# The Power of Adaptivity in Quantum Query Algorithms

Uma Girish
Princeton University
Princeton, USA
ugirish@cs.princeton.edu

Makrand Sinha
University of Illinois at Urbana-Champaign
Champaign, USA
msinha@illinois.edu

Avishay Tal
University of California at Berkeley
Berkeley, USA
atal@berkeley.edu

Kewen Wu
University of California at Berkeley
Berkeley, USA
shlw_kevin@hotmail.com

## ABSTRACT

Motivated by limitations on the depth of near-term quantum devices, we study the depth-computation trade-off in the query model, where depth corresponds to the number of adaptive query rounds and the computation per layer corresponds to the number of parallel queries per round. We achieve the strongest known separation between quantum algorithms with $r$ versus $r-1$ rounds of adaptivity. We do so by using the $k$-fold Forrelation problem introduced by Aaronson and Ambainis (SICOMP'18). For $k = 2r$, this problem can be solved using an $r$ round quantum algorithm with only one query per round, yet we show that any $r-1$ round quantum algorithm needs an exponential (in the number of qubits) number of parallel queries per round.

Our results are proven following the Fourier analytic machinery developed in recent works on quantum-classical separations. The key new component in our result are bounds on the Fourier weights of quantum query algorithms with bounded number of rounds of adaptivity. These may be of independent interest as they distinguish the polynomials that arise from such algorithms from arbitrary bounded polynomials of the same degree.

## CCS CONCEPTS

• **Theory of computation** → **Quantum computation theory**; **Quantum complexity theory**; *Pseudorandomness and derandomization.*

## KEYWORDS

Quantum Query Algorithms, Query Adaptivity, Forrelation, Quantum Advantages, Fourier Analysis of Boolean Functions

**ACM Reference Format:**
Uma Girish, Makrand Sinha, Avishay Tal, and Kewen Wu. 2024. The Power of Adaptivity in Quantum Query Algorithms. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing (STOC '24), June 24–28, 2024, Vancouver, BC, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3618260.3649621


*STOC '24, June 24–28, 2024, Vancouver, BC, Canada*

ACM ISBN 979-8-4007-0383-6/24/06
https://doi.org/10.1145/3618260.3649621

## 1 INTRODUCTION

The quantum query model, also known as the *black-box* or *oracle* model, has been a very successful test bed to develop quantum algorithms and to give provable guarantees on speedups over classical algorithms. In this model, a quantum algorithm has "black-box access" to the input and is only charged for quantum queries to the input, while any intermediate computation is considered free. Most well-known quantum algorithms, such as Grover search [21], Deutsch-Josza algorithm [15], Bernstein-Vazirani algorithm [6], Simon's Algorithm [40], and Shor's period-finding algorithm [38], are captured by this black-box access model. There are slightly different models of black-box access to the input and in this work, we consider the most basic access model where each query returns a *bit* of the input.

*$k$-Fold Forrelation.* Traditionally, the focus in the query model has been to compare quantum algorithms with classical ones. The culmination of this line of work led to the resolution of the following speedup question:

*What is the largest quantum speedup that is possible over classical algorithms?*

The motivation for this question stems from an attempt to pinpoint the exact limit of quantum speedups, and it has helped us develop a better understanding of the fundamental nature of quantum speedups. In particular, towards this question, Aaronson and Ambainis [1] introduced the $k$-fold Forrelation problem: In this problem one evaluates a degree-$k$ polynomial that we denote by $\mathrm{forr}_k$ (see [19, Definition A.1]) that measures the "Fourier correlation" between $k$ Boolean functions mapping $\{\pm1\}^m$ to $\{\pm1\}$. The algorithm can make superposition queries to any value in the truth table of these functions and must distinguish the case when the value of the polynomial is large from the case where it is close to zero. This defines a partial Boolean function or a promise problem.

Letting $n = 2^m$, this problem can be solved with $r = \lceil \frac{k}{2} \rceil$ quantum queries with $\mathrm{polylog}(n)$-sized quantum circuits, while [1] also showed that it can be solved with $O(n^{1-1/2r})$ classical queries. They conjectured that this is tight. Moreover, they also conjectured that one should be able to simulate any $r$-query quantum algorithm with $O(n^{1-1/2r})$ classical queries, making this a problem where quantum algorithms have the *maximal advantage*. Up to low-order terms, the first conjecture was proven for $k$-fold Forrelation[1] and

[1] We note that the $k = 2$ case was already resolved by Aaronson and Ambainis [1] and a different proof follows from the work [34] as well.

its variants by Sherstov, Storozhenko, and Wu [37] and Bansal and Sinha [5], building on the work of Raz and Tal [34] and Tal [44]. As a complement, Bravyi, Gosset, Grier, and Schaeffer [7] extended the simulation result to arbitrary quantum query algorithms, showing that any $r$-query quantum algorithm can be classically simulated with $O(n^{1-1/2r})$ queries. The $k$-fold Forrelation also turns out to be one of the most natural problems that is BQP-complete [1] and its variants have also been proposed as candidates for other separations in quantum complexity theory [28], making it a fundamental problem to study in its own right.

*The Power of Adaptivity.* In this work, our focus is to identify the exact limits of quantum depth in the query model, analogous to the quantum speedup question. One of the primary motivations for studying the power of depth comes from near-term quantum hardware which is restricted to quantum circuits of small depth in order to combat decoherence due to noise. Because of depth limitations, one needs to use wider circuits with more gates in each layer to perform computation, thus making parallel operations quite desirable. This makes optimizing the depth-width trade-off a fundamental task in quantum circuit synthesis for the near-term: Reducing circuit depth allows the computation to be completed before the qubits decohere too much, but it also requires more quantum gates per layer.

On the positive side, Cleve and Watrous [13] showed how to implement the quantum Fourier transform in a parallel fashion, which leads to the parallelization of Shor's factoring algorithm [39]. Also, in a recent related work, Regev [35] employed parallelization followed by polynomial-time classical post-processing, to design a more efficient quantum algorithm for factoring under certain number-theoretic conjectures. On the other hand, Moore and Nilsson [32] conjectured that certain staircase-shaped quantum circuits cannot be efficiently parallelized.

In the query model abstraction, the circuit depth corresponds to the number of adaptive rounds, denoted by $r$, and the circuit width corresponds to the maximal number of parallel queries, denoted by $t$, per round. An extreme case $r = 1$ is the non-adaptive quantum query algorithm, where all queries are made in parallel. Perhaps surprisingly, van Dam [45] showed that any $n$-bit Boolean function can be computed with bounded error using only $t \le n/2 + O(\sqrt{n})$ non-adaptive quantum queries, which is essentially tight for total functions [31]. Techniques have been developed to establish lower bounds for various problems in this non-adaptive setting [8, 27, 33], but less is known when we have more adaptive rounds. Zalka [47] considered the unordered search problem on $n$-bit database and showed that $t = \Omega(n/r^2)$ is needed. This matches the simple divide-and-search algorithm: Partition the space into $O(n/r^2)$ parts of $O(r^2)$ size each and execute Grover's algorithm [21] on each part in parallel in $r$ steps. Jeffery, Magniez, and de Wolf [25] proved tight $t = \Theta(n/r^{3/2})$ trade-off for the element distinctness problem and tight $t = \Theta(n/r^{1+1/k})$ trade-off for the $k$-sum problem.

The above results show that being more adaptive indeed reduces the need of quantum queries. However the improvement is quite marginal: Even if we double the number of rounds, the saving is still only a constant factor. This naturally leads to the following question:

*What is the largest possible saving in queries offered by more rounds of adaptivity?*

## 1.1 The Main Separation

We answer the above question in the strongest sense and along the way prove structural theorems about the Fourier spectrum of polynomials that arise from low-depth quantum algorithms.

Our main result shows that the aforementioned $k$-fold Forrelation problem separates different levels of quantum computational power, measured in terms of adaptivity. Informally, the saving in the number of parallel queries can be unbounded, even when we just have one more adaptive round.

Theorem 1.1. *For any constant $r \ge 2$, the $2r$-fold Forrelation problem on $n$-bit inputs*

1. *can be solved with advantage $2^{-10r}$ by $r$ adaptive rounds of queries with one quantum query per round, yet*
2. *any quantum query algorithm with $r-1$ adaptive rounds requires $\widetilde{\Omega}(n^{1/r^2})$ parallel queries to approximate it.*

**Remark 1.2.** Item 2 continues to hold even in the presence of a large amount of classical pre-processing. In more detail, we consider algorithms that are allowed to first make classical queries and based on the outputs, choose a quantum algorithm to run that has $k-1$ rounds of $t$ parallel queries each. We show that any such algorithm must either make $\Omega(n^{1/(2r)})$ classical queries or $\widetilde{\Omega}(n^{1/r^2})$ quantum queries. See [19, Appendix C] for more details.

**Remark 1.3.** We note two easy modifications of the above theorem that also follow from our work, which we do not state in the theorem statement above for brevity. First, in the first item above, one can boost the advantage of the quantum algorithm to any constant close to 1 by making $2^{O(r)}$ parallel queries per round without increasing the number of rounds since error amplification can be done by making parallel queries. Second, we can more generally obtain an $r$ versus $r'$ separation for any $r' < r$ where the lower bound in the second item improves as $r'$ decreases and is of the form $\widetilde{\Omega}(n^{c(r,r')})$ where

$$c(r,r') = \begin{cases} 1 - \frac{1}{r} & \text{for } r' = 1, \\ \frac{r-r'}{rr'+r/2} \ge \frac{1}{r^2} & \text{for } 2 \le r' \le r-1. \end{cases}$$

For example, reducing the number of rounds by a factor of 2, i.e., when $r = 2r'$, gives $c(r,r') = 1/(r+1)$. Furthermore, notice that the case when $r' = 1$ corresponds to a non-adaptive lower bound: Here we obtain that any non-adaptive quantum algorithm that solves $2r$-fold Forrelation must make $\widetilde{\Omega}(n^{1-1/r})$ parallel queries.

**Remark 1.4.** We recall that $k$-fold Forrelation is a partial function and being a partial function is necessary for Item 1. [25, 47] showed that for any total Boolean function $f$, the number of parallel quantum queries needed with $r$ rounds is $t = \Omega(\mathrm{bs}(f)/r^2)$, where $\mathrm{bs}(f)$ is the block sensitivity complexity of $f$. Note that Simon [41] proved that $\mathrm{bs}(f) = \Omega(\log n)$ if $f$ is a non-degenerate $n$-bit Boolean function. This implies that $t = \Omega(\log n)$ when $r$ is a constant. Similarly, Ambainis and de Wolf [3] showed that any non-degenerate $n$-bit total function requires $\Omega(\log n/\log\log n)$ quantum queries in total, which implies $t = \Omega(\log n/(r\log\log n))$. In summary, for total functions and constant rounds, the best possible separation is

only logarithmic-vs-polynomial, instead of the $O(1)$-vs-polynomial separation we obtain.

As mentioned previously, Item 1 of Theorem 1.1 was already known since the work of [1] and the crux of our result is the lower bound in Item 2. Lower bounds for $k$-fold Forrelation are quite non-trivial to prove even for classical query algorithms and the known techniques rely on the polynomial method. The polynomial method cannot be directly applied since $k$-fold Forrelation is a low-degree bounded polynomial and as such one needs to find a way to distinguish it from the polynomials of much higher degree that are computed by the computational model of interest. In particular, previous works [5, 34, 37, 44] identified that if the polynomials computed by a computational model satisfy a certain refined notion of "sparsity", in terms of *bounded Fourier Growth*, then the $k$-fold Forrelation problem cannot be solved in that model.

## 1.2 Fourier Growth of Low-Depth Quantum Algorithms

Recall that every Boolean function $f\colon \{\pm1\}^n \to [0,1]$ has a unique Fourier representation

$$f(x) = \sum_{S\subseteq[n]} \widehat{f}(S)\cdot\prod_{i\in S} x_i,$$

where $\widehat{f}(S) = \mathbb{E}[f(x)\cdot\prod_{i\in S} x_i]$ is the Fourier coefficient and the expectation is over uniform $x$ over the hypercube $\{\pm1\}^n$. The level-$\ell$ Fourier $\ell_1$-weight $L_{1,\ell}(f)$ is defined by

$$L_{1,\ell}(f) = \sum_{|S|=\ell} \left|\widehat{f}(S)\right|$$

and is a measure of the capability of the function $f$ to aggregate weak signals on $\ell$ bits. Let $C$ be a class of Boolean functions, then the Fourier growth of $C$ refers to the scaling of $\max_{f\in C} L_{1,\ell}(f)$ when $\ell$ grows.

Following [34, 44], Bansal and Sinha [5] successfully related the advantage of approximating $k$-fold Forrelation for $k = 2r$ with the (low-level) Fourier growth of the model of computation in question. Informally, if the Fourier weights grow slower than $(\sqrt{n})^{(1-1/k)\ell}$ (which is the Fourier growth of the $k$-fold Forrelation polynomial) up to level $\ell = k^2$, then it cannot approximate $k$-fold Forrelation. As a direct application of [5, Theorem 3.2 and Theorem 3.4], Item 2 of Theorem 1.1 follows from the following Fourier growth bounds. The detailed calculations can be found in [19, Appendix A].

Theorem 1.5. *Let $\mathcal{A}$ be a quantum query algorithm on $n$-bit inputs with arbitrarily many auxiliary qubits. Assume $\mathcal{A}$ has $r$ adaptive rounds of $t \le n$ parallel queries. Define $f\colon \{\pm1\}^n \to [0,1]$ by $f(x) = \mathbf{Pr}[\mathcal{A}$ accepts $x]$. Then*

$$L_{1,\ell}(f) \le O_{r,\ell}\left(t^\ell\cdot\left(\sqrt{n/t}\right)^{\lfloor(1-\frac{1}{2r})\ell\rfloor}\right).$$

*Moreover, this bound holds when some bits of $x$ are fixed in advance.*

**Remark 1.6.** In the non-adaptive case (i.e., $r = 1$), the bound in Theorem 1.5 can be improved (see [19, Subsection 2.2] for detail) to

$$L_{1,\ell}(f) \le O_\ell\left(t^{\ell/4}\cdot n^{\ell/4}\right).$$

This implies the non-adaptive lower bound of $\widetilde{\Omega}(n^{1-1/r})$ parallel queries for solving the $2r$-fold Forrelation problem, as mentioned in Remark 1.3. This is also tight as shown in [19, Appendix B].

The acceptance probability of any quantum query algorithm that makes $d$ queries can be expressed as a degree-$2d$ bounded polynomial. Most of the techniques in the literature do not distinguish polynomials of quantum algorithms from general bounded polynomials and we lack a sufficiently good understanding of such distinctions.

Our Fourier growth bounds are far better than the bounds that can be obtained by directly applying the Fourier growth estimates for low-degree bounded polynomials [16, 23]. Thus, this points to one way in which polynomials computed by low-depth quantum algorithms are different than general bounded polynomials of the same degree.

*Classically Simulating Low-depth Quantum Algorithms.* We mention an open problem related to the question of where the exact limits of the trade-offs between depth and the number of parallel queries lie. As mentioned before, if there is only one query per round ($t = 1$), then Aaronson and Ambainis [1] conjectured that any $r$-round quantum algorithm can be simulated with $O(n^{1-1/2r})$ classical queries and this conjecture was proved by [7].

Does such a classical simulation continue to exist for low-depth quantum algorithms that make multiple parallel queries per round? We believe this is the case and make the following conjecture.

Conjecture 1.7. *Any quantum query algorithm on $n$-bit inputs with $r$ adaptive rounds and $t$ parallel queries per round can be classically simulated with $\widetilde{O}_{t,r}\left(n^{1-1/2r}\right)$ queries.*

It is worth mentioning that the Fourier growth bounds of classical query models (aka decision trees) [37, 44] scales roughly like $(D\cdot\log n)^{\ell/2}$ where $D$ is the number of classical queries. Our Fourier bound matches the one for decision trees of depth $\widetilde{O}_{r,t}\left(n^{1-1/2r}\right)$ giving some support to the above conjecture.

## 1.3 Related Works

*Related Works in Communication Models.* Aside from the aforementioned results in the quantum query complexity, the round-query trade-off in the query model can also be deduced from the round-communication trade-off in the model of communication complexity. In this model, Alice and Bob are given $n$-bit inputs $x$ and $y$ separately and their goal is to evaluate some function $F(x,y)$ by communication.

Given such a communication task $F$, we immediately get a query task $f$ by letting $z = (x,y)$ and defining $f(z) = F(x,y)$. Then each quantum query to $z$ can be implemented in the communication setting by Alice and Bob exchanging one round of $O(\log n)$ qubits.[2] Therefore if $F$ requires sending $t\log n$ qubits in each round, then the corresponding $f$ requires $\Omega(t)$ parallel queries in each round. Via this reduction, the pointer chasing problem with $r$ jumps needs $\widetilde{\Omega}_r(n)$ parallel queries with $r-1$ adaptive rounds [24, 26], whereas

[2]Here $\log n$ is required for indexing an $n$-bit string in superposition, which is not needed classically. By switching the role of Alice and Bob between communication rounds, we can simulate $r$ queries in $r$ rounds of communication and one party in the end will compute the answer.

it can be solved with $r$ adaptive rounds of $O(\log n)$ queries. Since the pointer chasing problem is a total function, by Remark 1.4 this logarithmic-vs-polynomial separation cannot be further improved to an $O(1)$-vs-polynomial separation.

We remark that[3] it is possible to define a variant of the pointer chasing problem which only uses one quantum query per round. This is achieved by using the Bernstein-Vazirani trick (see [46]) to encode the address of each jump by the Hadamard code. Note that this is a partial function (due to the Bernstein-Vazirani trick), and it is conceivable that it will require $n^{\Omega(1)}$ queries if the number of adaptive rounds is reduced. In light of this, we highlight that our results generalize to the setting of *quantum query algorithms with classical preprocessing*, where the algorithm is allowed to first perform $n^{\Omega(1)}$ classical queries, then adaptively choose a quantum query algorithm with prescribed number of rounds and parallel queries. See details in [19, Appendix C]. In this setting, variants of the pointer chasing problem would be solved already in the classical preprocessing phase, whereas the $2r$-fold Forrelation problem still exhibits an $O(1)$-vs-polynomial separation.

*Related Works in Hybrid Models.* There is another line of work on hybrid quantum-classical query algorithms that is related to the questions studied here. In particular, this line of work [4, 12, 14, 22] considers the trade-off between quantum depth and the number of classical queries in a model that allows both. Although some of these works prove a fine-grained depth separation that seems similar to ours, the models considered in these works do not allow parallel queries (or only allow polylog$(n)$-parallel queries in [14]) and they do not study the trade-offs between depth and parallel quantum queries. Consequently, these results are not comparable to ours.

*Related Works in Fourier Growth.* The study of Fourier growth dates back to Mansour [30] for learning theoretic purposes. More recently it has been successfully applied in the study of pseudorandomness [2, 9–11] and quantum-classical separations [5, 17, 18, 34, 37, 44]. Moreover, Fourier growth bounds have been established for various models of computation: Boolean circuit classes [30, 43], branching programs [11, 29, 36, 42], query models [5, 20, 37, 44], communication models [17, 18], and more. We refer interested readers to [18] for detailed discussion.

*Full Version.* Full version of our paper [19] is available at https://arxiv.org/abs/2311.16057.

## 2 PROOF OVERVIEW

We provide a proof overview here and the details can be found in the full version of the paper [19].

### 2.1 High-Level Proof Sketch

*Describing Quantum Algorithms with Parallel Queries.* Quantum algorithms which make parallel queries have the following form. First, we have an initial state $|u\rangle$; this state has some registers to index coordinates of the input and some registers for workspace. The algorithm has several rounds, where each round consists of a few parallel oracle queries followed by a unitary operator. The parallel queries are modelled by $O_x^{\otimes t} \otimes \boldsymbol{I}$. Here, $O_x$ is an $(n+1) \times (n+1)$ unitary that maps $|i\rangle$ to $x_i |i\rangle$ for all $i \in [n]$ and keeps $|0\rangle$ fixed, and this is equivalent to the usual quantum query oracle. The operator $O_x^{\otimes t}$ implements $t$ parallel oracle queries and $\boldsymbol{I}$ acts as the identity matrix on the workspace. Finally, the algorithm applies some two-outcome measurement and returns the outcome as the output. See Figure 1 for depiction.

For simplicity, let us imagine that there is no workspace memory. Additionally, let us ignore the action of the oracle $O_x$ on the basis state $|0\rangle$ and treat $O_x$ as an $n \times n$ unitary matrix. These simplifications are only for the proof overview, and our proof works in full generality. In this case, the acceptance probability of this algorithm can be expressed as

$$f(x) = u^\dagger O_x^{\otimes t} M_1 O_x^{\otimes t} \cdots M_{k-1} O_x^{\otimes t} v, \tag{1}$$

where $k$ is twice the number of rounds, $u = v$ corresponds to the initial state, $M_1 = M_{k-1}^\dagger, M_2 = M_{k-2}^\dagger, \ldots, M_{k/2-1} = M_{k/2+1}^\dagger$ are the $\frac{k}{2} - 1$ unitary operators applied by the quantum algorithm and $M_{k/2}$ is the final measurement operator. For the rest of our proof, we can forget about the exact details of these matrices, we will only need that $M_1, \ldots, M_{k-1}$ have bounded operator norm and $u, v$ are unit vectors.

*Fourier Growth of Quantum Algorithms.* Let us now understand the Fourier growth of functions as in (1) where $M_1, \ldots, M_{k-1}$ have bounded operator norm and $u, v$ are unit vectors. We first set up some notation. We use $I \in [n]^t$ to denote a $t$-tuple of elements in $[n]$. We can view $I$ as an ordered multiset of $[n]$ of size $t$ (when counted with multiplicity). Accordingly, we use $\oplus I$ to denote the set of elements that appear an odd number of times in $I$ and use $\oplus I \oplus I'$ to denote $(\oplus I) \oplus (\oplus I')$ for $I, I' \in [n]^t$.

When we expand the matrix multiplication in (1), many variables cancel out due to the identity $x_i^2 = 1$. Assume for simplicity that $u$ and $v$ are real vectors, i.e., $(u[I])^* = u[I]$. Thus, for all $S \subseteq [n]$, the coefficient of the monomial $\prod_{i \in S} x_i$ in (1) is given by

$$\widehat{f}(S) = \sum_{\substack{I_1, \ldots, I_k \in [n]^t \\ \oplus I_1 \oplus \ldots \oplus I_k = S}} u[I_1] M_1[I_1, I_2] M_2[I_2, I_3] \cdots M_{k-1}[I_{k-1}, I_k] v[I_k].$$

Fix complex numbers $\alpha_S = \widehat{f}(S)^* / |\widehat{f}(S)|$ for each $S \subseteq [n]$ of size $\ell$. We wish to upper bound $L_{1,\ell}(f) = \sum_{|S|=\ell} \left|\widehat{f}(S)\right| = \sum_{|S|=\ell} \alpha_S \cdot \widehat{f}(S)$, which by the above is

$$L_{1,\ell}(f) = \sum_{\substack{I_1, \ldots, I_k \in [n]^t \\ |\oplus I_1 \oplus \cdots \oplus I_k| = \ell}} \alpha[\oplus I_1 \oplus \cdots \oplus I_k] \cdot u[I_1] M_1[I_1, I_2] M_2[I_2, I_3] \cdots M_{k-1}[I_{k-1}, I_k] v[I_k]. \tag{2}$$

To highlight the difficulties in bounding (2), we first present a few failed approaches and then describe our high-level proof approach. First, let us focus on the base case $k = 2$. For ease of notation, we will switch from indices $I_1, I_2$ to indices $I, J$ and from the matrix $M_1$ to $M$. Our goal is to upper bound

$$L_{1,\ell}(f) = \sum_{\substack{I, J \in [n]^t \\ |\oplus I \oplus J| = \ell}} \alpha[\oplus I \oplus J] \cdot u[I] M[I, J] v[J].$$

[3] We thank an anonymous QIP'24 reviewer for pointing this out.

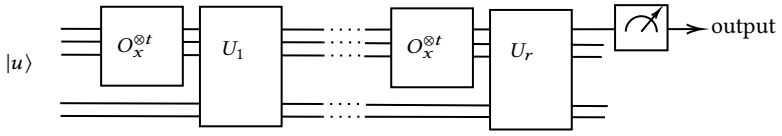


**Figure 1: Quantum algorithm with $r$ adaptive rounds of $t$ parallel queries each.**

One natural approach is to express $L_{1,\ell}(f)$ as a product of matrices (with bounded operator norms). One way to do this is to incorporate the phases $\alpha[\oplus I \oplus J]$ and the constraint $|\oplus I \oplus J| = \ell$ into the matrix $M[I, J]$. For instance, define $\widetilde{M}$ such that

$$\widetilde{M}[I, J] := \alpha[\oplus I \oplus J] \cdot 1[|\oplus I \oplus J| = \ell] \cdot M[I, J].$$

It is easy to see that $L_{1,\ell}(f) = u^\dagger \widetilde{M} v$ and consequently, $L_{1,\ell}(f) \leq \|\widetilde{M}\|$. What is the best upper bound that we can prove for $\|\widetilde{M}\|$? At first glance, it might seem that we cannot do better than $\sqrt{n^t}$. Indeed, given an $n^t \times n^t$ unitary matrix $M$, if we multiply each entry by arbitrary numbers in the unit disk, this could blow up the operator norm by as much as $\sqrt{n^t}$ (the Hadamard matrix gives a tight example of this). However, we can do much better. This is because the terms multiplying each entry of $M$ are highly constrained; the term multiplying the $(I, J)$-th entry depends only on $\oplus I \oplus J$.

To get an improved bound, consider the matrix $D$ whose rows and columns are indexed by all possible $\oplus I$ and $\oplus J$ respectively, and the $(\oplus I, \oplus J)$-th entry is $\alpha[\oplus I \oplus J] \cdot 1[|\oplus I \oplus J| = \ell]$. It is not too difficult to convince oneself that $\widetilde{M}$ is a sub-matrix of $M \otimes D$. Therefore, $\|\widetilde{M}\| \leq \|M\| \cdot \|D\| \leq \|D\|$. Now, what is the best upper bound we can show for $\|D\|$? Consider the row corresponding to $\oplus I = \emptyset$. For this row, we need to choose a column $\oplus J$ such that $|\oplus J| = \ell$ and there are $\binom{n}{\ell}$ such columns. This already means that $\|D\| \geq \sqrt{\binom{n}{\ell}}$ (and this turns out to be tight). While a bound of $L_{1,\ell}(f) \leq \sqrt{\binom{n}{\ell}}$ would already be a great improvement over the previous bound, it is still a trivial bound that holds for all bounded functions! Indeed, all Boolean functions which map into the complex unit disk satisfy $L_{1,\ell}(f) \leq \sqrt{\binom{n}{\ell}}$.

To get the optimal bound of $n^{\ell/4} \cdot t^{\ell/4}$, the idea is to reduce the operator norm of $D$. For instance, suppose we defined $\widetilde{D}$ to be $D$, except that we zero out entries for which $|\oplus I \setminus \oplus J| \neq \ell/2$ (or equivalently $|\oplus J \setminus \oplus I| \neq \ell/2$). In this case, for any fixed $\oplus I$, the number of possibilities for $\oplus J$ is only $\binom{n}{\ell/2} \cdot \binom{t}{\ell/2}$ and we can actually prove that $\|\widetilde{D}\| \leq n^{\ell/4} \cdot t^{\ell/4}$ as desired. Of course this doesn't suffice as we also need to sum over terms zeroed out.

In the full proof, the idea is to implicitly consider all possible values of $|\oplus I \setminus \oplus J|$. We fix any $\ell_1, \ell_2$ such that $|\oplus I \setminus \oplus J| = \ell_1$ and $|\oplus J \setminus \oplus I| = \ell_2$. Since $\ell_1 + \ell_2 = \ell$, either (1) $\ell_1 \leq \ell/2$ or (2) $\ell_2 \leq \ell/2$. We will define two different matrix product decompositions to handle each of these cases separately. It will turn out that the decomposition for case (1) satisfies an operator norm bound of $n^{\ell_1/2} \cdot t^{\ell_2/2}$ and the decomposition for case (2) satisfies a bound of $n^{\ell_2/2} \cdot t^{\ell_1/2}$. Together, taking the geometric mean of the two bounds would give the desired bound of $n^{\ell/4} \cdot t^{\ell/4}$.

We remark that our proof doesn't explicitly list out these cases; instead, it defines two different decompositions and simply takes the minimum of the two bounds which essentially captures these two cases. We describe the details of this in Subsection 2.2. For $k > 2$, it turns out that there is a subtle but crucial over-counting issue that is too technical to describe at this point. To address this, we need to introduce new matrices in the decompositions as well as carry out a step similar to Möbius inversion to undo the over-counting. We highlight this issue in Subsection 2.3.

## 2.2 Technical Proof Overview: $k = 2$

Recall from (2) that we wish to upper bound

$$L_{1,\ell}(f) = \sum_{\substack{I,J \in [n]^t \\ |\oplus I \oplus J| = \ell}} \alpha[\oplus I \oplus J] \cdot u[I] M[I, J] v[J]. \tag{3}$$

The high-level idea is as follows. We will express $L_{1,\ell}(f)$ as

$$\sum_{\substack{s_1, s_2 \in \mathbb{N} \\ s_1 + s_2 = \ell}} g(s)$$

for some function $g(s)$, where $s = (s_1, s_2)$ and we shall group the terms based on the sizes $s_1$ and $s_2$ of the sets $\oplus I \setminus \oplus J$ and $\oplus J \setminus \oplus I$ respectively. We shall then upper bound $g(s)$ for any $s_1, s_2 \in \mathbb{N}$ satisfying $s_1 + s_2 = \ell$. To do this, we will express $g(s)$ in two different ways, namely, as $u^\dagger W R' v$ and as $u^\dagger W' R v$, for some matrices $W, W', R, R'$ with bounded operator norms, and we will upper bound these by $\|u\| \|W\| \|R'\| \|v\|$ and $\|u\| \|W'\| \|R\| \|v\|$ respectively. Recall that $\|u\| = \|v\| = 1$. We will show that $\|R\|, \|R'\| \leq 1$ and

$$\|W\| \leq \sqrt{\binom{n}{s_2} \cdot \binom{t}{s_1}} \quad \text{and} \quad \|W'\| \leq \sqrt{\binom{n}{s_1} \cdot \binom{t}{s_2}}.$$

We upper bound the minimum of the two bounds by their geometric mean and use the fact that $s_1 + s_2 = \ell$ to obtain

$$g(s) \leq \sqrt{n^{s_2} t^{s_1} \cdot n^{s_1} t^{s_2}} = n^{\ell/4} t^{\ell/4}$$

as desired. We now describe the function $g(s)$ and the matrices $W, W', R, R'$ in more detail.

We group the terms in (3) based on the sizes of $\oplus I \setminus \oplus J$ and $\oplus J \setminus \oplus I$. For any $(s_1, s_2) \in \mathbb{N} \times \mathbb{N}$, define the indicator function $\mathrm{Size}^s(S_1, S_2)$ for any subsets $S_1, S_2 \subseteq [n]$ by

$$\mathrm{Size}^s(S_1, S_2) = 1\left[|S_1 \setminus S_2| = s_1 \text{ and } |S_2 \setminus S_1| = s_2\right].$$

We will consider $\mathrm{Size}^s(\oplus I, \oplus J)$ as depicted in Figure 2.

Let $g(s)$ denote the contribution to (3) from terms satisfying $\mathrm{Size}^s(\oplus I, \oplus J) = 1$, that is,

$$g(s) := \sum_{I,J \in [n]^t} \mathrm{Size}^s(\oplus I, \oplus J) \cdot \alpha[\oplus I \oplus J] \cdot u[I] M[I, J] v[J].$$

From (3), we have $L_{1,\ell}(f) = \sum_{\substack{s_1, s_2 \in \mathbb{N} \\ s_1 + s_2 = \ell}} g(s)$. Fix any $s_1, s_2 \in \mathbb{N}$ such that $s_1 + s_2 = \ell$. We will now bound $g(s)$. As described before, we will express $g(s)$ in two different ways, namely, as $u^\dagger W R' v$ and

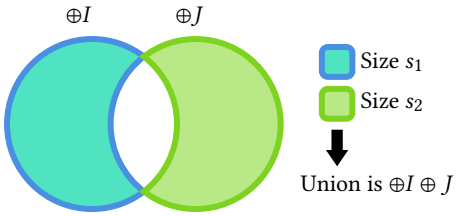


**Figure 2: The constraint** $\text{Size}^s(\oplus I, \oplus J) = 1$.

as $u^\dagger W' R v$, for some matrices $W, W', R, R'$ with bounded operator norms.

*Expressing $g(s)$ as $u^\dagger W R' v$.* The rows and columns of $W$ are indexed by $I$ and $(I', \oplus J)$ respectively, and those of $R'$ by $(I', \oplus J)$ and $J'$ respectively. These matrices are defined as follows

$$W[I, (I', \oplus J)] = 1\left[I = I'\right] \cdot \text{Size}^s(\oplus I, \oplus J) \cdot \alpha[\oplus I \oplus J],$$
$$R'[(I', \oplus J), J'] = 1\left[\oplus J' = \oplus J\right] \cdot M[I', J'].$$

Intuitively, $W$ is a matrix that multiplies by the signs $\alpha[\oplus I \oplus J]$ as well as enforces the $\text{Size}^s$ constraint on $\oplus I$ and $\oplus J$, and $R'$ is a matrix that implements the action of $M$, as well propagates information about $\oplus J$ backwards. This is depicted in Figure 3.

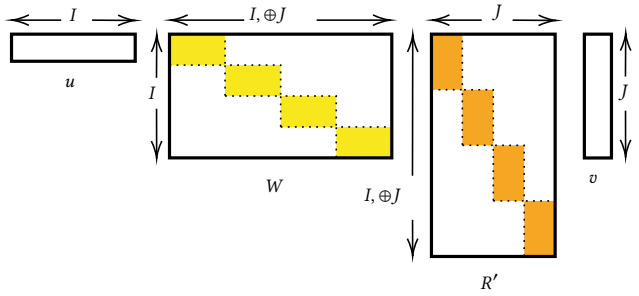


**Figure 3: Expressing** $g(s)$ **as** $u^\dagger W R' v$.

It is not too difficult to see that indeed $g(s) = u^\dagger W R' v$. We now show the desired upper bounds of $\|R'\| \le 1$ and $\|W\| \le \sqrt{\binom{n}{s_2} \cdot \binom{t}{s_1}}$.

- **Bounding $\|R'\|$:** We rearrange the columns of $R'$ according to $\oplus J$. Under this ordering of the columns, observe that $R'$ is a block diagonal matrix, where each block is a submatrix of $M$. Since $\|M\| \le 1$, this implies that $\|R'\| \le 1$.
- **Bounding $\|W\|$:** We rearrange the columns of $W$ according to $I$ and with this ordering, $W$ is block-diagonal. We now use the fact that $\|W\| \le \sqrt{\|W\|_1 \cdot \|W\|_\infty}$ where $\|W\|_1$ and $\|W\|_\infty$ are the max-column-norm and the max-row-norm respectively. Observe that $\|W\|_1 \le 1$, since each column has at most one non-zero entry, which in turn is of unit magnitude. We now bound $\|W\|_\infty$. For any row $I \in [n]^t$, observe that there are at most $\binom{n}{s_2} \cdot \binom{t}{s_1}$ many columns $\oplus J$ such that $\text{Size}^s(\oplus I, \oplus J) \neq 0$. Since each non-zero entry of $W$ is of unit magnitude, this implies that $\|W\|_\infty \le \binom{n}{s_2} \cdot \binom{t}{s_1}$.

This gives us the desired bound of

$$\|W\| \le \sqrt{\binom{n}{s_2} \cdot \binom{t}{s_1}}. \tag{4}$$

*Expressing $g(s)$ as $u^\dagger R W' v$.* The rows and columns of $R$ are indexed by $I$ and $(J', \oplus I')$ respectively and those of $W'$ are indexed by $(J', \oplus I')$ and $J$ respectively, and

$$W'[(J', \oplus I'), J] = 1\left[J = J'\right] \cdot \text{Size}^s(\oplus I', \oplus J) \cdot \alpha[\oplus I' \oplus J],$$
$$R[I, (J', \oplus I')] = 1\left[\oplus I = \oplus I'\right] \cdot M[I, J'].$$

Here, $W'$ implements $\alpha[\oplus I \oplus J]$ as well as enforces the $\text{Size}^s$ constraint on $\oplus I$ and $\oplus J$, and $R$ implements the action of $M$, as well propagates information about $\oplus I$ forward. This is depicted in Figure 4. A calculation similar to the previous case implies the desired bound of

$$\|W'\| \le \sqrt{\binom{n}{s_1} \cdot \binom{t}{s_2}}. \tag{5}$$

This completes the proof overview for $k = 2$.

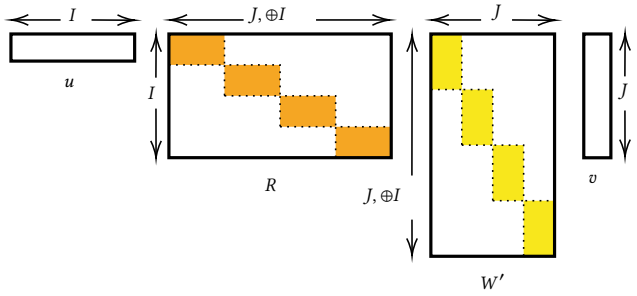


**Figure 4: Expressing** $g(s)$ **as** $u^\dagger R W' v$.

## 2.3 Technical Proof Overview: $k = 3$

For simplicity of notation, we will switch from indices $I_1, I_2, I_3$ to indices $I, J, K$. We need to upper bound

$$L_{1,\ell}(f) = \sum_{\substack{I,J,K \in [n]^t \\ |\oplus I \oplus J \oplus K| = \ell}} \alpha[\oplus I \oplus J \oplus K] \cdot u[I] M_1[I, J] M_2[J, K] v[K].$$

As before, we will express $L_{1,\ell}(f)$ as $\sum_{\substack{s_1,\ldots,s_4 \in \mathbb{N} \\ s_1+\cdots+s_4=\ell}} g(s)$ grouping terms based on sizes of certain sets and in order to bound each $g(s)$, we will try to express it in three different ways as $u^\dagger W_1 R_1' R_2' v$, $u^\dagger R_1 W_2 R_2' v$ and $u^\dagger R_1 R_2 W_3 v$.

It will turn out that $\|R_1\|, \|R_1'\|, \|R_2\|, \|R_2'\| \le 1$ and that

$$\|W_1\| \le \left(\frac{n}{t}\right)^{\frac{s_2+s_3}{2}} t^\ell, \|W_2\| \le \left(\frac{n}{t}\right)^{\frac{s_1+s_3}{2}} t^\ell, \|W_3\| \le \left(\frac{n}{t}\right)^{\frac{s_1+s_2}{2}} t^\ell. \tag{6}$$

Since $s_1 + s_2 + s_3 \le \ell$, taking the minimum of the three bounds would give us the desired bound of $(n/t)^{\ell/3} \cdot t^\ell$. There is an issue that comes up that we will later highlight. To describe it now in a nutshell, it turns out we *cannot* express $g(s)$ in the form of a matrix product with operator norms bounded as desired. Nevertheless, with some additional work, we can express a different function $h(s)$ in this form, furthermore, $h(s) = \sum_{s'} P[s, s'] g(s')$ for some

invertible matrix $P$ such that $P^{-1}$ has bounded norms. Therefore, using bounds on $h(s)$, we can derive the desired bounds on $g(s)$. We describe all this in more detail.

We start with the description of $g(s)$. Similar to the previous case, we will fix the sizes of certain sets in the Venn diagram of $\oplus I, \oplus J, \oplus K$ as depicted in Figure 5. More formally, let $s \in \mathbb{N}^4$. Define

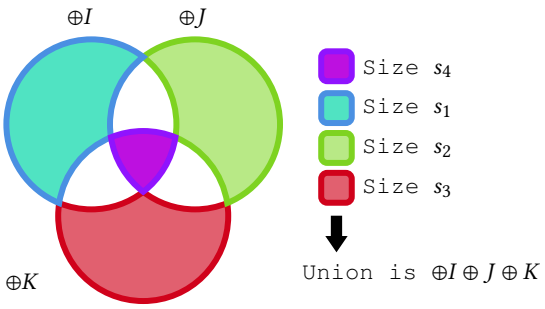


**Figure 5: The constraint** $\mathrm{Size}^s(\oplus I, \oplus J) = 1$.

$\mathrm{Size}^s(S_1, S_2, S_3)$ to be the indicator function of

$$|S_1 \setminus (S_2 \cup S_3)| = s_1, \quad |S_2 \setminus (S_1 \cup S_3)| = s_2,$$
$$|S_3 \setminus (S_1 \cup S_2)| = s_3, \quad |S_1 \cap S_2 \cap S_3| = s_4.$$

Let

$$g(s) := \sum_{\substack{I,J,K \in [n]^t \\ |\oplus I \oplus J \oplus K| = \ell}} \mathrm{Size}^s(\oplus I, \oplus J, \oplus K) \cdot \alpha[\oplus I \oplus J \oplus K] \cdot u[I] M_1[I,J] M_2[J,K] v[J].$$

We attempt to express $g(s)$ in three different ways as $u^\dagger W_1 R_1' R_2' v$, $u^\dagger R_1 W_2 R_2' v$, and $u^\dagger R_1 R_2 W_3 v$. The simplest to describe is the second expression. Here, we have matrices $R_1, W_2, R_2'$ whose indices are as depicted in Figure 6.

Based on the intuition from before, there is a very natural way to define these matrices, namely,

$$R_1[I, (\oplus I', J)] = 1\left[\oplus I = \oplus I'\right] \cdot M_1[I,J],$$
$$R_2'[(J, \oplus K), K'] = 1\left[\oplus K' = \oplus K\right] \cdot M_2[J,K'],$$

and

$$W_2[(J, \oplus I), (J', \oplus K)] = 1\left[J = J'\right] \cdot \mathrm{Size}^s(\oplus I, \oplus J, \oplus K) \cdot \alpha[\oplus I \oplus J \oplus K].$$

Note here that given the row $(J, \oplus I)$ and the column $(J, \oplus K)$, we can compute $\mathrm{Size}^s(\oplus I, \oplus J, \oplus K)$ and $\alpha[\oplus I \oplus J \oplus K]$. A similar calculation to before shows that $\|R_1\|, \|R_2'\| \le 1$ and

$$\|W_2\| \le \sqrt{\binom{n}{s_3} \cdot \binom{t}{s_1}\binom{t}{s_2}\binom{t}{s_4}} \cdot \sqrt{\binom{n}{s_1} \cdot \binom{t}{s_2}\binom{t}{s_3}\binom{t}{s_4}} = (n/t)^{(s_1+s_3)/2} \cdot t^\ell.$$

Let us try to define the other two decompositions $u^\dagger W_1 R_1' R_2' v$ and $u^\dagger R_1 R_2 W_2 v$ as depicted in Figure 7. Suppose we could define $W_1$ and $W_2$ such that

$$W_1[I, (I', \oplus J \oplus K)] = 1\left[I = I'\right] \mathrm{Size}^s(\oplus I, \oplus J, \oplus K) \alpha[\oplus I \oplus J \oplus K],$$
$$W_3[K, (K', \oplus I \oplus K)] = 1\left[K = K'\right] \mathrm{Size}^s(\oplus I, \oplus J, \oplus K) \alpha[\oplus I \oplus J \oplus K]. \quad (7)$$

Then, a calculation similar to the previous case would give the desired operator norm bounds on $W_1$ and $W_3$ as in (6). The problem is that we cannot define matrices $W_1, W_3$ that satisfy (7). We explain this issue for $W_1$. Given a row $I$ and a column $(I, \oplus J \oplus K)$, we cannot compute $\mathrm{Size}^s(\oplus I, \oplus J, \oplus K)$. After all, we only have the information about $\oplus I$ and $\oplus J \oplus K$, and hence the matrix $W_1$ can only enforce the constraints that $|\oplus I \setminus (\oplus J \oplus K)| = s_1 + s_4$ and $|(\oplus J \oplus K) \setminus \oplus I| = s_2 + s_3$, but it cannot enforce the constraints that $|\oplus J \setminus (\oplus I \cup \oplus K)| = s_2$ or $|\oplus K \setminus (\oplus I \cup \oplus J)| = s_3$. In particular, if we only define $W_1$ to enforce the constraints that it is able to enforce, we will end up counting terms corresponding to $I', J', K'$ which satisfy $\mathrm{Size}^{s'}(\oplus I, \oplus J, \oplus K)$ for $s'$ with $s_2' \ne s_2$ and $s_3' \ne s_3$. In this case, instead of estimating the target $g(s)$, we would be over-counting. We need two new ideas here.

(1) First we need to provide $W_1$ some additional information. One might hope that with a little extra information, $W_1$ can enforce $\mathrm{Size}^s$, but this turns out to be false. Giving this information will increase the operator norms by too much. Instead, the idea is to provide some information that enforces a variant of the $\mathrm{Size}^s$ constraint.
(2) This variant will allow us to bound a different function $h(s)$. This function is still an over-counting of $g(s)$, but the important point is that it is a predictable over-counting, that is, $h(s) = \sum_{s'} P[s,s'] g(s')$ for some invertible matrix $P$ such that $P^{-1}$ has bounded norm. Hence, we can derive bounds on $g(s)$ using bounds on $h(s)$.

We first explain step (2). Let

$$L(I,J,K) = \alpha[\oplus I \oplus J \oplus K] \cdot u[I] M_1[I,J] M_2[J,K] v[K].$$

While we would like to bound the expression

$$g(s) := \sum_{I,J,K \in [n]^t} L(I,J,K) \cdot \mathrm{Size}^s(I,J,K),$$

what we can bound turns out to be the expression

$$h(s) := \sum_{I,J,K \in [n]^t} L(I,J,K) \sum_{\substack{A,B,C,D \in [n]^t \\ A,B,C,D \text{ are disjoint} \\ A \cup B \cup C \cup D = \oplus I \oplus J \oplus K}} \mathrm{Subset}^s(A,B,C,D),$$

where $\mathrm{Subset}^s(A,B,C,D)$ is the indicator of the constraint that

$$A \subseteq \oplus I, |A| = s_1, \quad B \subseteq \oplus J, |B| = s_2, \quad C \subseteq \oplus K, |C| = s_3,$$
$$D \subseteq \oplus I \cap \oplus J \cap \oplus K, |D| = s_4. \quad (8)$$

This is depicted in Figure 8.

Observe that one of the terms in $h(s)$ is $A = \oplus I \setminus (\oplus J \cup \oplus K)$, $B = \oplus J \setminus (\oplus I \cup \oplus K)$, $C = \oplus K \setminus (\oplus I \cup \oplus J)$ and $D = \oplus I \cap \oplus J \cap \oplus K$. Hence, $h(s)$ consists of $g(s)$ plus some additional terms. For example, elements from $D$ can be moved to either $A$, $B$, or $C$ and still satisfy the constraints in (8). However, we can express

$$h(s) = \sum_{s'} P[s,s'] g(s')$$

for a structured matrix $P$. This matrix is invertible and has bounded $\|P^{-1}\|_1$. Therefore, our goal of bounding $\|g\|_1$ reduces to bounding $\|h\|_1$ as $h = Pg$. This is done in step (1) which we now explain.

We now explain how to bound $h(s)$. We will blow up the matrices in the decomposition $u^\dagger W_1 R_1' R_2' v$ to include information about $A, B, C, D \subseteq [n]$. We will also introduce new matrices $Q_1, Q_1', Q_2', Q_3'$

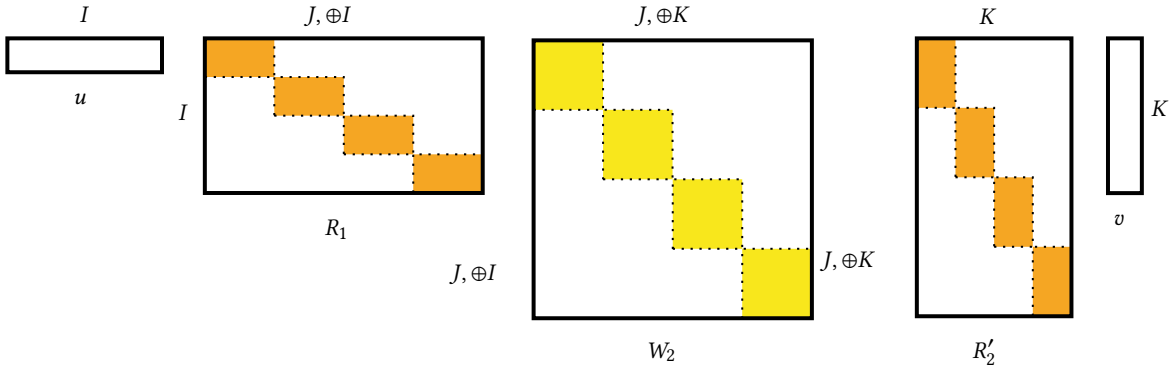


**Figure 6: Expressing $g(s)$ as $u^\dagger R_1 W_2 R_2' v$.**

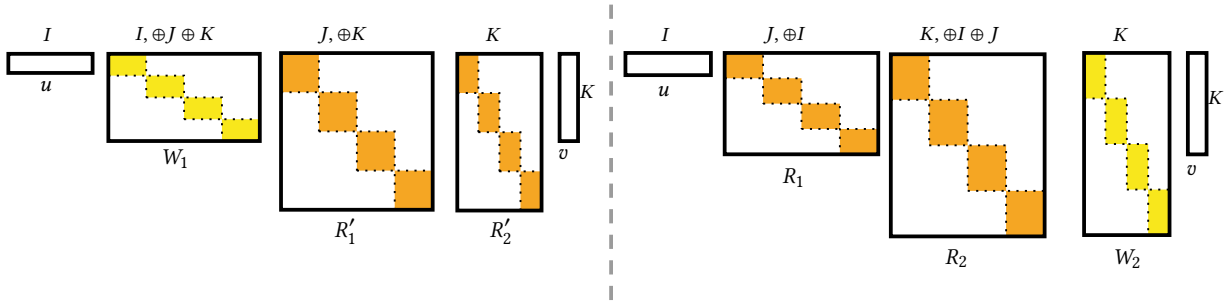


**Figure 7: Expressing $g(s)$ as $u^\dagger W_1 R_1' R_2' v$ and $u^\dagger R_1 R_2 W_2 v$ respectively.**

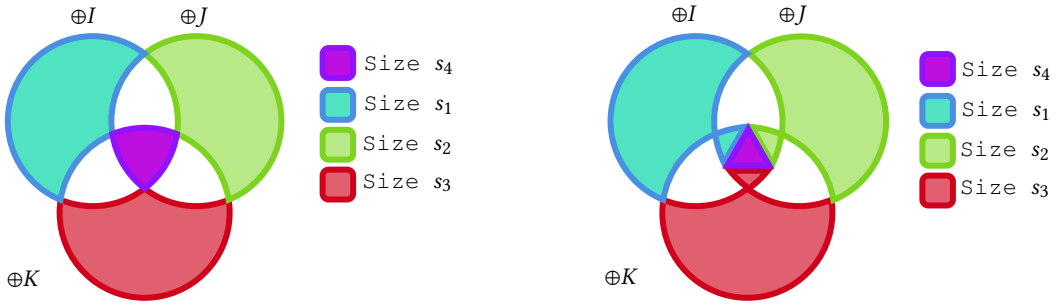


Figure (a) What we want to sum over  Figure (b) What we end up summing over

**Figure 8: The summation in $g(s)$ versus $h(s)$.**

to enumerate $A, B, C, D$ and verify that they satisfy the Subset$^s$ constraints in (8). Consider the expression $u^\dagger Q_1 W_1 Q_1' R_1' Q_2' R_2' Q_3' v$, where the matrices are as depicted in Figure 9.

The matrices $W_1, R_2', R_3'$ perform the same role as before and in addition, propagate information about the sets $A, B, C, D$. The matrices $Q_1, Q_1', Q_2', Q_3'$ impose constraints on $A, B, C, D$ as well as add and delete information as required. In more detail,

(1) $Q_1$ propagates $I$ and introduces $A, D$ such that that $A, D \subseteq \oplus I$, $|A| = s_1$ and $|D| = s_4$. Given $I$, there are at most $\binom{t}{s_1} \cdot \binom{t}{s_4}$ possibilities for $(A, D)$ and it follows that $\|Q_1\| \leq \sqrt{t^{s_1} \cdot t^{s_4}}$.

(2) $W_1$ enforces $A \cup B \cup C \cup D = \oplus I \oplus J \oplus K$ and the size constraints on $B, C$. It also applies $\alpha[\oplus I \oplus J \oplus K]$. For each $I, A, D$, there are at most $\binom{n}{s_2} \cdot \binom{n}{s_3}$ possibilities for $(B, C)$ and once we fix $A, B, C, D$ and $I$, we also fix $\oplus J \oplus K = \oplus I \oplus (A \cup B \cup C \cup D)$. So $\|W_1\| \leq \sqrt{n^{s_2} \cdot n^{s_3}}$.

(3) $Q_3'$ back-propagates $K$ and introduces $C, D$ such that $C, D \subseteq \oplus K$, $|C| = s_3$, and $|D| = s_4$. Given $K$, there are at most $\binom{t}{s_3} \cdot \binom{t}{s_4}$ possibilities for $(C, D)$ and hence $\|Q_3'\| \leq \sqrt{t^{s_3} \cdot t^{s_4}}$.

(4) $R_3'$ back-propagates $\oplus K, C, D$ and introduces $J$. It also applies the operator $M_2$. As before, $\|R_3'\| \leq 1$.

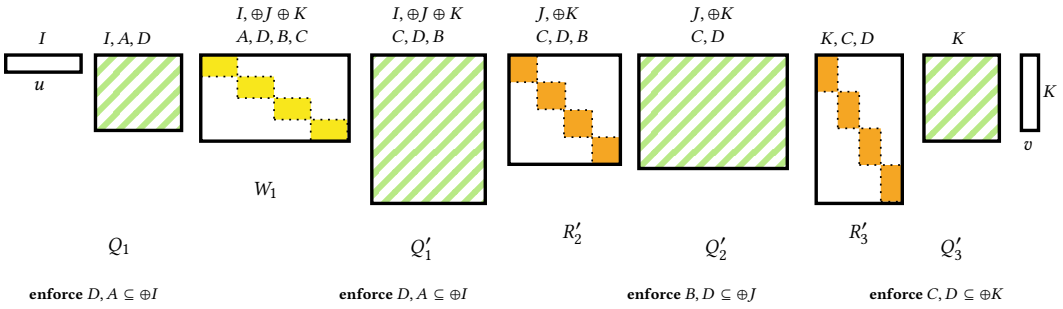


**Figure 9: Expressing $h(s) = u^\dagger Q_1 W_1 Q_1' R_1' Q_2' R_2' Q_3' v$.**

(5) $Q_2'$ back-propagates $J, \oplus K, C, D$, introduces $B$, and enforces that $B, D \subseteq \oplus J$ and $|B| = s_2$. Given $J$, there are at most $\binom{t}{s_2}$ possibilities for $B$, hence, $\|Q_2'\| \le \sqrt{t^{s_2}}$.
(6) $R_2'$ back-propagates $D, B, C, \oplus J \oplus K$, introduces $I$, and applies the operator $M_1$. As before, $\|R_2'\| \le 1$.
(7) $Q_1'$ back-propagates $D, B, C, \oplus J \oplus K, I$, introduces $A$, and enforces $D, A \subseteq \oplus I$ and $|A| = s_1$. Given $I$, there at most $\binom{t}{s_1}$ possibilities $A$, hence $\|Q_1'\| \le \sqrt{t^{s_1}}$.

Combining all these bounds gives us an upper bound on $h(s)$ of

$$\sqrt{n^{s_2+s_3}} \cdot t^{s_4+s_1+(s_2+s_3)/2} = (n/t)^{(s_2+s_3)/2} \cdot t^\ell.$$

By a symmetric argument, we blow up the matrices in the decomposition $u^\dagger R_1 R_2 W_3 v$ to include information about $A, B, C, D \subseteq [n]$ and get

$$h(s) \le (n/t)^{(s_1+s_2)/2} \cdot t^\ell.$$

Combining the three upper bounds on $h(s)$ we get

$$h(s) \le (n/t)^{\ell/3} \cdot t^\ell.$$

## ACKNOWLEDGMENTS

We thank anonymous QIP'24 and STOC'24 reviewers for helpful comments. KW also wants to thank Guangxu Yang and Penghui Yao for references in the quantum communication complexity. Part of this work was done while UG and MS are at the Simons Institute for the Theory of Computing.

UG is supported by the Simons Collaboration on Algorithms and Geometry, a Simons Investigator Award, by the National Science Foundation grants No. CCF-1714779, CCF-2007462 and by the IBM PhD Fellowship. MS is supported by a Simons-Berkeley Postdoctoral Fellowship. AT and KW are supported by a Sloan Research Fellowship and NSF CAREER Award CCF-2145474.

## REFERENCES

[1] Scott Aaronson and Andris Ambainis. 2015. Forrelation: A Problem that Optimally Separates Quantum from Classical Computing. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, Rocco A. Servedio and Ronitt Rubinfeld (Eds.). ACM, USA, 307–316. https://doi.org/10.1145/2746539.2746547
[2] Rohit Agrawal. 2020. Coin Theorems and the Fourier Expansion. *Chic. J. Theor. Comput. Sci.* 2020 (2020). http://cjtcs.cs.uchicago.edu/articles/2020/4/contents.html
[3] Andris Ambainis and Ronald de Wolf. 2014. How Low can Approximate Degree and Quantum Query Complexity be for Total Boolean Functions? *Comput. Complex.* 23, 2 (2014), 305–322. https://doi.org/10.1007/S00037-014-0083-2
[4] 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 2023, Orlando, FL, USA, June 20-23, 2023*, Barna Saha and Rocco A. Servedio (Eds.). ACM, USA, 1111–1124. https://doi.org/10.1145/3564246.3585153
[5] Nikhil Bansal and Makrand Sinha. 2021. k-forrelation optimally separates Quantum and classical query complexity. In *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, Samir Khuller and Virginia Vassilevska Williams (Eds.). ACM, USA, 1303–1316. https://doi.org/10.1145/3406325.3451040
[6] Ethan Bernstein and Umesh Vazirani. 1997. Quantum Complexity Theory. *SIAM J. Comput.* 26, 5 (1997), 1411–1473. https://doi.org/10.1137/S0097539796300921
[7] Sergey Bravyi, David Gosset, Daniel Grier, and Luke Schaeffer. 2022. Classical algorithms for Forrelation. *CoRR* (2022). arXiv:2102.06963 https://arxiv.org/abs/2102.06963
[8] Paul Burchard. 2019. Lower Bounds for Parallel Quantum Counting. *CoRR* abs/1910.04555 (2019). arXiv:1910.04555 http://arxiv.org/abs/1910.04555
[9] Eshan Chattopadhyay, Jason Gaitonde, Chin Ho Lee, Shachar Lovett, and Abhishek Shetty. 2021. Fractional Pseudorandom Generators from Any Fourier Level. In *36th Computational Complexity Conference, CCC 2021, July 20-23, 2021, Toronto, Ontario, Canada (Virtual Conference) (LIPIcs, Vol. 200)*, Valentine Kabanets (Ed.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, USA, 10:1–10:24. https://doi.org/10.4230/LIPICS.CCC.2021.10
[10] Eshan Chattopadhyay, Pooya Hatami, Kaave Hosseini, and Shachar Lovett. 2019. Pseudorandom Generators from Polarizing Random Walks. *Theory Comput.* 15 (2019), 1–26. https://doi.org/10.4086/TOC.2019.V015A010
[11] Eshan Chattopadhyay, Pooya Hatami, Shachar Lovett, and Avishay Tal. 2019. Pseudorandom Generators from the Second Fourier Level and Applications to AC0 with Parity Gates. In *10th Innovations in Theoretical Computer Science Conference, ITCS 2019, January 10-12, 2019, San Diego, California, USA (LIPIcs, Vol. 124)*, Avrim Blum (Ed.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, USA, 22:1–22:15. https://doi.org/10.4230/LIPICS.ITCS.2019.22
[12] Nai-Hui Chia and Shih-Han Hung. 2023. Non-Interactive Classical Verification of Quantum Depth: A Fine-Grained Characterization. *IACR Cryptol. ePrint Arch.* (2023), 1911. https://eprint.iacr.org/2023/1911
[13] Richard Cleve and John Watrous. 2000. Fast parallel circuits for the quantum Fourier transform. In *41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12-14 November 2000, Redondo Beach, California, USA*. IEEE Computer Society, USA, 526–536. https://doi.org/10.1109/SFCS.2000.892140
[14] Matthew Coudron and Sanketh Menda. 2020. Computations with greater quantum depth are strictly more powerful (relative to an oracle). In *Proccedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing, STOC 2020, Chicago, IL, USA, June 22-26, 2020*, Konstantin Makarychev, Yury Makarychev, Madhur Tulsiani, Gautam Kamath, and Julia Chuzhoy (Eds.). ACM, USA, 889–901. https://doi.org/10.1145/3357713.3384269
[15] David Deutsch and Richard Jozsa. 1992. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences* 439, 1907 (1992), 553–558. https://doi.org/10.1098/rspa.1992.0167
[16] Alexandros Eskenazis and Paata Ivanisvili. 2022. Learning low-degree functions from a logarithmic number of random queries. In *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, Stefano Leonardi and Anupam Gupta (Eds.). ACM, USA, 203–207. https://doi.org/10.1145/3519935.3519981
[17] Uma Girish, Ran Raz, and Avishay Tal. 2022. Quantum versus Randomized Communication Complexity, with Efficient Players. *Comput. Complex.* 31, 2 (2022), 17. https://doi.org/10.1007/S00037-022-00232-7
[18] Uma Girish, Makrand Sinha, Avishay Tal, and Kewen Wu. 2023. Fourier Growth of Communication Protocols for XOR Functions. In *64th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2023, Santa Cruz, CA, USA, November 6-9, 2023*. IEEE, USA, 721–732. https://doi.org/10.1109/FOCS57990.2023.00047

[19] Uma Girish, Makrand Sinha, Avishay Tal, and Kewen Wu. 2023. The Power of Adaptivity in Quantum Query Algorithms. *CoRR* abs/2311.16057 (2023). https://doi.org/10.48550/ARXIV.2311.16057 arXiv:2311.16057

[20] Uma Girish, Avishay Tal, and Kewen Wu. 2021. Fourier Growth of Parity Decision Trees. In *36th Computational Complexity Conference, CCC 2021, July 20-23, 2021, Toronto, Ontario, Canada (Virtual Conference) (LIPIcs, Vol. 200)*, Valentine Kabanets (Ed.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, USA, 39:1–39:36. https://doi.org/10.4230/LIPICS.CCC.2021.39

[21] Lov K. Grover. 1996. A Fast Quantum Mechanical Algorithm for Database Search. In *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, Gary L. Miller (Ed.). ACM, USA, 212–219. https://doi.org/10.1145/237814.237866

[22] Atsuya Hasegawa and François Le Gall. 2022. An Optimal Oracle Separation of Classical and Quantum Hybrid Schemes. In *33rd International Symposium on Algorithms and Computation, ISAAC 2022, December 19-21, 2022, Seoul, Korea (LIPIcs, Vol. 248)*, Sang Won Bae and Heejin Park (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, USA, 6:1–6:14. https://doi.org/10.4230/LIPICS.ISAAC.2022.6

[23] Siddharth Iyer, Anup Rao, Victor Reis, Thomas Rothvoss, and Amir Yehudayoff. 2021. Tight bounds on the Fourier growth of bounded functions on the hypercube. *CoRR* abs/2107.06309 (2021). arXiv:2107.06309 https://arxiv.org/abs/2107.06309

[24] Rahul Jain, Jaikumar Radhakrishnan, and Pranab Sen. 2002. The Quantum Communication Complexity of the Pointer Chasing Problem: The Bit Version. In *FST TCS 2002: Foundations of Software Technology and Theoretical Computer Science, 22nd Conference Kanpur, India, December 12-14, 2002, Proceedings (Lecture Notes in Computer Science, Vol. 2556)*, Manindra Agrawal and Anil Seth (Eds.). Springer, USA, 218–229. https://doi.org/10.1007/3-540-36206-1_20

[25] Stacey Jeffery, Frédéric Magniez, and Ronald de Wolf. 2017. Optimal Parallel Quantum Query Algorithms. *Algorithmica* 79, 2 (2017), 509–529. https://doi.org/10.1007/S00453-016-0206-Z

[26] Hartmut Klauck, Ashwin Nayak, Amnon Ta-Shma, and David Zuckerman. 2001. Interaction in quantum communication and the complexity of set disjointness. In *Proceedings on 33rd Annual ACM Symposium on Theory of Computing, July 6-8, 2001, Heraklion, Crete, Greece*, Jeffrey Scott Vitter, Paul G. Spirakis, and Mihalis Yannakakis (Eds.). ACM, USA, 124–133. https://doi.org/10.1145/380752.380786

[27] Pascal Koiran, Jürgen Landes, Natacha Portier, and Penghui Yao. 2010. Adversary lower bounds for nonadaptive quantum algorithms. *J. Comput. Syst. Sci.* 76, 5 (2010), 347–355. https://doi.org/10.1016/J.JCSS.2009.10.007

[28] William Kretschmer, Luowen Qian, Makrand Sinha, and Avishay Tal. 2023. Quantum Cryptography in Algorithmica. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing, STOC 2023, Orlando, FL, USA, June 20-23, 2023*, Barna Saha and Rocco A. Servedio (Eds.). ACM, USA, 1589–1602. https://doi.org/10.1145/3564246.3585225

[29] Chin Ho Lee, Edward Pyne, and Salil P. Vadhan. 2022. Fourier Growth of Regular Branching Programs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2022, September 19-21, 2022, University of Illinois, Urbana-Champaign, USA (Virtual Conference) (LIPIcs, Vol. 245)*, Amit Chakrabarti and Chaitanya Swamy (Eds.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, USA, 2:1–2:21. https://doi.org/10.4230/LIPICS.APPROX/RANDOM.2022.2

[30] Yishay Mansour. 1992. An $O(n^{\log\log n})$ Learning Algorithm for DNF Under the Uniform Distribution. (1992), 53–61. https://doi.org/10.1145/130385.130391

[31] Ashley Montanaro. 2010. Nonadaptive quantum query complexity. *Inf. Process. Lett.* 110, 24 (2010), 1110–1113. https://doi.org/10.1016/J.IPL.2010.09.009

[32] Cristopher Moore and Martin Nilsson. 2001. Parallel Quantum Computation and Quantum Codes. *SIAM J. Comput.* 31, 3 (2001), 799–815. https://doi.org/10.1137/S0097539799355053

[33] Harumichi Nishimura and Tomoyuki Yamakami. 2004. An Algorithmic Argument for Nonadaptive Query Complexity Lower Bounds on Advised Quantum Computation (Extended Abstract). In *Mathematical Foundations of Computer Science 2004, 29th International Symposium, MFCS 2004, Prague, Czech Republic, August 22-27, 2004, Proceedings (Lecture Notes in Computer Science, Vol. 3153)*, Jirí Fiala, Václav Koubek, and Jan Kratochvíl (Eds.). Springer, USA, 827–838. https://doi.org/10.1007/978-3-540-28629-5_65

[34] Ran Raz and Avishay Tal. 2022. Oracle Separation of BQP and PH. *J. ACM* 69, 4 (2022), 30:1–30:21. https://doi.org/10.1145/3530258

[35] Oded Regev. 2023. An Efficient Quantum Factoring Algorithm. *CoRR* abs/2308.06572 (2023). https://doi.org/10.48550/ARXIV.2308.06572 arXiv:2308.06572

[36] Omer Reingold, Thomas Steinke, and Salil P. Vadhan. 2013. Pseudorandomness for Regular Branching Programs via Fourier Analysis. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 16th International Workshop, APPROX 2013, and 17th International Workshop, RANDOM 2013, Berkeley, CA, USA, August 21-23, 2013. Proceedings (Lecture Notes in Computer Science, Vol. 8096)*, Prasad Raghavendra, Sofya Raskhodnikova, Klaus Jansen, and José D. P. Rolim (Eds.). Springer, USA, 655–670. https://doi.org/10.1007/978-3-642-40328-6_45

[37] Alexander A. Sherstov, Andrey A. Storozhenko, and Pei Wu. 2023. An Optimal Separation of Randomized and Quantum Query Complexity. *SIAM J. Comput.* 52, 2 (2023), 525–567. https://doi.org/10.1137/22M1468943

[38] Peter W. Shor. 1997. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM J. Comput.* 26, 5 (1997), 1484–1509. https://doi.org/10.1137/S0097539795293172

[39] Peter W. Shor. 1999. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM Rev.* 41, 2 (1999), 303–332. https://doi.org/10.1137/S0036144598347011

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

[41] Hans Ulrich Simon. 1983. A Tight Omega(loglog n)-Bound on the Time for Parallel Ram's to Compute Nondegenerated Boolean Functions. In *Fundamentals of Computation Theory, Proceedings of the 1983 International FCT-Conference, Borgholm, Sweden, August 21-27, 1983 (Lecture Notes in Computer Science, Vol. 158)*, Marek Karpinski (Ed.). Springer, USA, 439–444. https://doi.org/10.1007/3-540-12689-9_124

[42] Thomas Steinke, Salil P. Vadhan, and Andrew Wan. 2017. Pseudorandomness and Fourier-Growth Bounds for Width-3 Branching Programs. *Theory Comput.* 13, 1 (2017), 1–50. https://doi.org/10.4086/TOC.2017.V013A012

[43] Avishay Tal. 2017. Tight Bounds on the Fourier Spectrum of AC0. In *32nd Computational Complexity Conference, CCC 2017, July 6-9, 2017, Riga, Latvia (LIPIcs, Vol. 79)*, Ryan O'Donnell (Ed.). Schloss Dagstuhl - Leibniz-Zentrum für Informatik, USA, 15:1–15:31. https://doi.org/10.4230/LIPICS.CCC.2017.15

[44] Avishay Tal. 2020. Towards Optimal Separations between Quantum and Randomized Query Complexities. In *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, Sandy Irani (Ed.). IEEE, USA, 228–239. https://doi.org/10.1109/FOCS46700.2020.00030

[45] Wim van Dam. 1998. Quantum Oracle Interrogation: Getting All Information for Almost Half the Price. In *39th Annual Symposium on Foundations of Computer Science, FOCS '98, November 8-11, 1998, Palo Alto, California, USA*. IEEE Computer Society, USA, 362–367. https://doi.org/10.1109/SFCS.1998.743486

[46] John Watrous. 2009. Quantum Computational Complexity. In *Encyclopedia of Complexity and Systems Science*, Robert A. Meyers (Ed.). Springer, USA, 7174–7201. https://doi.org/10.1007/978-0-387-30440-3_428

[47] Christof Zalka. 1999. Grover's quantum searching algorithm is optimal. *Phys. Rev. A* 60 (Oct 1999), 2746–2751. Issue 4. https://doi.org/10.1103/PhysRevA.60.2746


Received 10-NOV-2023; accepted 2024-02-11