Quantum Advantage and Lower Bounds in Parallel Query Complexity

Joseph Carolan
University of Maryland, College Park, MD, USA

Amin Shiraz Gilani
University of Maryland, College Park, MD, USA

Mahathi Vempati
University of Maryland, College Park, MD, USA

Abstract

It is well known that quantum, randomized and deterministic (sequential) query complexities are polynomially related for total boolean functions. We find that significantly larger separations between the parallel generalizations of these measures are possible. In particular,

1. We employ the cheatsheet framework to obtain an unbounded parallel quantum query advantage over its randomized analogue for a total function, falsifying a conjecture of [Jeffery et al. 2017].
2. We strengthen 1 by constructing a total function which exhibits an unbounded parallel quantum query advantage despite having no sequential advantage, suggesting that genuine quantum advantage could occur entirely due to parallelism.
3. We construct a total function that exhibits a polynomial separation between 2-round quantum and randomized query complexities, contrasting a result of [Montanaro. 2010] that there is at most a constant separation for 1-round (nonadaptive) algorithms.
4. We develop a new technique for deriving parallel quantum lower bounds from sequential upper bounds. We employ this technique to give lower bounds for Boolean symmetric functions and read-once formulas, ruling out large parallel query advantages for them.

We also provide separations between randomized and deterministic parallel query complexities analogous to items 1-3.

2012 ACM Subject Classification Theory of computation → Quantum complexity theory

Keywords and phrases Computational complexity theory, quantum, lower bounds, parallel

Digital Object Identifier 10.4230/LIPIcs.ITCS.2025.31

Related Version *Full Version*: `https://arxiv.org/abs/2410.02665`

Funding *Joseph Carolan*: US Department of Energy grant no. DESC0020264.
Amin Shiraz Gilani: U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Accelerated Research in Quantum Computing and Quantum Testbed Pathfinder programs (award numbers DE-SC0020312 and DE-SC001904).
Mahathi Vempati: US Department of Energy grant no. DESC0020264 and the US Department of Energy Nuclear Energy University Programs award number DE-NE0009417.

Acknowledgements The authors thank Laxman Dhulipala for suggesting to investigate the relationship between quantum algorithms and parallelism, Luke Schaeffer and Chaitanya Karamchedu for helpful discussions, and Andrew Childs for valuable feedback on an earlier draft. We thank an anonymous reviewer for suggesting the idea for the function in Section 1.1.2, greatly simplifying our original construction. ASG additionally thanks Stacey Jeffery and Ronald de Wolf for helpful discussions.

1 Introduction

Quantum query complexity is a widely studied model for understanding the capabilities and limitations of quantum computers. Many of the important quantum algorithms and quantum-classical separation results are expressed in this model. For instance, the quantum period finding algorithm, a major ingredient in Shor's factoring algorithm [32], was first developed in the query model. There are many other examples of query problems for which quantum algorithms provably achieve exponential query advantage over their classical counterparts [33, 18].

However, these problems have partial domains, meaning they are not defined on most inputs. For functions whose domain includes all inputs (total functions), it is known that the quantum, randomized and deterministic query complexities are all polynomially related [10, 4]. Grover's seminal algorithm [24] for unstructured search achieves a quadratic speed-up over classical algorithms, and Ambainis et al. [6] and Aaronson et al. [3] construct functions that separate quantum and deterministic query complexities by a 4th power, and quantum and randomized query complexities by a 3rd power respectively.[1] These separations cannot be significantly improved in the sequential query model [10, 4].

A natural question is whether these algorithmic limitations are robust against generalizations of sequential query complexity. In this work, we consider a parallel model where an algorithm is allowed to make many queries at each step, as introduced in [27]. In particular, the p-parallel quantum query complexity, denoted by $\mathsf{Q}^{p\|}$, of a function is the minimum number of steps needed to compute it with bounded error, if at each step the algorithm is allowed to make p non-adaptive quantum queries.[2] We will also refer to this quantity by *p-query depth*, *p-query rounds* and *p-query layers* interchangeably. Notice that, for any function f and any parallelism p, $\mathsf{Q}(f)/p \leq \mathsf{Q}^{p\|}(f) \leq \mathsf{Q}(f)$ since any q-round p-parallel algorithm can be simulated by a pq-query sequential algorithm, and any q-query sequential algorithm can be simulated by a q-round p-parallel algorithm.[3]

This model is motivated by the fact that fault tolerant quantum computers must be inherently parallel: a large scale quantum computer which cannot do many gates at once cannot correct local errors faster than they occur. Therefore, quantum computers will necessarily be able to perform operations in parallel, meaning one should design algorithms to take maximum advantage of this capability. Further, near term quantum computations are limited to low depth due to short decoherence times. These factors make the parallel query model a more natural abstraction of real devices than a purely sequential model.

Prior work addressing parallel query complexity has indicated that quantum advantage tends to disappear as parallelism increases. Zalka [37] showed the parallel complexity for searching an unstructured list of size N is $\Theta\left(\sqrt{N/p}\right)$. This was generalized by Grover and Radhakrishnan [25], who proved that $\tilde{\Theta}\left(\sqrt{Nt/(p\cdot\min\{t,p\})}\right)$ p-parallel queries are necessary and sufficient to find t marked elements in a list of size N. Along with providing a systematic study of parallel quantum query complexity, Jeffery, Magniez and de Wolf [27] showed that the p-parallel quantum query complexity of k-SUM is $\tilde{\Theta}\left((N/p)^{k/k+1}\right)$.

---

[1] [3] originally showed a 2.5th power separation, which was boosted to a 3rd power via tight lower bounds on k-fold Forrelation shown in [8, 31].

[2] We define the deterministic parallel query complexity $\mathsf{D}^{p\|}$ and randomized bounded error parallel query complexity $\mathsf{R}^{p\|}$ similarly.

[3] Similar bounds hold for $\mathsf{R}^{p\|}$ and $\mathsf{D}^{p\|}$.

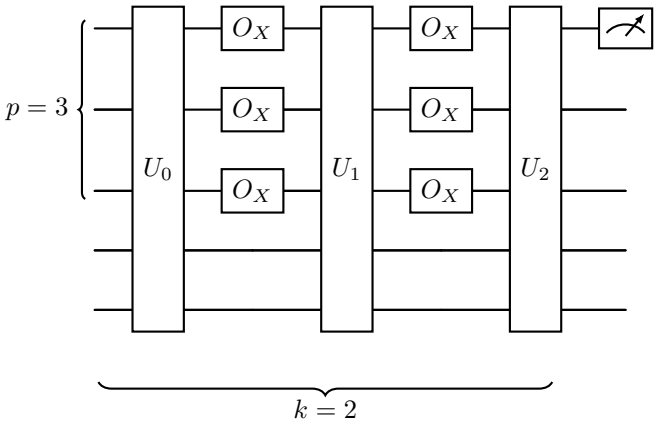

Figure 1 A parallel quantum algorithm with $k = 2$ adaptive steps and $p = 3$ queries per step; the last two wires are for workspace. In general, p denotes the number of non-adaptive queries per step, and k denotes the number of steps. The measure $Q^{p\|}$ denotes the minimal k needed to compute a function for a fixed p, while the measure $Q^{k\perp}$ denotes the minimal p needed to compute a function for a fixed k.

Furthermore, [27] introduce the following conjecture:

▶ **Conjecture 1** ([27]). *For any total function $f : \{0,1\}^N \to \{0,1\}$ and any p, there is at most a polynomial quantum advantage for p-parallel query algorithms. That is,*

$$\mathsf{D}^{p\|}(f) = \mathit{poly}(\mathsf{Q}^{p\|}(f)).$$

In fact, [27] proved this conjecture for all p polynomially smaller than the block sensitivity (bs), generalizing a result of Beals et al. [10].

The notion of p-parallel query complexity captures the minimum "depth" needed to compute some function for a given "width" (see Figure 1). An alternative complexity measure relevant to parallelism is the minimum "width" needed to compute some function for a fixed "depth", which we call the k-adaptive query complexity, and denote by $\mathsf{Q}^{k\perp}(f)$ (respectively $\mathsf{R}^{k\perp}(f)$, $\mathsf{D}^{k\perp}(f)$). Precisely, we define k-adaptive query complexity as the minimum p such that there is a p-parallel quantum (respectively randomized, deterministic) algorithm which makes k many p-parallel queries and computes f. This complexity measure is not widely studied for total functions. However, a result of Montanaro [28] fully characterizes the relevant 1-adaptive (or non-adaptive) query complexities, up to a factor of 2: for any total $f : \{0,1\}^N \to \{0,1\}$, $\mathsf{Q}^{1\perp}(f) \geq \mathsf{D}^{1\perp}(f))/2$. In the context of partial functions, a result by Girish et al. [22] characterized the maximal separation between $\mathsf{Q}^{k\perp}$ and $\mathsf{Q}^{k+1\perp}$.

In this paper, we present the high level overview of our work. See the full version on arxiv [14].

1.1 Main results

1.1.1 Unbounded parallel separations for total functions

Our first result involves adapting the cheatsheet framework of Aaronson et al. [3] to the parallel query setting, and using it to exhibit total function separations between parallel query complexity measures. In particular, we show that the canonical cheatsheet function of [3] exhibits unbounded separation between quantum and randomized parallel query complexities,

falsifying Conjecture 1. Further, we observe that this function demonstrates that 3-adaptive quantum algorithms can be more efficient in terms of total queries than even fully sequential randomized algorithms. More formally, we prove the following theorem.

▶ **Theorem 2.** *There exists a (total) Boolean function $h : \{0,1\}^N \to \{0,1\}$ such that for some $p = \tilde{O}(\mathsf{bs}(h))$ and some constants $\epsilon, \delta > 0$, we have*

(i) $\mathsf{R}^{p\|}(h) = \tilde{\Omega}(N^{\epsilon})$, **(ii)** $\mathsf{Q}^{p\|}(h) \leq 3$, **(iii)** $\mathsf{Q}^{3\perp}(h) = \tilde{O}\left(\mathsf{R}(h)^{1-\delta}\right)$.

It is worth noting that the large separation in parallel query complexity occurs when the parallelism almost exactly equals the block sensitivity $\mathsf{bs}(f)$, contrasting a result of Jeffery et al ([27], Theorem 12) which eliminates this possibility for $p = O(\mathsf{bs}(f)^{1-\epsilon})$ for any constant $\epsilon > 0$. Furthermore, the separation between $\mathsf{Q}^{3\perp}(f)$ and $\mathsf{R}(f)$ contrasts Montanaro's result ([28], Theorem 1) that for total functions, 1-adaptive quantum algorithms cannot be more than twice as efficient as the trivial 1-adaptive deterministic algorithm.

We also show a total function that achieves a similar separation between randomized and deterministic parallel query complexities. We do this by modifying the canonical cheatsheet function to instead embed a partial function with a large randomized-deterministic separation.

It is worth noting that this result is also sufficient to falsify Conjecture 1.

These results provide a recipe for achieving exponential parallel query advantages (from quantumness or randomness) for total functions by amplifying a polynomial sequential advantage for such functions. However, it is unclear whether an advantage can originate entirely from a quantum or randomized algorithm's ability to utilize parallelism more effectively than a randomized or deterministic algorithms. In other words, is there a function which has no sequential quantum query advantage, yet a large parallel quantum query advantage? We answer this question in the affirmative, even for total functions.

1.1.2 Unbounded genuine parallel separations

Our starting point is a partial function with the desired property. That is, we construct a partial function with (almost) equal randomized and quantum sequential query complexities, that does not admit any parallel advantage for randomized algorithms, yet admits near-maximal parallel advantage for quantum algorithms. In particular, we show the following theorem.

▶ **Theorem 3.** *There exists a (partial) Boolean function $f : \mathcal{D} \to \{0,1\}$ with $\mathcal{D} \subset \{0,1\}^N$ such that for some $p = O(\mathsf{Q}(f))$ and some constant $\epsilon > 0$, we have*

(i) $\mathsf{R}(f) = \tilde{\Theta}(N^{\epsilon})$, **(ii)** $\mathsf{Q}(f) = \tilde{\Omega}(\mathsf{R}(f))$, **(iii)** $\mathsf{R}^{p\|}(f) = \tilde{\Omega}(\mathsf{R}(f))$, **(iv)** $\mathsf{Q}^{p\|}(f) = 1$.

Note that the function f in the above theorem perfectly parallelizes quantumly but does not parallelize at all classically. We discuss the construction of f in more detail in the technical overview. We totalize f using the cheatsheet framework to give a total function with no sequential quantum advantage but unbounded parallel quantum advantage, giving rise to the following theorem.

▶ **Theorem 4.** *There exists a (total) Boolean function $h : \{0,1\}^N \to \{0,1\}$ such that for some p and some constants $\epsilon, \delta > 0$ (with $\delta < \epsilon$), we have*

(i) $\mathsf{R}(h) = \tilde{\Theta}(N^{\epsilon})$, **(ii)** $\mathsf{Q}(h) = \tilde{\Omega}(\mathsf{R}(h))$, **(iii)** $\mathsf{R}^{p\|}(h) = \tilde{\Omega}(N^{\delta})$, **(iv)** $\mathsf{Q}^{p\|}(h) \leq 3$.

We also obtain analogous separations between the randomized and deterministic query complexities using the same techniques.

1.1.3 Separations with two layers of adaptivity

We earlier noted (in Theorem 2) that for some total function h, $\mathsf{Q}^{3\perp}(h)$ can be polynomially smaller than $\mathsf{R}^{3\perp}(h)$ (and even $\mathsf{R}(h)$), however $\mathsf{Q}^{1\perp}(h) = \Theta(\mathsf{D}^{1\perp}(h))$ for all total functions h. A natural question to ask is whether it is possible for $\mathsf{Q}^{2\perp}(h)$ to be much smaller than $\mathsf{R}^{2\perp}(h)$ for some total h? We answer this question in the affirmative by presenting a general framework that transforms an (at least) polynomial separation between $\mathsf{Q}^{1\perp}(f)$ and $\mathsf{R}^{1\perp}(f)$ for some partial f to a polynomial separation between $\mathsf{Q}^{2\perp}(h)$ and $\mathsf{R}^{2\perp}(h)$ for some total h constructed from f. Since there exists a partial function f (such as Forrelation in [2]) with $\mathsf{Q}(f) = 1$ (so $\mathsf{Q}^{1\perp}(f) = 1$) and $\mathsf{R}(f) = \Omega(N^\epsilon)$ (so $\mathsf{R}^{1\perp}(f) = \Omega(N^\epsilon)$) for some $\epsilon > 0$, we get a total function h exhibiting a polynomial separation between $\mathsf{Q}^{2\perp}(h)$ and $\mathsf{R}^{2\perp}(h)$. Precisely, we show the following theorem:

▶ **Theorem 5.** *Let $f : \mathcal{D} \to \{0,1\}$ with $\mathcal{D} \subseteq \{0,1\}^N$ be a (partial) function that satisfies $\mathsf{Q}^{1\perp}(f) = \tilde{O}(N^\epsilon)$ and $\mathsf{R}^{1\perp}(f) = \tilde{\Omega}(N^\delta)$ for some constants $\epsilon < \delta$. Then there exists a total $h : \{0,1\}^M \to \{0,1\}$ (with $M = \Theta(N^3)$) and constants $\epsilon' < \delta'$ such that $\mathsf{Q}^{2\perp}(h) = \tilde{O}(M^{\epsilon'})$ and $\mathsf{R}^{2\perp}(h) = \tilde{\Omega}(M^{\delta'})$.*

We also obtain a similar separation between $\mathsf{R}^{2\perp}$ and $\mathsf{D}^{2\perp}$.

1.1.4 Quantum parallel lower bound framework and applications

Finally, we develop a new method for deriving parallel quantum query lower bounds from sequential quantum query upper bounds. It is simpler to reason about sequential algorithms than parallel algorithms, and upper bounds are often easier to give than lower bounds. Thus, our result allows reducing the hard problem of lower bounding parallel quantum query complexity to the potentially easier problem of upper bounding its sequential analogue. Our technique is based on the parallel spectral adversary method [27], and is especially well suited to problems with optimal adversary matrices that only distinguish input pairs that differ at a single index. We also show that, for any total function f, the combinatorial parallel adversary method [25, 13] fails to show a lower bound better than $\Omega\left(\sqrt{\left\lceil \frac{\mathsf{C}_0(f)}{p} \right\rceil \left\lceil \frac{\mathsf{C}_1(f)}{p} \right\rceil}\right)$ where C_b denotes the bth certificate complexity of f. Moreover, we show that our method surpasses this barrier and provides a better lower bound, for instance, to the And $\circ$ Or problem.

We let $\lambda(f)$ denote the spectral sensitivity of f, as defined in [4]. Let $\mathcal{F}^{(f)}_{p-\text{res}}$ denote the set of all functions obtained from restricting f to p input bits, and fixing the rest. We call this a $p-$restriction of f.

▶ **Theorem 6.** *For any total Boolean function $f : \{0,1\}^N \to \{0,1\}$, we have*

$$\mathsf{Q}^{p\|}(f) = \Omega\left(\lambda(f) \cdot \frac{1}{\max_{g \in \mathcal{F}^{(f)}_{p-res}} \lambda(g)}\right)$$

Recalling that $\lambda(f) = O(\mathsf{Q}(f))$, as shown by Aaronson et al. [4], we can write this as $\mathsf{Q}^{p\|}(f) = \Omega\left(\lambda(f) \cdot (\max_{g \in \mathcal{F}^{(f)}_{p-\text{res}}} \mathsf{Q}(g))^{-1}\right)$, where we note that the second term can be lower bounded by giving a (sequential) quantum algorithm for any $p-$restriction of f. We apply Theorem 6 to the following classes of well-studied functions.

1. *Read-once formulas*: While the combinatorial adversary method [5] is sufficient to lower bound the sequential quantum query complexity of the two-layer AND ∘ OR tree by $\Omega(\sqrt{N})$ [9], the best lower bound the parallel combinatorial adversary method can give for the Read-once formula problem $\mathsf{And}_{\sqrt{N}} \circ \mathsf{Or}_{\sqrt{N}}$ is $\Omega(\sqrt{N}/p)$. Using Theorem 6, we show a $\Omega\left(\sqrt{N/p}\right)$ lower bound for any read-once formula.
2. *Symmetric functions*: From Theorem 6, we get the tight lower bound of $\Omega\left(\sqrt{Nt/p \cdot \min\{t,p\}}\right)$ where t is the largest hamming weight less than $N/2$ such that the function value either differs on inputs of hamming weight t and $t+1$ or differs on inputs of hamming weight $N-t$ and $N-t-1$. This result recovers the combinatorial adversary lower bound implicit in [25] using an arguably simpler proof.

1.2 Related work

Prior work on parallel quantum algorithms in the query model has primarily demonstrated the need for quantum depth in solving certain problems.

Cryptography

There has been a recent line of work studying problems which can only be efficiently solved by quantum algorithms with high depth [15, 7, 16, 17, 21], constructing cryptographic proofs of quantum depth. In this vein, Chung et al [19] and Blocki et al [11] show parallel quantum lower bounds against producing the output of an iterated hash function to give a cryptographic proof of sequential work that is secure against quantum computers.

Partial functions

Burchard [13] gives a characterization of parallel quantum approximate counting, showing that, approximating to multiplicative error ϵ, the number K of marked elements in a list of size N requires $\Omega\left(\frac{\sqrt{N}}{\epsilon\sqrt{pK}}\right)$ p-parallel quantum queries. Girish et al [22] constructs a partial function which exhibits a large total query separation between r-adaptive and $r-1$-adaptive quantum algorithms for any constant r.

Circuit complexity

Another setting one could study the intersection of quantum and parallelism is in the circuit model. For instance, Cleve and Watrous [20] show how to implement the quantum fourier transform on n qubits in $O(\log n)$ depth, allowing for a highly parallel implementation of Shor's factoring algorithm. There are known examples of relational problems which can be computed in constant quantum depth, yet require at least logarithmic depth for classical circuits [12, 35, 36]. More generally, quantum circuit classes such as QAC_n and QNC_n have been extensively studied [23, 26, 34].

2 Technical overview

In this section, we outline the techniques used in showing our main results. For our results involving separations between query complexity measures, we will primarily describe the techniques involved in separating quantum from randomized query complexity measures unless the techniques for separating randomized and deterministic query complexity measures are significantly different.

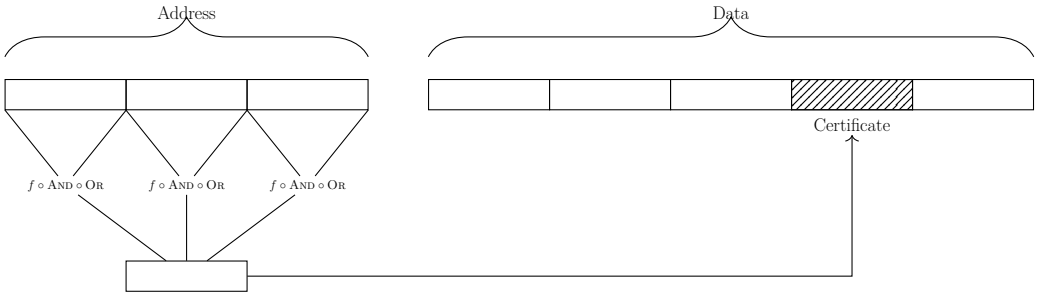

Figure 2 The canonical cheat sheet function f_{ccs} which lifts a partial function f to a total function, while retaining some of the speedup of f.

2.1 Unbounded parallel separations for total functions

The cheatsheet framework [3] lifts a partial function f, potentially with an exponential quantum query advantage to a total function that retains some of the advantage (although now polynomial). The natural way to do this is to define a total function such that when the input is in the domain of f, answer as f does, and when it is not answer 0. The cheat sheet framework does this "domain check" in a clever way so that the quantum advantage is not lost in the process. As shown in Figure 2, the input for the total function f_{ccs} is divided into two components, the Address section and the Data section. The partial function f is composed with a total function that has low certificate complexity, which in this case is $\textsc{And} \circ \textsc{Or}$. This composition $f \circ \textsc{And} \circ \textsc{Or}$ produces a single bit. This is repeated such that enough bits to address a block in the data component are obtained. If this block contains the certificate (or "cheatsheet") for the $\textsc{And} \circ \textsc{Or}$'s, and these certify that the input of f is in its domain, then the output of f_{ccs} is 1, else, the output is 0. As the data component is large, an algorithm is forced to solve the partial function f to obtain the address of the block, and can use the certificate present there to check whether the input of f is in the domain (and thus does not need to read the full Address to perform the domain check).

Thus, computing a cheatsheet function involves

1. Solving $f \circ \textsc{And} \circ \textsc{Or}$ to get the certificate location,
2. Reading out the certificate,
3. Checking the validity of the input of f.

For concreteness, consider the canonical cheatsheet function defined by Aaronson et al [3] composing $f = \textsc{Forrelation}$ [1, 2] on N bits with $\textsc{And} \circ \textsc{Or}$ on N^2 bits. This function was originally constructed to exhibit a superquadratic separation between sequential quantum and randomized query complexities. Observe that with $p = N^2$ parallelism, solving any given instance of the internal $\textsc{And} \circ \textsc{Or}$ can be done in a single p-parallel query by reading out all the inputs. A quantum algorithm can utilize this to quickly solve the whole composition, as Forrelation is quantumly easy. However, any classical algorithm would seem to need $\mathsf{R}(\textsc{Forrelation})$ p-parallel queries to solve this composition since it requires at least $p \cdot \mathsf{R}(\textsc{Forrelation})$ sequential queries. Thus, with p parallelism, step (1) is quantum efficient but not classically efficient. We show that for the canonical choice of parameters, steps (2) and (3) are also possible with few p-parallel queries, resulting in the desired separation.

The same technique can be used to give an exponential total function separation between randomized and deterministic parallel query complexity.

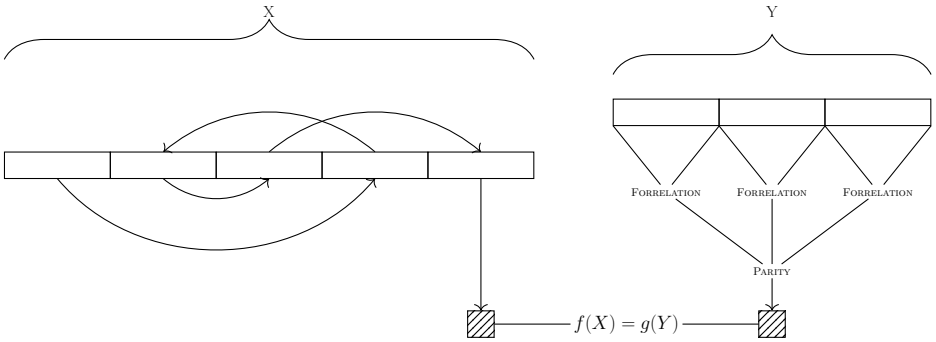

Figure 3 Partial function that admits unbounded genuine parallel separation.

Our result, combined with an aforementioned result of [27], which states that $\mathsf{D}^{p\|}(h) = \mathsf{poly}(\mathsf{Q}^{p\|}(h))$ for all total h and $p = O(\mathsf{bs}(h)^{1-\epsilon})$ with $\epsilon > 0$, provides a new way of upper bounding the block sensitivity of total Boolean functions. In particular, using our separation and a lower bound argument for the canonical cheatsheet function, we characterize its block sensitivity.

2.2 Unbounded genuine parallel separations

Is some polynomial sequential separation necessary for unbounded parallel quantum speedup, or can the advantage emerge purely from a quantum algorithm's ability to utilize parallelism (which we call a genuine parallel advantage)? We argue that the latter can indeed be the case, by first describing our construction for the partial function h that allows proving Theorem 3. As shown in Figure 3, the input to h here involves two components X and Y. The function f requires a sequence of adaptive queries to solve (in either model) whereas the function g achieves quantum-randomized separation and can be fully parallelized. We are promised that the output of both will be the same, so any algorithm can choose to solve either. We then argue that any algorithm will require solving at least one of f and g.

For concreteness, let $\epsilon > 0$ and suppose that $\mathsf{Q}(f) = \Theta(\mathsf{R}(f)) = \Theta(N^{\epsilon})$ and $\mathsf{R}^{p\|}(f) = \Omega(N^{\epsilon})$, and $\mathsf{Q}^{p\|}(g) = \Theta(N^{\epsilon})/p$ and $\mathsf{R}^{p\|}(g) = \Theta(N^{2\epsilon})/p$ for small enough p.[4] Then, a randomized sequential algorithm can choose to solve f, while a quantum sequential algorithm will not benefit from choosing to solve either f or g. Thus, $\mathsf{R}(h) = O(N^{\epsilon})$ and we would have $\mathsf{Q}(h) = \Omega(N^{\epsilon})$. Similarly, for $p = \mathsf{Q}(g)$, a p-parallel quantum algorithm can choose to solve g, while a p-parallel randomized algorithm will not benefit from choosing to solve either f or g. Therefore, $\mathsf{Q}^{p\|}(h) = 1$ and we would have $\mathsf{R}^{p\|} = \Omega(N^{\epsilon})$.

To show our desired randomized parallel and quantum lower bounds, we consider the general problem, which we call COR, where we are given inputs X and Y to arbitrary functions f and g respectively along with the promise that $f(X) = g(Y)$, our goal is to output $f(X) = g(Y)$. We use the hybrid argument to establish that any randomized parallel algorithm that solves $\textsc{cor}(f, g)$ will be able to either distinguish an input sampled from a hard 0-distribution for f from an input sampled from a hard 1-distribution for f, or will succeed at a similar distinguishing task for g. Therefore, we must have $\mathsf{R}^{p\|}(\textsc{cor}(f, g)) = \Omega(\min(\mathsf{R}^{p\|}(f), \mathsf{R}^{p\|}(g)))$. For the quantum lower bound, we show that that for any adversary

[4] For more concreteness, one may think of f as the pointer chasing function with chain length N^{ϵ} and g as the composition of parity on N^{ϵ} bits composed with FORRELATION on $N^{2\epsilon}$ bits as shown in Figure 3.

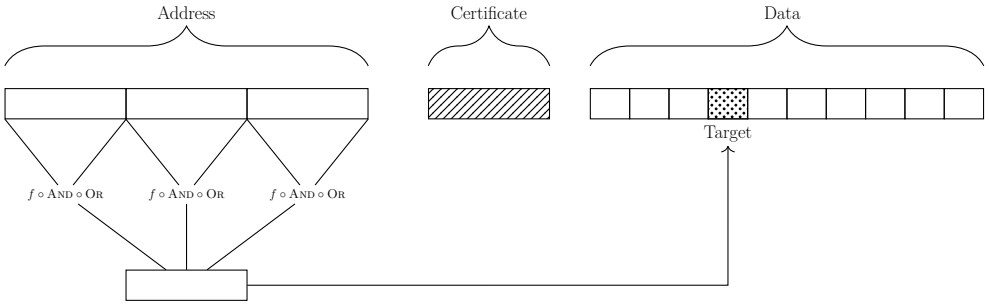

Figure 4 A framework that converts a partial function separation with one layer of adaptivity to a total function with two layers of adaptivity.

matrices $\Gamma^{(f)}$ and $\Gamma^{(g)}$ for f and g respectively, the matrix $\Gamma = \Gamma^{(f)} \otimes \Gamma^{(g)}$ is an adversary matrix for $\textsc{cor}(f,g)$ and $\max_i \|\Gamma_i\| = \max\left(\max_i \left\|\Gamma_i^{(f)}\right\|, \max_i \left\|\Gamma_i^{(g)}\right\|\right)$[5] It follows that $\mathsf{Adv}\left(\textsc{cor}(f,g)\right) = \Omega(\min(\mathsf{Adv}(f), \mathsf{Adv}(g)))$, which implies the desired lower bound.

Next, we describe a way to totalize the partial function h that we constructed in the previous section while maintaining some of its properties. We will use the cheatsheet framework. However, we will need a function with relatively small certificate complexity that does not admit any significant quantum speed-up so $\textsc{And} \circ \textsc{Or}$ will not work. Fortunately, [3] found a function, which they called Bkk, that satisfies $\mathsf{Q}(\textsc{Bkk}_N) = \tilde{\Theta}(\textsc{Bkk}_N) = \tilde{\Theta}(N)$ and $\mathsf{C}(\textsc{Bkk}_N) = \tilde{O}(\sqrt{N})$. We compose h on N bits with Bkk on N^2 bits, and then plug it into the cheatsheet framework. The desired upper bounds for $\mathsf{Q}^{p\|}$ and R in Theorem 4 are relatively straightforward and follows from the discussion in the previous sections. The quantum lower bound follows from quantum query complexity composition theorem and results in [3]. For the randomized parallel lower bound, we show a composition theorem where the inner function is Bkk.

The same lower bound techniques do not follow for the analogous separation between the randomized and deterministic query complexity measures. In particular, there is no general randomized composition theorem. Fortunately, for our constructions, the known randomized composition theorems suffice. For the deterministic parallel lower bound, we show a general composition theorem when the degree of the inner function is almost full, which might be of independent interest.

2.3 Separations with two layers of adaptivity

As we noted earlier (see Theorem 2), the cheat sheet framework can be employed to show a polynomial separation between $\mathsf{Q}^{3\perp}$ and $\mathsf{R}^{3\perp}$. We modify this framework to show a polynomial separation between $\mathsf{Q}^{2\perp}$ and $\mathsf{R}^{2\perp}$. As shown in Figure 4, our strategy is essentially to separate the cheatsheet into two parts: the first part contains the certificate, and the second just contains a long data string with one relevant index. Let f be a partial function with a polynomial separation between $\mathsf{Q}^{1\perp}(f)$ and $\mathsf{R}^{1\perp}(f)$. Informally, embedding f in our framework results in a function that requires (1) verifying the input to the f is in the domain (using the certificate) and if so, (2) outputting the bit of the data string present at the address obtained by solving the f.

[5] All the max are over relevant indices.

The quantum algorithm easily succeeds as follows. In the first round, it can read the certificate as well as solve f to obtain the address. In the second round, it verifies the certificate and queries the bit at the right address in the data string, hence is able to output the result. The randomized algorithm is unable to succeed in the same amount of parallelism because after the first round, it cannot compute f, so it would not find the right address by the first round, and can only make a query to the right address by the second round if it guesses the location correctly, which happens with very low probability. We also note that there is a caveat: the structure of a zero-certificate might be easy to distinguish from the structure of a one-certificate, allowing the algorithm to infer the inputs to the partial function from just the structure of the certificates. In that case, the randomized algorithm can use this to compute the partial function in the first round itself. To prevent this from happening, we use "bi-certificates" instead of certificates, where a "bi-certificate" corresponds to a set of indices that could certify both a zero and a one instance. This prevents the randomized algorithm from learning any information just by knowing the certificate.

Therefore, we present a framework to lift a partial function $\mathsf{Q}^{1\perp}$ vs $\mathsf{R}^{1\perp}$ separation to a total function $\mathsf{Q}^{2\perp}$ vs $\mathsf{R}^{2\perp}$ separation. The analogous result holds for randomized vs deterministic algorithms.

2.4 Quantum parallel lower bound framework and applications

There are few known techniques for lower bounding parallel quantum query complexity. Consider for instance lower bounding the quantum parallel query complexity for the balanced $\textsc{And} \circ \textsc{Or}$ function,

which is an example of a read-once formula. It is well known that this function has sequential quantum query complexity $\Theta(\sqrt{N})$ [9, 29]. Since both $\mathsf{Q}^{p\|}(\textsc{And})$ and $\mathsf{Q}^{p\|}(\textsc{Or})$ are $\Omega\left(\sqrt{N/p}\right)$, a natural guess for $\mathsf{Q}^{p\|}(\textsc{And} \circ \textsc{Or})$ is $\Omega\left(\sqrt{N/p}\right)$.

The bound $\mathsf{Q}(\textsc{And} \circ \textsc{Or}) = \Omega(\sqrt{N})$ was shown using the (sequential) combinatorial adversary method. However, as mentioned in Section 1.1.4, the corresponding parallel version fails to show a lower bound better than $\Omega\left(\sqrt{\left\lceil \frac{\mathsf{C}_0(f)}{p} \right\rceil \left\lceil \frac{\mathsf{C}_1(f)}{p} \right\rceil}\right)$ for any function f. This means that the best lower bound this method could help prove for $\textsc{And} \circ \textsc{Or}$ is $\Omega(\sqrt{N}/p)$, since $\mathsf{C}_0(\textsc{And} \circ \textsc{Or}) = \mathsf{C}_1(\textsc{And} \circ \textsc{Or}) = \sqrt{N}$.

We use the parallel spectral adversary method of [27], which is known to be optimal, to derive a method that is easier to apply and is sometimes stronger than the combinatorial adversary method (see Theorem 6). For the case of read-once formulas, we use the adversary sets of [9] to construct an adversary matrix Γ. It is easy to lower bound $\|\Gamma\|$ by a simple counting argument, but upper bounding $\|\Gamma_S\|$ for all $S \subseteq [N]$ with $|S| = p$ can be challenging. We show that if Γ satisfies the property that $\Gamma[x, y] = 0$ for all x, y that differs in more than 1 bit, then all the induced Γ_S can be rearranged to form block-diagonal matrices with blocks of size $2^p \times 2^p$. Moreover, each of these blocks are adversary matrices for some restricted function $g \in \mathcal{F}^{(f)}_{p-\text{res}}$ (Figure 5), where a restricted version of f is one where all but p input bits are fixed and known by the algorithm. Thus, we know that the spectral norm of these blocks is upper bounded by $\mathsf{Q}(g)$ (up to the normalizing factor of $\max_{i\in[2^p]} \|\Gamma_i\|$, and with a max taken over all $g \in \mathcal{F}^{(f)}_{p-\text{res}}$), which we can use to upper bound the spectral norm of any Γ_S. In our case, noting that g is a read-once formula of size p, we have $Q(g) = O(\sqrt{p})$ and we obtain the desired lower bound of $\Omega\left(\sqrt{N/p}\right)$. Hence, we are able to reduce the task of finding parallel lower bounds to the potentially easier task of finding sequential upper bounds.

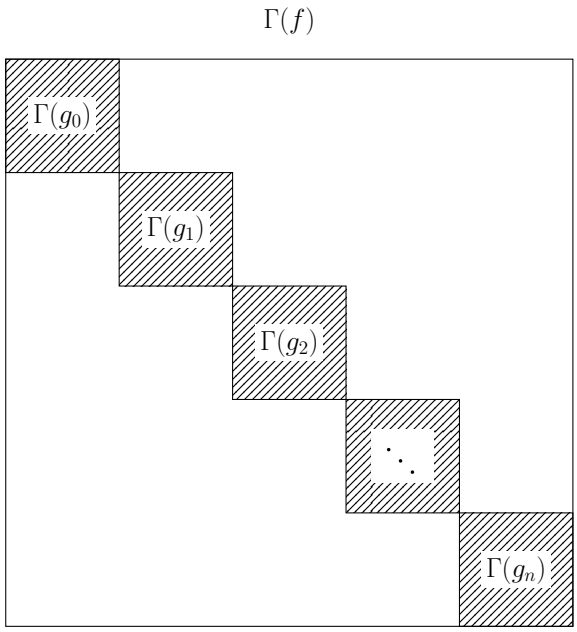

Figure 5 Nearest neighbor adversaries can be block-diagonalized, where the blocks are adversary matrices for restricted versions of the function f.

3 Conclusion

In this work, we have presented a few results on parallel quantum query complexity (See [14] for the full version). *A priori*, it seems unintuitive that an exponential advantage in number of rounds is possible for total functions, and also seems like lower bounding a simple function like AND ∘ OR in the parallel setting should be possible by standard techniques. Our results in Section 1.1.1 and Section 1.1.4 show that former is indeed true, and the latter is not the case and in fact requires "parallel-specific" thinking. Thus, one of our contributions is highlighting that there are still several fundamental query complexity questions unanswered in the parallel setting. We conclude by discussing a couple of these.

Limitations of parallel speedup

Can classical (randomized or deterministic) query algorithms simulate quantum query algorithms when allowed *both* a polynomial overhead in parallelism as well as number of rounds (as opposed to just a polynomial overhead in number of rounds as in Conjecture 1)? We conjecture that this is true:

▶ **Conjecture 7.** *For all total functions $f : \{0,1\}^M \to \{0,1\}$ and parallelism p, we have* $\mathsf{D}^{poly(p)\|}(f) = poly(\mathsf{Q}^{p\|}(f))$.

- When $p < \mathsf{bs}(f)^{1-\epsilon}$ for any $\epsilon > 0$, [27] prove that this is true.
- When $p \geq \mathsf{bs}(f) = \Omega(M^{\epsilon})$ for any $\epsilon > 0$, clearly this is true. This is the case in our result as well, where the canonical cheat sheet function was used to show Theorem 2 has $\mathsf{bs}(f) = \tilde{\Omega}(M^{1/6})$ and $\mathsf{D}^{p^{3/2}\|} = O(\mathsf{Q}^{p\|}(f)^3)$ for any p.
- Thus, the conjecture must be proven/falsified in the regime where bs and p are subpolynomial but superconstant. Moreover, what is the smallest power of p required to show this for all total f? Analogous questions remain open for $\mathsf{D}^{p\|}$ vs $\mathsf{R}^{p\|}$ as well.

Parallel composition theorem

It is known that $\mathsf{Q}(f \circ g) = \Theta(\mathsf{Q}(f) \cdot \mathsf{Q}(g))$ for boolean decision functions, a widely applicable and powerful result [30]. Is there an analogous theorem for p-parallel query complexity? In particular, do we have

$$\mathsf{Q}^{p\|}(f \circ g) = \Theta\left(\min_{q \in [p]} \mathsf{Q}^{q\|}(f) \cdot \mathsf{Q}^{\lceil p/q \rceil \|}(g)\right)$$

for boolean f, g? Such a result would suffice to reproduce our lower bound for $\textsc{And} \circ \textsc{Or}$, and likely be widely applicable for understanding parallel quantum query complexity. A straightforward attempt to show composition of the p-parallel adversary quantity using the reduction in [25] fails. This is because the partial function f' whose sequential query complexity characterizes the p-parallel complexity of f is no longer boolean.

Natural functions giving unbounded separation in rounds

Our results in Section 1.1.1 and Section 1.1.2 answer affirmatively whether it is possible for total functions to have unbounded separations in rounds, and whether this is still possible when no sequential separation exists. However, finding more natural functions to answer both these questions remains open and could help in understanding the interplay between quantum algorithms and parallel queries better.

References

1 Scott Aaronson. Bqp and the polynomial hierarchy. In *Proceedings of the Forty-Second ACM Symposium on Theory of Computing*, pages 141–150, 2010. doi:10.1145/1806689.1806711.

2 Scott Aaronson and Andris Ambainis. Forrelation: A problem that optimally separates quantum from classical computing. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, pages 307–316, 2015. doi:10.1145/2746539.2746547.

3 Scott Aaronson, Shalev Ben-David, and Robin Kothari. Separations in query complexity using cheat sheets. In *Proceedings of the Forty-Eighth Annual ACM Symposium on Theory of Computing*, pages 863–876. Association for Computing Machinery, 2016. doi:10.1145/2897518.2897644.

4 Scott Aaronson, Shalev Ben-David, Robin Kothari, Shravas Rao, and Avishay Tal. Degree vs. approximate degree and quantum implications of huang's sensitivity theorem. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1330–1342, 2021. doi:10.1145/3406325.3451047.

5 Andris Ambainis. Quantum lower bounds by quantum arguments. *Journal of Computer and System Sciences*, 64(4):750–767, 2002. doi:10.1006/jcss.2002.1826.

6 Andris Ambainis, Kaspars Balodis, Aleksandrs Belovs, Troy Lee, Miklos Santha, and Juris Smotrovs. Separations in query complexity based on pointer functions. *Journal of the ACM*, 64(5), 2017. doi:10.1145/3106234.

7 Atul Singh Arora, Andrea Coladangelo, Matthew Coudron, Alexandru Gheorghiu, Uttam Singh, and Hendrik Waldner. Quantum depth in the random oracle model. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023, pages 1111–1124, New York, NY, USA, 2023. Association for Computing Machinery. doi:10.1145/3564246.3585153.

8 Nikhil Bansal and Makrand Sinha. K-forrelation optimally separates quantum and classical query complexity. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1303–1316. Association for Computing Machinery, 2021. doi:10.1145/3406325.3451040.

9 Howard Barnum and Michael Saks. A lower bound on the quantum query complexity of read-once functions. *Journal of Computer and System Sciences*, 69(2):244–258, 2004. doi:10.1016/j.jcss.2004.02.002.

10 Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. In *Proceedings of the 39th Annual Symposium on Foundations of Computer Science*, 1998. doi:10.1145/502090.502097.

11 Jeremiah Blocki, Seunghoon Lee, and Samson Zhou. On the Security of Proofs of Sequential Work in a Post-Quantum World. In Stefano Tessaro, editor, *2nd Conference on Information-Theoretic Cryptography (ITC 2021)*, volume 199 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 22:1–22:27, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITC.2021.22.

12 Sergey Bravyi, David Gosset, and Robert Koenig. Quantum advantage with shallow circuits. *Science*, 362(6412):308–311, 2018. doi:10.1126/science.aar3106.

13 Paul Burchard. Lower bounds for parallel quantum counting. *ArXiv*, abs/1910.04555, 2019. arXiv:1910.04555.

14 Joseph Carolan, Amin Shiraz Gilani, and Mahathi Vempati. Quantum advantage and lower bounds for parallel query complexity, 2024. Full technical version of this paper, included in the QIP 2025 submission.

15 Nai-Hui Chia, Kai-Min Chung, and Ching-Yi Lai. On the need for large quantum depth. *J. ACM*, 70(1), January 2023. doi:10.1145/3570637.

16 Nai-Hui Chia and Shih-Han Hung. Classical verification of quantum depth, 2022. arXiv:2205.04656, doi:10.48550/arXiv.2205.04656.

17 Nai-Hui Chia and Shih-Han Hung. Non-interactive classical verification of quantum depth: A fine-grained characterization. Cryptology ePrint Archive, Paper 2023/1911, 2023. URL: https://eprint.iacr.org/2023/1911.

18 Andrew Childs, Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann, and Daniel A. Spielman. Exponential algorithmic speedup by a quantum walk. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, pages 59–68, 2003. doi:10.1145/780542.780552.

19 Kai-Min Chung, Serge Fehr, Yu-Hsuan Huang, and Tai-Ning Liao. On the compressed-oracle technique, and post-quantum security of proofs of sequential work. In Anne Canteaut and François-Xavier Standaert, editors, *Advances in Cryptology – EUROCRYPT 2021*, pages 598–629, Cham, 2021. Springer International Publishing. doi:10.1007/978-3-030-77886-6_21.

20 R. Cleve and J. Watrous. Fast parallel circuits for the quantum fourier transform. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*, pages 526–536, 2000. doi:10.1109/SFCS.2000.892140.

21 Matthew Coudron and Sanketh Menda. Computations with greater quantum depth are strictly more powerful (relative to an oracle). In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2020, pages 889–901, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3357713.3384269.

22 Uma Girish, Makrand Sinha, Avishay Tal, and Kewen Wu. The power of adaptivity in quantum query algorithms, 2023. doi:10.48550/arXiv.2311.16057.

23 Frederic Green, Steven T. Homer, Cristopher Moore, and Chris Pollett. Counting, fanout and the complexity of quantum acc. *Quantum Inf. Comput.*, 2:35–65, 2001. URL: https://api.semanticscholar.org/CorpusID:7821761, doi:10.26421/QIC2.1-3.

24 Lov Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, pages 212–219, 1996. doi:10.1145/237814.237866.

25 Lov K. Grover. A fast quantum mechanical algorithm for database search. In *Symposium on the Theory of Computing*, 1996. URL: https://api.semanticscholar.org/CorpusID:207198067.

26 Peter Høyer and Robert Špalek. Quantum fan-out is powerful. *Theory of Computing*, 1(5):81–103, 2005. doi:10.4086/toc.2005.v001a005.

27 Stacey Jeffery, Frederic Magniez, and Ronald Wolf. Optimal parallel quantum query algorithms. *Algorithmica*, 79(2):509–529, 2017. doi:10.1007/s00453-016-0206-z.

28 Ashley Montanaro. Nonadaptive quantum query complexity. *Information Processing Letters*, 110(24):1110–1113, 2010. doi:10.1016/J.IPL.2010.09.009.

29 Ben W. Reichardt. Faster quantum algorithm for evaluating game trees. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '11, pages 546–559, USA, 2011. Society for Industrial and Applied Mathematics. doi:10.1137/1.9781611973082.43.

30 Ben W. Reichardt. Reflections for quantum query algorithms. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '11, pages 560–569, USA, 2011. Society for Industrial and Applied Mathematics. doi:10.1137/1.9781611973082.44.

31 Alexander A. Sherstov, Andrey A. Storozhenko, and Pei Wu. An optimal separation of randomized and quantum query complexity. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1289–1302, 2021. doi:10.1145/3406325.3451019.

32 Peter Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 124–134, 1994. doi:10.1109/SFCS.1994.365700.

33 Daniel Simon. On the power of quantum computation. *SIAM Journal on Computing*, 26(5):1474–1483, 1997. doi:10.1137/S0097539796298637.

34 Yasuhiro Takahashi and Seiichiro Tani. Collapse of the hierarchy of constant-depth exact quantum circuits. *computational complexity*, 25(4):849–881, 2016. doi:10.1007/s00037-016-0140-0.

35 Adam Bene Watts, Robin Kothari, Luke Schaeffer, and Avishay Tal. Exponential separation between shallow quantum circuits and unbounded fan-in shallow classical circuits. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, STOC 2019, pages 515–526, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3313276.3316404.

36 Adam Bene Watts and Natalie Parham. Unconditional quantum advantage for sampling with shallow circuits, 2024. doi:10.48550/arXiv.2301.00995.

37 Christof Zalka. Grover's quantum searching algorithm is optimal. *Phys. Rev. A*, 60:2746–2751, October 1999. doi:10.1103/PhysRevA.60.2746.