) = x']$ for all i, x and x', one can compute the probability associated with both types of paths and conversely, given probabilities associated with the paths, one can compute $\Pr[H_i(x) = x']$ for all i, x and x'.

As is evident, working with $\mathscr{L}$ directly is cumbersome and we therefore define a simpler object, the *injective shuffler*. Fix sets

[17] The 0 and 1 represent where the first non-$\llcorner\lrcorner$ component sits.

[18] Recall, E is the event that the oracles H_0 and $H_1 \dots H_d$ are injective on Σ and $\bar{S}_0$ resp.

$S_{0i} \subseteq \Sigma^{d'}$ of size $|\Sigma^{d+2}|$ for all $i \in \{1, \dots d\}$. Let $H'_0 : \Sigma \to S_{01}$, $H'_i : S_{0i} \to S_{0,i+1}$ for all $i \in \{1, \dots d-1\}$ be injective functions and let $H'_d : S_{0d} \to \{0,1\}^n \cup \{\perp\}$ (which may not be injective) such that H'_d outputs $\perp$ for all paths originating from Σ (and no other).[19] We define the *injective shuffler*, $\mathcal{H}$ as $(H'_0, \dots H'_d)$.

Think of $\mathcal{H}$ as a simpler way to denote the relevant object associated with $\mathcal{L}|E$. What do we mean by the relevant object—not only is it injective, it also never reveals any information[20] about the values taken by $\tilde{H}$ in Σ. As alluded to at the beginning of this subsection, since the strings s_i arise from quantum parts which only get access to $\mathcal{L}$ via shadow oracles, the sampling argument only needs to be applied to parts of $\mathcal{L}$ outside of paths in $\tilde{H}$.

To state the sampling argument for the injective shuffler, we define (p, δ) non-β-uniform distributions $\mathbb{F}_{\text{inj}}^{(p,\delta)|\beta}$ for the injective shuffler (analogous to the way we defined them for permutations). We begin with the uniform distribution—it is simply a distribution which assigns equal probabilities to all the possible injective shufflers, given the sets $(S_{0i})_i$. As for β-uniform distributions, $\mathbb{F}_{\text{inj}}^{|\beta}$, we first need to define the "paths", β. Here, β will again be a set of "non-colliding paths" but formalising this requires some care (details in the full version [10]). Then a β-uniform distribution is the same as the uniform distribution except that the paths in β are fixed. Omitting further details, one can define $\mathbb{F}^{(p,\delta)|\beta}$ to be a distribution which is "δ close to" the β-uniform distribution with at most p many paths fixed (in addition to β).

The sampling argument for injective shufflers is the following. Suppose we start with $t \sim \mathbb{F}_{\text{inj}}^{\delta'|\beta}$ (i.e. a distribution which is "δ' close to" β-uniform) and are given some advice $h(t)$ which happens to be r with probability at least 2^{-m}. Then the distribution $\mathbb{F}_{\text{inj}}^{\delta'|\beta}$ conditioned on r is, roughly speaking, a convex combination[21] of $\mathbb{F}_{\text{inj}}^{(p,\delta+\delta')|\beta}$ distributions where the number of paths fixed (in addition to β) is at most $p = 2m/\delta$ and δ again is a free parameter. Using the previous shorthand, we have

$$\mathbb{F}_{\text{inj}}^{\delta'|\beta}|r \equiv \text{conv}(\mathbb{F}_{\text{inj}}^{(p,\delta+\delta')|\beta}).$$

Stitching everything together. As asserted before we described the sampling argument, one can replace all the oracles $\mathcal{L}$ in the quantum part of the circuit for $\text{BPP}^{\text{QNC}_d}$ with appropriate shadow oracles. Let $\mathcal{M}_{11}, \dots \mathcal{M}_{1d}$ denote the shadow oracles for the first quantum part, $\mathcal{M}_{21} \dots \mathcal{M}_{2d}$ for the second quantum part and so on. Suppose the paths queried by the ith classical part were β_i, the string outputted by the ith quantum part be s_i. Suppose $\mathcal{M}_{11} \dots \mathcal{M}_{1d} \dots \mathcal{M}_{i-1,1} \dots \mathcal{M}_{i-1,d}$ have been specified. Now, conditioned on s_i, the sampling argument says $\mathcal{L}|s_i$ behaves as a convex combination of injective shufflers with certain paths exposed, when restricted to base sets. Let $\beta(s_i)$ be the random variable which specifies these paths and occurs with the weights specified in the convex combination. One can define $\mathcal{M}_{i1} \dots \mathcal{M}_{id}$ as in the QNC_d case, ensuring the paths $\beta_1 \dots \beta_{i-1}$ and $\beta(s_1) \dots \beta(s_{i-1})$ have been exposed. Note crucially that s_i is obtained by a quantum part which only had access to $\mathcal{L}$ via shadow oracles so it does not change the distribution over $\tilde{H}$ (except for polynomially many paths which were already exposed, $\beta_1 \dots \beta_{i-1}$ and $\beta(s_1) \dots \beta(s_{i-1})$). Using a hybrid argument as in the QNC_d case, and using properties of the injective shuffler which is "δ close" to being uniform, one can apply the O2H lemma and conclude that the hybrids (again, defined as in the QNC_d case) are close in trace distance. Eventually, this yields that the initial circuit is close in trace distance to the circuit which only accesses $\mathcal{L}$ via the shadows $\mathcal{M}_{11} \dots \mathcal{M}_{1d} \dots \mathcal{M}_{m1} \dots \mathcal{M}_{md}$ in the quantum part (denote the number of quantum parts by $m \leq \text{poly}(\lambda)$). The latter circuit cannot solve d-CodeHashing again, because $\tilde{H}$ is only accessed by the classical parts of this circuit. More precisely, $\tilde{H}$ is only queried at at most $|\beta_1 \cup \dots \beta_m \cup \beta(s_1) \cup \dots \beta(s_m)| \leq \text{poly}(\lambda)$ locations and therefore the whole circuit can be simulated while only making polynomially many classical queries to $\tilde{H}$. From the soundness of CodeHashing, this entails d-CodeHashing cannot be solved.

2.1.5 d-CodeHashing $\notin \text{BPP}^{\text{QNC}_d^{\text{BPP}}}$. Just as the analysis of the $\text{BPP}^{\text{QNC}_d}$ case built on the QNC_d case, one can analyze the $\text{BPP}^{\text{QNC}_d^{\text{BPP}}}$ case by building on the $\text{QNC}_d^{\text{BPP}}$ case. While the high level idea stays the same, the details are more involved. This is partly because, in the QNC_d case, one could construct the shadow oracles $\mathcal{M}_1 \dots \mathcal{M}_d$ "all at once" since we were assuming the "worst case", i.e. the quantum algorithm learns everything there is to learn from the shadow oracles. However, in the $\text{QNC}_d^{\text{BPP}}$ case, to define $\mathcal{M}_i$, one had to know the behaviour of the classical algorithms in the hybrid circuits which involved $\mathcal{M}_1 \dots \mathcal{M}_{i-1}$ (in particular one has to know the "paths" that have been exposed). We show how one can account for this, but we leave the details to the main body.

2.1.6 Proof of Quantum Depth. In this subsection, we discuss how our complexity-theoretic separations also yield protocols for certifying quantum depth, i.e. *proofs of quantum depth*, in a way that is insensitive to classical polynomial depth. First, let us be a bit more precise about what we mean by proof of quantum depth.

Definition (informal)**.** A proof of d quantum depth is a two-message protocol involving two parties, a verifier and a prover. Both parties are assumed to have access to the random oracle H. The verifier is a PPT machine. The protocol satisfies the following, where λ is the security parameter.

- Completeness: There is a prover in BQP which makes the verifier accept with probability $1 - \text{negl}(\lambda)$.
- Soundness: No prover in $\text{BPP}^{\text{QNC}_d^{\text{BPP}}}$ makes the verifier accept with probability more than $\text{negl}(\lambda)$.

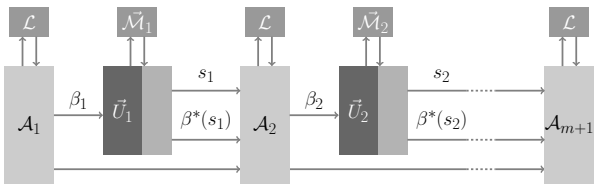


Figure 2: Here $\vec{\mathcal{M}}_i$ denotes the shadow oracles $(\mathcal{M}_{i1}, \dots \mathcal{M}_{id})$.

[19] i.e. $H'_d(x_d) = \perp$ iff $(x_0, x_1, \dots x_d, x_{d+1})$ is a Type 0 path (therefore $x_{d+1} = \perp$).
[20] Except for polynomially possibly many paths exposed by classical queries; we handle these shortly.
[21] Again, neglecting a component with weight at most 2^{-m}.

Let d be at most a fixed polynomial. Since d-CodeHashing is in NP, it immediately yields a proof of d quantum depth.

We conclude this discussion by illustrating the subtlety of considering proofs of quantum depth with more than two messages. Consider the following protocol.

Example 12. The verifier, Alice, prepares BB84 states $|b_i\rangle_{\theta_i} := H^{\theta_i}|b_i\rangle$ (b_i, θ_i are both chosen uniformly at random) for $i \in \{1, \dots n\}$ where H is the Hadamard operation (not to be confused with the random oracle). She sends them all to the prover, Bob.

Alice and Bob then engage in an n round protocol. In the i-th round, Alice sends θ_i and Bob sends b'_i. Alice accepts if $b_1 = b'_1, \dots b_n = b'_n$.

In this example,[22] it is not hard to see that Bob has to have n layers of unitaries. Could this simple construction already constitute a proof of quantum depth? Consider the following observations.

- *Spoofed by n single quantum depth devices.* It is easy to see that Bob can pass this test using n-many single-qubit quantum devices, each of which need only apply one quantum gate and make one computational basis measurement. The protocol works by simply delaying the application of the quantum gate and subsequent measurement. It is therefore difficult to call this a proof of quantum depth in any meaningful way.
- *Interaction seems superfluous.* The only use of the interaction is to introduce a delay. The same effect could be achieved with a single round protocol where Alice delays sending her message. Therefore, this procedure, at best, certifies "idle coherence" time.

The example shows how defining quantum depth in interactive settings can be quite subtle. We refer the reader back to the discussion in Section 1.3 for our proposal of what this definition should be.

2.1.7 Tighter Upper Bounds. Ideally, one would like to show the more fine-grained separation $\mathsf{BPP}^{\mathsf{QNC}_d^{\mathsf{BPP}}} \subsetneq \mathsf{BPP}^{\mathsf{QNC}_{d+1}^{\mathsf{BPP}}}$. Since the best known algorithm for solving YZ's CodeHashing uses polynomial depth, d-CodeHashing inherits this limitation. By using a different problem, we overcome this limitation and show the following.

Theorem 13. *Relative to a random oracle,* $\mathsf{QNC}_{2d+\mathcal{O}(1)} \not\subseteq \mathsf{BPP}^{\mathsf{QNC}_d^{\mathsf{BPP}}}$ *which implies* $\mathsf{BPP}^{\mathsf{QNC}_d^{\mathsf{BPP}}} \subsetneq \mathsf{BPP}^{\mathsf{QNC}_{2d+\mathcal{O}(1)}^{\mathsf{BPP}}}$.

We obtain the above by instantiating our lifting procedure, d-Rec$[\cdot]$, with a variant of the proof of quantumness from [17], which we refer to as CollisionHashing. It is straightforward to show that CollisionHashing also satisfies classical query soundness by using the main argument in [17] and the query lower bound for finding collisions proved in [4].

Let g be a $2 \to 1$ function for which it is hard to find a collision. Then, the (slightly simplified) problem is to produce a pair (y, r) such that $r \cdot (x_0 \oplus x_1) \oplus H(x_0) \oplus H(x_1) = 0$ where $\{x_0, x_1\} \in g^{-1}(y)$. This problem can be solved in $\mathsf{QNC}_{\mathcal{O}(1)}$ (assuming that calls to g take only depth 1) by preparing the superposition $\sum_x |g(x)\rangle |x\rangle$, measuring the second register in the standard basis, and the first in the Hadamard basis (detailed later).

We said simplified because in CollisionHashing, g is in fact a uniformly random function g (treated as an oracle) with a domain twice as large as the co-domain. Note that this is not a $2 \to 1$ function in general. However, with overwhelming probability, a constant fraction of the elements in the co-domain has exactly two pre-images. Then, we require a pair (y, r) such that either y has exactly two pre-images and (y, r) satisfies the "equation", or y does not have exactly two pre-images. The limitation of CollisionHashing is that solutions to the problem are not verifiable, so the problem cannot be used to obtain a fine-grained proof of quantum depth.

2.2 Separations Of Hybrid Quantum Depth Classes

2.2.1 Establishing $\mathsf{BPP}^{\mathsf{QNC}_{\mathcal{O}(1)}} \not\subseteq \mathsf{QNC}_d^{\mathsf{BPP}}$. We describe our second lifting procedure, called d-Ser$[\cdot]$. This takes any problem $\mathcal{P} \notin \mathsf{BPP}$ (relative to a random oracle) that satisfies offline soundness, and produces a new problem d-Ser$[\mathcal{P}] \notin \mathsf{QNC}_d^{\mathsf{BPP}}$ (see Lemma 7).

Denote by R_H the set of solutions to $\mathcal{P}$ (defined with respect to H). Then, the key idea is simple. The problem d-Ser$[\mathcal{P}]$ is to return a tuple $(c_0, c_1, \dots, c_d)$ such that: c_0 is a solution to $\mathcal{P}$, i.e. $c_0 \in R_{H(\cdot)}$; c_1 is a solution to $\mathcal{P}$ but with respect to $H(c_0||\cdot)$, i.e. $c_1 \in R_{H(c_0||\cdot)}$, and similarly until c_d, which should be such that $c_d \in R_{H(c_0 \dots c_{d-1}||\cdot)}$.

To be a bit more concrete, take $\mathcal{P}$ to be CollisionHashing. We know CollisionHashing $\in \mathsf{QNC}_{\mathcal{O}(1)}$. Clearly, d-Ser[CollisionHashing] $\in \mathsf{BPP}^{\mathsf{QNC}_{\mathcal{O}(1)}}$. This is because $\mathsf{BPP}^{\mathsf{QNC}_{\mathcal{O}(1)}}$ allows one to run polynomially many $\mathsf{QNC}_{\mathcal{O}(1)}$ circuits. Consequently, one can use the first circuit to obtain the classical output c_0, use the second circuit to find c_1 and so on. On the other hand, intuitively, we expect that d-Ser[CollisionHashing] $\notin \mathsf{QNC}_d^{\mathsf{BPP}}$. This is because to solve the $(i+1)$-th sub-problem, one seems to require the solution to all of the previous i sub-problems. Since there are $d+1$ sub-problems in total, $\mathsf{QNC}_d^{\mathsf{BPP}}$ does not seem to suffice (here of course we are implicitly using the fact that $\mathcal{P} \notin \mathsf{BPP}$). Formally, the argument proceeds in a similar way as for the lifting map d-Rec in Subsection 2.1.3, except for one subtlety which is handled by requiring that the problem $\mathcal{P}$ satisfies the extra property of offline soundness. We refer the reader to the main text for more details. We remark that offline soundness follows from classical query soundness and therefore both CollisionHashing and CodeHashing satisfy it.

The immediate consequence of the existence of the lifting map d-Ser$[\cdot]$ is that $\mathsf{BPP}^{\mathsf{QNC}_{\mathcal{O}(1)}} \not\subseteq \mathsf{QNC}_d^{\mathsf{BPP}}$ (first part of Theorem 4). However, we can also leverage d-Ser$[\cdot]$, together with the separation from the next subsection, to show that $\mathsf{BPP}^{\mathsf{QNC}_{\mathcal{O}(1)}^{\mathsf{BPP}}} \not\subseteq \mathsf{BPP}^{\mathsf{QNC}_d} \cup \mathsf{QNC}_d^{\mathsf{BPP}}$ (Theorem 5). This is done as follows.

In Subsection 2.2.2, we introduce the problem d-hCollisionHashing (which also satisfies offline soundness), and argue that it is in $\mathsf{QNC}_{\mathcal{O}(1)}^{\mathsf{BPP}}$, but not in $\mathsf{BPP}^{\mathsf{QNC}_d}$. Now, applying the lifting map to it gives d-Ser[d-hCollisionHashing] $\notin \mathsf{BPP}^{\mathsf{QNC}_d} \cup \mathsf{QNC}_d^{\mathsf{BPP}}$. To obtain the containment, notice that d-Ser yields a problem that can be solved by solving $d+1$

[22] While we used quantum communication in the protocol, one could (using known results) delegate the production of these states to the prover (under computational assumptions) and run a similar protocol using classical communication.

many instances of the original problem. Thus, it follows that d-Ser[d-hCollisionHashing] $\in \mathsf{BPP}^{\mathsf{QNC}^{\mathsf{BPP}}_{\mathcal{O}(1)}}$.

2.2.2 Establishing $\mathsf{QNC}^{\mathsf{BPP}}_{\mathcal{O}(1)} \not\subseteq \mathsf{BPP}^{\mathsf{QNC}_d}$. This is the more surprising of the two hybrid separations, and its proof is more involved. In this section, we fix $d \leq poly(\lambda)$. The problem that yields this separation is the following variation on CollisionHashing: given access to a 2-to-1 function g[23], and to $H_0, \ldots H_d$ (which specify h as $h = H_d \circ \cdots \circ H_0$), find a pair (y, r) such that

$$r \cdot (x_0 \oplus x_1) \oplus H(h(y)||x_0) \oplus H(h(y)||x_1) = 0\,,$$

where $\{x_0, x_1\} = g^{-1}(y)$. We refer to the new problem as d-hCollisionHashing.

Without relying on h (that is, requiring that the equation to be satisfied is just $r \cdot (x_0 \oplus x_1) \oplus H(x_0) \oplus H(x_1) = 0$), this problem is the same as CollisionHashing. This can be solved in $\mathsf{QNC}_{\mathcal{O}(1)}$ as follows:

(i) Evaluate g on a uniform superposition of inputs, obtaining $\sum_x |x\rangle |g(x)\rangle$,
(ii) Measure the image register obtaining some outcome y and a state $(|x_0\rangle + |x_1\rangle)|y\rangle$,
(iii) Query a phase oracle for H to obtain $((-1)^{H(x_0)}|x_0\rangle + (-1)^{H(x_1)}|x_1\rangle)|y\rangle$,
(iv) Make a Hadamard basis measurement of the first register, obtaining outcome r.

At a high level, in order to solve the new problem, which includes the evaluation of h as an input to H, one needs the ability to perform a (classical) depth d computation to evaluate $h(y)$ (since this requires the sequential evaluations of $H_0, \ldots, H_d$). Note that a $\mathsf{QNC}^{\mathsf{BPP}}$ algorithm can solve this problem: the only modification to the algorithm described above is that, at step (iii), the algorithm first computes $h(y)$ (using polynomial classical computation), and then queries the oracle H on a superposition of $(h(y), x_0)$ and $(h(y), x_1)$. One can easily verify that this leads to a valid (y, r) for the problem.

Next, we sketch how one can argue that the problem cannot be solved in $\mathsf{BPP}^{\mathsf{QNC}}$. The key technical ingredient is a "structure theorem" that characterizes the structure of efficient quantum strategies that are successful at CollisionHashing. Our structure theorem applies equally to the proof of quantumness protocol from [17] (recall that the latter is just a version of collision hashing where g is replaced by a 2-to-1 trapdoor claw-free function).

Theorem 14 (informal)**.** *Let P be any* BQP *prover that succeeds with $1 - \mathrm{negl}(n)$ probability at the proof of quantumness protocol from [17], by making q queries to the oracle H. Then, with $1 - \mathrm{negl}(n)$ probability over pairs (H, y), the following holds. Let $p_{y|H}$ be the probability that P^H outputs y, and let x_0,x_1 be the pre-images of y. Then, for all $b \in \{0, 1\}$, there exists $i \in [q]$ such that the state of the query register of P^H right before the i-th query has weight $\frac{1}{2}p_{y|H} \cdot (1 - \mathrm{negl}(n))$ on x_b.*

[23] Since we want our problem to be relative to a uniformly random oracle, in the formal description of the problem in the main text, we will not assume that g is exactly 2-to-1. Rather we will take g to be a uniformly random function with domain twice as large as the co-domain, and simply restrict our attention to y's in the co-domain that have exactly two pre-images (this is a constant fraction of the elements of the co-domain with overwhelming probability).

See the full version [10] for a formal statement of this result. This is a crucial strengthening of a Theorem from [26], and employs the compressed oracle technique [44]. A slight adaptation of this to our problem asserts that a successful strategy must be querying the random oracle H at a (close to) uniform superposition of $(h(y), x_0)$ and $(h(y), x_1)$.

Now let A be a $\mathsf{BPP}^{\mathsf{QNC}}$ algorithm that succeeds at d-hCollisionHashing with high probability and let q be the total number of queries to h made by the algorithm.

Then, one can show that, since the QNC part of the algorithm does not have sufficient depth to evaluate h (which is a sequential evaluation of $H_0, \ldots, H_d$), we can assume, without loss of generality, the QNC part of A has no access to h. In other words, all of the queries to h are classical.

Now, Theorem 14 says essentially that, for any y, the only way to succeed with high probability (conditioned on that y being the output) is to query (with as much weight as the probability of outputting y) a uniform superposition of $(h(y), x_0)$ and $(h(y), x_1)$. However, observe that, for any y, the only way for A to query H (with a high weight) at a uniform superposition of $(h(y), x_0)$ and $(h(y), x_1)$ is to correctly guess the value of $h(y)$. Since this value is uniformly random for any algorithm that has not queried h at y, it follows that querying H at the uniform superposition of $(h(y), x_0)$ and $(h(y), x_1)$ must necessarily happen *after* the algorithm has already queried h on y.

This implies that there must exist an $i^* \in [q]$ such that, with high probability, A outputs y, r such that y is contained in the list of classical queries made to h up to the i^*-th query. Denote such a list by L_{i^*}. Moreover, with high probability over L_{i^*}, the continuation of A (from that point on) queries H at a uniform superposition of $(h(y), x_0)$ and $(h(y), x_1)$ for some $y \in L_{i^*}$. We show that such an algorithm A can be leveraged to extract a collision for g.

The key observation is that, since A is a $\mathsf{BPP}^{\mathsf{QNC}}$ algorithm, and all of the queries to h happen in the BPP portion of A, the "state" of algorithm A right after the i^*-th query to h is entirely *classical*. Thus, one can take a "snapshot" of the state of A at that point (i.e. copy it), and simply run *two independent executions* of A from that point on (with independent classical randomness). By what we argued earlier, with high probability, there exists $y \in L_{i^*}$, such that the execution of A from that point on, queries H at a uniform superposition of $(h(y), x_0)$ and $(h(y), x_1)$. Since the two executions are identical and independent, it follows that measuring the query registers of H in both executions will yield distinct pre-images of y with significant probability.

Finding collisions of g is of course hard (for any query-bounded quantum algorithm) [4]. Hence, this yields a contradiction. For details, we refer the reader to the full version [10].

Acknowledgments

We are thankful to Joseph Slote, Ulysse Chabaud and Thomas Vidick for various discussions. ASA acknowledges support from IQIM, an NSF Physics Frontier Center (GBMF-1250002) and MURI grant FA9550-18-1-0161. While at ETH, AG was supported by Dr. Max Rössler, the Walter Haefner Foundation and the ETH Zürich Foundation. AC is a Quantum Postdoctoral Fellow at the Simons Institute for the Theory of Computing supported by NSF QLCI Grant No.

2016245, and by DARPA under agreement No. HR00112020023. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the United States Government or DARPA. US acknowledges the support by Polish National Science Center (NCN) (Grant No. 2019/35/B/ST2/01896). HW is supported by NSF award CNS-2154705 and the MC2 postdoctoral fellowship.

REFERENCES

[1] Scott Aaronson. 2010. BQP and the Polynomial Hierarchy. In *Proceedings of the Forty-Second ACM Symposium on Theory of Computing* (Cambridge, Massachusetts, USA) *(STOC '10)*. Association for Computing Machinery, 141–150. https://doi.org/10.1145/1806689.1806711

[2] Scott Aaronson and Andris Ambainis. 2009. The need for structure in quantum speedups. *arXiv preprint arXiv:0911.0996* (2009).

[3] Scott Aaronson and Andris Ambainis. 2015. Forrelation: A Problem That Optimally Separates Quantum from Classical Computing. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing* (Portland, Oregon, USA) *(STOC '15)*. Association for Computing Machinery, 307–316. https://doi.org/10.1145/2746539.2746547

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

[5] Dorit Aharonov, Alexei Kitaev, and Noam Nisan. 1998. Quantum circuits with mixed states. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*. 20–30.

[6] Dorit Aharonov and Yonathan Touati. 2018. Quantum circuit depth lower bounds for homological codes. *arXiv preprint arXiv:1810.03912* (2018).

[7] Miklós Ajtai. 1996. Generating hard instances of lattice problems. In *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*. 99–108.

[8] Andris Ambainis, Mike Hamburg, and Dominique Unruh. 2019. Quantum Security Proofs Using Semi-classical Oracles. In *Advances in Cryptology – CRYPTO 2019*, Alexandra Boldyreva and Daniele Micciancio (Eds.). Springer International Publishing, 269–295. https://doi.org/10.1007/978-3-030-26951-7_10

[9] Benny Applebaum, Yuval Ishai, and Eyal Kushilevitz. [n. d.]. Cryptography in $NC0$. 36, 4 ([n. d.]), 845–888. https://doi.org/10.1137/S0097539705446950

[10] Atul Singh Arora, Andrea Coladangelo, Matthew Coudron, Alexandru Gheorghiu, Uttam Singh, and Hendrik Waldner. 2022. Quantum Depth in the Random Oracle Model. arXiv:2210.06454 [quant-ph]

[11] Atul Singh Arora, Alexandru Gheorghiu, and Uttam Singh. [n. d.]. Oracle Separations of Hybrid Quantum-Classical Circuits. ([n. d.]). https://doi.org/10.48550/arXiv.2201.01904 arXiv:2201.01904 [quant-ph]

[12] Boaz Barak, Oded Goldreich, Russell Impagliazzo, Steven Rudich, Amit Sahai, Salil Vadhan, and Ke Yang. 2012. On the (im) possibility of obfuscating programs. *Journal of the ACM (JACM)* 59, 2 (2012), 1–48.

[13] Nir Bitansky, Ran Canetti, Henry Cohn, Shafi Goldwasser, Yael Tauman Kalai, Omer Paneth, and Alon Rosen. 2014. The impossibility of obfuscation with auxiliary input or a universal simulator. In *Annual Cryptology Conference*. Springer, 71–89.

[14] Nir Bitansky, Shafi Goldwasser, Abhishek Jain, Omer Paneth, Vinod Vaikuntanathan, and Brent Waters. 2016. Time-lock puzzles from randomized encodings. In *Proceedings of the 2016 ACM Conference on Innovations in Theoretical Computer Science*. 345–356.

[15] Jeremiah Blocki, Seunghoon Lee, and Samson Zhou. [n. d.]. On the Security of Proofs of Sequential Work in a Post-Quantum World. ([n. d.]). arXiv:2006.10972 [cs] http://arxiv.org/abs/2006.10972

[16] Dan Boneh, Joseph Bonneau, Benedikt Bünz, and Ben Fisch. 2018. Verifiable Delay Functions. In *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part I (Lecture Notes in Computer Science, Vol. 10991)*, Hovav Shacham and Alexandra Boldyreva (Eds.). Springer, 757–788. https://doi.org/10.1007/978-3-319-96884-1_25

[17] Zvika Brakerski, Venkata Koppula, Umesh V. Vazirani, and Thomas Vidick. 2020. Simpler Proofs of Quantumness. In *15th Conference on the Theory of Quantum Computation, Communication and Cryptography, TQC 2020, June 9-12, 2020, Riga, Latvia (LIPIcs, Vol. 158)*, Steven T. Flammia (Ed.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 8:1–8:14. https://doi.org/10.4230/LIPIcs.TQC.2020.8

[18] Hans J Briegel, David E Browne, Wolfgang Dür, Robert Raussendorf, and Maarten Van den Nest. 2009. Measurement-based quantum computation. *Nature Physics* 5, 1 (2009), 19–26.

[19] Ran Canetti, Yilei Chen, and Leonid Reyzin. 2016. On the correlation intractability of obfuscated pseudorandom functions. In *Theory of cryptography conference*. Springer, 389–415.

[20] Ran Canetti, Oded Goldreich, and Shai Halevi. 2004. The random oracle methodology, revisited. *Journal of the ACM (JACM)* 51, 4 (2004), 557–594.

[21] Nai-Hui Chia, Kai-Min Chung, and Ching-Yi Lai. [n. d.]. On the Need for Large Quantum Depth. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing* (New York, NY, USA, 2020-06-08) *(STOC 2020)*. Association for Computing Machinery, 902–915. https://doi.org/10.1145/3357713.3384291

[22] Nai-Hui Chia and Shih-Han Hung. 2022. Classical verification of quantum depth. https://doi.org/10.48550/ARXIV.2205.04656

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

[24] Kai-Min Chung, Serge Fehr, Yu-Hsuan Huang, and Tai-Ning Liao. 2021. On the compressed-oracle technique, and post-quantum security of proofs of sequential work. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 598–629.

[25] R. Cleve and J. Watrous. [n. d.]. Fast Parallel Circuits for the Quantum Fourier Transform. In *Proceedings 41st Annual Symposium on Foundations of Computer Science* (2000), Vol. 1. 526. https://doi.org/10.1109/SFCS.2000.892140

[26] Andrea Coladangelo, Shafi Goldwasser, and Umesh Vazirani. 2022. Deniable encryption in a Quantum world. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*. 1378–1391.

[27] Sandro Coretti, Yevgeniy Dodis, Siyao Guo, and John P. Steinberger. 2018. Random Oracles and Non-uniformity. In *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part I (Lecture Notes in Computer Science, Vol. 10820)*, Jesper Buus Nielsen and Vincent Rijmen (Eds.). Springer, 227–258. https://doi.org/10.1007/978-3-319-78381-9_9

[28] Matthew Coudron and Sanketh Menda. [n. d.]. 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* (2020) *(STOC 2020)*. 889–901. https://doi.org/10.1145/3357713.3384269

[29] Edward Farhi, Jeffrey Goldstone, and Sam Gutmann. 2014. A quantum approximate optimization algorithm. *arXiv preprint arXiv:1411.4028* (2014).

[30] Merrick Furst, James B Saxe, and Michael Sipser. 1984. Parity, circuits, and the polynomial-time hierarchy. *Mathematical systems theory* 17, 1 (1984), 13–27.

[31] Atsuya Hasegawa and François Le Gall. 2022. An optimal oracle separation of classical and quantum hybrid schemes. https://doi.org/10.48550/ARXIV.2205.04633

[32] John Hastad. 1986. Almost optimal lower bounds for small depth circuits. In *Proceedings of the eighteenth annual ACM symposium on Theory of computing*. 6–20.

[33] Richard Jozsa. 2005. An introduction to measurement based quantum computation. *Quantum Information Processing* 199 (09 2005).

[34] Yael Tauman Kalai, Guy N Rothblum, and Ron D Rothblum. 2017. From obfuscation to the security of Fiat-Shamir for proofs. In *Annual international cryptology conference*. Springer, 224–251.

[35] Peter Bro Miltersen. 1992. Circuit Depth Relative to a Random Oracle. *Inf. Process. Lett.* 42, 6 (1992), 295–298. https://doi.org/10.1016/0020-0190(92)90225-K

[36] Chris Peikert and Sina Shiehian. 2019. Noninteractive zero knowledge for NP from (plain) learning with errors. In *Annual International Cryptology Conference*. Springer, 89–114.

[37] Alberto Peruzzo, Jarrod McClean, Peter Shadbolt, Man-Hong Yung, Xiao-Qi Zhou, Peter J. Love, Alán Aspuru-Guzik, and Jeremy L. O'Brien. 2014. A variational eigenvalue solver on a photonic quantum processor. *Nature Communications* 5, 1 (7 2014). https://doi.org/10.1038/ncomms5213

[38] Robert Raussendorf and Hans J Briegel. 2001. A one-way quantum computer. *Physical review letters* 86, 22 (2001), 5188.

[39] Alexander A Razborov and Steven Rudich. 1994. Natural proofs. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*. 204–213.

[40] Ronald L Rivest, Adi Shamir, and David A Wagner. 1996. Time-lock puzzles and timed-release crypto. (1996).

[41] Gregory Rosenthal. 2020. Bounds on the QAC^0 Complexity of Approximating Parity. *arXiv preprint arXiv:2008.07470* (2020).

[42] Daniel R. Simon. 1997. On the Power of Quantum Computation. *SIAM J. Comput.* 26, 5 (1997), 1474–1483. https://doi.org/10.1137/S0097539796298637 arXiv:https://doi.org/10.1137/S0097539796298637

[43] Takashi Yamakawa and Mark Zhandry. [n. d.]. Verifiable Quantum Advantage without Structure. ([n. d.]). arXiv:2204.02063 [quant-ph] http://arxiv.org/abs/2204.02063

[44] Mark Zhandry. 2019. How to record quantum queries, and applications to quantum indifferentiability. In *Annual International Cryptology Conference*. Springer, 239–268.

Received 2022-11-07; accepted 2023-02-06