Quantum Search with Generalized Wildcards

Arjan Cornelissen
Simons Institute, University of California Berkeley, CA, USA

Nikhil S. Mande
School of Computer Science and Informatics, University of Liverpool, UK

Subhasree Patro
Department of Mathematics and Computer Science, Eindhoven University of Technology, The Netherlands

Nithish Raja
Department of Mathematics and Computer Science, Eindhoven University of Technology, The Netherlands

Swagato Sanyal
School of Computer Science, University of Sheffield, UK

Abstract

In the "search with wildcards" problem [Ambainis, Montanaro, Quantum Inf. Comput.'14], one's goal is to learn an unknown bit-string $x \in \{-1,1\}^n$. An algorithm may, at unit cost, test equality of any subset of the hidden string with a string of its choice. Ambainis and Montanaro showed a quantum algorithm of cost $O(\sqrt{n}\log n)$ and a near-matching lower bound of $\Omega(\sqrt{n})$. Belovs [Comput. Comp.'15] subsequently showed a tight $O(\sqrt{n})$ upper bound.

We consider a natural generalization of this problem, parametrized by a subset $\mathcal{Q} \subseteq 2^{[n]}$, where an algorithm may test whether $x_S = b$ for an arbitrary $S \in \mathcal{Q}$ and $b \in \{-1,1\}^S$ of its choice, at unit cost. We show the following:

- For all $k \in [n]$, when $\mathcal{Q}$ is the collection of all sets of size at most k, the quantum query complexity is $\Theta(n/\sqrt{k})$. In particular when $k = n$, this corresponds to the standard search with wildcards setting. This recovers and generalizes the tight characterization of Belovs, and Ambainis and Montanaro, using completely different techniques.
- When $\mathcal{Q}$ is the collection of contiguous blocks, the quantum query complexity is $\tilde{\Theta}(n)$.
- When $\mathcal{Q}$ is the collection of prefixes, the quantum query complexity is $\Theta(n)$.

All of these results are derived using a framework that we develop. We apply a symmetry reduction to the *primal* version of the negative-weight adversary bound, and show that the quantum query complexity of learning x is characterized, up to a constant factor, by a particular optimization program, which can be succinctly described as follows: 'maximize over all odd functions $f : \{-1,1\}^n \to \mathbb{R}$ the ratio of the maximum value of f to the maximum (over $T \in \mathcal{Q}$) standard deviation of f on a subcube whose free variables are exactly T.'

To the best of our knowledge, ours is the first work to use the primal version of the negative-weight adversary bound (which is a *maximization* program typically used to show lower bounds) to show new quantum query *upper* bounds without explicitly resorting to SDP duality.

2012 ACM Subject Classification Theory of computation → Quantum query complexity

Keywords and phrases quantum algorithms, quantum query complexity, adversary bound, symmetry reduction, substring queries

Digital Object Identifier 10.4230/LIPIcs.TQC.2026.3

Related Version *Full Version*: https://arxiv.org/abs/2511.04669

Funding *Arjan Cornelissen*: AC is supported by a Simons-CIQC postdoctoral fellowship through NSF QLCI Grant No. 2016245.
Nikhil S. Mande: NSM gratefully acknowledges support from the London Mathematical Society through a Scheme 7 grant, which enabled preliminary work that sparked this project.
Subhasree Patro: SP acknowledges the support from the Dutch Ministry of Education, Culture, and Science through Gravitation project "Challenges in Cyber Security – 024.006.037" for this work.

Nithish Raja: NR acknowledges the support from the Dutch Ministry of Education, Culture, and Science through Gravitation project "Challenges in Cyber Security – 024.006.037" for this work.

Acknowledgements The authors gratefully acknowledge OpenAI's ChatGPT (GPT-5), which assisted in developing several crucial ideas presented in this work. The authors bear full responsibility for any mistakes in the work. The authors also thank Nadezhda Voronova for pointing out that one of our lower bounds (when the query set is contiguous blocks) was already known, and for pointing us to the relevant references.

1 Introduction

Reconstructing an unknown string from limited access to its substrings is a natural and widely studied task across many areas of science and engineering. In computational biology, for example, genome assembly and DNA sequencing rely on reconstructing a long unknown strand from local substring measurements (see, for example, [22, 21, 9, 16] and the references therein). Beyond applied motivations, the problem provides a clean and mathematically tractable model for understanding the complexity of learning a hidden string via substring queries.

Most relevant to our work is the *search with wildcards* problem, introduced by Ambainis and Montanaro [1]. In this problem, the goal is to learn a hidden bitstring $x \in \{-1,1\}^n$ (i.e., an algorithm must correctly output x with high probability) using the minimum number of queries in the following *wildcard query* model. At unit cost, an algorithm may "check its guess for x" on any subset of the inputs. Formally, at unit cost, an algorithm may make a query (S,b) where $S \subseteq [n], b \in \{-1,1\}^S$. The query returns 1 if $x_S = b$, and 0 otherwise. Wildcard queries are a generalization of the usual query model, which only allows for $|S| = 1$. Classically even in the wildcard model every query provides at most 1 bit of information, yielding an information-theoretic lower bound of $\Omega(n)$ for learning x. In the usual quantum query model with $|S| = 1$, even though roughly a factor-2 speedup is possible [23], this is essentially the best speedup one can get [13].

Ambainis and Montanaro [1] showed that in the wildcard model, quantum algorithms can learn x using only $O(\sqrt{n}\log n)$ queries, exhibiting roughly a quadratic quantum speedup over the classical lower bound. However, their analysis and algorithm crucially rely on the ability to query *all subsets* of coordinates. In many practical settings, this assumption can be unrealistic. For instance, in *de novo* genome sequencing and related problems in computational biology [21], one has no promise on the input string, and the goal is to reconstruct it. In practical sequencing settings, the accessible measurements are limited to contiguous segments of the underlying strand; one cannot, for instance, simultaneously probe all odd positions at unit cost. Likewise, in more general settings one might be restricted to a fixed family of allowed subsets $\mathcal{Q} \subseteq 2^{[n]}$, and the structure of $\mathcal{Q}$ can significantly affect the power of the algorithm.

In this work, we study the quantum query complexities of learning an unknown string when the allowed query subsets are drawn from an arbitrary fixed collection $\mathcal{Q}$. This unifies previously studied models, including the standard query model $\mathcal{Q} = \{\{i\} : i \in [n]\}$, and the search-with-wildcards model $\mathcal{Q} = 2^{[n]}$ under a single framework.

1.1 Our Results

We refer the reader to Section 2 for formal definitions. We summarize here the notions that we need to state our main results. For an arbitrary collection of sets $\mathcal{Q} \subseteq 2^{[n]}$ and a function $F : \{-1,1\}^n \to E$, let $\mathsf{Q}^{\mathcal{Q}}(F)$ denote the quantum query query complexity of computing

F, where an algorithm can make unit-cost queries of the form $\mathbb{I}[x_S = b_S]$ for an arbitrary $S \in \mathcal{Q}$ and $b \in \{-1,1\}^S$ of its choice. We call $\mathsf{Q}^{\mathcal{Q}}(F)$ the *quantum $\mathcal{Q}$-substring query complexity of F*. Let $\oplus(\cdot)$ denote the parity function, i.e., $\oplus(x) = \prod_{i=1}^{n} x_i$ for $x \in \{-1,1\}^n$. A function $f : \{-1,1\}^n \to \mathbb{R}$ is said to be *odd* if $f(x) = -f(-x)$ for all $x \in \{-1,1\}^n$. We write $\mathbf{Var}(f) := \mathbf{Var}_{\mathbf{x}\sim\{-1,1\}^n}[f(\mathbf{x})]$ for the variance of the random variable $f(\mathbf{x})$, where $\mathbf{x}$ is sampled uniformly at random from $\{-1,1\}^n$.

Our main technical contribution is to show that the quantum $\mathcal{Q}$-substring query complexity of parity is characterized by the optimal value of an abstract analytic optimization program. Below, $f_{S|b}$ refers to the function f restricted to S by setting the bits in $\bar{S}$ to b.

▶ **Theorem 1.** *Let n be a positive integer and $\mathcal{Q} \subseteq 2^{[n]}$. Then $\mathsf{Q}^{\mathcal{Q}}(\oplus) = \Theta(\mathsf{val}_{\mathcal{Q}})$, where*

$$\mathsf{val}_{\mathcal{Q}} := \left(\max_{\substack{f:\{-1,1\}^n\to\mathbb{R} \\ f \text{ is an odd function}}} \frac{\|f\|_\infty}{\max\limits_{\substack{S\in\mathcal{Q}, \\ b\in\{-1,1\}^{\bar{S}}}} \sqrt{\mathbf{Var}(f_{S|b})}} \right). \tag{1}$$

In order to show this, we use the characterization of quantum query complexity by the negative-weight adversary bound [15, 20]. We show that for all $\mathcal{Q}$, the corresponding adversary bound *equals* $\mathsf{val}_{\mathcal{Q}}$ as defined above. We next analyze $\mathsf{val}_{\mathcal{Q}}$, as defined in Equation (1) for various interesting and natural settings of $\mathcal{Q}$ such as bounded-size sets, contiguous blocks, prefixes, and only the full set.

▶ **Theorem 2.** *Let n be a positive integer. Then, the following bounds hold for $\mathsf{val}_{\mathcal{Q}}$ for various classes of $\mathcal{Q} \subseteq 2^{[n]}$.*

Query Set	$\mathcal{Q}$	$\mathsf{val}_{\mathcal{Q}}$	*Reference*
Bounded-size sets	$\{S \subseteq [n] : \lvert S\rvert \leq k\}$	$\Theta\left(\frac{n}{\sqrt{k}}\right)$	*Theorem 16*
Contiguous blocks	$\{S = [i,j] : i \leq j \in [n]\}$	$\widetilde{\Theta}(n)$	*Theorem 18*
Prefixes	$\{[1,i] : i \in [n]\}$	$\Theta(n)$	*Full version of the paper*
Only full set	$\{[n]\}$	$2^{(n-1)/2}$	*Full version of the paper*

These results have immediate implications for the query complexity of the string-recovery problem in these query models. Let $\mathsf{rec} : \{-1,1\}^n \to \{-1,1\}^n$ be the (string recovery/learning) function defined by $\mathsf{rec}(x) = x$. Using Theorem 1, Theorem 2, and the Bernstein-Vazirani algorithm [6] to recover a bitstring in one query given query access to arbitrary parities of it, we obtain the following corollary.[1]

▶ **Corollary 3.** *Let n be a positive integer. Then, the quantum $\mathcal{Q}$-substring query complexity of learning an input string is characterized up to constants by $\mathsf{val}_{\mathcal{Q}}$.*

In particular, setting $k = n$ in the first bound above tightens the $O(\sqrt{n}\log n)$ algorithm of Ambainis and Montanaro [1] for quantum search with wildcards, and improves it to the optimal $O(\sqrt{n})$. We note that the optimal $O(\sqrt{n})$ bound in this case has already been shown by Belovs [3] using completely different techniques from ours. Additionally, our results in the prefix-query model prove that recovering any bitstring cannot be done with a sublinear

[1] The upper bounds for contiguous and prefix queries follow from the naive upper bound of querying one bit at a time (or an increasingly long prefix in the prefix query model) to learn an input string in $O(n)$ queries. The upper bound for full set queries follows from [14].

number of prefix queries, which resolves an open question raised in [24]. The result in the only full set case recovers the well-known tight $\Omega(\sqrt{N})$ lower bound for searching for a marked element in a database with N items [7] and the matching upper bound is given by [14].

A lower bound of $\Omega(n/\log(n))$, for learning the entire string, was shown by [11] and [8] when the query set is restricted to contiguous blocks. We recover this lower bound using our framework.

1.2 Our Techniques

We now discuss the techniques that we use to show Theorems 1 and 2. We then make a comparison with related work.

1.2.1 Sketch of Proof of Theorem 1

Our analysis begins from the characterization of quantum query complexity by the negative-weight adversary bound [20]. This characterization considers the quantum query complexity of a function $F : D \to E$, with respect to a query set $\mathcal{R}$, where every query $q \in \mathcal{R}$ is a function mapping D to a set of query outcomes. Reichardt [20] expresses the query complexity of F w.r.t. $\mathcal{R}$, up to a constant factor, in terms of the optimum value of a certain semidefinite optimization program known as the general adversary bound, which we refer to as the *primal adversary bound*, defined as

$$\mathrm{ADV}^{\mathcal{R},\pm}(F) := \max_{\Gamma \neq 0} \frac{\|\Gamma\|}{\max_{q \in \mathcal{R}} \|\Gamma_q\|}. \tag{2}$$

Here the maximization is over all matrices $\Gamma \in \mathbb{R}^{D \times D}$ with $F(x) = F(y) \implies \Gamma[x, y] = 0$, and the maximization in the denominator is over all available queries $\mathcal{R}$. For a query $q \in \mathcal{R}$, the matrix Γ_q is defined to be equal to Γ, with all those $[x, y]$ entries zeroed out where the outcome of the query q is the same on inputs x and y. This quantity was defined in [15], where it was shown to be a lower bound on quantum query complexity. Reichardt [20] subsequently showed that the query complexity of F with respect to query set $\mathcal{R}$ is equal to $\mathrm{ADV}^{\mathcal{R},\pm}(F)$ up to a constant factor.

It is well-known (see e.g., [19, Theorem 1.2]) that the negative-weight adversary bound can be written as a semidefinite program (SDP). We show in Lemma 6 that the feasible solutions of the maximization version of this SDP are directly related to the choices of Γ in Equation (2). It then follows that symmetry reductions on the SDP-level can be translated into additional symmetry constraints on the matrix Γ in Equation (2), without changing the optimal objective value. Note that attempting to naively apply such a symmetrization to the formulation in Equation (2) does not immediately work since, for example, the numerator could potentially reduce after symmetrizing. This is why all our symmetrization is done at the SDP level.

For the specific case where $F = \oplus$ and $\mathcal{R} = \{(S, b) : S \in \mathcal{Q}, b \in \{-1, 1\}^S\}$, with $\mathcal{Q} \subseteq 2^{[n]}$, we subsequently characterize in Lemmas 10 and 11 the symmetries in the primal adversary bound SDP, from which we deduce in Corollary 12 that a maximizing Γ can be assumed to be the communication matrix of an XOR function. That is, we can assume that $\Gamma = M_{f \circ \mathsf{XOR}}$ for some $f : \{-1, 1\}^n \to \mathbb{R}$, where $M_{f \circ \mathsf{XOR}}(x, y) := f(x \oplus y)$, i.e., the (x, y)'th entry only depends on the bitwise XOR of x and y. It is well known that such matrices are simultaneously diagonalizable by the Hadamard matrix, and hence diagonal in the Fourier basis, which motivates analyzing the numerator and denominators of Equation (2) using Fourier analysis.

It is known that the numerator of Equation (2) in this case, which is the spectral norm of $M_{f \circ \mathsf{XOR}}$, equals the (appropriately scaled) maximum Fourier coefficient of f [5]. For the denominator, we analyze the sparsity structure of the matrix Γ_q, for any query $(S, b) = q \in \mathcal{R}$. Through a series of calculations involving Fourier analysis of restricted Boolean functions, we show that the matrix Γ_q's spectral norm equals the (appropriately scaled) maximum, over all fixings of bits in $\bar{S}$, standard deviation of the restriction of f to the subcube that fixes these bits in $\bar{S}$. Putting everything together, we obtain $\mathsf{val}_{\mathcal{Q}} = \mathrm{ADV}^{\mathcal{R},\pm}(\oplus)$, which together with Reichardt's characterization [20] implies Theorem 1.

We provide a schematic illustration of the proof overview for Theorem 1 in Figure 1.

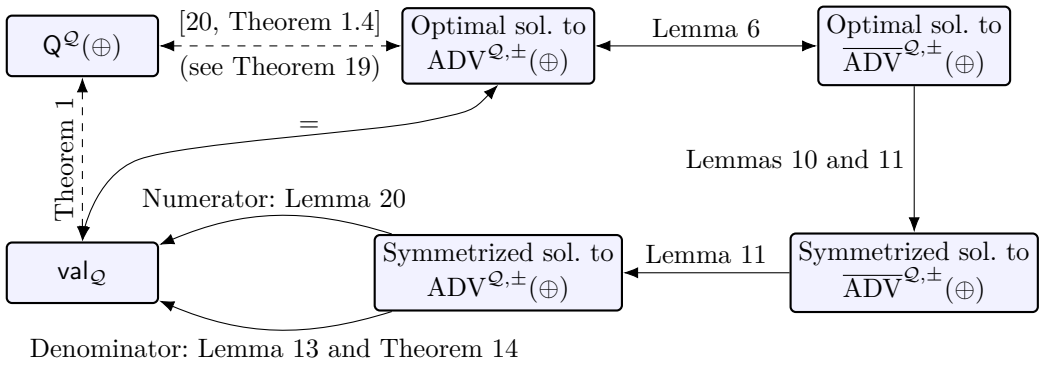

Figure 1 A schematic illustration to obtain our framework, which enables us to transform the problem of computing $\mathsf{Q}^{\mathcal{Q}}(\oplus)$, for an arbitrary collection of sets $\mathcal{Q} \subseteq 2^{[n]}$, into the task of analyzing an abstract analytic optimization program $\mathsf{val}_{\mathcal{Q}}$. The solid lines represent exact equalities, whereas the dashed lines represents characterizations up to constants.

1.2.2 Sketch of Proof of Theorem 2

Theorem 2 proves matching lower and upper bounds on $\mathsf{val}_{\mathcal{Q}}$ for four different classes of sets $\mathcal{Q}$. Note that $\mathsf{val}_{\mathcal{Q}}$ involves a maximization over all odd real-valued functions over the Boolean hypercube. The lower bounds are proven by exhibiting explicit functions for which every valid (for the class of $\mathcal{Q}$ considered) subcube restriction has low standard deviation with respect to the maximum absolute value of the function. The upper bounds are proven by showing that every odd real-valued function over the Boolean hypercube has a high standard deviation (with respect to its maximum absolute value) when restricted to some subcube that is valid for the class of $\mathcal{Q}$ considered. We now briefly discuss the proofs we give for various classes of $\mathcal{Q}$. In the main version of the paper we also discuss the "prefixes" and "only full set" classes, which are omitted in this version.

1.2.2.1 Bounded size

Theorem 16 deals with query sets of bounded size. The valid subcubes are the ones that fix at least $n - k$ variables. The lower bound is witnessed by the function that maps each bit-string to its Hamming weight minus $n/2$ (note that this is an odd function). For the upper bound, we first prove by an induction on the number of variables, that for every odd function there is some subcube (not necessarily valid) which simultaneously has a high variance and the absolute values of all its degree-1 Fourier coefficients are large. If this subcube is not valid (i.e., if it fixes less than $n - k$ variables), we argue via the probabilistic method and

Fourier analysis that the function can be further restricted by appropriately setting additional variables to a valid subcube, while simultaneously ensuring that the variance does not drop too much.

1.2.2.2 Contiguous blocks

Theorem 18 deals with query sets comprising consecutive variables (the same theorem statement holds true even if one is allowed wraparound). The valid subcubes are the ones that restrict consecutive variables. The upper bound of n follows from the observation that the query sets include all singletons, and Proposition 15 which verifies via the framework of Theorem 1 the obvious fact that if all the single bit queries are allowed, then the quantum query complexity is $O(n)$.

The lower bound is witnessed by a function f defined as follows. Given an input bit string of size n, view it as consisting of n/k disjoint blocks of size k, where $k = \log n - \log\log n + O(1)$. Let the blocks be denoted by $z^{(1)}, \ldots, z^{(n/k)}$. First, each $z^{(i)}$ is subjected to a function g that maps the string 1^k to 1, $(-1)^k$ to -1 and maps all other strings to 0. Finally f is defined to be $\sum_{i=1}^{n/k} g(z^{(i)})$. Note that each subcube setting contiguous variables (even if wraparound is allowed) touches, i.e., the endpoints of the block of queried indices are contained in, at most two of the blocks, so the restricted function is essentially a reduced version of f on a smaller number of blocks plus a bounded function on at most $2k$ bits. Thus, up to an additive shift by a bounded function, the (random variable corresponding to the) restriction of f is essentially the sum of independent random variables taking values in $\{0, 1, -1\}$ (the independence is due to the disjointness of the blocks), each of which is 0 except with probability $1/2^{k-1}$. Furthermore, the additional bounded summand is also independent of this sum and has $O(1)$ variance. We can thus apply linearity of variance for independent random variables to finish our argument.

1.3 Comparison With Related Work

The quantum $\mathcal{Q}$-substring query complexity of rec with $\mathcal{Q} = 2^{[n]}$ was introduced and studied by Ambainis and Montanaro [1], who dubbed this problem the "search with wildcards" problem and gave an $O(\sqrt{n}\log n)$ upper bound and a near matching $\Omega(\sqrt{n})$ lower bound. The algorithm by Ambainis and Montanaro uses the Pretty Good Measurement to iteratively learn $O(\sqrt{n})$ bits and fixes any errors made via binary search. In the search with wildcards setting, binary search can be performed with just $O(\log n)$ queries. Therefore, the algorithm learns the entire string in $O(\sqrt{n}\log n)$ queries. Setting $k = n$ in our Corollary 3 gives a tight $O(\sqrt{n})$ bound for this task. Ambainis and Montanaro also gave quantum query algorithms for the combinatorial group testing problem, which they observed to be a generalization of the $\mathcal{Q}$-substring query complexity of rec with $\mathcal{Q} = 2^{[n]}$. Subsequently, Belovs [3] gave, among other results, tight bounds on the quantum query complexity of combinatorial group testing, which implies a tight $O(\sqrt{n})$ upper bound on the quantum $2^{[n]}$-substring query complexity of rec. In contrast with the algorithms of Ambainis and Montanaro, Belovs showed a $O(\sqrt{n})$ upper bound on the dual of the negative-weight adversary bound. Since the negative-weight adversary bound is known to be asymptotically equal to quantum query complexity [20], this immediately gives an upper bound on $\mathsf{Q}^{\mathcal{Q}}(\mathsf{rec})$.

In contrast to the work of Ambainis and Montanaro [1] who construct an explicit algorithm, and that of Belovs [3] who constructs an explicit solution to the dual adversary bound, we deal, perhaps counter-intuitively, completely with the *primal* version of the negative-weight adversary bound [15]. Our work does have the drawback that it does not give explicit

algorithms for any of our upper bounds. However, as far as we are aware, ours is the first work to prove novel quantum query *upper* bounds using the *primal* version of the negative-weight adversary bound (which is primarily used as a lower bound technique) without explicitly resorting to SDP duality.

When $\mathcal{Q}$ is the set of all contiguous blocks, a lower bound of $\Omega(n/\log(n))$ was previously shown by [11, Lemma 7] via the positive-weight adversary method. The same lower bound, even for computing Parity, was also shown by [8, Theorem 19], where they lower bound the approximate degree and obtain the lower bound via the polynomial method [2]. Furthermore, their lower bound holds even in the setting where the success probability of the algorithm is inverse polynomially close to 1/2. In this work, we obtain the same $\Omega(n/\log(n))$ query lower bound (in the bounded-error setting) using our framework, using techniques different from the previous works.

Another work of relevance to ours is [4], who showed classical lower bounds for composed functions using quantum upper bounds for combinatorial group testing [3]. It is not immediately clear to us if our results have any new consequences for classical lower bounds using their framework, but our work can be viewed as partial progress towards answering their question [4, Section V] "Are there nontrivial quantum algorithms that use g-queries to learn x for some function $g \notin \{\mathsf{OR}, \mathsf{XOR}\}$? Are there nontrivial quantum algorithms that use g-queries to compute some other function $h(x)$ of the input?" Theorem 1 gives a complete framework that allows one to easily analyze the complexity of computing Parity (which is often equivalent to the complexity of learning the full input string) when g is restricted to subcube queries. Our framework gives provably tight bounds in this setting for all classes of allowed subcubes to query (with the constraint that if the algorithm is allowed to query if x is in a subcube whose fixed variables are S, all fixing of variables in S are allowed queries). The key advantage of our framework is that it replaces quantum and SDP-based reasoning with a simple, combinatorial analysis of real-valued functions on the Boolean hypercube. We justify this simpler viewpoint via Theorem 2, which establishes tight bounds within this framework for several natural classes of allowed subcube query sets.

1.4 Outlook

While our results do not apply in the combinatorial group testing setting, where only OR queries are allowed and the input string is promised to have bounded Hamming weight, we consider a generalization of the search with wildcards problem in a different direction, namely when a query algorithm is restricted on the support of the substring queries it can make. Our results are specifically of interest in the context of biological applications such as genome reconstruction, DNA sequencing, and RNA sequencing: In these models, a unit-cost task is often assumed to be observing a contiguous substring of the input string [22, 21, 9, 16]. Our second bound in Corollary 3 shows that quantum algorithms cannot provide an advantage over classical ones in this setting.

We view our work as an invitation to explore upper bounds through the *primal* adversary formulation itself, by studying the structural properties that optimal adversary matrices must satisfy. While this formulation can tend to be regarded as less convenient than its dual, our results illustrate that, with sufficient symmetrization and structural insight, the primal characterization can in fact yield tight bounds and reveal new patterns in how optimal Γ matrices arise. This suggests that it may be a more practical tool for upper bounds than previously thought.

2 Preliminaries

All logarithms in this paper are taken base 2. For a positive integer n and integers $i \leq j \in [n]$, we use the notation $[i,j] := \{k \in [n] : i \leq k \leq j\}$. For strings, $x, y \in \{-1,1\}^n$, we use the notation $x \oplus y$ to denote the bitwise XOR of x and y, or equivalently, the string obtained by taking component-wise multiplication of x and y. For a string $x \in \{-1,1\}^n$, we use the notation $|x|$ to denote the Hamming weight of x, i.e., the number of -1's in x. For a string $x \in \{-1,1\}^n$ and $S \subseteq [n]$, we use x_S to denote the string x restricted to the bits in S. For an expression X, we define $\mathbb{I}[X]$ to equal 1 if X is true, and 0 if X is false. For a string $x = (x_1, x_2, \ldots, x_n) \in \{-1,1\}^n$, we use the notation $-x$ to denote the string $(-x_1, -x_2, \ldots, -x_n)$. We denote vectors with the ket-notation, $|v\rangle$, and refer to their conjugate transpose with the bra-notation, i.e., $\langle v|$.

A function $f : \{-1,1\}^n \to \mathbb{R}$ is said to be an *odd function* if $f(x) = -f(-x)$ for all $x \in \{-1,1\}^n$. We abuse notation and use f to interchangeably mean a function as well as a random variable corresponding to picking a value of f uniformly at random. That is, for $f : \{-1,1\}^n \to \mathbb{R}$, we use $\mathbb{E}[f] := \mathbb{E}_{x \in \{-1,1\}^n}[f(x)]$, $\mathbb{E}[f^2] := \mathbb{E}_{x \in \{-1,1\}^n}[f(x)^2]$, and $\mathbf{Var}(f) := \mathbb{E}[f^2] - \mathbb{E}[f]^2$. We use $\|f\|_\infty$ to denote $\max_{x \in \{-1,1\}^n} |f(x)|$. For a positive integer n and a 2^n-dimensional real matrix M whose rows and columns are indexed by n-bit strings, we say that a M is the *communication matrix of an* XOR *function* if there exists a function $f : \{-1,1\}^n \to \mathbb{R}$ such that $M(x,y) = f(x \oplus y)$. We use the notation $M_{f \circ \mathsf{XOR}}$ to denote the communication matrix of $f \circ \mathsf{XOR}$.

A collection of sets $\mathcal{Q} \subseteq 2^{[n]}$ is said to be *downward closed* if for all $Q \in \mathcal{Q}$, all subsets of Q are also in $\mathcal{Q}$. For an n-dimensional square matrix M, we denote its spectral norm by $\|M\|$. We provide a brief overview of quantum algorithms and their query models in Appendix A, and we state the results from Fourier analysis that we need in Appendix B.

3 The Framework

In this section we build our framework that allows us to analyze the $\mathcal{Q}$-substring query complexity of learning input strings, for arbitrary $\mathcal{Q}$.

3.1 Learning vs. Parity

We first make the simple observation that for every collection of sets $\mathcal{Q} \subseteq 2^{[n]}$, the quantum query complexity of learning x is bounded from below by the quantum query complexity of computing the parity of all of the bits of x.

▶ **Observation 4.** *Let* $\mathcal{Q} \subseteq 2^{[n]}$. *Then* $\mathsf{Q}^{\mathcal{Q}}(\mathsf{rec}) \geq \mathsf{Q}^{\mathcal{Q}}(\oplus)$.

This immediately follows from the fact that once a string has been learned, the parity of its input bits can be computed with no extra queries. We remark that the same argument in fact yields $\mathsf{Q}^{\mathcal{Q}}(\mathsf{rec}) \geq \max_{S \subseteq [n]} \mathsf{Q}^{\mathcal{Q}}(\oplus_S)$, but we do not use this.

Next we observe that in the case when $\mathcal{Q}$ is downward closed, these complexities are actually equal.

▶ **Lemma 5.** *For all* $\mathcal{Q} \subseteq 2^{[n]}$ *that is downward closed, we have* $\mathsf{Q}^{\mathcal{Q}}(\mathsf{rec}) = \mathsf{Q}^{\mathcal{Q}}(\oplus)$.

Proof. Observation 4 provides the lower bound on $\mathsf{Q}^{\mathcal{Q}}(\mathsf{rec})$, so it remains to prove the upper bound. Note that we can recover a bit string if we are able to compute the parity on each of the subsets in superposition, by the Bernstein-Vazirani algorithm. Thus, it remains to prove that $\mathsf{Q}^{\mathcal{Q}}(\oplus_S) \leq \mathsf{Q}^{\mathcal{Q}}(\oplus)$, for all $S \subseteq [n]$.

To that end, let $S \subseteq [n]$. For any $x \in \{-1,1\}^n$, let $y(x) \in \{-1,1\}^n$ be the input that is equal to x on S, and identically 1 on $\bar{S}$. Then, $\oplus_S(x) = \oplus(y(x))$, so it remains to run the algorithm on $y(x)$. To that end, note that we can simulate a substring query (T, b) to $y(x)$ as follows. First, we check if b is identically 1 on $T \setminus S$, else we simply output 1 (False). Otherwise, we perform the query $(S \cap T, b_S)$ on x, and we observe that $y(x)_T = b$ if and only if $y(x)_{S\cap T} = b_S$. Since $\mathcal{Q}$ is downward-closed and $S \cap T \subseteq T$, we indeed have access to this query, and so we can simulate the parity-algorithm on $y(x)$ without additional overhead. ◀

We conclude that in order to characterize the $\mathcal{Q}$-complexity of bit string recovery, it suffices to characterize the $\mathcal{Q}$-complexity of computing the parity function, as long as $\mathcal{Q}$ is downward closed.

3.2 Simplification by Symmetry Reduction

In this subsection we perform a symmetry reduction on the primal adversary bound for the quantum $\mathcal{Q}$-substring quantum query complexity of computing parity, for an arbitrary $\mathcal{Q} \subseteq 2^{[n]}$. We show that the maximizing matrix Γ can be assumed to be the communication matrix of a XOR function, i.e., we may assume that $\Gamma = M_{f\circ\mathsf{XOR}}$ for some $f : \{-1,1\}^n \to \mathbb{R}$. Recall that $M_{f\circ\mathsf{XOR}}[x,y] = f(x \oplus y)$ for all $x, y \in \{-1,1\}^n$.

3.2.1 Alternate formulation of adversary bound for functions with Boolean output

Here we use an alternative formulation of the adversary bound that was shown by Cornelissen [12, Theorem 6.2.4]. This formulation allows us to average over optimal solutions without any loss in the optimal value. An important detail to note is that this formulation holds only for functions with Boolean output.[2]

Alternate formulation of primal adversary bound for functions with Boolean output

$$\begin{aligned}
\overline{\mathrm{ADV}}^{\mathcal{R},\pm}(F) = \quad & \max_{\Gamma,\beta} \quad \sum_{x,y\in D} \Gamma[x,y] && (3)\\
& \text{subject to} \quad \Gamma \in \mathbb{R}^{D\times D}, \beta \in \mathbb{R}^D_{\geq 0},\\
& \qquad \Gamma \text{ is symmetric},\\
& \qquad \Gamma[x,y] = 0, && (\forall\, x,y \in D \text{ with } F(x) = F(y))\\
& \qquad \mathsf{diag}(\beta) - \Gamma \circ \Delta_q \succeq 0, && (\forall q \in \mathcal{R})\\
& \qquad \sum_{x\in f^{-1}(1)} \beta[x] = \sum_{x\in f^{-1}(-1)} \beta[x] = \frac{1}{2}.
\end{aligned}$$

▶ **Lemma 6.** *Let $D \subseteq \{-1,1\}^n$ and $F : D \to \{-1,1\}$ be a Boolean function. Then $\mathrm{ADV}^{\mathcal{R},\pm}(F) = \overline{\mathrm{ADV}}^{\mathcal{R},\pm}(F)$. Moreover, if (β, Γ) is an optimal solution for $\overline{\mathrm{ADV}}^{\mathcal{R},\pm}(F)$, then Γ', defined as*[3]

$$\Gamma'[x,y] = \frac{\Gamma[x,y]}{\sqrt{\beta[x]\beta[y]}},$$

is an optimal solution for $\mathrm{ADV}^{\mathcal{R},\pm}(F)$.

[2] A similar optimization program can be derived for functions with non-Boolean output. Since its presentation is slightly more cumbersome and we don't need it in this work, we merely present the Boolean version here.

[3] Here, we use the convention that $0/0 = 0$.

The proof of this lemma essentially appears in the proof of [12, Theorem 6.2.4], but not with the exact formulation above. We provide a proof of Lemma 6 in the main version of the paper.

The benefit of this alternate formulation of the primal adversary bound is that it is a convex optimization problem. Therefore, it is always possible to take a linear combination of two feasible solutions, say with objective values c_1 and c_2, and obtain a new feasible solution with an objective value that is at least as big as $\min\{c_1, c_2\}$. We will use this to perform a symmetry reduction on the level of the SDP. In particular, we show how symmetries in the function and in the query set translate to additional constraints that we can impose on the optimal solutions to the alternate formulation of the primal adversary bound.

Let $\sigma : D \to D$ be a permutation of the domain D. We abuse notation to use σ to refer to the permutation function as well as the corresponding permutation matrix. It will be clear based on context whether σ is treated as a function or a matrix. When treated as a matrix, right-multiplying by σ^T corresponds to permuting columns according to the permutation σ and left-multiplying by σ corresponds to permuting rows according to σ.

Next, we define two types of symmetry that our query problem could exhibit.

▶ **Definition 7** (Function symmetry). *A permutation $\sigma : D \to D$ is said to be* function symmetric with respect to a function $F : D \to E$ *if* $F(\sigma(x)) = F(\sigma(y)) \iff F(x) = F(y)$, $\forall x, y \in D$.

▶ **Definition 8** (Query symmetry). *A permutation $\sigma : D \to D$ is said to be* query symmetric with respect to the set of queries $\mathcal{R}$ *if* $\{\sigma\Delta_q\sigma^T \mid q \in \mathcal{R}\} = \{\Delta_q \mid q \in \mathcal{R}\}$.

We now define a group that we use for the rest of this section and in the appendices.

▶ **Definition 9.** *Let G denote the group generated by bit-flip permutations, i.e., $G = \{\sigma_y \mid y \in \{-1,1\}^n\}$ where $\sigma_y : x \mapsto x \oplus y$.*

We now show that computing the parity function using $\mathcal{Q}$-substring queries exhibits these symmetries. The proofs of the following lemmas can be found in the full version.

▶ **Lemma 10.** *Let $n \in \mathbb{N}$ and $\mathcal{Q} \subseteq 2^{[n]}$. Then, every $\sigma \in G$ is function-symmetric w.r.t. $\oplus$ and query-symmetric w.r.t. $\mathcal{Q}$.*

▶ **Lemma 11.** *Let n be a positive integer and let $\mathcal{R}$ be a query set. Let G be the group generated by bit flip permutations. If every permutation $\sigma \in G$ is query symmetric with respect to $\mathcal{R}$, then there exists an $f : \{-1,1\}^n \to \mathbb{R}$ such that*

- *$f(x) = 0$ for all x of even Hamming weight, and*
- *$M_{f\circ\mathsf{XOR}}$ is an optimal solution for the adversary bound $\mathrm{ADV}^{\mathcal{R},\pm}(\oplus)$.*

▶ **Corollary 12.** *Let $n \in \mathbb{N}$ and $\mathcal{Q} \subseteq 2^{[n]}$. Then*

$$\mathrm{ADV}^{\mathcal{Q},\pm}(\oplus) = \max_{\substack{f:\{-1,1\}^n\to\mathbb{R} \\ |z| \text{ even} \implies f(z)=0}} \frac{\|M_{f\circ\mathsf{XOR}}\|}{\max\limits_{\substack{S\in\mathcal{Q}, \\ b\in\{-1,1\}^S}} \|M_{f\circ\mathsf{XOR}} \circ \Delta_{S,b}\|}.$$

Proof. We first show that $M_{f\circ\mathsf{XOR}}$ is a feasible solution to $\mathrm{ADV}^{\mathcal{Q},\pm}(\oplus)$, for all $f : \{-1,1\}^n \to \mathbb{R}$ satisfying $f(x) = 0$ when $|x|$ is even. The matrix $M_{f\circ\mathsf{XOR}}$ is symmetric since $M[x,y] = f(x \oplus y) = f(y \oplus x) = M[y,x]$ for all $x, y \in \{-1,1\}^n$. We know that $|x \oplus y|$ is even if both $|x|, |y|$ are even or if both $|x|, |y|$ are odd. Therefore,

$$\oplus(x) = \oplus(y) \implies |x \oplus y| \text{ is even} \implies M_{f\circ\mathsf{XOR}}[x,y] = 0.$$

Lemma 11 (which is applicable since Lemma 10 guarantees all elements of G to be query symmetric w.r.t. $\mathcal{Q}$ and function symmetric w.r.t. $\oplus$) yields the corollary. ◀

3.3 Simplification by Fourier Analysis

In this subsection we show that using the structure we derived in the last subsection, we can simplify the expression on the right-hand side of Corollary 12 to a concise analytic optimization problem over the Boolean hypercube. Lemma 20 says that the numerator equals $2^n \max_{S\subseteq[n]} |\widehat{f}(S)|$. For the denominator, we use the following lemma, whose proof is available in the main version of the paper.

▶ **Lemma 13.** *For all $f : \{-1,1\}^n \to \mathbb{R}$, and all $S \subseteq [n]$ and $b \in \{-1,1\}^S$, we have*

$$\|M_{f\circ\mathsf{XOR}} \circ \Delta_{S,b}\| = 2^{n-|S|} \max_{x\in\{-1,1\}^{\bar{S}}} \left\| \left(\widehat{f_{\bar{S}|y\oplus b}}(x)\right)_{y\in\{-1,1\}^S\setminus\{b\}} \right\|_2 .$$

Now, we can prove the required bound on the operator norm of the denominator in Corollary 12.

▶ **Theorem 14.** *For all $f : \{-1,1\}^n \to \mathbb{R}, S \subseteq [n]$ and all $b \in \{-1,1\}^S$, we have*

$$\|M_{f\circ\mathsf{XOR}} \circ \Delta_{S,b}\| = 2^n \max_{x\in\{-1,1\}^{\bar{S}}} \sqrt{\mathbf{Var}(\hat{f}_{S|x})}.$$

Proof. For $x \in \{-1,1\}^S$ and $y \in \{-1,1\}^{\bar{S}}$, let xy denote the bit string that equals x on S and y on $\bar{S}$. For the rest of this proof we identify a bitstring with its characteristic set. Specifically, this means for every $f : \{-1,1\}^n \to \mathbb{R}$, we use the notation $\hat{f}(x) := \hat{f}(\{j \in [n] : x_j = -1\})$. The spectral norm of a symmetric matrix with a single non-zero row is simply the norm of that row. We can thus use Lemma 13 to obtain that

$$\begin{aligned}
\|M_{f\circ\mathsf{XOR}} \circ \Delta_{S,b}\| &= 2^{n-|S|} \max_{x\in\{-1,1\}^{\bar{S}}} \left\| \left(\widehat{f_{\bar{S}|y\oplus b}}(x)\right)_{y\in\{-1,1\}^S\setminus\{b\}} \right\|_2 \\
&= 2^{n-|S|} \max_{x\in\{-1,1\}^{\bar{S}}} \sqrt{\sum_{y\in\{-1,1\}^S\setminus\{b\}} \widehat{f_{\bar{S}|y\oplus b}}(x)^2} \\
&= 2^{n-|S|} \max_{x\in\{-1,1\}^{\bar{S}}} \sqrt{\sum_{y\in\{-1,1\}^S\setminus\{1^S\}} \widehat{f_{\bar{S}|y}}(x)^2} \quad \text{by variable substitution} \\
&= 2^{n-|S|} \max_{x\in\{-1,1\}^{\bar{S}}} \sqrt{\sum_{y\in\{-1,1\}^S} \widehat{f_{\bar{S}|y}}(x)^2 - \widehat{f_{\bar{S}|1^S}}(x)^2} \\
&= 2^{n-|S|} \max_{x\in\{-1,1\}^{\bar{S}}} \sqrt{2^{|S|}\mathbb{E}_{y\in\{-1,1\}^S}[\widehat{f_{\bar{S}|y}}(x)^2] - \widehat{f_{\bar{S}|1^S}}(x)^2} \\
&= 2^{n-|S|} \max_{x\in\{-1,1\}^{\bar{S}}} \sqrt{2^{|S|}\sum_{y\in\{-1,1\}^S} \widehat{f}(xy)^2 - \left(\sum_{y\in\{-1,1\}^S} \widehat{f}(xy)\right)^2} \\
&\qquad \text{By Proposition 22 and Corollary 23} \\
&= 2^{n-|S|} \max_{x\in\{-1,1\}^{\bar{S}}} \sqrt{2^{2|S|}\mathbb{E}[\hat{f}^2_{S|x}] - 2^{2|S|}\mathbb{E}[\hat{f}_{S|x}]^2} \\
&= 2^n \max_{x\in\{-1,1\}^{\bar{S}}} \sqrt{\mathbf{Var}(\hat{f}_{S|x})}.
\end{aligned}$$

◀

Observe that the RHS in the theorem above is independent of b. Using this derived expression for the denominator, we are able to now state and prove Theorem 1. We refer to the main version of the paper for the formal version of the proof.

4 Applications of the Framework

In this section we use the framework developed in the last section to analyze $\mathsf{val}_{\mathcal{Q}}$, and thereby the quantum $\mathcal{Q}$-substring query complexity of computing parity, for various interesting collections $\mathcal{Q}$. We consider a wider variety of collections $\mathcal{Q}$ in the main version of the paper.

4.1 Contains All Singletons

In the case that the allowed substring queries include all singletons, there is an easy quantum query upper bound of n: simply query all of the variables one at a time. As a sanity check, we first show how this can be recovered by our framework.

▶ **Proposition 15.** *Let $\mathcal{Q} \supseteq \{\{i\} : i \in [n]\}$. Then $\mathsf{val}_{\mathcal{Q}} \leq n$.*

Proof. Since f is odd, there is always an x^* such that $f(x^*) = \|f\|_\infty > 0$. Let $M = \max_x f(x)$ be attained at x^*. As f is odd, $f(-x^*) = -M$.

Consider a shortest path from x^* to $-x^*$ in the hypercube, this has length n. Along this path the total value of f decreases by $2M$, thus there must exist an edge with a drop of at least $2M/n$. That edge corresponds to flipping one coordinate, say $i \in [n]$. This means there exists an $z \in \{-1,1\}^n$ such that $|f(z) - f(z \oplus e_i)| \geq 2M/n$. Choose the subcube that fixes all variables but i to be consistent with z, and leave the i'th variable free. Note that this is a valid subcube to consider as $\mathcal{Q}$ contains all singletons. For this choice of subcube we have the variance (using the formula $\mathbf{Var}(g) = \frac{1}{2}\mathbb{E}[(g(x) - g(y))^2]$) to be $\frac{(f(z)-f(z\oplus e_i))^2}{4} \geq \frac{4M^2}{4n^2}$. Thus $\mathsf{val}_{\mathcal{Q}} \leq \frac{M}{\sqrt{4M^2/4n^2}} = n$. ◀

4.2 Sets of Bounded Size

The setting $\mathcal{Q} = \left\{S \subseteq 2^{[n]} : |S| \leq k\right\}$ corresponds to the model where the allowed substrings to be queried are bounded in size.

In particular, $k = n$ corresponds to the usual search with wildcards setting, where an algorithm is allowed to query OR's of arbitrary sets of literals.

▶ **Theorem 16.** *Let $1 \leq k \leq n$ be an integer, and let $\mathcal{Q} = \left\{S \subseteq 2^{[n]} : |S| \leq k\right\}$. Then,*

$$\frac{n}{\sqrt{k}} \leq \mathsf{val}_{\mathcal{Q}} \leq \frac{2n}{\sqrt{k}}.$$

We show the lower bound using the (appropriately shifted) Hamming weight function. We prove the upper bound by induction on n. We first state a convenient induction hypothesis, then show how it implies the upper bound, and finally prove the induction hypothesis.

▶ **Lemma 17.** *For all integers $n > 0$ and all functions $f : \{-1,1\}^n \to \mathbb{R}$ with $\mathbb{E}[f(x)] = 0$, there exists a $S \subseteq [n]$ and $b \in \{-1,1\}^{\bar{S}}$ where:*

- $\mathbf{Var}(f_{S|b}) \geq \frac{\|f\|_\infty^2}{4n}$, *and*
- $\left|\widehat{f_{S|b}}(\{i\})\right| \geq \frac{\|f\|_\infty}{2n}$ *for all $i \in S$.*

In other words, there exists a subcube where the restricted function has large variance, as well as large degree-1 Fourier coefficients. We feel this is an independently interesting result in Fourier analysis. We first show how Theorem 16 follows from Lemma 17 (note that Lemma 17 gives no guarantee on $|S|$, which we require for proving Theorem 16).

Proof of Theorem 16. We first show the lower bound on $\mathsf{val}_{\mathcal{Q}}$, and then the upper bound.

- **Lower bound:** Consider the function $f : \{-1,1\}^n \to \mathbb{R}$ defined by $f(x) = |x| - (n/2)$. It is clear that f is an odd function since $f(-x) = |-x| - (n/2) = (n - |x|) - (n/2) = n/2 - |x| = -f(x)$. Let $m \leq k$. For any subcube that fixes all but a collection of m variables, it is easy to see that the random variable obtained by restricting f to this subcube and picking a uniformly random function value in the restricted subcube is simply a constant additive shift of the binomial random variable $\mathrm{Bin}(m, 1/2)$. Since such shifts do not affect variance, the variance in this subcube equals $m/4 \leq k/4$, since the only subcubes under consideration are those where the number of free variables is at most k. Since $\max |f(x)| = n/2$, this function f witnesses

$$\mathsf{val}_{\mathcal{Q}} \geq \frac{n/2}{\sqrt{k/4}} = \frac{n}{\sqrt{k}}.$$

- **Upper bound:** Let $M := \|f\|_\infty = \max_{x\in\{-1,1\}^n} |f(x)|$. Let $S \subseteq [n]$ and $b \in \{-1,1\}^{\bar{S}}$ be obtained from Lemma 17 (which is applicable since f is odd, and hence balanced). If $|S| \leq k$, then this is a valid subcube that can be chosen in the denominator of $\mathsf{val}_{\mathcal{Q}}$, and we get $\mathsf{val}_{\mathcal{Q}} \leq \frac{M}{\sqrt{M^2/4n}} = 2\sqrt{n} \leq \frac{2n}{\sqrt{k}}$, which proves the theorem. So for the rest of the proof we may assume that $|S| > k$. For convenience, define $g := f_{S|b} : \{-1,1\}^S \to \mathbb{R}$. Let T be any subset of S of size k, and define $\bar{T} := S \setminus T$ (i.e., the complement of T with respect to the universe S). For an arbitrary $x \in \{-1,1\}^{\bar{T}}$, consider the function $g_{T|x} : \{-1,1\}^T \to \mathbb{R}$ obtained by restricting g further by setting the variables in $\bar{T}$ to x. We have for each $i \in T \subseteq S$,

$$\mathbb{E}_x[\widehat{g_{T|x}}(\{i\})^2] = \sum_{A\subseteq\bar{T}} \widehat{g}(A \cup \{i\})^2 \geq \widehat{g}(\emptyset \cup \{i\})^2 = \widehat{g}(\{i\})^2 \geq \frac{M^2}{4n^2},$$

where the equality is due to Corollary 23 and the last inequality holds by Lemma 17. By linearity of expectation, and since $|T| = k$, we have

$$\mathbb{E}_x\left[\sum_{i\in T} \widehat{g_{T|x}}(\{i\})^2\right] = \sum_{i\in T} \mathbb{E}_x\left[\widehat{g_{T|x}}(\{i\})^2\right] \geq \frac{M^2 k}{4n^2}.$$

Therefore there exists a $x^* \in \{-1,1\}^{\bar{T}}$ such that $\sum_{i\in T}\left(\widehat{g_{T|x^*}}(\{i\})^2\right) \geq \frac{M^2k}{4n^2}$. Since $\mathbf{Var}(h) = \sum_{S\neq\emptyset} \widehat{h}(S)^2$ for all functions h by Lemma 25, we have

$$\mathbf{Var}(g_{T|x^*}) = \sum_{S\subseteq T, S\neq\emptyset} \widehat{g_{T|x^*}}(S)^2 \geq \sum_{i\in T} \widehat{g_{T|x^*}}(\{i\})^2 \geq \frac{M^2k}{4n^2}.$$

Note that $g_{T|x^*}$ is the restriction of f to the subcube T, by setting the variables in $\bar{S}$ and $\bar{T}$ to b and x^*, respectively. This is a valid subcube for the denominator of $\mathsf{val}_{\mathcal{Q}}$ (see Equation (1)), and so this subcube witnesses

$$\mathsf{val}_{\mathcal{Q}} \leq \frac{M}{\sqrt{\frac{M^2k}{4n^2}}} = \frac{2n}{\sqrt{k}},$$

which completes the proof. ◀

It only remains to prove our induction hypothesis, Lemma 17, which we do now.

Proof of Lemma 17. We prove the lemma by induction on n. Let $\|f\|_\infty = M$ and assume that $|f(x^*)| = M$.

- **Base case**: $n = 1$. Thus we have $f(-1) = M$ and $f(1) = -M$ (or the opposite, in which case essentially the same proof works). We have the variance over the whole cube as $\mathbf{Var}(f) = M^2 > \frac{M^2}{4}$. We also have $\widehat{f}(\{1\}) =$ either M or $-M$. In either case, $|\widehat{f}(\{1\})| = M \geq \frac{M}{2}$. This proves the base case.
- **Assumption for inductive step**: We assume the induction hypothesis to be true for $n = \ell$, i.e., for every balanced function f on ℓ variables, there exists a $S \subseteq [\ell]$ and $b \in \{-1,1\}^{\bar{S}}$ such that $\mathbf{Var}(f_{S|b}) \geq \frac{M^2}{4\ell}$, and $\left|\widehat{f_{S|b}}(\{i\})\right| \geq \frac{M}{2\ell}$ for all $i \in S$.
- **Inductive step**: For the inductive step, we assume that the hypothesis holds for $n = \ell$ (see above) and show it is true for $n = \ell + 1$. Define $\mu_{i,b} := \mathbb{E}_{x:x_i=b}[f(x)]$ for $b \in \{-1,1\}$. We consider two cases.
  - **Case 1:** $\exists i \in [\ell+1]$ such that $|\mu_{i,x_i^*} - \mu_{i,-x_i^*}| \leq \frac{M}{\ell+1}$. Since we know f to be balanced, we have $\mu_{i,x_i^*} + \mu_{i,-x_i^*} = 0$, and hence by the triangle inequality

$$|\mu_{i,x_i^*}| = \left|\frac{\mu_{i,x_i^*} - \mu_{i,-x_i^*}}{2} + \frac{\mu_{i,x_i^*} + \mu_{i,-x_i^*}}{2}\right| \leq \frac{|\mu_{i,x_i^*} - \mu_{i,-x_i^*}|}{2} + \frac{|\mu_{i,x_i^*} + \mu_{i,-x_i^*}|}{2} \leq \frac{M}{2(\ell+1)}.$$

    Define $f_i := f|_{x_i=x_i^*} - \mu_{i,x_i^*}$. Note that we have $\mathbb{E}[f_i] = 0$ and $\|f_i\|_\infty \geq M - \frac{M}{2(\ell+1)}$, where we used that the maximum absolute value of f is attained in the domain of f_i by construction (i.e., x^* is in the domain of f_i). Applying the induction hypothesis to f_i, we get a $S \subseteq [\ell+1]$ and $b \in \{-1,1\}^{\bar{S}}$ (with $i \in \bar{S}$ and $b_i = x_i^*$) such that the following two properties hold.
    - We write $b' := b|_{\bar{S}\setminus\{i\}}$, i.e., the bit string b without the bit at position i, and obtain

$$\mathbf{Var}((f_i)_{S|b'}) \geq \frac{\left(M\left(1 - \frac{1}{2(\ell+1)}\right)\right)^2}{4\ell} \geq \frac{\left(M\left(\sqrt{1 - \frac{1}{\ell+1}}\right)\right)^2}{4\ell} = \frac{M^2}{4\ell} \cdot \frac{\ell}{\ell+1} = \frac{M^2}{4(\ell+1)}.$$

      Here we are using $1 - \frac{1}{2\alpha} \geq \sqrt{1 - \frac{1}{\alpha}}$ for all $\alpha > 1$, which can easily be verified by squaring both sides and comparing terms. Since $f_{S|b} = (f_i)_{S|b'} + \mu_{i,x_i^*}$ and variance is unaffected by a constant additive shift, this implies $\mathbf{Var}(f_{S|b}) \geq \frac{M^2}{4(\ell+1)}$ as well.
    - First note that for any function $g : \{-1,1\}^n \to \mathbb{R}$ and any constant c, all Fourier coefficients of g are equal to the corresponding Fourier coefficients of $g - c$, except for the empty coefficient which gets subtracted by c. Thus, we again write $b' := b|_{\bar{S}\setminus\{i\}}$, to find that for all $j \in S$,

$$|\widehat{f_{S|b}}(\{j\})| = |\widehat{(f_i)_{S|b'}}(\{j\})| \geq \frac{M\left(1 - \frac{1}{2(\ell+1)}\right)}{2\ell} = \frac{M \cdot \frac{2\ell+1}{2(\ell+1)}}{2\ell} > \frac{M \cdot 2\ell}{2(\ell+1)} \cdot \frac{1}{2\ell} = \frac{M}{2(\ell+1)}.$$

  - **Case 2:** For all $i \in [\ell+1]$, we have $|\mu_{i,x_i^*} - \mu_{i,-x_i^*}| > \frac{M}{\ell+1}$. In that case, we choose $S = [\ell+1]$ and b becomes the empty string, so that $f_{S|b} = f$. Then, we find
    - By definition of Fourier coefficients, we have for all $i \in [\ell+1]$,

$$|\widehat{f}(\{i\})| = |\mathbb{E}_x[f(x) \cdot x_i]| = \left|\frac{x_i^*}{2}(\mu_{i,x_i^*} - \mu_{i,-x_i^*})\right| > \frac{M}{2(\ell+1)}.$$

    - By Lemma 25, we have the variance of the whole subcube as

$$\mathbf{Var}(f) = \sum_{S \neq \emptyset} \widehat{f}(S)^2 \geq \sum_{i \in [\ell+1]} \widehat{f}(\{i\})^2 > (\ell+1)\frac{M^2}{4(\ell+1)^2} = \frac{M^2}{4(\ell+1)}.$$

◀

4.3 Contiguous Blocks

The setting $\mathcal{Q} = \{[i,j] : i \leq j \in [n]\}$ corresponds to the model where the allowed substrings to be queried form contiguous blocks. While we don't phrase it as such, our results in this subsection also hold when the notion of contiguity allows for wraparound.

The following is our main result of this subsection.

▶ **Theorem 18.** *Let n be a positive integer and let $\mathcal{Q} = \{[i,j] : i \leq j \in [n]\}$. Then there exists a universal constant c such that*

$$\frac{cn}{\log n} \leq \mathsf{val}_{\mathcal{Q}} \leq n.$$

Proof. Since $\mathcal{Q}$ contains all singletons, Proposition 15 implies the upper bound. For the lower bound, define $f : \{-1,1\}^{mk} \to \mathbb{R}$ as $f(z^{(1)}, \ldots, z^{(m)}) = \sum_{i=1}^{m} g(z^{(i)})$. Here each $z^{(i)} \in \{-1,1\}^k$, and

$$g(z^{(i)}) = \begin{cases} 1, & z^{(i)} = 1^k, \\ -1, & z^{(i)} = (-1)^k, \\ 0, & \text{otherwise.} \end{cases}$$

First note that f is odd and $\max |f(x)| = m = n/k$. We show that every valid subcube in the denominator of the RHS of the expression of $\mathsf{val}_{\mathcal{Q}}$ (see Equation (1)) has small variance, which proves the theorem.

For any subcube (S,b) where S is a contiguous block, since a contiguous block can partially intersect at most 2 blocks (let the free variables in these blocks be denoted by strings $y^{(1)}$ and $y^{(2)}$), we can assume that the restriction of f to the subcube (S,b) has the following form:

$$f_{S|b}(y^{(1)}, y^{(2)}, z^{(1)}, z^{(2)}, \ldots z^{(t)}) = h_1(y^{(1)}) + h_2(y^{(2)}) + \sum_{i=1}^{t} g(z^{(i)}) + c, \tag{4}$$

where $h_1, h_2 \in \{r_1, r_{-1}, r_0\}$ depending on the values of the set variables in the corresponding blocks, where for all $b \in \{-1,0,1\}$,

$$r_b(y) := \begin{cases} b, & \text{if } y \text{ is the all-}b \text{ string,} \\ 0, & \text{otherwise.} \end{cases}$$

In particular, r_0 is the constant 0 function. Since each block in the restricted function contains disjoint variables, the variance of the sum on the RHS of Equation (4) equals the sum of variances of the individual terms, which equals

$$= \mathbf{Var}(h_1) + \mathbf{Var}(h_2) + \sum_{i=1}^{t} \mathbf{Var}(g) + 0 = O(1) + t \cdot \mathbb{E}[g^2] = O(1) + m \cdot \frac{1}{2^{k-1}},$$

where the second equality follows since since h_i is a $\{-1,0,1\}$-valued function, and each g is a balanced function, and the last equality follows since $t \leq m$ and g^2 takes value 1 at precisely two points and 0 everywhere else.

If we set n to be the number of variables for f, which equals mk, then the above variance equals $O(1) + \frac{n/k}{2^{k-1}}$. Thus we have

$$\mathsf{val}_{\mathcal{Q}} = \frac{\|f\|_\infty}{\sqrt{\max\limits_{\substack{S \in \mathcal{Q}, \\ b \in \{-1,1\}^{\bar{S}}}} \mathbf{Var}(f_{S|b})}} = \frac{n/k}{\sqrt{O(1) + \frac{n/k}{2^{k-1}}}} = \Omega\left(\min\left\{\frac{n}{k}, \sqrt{\frac{n}{k}} \cdot 2^{\frac{k-1}{2}}\right\}\right).$$

Setting k to maximize this quantity asymptotically, we get $\frac{n}{k} = 2^{k-1}$, which implies $n = k \cdot 2^{k-1}$, implying $k = \log n - \log\log n + O(1)$, and hence $\mathsf{val}_{\mathcal{Q}} = \Omega(n/\log n)$, proving the theorem. ◀

5 Conclusions

This work introduces a general framework for analyzing query algorithms for learning input strings in models that extend the search with wildcards setting [1]. As applications of our framework we are able to recover existing upper bounds [1, 3] in the search-with-wildcards setting and also show new bounds in generalized search-with-wildcards settings. Interestingly, our upper bounds in this paper are all shown (without explicitly resorting to SDP duality) using the *primal* version of the negative-weight adversary bound [15], which is a maximization SDP, typically used to show *lower bounds*. To the best of our knowledge, ours is the first work to use the primal adversary bound to establish novel upper bounds. Even though some of our results were already known from previous works [1, 3, 11, 8] through various differing techniques, we are able to recover the same bounds through a unifying framework. Our work makes progress towards answering an open question of [4], who asked if one could devise algorithms for learning input strings or computing some function of the input string, assuming non-standard query access to the input. It would be interesting to see if our results can be used to show any classical lower bounds using the framework developed in [4].

An interesting related direction is that of the AND/OR-query complexity of computing Boolean functions rather than learning an input string. It is not hard to show that the query model in the search-with-wildcards setting is equivalent to allowing unit-cost query access to arbitrary ORs and arbitrary ANDs of inputs. It was recently shown [10] that randomness does not give a superpolynomial advantage over determinism for computing total Boolean functions in this model. It would be interesting to see if ideas from our work could be generalized to show that quantumness also does not provide a superpolynomial advantage over randomness/determinism. Our framework relies on the optimizing matrix for the primal adversary bound being the communication matrix of an XOR function, which crucially uses symmetries of Parity (the function being computed), so it is not clear how to generalize our framework in this direction easily.

References

1 Andris Ambainis and Ashley Montanaro. Quantum algorithms for search with wildcards and combinatorial group testing. *Quantum Inf. Comput.*, 14(5-6):439–453, 2014. doi:10.26421/QIC14.5-6-4.

2 Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. *J. ACM*, 48(4), 2001. doi:10.1145/502090.502097.

3 Aleksandrs Belovs. Quantum algorithms for learning symmetric juntas via the adversary bound. *Comput. Complex.*, 24(2):255–293, 2015. doi:10.1007/S00037-015-0099-2.

4 Shalev Ben-David, Adam Bouland, Ankit Garg, and Robin Kothari. Classical lower bounds from quantum upper bounds. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 339–349. IEEE, 2018. doi:10.1109/FOCS.2018.00040.

5 Anna Bernasconi and Bruno Codenotti. Spectral analysis of boolean functions as a graph eigenvalue problem. *IEEE transactions on computers*, 48(3):345–351, 2002.

6 Ethan Bernstein and Umesh V. Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997. Earlier version in STOC'93. doi:10.1137/S0097539796300921.

7 Michel Boyer, Gilles Brassard, Peter Høyer, and Alain Tapp. Tight bounds on quantum searching. *Fortschritte der Physik: Progress of Physics*, 46(4-5):493–505, 1998.

8 Mark Bun and Nadezhda Voronova. Approximate degree lower bounds for oracle identification problems, 2023. doi:10.48550/arXiv.2303.03921.

9 Mark JP Chaisson, Richard K Wilson, and Evan E Eichler. Genetic variation and the de novo assembly of human genomes. *Nature Reviews Genetics*, 16(11):627–640, 2015.

10 Arkadev Chattopadhyay, Yogesh Dahiya, Nikhil S. Mande, Jaikumar Radhakrishnan, and Swagato Sanyal. Randomized versus deterministic decision tree size. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing (STOC)*, pages 867–880. ACM, 2023. doi:10.1145/3564246.3585199.

11 Richard Cleve, Kazuo Iwama, François Le Gall, Harumichi Nishimura, Seiichiro Tani, Junichi Teruyama, and Shigeru Yamashita. Reconstructing strings from substrings with quantum queries. In Fedor V. Fomin and Petteri Kaski, editors, *Algorithm Theory – SWAT 2012*. Springer Berlin Heidelberg, 2012.

12 Arjan Cornelissen. *Quantum multivariate estimation and span program algorithms*. University of Amsterdam, 2023.

13 Edward Farhi, Jeffrey Goldstone, Sam Gutmann, and Michael Sipser. Bound on the number of functions that can be distinguished with k quantum queries. *Physical Review A*, 60(6):4331, 1999.

14 Lov K Grover. A fast quantum mechanical algorithm for database search. In *Proceedings of the twenty-eighth annual ACM Symposium on Theory of Computing (STOC)*, pages 212–219, 1996. doi:10.1145/237814.237866.

15 Peter Høyer, Troy Lee, and Robert Špalek. Negative weights make adversaries stronger. In *Proceedings of the thirty-ninth annual ACM Symposium on Theory of Computing (STOC)*, pages 526–535, 2007.

16 Heng Li and Richard Durbin. Genome assembly in the telomere-to-telomere era. *Nature Reviews Genetics*, 25(9):658–670, 2024.

17 Nikhil Shekhar Mande. *Communication complexity of XOR functions*. PhD thesis, PhD thesis, Tata Institute of Fundamental Research Mumbai, 2018.

18 Ryan O'Donnell. *Analysis of boolean functions*. Cambridge University Press, 2014.

19 Ben W. Reichardt. Span programs and quantum query complexity: the general adversary bound is nearly tight for every Boolean function. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science—FOCS*, pages 544–551. IEEE Computer Soc., Los Alamitos, CA, 2009. doi:10.1109/FOCS.2009.55.

20 Ben W. Reichardt. Reflections for quantum query algorithms. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 560–569. SIAM, Philadelphia, PA, 2011. doi:10.1137/1.9781611973082.44.

21 Gordon Robertson, Jacqueline Schein, Readman Chiu, Richard Corbett, Matthew Field, Shaun D Jackman, Karen Mungall, Sam Lee, Hisanaga Mark Okada, Jenny Q Qian, et al. De novo assembly and analysis of RNA-seq data. *Nature methods*, 7(11):909–912, 2010.

22 Steven S Skiena and Gopalakrishnan Sundaram. Reconstructing strings from substrings. *Journal of Computational Biology*, 2(2):333–353, 1995. doi:10.1089/CMB.1995.2.333.

23 Wim van Dam. Quantum oracle interrogation: Getting all information for almost half the price. In *Proceedings 39th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 362–367. IEEE, 1998. doi:10.1109/SFCS.1998.743486.

24 Yongzhen Xu, Shihao Zhang, and Lvzhou Li. Quantum algorithm for learning secret strings and its experimental demonstration. *Physica A: Statistical Mechanics and its Applications*, 609:128372, 2023.

A Quantum Query Complexity

Let $\mathcal{H}$ be a finite-dimensional Hilbert space. Any quantum algorithm $\mathcal{A}$ can be represented as a sequence of unitary operators $\{U_i\}_{i=0}^{T} \in \mathcal{L}(\mathcal{H})$ on an initial state $|\psi_0\rangle \in \mathcal{H}$. The algorithm is terminated by a measurement on $|\phi\rangle = U_T U_{T-1} \cdots U_1 U_0 |\psi_0\rangle$ to observe an outcome.

Let D and E be finite sets and $F : D \to E$ be a function. For every $x \in D$, let O_x be a unitary operator that depends on x, referred to as the *oracle*. A quantum query algorithm $\mathcal{A}$ alternates input-independent unitary operators with calls to the oracle, i.e., it applies the operations U_0, O_x, U_1, O_x, etc. The oracle calls are referred to as *queries* to the input x. The algorithm $\mathcal{A}$ is a bounded-error, quantum query algorithm for computing F if $\forall x \in D$ the output of $\mathcal{A}$ is $F(x)$ with probability at least $2/3$. The quantum query complexity of computing F is the minimum number of queries required to compute F with probability $2/3$. This is denoted by $\mathsf{Q}(F)$.

In this work, we consider algorithms with access to different *query sets* and therefore use $\mathsf{Q}^{\mathcal{R}}(F)$ to refer to the quantum query complexity of computing F using query set $\mathcal{R}$. Formally a query set $\mathcal{R}$ is a set consisting of functions $r : \{-1,1\}^n \to \mathcal{O}$ for some discrete range $\mathcal{O}$. The usual query model is captured by the query set $\mathcal{R} = \{r_i : \{-1,1\}^n \to \{-1,1\} \mid i \in [n]\}$ where $r_i(x) := x_i$. The final state of a quantum query algorithm with query set $\mathcal{R}$ looks like

$$|\phi\rangle = U_T O_x \dots U_2 O_x U_1 O_x U_0 |\psi_0\rangle$$

where the action of O_x on each basis state is

$$O_x : |r\rangle |b\rangle \to |r\rangle |b + r(x) \mod |\mathcal{O}|\rangle \qquad \forall r \in \mathcal{R}, b \in \{0, 1, \dots, |\mathcal{O}|\}.$$

Let $x \in \{-1,1\}^n$ be an input string and $\mathcal{Q} \subseteq 2^{[n]}$. We abuse notation and use $\mathcal{Q}$ to be the query set consisting of all queries $q = (S, b_S)$ of the form $q(x) = \mathbb{I}[x_S = b_S]$ where $S \in \mathcal{Q}$, $b_S \in \{-1,1\}^S$ and x_S is the substring of x restricted to indices in S. Throughout this work, we only work with queries of this form. Thus, for $\mathcal{Q} \subseteq 2^{[n]}$, we abuse notation and use $\mathcal{Q}$ to also denote the query set $\left\{q = (S, b) : S \in \mathcal{Q}, b \in \{-1,1\}^S\right\}$.

The primal adversary bound is a semi-definite program that characterizes the quantum query complexity of computing function $F : D \to E$, relative to a query set $\mathcal{R}$. To that end, $\forall q \in \mathcal{R}, \Delta_q \in \{0,1\}^{D \times D}$ is the matrix defined as:

$$\Delta_q[x,y] = \begin{cases} 1, & q(x) \neq q(y), \\ 0, & \text{otherwise.} \end{cases}$$

That is, $\Delta_q[x,y] = 1$ if and only if the query q can distinguish between inputs x and y. Throughout this work, we use the notation $\mathrm{ADV}^{\mathcal{R},\pm}$ to denote the adversary bound relative to query set $\mathcal{R}$. Now we state the primal version of the general adversary bound, which we will refer to as the *primal adversary bound*.

Primal adversary bound

$$\mathrm{ADV}^{\mathcal{R},\pm}(F) = \max_{\Gamma} \quad \frac{\|\Gamma\|}{\max\limits_{q \in \mathcal{R}} \|\Gamma \circ \Delta_q\|} \tag{5}$$

$$\begin{aligned} \text{subject to} \quad & \Gamma \in \mathbb{R}^{D \times D}, \\ & \Gamma \text{ is symmetric,} \\ & \Gamma[x,y] = 0. \qquad (\forall\, x, y \in D \text{ with } F(x) = F(y)) \end{aligned}$$

Since the objective function is invariant under multiplying Γ with a constant factor, we can without loss of generality assume that the denominator is equal to 1. Thus, we can equivalently write the SDP computing $\mathrm{ADV}^{\mathcal{R},\pm}$ with objective function $\max_\Gamma \|\Gamma\|$, and the additional constraint that $\forall q \in \mathcal{R}, \|\Gamma \circ \Delta_q\| \leq 1$.

Reichardt [20] showed that the primal adversary bound is a tight characterization of quantum query complexity.

▶ **Theorem 19** ([20, Theorem 1.4]). *Let D and E be finite sets and $F : D \to E$ be a function. The bounded-error, quantum query complexity of computing F using query set $\mathcal{R}$ is characterized by the adversary bound.*

$$\mathsf{Q}^{\mathcal{R}}(F) = \Theta(\mathrm{ADV}^{\mathcal{R},\pm}(F)).$$

Although [20, Theorem 1.4] states the adversary bound for index queries, it can be observed that the adversary bound characterizes the quantum query complexity for any query set and therefore we present it as such. Let $\mathcal{R}$ be a query set (assume for simplicity that all queries output values in $\{-1,1\}$. The argument below can easily be generalized if this is not the case). For every $x \in D$, define the string $\mathcal{R}(x) = (r(x))_{r\in\mathcal{R}}$. Define the partial function $G : \{-1,1\}^{\mathcal{R}} \to E$ on the input domain $\left\{z \in \{-1,1\}^{\mathcal{R}} : z = \mathcal{R}(x) \text{ for some } x \in D\right\}$ as $G(\mathcal{R}(x)) := F(x)$. Observe that querying a input variable, say $R \in \mathcal{R}$, to G on input $\mathcal{R}(x)$ is exactly the same as making the query R on input x. Thus, $\mathsf{Q}^{\mathcal{R}}(F) = \mathsf{Q}(G)$ and $\mathrm{ADV}^{\mathcal{R},\pm}(F) = \mathrm{ADV}^{\pm}(G)$, and therefore the result follows by applying [20, Theorem 1.4] to G.

When $\mathcal{Q} \subseteq 2^{[n]}$, and the query set consists of all queries of the form $\mathbb{I}[x_S = b]$ where $S \in \mathcal{Q}$ and $b \in \{-1,1\}^S$ are arbitrary, recall that we abuse notation and use $\mathcal{Q}$ to also denote the query set $\left\{q = (S,b) : S \in \mathcal{Q}, b \in \{-1,1\}^S\right\}$. Thus $\mathrm{ADV}^{\mathcal{Q},\pm}(\cdot)$ and $\mathsf{Q}^{\mathcal{Q}}(\cdot)$ denote the negative-weight adversary bound and the quantum query complexity, respectively, with respect to this query set.

B Fourier Analysis of Boolean Functions and Restrictions

Below are some relevant formal definitions and Fourier-analytic properties of restricted functions, adapted from [18].

Consider the vector space of functions from $\{-1,1\}^n$ to $\mathbb{R}$, equipped with the following inner product.

$$\langle f, g\rangle := \mathbb{E}_{x\in\{-1,1\}^n}[f(x)g(x)] = \frac{1}{2^n}\sum_{x\in\{-1,1\}^n} f(x)g(x).$$

For a set $S \subseteq [n]$, define the parity function $\chi_S : \{-1,1\}^n \to \mathbb{R}$ by $\chi_S(x) = \prod_{i\in S} x_i$. It is not hard to show that the set of all parities $\{\chi_S : S \subseteq [n]\}$ forms an orthonormal basis of the space above. Thus, every function $f : \{-1,1\}^n \to \mathbb{R}$ has a unique expansion with respect to this basis, $f = \sum_{S\subseteq[n]} \widehat{f}(S)\chi_S$. This expansion is called the Fourier expansion of f and the coefficients $\left\{\widehat{f}(S) : S \subseteq [n]\right\}$ are called the Fourier coefficients of f.

We list several properties of Fourier coefficients.

▶ **Lemma 20** ([5], also see [17, Lemma 2.2.3]). *Let $f : \{-1,1\}^n \to \mathbb{R}$ be a function and let M be the communication matrix of $f \circ \mathsf{XOR}$. Then,*

$$\|M\| = 2^n \max_{S\subseteq[n]} |\widehat{f}(S)|.$$

▶ **Definition 21.** *Let $f : \{-1,1\}^n \to \mathbb{R}$ be a function, and let $(J, \bar{J})$ be a partition of $[n]$. Let $z \in \{-1,1\}^{\bar{J}}$. Then we write $f_{J|z} : \{-1,1\}^J \to \mathbb{R}$ (pronounced "the restriction of f to J using z") for the subfunction of f obtained by fixing the coordinates in $\bar{J}$ to the bit values z, i.e.,*

$$\forall x \in \{-1,1\}^J, f_{J|z}(x) := f(x_J z_{\bar{J}})$$

where $x_J z_{\bar{J}}$ represents the string that equals x on J and z on $\bar{J}$.

▶ **Proposition 22** ([18, Proposition 3.21]). *Let $f : \{-1,1\}^n \to \mathbb{R}$ and let $(J, \bar{J})$ be a partition of $[n]$. For $z \in \{-1,1\}^{\bar{J}}$ and $S \subseteq J$ we have*

$$\widehat{f_{J|z}}(S) = \sum_{T \subseteq \bar{J}} \widehat{f}(S \cup T)\chi_T(z).$$

▶ **Corollary 23** ([18, Corollary 3.22]). *Let $(J, \bar{J})$ be a partition of $[n]$ and fix $S \subseteq J$. Suppose $z \sim \{-1,1\}^{\bar{J}}$ is chosen uniformly at random. Then,*

$$\mathbb{E}_z[\widehat{f_{J|z}}(S)^2] = \sum_{T \subseteq \bar{J}} \widehat{f}(S \cup T)^2.$$

The proof of the following lemma is available in the full version of the paper.

▶ **Lemma 24.** *For $f : \{-1,1\}^n \to \mathbb{R}$, the following conditions are equivalent:*

- *For all x with $|x|$ even, $f(x) = 0$.*
- *For all $S \subseteq [n]$, we have $\widehat{f}(S) = -\widehat{f}(\bar{S})$.*

We require the following well-known expression for the variance of a function. See, for example, [18, Proposition 1.13] for a simple proof.

▶ **Lemma 25.** *Let $f : \{-1,1\}^n \to \mathbb{R}$ be a function. Then,*

$$\mathbf{Var}(f) = \sum_{S \neq \emptyset} \widehat{f}(S)^2.$$