

Fourier Spectrum of Noisy Quantum Algorithms

Uma Girish*

Abstract

Quantum computing promises exponential speedups for certain problems, yet fully universal quantum computers remain out of reach and near-term devices are inherently noisy. Motivated by this, we study noisy quantum algorithms and the landscape between BQP and BPP. We build on a powerful technique to differentiate quantum and classical algorithms called the level- ℓ Fourier growth (the sum of absolute values of Fourier coefficients of sets of size ℓ) and show that it can also be used to differentiate quantum algorithms based on the types of resources used. We show that noise acting on a quantum algorithm dampens its Fourier growth in ways intricately linked to the type of noise.

Concretely, we study noisy models of quantum computation where highly mixed states are prevalent, namely: DQC_k algorithms, where k qubits are clean and the rest are maximally mixed, and $\frac{1}{2}\text{BQP}$ algorithms, where the initial state is maximally mixed, but the algorithm is given knowledge of the initial state at the end of the computation. We establish upper bounds on the Fourier growth of DQC_k , $\frac{1}{2}\text{BQP}$ and BQP algorithms and leverage the differences between these bounds to derive oracle separations between these models. In particular, we show that 2-FORRELATION and 3-FORRELATION require $N^{\Omega(1)}$ queries in the DQC_1 and $\frac{1}{2}\text{BQP}$ models respectively. Our results are proved using a new matrix decomposition lemma that might be of independent interest.

Contents

1	Introduction	3
1.1	The Space Below BQP	3
1.2	Fourier Growth	5
1.3	Our Results	7
1.4	Applications	9
1.5	Technical Highlight: Matrix Decomposition Lemma	10
1.6	Proof Sketch	11
1.7	Outlook & Future Directions	14
1.8	Organization.	15
2	Preliminaries & Notation	15
2.1	Fourier Growth	16
2.2	Quantum Query Complexity	17
2.3	DQC_k algorithms	18
2.4	$\frac{1}{2}\text{BQP}$ algorithms	19
3	Main Technical Tool: Matrix Decomposition Lemma	20
3.1	An Improved Matrix Decomposition Lemma	22

*Columbia University. Email: ug2150@columbia.edu

4	Fourier Growth of DQC_k: Proof of Theorem 1.5	22
4.1	Level- ℓ Fourier Growth	23
4.2	Tightness of our Bounds for DQC_1	26
5	Fourier Growth of $\frac{1}{2}\text{BQP}$: Proof of Theorem 1.6	27
5.1	Level-3 Fourier Growth	27
5.2	Level-6 Fourier Growth	30
6	Fourier Growth of BQP: Proof of Theorem 1.7	33
6.1	Level- ℓ Fourier Growth	34
7	Acknowledgements	36
A	Appendix	39
A.1	Quantum Algorithms with Classical Pre-Processing.	39
A.2	Simulating DQC_k algorithms by DQC_{k-t} algorithms.	42
A.3	Proof of Improved Matrix Decomposition Lemma	42
A.4	Acceptance Probability of Quantum Algorithms	46
A.5	Fourier Coefficients of Quantum Algorithms	48

1 Introduction

Quantum computing promises to solve certain problems exponentially faster than classical computers, as evidenced by numerous query complexity separations or oracle separations [DJ92, BV97, Sim97, Aar10, AA15]. Yet, we haven't been able to harness this, as we are far from being able to build fully universal quantum computers. While BQP algorithms generally assume noiseless computation, noise is arguably the most significant issue faced by near-term quantum computers and all current quantum devices are inherently noisy. To better understand what quantum resources are truly responsible for quantum advantage, researchers have proposed numerous intermediate models of quantum computing like IQP, DQC_1 , NISQ and Boson Sampling [SB08, KL98, CCHL23, AA11, ABKM17]. These models isolate specific quantum features – such as having a few clean qubits or limited adaptivity – and allow us to probe the quantum landscape below BQP. Although these models likely do not capture the full power of quantum computing, their precise relationship to BQP and to each other remains poorly understood. This raises a natural question:

What does the landscape of quantum computation below BQP look like?

In our work, we study this question from a Fourier analytic perspective. In particular, we study the level- ℓ Fourier growth of the acceptance probability of algorithms (Definition 1.2). This is a measure of how well-spread the Fourier coefficients are. In our work, we show that Fourier growth is not just a tool for distinguishing quantum and classical models; it is a fine-grained tool capable of differentiating quantum models based on the kinds of quantum resources they utilize. We focus on noisy quantum algorithms and demonstrate that noise dampens the Fourier growth in ways that are intricately linked to the type of noise present.

In particular, we study noisy models like DQC_k , where k qubits are clean and the rest are maximally mixed [KL98, MFF14], and $\frac{1}{2}\text{BQP}$, where the initial state is maximally mixed, i.e., a uniformly random computational basis state, but the algorithm is given knowledge of this initial state at the end of the computation [ABKM17, JM24]. We prove Fourier growth bounds on the acceptance probability of DQC_k , $\frac{1}{2}\text{BQP}$ and BQP algorithms (Theorems 1.5 to 1.7) and use the differences in these bounds to derive oracle separations between these models. In particular, we show that 2-FORRELATION and 3-FORRELATION, which can be solved with two queries in the $\frac{1}{2}\text{BQP}$ and BQP models respectively, require $N^{\Omega(1)}$ queries in the DQC_1 and $\frac{1}{2}\text{BQP}$ models respectively (Corollaries 1.9 and 1.10), resolving two conjectures from [JM24] and establishing the first oracle separation between $\frac{1}{2}\text{BQP}$ and DQC_k , as well as a new oracle separation between BQP and $\frac{1}{2}\text{BQP}$.

We believe that the noise-induced dampening of Fourier growth is a more general phenomenon, and that the techniques developed here could shed light on other noisy models such as NISQ. Our results are proved using a new matrix decomposition lemma that encodes information about indices in a matrix product that might be of independent interest.

1.1 The Space Below BQP

The landscape of computational models between BQP and BPP is vast and intricate. There are numerous intermediate models of quantum computation like IQP, DQC_1 , NISQ and Boson Sampling [SB08, KL98, CCHL23, AA11] with constraints on the quantum resources. The study of such intermediate models serves two key purposes: (1) to systematically delineate the boundary between classical and quantum algorithms and pinpoint the minimal resources for quantum speedups, and (2) to model the physical constraints of near-term quantum devices and reason about them.

One important issue that affects near-term quantum computers is noise. Unlike classical systems, quantum computers are highly susceptible to various types of errors due to decoherence,

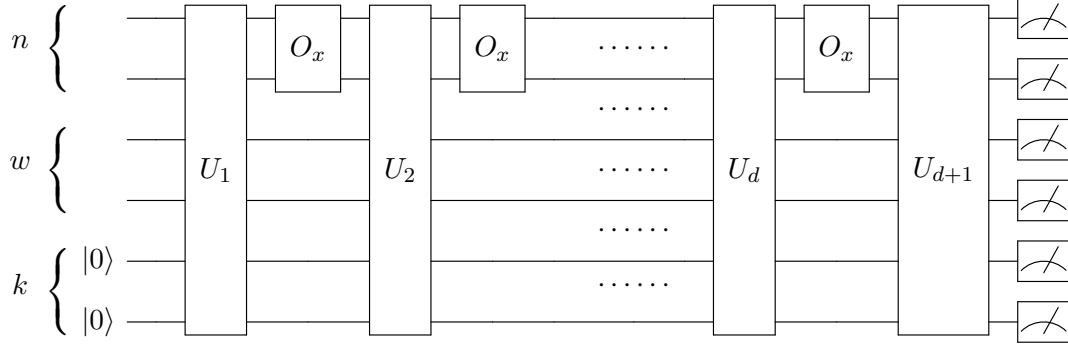


Figure 1: A d -query DQC_k algorithm. The initial state on the first $n + w$ qubits is maximally mixed.

imperfect gates, and environmental interactions. This prompts a natural question – how much noise can quantum algorithms tolerate? How does noise change the quantum computational complexity landscape? This is challenging to answer in general, as there are many different kinds of noise that affect quantum algorithms. One way to simplify this challenge is to consider models that have an extreme amount of noise. In this endeavor, researchers have proposed highly noisy models of quantum computation like DQC_k and $\frac{1}{2}\text{BQP}$ [KL98, MFF14, JM24] where all the noise is pushed onto the initial state – the qubits start maximally or nearly maximally mixed, while the gates are noiseless. These models provide a framework for understanding the minimal number of clean qubits required to achieve quantum speedups. We describe these models below.

DQC_k Drawing inspiration from the NMR approach to quantum computing where mixed states are ubiquitous, Knill and Laflamme [KL98] introduced the one-clean qubit or DQC_1 model as an idealized version of a noisy quantum computer. In this model, one qubit is clean (noiseless) and the rest are maximally noisy, and the algorithm can apply (noiseless) unitary gates on these qubits and measure at the end. This model was later generalized to DQC_k to allow k clean qubits [MFF14, FKM⁺15]. This model does not seem to be universal for quantum computing since all qubits except a few are maximally noisy and many oracle problems like Simon’s problem and order finding are not believed to be solvable in this model. Despite this, DQC_1 can solve problems that are believed to be classically hard, like estimating the trace and Pauli coefficients of a unitary matrix described by a quantum circuit [KL98, DFC05], Jones polynomials [SJ08], partition functions [CSS21]. Under complexity theoretic assumptions, this model is not classically simulable [MFF14, FKM⁺18, Mor17]. There are exponential oracle separations between DQC_1 and BPP [She10]. The communication version of the one clean qubit model provides exponential speedups over classical randomized communication [AGL23]. The fact that quantum speedups persist even under such extreme noise makes DQC_1 a particularly intriguing model for further study – it challenges our understanding of what minimal quantum resources are required for speedups.

$\frac{1}{2}\text{BQP}$ The $\frac{1}{2}\text{BQP}$ model was originally defined by [ABKM17] to capture the power of permutational computations on special input states. This model was revisited by [JM24] in the context of delineating the boundary between BQP and DQC_1 . In this model, the initial state is maximally mixed, i.e., a uniformly random computational basis state, but the algorithm learns this state at the end of the computation and decides whether to accept or reject. One can equivalently define this model as a quantum algorithm acting on one half of a maximally entangled EPR state and in the end, we measure both halves and do classical postprocessing on the measurement outcomes. This model is not believed to be universal for quantum computing as it allows a significant amount

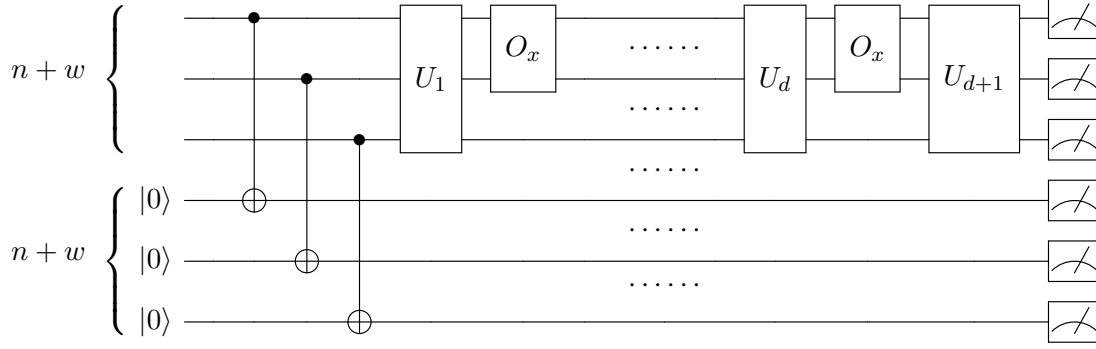


Figure 2: A d -query $\frac{1}{2}$ BQP algorithm. The initial state on the first $n + w$ qubits can be thought of as maximally mixed, or as the pure state $2^{-(n+w)/2} \sum_{I \in \{0,1\}^{n+w}} |I\rangle$; the resulting circuits are equivalent.

of noise, yet, this model encapsulates many known quantum speedups. It can solve the factoring problem and numerous oracle-based problems including Simon’s problem, Deutsch-Jozsa, order finding, and the Forrelation problem and can simulate DQC_k for any small k as well as IQP [JM24]. It appears to be the weakest quantum model that is unlikely to be universal and yet captures most known BQP speedups despite operating on maximally mixed states.

A powerful and natural framework to study the differences between $\text{DQC}_k, \frac{1}{2}\text{BQP}$ and BQP is query complexity. In this setting, there is a boolean function $f : \{-1, 1\}^N \rightarrow \{-1, 1\}$ and the goal is to compute $f(x)$ for $x \in \{-1, 1\}^N$ while minimizing the number of queries to the oracle O_x . This model strips away implementation details and captures the essence of what makes different computational models powerful. The aforementioned quantum models can be formalized using this framework and are depicted in Figures 1 and 2. (See Definitions 2.11 and 2.13 for more details.) Query complexity has long been one of the most fruitful arenas for understanding the differences between quantum and classical computation and gives us strong evidence for quantum advantage, including provable exponential oracle separations between BQP and BPP. Over the years, the field has also developed an impressive arsenal of lower-bound techniques for both quantum and classical algorithms. While these techniques are powerful for distinguishing quantum from classical, they are not designed to distinguish between quantum algorithms. Indeed, many of these methods – including the polynomial method – apply uniformly to all bounded low-degree polynomials and cannot capture the subtle differences between $\text{DQC}_1, \frac{1}{2}\text{BQP}$ and BQP. This motivates the search for more fine-grained analytic techniques.

The central contribution of this paper is to show that a Fourier analytic concept known as Fourier growth provides exactly such a tool. While Fourier growth was historically used to distinguish between quantum and classical algorithms, we demonstrate that it can also serve as a lens to separate quantum models from each other. We show that noise dampens the Fourier growth of quantum algorithms in ways that are intricately tied to the noise patterns. We now introduce Fourier growth, provide its historical context and describe its importance.

1.2 Fourier Growth

Fourier growth has emerged as a central concept that allows us to distinguish quantum and classical algorithms. To formally define Fourier growth, recall that every boolean function $f :$

$\{-1, 1\}^N \rightarrow \mathbb{R}$ can be uniquely represented as a multi-linear polynomial

$$f(x) = \sum_{S \subseteq [N]} \hat{f}(S) \cdot \prod_{i \in S} x_i$$

where $\hat{f}(S)$ are called the Fourier coefficients of f .

Definition 1.1 (Signed Fourier Growth). *For level $\ell \in \mathbb{N}$, and signs $\alpha_S \in [-1, 1]$ for $S \subseteq [N]$ with $|S| = \ell$, define the α -signed level- ℓ Fourier growth of f , denoted by $L_{1,\ell}^\alpha(f)$ as*

$$L_{1,\ell}^\alpha(f) := \sum_{\substack{S \subseteq [N] \\ |S| = \ell}} \alpha_S \cdot \hat{f}(S),$$

Definition 1.2 (Fourier Growth). *For level $\ell \in \mathbb{N}$, the level- ℓ Fourier growth of f , denoted by $L_{1,\ell}(f)$, is the ℓ_1 -norm of the level- ℓ Fourier coefficients of f ,*

$$L_{1,\ell}(f) := \sum_{\substack{S \subseteq [N] \\ |S| = \ell}} |\hat{f}(S)| = \max_{\alpha \in [-1, 1]^{\binom{N}{\ell}}} L_{1,\ell}^\alpha(f).$$

Fourier growth bounds have been extensively studied and established for various classical models¹, including small-width DNFs/CNFs [Man95], AC⁰ circuits [Tal17], low-depth decision trees [Tal20, SSW23], low-degree GF(2) polynomials [CHLT19], low-depth parity decision trees [GTW21], low-degree bounded polynomials [IRR⁺21], and more. Upper bounds on the Fourier growth, even for the first few levels, give rise to quantum versus classical separations. Intuitively, while both quantum and classical algorithms of small query complexity can be represented by low-degree polynomials, the polynomials associated with quantum algorithms are a lot “denser” compared to their classical analogues, and this density is captured by Fourier growth. In particular, it was shown by [Tal20, SSW23] that for d -query classical algorithms, $L_{1,\ell}(f)$ is at most $\tilde{O}_\ell(d^{\ell/2})$; on the other hand, for d -query quantum algorithms, $L_{1,\ell}(f)$ is at most $O_\ell(d^\ell) \cdot N^{\ell/2-1}$ [IRR⁺21] and this can be tight for certain algorithms. A key problem that exploits this difference in the Fourier growth is the Forrelation problem. This was originally introduced by Aaronson and Ambainis [Aar10, AA15] to show an oracle separation between BQP and BPP and was subsequently used by Raz and Tal [RT22] in their breakthrough oracle separation of BQP and PH. Building on this, [BS21] generalized this to the k -Forrelation problem and used it to show optimal separations between BQP and BPP. We describe this problem below.

Definition 1.3 (k -Forrelation function). *Let $N = 2^n$. For $x^{(1)}, \dots, x^{(k)} \in \{-1, 1\}^N$, define*

$$\text{forr}^{(k)}(x^{(1)}, \dots, x^{(k)}) := \langle v | H_N \cdot O_{x^{(1)}} \cdot H_N \cdot O_{x^{(2)}} \cdots H_N \cdot O_{x^{(k)}} \cdot H_N | v \rangle$$

where H_N is the $N \times N$ unitary Hadamard matrix as in Definition 2.1 and $|v\rangle = |0 \dots 0\rangle$.

Definition 1.4 (k -FORRELATION problem with parameter $\varepsilon = \Theta(1/\log^k N)$). *Given input $x \in \{-1, 1\}^{kN}$, return -1 if $\text{forr}^{(k)}(x) \geq 2\varepsilon$ and 1 if $\text{forr}^{(k)}(x) \leq \varepsilon$.*

¹By Fourier growth of a model, we refer to the Fourier growth of the acceptance probability of an algorithm in this model.

Fourier Growth of d -query algorithms	$\ell = 1$	$\ell = 2$	$\ell = 3$
BQP [IRR ⁺ 21], Theorem 1.7	d	$d^2\sqrt{N}$	d^3N
$\frac{1}{2}$ BQP Theorem 1.6	d	$d^2\sqrt{N}$	$d^3\sqrt{N}$
DQC ₁ Theorem 1.5	d	d^2	$d^3\sqrt{N}$
BPP [Tal20, SSW23]	$\sqrt{d}$	$d\sqrt{\log N}$	$\sqrt{d^3} \log N$

Table 1: Upper Bounds on the Fourier growth of the acceptance probability of various d -query algorithms, up to $O(1)$ factors.

Quantum algorithms in the BQP model can solve k -FORRELATION using $\lceil k/2 \rceil$ quantum queries. Furthermore, the results of [RT22, CHLT19, RT22, BS21] imply that any family of algorithms solving k -FORRELATION must have large Fourier growth at levels $k, 2k, \dots, k(k-1)$ (see Theorem 2.6 and Theorem 2.8 for a precise statement). These results effectively reduce the task of proving lower bounds for the Forrelation problem to the task of establishing Fourier growth bounds. In particular, 2-FORRELATION involves level-2 bounds and 3-FORRELATION involves level-3 and level-6 bounds. Since classical algorithms have small Fourier growth at all levels, it follows from the aforementioned works that they cannot solve the Forrelation problem.

1.3 Our Results

In our work, we go beyond the idea of using Fourier growth to distinguish between quantum and classical algorithms. We show that although Fourier growth can be large for quantum algorithms, just how large it can be depends on the kind of quantum resources used and the types of noise present. In particular, we establish Fourier growth bounds for DQC _{k} , $\frac{1}{2}$ BQP and BQP algorithms. The bounds we obtain for $\ell = 1, 2, 3$ are summarized in Table 1 and depicted in Figure 3.

DQC _{k} algorithms.

Theorem 1.5. *Let $f(x)$ be the acceptance probability of a d -query DQC _{k} algorithm and $\rho \in \{-1, 1, *\}^N$ be any restriction. Then, for all $\ell \geq 2$, we have*

$$L_{1,\ell}(f|_\rho) \leq \min\left(2^{k/2}, \sqrt{N}\right) \cdot \binom{2d}{\ell} \cdot N^{(\ell-2)/2}.$$

We prove this in Section 4 and show that the dependence on k and N are individually optimal in Section 4.2. Here, the dependence on N is particularly interesting. As we will see in Theorem 1.7, the Fourier growth of DQC₁ algorithms falls short of the growth of general BQP algorithms by a factor of $\sqrt{N}$ at each level.

$\frac{1}{2}$ BQP algorithms. For the $\frac{1}{2}$ BQP model, we are unable to prove $L_{1,3}$ and $L_{1,6}$ bounds. Currently, we do not have any upper bounds on $L_{1,3}, L_{1,6}$ that are stronger than the ones for general BQP

algorithms. Nevertheless, for our applications to Forrelation lower bounds, it turns out that we only need to deal with a certain family of signs, which we are able to do ¹.

Theorem 1.6. *Let $f(x)$ be the acceptance probability of a d -query $\frac{1}{2}$ BQP algorithm and $\rho \in \{-1, 1, *\}^{3N}$ be any restriction. Let $\gamma \in [-1, 1]^{3N}$ and $\alpha(\gamma) \in [-1, 1]^{\binom{3N}{3}}, \beta(\gamma) \in [-1, 1]^{\binom{3N}{6}}$ be signs as in [Definition 2.7](#). Then,*

$$L_{1,3}^{\alpha(\gamma)}(f|_{\rho}) \leq O(d^3) \cdot \sqrt{N},$$

$$L_{1,6}^{\beta(\gamma)}(f|_{\rho}) \leq O(d^6) \cdot \sqrt{N^3}.$$

We prove this in [Section 5](#). We are unaware if this bound is tight, or if one can derive a similar bound for all families of signs (see [Section 1.7](#))².

BQP algorithms.

Theorem 1.7. *Let $f(x)$ be the acceptance probability of a d -query BQP algorithm and $\rho \in \{-1, 1, *\}^N$ be any restriction. Then,*

$$L_{1,\ell}(f|_{\rho}) \leq \binom{2d}{\ell} \cdot N^{(\ell-1)/2}.$$

We prove this in [Section 6](#). The dependence on N is tight due to the k -FORRELATION problem. The best-known bound prior to this work is an upper bound of $d^{\ell} \cdot \exp\left(\binom{\ell+1}{2}\right) \cdot N^{(\ell-1)/2}$ for bounded degree- d polynomials due to [\[IRR⁺21\]](#). We see in this expression that the dependence on d, ℓ is of the form $d^{\ell} \cdot \exp(\ell^2/2)$, which is quite large for $\ell \gtrsim \sqrt{d}$, in contrast to our dependence, which is at most $\binom{2d}{\ell} \leq d^{\ell} \cdot \exp(-\ell)$. We are not aware if this dependence is tight and leave this for future work (see [Section 1.7](#)).

We remark that variants of [Theorems 1.5 to 1.7](#) hold even with classical pre-processing. The proof of this is quite simple and is deferred to [Appendix A.1](#).

Comparison to Prior Works. While Fourier growth has been extensively studied for classical algorithms, we are aware of only a few works that explicitly study the Fourier growth of quantum algorithms [\[AG23, GSTW24, IRR⁺21\]](#). Among these, [\[IRR⁺21\]](#) and [\[GSTW24\]](#) are closely related to our work. As mentioned before, [\[IRR⁺21\]](#) establishes bounds on the Fourier growth of BQP algorithms that is slightly weaker than ours; furthermore, their bounds apply to all bounded low-degree polynomials and consequently cannot be used to distinguish between BQP, $\frac{1}{2}$ BQP and DQC₁.

The work of [\[GSTW24\]](#) is especially closely related to our work. They study quantum algorithms with k rounds of parallel queries and show that reducing the number of rounds even by one can cause a large blowup in the quantum query complexity. They achieve this by showing Fourier growth bounds for k -round quantum algorithms and leveraging the differences between the bounds for different k . Our work shares some conceptual similarities with their work, particularly in leveraging Fourier growth bounds to distinguish between quantum models, and also in using similar

¹ We observe that [\[BS21\]](#) show that to establish lower bounds for 3-FORRELATION, one only needs to prove signed-Fourier growth bounds for a particular family of signs. (See [Definition 2.7](#) and [Theorem 2.8](#) for more details.) When we refer to the Fourier growth of $\frac{1}{2}$ BQP algorithms, we typically mean signed-Fourier growth for signs as in [Theorem 2.8](#) and [Definition 2.7](#).

²We remark for this family of signs, BQP algorithms can already achieve a significantly larger Fourier growth. In particular, consider the acceptance probability $f(x)$ of the two-query BQP algorithm that solves 3-FORRELATION. For $\gamma = (1, \dots, 1)$, one can show that $L_{1,3}^{\alpha(\gamma)}(f) = \Omega(N)$.

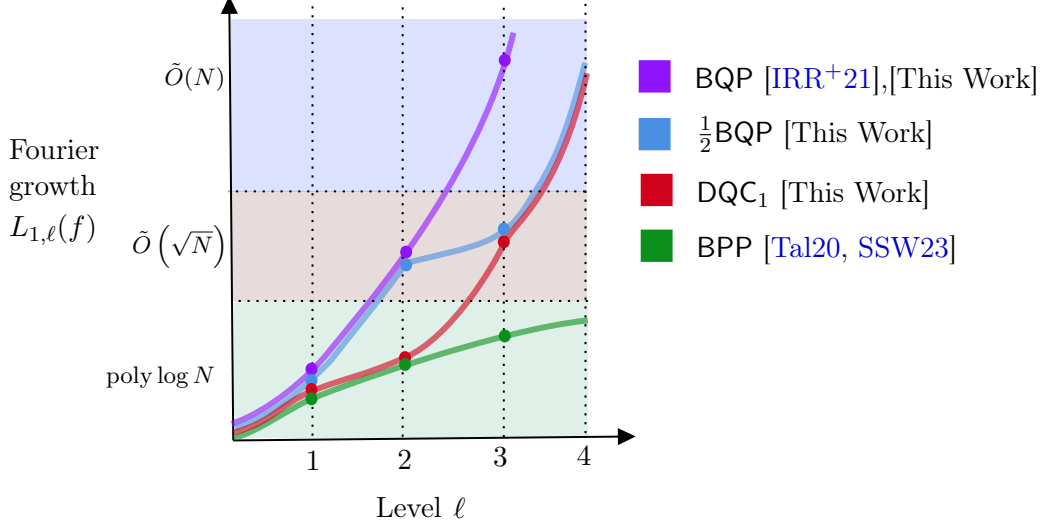


Figure 3: Fourier growth of acceptance probability of algorithms with $d = \text{poly log } N$ queries.

techniques for storing information about parities within matrix products. However, the models of quantum computation we consider are completely different. In [GSTW24], the number of rounds is constrained, while the number of clean qubits is unlimited and the initial state is the all-zeroes state. In contrast, in our setting, the number of clean qubits is constrained and the initial state is forced to be highly mixed, while the number of rounds is allowed to be large. These differing constraints lead to fundamentally different behaviors. Consequently, our techniques diverge from theirs and we require distinct ideas and develop new techniques.

It is worth emphasizing that the idea of using Fourier growth to distinguish between low-degree polynomials arising from different types of algorithms traces back to the landmark oracle separation of BQP and PH [RT22]. The core challenge in that setting was that both models admit low-degree polynomial approximations, and Fourier growth was used precisely to tell these polynomials apart.

1.4 Applications

We study the complexity of the Forrelation problem and its variants in the DQC_k and $\frac{1}{2}\text{BQP}$ models. Combining our Fourier growth bound (Theorem 1.5) with the results of [RT22, CHLT19] (see Theorem 2.6) and the upper bounds on 2-FORRELATION from [Aar10, AA15], we immediately obtain the following corollary.

Corollary 1.8. *For any $k \in \mathbb{N}$, the 2-FORRELATION problem on 2^k -bit inputs can be solved by a DQC_k algorithm with success probability at least $2/3$ by making one quantum query, however, any DQC_{k-t} algorithm that makes d quantum queries has success probability at most $\frac{1}{2} + \tilde{O}(d^2) \cdot 2^{-t/2}$.*

In particular, any DQC_{k-t} algorithm that succeeds with probability at least $2/3$ must make at least $\tilde{\Omega}(2^{t/4})$ queries. Setting $k = \log N$, we obtain the following corollary.

Corollary 1.9. *The 2-FORRELATION problem on N -bit inputs, which can be solved with $\log N$ clean qubits and one quantum query, requires $\tilde{\Omega}(N^{c/4})$ queries in the $\text{DQC}_{(1-c)\log N}$ model for all constants $c < 1$. In particular, any DQC_1 algorithm for 2-FORRELATION must make $\tilde{\Omega}(N^{1/4})$ queries.*

We remark that Corollary 1.9 holds even if the algorithm is allowed to make $\tilde{\Omega}(N^{c/4})$ classical

pre-processing queries in advance (using clean bits). We derive the following implications of [Corollary 1.9](#).

A Hierarchy Theorem for DQC_k . In this work, we quantify the power that each additional clean qubit gives to quantum algorithms. It is not too difficult to show that any DQC_k algorithm can be simulated by a DQC_{k-t} algorithm without additional queries but with a loss of $2^{\Theta(t)}$ in the advantage ([Claim A.3](#)). [Corollary 1.8](#) shows that this is tight, up to a constant in the exponent. This shows that the number of clean qubits in a quantum algorithm cannot be efficiently reduced, even with a large amount of classical pre-processing on clean bits.

The First Oracle separation between $\frac{1}{2}\text{BQP}$ and DQC_1 . We give the first oracle separation between $\frac{1}{2}\text{BQP}$ and DQC_1 , resolving a conjecture of [\[JM24\]](#). In particular, Jacobs and Mehraban showed that 2-FORRELATION on N -bit inputs is solvable in the $\frac{1}{2}\text{BQP}$ model with two quantum queries and conjectured that it requires $N^{\Omega(1)}$ queries in the DQC_1 model (see open question #1 on page 8 [\[JM24\]](#)). Our work ([Corollary 1.8](#)) proves this conjecture.

A New Oracle separation between BQP and $\frac{1}{2}\text{BQP}$. Jacobs and Mehraban conjectured (see open question #5 on page 8 [\[JM24\]](#)) that 3-FORRELATION is not in $\frac{1}{2}\text{BQP}$ and our work ([Corollary 1.10](#)) resolves this. By combining our Fourier growth bound ([Theorem 1.6](#)) with the results of [\[BS21\]](#) ([Theorem 2.8](#)), we immediately obtain the following corollary.

Corollary 1.10. *The 3-FORRELATION problem on $3N$ -bit inputs, which can be solved by a BQP algorithm with two quantum queries, requires $\tilde{\Omega}(N^{1/12})$ queries in the $\frac{1}{2}\text{BQP}$ model.*

We also remark that [Corollary 1.10](#) holds even if the algorithm is allowed to make $\tilde{\Omega}(N^{1/12})$ classical pre-processing queries in advance.

1.5 Technical Highlight: Matrix Decomposition Lemma

The main recurring technique in our paper is the use of a matrix decomposition lemma (see [Lemma 3.1](#)). This lemma offers a way to encode information about the indices involved in a matrix product and arises naturally in the context of quantum algorithms, as it allows us to encode information about the Fourier coefficients within a sequence of matrix products. We think it might be of independent interest.

Firstly, we observe that the acceptance probability of quantum algorithms can be expressed as a product of matrices with bounded operator norms. To give some intuition, fix $i_1, i_{d+1} \in [N]$. Consider a sequence of unitary matrices $U_1, \dots, U_d$ and let $U[i|j]$ denote the $(i, j)^{\text{th}}$ -entry of U . Consider a BQP algorithm that starts with the initial state $|i_1\rangle$, evolves it according to the unitary operators $U_1, \dots, U_d$, interleaved with phase oracles O_x and finally measures the qubits and accepts if the outcome is $|i_{d+1}\rangle$. The acceptance probability of this algorithm is given by $f(x)^2$ where

$$\begin{aligned} f(x) &:= \langle i_1 | U_1 \cdot O_x \cdot U_2 \cdot O_x \cdots O_x \cdot U_d | i_{d+1} \rangle \\ &= \sum_{i_2, \dots, i_d} \left(\prod_{t \in [d]} U_t[i_t | i_{t+1}] \right) \cdot \left(\prod_{t \in [d] \setminus \{1\}} x_{i_t} \right) \end{aligned}$$

More generally, by allowing the matrices $U_1, \dots, U_d$ to be arbitrary matrices with spectral norm at most 1 and by adding workspace, we can produce a similar expression for $f(x)$ which equals the

acceptance probability of an arbitrary $\lfloor d/2 \rfloor$ -query BQP algorithm (see [Claim 2.10](#)). There are other expressions for capturing the acceptance probability of DQC_k and $\frac{1}{2}$ BQP algorithms using matrix products (see [Claim 2.12](#) and [Claim 2.14](#)). Now that we have an expression for the acceptance probability, we need to compute the Fourier coefficients. Observe that for all $S \subseteq [N]$,

$$\hat{f}(S) = \sum_{i_2, \dots, i_d} \prod_{t \in [d]} U_t[i_t|i_{t+1}] \cdot \mathbb{1}[S = \{i_2\} \oplus \dots \oplus \{i_d\}].$$

Our main idea is to try and encode information about the Fourier coefficients inside a product of matrices with bounded norms. The hope is that since $f(x)$ itself is a product of matrices with bounded norms, so are its Fourier coefficients. To illuminate the main idea, say we wish to multiply the matrices $U_1, \dots, U_d$ to get a matrix U where

$$U[i_1|i_{d+1}] = \sum_{i_2, \dots, i_d} \prod_{t \in [d]} U_t[i_t|i_{t+1}]$$

but additionally, we wish to retain information about the symmetric difference of the intermediate indices $\{i_2\}, \dots, \{i_d\}$ until the very end. More formally, we wish to design a matrix $\tilde{U}$ whose rows are indexed by i_1 and columns by $i_{d+1}S_{d+1}$ such that

$$\tilde{U}[i_1|i_{d+1}S_{d+1}] = \sum_{i_2, \dots, i_d} \prod_{t \in [d]} U_t[i_t|i_{t+1}] \cdot \mathbb{1}[S_{d+1} = \{i_2\} \oplus \dots \oplus \{i_d\}].$$

Here, the indicator function ensures that for each S_{d+1} , the corresponding entry of the final matrix only involves contributions from indices that satisfy the parity condition with respect to S_{d+1} . The reason we want to do this is clear; the entry $\tilde{U}[i_1|i_{d+1}S]$ precisely equals the Fourier coefficient $\hat{f}(S)$. Thus, by reading off the entries of matrix $\tilde{U}$ restricted to rows corresponding to i_1 and columns corresponding to i_{d+1} , we would obtain the list of all Fourier coefficients. The challenge lies in constructing such a matrix $\tilde{U}$ with bounded norms and this is precisely achieved by [Lemma 3.1](#). It embeds the required combinatorial information about the indices within a matrix product while maintaining control over the norms of $\tilde{U}$. We also show an improved matrix decomposition lemma ([Lemma 3.3](#)) that allows slightly more complex predicates of the indices being summed over – in particular, we allow the imposition of equality constraints between indices as well as memory constraints on indices.

1.6 Proof Sketch

In general, proving Fourier growth bounds is quite challenging and technically involved. A major challenge arises from the need to incorporate the signs $\alpha_S \in [-1, 1]$ into the matrix product given by the matrix decomposition lemma, and also from the need to sum over all sets S of size ℓ . Introducing the signs in a naive fashion often blows up the operator norms of the underlying matrices, making it difficult to maintain control over the Fourier growth. The heart of our proof involves techniques to incorporate these signs while keeping the operator norms bounded. This step turns out to be especially challenging for $\frac{1}{2}$ BQP algorithms and we are unable to handle arbitrary signs α_S . However, we are able to successfully encode the signs that arise from the 3-FORRELATION problem. For level-6, we run into additional difficulties that require developing more complex ways of storing information within matrix products and this is handled by the improved matrix decomposition lemma ([Lemma 3.3](#)).

In this section, we present the simplest part of our proof: using the matrix decomposition lemma ([Lemma 3.1](#)) to establish Fourier growth bounds for DQC_1 algorithms. We will make some

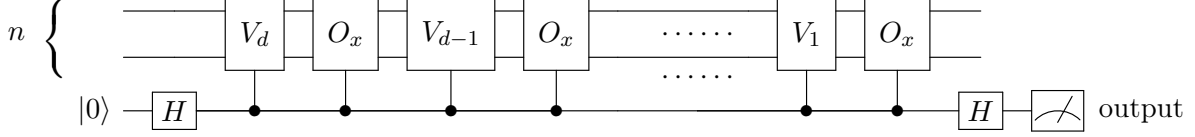


Figure 4: A simple example of a d -query DQC_1 algorithm. The initial state on the first n qubits is maximally mixed.

simplifications: we only focus on level $\ell = 2$; we will assume that there is no restriction ρ on the inputs; and we will only consider algorithms with one clean qubit of a special form in [Figure 4](#). These simplifications are only for the proof sketch and still give enough intuition for the general case.

Firstly, it is not too difficult to derive an expression for acceptance probability of the algorithm in [Figure 4](#). This is given by $\frac{1}{2} + \frac{1}{2}f(x)$ where

$$\begin{aligned} f(x) &:= \frac{1}{N} \text{Tr} (O_x \cdot V_1 \cdot O_x \cdot V_2 \cdots O_x \cdot V_d) \\ &= \frac{1}{N} \sum_{i_1, \dots, i_d \in [N]} \left(\prod_{t \in [d]} V_t[i_t | i_{t+1}] \right) \cdot \left(\prod_{t \in [d]} x_{i_t} \right) \end{aligned} \quad (1)$$

where $V_1 \dots, V_d$ are the $N \times N$ unitary matrices applied by the algorithm and we use the convention that $i_{d+1} = i_1$. One can derive a similar expression for the acceptance probability of an arbitrary DQC_k algorithm (see [Claim 2.12](#) for more details). Let us now compute the Fourier coefficients of the acceptance probability, which equals (up to a factor of $1/2$) the Fourier coefficients of $f(x)$, which are easy to read off of [Equation \(1\)](#). For any $S \subseteq [N]$, the S -th Fourier coefficient of f is given by

$$\hat{f}(S) = \frac{1}{N} \sum_{i_1, \dots, i_d \in [N]} \left(\prod_{t \in [d]} V_t[i_t | i_{t+1}] \right) \cdot \mathbb{1}[\{i_1\} \oplus \dots \oplus \{i_d\} = S]. \quad (2)$$

The quantity we wish to bound is the level-2 Fourier growth of f , i.e., $L_{1,2}(f) = \max_{\alpha} L_{1,2}^{\alpha}(f)$, where

$$L_{1,2}^{\alpha}(f) \triangleq \sum_{|S|=2} \alpha_S \cdot \hat{f}(S) \quad (3)$$

for signs $\alpha_S \in [-1, 1]$ for $S \subseteq [n]$ of size 2. Fix any such signs α . Substituting the expression for Fourier coefficients $\hat{f}(S)$ ([Equation \(2\)](#)) in the expression for $L_{1,2}^{\alpha}(f)$ ([Equation \(3\)](#)), we see that our goal is to upper bound

$$L_{1,2}^{\alpha}(f) = \sum_{|S|=2} \alpha_S \cdot \frac{1}{N} \sum_{i_1, \dots, i_d \in [N]} \left(\prod_{t \in [d]} V_t[i_t | i_{t+1}] \right) \cdot \mathbb{1}[\{i_1\} \oplus \dots \oplus \{i_d\} = S]. \quad (4)$$

Decomposing $L_{1,2}^{\alpha}$ into a few terms. First, we will group the terms in [Equation \(4\)](#) into a few terms. We will express $L_{1,2}(f)^{\alpha}$ as a sum over pairs (t_1, t_2) such that $t_1 \neq t_2 \in [d]$ of a quantity $\Delta_{t_1, t_2}^{\alpha}$. We describe this below.

Observe that for a term to contribute to [Equation \(4\)](#), the symmetric difference of $i_1, \dots, i_d$ has size 2. In this case, there must exist a pair of indices $t_1 < t_2 \in [d]$ such that i_{t_1} and i_{t_2}

are distinct and the symmetric difference of the rest of the i_t is the empty set. More precisely, if $\{i_1\} \oplus \dots \oplus \{i_d\} = S$ for a set S of size 2, then

$$\exists t_1 < t_2 \in [d] \text{ such that } \{i_{t_1}, i_{t_2}\} = S \text{ and } \bigoplus_{t \in [d] \setminus \{t_1, t_2\}} \{i_t\} = \emptyset.$$

Conversely, any such $t_1, t_2 \in [d]$ and $i_1, \dots, i_d$ satisfying the above equation defines a unique $S = \{i_{t_1}, i_{t_2}\}$. For any pair of indices $t_1 < t_2 \in [d]$, let Δ_{t_1, t_2}^α be the contribution of the corresponding terms to $L_{1,2}^\alpha(f)$, i.e.,

$$\Delta_{t_1, t_2}^\alpha := \frac{1}{N} \sum_{i_{t_1} \neq i_{t_2} \in [N]} \alpha_{\{i_{t_1}, i_{t_2}\}} \cdot \sum_{\substack{i_{t_1+1}, \dots, i_{t_2-1} \in [N] \\ i_{t_2+1}, \dots, i_{t_1-1} \in [N]}} \left(\prod_{t \in [d]} V_t[i_t | i_{t+1}] \right) \cdot \mathbb{I} \left[\bigoplus_{t \in [d] \setminus \{t_1, t_2\}} \{i_t\} = \emptyset \right]. \quad (5)$$

Clearly, we have $L_{1,2}^\alpha(f) = \sum_{t_1 < t_2 \in [d]} \Delta_{t_1, t_2}^\alpha$. Observe that there are $O(d^2)$ choices of $t_1 < t_2 \in [d]$. For any such choice, we will show in the second step that $\Delta_{t_1, t_2}^\alpha \leq 1$, obtaining $L_{1,2}^\alpha(f) \leq O(d^2)$ as desired.

Showing that $\Delta_{t_1, t_2}^\alpha \leq 1$. This is where we will use the matrix decomposition lemma ([Lemma 3.1](#)). We will group the terms $t \in [d]$ into circular intervals $[t_1, t_2)$ and $[t_2, t_1)^3$. We will apply the matrix decomposition lemma on $V_{t_1}, \dots, V_{t_2-1}$ to remember the symmetric difference of $\{i_t\}$ for $t \in (t_1, t_2)$ and similarly on the matrices $V_{t_2}, \dots, V_{t_1-1}$ to remember the symmetric difference of $\{i_t\}$ for $t \in (t_2, t_1)$ and then enforce equality between these sets. More precisely, apply [Lemma 3.1](#) (with $T = \emptyset$) on the matrices $V_{t_1}^\rho, \dots, V_{t_2-1}^\rho$ to obtain $\tilde{V}_{[t_1, t_2)}$ and to $V_{t_2}^\rho, \dots, V_{t_1-1}^\rho$ backwards to obtain $\tilde{V}_{[t_2, t_1)}$ such that for all $i_{t_1}, i_{t_2} \in [N]$, $S_{t_2} \subseteq [N]$,

$$\tilde{V}_{[t_1, t_2)}[i_{t_1} | i_{t_2} S_{t_2}] = \sum_{i_t \in [N] \text{ for } t \in (t_1, t_2)} \left(\prod_{t \in [t_1, t_2)} V_t[i_t | i_{t+1}] \right) \cdot \mathbb{I} \left[\bigoplus_{t \in (t_1, t_2)} \{i_t\} = S_{t_2} \right], \quad (6)$$

$$\tilde{V}_{[t_2, t_1)}[i_{t_1} | i_{t_2} S_{t_2}] = \sum_{i_t \in [N] \text{ for } t \in (t_2, t_1)} \left(\prod_{t \in [t_2, t_1)} V_t[i_t | i_{t+1}] \right) \cdot \mathbb{I} \left[\bigoplus_{t \in (t_2, t_1)} \{i_t\} = S_{t_2} \right]. \quad (7)$$

Substituting [Equations \(6\) and \(7\)](#) in [Equation \(5\)](#), we see that

$$\begin{aligned} \Delta_{t_1, t_2}^\alpha &\triangleq \frac{1}{N} \sum_{i_{t_1} \neq i_{t_2} \in [N]} \alpha_{\{i_{t_1}, i_{t_2}\}} \sum_{S_{t_2} \subseteq [N]} \tilde{V}_{[t_1, t_2)}[i_{t_1} | i_{t_2} S_{t_2}] \cdot \tilde{V}_{[t_2, t_1)}[I_{t_1} | I_{t_2} S_{t_2}] \\ &\leq \frac{1}{N} \sum_{i_{t_1} \neq i_{t_2} \in [N]} \sum_{S_{t_2} \subseteq [N]} \left| \tilde{V}_{[t_1, t_2)}[i_{t_1} | i_{t_2} S_{t_2}] \right| \cdot \left| \tilde{V}_{[t_2, t_1)}[I_{t_1} | I_{t_2} S_{t_2}] \right| \quad (\text{since } \alpha_{\{i_{t_1}, i_{t_2}\}} \in [-1, 1]) \\ &\leq \frac{1}{N} \cdot \left\| \tilde{V}_{[t_1, t_2)} \right\|_{\text{frob}} \cdot \left\| \tilde{V}_{[t_2, t_1)} \right\|_{\text{frob}} \end{aligned} \quad (\text{Fact 2.3})$$

Firstly, observe that

$$\max \left(\left\| \tilde{V}_{[t_1, t_2)} \right\|_{\text{frob}}, \left\| \tilde{V}_{[t_2, t_1)} \right\|_{\text{frob}} \right) \leq \sqrt{N}.$$

This is because both matrices have operator norm at most one and either have at most N rows or N columns. This implies that $\Delta_{t_1, t_2}^\alpha \leq N^{-1} \cdot N \leq 1$. This completes the proof sketch.

³We arrange $1, \dots, d$ in a clock-wise circle and define the intervals clock-wise. For instance, the interval $[d-2, 2]$ refers to the set $\{d-2, d-1, d, 1, 2\}$. The intervals (t_1, t_2) and (t_2, t_1) are well-defined but would be empty if $t_2 = t_1 \pm 1$ modulo d . In each of these cases, it is understood that the summation over $I_{t_1+1}, \dots, I_{t_2-1}$ and $I_{t_2+1}, \dots, I_{t_1-1}$ respectively is to be ignored.

We now describe some of the additional ideas involved in generalizing this proof.

Generalizing to higher levels. Proving bounds for higher levels for DQC_k algorithms requires one additional new idea that involves an improved matrix decomposition lemma, where in addition to remembering parity information, we store the values of certain subsets of indices until the very end, furthermore, to get the optimal dependence on k , we need an improved bound on the Frobenius norm. (See [Section 4](#) for more details.) The proof strategy is quite similar for BQP algorithms as well (see [Section 6](#)).

$\frac{1}{2}$ BQP algorithms. It is not too hard to show that the expression for the acceptance probability of a d -query $\frac{1}{2}$ BQP algorithm is quite similar to [Equation \(1\)](#), except, there are $2d$ matrices $V_1, \dots, V_{2d}$, and more importantly, there is an extra term of the form $F_{i_1, i_d} \in \{0, 1\}$ inside the summation, which corresponds to the post-processing of the measurement outcomes of the initial and final states. (See [Equation \(53\)](#) and [Claim 2.14](#) for a formal expression.) This additional term F_{i_1, i_d} is challenging to incorporate while keeping the norms bounded. As a result, proving bounds for $\frac{1}{2}$ BQP algorithms turns out to be more technically involved. We need to use an improved matrix decomposition lemma ([Lemma 3.3](#)) where we enforce memory constraints as well as equality constraints on the indices being summed over.

Furthermore, we are only able to prove level-3 and level-6 Fourier growth bounds for a particular family of signs as in [Definition 2.7](#). The reason why the signs $\alpha(\gamma)$ and $\beta(\gamma)$ in [Definition 2.7](#) are easier to deal with than general signs, is that once we fix i_2 , $\alpha(\gamma)_{i_1, i_2, i_3}$ becomes a product of three terms, the first depending only on i_1 , the second on i_3 and the third on γ in a product fashion. Similarly, once we fix i_2, i_5 , then $\beta(\gamma)_{i_1, \dots, i_6}$ becomes a product of five terms, the first depending only on i_1 , the second on i_4 , the third on i_3 , the fourth on i_6 , and the fifth on γ in a product fashion. These kind of signs that are products across the indices are much easier to handle than general families of signs and often exhibit a Fourier growth that is much smaller than the Fourier growth for arbitrary signs⁴. We then show that summing over the i_2 , or over the i_2, i_5 doesn't blow up the Fourier growth by much. (See [Section 5](#) for more details.)

1.7 Outlook & Future Directions

Broadly, our results suggest that Fourier growth provides a powerful analytic lens to separate models of quantum computation. Several natural next steps emerge in this direction and we highlight some open questions in this section.

1. **Fourier Growth of NISQ.** Researchers have attempted to model NISQ (noisy intermediate scale quantum) algorithms through the lens of query complexity, in the hopes of understanding the computational power of near-term quantum devices [[CCHL23](#), [CHHK24](#)]. There has been recent interest in using 2-FORRELATION to show quantum advantages in near-term experiments [[Geo25](#), [Shu25](#)] and this prompts the natural question, can we solve 2-FORRELATION in NISQ? If not, can we prove bounds on the Fourier growth of NISQ?
2. **The Power of DQC_1 .** Where does DQC_1 fit within the landscape of classical complexity, and in particular, is it contained in PH? The differences between the Fourier growth of DQC_1 and PH are quite stark, but it is not clear how to leverage this into an oracle separation, as existing approaches rely on the Forrelation problem, which is hard for DQC_1 . Developing new

⁴Indeed, for general bounded degree- d polynomials, the level- ℓ Fourier growth with arbitrary signs can be as large as $N^{\Omega(\ell)}$, whereas for signs that are a product across the indices, the Fourier growth is at most $d^{O(\ell)}$ [[IRR⁺21](#)].

techniques here would not only clarify the power of DQC_1 , but also expand the toolkit for proving lower bounds on classical computation.

3. **The Power of IQP.** Another intriguing intermediate model is IQP, whose power derives from its ability to perform Fourier sampling. How does this model compare to DQC_1 and $\frac{1}{2}\text{BQP}$? Understanding the relationship between these models would help chart the intermediate landscape between BPP and BQP and reveal the relative power of various quantum capabilities like Fourier sampling and trace estimation. It was shown by [JM24] that IQP can be simulated by $\frac{1}{2}\text{BQP}$ and they conjectured that this containment is strict. Is 2-FORRELATION solvable in IQP and if not, can we prove Fourier growth bounds?
4. **Tight Bounds on the Fourier Growth of Quantum Algorithms.** Finally, many of our upper bounds on the Fourier growth are not known to be tight. Are the dependencies on d and ℓ tight in Theorems 1.5 to 1.7? What is the Fourier growth of $\frac{1}{2}\text{BQP}$ with respect to arbitrary families of signs? Tight bounds on Fourier growth could provide a precise handle for quantum computational power, and help map the landscape between classical, intermediate, and fully quantum models.

1.8 Organization.

Section 2 consists of preliminaries, where we formally describe the various models of computation and state the results we need from prior works on Forrelation. In Section 3, we describe and prove the matrix decomposition lemmas (Lemmas 3.1 and 3.3). We prove our Fourier growth bounds for DQC_k in Section 4 (proof of Theorem 1.5), $\frac{1}{2}\text{BQP}$ in Section 5 (proof of Theorem 1.6) and BQP in Section 6 (proof of Theorem 1.7).

2 Preliminaries & Notation

Restrictions. For a restriction $\rho \in \{-1, 1, *\}^N$ and a vector $x \in \{-1, 1\}^N$, the i -th coordinate of the restricted vector $\rho(x) \in \{-1, 1\}^N$ is ρ_i if $\rho_i \in \{-1, 1\}$ and x_i if $\rho_i = *$ for $i \in [N]$. For a boolean function $f : \{-1, 1\}^N \rightarrow \mathbb{R}$, and a restriction $\rho \in \{-1, 1, *\}^N$, we use $f|_\rho$ to denote the restricted function which maps x to $f(\rho(x))$ for $x \in \{-1, 1\}^N$.

Sets. For $x \in \mathbb{R}^N$ and $S \subseteq [N]$, we use $\chi_S(x)$ or x_S to denote $\prod_{i \in S} x_i$. For indices $i_1, \dots, i_k \in [N]$, we use $\{i_1\} \oplus \dots \oplus \{i_k\}$ to denote the symmetric difference $\oplus_{t \in [k]} \{i_t\}$ and similarly $S_1 \oplus S_2$ denotes the symmetric difference of the sets S_1 and S_2 .

We will often use uppercase letters to denote 2 to the power of lowercase letters, in particular, $N = 2^n$, $W = 2^w$, $K = 2^k$ and $M = 2^m$.

Circular Intervals. For $i, j \in [n]$, we use $[i, j]$ to denote the clockwise sequence of points from i to j when $1, \dots, n$ are arranged clock-wise in a circle. For example, $[n, 2] = \{n, 1, 2\}$ and $[1, 3] = \{1, 2, 3\}$. We use $(,]$ and $[,)$ and $(,)$ to denote half-open or open intervals.

Vectors and Inner Products. We identify the space $\{0, 1\}^n$ with $[N]$ under the natural correspondence $(a_1, \dots, a_n) \rightarrow 1 + \sum_i a_i 2^i$. We also identify $\{0, 1\}^n$ with $\{-1, 1\}^n$ under the correspondence that maps 0 to 1 and 1 to -1 . For $u, v \in [N]$, we use $\langle u, v \rangle_2 := \sum_{i \in [n]} u_i v_i \pmod 2$ to denote the inner product over $\mathbb{F}_2$ under the aforementioned correspondence. For $u \in \mathbb{C}^N$ and $U \in \mathbb{C}^{N \times N}$,

we use $u^\dagger, U^\dagger$ to denote the conjugate-transpose. For complex vectors $u, v \in \mathbb{C}^N$, we use $\langle u | v \rangle$, $v^\dagger u$, and $\langle u, v \rangle$ to denote $\sum_i u_i \bar{v}_i$, the complex inner product.

Matrices. We use $\mathbf{I}$ to denote the identity matrix, where the dimensions are clear from context. We will often encounter matrices whose rows and columns are indexed by (i, w) for $i \in [N], w \in [W]$, or by (i, w, k) for $i \in [N], w \in [W], k \in [K]$. For ease of notation, we use I as a shorthand for (i, w) or (i, w, k) , where the distinction will be clear from the context. For $I_t, I_{t+1} \in [M]$, we use either $U_t[I_t | I_{t+1}]$ or $U_t[I_t, I_{t+1}]$ to denote the (I_t, I_{t+1}) -the entry of U_t . For matrices $U_1, \dots, U_d$, we use $U_{[t_1, t_2]}$ to denote the product $\prod_{t \in [t_1, t_2]} U_t = U_{t_1} \cdots U_{t_2}$ of the matrices in the circular interval $[t_1, t_2]$ in clockwise order. We define $U_{[t_1, t_2]}, U_{(t_1, t_2]}, U_{[t_1, t_2)}$ analogously.

Definition 2.1 (Hadamard Matrix). *For $N = 2^n$, the Hadamard matrix H_N is defined to be*

$$H_N = \frac{1}{\sqrt{N}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}^{\otimes n}.$$

Matrix Norms & Inequalities. Let $\|\cdot\|_{\text{op}}$ and $\|\cdot\|_{\text{frob}}$ denote the spectral and Frobenius norm, or equivalently, the Schatten- ∞ and Schatten-2 norms. The following basic fact follows from Holder's Inequality for Schatten norms.

Fact 2.2. *Let A, B, C be rectangular matrices with $A = BC$. Then, $\|A\|_{\text{frob}} \leq \|B\|_{\text{op}} \cdot \|C\|_{\text{frob}}$.*

The Cauchy-Schwarz inequality implies the following fact.

Fact 2.3. *For rectangular matrices A, B , and any subset T of indices, we have*

$$\sum_{(i,j) \in T} |A[i|j]| \cdot |B[i|j]| \leq \|A\|_{\text{frob}} \cdot \|B\|_{\text{frob}}.$$

We use the following basic facts about the spectral norms of matrices.

Fact 2.4. *For any submatrix B of A , we have $\|B\|_{\text{op}} \leq \|A\|_{\text{op}}$.*

Fact 2.5. *For any block diagonal matrix A consisting of blocks $A_1, \dots, A_t$, we have $\|A\|_{\text{op}} \leq \max_{i \in [t]} \|A_i\|_{\text{op}}$.*

2.1 Fourier Growth

Recall the definition of the Fourier growth as in [Definition 1.1](#) and [Definition 1.2](#). For a family of functions $\mathcal{F}$, we use $L_{1,\ell}(\mathcal{F})$ to denote $\max_{f \in \mathcal{F}} L_{1,\ell}(f)$.

Lower Bounds for Forrelation from Fourier Growth. The results of [\[RT22, CHLT19\]](#) imply that to show lower bounds on the 2-FORRELATION problem, it suffices to prove Fourier growth bounds for level 2.

Theorem 2.6 ([\[RT22, CHLT19\]](#)). *Let $\mathcal{F}$ be any family of $2N$ -variate boolean functions closed under restrictions. Then, the maximum advantage with which $\mathcal{F}$ solves 2-FORRELATION is at most*

$$O\left(\frac{L_{1,2}(\mathcal{F})}{\sqrt{N}}\right).$$

The results of [BS21] imply that to show lower bounds on the 3-FORRELATION problem, it suffices to prove signed-Fourier growth bounds for level 3 and 6, for the following family of signs.

Definition 2.7. Partition $[3N]$ into $A := [N], B := (N, 2N], C := (2N, 3N]$. Let $\gamma \in [-1, 1]^{3N}$. Define $\alpha(\gamma) \in [-1, 1]^{\binom{3N}{3}}$ and $\beta(\gamma) \in [-1, 1]^{\binom{3N}{6}}$ as follows. Let $\overline{H} \in \{-1, 1\}^{N \times N}$ be the matrix whose (i, j) -th entry is $(-1)^{\langle i, j \rangle_2} = \text{sign}(H_N[i, j])$ for $i, j \in [N]$. For $i_1, i_2, i_3 \in [3N]$, let

$$\alpha(\gamma)_{i_1, i_2, i_3} := \begin{cases} \overline{H}(i_2, i_1) \cdot \overline{H}(i_2, i_3) \cdot \left(\prod_{t \in [3]} \gamma_{i_t} \right) & \text{if } i_1 \in A, i_2 \in B, i_3 \in C \\ 0 & \text{otherwise.} \end{cases}$$

For $i_1, \dots, i_6 \in [3N]$, let

$$\beta(\gamma)_{i_1, \dots, i_6} := \begin{cases} \alpha(\gamma)_{i_1, i_2, i_3} \cdot \alpha(\gamma)_{i_4, i_5, i_6} & \text{if } i_1 \neq i_4 \in A, i_2 \neq i_5 \in B, i_3 \neq i_6 \in C \\ 0 & \text{otherwise.} \end{cases}$$

The following theorem is implicit in [BS21].⁵

Theorem 2.8 (Implicit in [BS21]). Let $\mathcal{F}$ be any family of $3N$ -variate boolean functions that is closed under restrictions. Let $\gamma \in [-1, 1]^{3N}$ and $\alpha(\gamma) \in [-1, 1]^{\binom{3N}{3}}, \beta(\gamma) \in [-1, 1]^{\binom{3N}{6}}$ be as in [Definition 2.7](#). Then, the maximum advantage with which $\mathcal{F}$ solves 3-FORRELATION is at most

$$\max_{\gamma \in [-1, 1]^{3N}} O \left(\frac{L_{1,3}^{\alpha(\gamma)}(\mathcal{F})}{N} + \frac{L_{1,6}^{\beta(\gamma)}(\mathcal{F})}{N^2} \right).$$

2.2 Quantum Query Complexity

In the setting of quantum query complexity, the input is accessed by an oracle. This oracle is typically an operator $\tilde{O}_x$ for $x \in \{0, 1\}^N$ which maps $|b\rangle |i\rangle \rightarrow |b \oplus x_i\rangle |i\rangle$ for $b \in \{0, 1\}, i \in [N]$. One can alternatively define an oracle O_x for $x \in \{-1, 1\}^N$ which maps $|b\rangle |i\rangle$ to itself if $b = 0$ and to $|b\rangle |i\rangle x_i$ if $b = 1$ and $i \in [N]$. It is not too difficult to show that these two definitions are equivalent, up to a Hadamard gate on the first qubit. We will work with the oracle O_x and later introduce some additional simplifications.

The most general model of a quantum query algorithm is the BQP model defined below. For the following definition, we interpret n as the number of qubits on which the oracle acts and w as the number of qubits of extra workspace. As mentioned before, we use $\mathbf{I}$ to denote the identity matrix, where the dimension is implicit.

Definition 2.9 (BQP Algorithm with d Queries). Let $n, w \in \mathbb{N}$, $N = 2^n, W = 2^w$ and $M = NW$. A BQP algorithm acts on $n + w$ qubits initialized to $|0 \dots 0\rangle$. Let $U_1, \dots, U_{d+1} \in \mathbb{C}^{M \times M}$ be $M \times M$ unitary matrices. The algorithm applies the unitary operators $U_1, \dots, U_{d+1}$ interleaved with the oracle $O_x \otimes \mathbf{I}$ and measures all the qubits at the end to obtain an outcome I_{d+1} . The algorithm accepts iff $I_{d+1} \in \mathcal{F}$ where $\mathcal{F} \subseteq [M]$ is a subset. (See [Figure 5](#) for a depiction.)

The following claim expresses the acceptance probability of a d -query BQP algorithm and is proved in [Appendix A.4](#).

⁵In particular, see equation (5.7) and the equation above in [BS21] for the level-3 contribution and equation (5.13) and the preceding paragraph for the level-6 contribution.

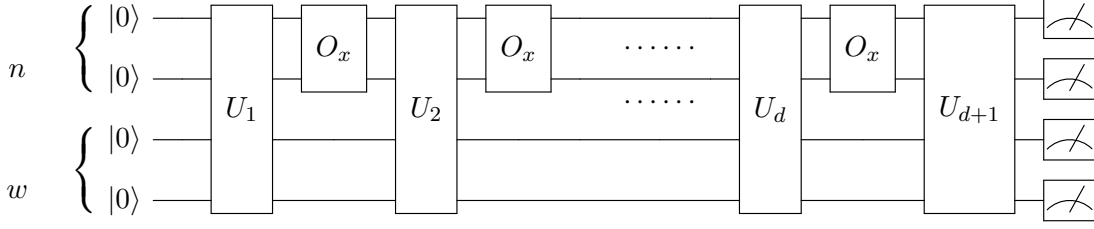


Figure 5: A d -query BQP algorithm.

Claim 2.10. *The acceptance probability of a d -query BQP algorithm can be expressed as*

$$f(x) := \langle I_1 | V_1 \cdot O \cdots O \cdot V_{2d+1} | I_1 \rangle$$

where $O = O_x \otimes \mathbf{I}$, $V_1, \dots, V_{2d+1} \in \mathbb{C}^{M \times M}$ are matrices with $\|V_t\|_{\text{op}} \leq 1$ for all $t \in [d]$ and $|I_1\rangle = |0 \dots 0\rangle$.

In the following sections, we will define DQC_k and $\frac{1}{2}\text{BQP}$ algorithms.

2.3 DQC_k algorithms

We interpret n as the number of qubits on which the oracle acts, k as the number of clean qubits, and w as the number of qubits of extra workspace.

Definition 2.11 (DQC_k Algorithm with d Queries). *Let $n, w, k \in \mathbb{N}$ and $N = 2^n, W = 2^w, K = 2^k$ and $M = NWK$. A DQC_k algorithm acts on k clean qubits initialized to the $|0 \dots 0\rangle$ state and $n + w$ maximally noisy qubits which consist of n qubits on which the oracle acts and w qubits of workspace. Let $U_1, \dots, U_{d+1} \in \mathbb{C}^{M \times M}$ be $M \times M$ unitary matrices. Let $\mathcal{S} = [NW] \times \{1\}$ be the set of all possible starting basis states of the algorithm and $\mathcal{F} \subseteq [NWK]$ be the subset of final basis states that is accepted by the algorithm. The algorithm starts with a uniformly random basis state sampled from $\mathcal{S}$, applies the unitary operators $U_1, \dots, U_{d+1}$, interleaved with the oracle $O_x \otimes \mathbf{I}$, measures all the qubits at the end and accepts if the outcome is in $\mathcal{F}$. (See Figure 1 for a depiction.)*

Remark. In our model, the oracles are not allowed to directly act on the clean qubits, nevertheless, we can effectively implement this type of operation by swapping the clean qubits with the noisy qubits, applying the oracle on those noisy qubits and swapping them back with the clean qubits. While this transformation does require the use of k extra (potentially noisy) qubits to do the swap operation, our formalism has the advantage that we can talk about oracle separations where k , the number of clean qubits is significantly smaller than n , where the length of the input is 2^n . This is important, since when $k \gg n$, many problems become solvable with a few quantum queries with $O(k)$ clean qubits.

We will now provide an expression for the acceptance probability of a DQC_k algorithm, which we will prove in the appendix (Appendix A.4). As mentioned before, estimating the trace of a unitary matrix described by a quantum circuit is known to be complete for the class DQC_1 [KL98] and a similar statement is true in query complexity as well.

Claim 2.12. *The acceptance probability of a d -query a DQC_k algorithm can be expressed as*

$$f(x) = (NW)^{-1} \cdot \text{Tr}(O \cdot V_1 \cdots O \cdot V_{2d})$$

where $O = O_x \otimes \mathbf{I}$, $V_1, \dots, V_{2d} \in \mathbb{C}^{M \times M}$ satisfy $\|V_t\|_{\text{op}} \leq 1$ for $t \in [2d]$, furthermore, $\|V_1\|_{\text{frob}} \leq \sqrt{NW}$.

2.4 $\frac{1}{2}$ BQP algorithms

We interpret n as the number of qubits on which the oracle acts and w as the number of qubits of extra workspace.

Definition 2.13 ($\frac{1}{2}$ BQP Algorithm with d Queries). *Let $n, w \in \mathbb{N}$, $N = 2^n$, $W = 2^w$ and $M = NW$. A $\frac{1}{2}$ BQP algorithm acts on $n + w$ qubits initialized to $|I_1\rangle$ for a uniformly random $I_1 \sim [M]$. The algorithm does not have knowledge of I_1 . Let $U_1, \dots, U_{d+1} \in \mathbb{C}^{M \times M}$ be $M \times M$ unitary matrices. The algorithm applies the unitary operators $U_1, \dots, U_{d+1}$ interleaved with the oracle $O_x \otimes \mathbf{I}$ and measures all the qubits at the end to obtain an outcome I_{d+2} . Finally, the algorithm then learns I_1 . The algorithm accepts iff $(I_1, I_{d+2}) \in \mathcal{F}$ where $\mathcal{F} \subseteq [M] \times [M]$ is a subset. (See [Figure 2](#) for a depiction.)*

We provide an expression for the acceptance probability of a d -query $\frac{1}{2}$ BQP algorithm, which is proved in [Appendix A.4](#).

Claim 2.14. *The acceptance probability of a d -query $\frac{1}{2}$ BQP algorithm can be expressed as*

$$f(x) := M^{-1} \sum_{I_1, I_{d+2} \in [M]} F_{I_1, I_{d+2}} \cdot \langle I_1 | U_1^\dagger \cdot O \cdots O \cdot U_{d+1}^\dagger | I_{d+2} \rangle \langle I_{d+2} | U_{d+1} \cdot O \cdots O \cdot U_1 | I_1 \rangle$$

where $O = O_x \otimes \mathbf{I}$, and $U_1, \dots, U_{d+1} \in \mathbb{C}^{M \times M}$ are matrices with $\|U_t\|_{\text{op}} \leq 1$ for all $t \in [d+1]$.

Some Remarks.

- While our way of defining DQC_k and $\frac{1}{2}$ BQP doesn't clearly subsume BPP, there is a simple way to fix this. We can define variants of these models where the algorithm is allowed to make up to d classical pre-processing queries on clean bits, and based on the query outcomes, choose a d -query quantum algorithm to run. When defined this way, these models immediately subsume BPP, since we can implement any BPP algorithm in the pre-processing part. Interestingly, many of the results in our paper, especially the lower bounds hold even for algorithms with a large amount of classical pre-processing. See [Appendix A.1](#) for more details.
- Unlike [\[GSTW24\]](#), our model does not allow parallel queries. This is without loss of generality, as our model has unrestricted depth and we can simulate k parallel queries by k adaptive queries. If we allow parallel queries but limit the depth, we suspect that it might lead improved Fourier growth bounds in terms of the depth of the algorithm, but we leave this to future work.
- In the rest of this paper, we will work with the oracle O'_x which maps $|i\rangle$ to $|i\rangle x_i$ for all $i \in [N]$ where x is of length N . Note that the aforementioned oracle O_x is the controlled version of O'_x and generally offers more functionality than O'_x . However, in all our proofs, it suffices to work with the oracle O'_x since we allow restrictions $\rho \in \{-1, 1, *\}^N$ to act on our input. In particular, if we consider O'_x for bit-strings of length $2N$ and apply the restriction which fixes the first N coordinates to 1, we obtain the oracle O_x on bit-strings of length N as desired. Since all our Fourier growth bounds work even under restrictions of the input, it suffices to work with oracles of the form O'_x and all our Fourier growth bounds will carry over to oracles of the form O_x if N is replaced by $2N$. Henceforth, we will refer to the oracle O'_x as O_x and work with this oracle.

3 Main Technical Tool: Matrix Decomposition Lemma

The following matrix decomposition lemma is a recurring tool in this paper. It allows us to encode information about the indices in a matrix multiplication by embedding them inside a larger matrix multiplication. In this lemma, we have matrices $U_1, \dots, U_d$ where the rows and columns of U_t are indexed by I_t and I_{t+1} respectively. Here, I is a shorthand for either (i, w, k) or (i, w) where $i \in [N]$ corresponds to indices we want to remember information about and $w \in [W], k \in [K]$ corresponds to auxiliary workspace indices. The set T corresponds to the complement of matrices whose index information we want to retain, i.e., we don't care about the matrices in T . The number $\tilde{N}$ indicates that we do not store parity information for indices i_t with $i_t > \tilde{N}$ and the set S_{d+1} corresponds to the information aggregated after multiplying the matrices.

Lemma 3.1. *Let $U_1, \dots, U_d$ be $M \times M$ matrices with $\|U_t\|_{\text{op}} \leq 1$ for $t \in [d]$ and let $T \subseteq [d]$ and $\tilde{N} \leq N$. Then, there exist matrices $\tilde{U}_1, \dots, \tilde{U}_d$ and $\tilde{U} = \tilde{U}_1 \cdots \tilde{U}_d$ such that for all $I_1, I_{d+1} \in [M], S_{d+1} \subseteq [N]$,*

$$\tilde{U}[I_1|I_{d+1}S_{d+1}] = \sum_{I_2, \dots, I_d \in [M]} \left(\prod_{t \in [1, d]} U_t[I_t|I_{t+1}] \right) \cdot \mathbb{1} \left[S_{d+1} = \bigoplus_{\substack{t \in [2, d] \setminus T \\ i_t \leq \tilde{N}}} \{i_t\} \right].$$

Furthermore, $\max_{t \in [d]} \|\tilde{U}_t\|_{\text{op}} \leq 1$ and $\|\tilde{U}\|_{\text{frob}} \leq \min_{t \in [d]} \|U_t\|_{\text{frob}}$.

Proof of Lemma 3.1. We first describe a function that updates the information we need to remember about the parity of the indices.

Definition 3.2. *Let update_t be the function which for $S \subseteq [N], i_t \in [N], t \in [d]$ satisfies*

$$\text{update}_t(S, i_t) = \begin{cases} S \oplus \{i_t\} & \text{if } t \in [2, d] \setminus T \text{ and } i_t \leq \tilde{N} \\ S & \text{otherwise.} \end{cases}$$

Set $S_1 = \emptyset$. For $t \in [1, d]$, define a matrix $\tilde{U}_t$ with rows and columns indexed by $I_t S_t$ and $I_{t+1} S_{t+1}$ respectively where $I_t, I_{t+1} \in [M], S_t, S_{t+1} \subseteq [N]$ and

$$\tilde{U}_t[I_t S_t | I_{t+1} S_{t+1}] = U_t[I_t | I_{t+1}] \cdot \mathbb{1}[S_{t+1} = \text{update}_t(S_t, i_t)]$$

Let $\tilde{U} := \tilde{U}_{[1, d]} \triangleq \tilde{U}_1 \cdots \tilde{U}_d$. Observe that for any $I_1, I_{d+1} \in [M], S_{d+1} \subseteq [N]$, we have

$$\begin{aligned} \tilde{U}[I_1|I_{d+1}S_{d+1}] &\triangleq \left(\prod_{t=1}^d \tilde{U}_t \right) [I_1|I_d S_d] \\ &= \sum_{\substack{I_2, \dots, I_d \in [M] \\ S_2, \dots, S_d \subseteq [N]}} \prod_{t=1}^d \tilde{U}_t[I_t S_t | I_{t+1} S_{t+1}] \\ &\triangleq \sum_{I_2, \dots, I_d \in [M]} \prod_{t=1}^{d-1} U_t[I_t | I_{t+1}] \cdot \sum_{S_2, \dots, S_d \subseteq [N]} \prod_{t=1}^d \mathbb{1}[S_{t+1} = \text{update}_t(S_t, i_t)] \\ &= \sum_{I_2, \dots, I_d \in [M]} \prod_{t=1}^d U_t[I_t | I_{t+1}] \cdot \mathbb{1} \left[S_{d+1} = \bigoplus_{\substack{t \in [2, d] \setminus T \\ i_t \leq \tilde{N}}} \{i_t\} \right]. \end{aligned}$$

This shows that $\tilde{U}_{[1,d]}$ satisfies the defining equation in [Lemma 3.1](#). In fact, we proved the stronger result that for all $t \in [d]$ and $I_1, I_{t+1} \in [M], S_{t+1} \subseteq [N]$, we have

$$\tilde{U}_{[1,t]}[I_1|I_{t+1}S_{t+1}] = \sum_{I_2, \dots, I_t \in [M]} \left(\prod_{t' \in [1,t]} U_{t'}[I_{t'}|I_{t'+1}] \right) \cdot \mathbb{1} \left[S_{t+1} = \bigoplus_{\substack{t' \in [2,t] \setminus T \\ i_{t'} \leq \tilde{N}}} \{i_{t'}\} \right] \quad (8)$$

where as mentioned before, we use $\tilde{U}_{[1,t]}$ to denote $\tilde{U}_1 \cdots \tilde{U}_t$.

Bound on the spectral norm. We now show that $\|\tilde{U}_t\| \leq 1$ for all $t \in [d]$. This is clearly true for $\tilde{U}_1$, since $\tilde{U}_1$ is a block-diagonal matrix with respect to S_2 and the only non-zero block corresponds to $S_2 = \emptyset$ and is given by U_1 . For any $t \in [2, d]$, consider $\tilde{U}_t$. The rows and columns are indexed by $I_t S_t$ and $I_{t+1} S_{t+1}$ respectively. If $t \in T$, then the matrix is block diagonal with respect to S_t , since $S_{t+1} = S_t$ and each block is a copy of U_t . If $t \notin T$, rearrange the rows $I_t S_t$ into groups according to $\text{update}_t(S_t, i_t)$. Under this rearrangement, the matrix is block diagonal with respect to S_{t+1} since the non-zero entries correspond to $S_{t+1} = \text{update}_t(S_t, i_t)$. We will now show that each block is a sub-matrix of U_t . Fix a block corresponding to S_{t+1} . If $i_t \leq \tilde{N}$ then $S_t = S_{t+1} \oplus \{i_t\}$ and otherwise $S_t = S_{t+1}$, hence, fixing S_{t+1} and i_t uniquely determines S_t . In other words, any row of U_t can appear at most once within a block. Thus, the operator norm of each block is at most 1 by [Fact 2.4](#) and this proves that $\|\tilde{U}_t\| \leq 1$ by [Fact 2.5](#).

Bound on the Frobenius norm. Finally, we bound the Frobenius norm of $\tilde{U}$. Fix any $t \in [d]$. Since $\|\tilde{U}_t\|_{\text{op}} \leq 1$ for all $t \in [d]$, by [Fact 2.2](#), we have

$$\|\tilde{U}\|_{\text{frob}} \triangleq \|\tilde{U}_1 \cdots \tilde{U}_d\|_{\text{frob}} \leq \|\tilde{U}_1 \cdots \tilde{U}_t\|_{\text{frob}} \triangleq \|\tilde{U}_{[1,t]}\|_{\text{frob}}.$$

Ideally, we would have liked to argue that $\|\tilde{U}_t\|_{\text{frob}} \leq \|U_t\|_{\text{frob}}$, but this is not necessarily true. This is because $\tilde{U}_t$ is a matrix with rows indexed by $I_t S_t$ and columns by $I_{t+1} S_{t+1}$ and contains within itself several copies of sub-matrices of $\tilde{U}_t$ across the various possibilities for S_t . We can only guarantee that $\|\tilde{U}_1\| \leq \|U_1\|_{\text{frob}}$, since $\tilde{U}_1$ consists of only one copy of U_1 . To get around this, we will apply [Lemma 3.1](#) in reverse i.e., to the matrices $U_t^T, U_{t-1}^T, \dots, U_1^T$ in this order. We obtain a matrix⁶ $\tilde{U}' := \tilde{U}'_t \cdot \tilde{U}'_{t-1} \cdots \tilde{U}'_1$, such that for all $I_1, I_{t+1} \in [M]$ and $S_1 \subseteq [N]$, we have

$$\begin{aligned} \tilde{U}'[I_{t+1}|I_1 S_1] &= \sum_{I_t, I_{t-1}, \dots, I_2 \in [M]} \left(\prod_{t'=t}^1 U_{t'}^T[I_{t'+1}|I_{t'}] \right) \cdot \mathbb{1} \left[S_1 = \bigoplus_{\substack{t' \in \{t, t-1, \dots, 2\} \setminus T \\ i_{t'} \leq \tilde{N}}} \{i_{t'}\} \right] \\ &= \sum_{I_2, \dots, I_t \in [M]} \left(\prod_{t' \in [1,t]} U_{t'}[I_{t'}|I_{t'+1}] \right) \cdot \mathbb{1} \left[S_1 = \bigoplus_{\substack{t' \in [2,t] \setminus T \\ i_{t'} \leq \tilde{N}}} \{i_{t'}\} \right]. \end{aligned}$$

⁶We remark $\tilde{U}'_t$ is not the transpose of $\tilde{U}_t$. In particular, $\tilde{U}'_t$ will be a matrix with rows indexed by I_{t+1} and columns by $I_t S_t$, whereas $\tilde{U}_t$ is a matrix with rows indexed by $I_t S_t$ and columns by $I_{t+1} S_{t+1}$. Furthermore, $\tilde{U}'_t$ essentially consists of one copy of U_t , while $\tilde{U}_t$ consists of several copies of sub-matrices of U_t for each possible S_t . This distinction turns out to be essential.

$$\triangleq \tilde{U}_{[1,t]}[I_1|I_{t+1}S_1] \quad (\text{by Equation (8)})$$

We observe that the entries of $\tilde{U}'$ and $\tilde{U}_{[1,t]}$ are the same, just arranged differently, hence, their Frobenius norms are equal. Thus,

$$\|\tilde{U}_{[1,t]}\|_{\text{frob}} = \|\tilde{U}'\|_{\text{frob}} \triangleq \|\tilde{U}'_t \cdot \tilde{U}'_{t-1} \cdots \tilde{U}'_1\|_{\text{frob}} \leq \|\tilde{U}'_t\|_{\text{frob}}.$$

We now recall the construction of $\tilde{U}'_t$ from Lemma 3.1 and recall that $\tilde{U}'_t$ is identical to U_t when restricted to columns $S_t = \emptyset$ and zero on the other columns. This implies that $\|\tilde{U}_{[1,t]}\|_{\text{frob}} = \|U_t\|_{\text{frob}}$ and completes the proof. $\square$

3.1 An Improved Matrix Decomposition Lemma

Looking ahead, it turns out that we need a variant of Lemma 3.1, where we have p equality constraints and q memory constraints: for a list of indices $s_1 < t_1, \dots, s_p < t_p \in [2, d]$, we wish to only sum over indices that satisfy $i_{s_j} = i_{t_j}$ for $j \in [p]$, and for the indices $r_1, \dots, r_q$, we wish to retain information about i_{r_j} for $j \in [q]$ until the very end.

Lemma 3.3. *Let $T \subseteq [d]$ and $\tilde{N} \leq N$. Let $p, q \in \mathbb{N} \cup \{0\}$ and let $s_1, t_1, \dots, s_p, t_p \in [2, d]$ with $s_1 < t_1, \dots, s_k < t_k$ and $r_1, \dots, r_q \in [2, d]$. Assume that $s_1, t_1, \dots, s_p, t_p, r_1, \dots, r_q$ are all distinct.*

Let $U_1, \dots, U_d$ be $M \times M$ matrices with $\|U_t\|_{\text{op}} \leq 1$ for all $t \in [d]$. Then, there exist matrices $\tilde{U}_1, \dots, \tilde{U}_d$ and $\tilde{U} = \tilde{U}_1 \cdots \tilde{U}_d$ such that for all $I_1, I_{d+1} \in [M]$, $S_1, S_{d+1} \subseteq [N]$, and $B_{d+1} \in [N]^q$.

$$\begin{aligned} \tilde{U}[I_1 S_1 | I_{d+1} S_{d+1} B_{d+1}] &= \sum_{I_2, \dots, I_d \in [M]} \left(\prod_{t \in [1, d]} U_t[I_t | I_{t+1}] \right) \cdot \mathbb{1} \left[S_{t+1} = S_1 \bigoplus_{\substack{t \in [2, d] \setminus T \\ i_t \leq \tilde{N}}} \{i_t\} \right] \\ &\quad \cdot \mathbb{1} [i_{s_j} = i_{t_j} \text{ for all } j \in [p]] \quad (\text{equality constraints}) \\ &\quad \cdot \mathbb{1} [B_{d+1}(j) = i_{r_j} \text{ for all } j \in [q]]. \quad (\text{memory constraints}) \end{aligned}$$

Furthermore, $\|\tilde{U}_t\|_{\text{op}} \leq 1$ for all $t \in [d]$. Let $\tilde{U}^\emptyset$ be the submatrix of $\tilde{U}$ obtained by taking rows that satisfy $S_1 = \emptyset$. Then, $\|\tilde{U}^\emptyset\|_{\text{frob}} \leq \min_{t \in [d]} \|U_t\|_{\text{frob}}$.

The proof of this lemma is deferred to Appendix A.3. The main ideas behind incorporating the additional constraints is as follows. For memory constraints, the approach is very similar to how we updated the parity information using `update`. We keep a set B_t of all the indices remembered until this point and at time t , we append the index i_t into B_t if it needs to be remembered. For equality constraints, we use another set A_t to store the various $i_{t'}$ for $t' \leq t$ that we have seen until this point and for which we are yet to enforce equality constraints. Suppose at time t , we find that $(i_{t'}, i_{t+1})$ was a pair of equality constraints that we need to impose for some $t' < t + 1$, we use A_t to enforce equality between $i_{t'}$ and i_{t+1} , then remove $i_{t'}$ from A_t to obtain A_{t+1} , and proceed. See Appendix A.3 for more details.

4 Fourier Growth of DQC_k : Proof of Theorem 1.5

Since DQC_k algorithms are a sub-class of BQP algorithms, the bounds from Theorem 1.7 immediately apply to DQC_k algorithms and complete the proof when $\min(2^{k/2}, \sqrt{N}) = \sqrt{N}$. It suffices to handle the other case, i.e., $\min(2^{k/2}, \sqrt{N}) = 2^{k/2}$ which will be the focus of this section.

Throughout this section, to simplify notation, we use the shorthand I_t to denote (i_t, w_t, k_t) where $i_t \in [N], w_t \in [W], k_t \in [K]$ for $N = 2^n, W = 2^w, K = 2^k$. We use $\mathbf{I}$ to denote the identity matrix, where the dimension is implicit.

Let $f(x)$ be the acceptance probability of a DQC_k algorithm and ρ be any restriction of the input variables. We will now derive an expression for the Fourier coefficients of $f|_\rho(x)$. We may assume without loss of generality that the first $\tilde{N}$ coordinates are unfixed and the rest are fixed, by permuting the matrices applied by the quantum algorithm appropriately. Thus, only Fourier coefficients corresponding to $S \subseteq [\tilde{N}]$ are non-zero and are described by the following claim.

Claim 4.1. *Let $f(x)$ be the acceptance probability of a d -query DQC_k algorithm and let $\rho \in \{-1, 1, *\}^N$ be any restriction that leaves the first $\tilde{N}$ coordinates unfixed. Then, there exist matrices $V_1^\rho, \dots, V_{2d}^\rho$ such that for all $S \subseteq [\tilde{N}]$,*

$$\widehat{f|_\rho}(S) = (NW)^{-1} \sum_{I_1, \dots, I_{2d} \in [M]} \left(\prod_{t \in [2d]} V_t^\rho[I_t | I_{t+1}] \right) \cdot \mathbb{1} \left[\bigoplus_{\substack{t \in [2d] \\ \text{with } i_t \leq \tilde{N}}} \{i_t\} = S \right].$$

where $V_1^\rho, \dots, V_{2d}^\rho \in \mathbb{C}^{M \times M}$ satisfy $\|V_t^\rho\|_{\text{op}} \leq 1$ for $t \in [2d]$ and $\|V_1^\rho\|_{\text{frob}} \leq \sqrt{NW}$

The proof of this is fairly simple and is deferred to [Appendix A.5](#).

4.1 Level- ℓ Fourier Growth

In this section, we will establish $L_{1,\ell}$ bounds for DQC_k algorithms for general $\ell \geq 2$ and complete the proof of [Theorem 1.5](#). The goal of this section is to upper bound

$$L_{1,\ell}(f|_\rho) \triangleq \max_{\alpha \in [-1, 1]^{\binom{N}{\ell}}} L_{1,\ell}^\alpha(f|_\rho) = \max_{\alpha \in [-1, 1]^{\binom{N}{\ell}}} \sum_{S \in \binom{[N]}{\ell}} \alpha_S \cdot \widehat{f|_\rho}(S). \quad (9)$$

Fix any $\alpha_S \in [-1, 1]$ for each $S \in \binom{[\tilde{N}]}{\ell}$. From [Equation \(9\)](#) and [Claim 4.1](#), we see that our goal is to upper bound

$$L_{1,\ell}^\alpha(f|_\rho) = \sum_{\substack{S \subseteq [\tilde{N}] \\ |S| = \ell}} (NW)^{-1} \sum_{I_1, \dots, I_{2d} \in [M]} \left(\prod_{t \in [2d]} V_t^\rho[I_t | I_{t+1}] \right) \cdot \mathbb{1} \left[\bigoplus_{\substack{t \in [2d] \\ \text{with } i_t \leq \tilde{N}}} \{i_t\} = S \right] \cdot \alpha_S \quad (10)$$

Observe that if $\bigoplus_{t \in [2d], i_t \leq \tilde{N}} \{i_t\} = S$, then in particular, there must exist a subset $T \subseteq [2d]$ of size ℓ such that $\{i_t : t \in T\}$ is a sequence of ℓ distinct elements in $[\tilde{N}]$ and $\bigoplus_{\substack{t \in [2d] \setminus T \\ \text{with } i_t \leq \tilde{N}}} \{i_t\} = \emptyset$. Conversely, for any T and $\{i_t\}_{t \in T}$ satisfying the above conditions, it defines a unique $S = \{i_t : t \in T\}$. Fix $T \subseteq [2d]$ of size ℓ (this can be done in $\binom{2d}{\ell}$ ways). Let the elements of T be $t_1 < \dots < t_\ell$. Define

$$\begin{aligned} \Delta_T := & \sum_{I_1, \dots, I_{2d} \in [M]} \left(\prod_{t \in [2d]} V_t^\rho[I_t | I_{t+1}] \right) \cdot \mathbb{1} \left[\bigoplus_{\substack{t \in [2d] \setminus T \\ \text{with } i_t \leq \tilde{N}}} \{i_t\} = \emptyset \right] \\ & \cdot \mathbb{1} [i_{t_1}, \dots, i_{t_\ell} \in [\tilde{N}] \text{ are distinct}] \cdot \alpha_{\{i_{t_1}, \dots, i_{t_\ell}\}}. \end{aligned} \quad (11)$$

From the above paragraph, it follows that

$$L_{1,\ell}^\alpha(f|\rho) = (NW)^{-1} \sum_{T \in \binom{[2d]}{\ell}} \Delta_T \leq \binom{2d}{\ell} \cdot (NW)^{-1} \cdot \max_{T \in \binom{[2d]}{\ell}} \Delta_T.$$

We will now show that for all $T \in \binom{[2d]}{\ell}$, we have $\Delta_T \leq M \cdot K^{-1/2} \cdot N^{(\ell-2)/2}$. This, along with the above equation (and the fact that $M = KNW$) would imply that $L_{1,\ell}^\alpha(f|\rho) \leq \sqrt{K} \cdot \binom{2d}{\ell} \cdot N^{(\ell-2)/2}$ as desired. We now show the desired bound of $\Delta_T \leq M \cdot K^{-1/2} \cdot N^{(\ell-2)/2}$.

We will group the terms $t \in [2d]$ into circular intervals $[t_1, t_2), [t_2, t_3)$ and so on until $[t_\ell, t_1)$. Since these intervals cover $[2d]$, 1 must belong to either $[t_1, t_2)$ (this happens when $t_1 = 1$) or $[t_\ell, t_1)$ (this happens when $t_1 > 1$). Assume without loss of generality that $1 \in [t_1, t_2)$, the argument for the other case is similar. We apply [Lemma 3.1](#) to the matrices $V_{t_1}^\rho, \dots, V_{t_{2-1}}^\rho$ in this order (with parameter T) to obtain $\tilde{V}_{[t_1, t_2)}$ such that for all $I_{t_1}, I_{t_2} \in [M], S_{t_2} \subseteq [N]$, we have

$$\tilde{V}_{[t_1, t_2)}[I_{t_1} | I_{t_2} S_{t_2}] = \sum_{I_{t_1+1}, \dots, I_{t_{2-1}} \in [M]} \left(\prod_{t \in [t_1, t_2)} V_t^\rho[I_t | I_{t+1}] \right) \cdot \mathbb{1} \left[S_{t_2} = \bigoplus_{\substack{t \in (t_1, t_2) \\ i_t \leq \tilde{N}}} \{i_t\} \right]. \quad (12)$$

Since we assumed that $1 \in [t_1, t_2)$ and since $\|V_1^\rho\|_{\text{frob}} \leq \sqrt{M/K}$, [Lemma 3.1](#) implies that

$$\|\tilde{V}_{[t_1, t_2)}\|_{\text{frob}} \leq \sqrt{M/K} \quad (13)$$

Define a matrix $\tilde{V}_{[t_2, t_1)}$ so that for all $I_{t_1}, I_{t_2} \in [M], S_{t_2} \subseteq [N]$, we have

$$\begin{aligned} \tilde{V}_{[t_2, t_1)}[I_{t_2} S_{t_2} | I_{t_1}] &= \sum_{I_{t_2+1}, \dots, I_{t_1-1} \in [M]} \left(\prod_{t \in [t_2, t_1)} V_t^\rho[I_t | I_{t+1}] \right) \cdot \mathbb{1} \left[S_{t_2} = \bigoplus_{\substack{t \in (t_2, t_1) \setminus T \\ i_t \leq \tilde{N}}} \{i_t\} \right] \\ &\quad \cdot \mathbb{1} [i_{t_1}, \dots, i_{t_\ell} \in [\tilde{N}] \text{ are distinct}] \cdot \alpha_{\{i_{t_1}, \dots, i_{t_\ell}\}}. \end{aligned} \quad (14)$$

The above equation is well defined since $t_1, \dots, t_\ell$ belong to the circular interval $[t_2, t_1]$. From [Equation \(11\)](#), we have

$$\begin{aligned} \Delta_T &= \sum_{\substack{I_{t_1}, I_{t_2} \in [M] \\ S_{t_2} \subseteq [N]}} \tilde{V}_{[t_1, t_2)}[I_{t_2} S_{t_2} | I_{t_1}] \cdot \tilde{V}_{[t_2, t_1)}[I_{t_1} | I_{t_2} S_{t_2}] \\ &\leq \|\tilde{V}_{[t_1, t_2)}\|_{\text{frob}} \cdot \|\tilde{V}_{[t_2, t_1)}\|_{\text{frob}} \quad (\text{by Fact 2.3}) \\ &\leq \sqrt{M/K} \cdot \|\tilde{V}_{[t_2, t_1)}\|_{\text{frob}} \quad (\text{by Equation (13)}) \end{aligned}$$

We will now control the second term $\|\tilde{V}_{[t_2, t_1)}\|_{\text{frob}}$. Firstly, if $\ell = 2$, then the proof is quite simple. Observe that $\tilde{V}_{[t_2, t_1)}$ is almost identical to the matrix $\tilde{V}'_{[t_2, t_1)}$ that one would get on applying [Lemma 3.3](#) on the matrices $V_{t_2}, \dots, V_{t_1-1}$ backwards, except, we need to multiply by $\mathbb{1} [i_{t_1}, i_{t_2} \in [\tilde{N}] \text{ are distinct}]$ and by a sign $\alpha_{\{i_{t_1}, i_{t_2}\}}$. We have $\|\tilde{V}_{[t_2, t_1)}\|_{\text{frob}} \leq \sqrt{M}$ by [Lemma 3.3](#). Observe that multiplying by the aforementioned terms has the effect of zeroing some entries of $\tilde{V}'$

and multiplying some entries of $\tilde{V}'$ by signs, neither of which increase the Frobenius norm. Hence, $\|\tilde{V}_{[t_2, t_1]}\| \leq \sqrt{M}$ and this would complete the proof for $\ell = 2$.

For levels $\ell > 2$, the argument is more involved since we are multiplying by terms that involve the indices being summed over. To handle this, we require an extra step. We will apply [Lemma 3.3](#) on the matrices $V_{t_3}^\rho, \dots, V_{t_1-1}^\rho$ backwards (with parameter T) with memory constraints corresponding to $t_4, \dots, t_\ell, t_1$ (this is well-defined as $t_4, \dots, t_1 \in [t_3, t_1]$). We obtain a matrix $\tilde{V}'_{[t_3, t_1]}$ such that for all $I_{t_3}, I_{t_1} \in [M]$ and $S_{t_1}, S_{t_3} \subseteq [N]$ and $B_{t_3} \in [\tilde{N}]^{\ell-2}$, we have

$$\begin{aligned} \tilde{V}'_{[t_3, t_1]}[I_{t_3} S_{t_3} B_{t_3} | I_{t_1} S_{t_1}] &= \sum_{I_t \text{ for } t \in (t_3, t_1)} \left(\prod_{t \in [t_3, t_1]} V_t^\rho[I_t | I_{t+1}] \right) \cdot \mathbb{1} \left[S_{t_1} = S_{t_3} \bigoplus_{\substack{t \in (t_3, t_1) \setminus T \\ i_t \leq \tilde{N}}} \{i_t\} \right] \\ &\quad \cdot \mathbb{1} [B_{t_3} = (i_{t_4}, i_{t_5}, \dots, i_{t_\ell}, i_{t_1})], \end{aligned} \quad (15)$$

furthermore,

$$\|\tilde{V}'_{[t_3, t_1]}\|_{\text{op}} \leq 1. \quad (17)$$

Similarly, we will apply [Lemma 3.1](#) on the matrices $V_{t_2}^\rho, \dots, V_{t_3-1}^\rho$ to obtain a matrix $\tilde{V}'_{[t_2, t_3]}$ such that for all $I_{t_2}, I_{t_3} \in [M]$ and $S_{t_3} \subseteq [N]$, we have

$$\tilde{V}'_{[t_2, t_3]}[I_{t_2} | I_{t_3} S_{t_3}] = \sum_{I_t \text{ for } t \in (t_2, t_3)} \left(\prod_{t \in [t_2, t_3]} V_t^\rho[I_t | I_{t+1}] \right) \cdot \mathbb{1} \left[S_{t_3} = \bigoplus_{\substack{t \in (t_2, t_3) \\ i_t \leq \tilde{N}}} \{i_t\} \right],$$

furthermore, $\|\tilde{V}'_{[t_2, t_3]}\|_{\text{frob}} \leq \sqrt{M}$. Define a new matrix $\tilde{V}''_{[t_2, t_3]}$ with rows indexed by I_{t_2} and columns by $I_{t_3} S_{t_3}, B_{t_3}$ such that for all $I_{t_2} \in [M]$ and $I_{t_3} \in [M], S_{t_3} \subseteq [N], B_{t_3} \in [N]^{\ell-2}$, we have

$$\begin{aligned} \tilde{V}''_{[t_2, t_3]}[I_{t_2} | I_{t_3} S_{t_3} B_{t_3}] &:= \tilde{V}'_{[t_2, t_3]}[I_{t_2} | I_{t_3} S_{t_3}] \\ &\quad \cdot \mathbb{1} [\{i_{t_2}, i_{t_3}\} \cup B_{t_3} \text{ has } \ell \text{ distinct elements}] \cdot \alpha_{\{\{i_{t_2}, i_{t_3}\} \cup B_{t_3}\}}. \end{aligned} \quad (18)$$

In other words, the matrix $\tilde{V}''_{[t_2, t_3]}$ consists of $N^{\ell-2}$ blocks corresponding to the various $B_{t_3} \in [N]^{\ell-2}$, and each block is a submatrix of $\tilde{V}'_{[t_2, t_3]}$ with some entries zeroed out and some multiplied by elements in $[-1, 1]$ coming from α . Since $\|\tilde{V}'_{[t_2, t_3]}\|_{\text{frob}} \leq \sqrt{M}$, it follows that

$$\|\tilde{V}''_{[t_2, t_3]}\|_{\text{frob}} \leq \sqrt{M} \cdot \sqrt{N^{\ell-2}}. \quad (19)$$

Finally, we observe that

$$\begin{aligned} (\tilde{V}''_{[t_2, t_3]} \cdot \tilde{V}'_{[t_3, t_1]})[I_{t_2} | I_{t_1} S_{t_1}] &= \sum_{\substack{I_{t_3} \in [M] \\ B_{t_3} \in [N]^{\ell-2}}} \tilde{V}''_{[t_2, t_3]}[I_{t_2} | I_{t_3} S_{t_3} B_{t_3}] \cdot \tilde{V}'_{[t_3, t_1]}[I_{t_3} S_{t_3} B_{t_3} | I_{t_1} S_{t_1}] \\ &= \sum_{I_t \text{ for } t \in (t_2, t_1)} \left(\prod_{t \in [t_2, t_1]} V_t^\rho[I_t | I_{t+1}] \right) \cdot \mathbb{1} \left[S_{t_1} = \bigoplus_{\substack{t \in (t_2, t_1) \\ i_t \leq \tilde{N}}} \{i_t\} \right] \end{aligned}$$

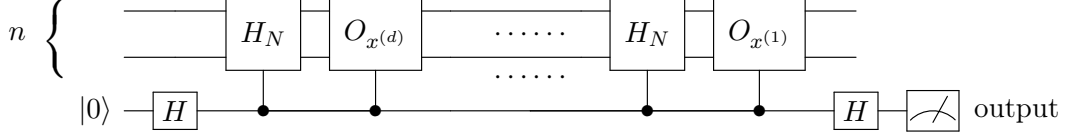


Figure 6: A d -query DQC_1 algorithm with n maximally mixed qubits.

$$\begin{aligned}
& \cdot \mathbb{1} \left[i_{t_1}, \dots, i_{t_\ell} \in [\tilde{N}] \text{ are distinct} \right] \cdot \alpha_{\{i_{t_1}, \dots, i_{t_\ell}\}} \\
& \hspace{15em} \text{(by Equations (16) and (18))} \\
& \triangleq \tilde{V}_{[t_2, t_1]}[I_{t_2} S_{t_1} | I_{t_1}]. \hspace{15em} \text{(by Equation (14))}
\end{aligned}$$

Thus, we see that the entries of $\tilde{V}_{[t_2, t_3]}'' \cdot \tilde{V}_{[t_3, t_1]}'$ and that of $\tilde{V}_{[t_2, t_1]}$ are the same, just arranged differently. Thus, we have

$$\begin{aligned}
\|\tilde{V}_{[t_2, t_1]}\|_{\text{frob}} &= \|\tilde{V}_{[t_2, t_3]}'' \cdot \tilde{V}_{[t_3, t_1]}'\|_{\text{frob}} \\
&\leq \|\tilde{V}_{[t_2, t_3]}''\|_{\text{frob}} \cdot \|\tilde{V}_{[t_3, t_1]}'\|_{\text{op}} \hspace{10em} \text{(by Fact 2.3)} \\
&\leq \sqrt{M} \cdot N^{(\ell-2)/2}. \hspace{10em} \text{(by Equations (17) and (19).)}
\end{aligned}$$

This completes the proof.

4.2 Tightness of our Bounds for DQC_1

In this section, we will show that the dependence on k and N is tight in [Theorem 1.5](#).

Dependence on N . First, we consider the case $k = 1$ and show that DQC_1 algorithms can indeed achieve level- ℓ Fourier growth of roughly $N^{(\ell-2)/2}$, i.e., the dependence on N is tight in [Theorem 1.5](#). We will do so by producing an algorithm on inputs of length dN which makes d oracle queries and whose level- ℓ Fourier growth for $\ell = d$ is $\Omega(N^{(\ell-2)/2})$.

Let H_N be the Hadamard matrix as in [Definition 2.1](#) and view this matrix as an n -qubit unitary operator. For $t \in [d]$, let N_t denote the interval $((t-1)N, tN]$ so that $N_1 \sqcup \dots \sqcup N_d = [dN]$. We view the input $x \in \{0, 1\}^{Nd}$ as comprising of d input strings $x^{(1)}, \dots, x^{(d)}$ of length N each such that $x^{(t)}$ is supported on N_t . Instead of the oracle O_x , we will consider d oracles $O_{x^{(1)}}, \dots, O_{x^{(d)}}$. Consider the d -query DQC_1 algorithm as in [Figure 6](#).⁷ As we saw in [Equation \(1\)](#), is not too difficult to show the bias of this algorithm is precisely

$$f(x) = \frac{1}{2N} \text{Tr}(O_{x^{(1)}} \cdot H_N \cdots O_{x^{(d)}} \cdot H_N).$$

We observe the Fourier coefficients of f correspond to subsets $S \subseteq [Nd]$ that pick exactly one element from each N_t . There are N^d such non-zero Fourier coefficients and they are given by

$$\hat{f}(S) = \frac{1}{2N} \sum_{\substack{i_t \in N_t \\ \text{for } t \in [d]}} (-1)^{\langle i_1, i_2 \rangle + \dots + \langle i_d, i_1 \rangle} \cdot \frac{1}{N^{d/2}} \cdot \mathbb{1}[S = \{i_1, \dots, i_d\}].$$

⁷Typically, we express DQC_1 in terms of a single oracle O_x , as opposed to d smaller oracles $O_{x^{(1)}}, \dots, O_{x^{(d)}}$, nevertheless, it is easy to embed the circuit in [Figure 6](#) into a larger one consisting only of O_x oracle calls for $x = (x^{(1)}, \dots, x^{(d)})$ by applying the following sequence of operators d times: $H_N \otimes \mathbf{I}$, followed O_x , followed by the permutation matrix Π that maps $|i\rangle \rightarrow |i - N \pmod{Nd}\rangle$ for all computational basis states $i \in [Nd]$.

Each such S uniquely identifies $i_1 \in N_1, \dots, i_d \in N_d$ and we set $\alpha_S := (-1)^{\langle i_1, i_2 \rangle + \dots + \langle i_d, i_1 \rangle}$. Thus, we obtain that the level- d Fourier growth is at least

$$N^d \cdot \frac{1}{2N} \cdot \frac{1}{N^{d/2}} \geq \Omega\left(N^{(d-2)/2}\right).$$

This completes the proof.

Dependence on k . It is clear to see that a DQC_k algorithm can solve the Forrelation problem on inputs of length 2^k , since we can run the k -qubit Forrelation circuit on the clean qubits. As the Forrelation function on 2^k -bit inputs has level-two Fourier growth of $2^{k/2}$, this saturates the bound from [Theorem 1.5](#) for level two.

5 Fourier Growth of $\frac{1}{2}$ BQP: Proof of [Theorem 1.6](#)

In this section, we will show Fourier growth bounds on $\frac{1}{2}$ BQP algorithms. The level-3 bound uses the basic matrix decomposition lemma ([Lemma 3.1](#)) from earlier and is presented in [Section 5.1](#). Since the level-6 bound is more involved, it requires the improved matrix decomposition lemma ([Lemma 3.3](#)) and is presented in [Section 5.2](#).

Throughout this section, to simplify notation, we use the shorthand I_1 to denote (i_1, w_1) where $i \in [N], w \in [W]$ for $N = 2^n, W = 2^w$. We use $\mathbf{I}$ to denote the identity matrix, where the dimension is implicit.

Given the expression for the acceptance probability of a d -query $\frac{1}{2}$ BQP algorithm ([Claim 2.14](#)), it is not too difficult to derive an expression for the Fourier coefficients under any restriction – this part is similar to the proof of [Claim 4.1](#) from [Claim 2.12](#). We obtain the following claim, whose proof is deferred to [Appendix A.5](#). As before, we can assume without loss of generality that the restriction ρ fixes all but the first $\tilde{N}$ coordinates for some $\tilde{N} \leq N$, by permuting the matrices applied by the quantum algorithm appropriately.

Claim 5.1. *Let $f(x)$ be the acceptance probability of a d -query $\frac{1}{2}$ BQP algorithm and $\rho \in \{-1, 1, *\}^N$ be any restriction that leaves the first $\tilde{N}$ coordinates free and fixes the rest. Then, there exist matrices $V_1^\rho, \dots, V_{2d+2}^\rho \in \mathbb{C}^M$ such that for all $S \subseteq [\tilde{N}]$,*

$$\widehat{f|_\rho}(S) = M^{-1} \sum_{I_1, I_{d+2} \in [M]} \sum_{\substack{I_2, \dots, I_{d+1} \in [M] \\ I_{d+3}, \dots, I_{2d+2} \in [M]}} F_{I_1, I_{d+2}} \cdot \prod_{t \in [2d+2]} V_t^\rho[I_t | I_{t+1}] \cdot \mathbb{1} \left[\bigoplus_{\substack{t \in [2d+2] \setminus \{1, d+2\} \\ i_t \leq \tilde{N}}} \{i_t\} = S \right] \quad (20)$$

where $\|V_t^\rho\|_{\text{op}} \leq 1$ for all $t \in [2d+2]$.

Now that we have an expression for the Fourier coefficients, we turn our attention to proving Fourier growth bounds.

5.1 Level-3 Fourier Growth

As mentioned before, we will only be able to bound $L_{1,3}^{\alpha(\gamma)}(f|_\rho)$ where $\gamma \in [-1, 1]^{3N}$ and $\alpha(\gamma)$ is as in [Definition 2.7](#). Fix any such $\alpha(\gamma)$. As before, if $\bigoplus_{\substack{t \in [2d+2] \setminus \{1, d+2\} \\ i_t \leq \tilde{N}}} \{i_t\} = S$, then there exist

distinct $t_1, t_2, t_3 \in [2d+2] \setminus \{1, d+2\}$ such that

$$i_{t_1}, i_{t_2}, i_{t_3} \in [\tilde{N}] \text{ are distinct and } \bigoplus_{\substack{t \in [2d+2] \setminus \{1, d+2, t_1, t_2, t_3\} \\ i_t \leq \tilde{N}}} \{i_t\} = \emptyset,$$

conversely, any $t_1, t_2, t_3, i_{t_1}, i_{t_2}, i_{t_3}$ satisfying the above condition defines a unique S , up to a permutation of the t_1, t_2, t_3 . There are at most $O(d^3)$ possibilities for distinct $t_1, t_2, t_3 \in [2d+2] \setminus \{1, d+2\}$. Fix any such t_1, t_2, t_3 . We now recall [Definition 2.7](#). Let $\tilde{A} = A \cap [\tilde{N}]$, $\tilde{B} = B \cap [\tilde{N}]$, $\tilde{C} = C \cap [\tilde{N}]$. For any $i_{t_1}, i_{t_2}, i_{t_3} \in [\tilde{N}]$, for $\alpha(\gamma)_{i_{t_1}, i_{t_2}, i_{t_3}}$ to be non-zero, one of $\{i_{t_1}, i_{t_2}, i_{t_3}\}$ must lie in $\tilde{A}, \tilde{B}, \tilde{C}$ each. Without loss of generality, $i_{t_1} \in \tilde{A}, i_{t_2} \in \tilde{B}, i_{t_3} \in \tilde{C}$. Now, S uniquely identifies $i_{t_1}, i_{t_2}, i_{t_3}$.

Fix any $i_{t_2}^* \in \tilde{B}$. (There are at most N possibilities for such $i_{t_2}^*$.) Define

$$\begin{aligned} \Delta_{t_1, t_2, t_3, i_{t_2}^*}^\gamma &:= \sum_{\substack{I_{t_1} \in \tilde{A} \times [W] \\ I_{t_2} \in \{i_{t_2}^*\} \times [W] \\ I_{t_3} \in \tilde{C} \times [W]}} \sum_{\substack{I_t \in [M] \text{ for } t \text{ in} \\ [2d+2] \setminus \{t_1, t_2, t_3\}}} F_{I_1, I_{d+2}} \cdot \prod_{t \in [2d+2]} V_t^\rho[I_t | I_{t+1}] \\ &\quad \cdot \mathbb{1} \left[\bigoplus_{\substack{t \in [2d+2] \setminus \{1, d+2, t_1, t_2, t_3\} \\ \text{with } i_t \leq \tilde{N}}} \{i_t\} = \emptyset \right] \cdot \alpha(\gamma)_{i_{t_1}, i_{t_2}^*, i_{t_3}}. \end{aligned} \tag{21}$$

Substituting this in the expression for the Fourier growth, we have

$$\begin{aligned} L_{1,3}^{\alpha(\gamma)}(f|_\rho) &\triangleq \sum_{|S|=3} \alpha(\gamma)_S \cdot \widehat{f|_\rho}(S) \\ &= M^{-1} \sum_{\substack{\text{distinct } t_1, t_2, t_3 \\ \text{in } [2d+2] \setminus \{1, d+2\}, \\ i_{t_2}^* \in \tilde{B}}} \Delta_{t_1, t_2, t_3, i_{t_2}^*}^\gamma \quad (\text{from } \text{Equation (21)} \text{ and } \text{Claim 5.1}) \\ &\leq M^{-1} \cdot O(d^3 N) \cdot \max_{\substack{\text{distinct } t_1, t_2, t_3 \\ \text{in } [2d+2] \setminus \{1, d+2\}, \\ i_{t_2}^* \in \tilde{B}}} \Delta_{t_1, t_2, t_3, i_{t_2}^*}^\gamma. \end{aligned}$$

We will show that for each distinct $t_1, t_2, t_3 \in [2d+2] \setminus \{1, d+2\}$ and $i_{t_2}^* \in \tilde{B}$, we have $\Delta_{t_1, t_2, t_3, i_{t_2}^*}^\gamma \leq \sqrt{MW}$. Substituting this above, we would get

$$L_{1,3}^{\alpha(\gamma)}(f|_\rho) \leq M^{-1} \cdot O(d^3 N) \cdot \sqrt{MW} = O(d^3) \cdot \sqrt{N} \cdot \sqrt{NMW} \cdot M^{-1} \leq O(d^3) \cdot \sqrt{N},$$

where we used the fact that $M = NW$. It now suffices prove the bound $\Delta_{t_1, t_2, t_3, i_{t_2}^*}^\gamma \leq \sqrt{MW}$.

Fix any distinct $t_1, t_2, t_3 \in [2d+2] \setminus \{1, d+2\}$ and $i_{t_2}^* \in \tilde{B}$. We will now use [Definition 2.7](#) to get

$$\alpha(\gamma)_{i_{t_1}, i_{t_2}, i_{t_3}} = \overline{H}(i_{t_1}, i_{t_2}^*) \cdot \overline{H}(i_{t_3}, i_{t_2}^*) \cdot \gamma_{i_{t_1}} \cdot \gamma_{i_{t_2}^*} \cdot \gamma_{i_{t_3}}$$

We will use this to encode the action of multiplication by $\alpha(\gamma)_{i_{t_1}, i_{t_2}^*, i_{t_3}}$ using a matrix product with diagonal matrices. Define matrices P_t as follows. Firstly, for $t \in [2d+2] \setminus \{t_1, t_2, t_3\}$, we have $P_t = V_t^\rho$. Let $D_{t_1}, D_{t_2}, D_{t_3}$ be $M \times M$ diagonal matrices with $[-1, 1]$ -valued entries defined as follows. For $I_{t_1}, I_{t_2}, I_{t_3} \in [M]$, let

$$D_{t_1}[I_{t_1} | I_{t_1}] = \begin{cases} \gamma_{i_{t_1}} \cdot \overline{H}(i_{t_1}, i_{t_2}^*) & \text{if } i_{t_1} \in \tilde{A} \\ 0 & \text{otherwise.} \end{cases} \tag{22}$$

$$D_{t_2}[I_{t_2}|I_{t_2}] = \begin{cases} \gamma_{i_{t_2}^*} & \text{if } i_{t_2} = i_{t_2}^* \\ 0 & \text{otherwise.} \end{cases} \quad (23)$$

$$D_{t_3}[I_{t_3}|I_{t_3}] = \begin{cases} \gamma_{i_{t_3}} \cdot \overline{H}(i_{t_3}, i_{t_2}^*) & \text{if } i_{t_3} \in \tilde{C} \\ 0 & \text{otherwise.} \end{cases} \quad (24)$$

Let $P_{t_1} := D_{t_1} \cdot V_{t_1}^\rho$, $P_{t_2} := D_{t_2} \cdot V_{t_2}^\rho$, and $P_{t_3} := D_{t_3} \cdot V_{t_3}^\rho$. Firstly, observe that

$$\|P_t\|_{\text{op}} \leq 1 \text{ for all } t \in [d], \quad (25)$$

since $\gamma \in [-1, 1]^{3N}$ and $\|D_t\|_{\text{op}}, \|V_t^\rho\|_{\text{op}} \leq 1$. Secondly, observe that

$$\|P_{t_2}\|_{\text{frob}} \leq \sqrt{W}, \quad (26)$$

since $P_{t_2} = D_{t_2} \cdot V_{t_2}^\rho$ and multiplying by the matrix D_{t_2} has the effect of zeroing out all but W rows (only the rows indexed by $i_{t_2}^*$ survive), and each row of V_{t_2} has norm at most one. This construction allows us to simplify Equation (21) as

$$\Delta_{t_1, t_2, t_3, i_{t_2}^*}^\gamma = \sum_{I_1, \dots, I_{2d+2} \in [M]} F_{I_1, I_{d+2}} \cdot \prod_{t \in [2d+2]} P_t[I_t|I_{t+1}] \cdot \mathbb{1} \left[\bigoplus_{\substack{t \in [2d+2] \setminus \{1, d+2, t_1, t_2, t_3\} \\ \text{with } i_t \leq \tilde{N}}} \{i_t\} = \emptyset \right]. \quad (27)$$

We now break up this summation into terms $I_1, \dots, I_{d+2}$ and $I_{d+3}, \dots, I_{2d+2}$. We apply Lemma 3.1 to the matrices $P_1, \dots, P_{d+1}$ in this order with $T = \{t_1, t_2, t_3\}$ to obtain a matrix $\tilde{P}_{[1, d+1]}$ and to the matrices $P_{d+2}, \dots, P_{2d+2}$ in reverse order with $T = \{t_1, t_2, t_3\}$ to obtain a matrix $\tilde{P}_{[d+2, 2d+2]}$ such that for all $I_1, I_{d+2} \in [N]$, $S_{d+2} \subseteq [N]$, we have

$$\tilde{P}_{[1, d+1]}[I_1|I_{d+2}S_{d+2}] = \sum_{I_2, \dots, I_{d+1} \in [M]} \left(\prod_{t \in [1, d+1]} P_t[I_t|I_{t+1}] \right) \cdot \mathbb{1} \left[S_{d+2} = \bigoplus_{\substack{t \in [2, d+1] \setminus \{t_1, t_2, t_3\} \\ \text{with } i_t \leq \tilde{N}}} \{i_t\} \right] \quad (28)$$

$$\tilde{P}_{[d+2, 2d+2]}[I_{d+2}S_{d+2}|I_1] = \sum_{I_{d+3}, \dots, I_{2d+2} \in [M]} \left(\prod_{t \in [d+2, 2d+2]} P_t[I_t|I_{t+1}] \right) \cdot \mathbb{1} \left[S_{d+2} = \bigoplus_{\substack{t \in [d+3, 2d+2] \setminus \{t_1, t_2, t_3\} \\ \text{with } i_t \leq \tilde{N}}} \{i_t\} \right] \quad (29)$$

$$\|\tilde{P}_{[1, d+1]}\|_{\text{frob}} \leq \min_{t \in [1, d+1]} \|P_t\|_{\text{frob}} \quad \text{and} \quad \|\tilde{P}_{[d+2, 2d+2]}\|_{\text{frob}} \leq \min_{t \in [d+2, 2d+2]} \|P_t\|_{\text{frob}}. \quad (30)$$

Observe that $[2, d+1] \cup [d+3, 2d+2] = [2d+2] \setminus \{1, d+2\}$. Plugging in Equations (28) and (29) into Equation (27), we have

$$\begin{aligned} \Delta_{t_1, t_2, t_3, i_{t_2}^*}^\gamma &= \sum_{\substack{I_1, I_{d+2} \in [M] \\ S_{d+2} \subseteq [N]}} F_{I_1, I_{d+2}} \cdot \tilde{P}_{[1, d+1]}[I_1|I_{d+2}S_{d+2}] \cdot \tilde{P}_{[d+2, 2d+2]}[I_{d+2}S_{d+2}|I_1] \\ &\leq \sum_{\substack{I_1, I_{d+2} \in [M] \\ S_{d+2} \subseteq [N]}} \left| \tilde{P}_{[1, d+1]}[I_1|I_{d+2}S_{d+2}] \right| \cdot \left| \tilde{P}_{[d+2, 2d+2]}[I_{d+2}S_{d+2}|I_1] \right| \quad (\text{since } F_{I_1, I_{d+2}} \in \{0, 1\}) \end{aligned}$$

$$\leq \|\tilde{P}_{[1,d+1]}\|_{\text{frob}} \cdot \|\tilde{P}_{[d+2,2d+2]}\|_{\text{frob}}. \quad (\text{by Fact 2.3})$$

As before, it is easy to see that

$$\max \left(\|\tilde{P}_{[1,d+1]}\|_{\text{frob}}, \|\tilde{P}_{[d+2,2d+2]}\|_{\text{frob}} \right) \leq \sqrt{M}.$$

since these matrices have operator norm at most 1 (due to Equation (25)) and have either at most M rows or at most M columns. This already tells us that $\Delta_{t_1, t_2, t_3, i_{t_2}^*}^\gamma \leq M$. We will now derive the improved bound of $\Delta_{t_1, t_2, t_3, i_{t_2}^*}^\gamma \leq \sqrt{MW}$ by showing that

$$\min \left(\|\tilde{P}_{[1,d+1]}\|_{\text{frob}}, \|\tilde{P}_{[d+2,2d+2]}\|_{\text{frob}} \right) \leq \sqrt{W}.$$

Since $[1, d+1] \cup [d+2, 2d+2] = [2d+2]$, t_2 must belong to either $[1, d+1]$ or $[d+2, 2d+2]$. Assume without loss of generality that $t_2 \in [1, d+1]$, the analysis for the other case is similar. Recall that $\tilde{P}_{[1,d+1]}$ was obtained by applying Lemma 3.1 on the matrices $P_1, \dots, P_{d+1}$. Since t_2 appears in $[1, d+1]$, Lemma 3.1 along with Equation (26) implies that

$$\|\tilde{P}_{[1,d+1]}\|_{\text{frob}} \leq \|P_{t_2}\|_{\text{frob}} \leq \sqrt{W}.$$

This completes the proof for level 3. Next, we will prove the level-6 Fourier growth bound.

5.2 Level-6 Fourier Growth

Let $\gamma \in [-1, 1]^{3N}$ and $\beta(\gamma)$ be as in Definition 2.7. We wish to upper bound

$$L_{1,6}^{\beta(\gamma)}(f|_\rho) = \sum_{|S|=6} \beta(\gamma)_S \cdot \widehat{f|_\rho}(S). \quad (31)$$

Recall from Equation (20) in Claim 5.1 that the only level-6 non-zero Fourier coefficients correspond to $S \subseteq \binom{[\tilde{N}]}{6}$ and are given by

$$\widehat{f|_\rho}(S) = M^{-1} \sum_{I_1, I_{d+2} \in [M]} \sum_{\substack{I_2, \dots, I_{d+1} \in [M] \\ I_{d+3}, \dots, I_{2d+2} \in [M]}} F_{I_1, I_{d+2}} \cdot \prod_{t \in [2d+2]} P_t[I_t | I_{t+1}] \cdot \mathbb{1} \left[\bigoplus_{\substack{t \in [2d+2] \setminus \{1, d+2\} \\ i_t \leq \tilde{N}}} \{i_t\} = S \right]. \quad (32)$$

Let $S \subseteq \binom{[\tilde{N}]}{6}$. As before, if $\bigoplus_{\substack{t \in [2d+2] \setminus \{1, d+2\} \\ i_t \leq \tilde{N}}} i_t = S$, then there exist six distinct $t_1, \dots, t_6 \in [2d+2] \setminus \{1, d+2\}$ such that

$$i_{t_1}, \dots, i_{t_6} \in [\tilde{N}] \text{ are distinct and } \bigoplus_{\substack{t \in [2d+2] \setminus \{1, d+2, t_1, \dots, t_6\} \\ i_t \leq \tilde{N}}} \{i_t\} = \emptyset,$$

conversely, any $t_1, \dots, t_6, i_{t_1}, \dots, i_{t_6}$ satisfying the above condition defines a unique S , up to a permutation of the t_i 's. There are at most $O(d^6)$ possibilities for distinct $t_1, \dots, t_6 \in [2d+2] \setminus \{1, d+2\}$. Fix any such $t_1, \dots, t_6$. Let $\tilde{A} = A \cap [\tilde{N}]$, $\tilde{B} = B \cap [\tilde{N}]$, $\tilde{C} = C \cap [\tilde{N}]$ as before. For any $i_{t_1}, \dots, i_{t_6} \in [\tilde{N}]$, for $\beta(\gamma)_{i_{t_1}, \dots, i_{t_6}}$ to be non-zero, we must have two of $\{i_{t_1}, \dots, i_{t_6}\}$ must lie

in $\tilde{A}, \tilde{B}, \tilde{C}$ each. Without loss of generality, $i_{t_1} \neq i_{t_4} \in \tilde{A}, i_{t_2} \neq i_{t_5} \in \tilde{B}, i_{t_3} \neq i_{t_6} \in \tilde{C}$. With this notation, S uniquely identifies $i_{t_1}, \dots, i_{t_6}$ up to the following swaps $i_{t_1} \leftrightarrow i_{t_4}, i_{t_2} \leftrightarrow i_{t_5}, i_{t_3} \leftrightarrow i_{t_6}$.

Fix any $i_{t_2}^* \neq i_{t_5}^* \in B$. (There are at most N^2 possibilities for $(i_{t_2}^*, i_{t_5}^*)$.) Define

$$\begin{aligned} \Delta_{t_1, \dots, t_6, i_{t_2}^*, i_{t_5}^*}^\gamma := & \sum_{\substack{I_{t_1}, I_{t_4} \in \tilde{A} \times [W] \\ I_{t_2} \in \{i_{t_2}^*\} \times [W] \\ I_{t_5} \in \{i_{t_5}^*\} \times [W] \\ I_{t_3}, I_{t_6} \in \tilde{C} \times [W]}} \beta(\gamma)_{i_{t_1}, i_{t_2}^*, i_{t_3}, i_{t_4}, i_{t_5}^*, i_{t_6}} \sum_{\substack{I_t \in [M] \text{ for } t \text{ in} \\ [2d+2] \setminus \{t_1, \dots, t_6\}}} F_{I_1, I_{d+2}} \cdot \prod_{t \in [2d+2]} V_t^\rho[I_t | I_{t+1}] \\ & \cdot \mathbb{1} \left[\bigoplus_{\substack{t \in [2d+2] \setminus \{1, d+2, t_1, \dots, t_6\} \\ \text{with } i_t \leq \tilde{N}}} \{i_t\} = \emptyset \right] \cdot \mathbb{1}[i_{t_1} \neq i_{t_4} \text{ and } i_{t_3} \neq i_{t_6}]. \end{aligned} \quad (33)$$

Substituting Equations (32) and (33) in Equation (31), we have

$$\begin{aligned} L_{1,6}^{\beta(\gamma)}(f|_\rho) & \leq M^{-1} \cdot O(1) \sum_{\substack{\text{distinct } t_1, \dots, t_6 \\ \text{in } [2d+2] \setminus \{1, d+2\}, \\ i_{t_2}^* \neq i_{t_5}^* \in \tilde{B}}} \Delta_{t_1, \dots, t_6, i_{t_2}^*, i_{t_5}^*}^\gamma \\ & \leq M^{-1} \cdot O(d^6 N^2) \max_{\substack{\text{distinct } t_1, \dots, t_6 \\ \text{in } [2d+2] \setminus \{1, d+2\}, \\ i_{t_2}^* \neq i_{t_5}^* \in \tilde{B}}} \Delta_{t_1, \dots, t_6, i_{t_2}^*, i_{t_5}^*}^\gamma. \end{aligned}$$

We will now show that $\Delta_{t_1, \dots, t_6, i_{t_2}^*, i_{t_5}^*}^\gamma \leq O(\sqrt{MW})$ for all distinct $t_1, \dots, t_6 \in [2d+2] \setminus \{1, d+2\}$ and $i_{t_2}^* \neq i_{t_5}^* \in B$. Substituting this in the above would imply that

$$L_{1,6}^{\beta(\gamma)}(f|_\rho) \leq O(d^6) \cdot N^2 \cdot \sqrt{MW} \cdot M^{-1} \leq O(d^6) \cdot N^{1.5}$$

where we used the fact that $M = NW$.

Fix any distinct $t_1, \dots, t_6 \in [2d+2] \setminus \{1, d+2\}$ and $i_{t_2}^* \neq i_{t_5}^* \in \tilde{B}$. We now turn our attention to proving that $\Delta_{t_1, \dots, t_6, i_{t_2}^*, i_{t_5}^*}^\gamma \leq O(\sqrt{MW})$. The first natural attempt is to apply Lemma 3.1 to express $\Delta_{t_1, \dots, t_6, i_{t_2}^*, i_{t_5}^*}^\gamma$ as a matrix product. Recall from Definition 2.7 that

$$\beta(\gamma)_{i_{t_1}, i_{t_2}^*, i_{t_3}, i_{t_4}, i_{t_5}^*, i_{t_6}} = \alpha(\gamma)_{\{i_{t_1}, i_{t_2}^*, i_{t_3}\}} \cdot \alpha(\gamma)_{\{i_{t_4}, i_{t_5}^*, i_{t_6}\}} \cdot \mathbb{1}[i_{t_1} \neq i_{t_4} \text{ and } i_{t_3} \neq i_{t_6}].$$

Inspired by the level-3 approach, let us define matrices P_t^ρ as follows. Firstly, for $t \notin \{t_1, \dots, t_6\}$, we have $P_t^\rho = V_t^\rho$. Let $D_{t_1}, D_{t_2}, D_{t_3}$ be diagonal matrices as earlier in Equations (22) to (24) and let $D_{t_4}, D_{t_5}, D_{t_6}$ be diagonal matrices defined similarly to in Equations (22) to (24) but we change the indices to I_{t_1} to I_{t_4}, I_{t_2} to $I_{t_5}, i_{t_2}^*$ to $i_{t_5}^*$, and I_{t_3} to I_{t_6} . Similarly to Equations (25) and (26), we have

$$\|P_t\|_{\text{op}} \leq 1 \text{ for all } t \in [2d+2] \quad \text{and} \quad \|P_{t_2}\|_{\text{frob}}, \|P_{t_5}\|_{\text{frob}} \leq \sqrt{W}, \quad (34)$$

and, we can simplify the expression for $\Delta_{t_1, \dots, t_6, i_{t_2}^*, i_{t_5}^*}^\gamma$ from Equation (33) as follows.

$$\begin{aligned} \Delta_{t_1, \dots, t_6, i_{t_2}^*, i_{t_5}^*}^\gamma &= \sum_{I_1, \dots, I_{2d+2} \in [M]} F_{I_1, I_{d+2}} \cdot \prod_{t \in [2d+2]} P_t[I_t | I_{t+1}] \\ &\quad \cdot \mathbb{1} \left[\bigoplus_{\substack{t \in [2d+2] \setminus \{1, d+2, t_1, \dots, t_6\} \\ \text{with } i_t \leq \tilde{N}}} \{i_t\} = \emptyset \right] \cdot \mathbb{1}[i_{t_1} \neq i_{t_4} \text{ and } i_{t_3} \neq i_{t_6}]. \end{aligned} \quad (35)$$

The quantity $\Delta_{t_1, \dots, t_6, i_{t_2}^*, i_{t_5}^*}^\gamma$ in Equation (35) is thus captured by a matrix product of the P_t 's and we would like to apply Lemma 3.1 as before, but the issue is that we need to enforce the constraints that $i_{t_1} \neq i_{t_4}$ and $i_{t_3} \neq i_{t_6}$. As such, Lemma 3.1 is unable to enforce non-equality constraints between the indices being summed over. However, it turns out that a variant of this lemma can enforce *equality* constraints between indices. Inspired by this, we use the Inclusion-Exclusion principle to get

$$\mathbb{1}[i_{t_1} \neq i_{t_4} \text{ and } i_{t_3} \neq i_{t_6}] = 1 - \mathbb{1}[i_{t_1} = i_{t_4}] - \mathbb{1}[i_{t_3} = i_{t_6}] + \mathbb{1}[i_{t_1} = i_{t_4} \text{ and } i_{t_3} = i_{t_6}] \quad (36)$$

For $a \in [4]$, we now define $\Delta_{t_1, \dots, t_6, i_{t_2}^*, i_{t_5}^*}^{\gamma, a}$ to be identical to $\Delta_{t_1, \dots, t_6, i_{t_2}^*, i_{t_5}^*}^\gamma$ as in Equation (35), but we

$$\text{replace the term } \mathbb{1}[i_{t_1} \neq i_{t_4} \text{ and } i_{t_3} \neq i_{t_6}] \text{ by } \begin{cases} 1 & \text{if } a = 1 \\ \mathbb{1}[i_{t_1} = i_{t_4}] & \text{if } a = 2 \\ \mathbb{1}[i_{t_3} = i_{t_6}] & \text{if } a = 3 \\ \mathbb{1}[i_{t_1} = i_{t_4} \text{ and } i_{t_3} = i_{t_6}] & \text{if } a = 4. \end{cases} \quad (37)$$

From the inclusion-exclusion principle as in Equation (36), we see that

$$\Delta_{t_1, \dots, t_6, i_{t_2}^*, i_{t_5}^*}^\gamma = \sum_{a \in [4]} \Delta_{t_1, \dots, t_6, i_{t_2}^*, i_{t_5}^*}^{\gamma, a} \cdot (-1)^{\mathbb{1}[a \in \{2, 3\}]} \leq \sum_{a \in [4]} \left| \Delta_{t_1, \dots, t_6, i_{t_2}^*, i_{t_5}^*}^{\gamma, a} \right|.$$

Thus, for each $a \in [4]$, it suffices to show that

$$\left| \Delta_{t_1, \dots, t_6, i_{t_2}^*, i_{t_5}^*}^{\gamma, a} \right| \leq \sqrt{MW}.$$

Ideally, we would like to use a decomposition similar to the one for the level-3 case but there is a key difference, namely the equality constraints imposed by a . This is where Lemma 3.3 comes into play. The main idea is to apply Lemma 3.3 for each $a \in [4]$ with appropriate constraints to the matrices $P_1, \dots, P_{d+1}$ in this order to obtain $\tilde{P}_{[1, d+1]}^a$ and to $P_{d+2}, \dots, P_{2d+2}$ in reverse order to obtain $\tilde{P}_{[d+2, 2d+2]}^a$ such that

$$\Delta_{t_1, \dots, t_6, i_{t_2}^*, i_{t_5}^*}^{\gamma, a} = \sum_{I_1 I_{d+2} \in [M]} F_{I_1, I_{d+2}} \sum_{\substack{S_{d+2} \subseteq [N] \\ B_{d+2} \in [N]^*}} \tilde{P}_{[1, d+1]}^a[I_1 | I_{d+2} S_{d+2} B_{d+2}] \cdot \tilde{P}_{[d+2, 2d+2]}^a[I_{d+2} S_{d+2} B_{d+2} | I_1]. \quad (38)$$

Imagine for now that we are able to do this. From here on, the proof is identical the level-3 approach. In more detail, from Equation (38), we get

$$\left| \Delta_{t_1, \dots, t_6, i_{t_2}^*, i_{t_5}^*}^{\gamma, a} \right| \leq \|\tilde{P}_{[1, d+1]}^a\|_{\text{frob}} \cdot \|\tilde{P}_{[d+2, 2d+2]}^a\|_{\text{frob}}.$$

As before, we have

$$\max \left(\|\tilde{P}_{[1,d+1]}^a\|_{\text{frob}}, \|\tilde{P}_{[d+2,2d+2]}^a\|_{\text{frob}} \right) \leq \sqrt{M}$$

since these matrices have operator norm at most 1 and have either at most M rows or at most M columns. Furthermore, from [Equation \(34\)](#) and [Lemma 3.3](#), since t_2 must belong to either $[1, d+1]$ or $[d+2, 2d+2]$, we have

$$\min \left(\|\tilde{P}_{[1,d+1]}^a\|_{\text{frob}}, \|\tilde{P}_{[d+2,2d+2]}^a\|_{\text{frob}} \right) \leq \sqrt{W}.$$

Altogether, we'd get $\left| \Delta_{t_1, \dots, t_6, i_{t_2}^*, i_{t_5}^*}^{\gamma, a} \right| \leq \sqrt{MW}$ as desired and this would complete the proof.

We will now show how to set the parameters and enforce constraints so that [Equation \(38\)](#) is satisfied. We start with $p = p' = 0, q = q' = 0$. Set $T = \{1, d+2, t_1, \dots, t_6\}$.

- Suppose $a = 1$, we do not need to impose any equality constraints and we leave p, p', q, q' untouched and apply [Lemma 3.3](#) for both $\tilde{P}_{[1,d+1]}^a$ and $\tilde{P}_{[d+2,2d+2]}^a$.
- Suppose $a = 2$, then we need to impose the constraint $\mathbb{1}[i_{t_1} = i_{t_4}]$. There are two cases.
 - Case 1: Both t_1, t_4 lie in the same interval in $\{[2, d+1], [d+3, 2d+2]\}$. Without loss of generality, assume that both t_1, t_4 lie in the first interval $[2, d+1]$, the analysis for the other case is similar. We set the parameters $p \leftarrow p+1, q \leftarrow q$ and $(s_1, t_1) \leftarrow (t_1, t_4)$ in the construction of $\tilde{P}_{[1,d+1]}^a$. We leave p', q' untouched for the construction of $\tilde{P}_{[d+2,2d+2]}^a$. This has the effect of imposing the equality constraint $i_{t_1} = i_{t_4}$ within $\tilde{P}_{[1,d+1]}^a$.
 - Case 2: t_1, t_4 lie in different intervals in $\{[2, d+1], [d+3, 2d+2]\}$. Without loss of generality, assume that $t_1 \in [2, d+1]$ and $t_4 \in [d+3, 2d+2]$, the analysis for the other case is similar. We will set the parameters $p \leftarrow p, q \leftarrow q+1$ and $r_1 \leftarrow t_1$ in the construction of $\tilde{P}_{[1,d+1]}^a$ and set $p' \leftarrow p', q' \leftarrow q'+1$ and $r'_1 \leftarrow t_4$ in the construction of $\tilde{P}_{[d+2,2d+2]}^a$. This has the effect of storing i_{t_1} in B_{d+2} from the first half $\tilde{P}_{[1,d+1]}^a$ and storing i_{t_4} in B_{d+2} from the second half $\tilde{P}_{[d+2,2d+2]}^a$ and enforcing equality in between using B_{d+2} .
- The analysis for $a = 3$ is identical to the case $a = 2$ by replacing t_1 by t_3 and t_4 by t_6 . We also replace $(s_1, t_1), (s'_1, t'_1)$ by $(s_2, t_2), (s'_2, t'_2)$ and r_1, r'_1 by r_2, r'_2 .
- Suppose $a = 4$, then we carry out the $a = 2$ step for the constraints t_1, t_4 followed by the $a = 3$ step for t_3, t_6 .

It is not too difficult to see that this indeed ensures that [Equation \(38\)](#) is satisfied. This completes the proof.

6 Fourier Growth of BQP: Proof of [Theorem 1.7](#)

In this section we will show Fourier growth bounds on BQP algorithms using our improved matrix decomposition lemma ([Lemma 3.3](#)). Given the expression for the acceptance probability of d -query BQP algorithm ([Claim 2.10](#)), it is not too difficult to derive an expression for the Fourier coefficients under any restriction – this part is similar to the proof of [Claim 4.1](#) from [Claim 2.12](#). We obtain the following claim, whose proof is deferred to [Appendix A.5](#). As before, we can assume without loss of generality that the restriction ρ fixes all but the first $\tilde{N}$ coordinates for some $\tilde{N} \leq N$, by permuting the matrices applied by the quantum algorithm appropriately.

Claim 6.1. Let $f(x)$ be the acceptance probability of a d -query BQP algorithm and $\rho \in \{-1, 1, *\}^N$ be any restriction that leaves the first $\tilde{N}$ coordinates unfixed. Then, there exist matrices $V_1^\rho, \dots, V_{2d+1}^\rho$ such that for all $S \subseteq [\tilde{N}]$,

$$\widehat{f|_\rho}(S) = \sum_{I_1, \dots, I_{2d+2}} v[I_1] \cdot \left(\prod_{t \in [2d+1]} V_t^\rho[I_t | I_{t+1}] \right) \cdot v[I_{2d+2}] \cdot \mathbb{1} \left[S = \bigoplus_{\substack{t \in [2, 2d+1] \\ i_t \leq \tilde{N}}} \{i_t\} \right], \quad (39)$$

where $V_1^\rho, \dots, V_{2d+1}^\rho \in \mathbb{C}^{M \times M}$ satisfy $\|V_t^\rho\|_{\text{op}} \leq 1$ for $t \in [2d+1]$ and $v \in \mathbb{C}^M$ is a vector with first coordinate 1 and the rest zeroes.

Now that we have an expression for the Fourier coefficients, we turn our attention to proving Fourier growth bounds.

6.1 Level- ℓ Fourier Growth

Fix any signs $\alpha_S \in [-1, 1]$ for each $S \in \binom{[\tilde{N}]}{\ell}$. Our goal is to upper bound

$$L_{1,\ell}^\alpha(f) = \sum_{|S|=\ell} \alpha_S \cdot \widehat{f|_\rho}(S).$$

From Equation (39) in Claim 6.1, we see that our goal is to upper bound

$$L_{1,\ell}^\alpha(f|_\rho) = \sum_{\substack{S \subseteq [\tilde{N}] \\ |S|=\ell}} \sum_{I_1, \dots, I_{2d+2} \in [M]} v[I_1] \cdot \left(\prod_{t \in [2d+1]} V_t^\rho[I_t | I_{t+1}] \right) \cdot v[I_{2d+2}] \cdot \mathbb{1} \left[S = \bigoplus_{\substack{t \in [2, 2d+1] \\ i_t \leq \tilde{N}}} \{i_t\} \right] \cdot \alpha_S. \quad (40)$$

Observe that if $\bigoplus_{t \in [2, 2d+1], i_t \leq \tilde{N}} \{i_t\} = S$, then in particular, there must exist a subset $T \subseteq [2, 2d+1]$ of size ℓ such that $\{i_t : t \in T\}$ is a sequence of ℓ distinct elements in $[\tilde{N}]$ and $\bigoplus_{\substack{t \in [2, 2d+1] \setminus T \\ \text{with } i_t \leq \tilde{N}}} \{i_t\} = \emptyset$.

Conversely, for any T and $\{i_t\}_{t \in T}$ satisfying the above conditions, it defines a unique $S = \{i_t : t \in T\}$. Fix $T \subseteq [2, 2d+1]$ of size ℓ (this can be done in $\binom{2d}{\ell}$ ways). Define

$$\begin{aligned} \Delta_T := & \sum_{I_1, \dots, I_{2d+1} \in [M]} v[I_1] \cdot \left(\prod_{t \in [2d+1]} V_t^\rho[I_t | I_{t+1}] \right) \cdot v[I_{2d+2}] \\ & \cdot \mathbb{1} \left[\bigoplus_{\substack{t \in [2, 2d+1] \setminus T \\ \text{with } i_t \leq \tilde{N}}} \{i_t\} = \emptyset \right] \cdot \alpha_{\{i_{t_1}, \dots, i_{t_\ell}\}} \cdot \mathbb{1} [i_{t_1}, \dots, i_{t_\ell} \in [\tilde{N}] \text{ are distinct}]. \end{aligned} \quad (41)$$

From the above paragraph, it follows that

$$L_{1,\ell}^\alpha(f|_\rho) = \sum_{T \in \binom{[2, 2d+1]}{\ell}} \Delta_T \leq \binom{2d}{\ell} \cdot \max_{T \in \binom{[2, 2d+1]}{\ell}} \Delta_T.$$

We will now show that for all $T \in \binom{[2, 2d+1]}{\ell}$, we have $\Delta_T \leq N^{(\ell-1)/2}$. This, along with the above equation would imply that $L_{1,\ell}^\alpha(f|\rho) \leq \binom{2d}{\ell} \cdot N^{(\ell-1)/2}$ as desired. We now show the desired bound of $\Delta_T \leq N^{(\ell-1)/2}$.

Let $T = \{t_1, \dots, t_\ell\}$ where $2 \leq t_1 < \dots < t_\ell \leq 2d+1$. We partition $[1, 2d+2]$ into intervals $[1, t_1) \sqcup [t_1, t_\ell) \sqcup [t_\ell, 2d+2]$. Apply [Lemma 3.1](#) to the matrices $V_1, \dots, V_{t_1-1}$ in this order to obtain $\tilde{V}_{[1, t_1)}$ and the matrices $V_{t_\ell}, \dots, V_{2d+1}$ in reverse order to obtain $\tilde{V}_{[t_\ell, 2d+2]}$ such that

$$\tilde{V}_{[1, t_1)}[I_1 | I_{t_1} S_{t_1}] = \sum_{I_2, \dots, I_{t_1-1}} \left(\prod_{t \in [1, t_1)} V_t^\rho[I_t | I_{t+1}] \right) \cdot \mathbb{1} \left[S_{t_1} = \bigoplus_{\substack{t \in (1, t_1) \\ i_t \leq \tilde{N}}} \{i_t\} \right] \quad (42)$$

$$\begin{aligned} \tilde{V}_{[t_\ell, 2d+2]}[I_{t_\ell} S_{t_\ell} | I_{2d+2}] &= \sum_{I_{t_\ell+1}, \dots, I_{2d+1}} \left(\prod_{t \in [t_\ell, 2d+2]} V_t^\rho[I_t | I_{t+1}] \right) \cdot \mathbb{1} \left[S_{t_\ell} = \bigoplus_{\substack{t \in (t_\ell, 2d+2) \\ i_t \leq \tilde{N}}} \{i_t\} \right], \\ \|\tilde{V}_{[1, t_1)}\|_{\text{op}} &\leq 1 \quad \text{and} \quad \|\tilde{V}_{[t_\ell, 2d+2]}\|_{\text{op}} \leq 1 \end{aligned} \quad (43)$$

Define a third matrix $\tilde{V}'_{[t_1, t_\ell]}$ with rows indexed by $I_{t_1} S_{t_1}$ and columns by $I_{t_\ell} S_{t_\ell}$ so that

$$\begin{aligned} \tilde{V}'_{[t_1, t_\ell]}[I_{t_1} S_{t_1} | I_{t_\ell} S_{t_\ell}] &:= \sum_{I_{t_1+1}, \dots, I_{t_\ell-1}} \left(\prod_{t \in [t_1, t_\ell)} V_t^\rho[I_t | I_{t+1}] \right) \cdot \mathbb{1} \left[S_{t_\ell} = S_{t_1} \bigoplus_{\substack{t \in (t_1, t_\ell) \setminus T \\ i_t \leq \tilde{N}}} \{i_t\} \right] \\ &\cdot \alpha_{\{i_{t_1}, \dots, i_{t_\ell}\}} \cdot \mathbb{1} \left[i_{t_1}, \dots, i_{t_\ell} \in [\tilde{N}] \text{ are distinct} \right]. \end{aligned} \quad (44)$$

By combining [Equations \(41\), \(42\) and \(44\)](#), we see that

$$\begin{aligned} \Delta_T &\triangleq \sum_{\substack{I_1, I_{2d+2} \in [M] \\ I_{t_1}, I_{t_\ell} \in [M] \\ S_{t_1}, S_{t_\ell} \subseteq [N]}} v[I_1] \cdot \tilde{V}_{[1, t_1)}[I_1 | I_{t_1} S_{t_1}] \cdot \tilde{V}'_{[t_1, t_\ell]}[I_{t_1} S_{t_1} | I_{t_\ell} S_{t_\ell}] \cdot \tilde{V}_{[t_\ell, 2d+2]}[I_{t_\ell} S_{t_\ell} | I_{2d+2}] \cdot v[I_{2d+2}] \\ &\leq \|v\| \cdot \|\tilde{V}_{[1, t_1)}\|_{\text{op}} \cdot \|\tilde{V}'_{[t_1, t_\ell]}\|_{\text{op}} \cdot \|\tilde{V}_{[t_\ell, 2d+2]}\|_{\text{op}} \cdot \|v\| \\ &\leq \|\tilde{V}'_{[t_1, t_\ell]}\|_{\text{op}}. \end{aligned} \quad (\text{by [Equation \(43\)](#) and since } \|v\| = 1.)$$

It thus suffices to bound the operator norm of $\tilde{V}'_{[t_1, t_\ell]}$. Observe that $\tilde{V}'_{[t_1, t_\ell]}$ is almost equal to the matrix one would get by applying [Lemma 3.3](#) on the matrices $V_{t_1}^\rho, \dots, V_{t_\ell-1}^\rho$, but not quite – the difference is that we only need to sum over *distinct* $i_{t_1}, \dots, i_{t_\ell} \in [\tilde{N}]$, as well as multiply by a sign $\alpha_{i_{t_1}, \dots, i_{t_\ell}}$. To get around this, we will use an idea similar to the one we used in the proof of [Theorem 1.5](#). We will apply [Lemma 3.3](#) to the matrices $V_{t_1}^\rho, \dots, V_{t_\ell-1}^\rho$ with memory constraints defined by $i_{t_1}, i_{t_2}, \dots, i_{t_{\ell-1}}$. There are $\ell-1$ memory constraints that are well defined since $i_{t_1}, \dots, i_{t_{\ell-1}} \in [i_{t_1}, i_{t_\ell}]$. We obtain a matrix $\tilde{V}_{[t_1, t_\ell]}$ with $\|\tilde{V}_{[t_1, t_\ell]}\|_{\text{op}} \leq 1$ whose rows are indexed by

$I_{t_1} \in [M], S_{t_1} \subseteq [N]$ and columns by $I_{t_\ell} \in [M], S_{t_\ell} \subseteq [N], B_{t_\ell} \in [N]^{\ell-1}$, with entries satisfying

$$\begin{aligned} \tilde{V}_{[t_1, t_\ell]}[I_{t_1} S_{t_1} | I_{t_\ell} S_{t_\ell} B_{t_\ell}] &= \left(\prod_{t \in [1, t_1]} V_t^\rho[I_t | I_{t+1}] \right) \cdot \mathbb{1} \left[S_{t_\ell} = \bigoplus_{\substack{t \in (t_1, t_\ell) \setminus T \\ i_t \leq \tilde{N}}} \{i_t\} \right] \\ &\quad \cdot \mathbb{1} [B_{t_\ell} = (i_{t_1}, i_{t_2}, \dots, i_{t_{\ell-1}})] . \end{aligned} \quad (45)$$

We now define a matrix V' with rows indexed by $I_{t_\ell} \in [M], S_{t_\ell} \subseteq [N], B_{t_\ell} \in [N]^{\ell-1}$ and columns by $I'_{t_\ell} \in [M], S'_{t_\ell} \subseteq [N]$ whose entries are given by

$$V'[I_{t_\ell} S_{t_\ell} B_{t_\ell} | I'_{t_\ell} S'_{t_\ell}] = \mathbb{1} [I_{t_\ell} = I'_{t_\ell} \text{ and } S_{t_\ell} = S'_{t_\ell}] \cdot \mathbb{1} [\{i_{t_\ell}\} \cup B_{t_\ell} \text{ has } \ell \text{ distinct elements}] \cdot \alpha_{\{i_{t_\ell}\} \cup B_{t_\ell}} . \quad (46)$$

Observe from [Equations \(44\) to \(46\)](#) that $\tilde{V}'_{[t_1, t_\ell]} = \tilde{V}_{[t_1, t_\ell]} \cdot V'$ and thus,

$$\|\tilde{V}'_{[t_1, t_\ell]}\|_{\text{op}} \leq \|\tilde{V}_{[t_1, t_\ell]}\|_{\text{op}} \cdot \|V'\|_{\text{op}} \leq \|V'\|_{\text{op}} .$$

Finally, we show that $\|V'\|_{\text{op}} \leq N^{(\ell-1)/2}$. To see this, observe that V' is block-diagonal with respect to $I_{t_\ell} S_{t_\ell}$. Fix any such $I_{t_\ell} S_{t_\ell}$ and consider the resulting block. This is a $N^{\ell-1} \times 1$ matrix whose entries are $[-1, 1]$ -valued, hence, we have $\|V'\|_{\text{op}} \leq N^{(\ell-1)/2}$. This completes the proof.

7 Acknowledgements

The author would like to thank Vishnu Iyer, Dale Jacobs, Natalie Parham, Makrand Sinha, Kewen Wu and Henry Yuen for very helpful comments and discussions.

References

- [AA11] Scott Aaronson and Alex Arkhipov. The computational complexity of linear optics. In *Proceedings of the Forty-Third Annual ACM Symposium on Theory of Computing, STOC '11*, page 333–342, New York, NY, USA, 2011. Association for Computing Machinery. [3](#)
- [AA15] Scott Aaronson and Andris Ambainis. Forrelation: A problem that optimally separates quantum from classical computing. In *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing*, pages 307–316, 2015. [3](#), [6](#), [9](#)
- [Aar10] Scott Aaronson. BQP and the polynomial hierarchy. In *Proceedings of the 61st IEEE Annual Symposium on Foundations of Computer Science*, pages 141–150, 2010. [3](#), [6](#), [9](#)
- [ABKM17] Scott Aaronson, Adam Bouland, Greg Kuperberg, and Saeed Mehraban. The computational complexity of ball permutations. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 317–327. ACM, 2017. [3](#), [4](#)
- [AG23] Srinivasan Arunachalam and Uma Girish. Trade-offs between entanglement and communication. In *Proceedings of the 38th Computational Complexity Conference, CCC '23*, Dagstuhl, DEU, 2023. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. [8](#)

- [AGL23] Srinivasan Arunachalam, Uma Girish, and Noam Lifshitz. One Clean Qubit Suffices for Quantum Communication Advantage. 10 2023. 4
- [BS21] Nikhil Bansal and Makrand Sinha. k -forrelation optimally separates quantum and classical query complexity. In Samir Khuller and Virginia Vassilevska Williams, editors, *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1303–1316, 2021. 6, 7, 8, 10, 17
- [BV97] Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997. 3
- [CCHL23] Sitan Chen, Jordan Cotler, Hsin-Yuan Huang, and Jerry Li. The complexity of nisq . *Nature Communications*, 14(1):6001, 2023. 3, 14
- [CHHK24] Nai-Hui Chia, Min-Hsiu Hsieh, Shih-Han Hung, and En-Jui Kuo. Oracle separation between noisy quantum polynomial time and the polynomial hierarchy, 2024. 14
- [CHLT19] Eshan Chattopadhyay, Pooya Hatami, Shachar Lovett, and Avishay Tal. Pseudorandom generators from the second fourier level and applications to AC^0 with parity gates. In Avrim Blum, editor, *Proceedings of the 10th Innovations in Theoretical Computer Science Conference*, volume 124 of *LIPICs*, pages 22:1–22:15, 2019. 6, 7, 9, 16
- [CSS21] Anirban N. Chowdhury, Rolando D. Somma, and Yigit Subasi. Computing partition functions in the one-clean-qubit model. *Phys. Rev. A*, 103:032422, Mar 2021. 4
- [DFC05] Animesh Datta, Steven T. Flammia, and Carlton M. Caves. Entanglement and the power of one qubit. *Phys. Rev. A*, 72:042316, Oct 2005. 4
- [DJ92] David Deutsch and Richard Jozsa. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 439:553–558, 1992. 3
- [FKM⁺15] Keisuke Fujii, Hirotada Kobayashi, Tomoyuki Morimae, Harumichi Nishimura, Shuhei Tamate, and Seiichiro Tani. Power of quantum computation with few clean qubits. *arXiv preprint arXiv:1509.07276*, 2015. 4
- [FKM⁺18] Keisuke Fujii, Hirotada Kobayashi, Tomoyuki Morimae, Harumichi Nishimura, Shuhei Tamate, and Seiichiro Tani. Impossibility of classically simulating one-clean-qubit model with multiplicative error. *Physical review letters*, 120(20):200502, 2018. 4
- [Geo25] Alexandru Georghiu. Verifiable quantum advantage: old and new ideas, 2025. Link to recording of talk: <https://youtu.be/7NqAcaSwKf8?si=2ZBkEkbiVaUMSzMJ>. 14
- [GSTW24] Uma Girish, Makrand Sinha, Avishay Tal, and Kewen Wu. The power of adaptivity in quantum query algorithms. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 1488–1497, 2024. 8, 9, 19
- [GTW21] Uma Girish, Avishay Tal, and Kewen Wu. Fourier growth of parity decision trees. In *36th Computational Complexity Conference (CCC 2021)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2021. 6
- [IRR⁺21] Siddharth Iyer, Anup Rao, Victor Reis, Thomas Rothvoss, and Amir Yehudayoff. Tight bounds on the fourier growth of bounded functions on the hypercube. *CoRR*, abs/2107.06309, 2021. 6, 7, 8, 9, 14

- [JM24] Dale Jacobs and Saeed Mehraban. Space just above one clean qubit, 2024. [3](#), [4](#), [5](#), [10](#), [15](#)
- [KL98] E. Knill and R. Laflamme. Power of one bit of quantum information. *Phys. Rev. Lett.*, 81:5672–5675, Dec 1998. [3](#), [4](#), [18](#)
- [Man95] Yishay Mansour. An $O(n^{\log \log n})$ learning algorithm for DNF under the uniform distribution. *J. Comput. Syst. Sci.*, 50(3):543–550, 1995. Appeared in COLT, 1992. [6](#)
- [MFF14] Tomoyuki Morimae, Keisuke Fujii, and Joseph F Fitzsimons. Hardness of classically simulating the one-clean-qubit model. *Physical review letters*, 112(13):130502, 2014. [3](#), [4](#)
- [Mor17] Tomoyuki Morimae. Hardness of classically sampling the one-clean-qubit model with constant total variation distance error. *Physical Review A*, 96(4):040302, 2017. [4](#)
- [RT22] Ran Raz and Avishay Tal. Oracle separation of BQP and PH. *J. ACM*, 69(4):30:1–30:21, 2022. [6](#), [7](#), [9](#), [16](#)
- [SB08] Dan J. Shepherd and Michael J. Bremner. Temporally unstructured quantum computation. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 465:1413 – 1439, 2008. [3](#)
- [She10] Daniel James Shepherd. Quantum complexity: restrictions on algorithms and architectures. *CoRR*, abs/1005.1425, 2010. [4](#)
- [Shu25] Noah Shatty. Simons institute summer cluster on quantum computing: Lightning talks, 2025. Link to recording of talk: <https://www.youtube.com/live/7F5LBNGDRmk?si=NhPVNL25qGTtcngP&t=2651>. [14](#)
- [Sim97] Daniel R. Simon. On the power of quantum computation. *SIAM J. Comput.*, 26(5):1474–1483, October 1997. [3](#)
- [SJ08] Peter W. Shor and Stephen P. Jordan. Estimating jones polynomials is a complete problem for one clean qubit. *Quantum Info. Comput.*, 8(8):681–714, September 2008. [4](#)
- [SSW23] Alexander A. Sherstov, Andrey A. Storozhenko, and Pei Wu. An optimal separation of randomized and quantum query complexity. *SIAM J. Comput.*, 52(2):525–567, 2023. [6](#), [7](#), [9](#)
- [Tal17] Avishay Tal. Tight bounds on the Fourier spectrum of AC0. In *Computational Complexity Conference*, volume 79 of *LIPICs*, pages 15:1–15:31. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. [6](#)
- [Tal20] Avishay Tal. Towards optimal separations between quantum and randomized query complexities. In *Proceedings of the 61st IEEE Annual Symposium on Foundations of Computer Science*, pages 228–239, 2020. [6](#), [7](#), [9](#)

A Appendix

A.1 Quantum Algorithms with Classical Pre-Processing.

In this section, we prove a variant of [Theorems 1.5 to 1.7](#) in a more general setting of algorithms that can perform classical pre-processing. We now describe this model more formally. A d -query DQC_k (respectively $\frac{1}{2}\text{BQP}$, BQP) algorithm with classical pre-processing consists of two phases:

- **CLASSICAL PHASE:** The algorithm performs d classical queries on clean workspace.
- **QUANTUM PHASE:** Based on the results, the algorithm chooses a d -query DQC_k (respectively $\frac{1}{2}\text{BQP}$, BQP) algorithm to run and returns the output.

Theorem A.1. *Let $\mathcal{F}$ denote the family of acceptance probabilities of a class of algorithms without classical pre-processing. Let $f(x)$ be the acceptance probability of an algorithms with d classical pre-processing queries. Let $\rho \in \{-1, 1, *\}^N$ be any restriction and $\alpha \in [-1, 1]^{\binom{N}{\ell}}$ signs. Then, there exist $f' \in \mathcal{F}$ such that*

$$L_{1,\ell}^\alpha(f|\rho) \leq \sum_{k=0}^{\ell} \binom{d}{\ell-k} \cdot \max_{\alpha'} L_{1,k}^{\alpha'}(f'|\rho),$$

where the maximum is over α' , another family of signs. Furthermore, if α are level- ℓ signs as in [Definition 2.7](#), then so is α' when $k = \ell$

Corollary A.2. *Analogues of [Theorems 1.5 to 1.7](#) hold even for algorithms with classical pre-processing.*

Proof of Theorem A.1. We view the classical phase as a decision tree of depth d with 2^d leaves where each leaf y selects an algorithm f_y to run. Furthermore, we view each leaf y as a partial assignment in $\{-1, 1, *\}^N$ where the coordinates that are queried are assigned ± 1 depending on the outcome of the query, and the coordinates not queried are assigned $*$. We use $y^{-1}(*)$ to denote the coordinates of y that are alive. We know that $|y^{-1}(*)| \geq N - d$. This defines a restriction $\rho_y \in \{-1, 1, *\}^N$ of the variables which restricts the i -th coordinate to y_i if $y_i \in \{-1, 1\}$ and leaves it alive otherwise. We can assume that any y that is ever traversed is consistent with ρ . For any such y , let $f_y(x)$ be the acceptance probability of the algorithm chosen conditioned on receiving y in the first stage. Consider:

$$\begin{aligned} & L_{1,\ell}^\alpha(f|\rho) \\ &= \mathbb{E}_{x \sim \{-1, 1\}^N} \left[\sum_{|S|=\ell} \alpha_S \cdot f|_\rho(x) \cdot \chi_S(x) \right] \\ &= \mathbb{E}_{y \text{ consistent with } \rho} \left[\mathbb{E}_{\substack{x \sim \{-1, 1\}^N \\ \text{consistent with } y}} \left[\sum_{|S|=\ell} \alpha_S \cdot f_y(\rho(x)) \cdot \chi_S(x) \right] \right] \\ &= \mathbb{E}_{y \text{ consistent with } \rho} \left[\mathbb{E}_{\substack{x \sim \{-1, 1\}^N \\ \text{consistent with } y}} \left[\sum_{k=0}^{\ell} \sum_{\substack{S_1 \subseteq y^{-1}(*) \\ S_2 \subseteq [N] \setminus y^{-1}(*) \\ |S_1|=k, |S_2|=\ell-k}} \alpha_{S_1 \cup S_2} \cdot f_y(\rho(x)) \cdot \chi_{S_1}(x) \cdot \chi_{S_2}(x) \right] \right]. \end{aligned}$$

Fix a leaf y that maximizes the above quantity. Since we are only taking expectations over x consistent with y , we can replace $\chi_{S_2}(x)$ by $\chi_{S_2}(y)$ in the R.H.S. above and similarly, $\rho(x)$ only depends on the variables in S_1 . Once we do this, x is completely free of y and we can replace the expectation of $x \sim \{-1, 1\}^N$ consistent with y by simply $x \sim \{-1, 1\}^N$. We obtain

$$L_{1,\ell}^\alpha(f|\rho) \leq \mathbb{E}_{x \sim \{-1,1\}^N} \left[\sum_{k=0}^{\ell} \sum_{\substack{S_1 \subseteq y^{-1}(\ast) \\ S_2 \subseteq [N] \setminus y^{-1}(\ast) \\ |S_1|=k, |S_2|=\ell-k}} \alpha_{S_1 \cup S_2} \cdot \chi_{S_2}(y) \cdot f_y(\rho(x)) \cdot \chi_{S_1}(x) \right] \quad (47)$$

Since $|\chi_{S_2}(y)| \leq 1$, applying Triangle Inequality gives

$$L_{1,\ell}^\alpha(f|\rho) \leq \sum_{k=0}^{\ell} \sum_{\substack{S_2 \subseteq [N] \setminus y^{-1}(\ast) \\ |S_2|=\ell-k}} \left| \mathbb{E}_{x \sim \{-1,1\}^N} \left[\sum_{\substack{S_1 \subseteq y^{-1}(\ast) \\ |S_1|=k}} \alpha_{S_1 \cup S_2} \cdot f_y(\rho(x)) \cdot \chi_{S_1}(x) \right] \right| \quad (48)$$

Define $\gamma \in [-1, 1]^N$ by $\gamma_i = 1$ if $i \in y^{-1}(\ast)$ and 0 otherwise. For any fixed $k \in \{0, \dots, \ell\}$ and $S_2 \subseteq [N] \setminus y^{-1}(\ast)$ of size $\ell - k$, define signs α^{S_2} that are non-zero only for $S_1 \subseteq [N]$ with size k so that

$$\alpha_{S_1}^{S_2} := \alpha_{S_1 \cup S_2} \cdot \chi_{S_1}(\gamma).$$

Observe that $\chi_{S_1}(\gamma) = 1$ if $S_1 \subseteq y^{-1}(\ast)$ and 0 otherwise. Thus,

$$\sum_{\substack{S_1 \subseteq y^{-1}(\ast) \\ |S_1|=k}} \alpha_{S_1 \cup S_2} \cdot \chi_{S_1}(x) = \sum_{\substack{S_1 \subseteq [N] \\ |S_1|=k}} \alpha_{S_1 \cup S_2} \cdot \chi_{S_1}(\gamma) \cdot \chi_{S_1}(x) \triangleq \sum_{\substack{S_1 \subseteq [N] \\ |S_1|=k}} \alpha_{S_1}^{S_2} \cdot \chi_{S_1}(x).$$

Finally, we observe that

$$\mathbb{E}_{x \sim \{-1,1\}^N} \left[\sum_{\substack{S_1 \subseteq [N] \\ |S_1|=k}} \alpha_{S_1}^{S_2} \cdot f_y(\rho(x)) \cdot \chi_{S_1}(x) \right] \triangleq \sum_{\substack{S_1 \subseteq [N] \\ |S_1|=k}} \alpha_{S_1}^{S_2} \cdot \widehat{f_y|_\rho}(S_1) \triangleq L_{1,k}^{\alpha^{S_2}}(f_y|_\rho).$$

Substituting this in [Equation \(48\)](#), we get

$$L_{1,\ell}^\alpha(f) \leq \sum_{k=0}^{\ell} \binom{d}{\ell-k} \cdot \max_{\alpha'} L_{1,k}^{\alpha'}(f_y|_\rho),$$

where we used the fact that $N - |y^{-1}(\ast)| \leq d$. Furthermore, when $k = \ell$ (i.e., when $S_2 = \emptyset$), it is easy to see that if α is a family of signs as in [Definition 2.7](#), so is the result family $\alpha^\emptyset$. This completes the proof. $\square$

Proof of [Corollary A.2](#) from [Lemma 3.3](#). Let $\mathcal{F}$ (respectively $\mathcal{F}'$) denote the class of d -query DQC_k algorithms with (respectively without) classical pre-processing. Applying [Theorem A.1](#), we have

$$L_{1,\ell}(\mathcal{F}) \leq \sum_{k=0}^{\ell} \binom{d}{\ell-k} \cdot L_{1,k}^{\alpha'}(f|_{\rho'}).$$

We now apply [Theorem 1.5](#) to bound each $L_{1,k}^{\alpha'}(f|_{\rho'})$ and this gives

$$L_{1,\ell}(\mathcal{F}) \leq \sum_{k=0}^{\ell} \binom{d}{\ell-k} \cdot \binom{2d}{k} \cdot N^{(k-2)/2} \leq \binom{3d}{\ell} \cdot N^{(\ell-2)/2}$$

as desired. The proof for BQP algorithms is identical and we obtain a bound of $\binom{3d}{\ell} \cdot N^{(\ell-1)/2}$.

Let $\mathcal{F}$ (respectively $\mathcal{F}'$) denote the class of d -query $\frac{1}{2}$ BQP algorithms with (respectively without) classical pre-processing. Let α be a valid signing of the level-3 Fourier coefficients as in [Definition 2.7](#). Applying [Theorem A.1](#), we have

$$L_{1,3}^{\alpha}(\mathcal{F}) \leq \max_{\alpha'} \left(d^3 + d^2 \cdot L_{1,1}^{\alpha'}(\mathcal{F}') + d \cdot L_{1,2}^{\alpha'}(\mathcal{F}') + L_{1,3}^{\alpha'}(\mathcal{F}') \right).$$

By the guarantee of [Theorem A.1](#), α' is a valid set of signs for level-3 as long as α is a valid set of signs for level-3 as in [Definition 2.7](#). Thus, applying [Theorem 1.6](#) gives us $L_{1,3}^{\alpha'}(\mathcal{F}') \leq O(d^3 \sqrt{N})$. For levels $k < 3$, we can apply [Theorem 1.7](#) to conclude that $L_{1,1}^{\alpha'}(\mathcal{F}') \leq O(d)$ and $L_{1,2}^{\alpha'}(\mathcal{F}') \leq O(d^2 \sqrt{N})$. Combining these, we get $L_{1,3}^{\alpha}(\mathcal{F}) \leq O(d^3) \cdot \sqrt{N}$ as desired.

For level-6, doing a similar calculation gives us

$$L_{1,6}^{\alpha}(\mathcal{F}) \leq \max_{\alpha'} \left(\sum_{k=0}^6 d^{\ell-k} \cdot L_{1,k}^{\alpha'}(\mathcal{F}') \right).$$

For $k = 6$, we use [Theorem 1.6](#) to conclude that $L_{1,6}^{\alpha'}(\mathcal{F}') \leq O(d^6) \cdot N^{1.5}$. For levels $k \leq 4$, we use [Theorem 1.7](#) to conclude that $L_{1,k}^{\alpha'}(\mathcal{F}') \leq O(d^k) \cdot N^{(k-1)/2} \leq O(d^5) \cdot N^{1.5}$. We run into a complication here which is that for level-5, the bound given by [Theorem 1.7](#) is already $\Omega(N^2)$. We will instead show an upper bound of $O(d^7) \cdot N^{1.5}$, which is d times the bound given by [Theorem 1.6](#) for algorithms without pre-processing. To show this, we will revisit the proof of [Theorem A.1](#) and in particular consider [Equation \(48\)](#). By the above discussion, we only need to consider the contribution of terms $S_2 \subseteq [N] \setminus y^{-1}(*)$ with $|S_2| = 1$. Thus, it suffices to bound $\Delta := \sum_{i \in [N] \setminus y^{-1}(*)} |\Delta_i|$, where

$$\Delta_i = \mathbb{E}_{x \sim \{-1,1\}^N} \left[\sum_{\substack{S_1 \subseteq y^{-1}(*) \\ |S_1|=5}} \alpha_{S_1 \cup \{i\}} \cdot f_y(\rho(x)) \cdot \chi_{S_1}(x) \right] \triangleq \sum_{\substack{S_1 \subseteq y^{-1}(*) \\ |S_1|=5}} \alpha_{S_1 \cup \{i\}} \cdot \widehat{f_y|_{\rho}}(S_1).$$

Since $|[N] \setminus y^{-1}(*)| \leq d$, we have $\Delta \leq d \cdot \max_i \Delta_i$. Fix any $i \in [N] \setminus y^{-1}(*)$ that maximizes this and we will show a bound on Δ_i as follows. Recall that $f_y(\rho(x))$ is the acceptance probability of a certain d -query $\frac{1}{2}$ BQP algorithm $\mathcal{A}$. Consider the $(d+1)$ -query $\frac{1}{2}$ BQP algorithm $\mathcal{A}'$ that applies $\mathcal{A}$, computes x_i using an additional qubit and conditioned on this qubit starting in the $|0\rangle$ state, XORs the outcome x_i with the output of $\mathcal{A}$. It is not too difficult to show that the acceptance probability of $\mathcal{A}'$ is given by

$$\begin{aligned} f'(x) &:= \frac{1}{2} \cdot f_y(\rho(x)) + \frac{(1+x_i)}{4} \cdot f_y(\rho(x)) + \frac{(1-x_i)}{4} \cdot (1 - f_y(\rho(x))) \\ &= \frac{1}{4} - \frac{1}{4} \cdot x_i + \frac{1}{2} \cdot f_y(\rho(x)) + \frac{1}{2} \cdot x_i \cdot f_y(\rho(x)). \end{aligned}$$

We now consider the signed level-6 Fourier growth of both sides of the above equation with respect to α . Since $i \in [N] \setminus y^{-1}(*)$ and ρ restricts all such coordinates, we get

$$L_{1,6}^{\alpha}(f') = \frac{1}{2} L_{1,6}^{\alpha}(f_y|_{\rho}) + \frac{1}{2} \sum_{\substack{S_1 \subseteq y^{-1}(*) \\ |S_1|=5}} \alpha_{\{i\} \cup S_1} \cdot \widehat{f_y|_{\rho}}(S_1) = \frac{1}{2} L_{1,6}^{\alpha}(f_y|_{\rho}) + \frac{1}{2} \Delta_i.$$

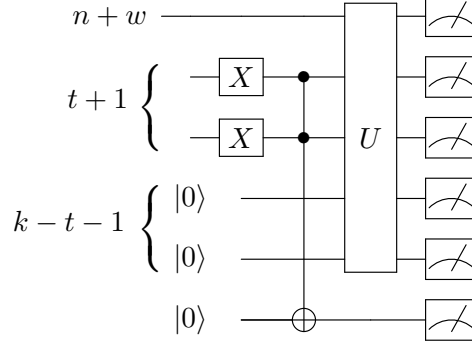


Figure 7: Simulating a DQC_k algorithm by a DQC_{k-t} algorithm.

Rearranging this equation and applying [Theorem 1.6](#) on $\mathcal{A}$ and $\mathcal{A}'$ implies that

$$|\Delta_i| \leq O(d^6) \cdot N^{1.5}.$$

This completes the proof and results in an overall bound of $O(d^7) \cdot N^{1.5}$ for the level-6 Fourier growth of d -query $\frac{1}{2}\text{BQP}$ algorithms with d classical pre-processing queries. $\square$

A.2 Simulating DQC_k algorithms by DQC_{k-t} algorithms.

Claim A.3. *Let $g(x)$ be the bias of a d -query DQC_k algorithm. Then, there is a d -query DQC_{k-t} algorithm whose bias is $g(x) \cdot 2^{-t-1}$.*

Proof of Claim A.3. Given a d -query DQC_k algorithm with $n + w$ noisy bits, consider a DQC_{k-t} algorithm which uses $n + w + t + 1$ noisy bits and $k - t$ clean qubits as follows. Firstly, the algorithm applies the X gate to the last $t + 1$ noisy qubits and applies a Toffoli controlled on these qubits with the target as the final clean qubit. Then, apply the DQC_k algorithm on the first $n + w$ noisy qubits and the first k clean qubits. Finally, measure the last clean qubit. If it results in an outcome 1, then return the outcome of the DQC_k algorithm, otherwise, return a random bit (by taking an additional noisy qubit for instance).

Observe that this algorithm behaves identically to the original one whenever the $t + 1$ noisy qubits are in the all-zeroes state, which happens with probability 2^{-t-1} . In all other cases, the algorithm returns a uniformly random bit. Thus, the bias of the resulting algorithm is $2^{-t-1} \cdot g(x)$. $\square$

A.3 Proof of Improved Matrix Decomposition Lemma

Proof of Lemma 3.3. To prove this lemma, we will show by induction on $t \in [d]$ that there exist matrices $\tilde{U}_1, \dots, \tilde{U}_d$ with spectral norm at most 1 such that for all $t \in [d]$, for all $I_1, I_{t+1} \in [M]$, $S_t, S_{t+1} \subseteq [N]$, and $A_{t+1} \in \{0, \dots, N\}^p$, $B_{t+1} \in \{0, \dots, N\}^q$, we have

$$\begin{aligned} \tilde{U}_{[1,t]}[I_1 S_1 | I_{t+1} S_{t+1} A_{t+1} B_{t+1}] &= \sum_{I_2, \dots, I_t \in [M]} \left(\prod_{t' \in [1,t]} U_{t'}[I_{t'} | I_{t'+1}] \right) \\ &\quad \cdot \mathbb{1} \left[S_{t+1} = S_1 \bigoplus_{\substack{t' \in [2,t] \setminus T \\ i_{t'} \leq \tilde{N}}} \{i_{t'}\} \right] \end{aligned} \tag{a}$$

$$\cdot \mathbb{1} [i_{s_j} = i_{t_j} \text{ for all } j \in [p] \text{ with } t_j \leq t + 1] \quad (\text{b})$$

$$\cdot \mathbb{1} \left[A_{t+1}(j) = \begin{cases} i_{s_j} & \text{for } j \in [p] \text{ with } s_j \leq t, t + 1 < t_j \\ 0 & \text{otherwise} \end{cases} \right] \quad (\text{c})$$

$$\cdot \mathbb{1} \left[B_{t+1}(j) = \begin{cases} i_{r_j} & \text{for } j \in [q] \text{ with } r_j \leq t \\ 0 & \text{otherwise.} \end{cases} \right] \quad (\text{d})$$

where as mentioned before, we use $\tilde{U}_{[1,t]}$ to denote $\tilde{U}_1 \cdots \tilde{U}_t$. We explain these conditions below.

Think of t as a clock that runs in $\{1, \dots, d\}$ and think of the the $s_1, t_1, \dots, s_p, t_p, r_1, \dots, r_q$ as moments in time. At the start of the t -th timestep, we get to see $I_t S_t A_t B_t$ and at the end of the timestep, we see $I_{t+1} S_{t+1} A_{t+1} B_{t+1}$. Suppose the current time is t and we are considering the matrix $\tilde{U}_{[1,t]}$. Then, **Equations (a) to (d)** impose the following constraints on the $I_2, \dots, I_t$:

- **Equation (a)**: The set S_{t+1} must be equal to the symmetric difference of S_1 and all the $\{i_{t'}\}$ for $t' \in [2, d] \setminus T$ with $i_{t'} \leq \tilde{N}$ – these are the relevant $i_{t'}$ that have appeared by the start of the current timestep t .
- **Equation (b)**: For all $t_j \leq t + 1$ that we will have been seen by the end of the current time step, we must have enforced the constraint that $i_{s_j} = i_{t_j}$.
- **Equation (c)**: For all s_j that we have seen by the start of the current timestep t and any t_j that lies ahead beyond time $t + 1$, we must retain information about i_{s_j} using $A_{t+1}(j)$ so that we can check equality with i_{t_j} in the future.
- **Equation (d)**: For all r_j that we have seen by the start the current timestep t , we must retain information about I_{r_j} using $B_{t+1}(j)$ until the very end.

Finally, when $t = d$, we will have imposed the constraints that $i_{s_j} = i_{t_j}$ for all $j \in [p]$, A_{d+1} will be the all-zeroes string since there no more time left, and we will have remembered all the i_{r_j} for $j \in [q]$ inside B_{d+1} . We will now show how to define $\tilde{U}_1, \dots, \tilde{U}_d$ so as to satisfy all these constraints. First, we first define some auxiliary functions that turn out to be useful.

update: For $t \in [d]$, define the **update** function which for $T \subseteq [N]$ and $i_t \in [N]$ satisfies

$$\text{update}_t(S, i_t) = \begin{cases} S \oplus \{i_t\} & \text{if } t \in [2, d] \setminus T \text{ and } i_t \leq \tilde{N} \\ S & \text{otherwise.} \end{cases}$$

This function captures the information we need to remember about the parity of the indices.

add: Let $\text{add}_j(A, i)$ denote the function that takes $A \in \{0, \dots, N\}^p$ and $i \in [N]$ and replaces the j -th entry by i for $j \in [p]$. We overload notation and use $\text{add}_j(B, i)$ to denote the function that takes $B \in \{0, \dots, N\}^q$ and replaces the j -th entry by $i \in [N]$ for $j \in [q]$.

remove: Finally, let $\text{remove}_j(A), \text{remove}_j(B)$ be the function that takes A, B and replaces the j -th element by 0.

For $t \in [d]$, we will define a matrix $\tilde{U}_t$ with rows and columns indexed by $I_t S_t A_t B_t$ and $I_{t+1} S_{t+1} A_{t+1} B_{t+1}$ respectively where $I_t, I_{t+1} \in [M], S_t, S_{t+1} \subseteq [N]$ and $B_t, B_{t+1} \in \{0, \dots, N\}^q, A_t, A_{t+1} \in \{0, \dots, N\}^p$. Set A_1, B_1 to be the all-zeroes string and $S_1 = \emptyset$. Set $\tilde{U}_t[I_t S_t A_t B_t | I_{t+1} S_{t+1} A_{t+1} B_{t+1}]$ to be either $U_t[I_t | I_{t+1}]$ or 0, where it is the former if and only if the following conditions are satisfied.

1. Firstly, for all $t \in [d]$, we always have $S_{t+1} = \text{update}_t(S_t, i_t)$. This, by induction, ensures that condition (a) is satisfied. We will now describe the constraints on A_t, B_t .
2. If $t \neq r_1, \dots, r_q, s_1, \dots, s_p, t_1 - 1, \dots, t_p - 1$, then $A_{t+1} = A_t, B_{t+1} = B_t$.
3. If t is equal r_j for some $j \in [q]$, then we enforce $B_{t+1} = \text{add}_j(B_t, i_t)$ and $A_{t+1} = A_t$. This has the effect of adding i_t to B_{t+1} and by induction ensures that condition (d) is satisfied.
4. Similarly, if t is equal s_j for some $j \in [p]$, we enforce $A_{t+1} = \text{add}_j(A_t, i_t)$ and $B_{t+1} = B_t$. This has the effect of adding i_t to A_{t+1} and by induction ensures that condition (c) is satisfied.
5. If $t+1 = t_j$ for some $j \in [p]$, then we enforce $A_t(j) = i_{t+1}$, $A_{t+1} = \text{remove}_j(A_t)$ and $B_{t+1} = B_t$. This has the effect of comparing i_{t+1} to the value of i_{s_j} that we have stored in A_t (since s_j must have appeared already), enforcing equality and erasing this value from A_{t+1} and by induction ensures that conditions (b), (c) are satisfied.
6. We remark that it is possible that conditions 4 and 5 hold simultaneously, this happens when $t = s_j$ for some $j \in [p]$ and $t+1 = t_{j'}$ for some $j' \in [p]$ with $j' \leq j$. Note that for each point in time, there can only be one j, j' for which this collision happens. In this case, we must enforce both 4 and 5 in this order. More precisely, we enforce $A_{t+1} = \text{remove}_{j'}(\text{add}_j(A_t, i_t))$, $(\text{add}_j(A_t, i_t))(j') = i_{t+1}$ and $B_{t+1} = B_t$. This by induction ensures that conditions (b), (c) are satisfied. Similarly, conditions 3 and 5 could also hold simultaneously, in which case we enforce both. More precisely, we enforce $B_{t+1} = \text{add}_j(B_t, i_t)$ and $A_{t+1} = \text{remove}_j(A_t)$ and $A_t(j) = i_{t+1}$. This by induction ensures that conditions (c), (d) are satisfied.

The above discussion shows that this choice of $\tilde{U}_1, \dots, \tilde{U}_t$ ensures that the inductive step holds. We will now turn our attention to proving bounds on the norms of the matrices $\tilde{U}_1, \dots, \tilde{U}_d$.

Bounds on the Spectral Norm. We now bound the spectral norm of the matrices and will show that $\|\tilde{U}_t\|_{\text{op}} \leq 1$ for all $t \in [d]$. For any $t \in [2, d]$, consider $\tilde{U}_t$. The rows and columns are indexed by $I_t S_t, A_t, B_t$ and $I_{t+1} S_{t+1}, A_{t+1}, B_{t+1}$ respectively. We now examine each block in this matrix and do a case-by-case analysis depending on which of the above conditions **Items 1 to 6** are satisfied.

1. In **Item 1**, if we rearrange the rows either according to groups of $\text{update}(S_t, i_t)$, the matrix is block-diagonal with respect to S_{t+1} since the non-zero entries correspond to $S_{t+1} = \text{update}_t(S_t, i_t)$. We now bound the operator norm of each block.
2. Suppose **Item 2** holds, i.e., $t \neq r_1, \dots, r_q, s_1, \dots, s_p, t_1 - 1, \dots, t_p - 1$, and the matrix is block-diagonal with respect to A_t, B_t since the non-zero entries correspond to $A_{t+1} = A_t, B_{t+1} = B_t$. Furthermore, each block is a sub-matrix of U_t (by the same argument as in the proof of **Lemma 3.1**). Hence, $\|\tilde{U}_t\|_{\text{op}} \leq 1$.
3. Suppose **Item 3** holds, i.e., t is equal r_j for some $j \in [q]$. We see that the non-zero entries correspond to $A_t = A_{t+1}$ and hence, the matrix is block-diagonal with respect to A_t . Since this is the first point in time that r_j is seen, we have enforced $B_t(j) = 0$ and $B_{t+1}(j) = i_t$

and for all other $j' \neq j$, we have $B_t(j') = B_{t+1}(j')$. This means that the rows can be rearranged into groups according to i_t which makes the matrix is block-diagonal with respect to B_{t+1} , furthermore, each block is a sub-matrix of U_t (by the same argument as in the proof of [Lemma 3.1](#)) and hence $\|\tilde{U}_t\|_{\text{op}} \leq 1$.

4. Similarly, if t is equal s_j for some $j \in [p]$, the analysis for is identical to the above one using [Item 4](#) and we obtain $\|\tilde{U}_t\|_{\text{op}} \leq 1$.
5. Suppose [Item 5](#) holds, i.e., $t+1 = t_j$ for some $j \in [p]$. From [Item 5](#), the matrix is block-diagonal with respect to B_t since we enforce $B_{t+1} = B_t$. Since $t+1 = t_j$, this means that we had already seen s_j by the start of the current step and had set $A_t(j) = i_{s_j}$. Therefore, when we enforce $A_t(j) = i_{t+1}$, we are enforcing $i_{t+1} = i_{s_j}$. This means that given columns labels A_{t+1} and i_{t+1} , it uniquely identifies a row label $A_t = \text{add}_j(A_{t+1}, i_{t+1})$. Thus, rearranging the columns according to groups of $\text{add}_j(A_{t+1}, i_{t+1})$, the matrix becomes block diagonal with respect to A_t as well. Furthermore, each block is a sub-matrix of U_t (by the same argument as in the proof of [Lemma 3.1](#)) and hence $\|\tilde{U}_t\|_{\text{op}} \leq 1$.
6. The analysis for [Item 6](#) involves carrying out [Item 4](#) (or [Item 3](#)) and [Item 5](#) one after the other.

Bounds on the Frobenius Norm. We now restrict our attention to the matrix $\tilde{U}^\emptyset$ obtained by taking the rows of $\tilde{U}$ corresponding to $S_1 = \emptyset$. Let $\tilde{U}_1^\emptyset$ be the matrix obtained by taking the rows of $\tilde{U}_1$ with $S_1 = \emptyset$. We have $\tilde{U}^\emptyset = \tilde{U}_1^\emptyset \cdot \tilde{U}_2 \cdots \tilde{U}_d$. Fix any $t \in [d]$. As before, since $\|\tilde{U}_t\|_{\text{op}} \leq 1$ for all $t \in [d]$, by [Fact 2.2](#), we have

$$\|\tilde{U}^\emptyset\|_{\text{frob}} \triangleq \|\tilde{U}_1^\emptyset \cdot \tilde{U}_2 \cdots \tilde{U}_d\|_{\text{frob}} \leq \|\tilde{U}_1^\emptyset \cdot \tilde{U}_2 \cdots \tilde{U}_t\|_{\text{frob}} \triangleq \|\tilde{U}_{[1,t]}^\emptyset\|_{\text{frob}}.$$

To control $\|\tilde{U}_{[1,t]}^\emptyset\|_{\text{frob}}$, we like to use the earlier approach of applying [Lemma 3.3](#) in reverse i.e., to the matrices $U_t^T, U_{t-1}^T, \dots, U_1^T$ in this order. To do this, we will need to define the new parameters s'_j, t'_j, r'_j that correspond to the reverse of s_j, t_j, r_j .

First, rearrange the (s_j, t_j) for $j \in [p]$ so that for the first p' pairs, we have $t_j \leq t+1$ (these are the pairs for which $\tilde{U}_{[1,t]}$ will have imposed equality constraints due to [Equation \(b\)](#)) and for the next q''' pairs, we have $s_j \leq t, t+1 < t_j$ (these are the elements that $\tilde{U}_{[1,t]}$ retains information about in A_{t+1} due to [Equation \(c\)](#)). Rearrange the r_j so that for the first q'' elements, we have $r_j \leq t$ (these are the elements that $\tilde{U}_{[1,t]}$ retains information about in B_{t+1} due to [Equation \(d\)](#)). Set $q' = q'' + q'''$. We will define $s'_1 > t'_1, \dots, s'_{p'} > t'_{p'}$ and $r'_1 > \dots > r'_{q'+q''}$ as follows.

- For $j \in [p']$ set $s'_j := t_j$ and $t'_j := s_j$.
- For $j \in [q'']$ set $r'_j = r_j$.
- For $j \in [q''']$ set $r'_{q''+j} := s_{p'+j}$.
- Set T' to be T .

We would like apply [Lemma 3.3](#) in reverse i.e., to the matrices $U_t^T, U_{t-1}^T, \dots, U_1^T$ in this order with these parameters. Set $S_{t+1} = \emptyset$. This gives us a matrix $\tilde{U}'$ such that

$$\tilde{U}'[I_{t+1}|I_1 S_1 B_1] = \sum_{I_2, \dots, I_t \in [M]} \left(\prod_{t'=t}^1 U_{t'}^T[I_{t'+1}|I_{t'}] \right) \cdot \mathbb{1} \left[S_1 = \bigoplus_{\substack{t' \in [2,t] \setminus T \\ i_{t'} \leq \tilde{N}}} \{i_{t'}\} \right]$$

$$\begin{aligned}
& \cdot \mathbb{1} \left[i_{s'_j} = i_{t'_j} \text{ for all } j \in [p'] \right] \\
& \cdot \mathbb{1} \left[B_1(j) = i_{r'_j} \text{ for all } j \in [q''] \right] \\
& \cdot \mathbb{1} \left[B_1(q'' + j) = i_{r'_{q''+j}} \text{ for all } j \in [q'''] \right] \quad (\text{by Lemma 3.3}) \\
& = \sum_{I_2, \dots, I_t \in [M]} \left(\prod_{t'=1}^t U_{t'}[I_{t'} | I_{t'+1}] \right) \cdot \mathbb{1} \left[S_1 = \bigoplus_{\substack{t' \in [2, t] \setminus T \\ i_{t'} \leq \tilde{N}}} \{i_{t'}\} \right] \\
& \cdot \mathbb{1} \left[i_{s_j} = i_{t_j} \text{ for all } j \in [p] \text{ with } t_j \leq t+1 \right] \\
& \cdot \mathbb{1} \left[B_1(j) = \begin{cases} i_{r_j} & \text{for } j \in [q] \text{ with } r_j \leq t \\ 0 & \text{otherwise.} \end{cases} \right] \\
& \cdot \mathbb{1} \left[B_1(q'' + j) = \begin{cases} i_{s_j} & \text{for } j \in [p] \text{ with } s_j \leq t, t+1 < t_j \\ 0 & \text{otherwise} \end{cases} \right] \quad (\text{by construction}) \\
& = \tilde{U}_{[1, t]}[I_1 | I_{t+1}, A'_1, B'_1] \quad (\text{by Equations (a) to (d)})
\end{aligned}$$

where B'_1 consists of B_1 restricted to coordinates $j \in [q''']$ and zeroes everywhere else and A'_1 consists of B_1 restricted to the coordinates $j \in [q'' + 1, q']$ and zeroes everywhere else. Observe that this gives a bijective correspondence between (B'_1, A'_1) and B_1 . Thus, we see that the entries of $\tilde{U}'$ and $\tilde{U}_{[1, t]}$ are the same, just arranged differently. Hence, $\|\tilde{U}'\|_{\text{frob}} = \|\tilde{U}_{[1, t]}\|_{\text{frob}}$. Consider

$$\|\tilde{U}'\|_{\text{frob}} \triangleq \|\tilde{U}'_t \cdot \tilde{U}'_{t-1} \cdots \tilde{U}'_1\|_{\text{frob}} \leq \|\tilde{U}'_t\|_{\text{frob}}.$$

We now recall the construction of $\tilde{U}'_t$ from Lemma 3.3. Recall that $\tilde{U}'_t$ satisfies for all $I_t, I_{t+1} \in [M], S_t \subseteq [M]$, we have

$$\tilde{U}'_t[I_{t+1} | I_t S_t A_t B_t] = \begin{cases} U_t^T[I_{t+1} | I_t] & \text{if } S_t = \emptyset, A_t = B_t = 0 \\ 0 & \text{otherwise.} \end{cases}$$

We see that $\tilde{U}'_t$ is identical to U_t^T when restricted to columns $S_t = \emptyset$ and $A_t = B_t = 0$ and zero on the other columns. This implies that $\|\tilde{U}'_t\|_{\text{frob}} = \|U_t\|_{\text{frob}}$ and completes the proof. $\square$

A.4 Acceptance Probability of Quantum Algorithms

DQC_k algorithms.

Proof of Claim 2.12. Consider a d -query DQC_k algorithm and let $U_1, \dots, U_{d+1}$ be the unitary operators of the algorithm and $\mathcal{S} = [NW] \times \{1\}, \mathcal{F} \subseteq [NWK]$ be the set of initial and accepting final states as in Definition 2.11 and Figure 1. The final state of the algorithm can be expressed as a uniform mixture over $I_1 \in \mathcal{S}$ of the pure state $U_{d+1} \cdot (O_x \otimes \mathbf{I}) \cdot U_d \cdots (O_x \otimes \mathbf{I}) \cdot U_1 |I_1\rangle$. Let $\mathcal{F} \subseteq [NWK]$ be the subset of final basis states that is accepted by the algorithm. We can thus express the acceptance probability of the algorithm as an average over $I_1 \in \mathcal{S}$ of

$$\sum_{I_{d+2} \in \mathcal{F}} |\langle I_{d+2} | U_{d+1} \cdot O \cdot U_d \cdots O \cdot U_1 | I_1 \rangle|^2$$

Since there are NW elements in $\mathcal{S}$, the overall acceptance probability of the algorithm is given by

$$\begin{aligned}
f(x) &:= \frac{1}{NW} \sum_{\substack{I_1 \in \mathcal{S} \\ I_{d+2} \in \mathcal{F}}} |\langle I_{d+2} | U_{d+1} \cdot O \cdot U_d \cdots O \cdot U_1 | I_1 \rangle|^2 \\
&= \frac{1}{NW} \sum_{\substack{I_1 \in \mathcal{S} \\ I_{d+2} \in \mathcal{F}}} \langle I_1 | U_1^\dagger \cdot O \cdots U_d^\dagger \cdot O \cdot U_{d+1}^\dagger | I_{d+2} \rangle \cdot \langle I_{d+2} | U_{d+1} \cdot O \cdot U_d \cdots O \cdot U_1 | I_1 \rangle \\
&= \frac{1}{NW} \sum_{\substack{I_1 \in \mathcal{S} \\ I_{d+2} \in \mathcal{F}}} \text{Tr} \left(U_1 | I_1 \rangle \langle I_1 | U_1^\dagger \cdot O \cdots U_d^\dagger \cdot O \cdot U_{d+1}^\dagger | I_{d+2} \rangle \langle I_{d+2} | U_{d+1} \cdot O \cdot U_d \cdots U_2 \cdot O \right) \\
&= \frac{1}{NW} \text{Tr} \left(U_1 \left(\sum_{I_1 \in \mathcal{S}} | I_1 \rangle \langle I_1 | \right) U_1^\dagger \cdot O \cdots O \cdot U_{d+1}^\dagger \left(\sum_{I_{d+2} \in \mathcal{F}} | I_{d+2} \rangle \langle I_{d+2} | \right) U_{d+1} \cdot O \cdots U_2 \cdot O \right).
\end{aligned}$$

We will further simplify this expression by introducing $M \times M$ matrices $V_1, \dots, V_{2d}$ as follows. Let $V_1 = \sum_{I_1 \in \mathcal{S}} U_1 | I_1 \rangle \langle I_1 | U_1^\dagger$. For $t \in [2, d]$, let $V_t := U_t^\dagger$. Let $V_{d+1} = \sum_{I_{d+2} \in \mathcal{F}} U_{d+1}^\dagger | I_{d+2} \rangle \langle I_{d+2} | U_{d+1}$ and for $t \in [d-1]$, let $V_{d+1+t} := U_{d-t+1}$. This allows us to express $f(x)$ as

$$f(x) = (NW)^{-1} \cdot \text{Tr} (V_1 \cdot O \cdot V_2 \cdot O \cdots V_{2d} \cdot O).$$

This gives us the desired expression. Finally we observe that $\|V_t\|_{\text{op}} \leq 1$ for all t , and V_1 is (up to multiplication by unitary matrices) equal to a diagonal matrix with at most $|\mathcal{S}| = NW$ non-zero entries of value 1, hence $\|V_1\|_{\text{frob}} \leq \sqrt{NW}$. $\square$

$\frac{1}{2}$ BQP algorithms.

Proof of Claim 2.14. Let $\mathcal{F}$ be the accepting pairs of initial and final states of a $\frac{1}{2}$ BQP algorithm and $U_1, \dots, U_{d+1}$ be unitary operators as in Definition 2.13 and Figure 2. For I_1, I_{d+1} , we use $F_{I_1, I_{d+1}}$ to denote 1 when $(I_1, I_{d+1}) \in \mathcal{F}$ and 0 otherwise. It is fairly straightforward to see that the acceptance probability $f(x)$ of the algorithm is given by

$$\begin{aligned}
f(x) &:= M^{-1} \sum_{I_1, I_{d+2} \in [M]} F_{I_1, I_{d+2}} \cdot |\langle I_{d+2} | U_{d+1} \cdot O \cdot U_d \cdots O \cdot U_1 | I_1 \rangle|^2 \\
&= M^{-1} \sum_{I_1, I_{d+2} \in [M]} F_{I_1, I_{d+2}} \cdot \langle I_1 | U_1^\dagger \cdot O \cdots O \cdot U_{d+1}^\dagger | I_{d+2} \rangle \langle I_{d+2} | U_{d+1} \cdot O \cdots O \cdot U_1 | I_1 \rangle
\end{aligned}$$

as desired. $\square$

BQP algorithms.

Proof of Claim 2.10. Let $U_1, \dots, U_{d+1}$ be the $M \times M$ unitary matrices applied by the algorithm and $F \subseteq [M]$ be the set of accepting final states as in Definition 2.9 and Figure 5. Let Π_F be the $M \times M$ diagonal matrix whose i -th entry is 0 if $i \notin F$ and 1 otherwise. Let $I_1 = |0 \dots 0\rangle$. Observe that the acceptance probability of the algorithm on input x is precisely

$$f(x) := \langle I_1 | U_1^\dagger \cdot O \cdots U_d^\dagger \cdot O \cdot U_{d+1}^\dagger \cdot \Pi_F \cdot U_{d+1} \cdot O \cdots O \cdot U_1 | I_1 \rangle$$

where $O = O_x \otimes \mathbf{I}$. Define matrices V_i for $i \in [2d+1]$ as follows. For $i \in [d]$, $V_i := U_i^\dagger$, $V_{d+1} = U_{d+1}^\dagger \cdot \Pi_F \cdot U_{d+1}$, and for $i \in [d]$, $V_{2d+2-i} = U_i$. Observe that $\|V_i\|_{\text{op}} \leq 1$ for all $i \in [2d+1]$, furthermore,

$$f(x) := \langle I_1 | V_1 \cdot O \cdots O \cdot V_{2d+1} | I_1 \rangle.$$

This completes the proof. $\square$

A.5 Fourier Coefficients of Quantum Algorithms

DQC_k Algorithms.

Proof of Claim 4.1. From Claim 2.12, the acceptance probability $f(x)$ of a d -query DQC_k algorithm is given by $f(x)$ where

$$\begin{aligned} f(x) &= (NW)^{-1} \cdot \text{Tr}((O_x \otimes \mathbf{I}) \cdot V_1 \cdots (O_x \otimes \mathbf{I}) \cdot V_{2d}) \\ &= (NW)^{-1} \sum_{I_1, \dots, I_{2d} \in [M]} \prod_{t \in [2d]} (V_t[I_t|I_{t+1}] \cdot x_{i_t}) \end{aligned} \quad (49)$$

with the convention that $I_{2d+1} = I_1$. We now replace x by $\rho(x)$ in Equation (49) to obtain

$$f(\rho(x)) = (NW)^{-1} \sum_{I_1, \dots, I_{2d} \in [M]} \prod_{t \in [2d]} (V_t[I_t|I_{t+1}] \cdot \rho(x)_{i_t}) \quad (50)$$

Since the first $\tilde{N}$ coordinates are unfixed and the rest are fixed,

$$\rho(x)_{i_t} = \begin{cases} x_{i_t} & \text{if } i_t \leq \tilde{N} \\ \rho_{i_t} & \text{if } i_t > \tilde{N} \end{cases}.$$

In particular,

$$\prod_{t \in [2d]} \rho(x)_{i_t} = \left(\prod_{\substack{t \in [2d] \\ \text{with } i_t > \tilde{N}}} \rho_{i_t} \right) \cdot \left(\prod_{\substack{t \in [2d] \\ \text{with } i_t \leq \tilde{N}}} x_{i_t} \right)$$

Substituting this in Equation (50), we get

$$f(\rho(x)) = (NW)^{-1} \sum_{I_1, \dots, I_{2d} \in [M]} \prod_{t \in [2d]} (V_t[I_t|I_{t+1}]) \cdot \left(\prod_{\substack{t \in [2d] \\ \text{with } i_t > \tilde{N}}} \rho_{i_t} \right) \cdot \left(\prod_{\substack{t \in [2d] \\ \text{with } i_t \leq \tilde{N}}} x_{i_t} \right). \quad (51)$$

To simplify this expression and get rid of the ρ_{i_t} , we will define a $M \times M$ diagonal matrix D^ρ and $M \times M$ unitary matrices V_t^ρ for $t \in [2d]$ as follows. For $I \in [M]$, define D^ρ to be a diagonal matrix whose I -th diagonal entry is ρ_i if $i > \tilde{N}$ and 1 otherwise. Define $V_t^\rho = D^\rho \cdot V_t$ for all $t \in [2d]$. Observe this allows us to simplify Equation (51) and obtain

$$f(\rho(x)) = (NW)^{-1} \sum_{I_1, \dots, I_{2d} \in [M]} \left(\prod_{t \in [2d]} V_t^\rho[I_t|I_{t+1}] \right) \cdot \left(\prod_{\substack{t \in [2d] \\ \text{with } i_t \leq \tilde{N}}} x_{i_t} \right)$$

From here, we see that the only non-zero Fourier coefficients correspond to $S \subseteq [\tilde{N}]$ and satisfy the defining equation as in Claim 4.1. The bounds on the norms of V_t^ρ follow immediately from the corresponding bounds on V_t from Claim 2.12 and the fact that $\|D^\rho\|_{\text{op}} \leq 1$. $\square$

$\frac{1}{2}$ BQP Algorithms.

Proof of Claim 5.1. Recall from Claim 2.14 that the acceptance probability of a d -query $\frac{1}{2}$ BQP algorithm is given by $f(x)$ where

$$f(x) := M^{-1} \sum_{I_1, I_{d+2} \in [M]} F_{I_1, I_{d+2}} \cdot \langle I_1 | U_1^\dagger \cdot O \cdots O \cdot U_{d+1}^\dagger | I_{d+2} \rangle \langle I_{d+2} | U_{d+1} \cdot O \cdots O \cdot U_1 | I_1 \rangle. \quad (52)$$

To simplify notation, for all $t \in [d+1]$, we define $V_t := U_t^\dagger$ and $V_{2d+3-t} = U_t$. Substituting this in Equation (52), we get

$$f(x) := M^{-1} \sum_{I_1, I_{d+2} \in [M]} F_{I_1, I_{d+2}} \sum_{\substack{I_2, \dots, I_{d+1} \in [M] \\ I_{d+3}, \dots, I_{2d+2} \in [M]}} \left(\prod_{t \in [2d+2]} V_t[I_t | I_{t+1}] \right) \cdot \left(\prod_{t \in [2d+2] \setminus \{1, d+2\}} x_{i_t} \right). \quad (53)$$

Substituting $\rho(x)$ in place of x in Equation (53), we get

$$\begin{aligned} f(\rho(x)) = M^{-1} \sum_{I_1, I_{d+2} \in [M]} F_{I_1, I_{d+2}} \sum_{\substack{I_2, \dots, I_{d+1} \in [M] \\ I_{d+3}, \dots, I_{2d+2} \in [M]}} & \left(\prod_{t \in [2d+2]} V_t[I_t | I_{t+1}] \right) \\ & \cdot \left(\prod_{\substack{t \in [2d+2] \setminus \{1, d+2\} \\ \text{with } i'_t > \tilde{N}}} \rho_{i'_t} \right) \cdot \left(\prod_{\substack{t \in [2d+2] \setminus \{1, d+2\} \\ \text{with } i_t \leq \tilde{N}}} x_{i_t} \right). \end{aligned} \quad (54)$$

As in the proof of Claim 4.1, we will simplify this expression by defining D^ρ to be a diagonal matrix whose I -th diagonal entry is 1 if $i \leq \tilde{N}$ and ρ_i otherwise and let $V_1^\rho = V_1$, $V_{d+2}^\rho = V_{d+2}$ and let $V_t^\rho = D^\rho \cdot V_t$ for $t \neq 1, d+2$. This allows us to simplify Equation (54) as

$$f(\rho(x)) = M^{-1} \sum_{I_1, I_{d+1} \in [M]} F_{I_1, I_{d+1}} \sum_{\substack{I_2, \dots, I_d \in [M] \\ I_{d+2}, \dots, I_{2d+2} \in [M]}} \prod_{t \in [2d]} V_t^\rho[I_t | I_{t+1}] \cdot \left(\prod_{\substack{t \in [2d] \setminus \{1, d+1\} \\ \text{with } i_t \leq \tilde{N}}} x_{i_t} \right).$$

From here, we see that only Fourier coefficients with $S \subseteq [\tilde{N}]$ are non-zero and are given by the defining equation in Claim 5.1. The norm bounds on V_t^ρ follow immediately from the corresponding bounds in Claim 2.14. This completes the proof. $\square$

BQP Algorithms.

Proof of Claim 6.1. Let $f(x)$ be the acceptance probability of a d -query BQP algorithm. We will derive an expression for the Fourier coefficients of f . Let $v = |0 \dots 0\rangle$. Recall from Claim 2.10 that

$$\begin{aligned} f(x) &:= v^\dagger \cdot V_1 \cdot O_x \cdots V_{2d} \cdot O_x \cdot V_{2d+1} \cdot v \\ &= \sum_{I_1, \dots, I_{2d+2}} v[I_1] \cdot \left(\prod_{t \in [2d+1]} V_t[I_t | I_{t+1}] \right) \cdot v[I_{2d+2}] \cdot \left(\prod_{t \in [2, 2d+1]} x_{i_t} \right) \end{aligned} \quad (55)$$

for matrices $V_1, \dots, V_{2d+1} \in \mathbb{C}^{M \times M}$ with spectral norm at most 1 and $v \in \mathbb{C}^M$ with $\|v\|_2 \leq 1$.

Let $\rho \in \{-1, 1, *\}^N$ be any restriction such that the first $\tilde{N}$ coordinates are free and the rest are fixed. Define an $M \times M$ diagonal matrix D^ρ exactly as in the proof of [Claim 4.1](#), i.e., the i -th entry is ρ_i if $i > \tilde{N}$ and 1 otherwise. Define $V_t^\rho = D^\rho \cdot V_t$ for $t \in [2, 2d+1]$ and $V_t^\rho = V_t$ for $t \in \{1, 2d+2\}$. With this notation, from [Equation \(55\)](#), we have

$$f|_\rho(x) = \sum_{I_1, \dots, I_{2d+2}} v[I_1] \cdot \left(\prod_{t \in [2d+1]} V_t^\rho[I_t | I_{t+1}] \right) \cdot v[I_{2d+2}] \cdot \left(\prod_{\substack{t \in [2, 2d+1] \\ i_t \leq \tilde{N}}} x_{i_t} \right).$$

From here, it is easy to see that the only non-zero Fourier coefficients of $f|_\rho$ correspond to $S \subseteq [\tilde{N}]$ and satisfy the defining equation in [Claim 6.1](#). The norm bounds on V_t^ρ follow immediately from the corresponding bounds on V_t from [Claim 2.10](#). $\square$