A Framework for Ruling out Quantum Speedups

Thomas Huffstutler
Department of Computer Science, Stony Brook University, NY, USA

Upendra Kapshikar
Department of Mathematics and Statistics, University of Ottawa, Canada

David Miloschewsky
Department of Computer Science, Stony Brook University, NY, USA

Supartha Podder
Department of Computer Science, Stony Brook University, NY, USA

Abstract

We study when partial Boolean functions can (and cannot) exhibit superpolynomial quantum query speedups, and develop a general framework for ruling out such speedups via two complementary lenses: *promise-aware* complexity measures and *function completions*.

First, we introduce promise versions of standard combinatorial measures (including block sensitivity and related variants) and prove that if the relevant promise and completion measures "collapse", then deterministic and quantum query complexities are necessarily polynomially related, i.e., $\mathrm{D}(f) = \mathrm{poly}(\mathrm{Q}(f))$. We then analyze structured families of promises, including symmetric partial functions and promises supported on Hamming slices, obtaining sharp (up to polynomial factors) characterizations in terms of a single gap parameter for the symmetric case and refined slice-dependent bounds for k-slice domains.

Next, we formalize *completion complexity* as the minimum of a measure over total completions of a partial function, and show that completability of a measure captures the possibility of superpolynomial quantum speedups. Finally, we apply this viewpoint to derive broad non-speedup criteria for some classes of functions admitting well-behaved completions, such as functions with low maximum influence on both the standard and p-biased hypercubes and functions with efficiently identifiable domains, and then show some hardness results for general completion techniques.

2012 ACM Subject Classification Theory of computation → Quantum query complexity

Keywords and phrases Complexity, Boolean Analysis, Quantum Speedup

Digital Object Identifier 10.4230/LIPIcs.MFCS.2026.10

Related Version *Full Version*: `https://arxiv.org/abs/2603.29256`

Funding *Upendra Kapshikar*: Supported by the Natural Sciences and Engineering Research Council of Canada (NSERC)(ALLRP-578455-2022, RGPIN-2022-05167) and the Air Force Office of Scientific Research under award number FA9550-20-1-0375.
Supartha Podder: This work was supported by US Department of Energy (grant no DE-SC0023179) and US National Science Foundation (award no 1954311).

Acknowledgements We thank Srijita Kundu for many helpful discussions during the early phase of this project. We also thank Nengkun Yu and Kunal Marwaha for helpful discussions.

1 Introduction

One of the central questions in quantum complexity theory is identifying problems which exhibit superpolynomial quantum speedups. While polynomial quantum speedups are relatively common and can be constructed for many problems, finding exponential quantum speedups is significantly more challenging. When it comes to showing exponential quantum speedups and/or designing quantum algorithms, the query complexity model is often the

preferred choice, as it is simple enough to facilitate proving lower bounds, yet expressive enough to capture the varying levels of problem difficulty. In this model, query access is given to an unknown n-bit input x with the goal of computing a predefined function f on x. The queries are given in a bit-wise manner, i.e., given an index i as a query, we get x_i as the response. The goal is to minimize the number of queries necessary to compute $f(x)$ in the worst case. In the randomized setting, algorithms are allowed to output incorrect answers with probability $\frac{1}{3}$. When we consider quantum algorithms in this model, we allow the algorithms to make queries in superposition, thus rendering the question of how much advantage quantum algorithms provide over classical algorithms in this setting.

Over the years, only a handful of structured problems – such as period-finding in Shor's algorithm or Simon's problem [33, 34] – have been shown to admit exponential quantum speedups. In contrast, unstructured problems generally allow only polynomial speedups. Despite being an important problem, only a few works have attempted to formalize quantum speedups and characterize what constitutes a superpolynomial quantum speedup. It was shown by [6] that when Boolean functions are defined on all inputs (i.e., $f : \{0,1\}^n \to \{0,1\}$), quantum and deterministic query complexities are polynomially (power 6) related. The relation was later improved to power 4 by [3], using Huang's sensitivity theorem [19]. [29] showed that deterministic and randomized complexities are polynomially related for total functions and in fact, for total functions, all well-studied query complexity measures are polynomially related[1] [10]. Thus, to achieve superpolynomial quantum speedups, we must consider partial functions (i.e., Boolean functions defined only on a subset of inputs, $f : X \subseteq \{0,1\}^n \to \{0,1\}$). In fact, we know[2] that the domain of any function achieving superpolynomial quantum speedup needs to be large [2].

However, not all partial functions (even on large domain size) exhibit quantum speedups. Thus, an important question is determining which partial functions exhibit superpolynomial quantum speedups, which do not, and more importantly, why. In pursuit of this, [1] showed that any partial symmetric function does not exhibit such speedup[3]. [12] later strengthened this result to a more general notion of symmetry and showed that certain types of permutation invariant functions admit no superpolynomial speedup. However, symmetric functions appear highly irregular as the function remains invariant under all possible permutations of the indices. [8] raised the question of how symmetric must a function be before it cannot exhibit a large quantum speedup and characterized functions invariant under different group actions, studying which ones allow superpolynomial speedups and which do not.

On the other hand, symmetric partial functions, even under various group actions, are only a tiny fraction of all possible partial functions. To shed light on the quantum speedups of non-symmetric functions, [7] showed that functions defined on the slice, i.e., $f : \binom{[n]}{k} \to \{0,1\}$, for $k \in [n]$, do not admit superpolynomial quantum speedups. This can be viewed as a generalization of [6]. Arguably in this case, the structure is on the domain, where every input x is *promised* to have Hamming weight k, but the function is free to take any possible value on these inputs.

In another direction, [2] studied the quantum advantage question from a different perspective. Given a problem which is intractable for both quantum and classical algorithms, can we find a sub-problem for which quantum algorithms provide an exponential advantage? Towards this end, they introduced the idea of *sculpting* functions, and asked if it is possible

[1] A table of polynomial relationships between well-studied measures for total functions can be found in [3]
[2] This is because $\mathrm{D}(f) = O\left(\mathrm{Q}(f)^2 \cdot \log^2 |\mathrm{Dom}(f)|\right)$ [2].
[3] They showed it for randomized vs. quantum.

to turn a total function into a partial function by removing some of the inputs, so that the newly formed partial function has a superpolynomial quantum speedup. They characterized the total Boolean functions for which an exponential quantum speedup can always be found in this manner and, as an implication, showed that when the domain size of a partial function is small, i.e., it contains only $\text{poly}(n)$ inputs, it cannot admit superpolynomial quantum speedups.

Other related works, such as [36, 9, 7, 5, 23, 31], explore quantum speedups from other perspectives, such as in the context of relational problems or under fixed query budgets etc. Before the work of [36], it was widely believed that achieving exponential quantum advantage required some form of global regularity or structure. However, their result suggests that quantum speedup is more subtle and less well-understood than previously thought. In this work, we revisit Boolean functions and provide alternative approaches to proving the absence of superpolynomial quantum speedups.

1.1 Our contributions

For the first result, we look at the promise version of block sensitivity. Consider a function $f : \{0,1\}^n \to \{0,1\}$. A *block* B is a subset of indices, $B \subset [n]$. For a total function, a block B is considered sensitive for input x if $f(x) \neq f(x^B)$, where x^B is obtained by flipping coordinates in B. For partial functions, we relax this notion: a block is *also* considered sensitive if flipping it causes the input to leave the promise (the function output is undefined) [13]. We call this measure *promised block sensitivity* and label it $\text{bs}'(f)$. A formal definition of $\text{bs}'(f)$ and other measures discussed may be found in Section 2.2. For two measures A and B, we use $A =_{\text{poly}} B$ to denote that they are polynomially related. Using the promise version of block sensitivity we show that large quantum query advantages cannot arise when other complexity measures collapse.

▶ **Result 1 (Requirements for superpolynomial quantum speedup**; Theorem 23**).** *For any partial Boolean function f, if either (i) critical block sensitivity* $\text{cbs}(f)$ *is polynomially related to* promised block sensitivity $\text{bs}'(f)$*, or (ii) critical block sensitivity is polynomially related to critical certificate complexity* $\text{C}_{\text{cri}}(f)$*, then quantum and deterministic query complexities are polynomially related. Formally,*

$$\begin{aligned} \text{cbs}(f) =_{\text{poly}} \text{bs}'(f) \ \textbf{\textit{or}}\ \text{cbs}(f) &=_{\text{poly}} \text{C}_{\text{cri}}(f) \\ \Downarrow & \\ \text{D}(f) \ =_{\text{poly}} \text{Q}(f). & \end{aligned}$$

Note that there are multiple ways one may define measures for partial functions which are equivalent for total functions [13, 4], we chose our definitions as we found them the most natural for studying the role of $\text{Dom}(f)$ when comparing $\text{D}(f)$ and $\text{Q}(f)$.

Many natural Boolean functions exhibit a high degree of permutation symmetry: their value depends only on the Hamming weight of the input, or they could be defined only on a single orbit under the action of the symmetric group S_n. Formally, we study two related families of partial functions with S_n-symmetry:

- *fully symmetric* partial functions whose value depends only on $\text{wt}(x)$, and
- functions whose domain is a single k-slice of the Boolean cube (equivalently, an S_n-orbit), where the function may be arbitrary on that slice.

Our first set of results gives a clean, tight characterization (up to polynomial factors) of several query complexity measures for partial symmetric functions in terms of a single parameter we denote GAP,

$$\mathrm{GAP}(f) \;=\; \min_{x,y\in \mathrm{Dom}(f),\, f(x)\neq f(y)} |\mathrm{wt}(x)-\mathrm{wt}(y)|.$$

Informally, $\mathrm{GAP}(f)$ captures the minimal Hamming-weight separation between any two inputs on which the function differs.

Secondly, we look at the family of functions defined on a single slice. These functions were previously studied by [7], who proved a general bound $\mathrm{D}(f) = O(\mathrm{Q}(f)^{18})$.

▶ **Result 2 (Partial functions and S_n; Lemma 27, 28, 31).** *For any partial symmetric function f,*

$$\widetilde{\deg}(f) =_{\mathrm{poly}} \mathrm{Q}(f) =_{\mathrm{poly}} \mathrm{R}(f) =_{\mathrm{poly}} \frac{n}{\mathrm{GAP}} \qquad \textit{and}$$
$$\mathrm{D}(f) =_{\mathrm{poly}} n - \mathrm{GAP}.$$

Combining these results, we get a complete picture of when quantum speedups over deterministic algorithms are possible for symmetric functions. As a corollary, we also obtain that, for partial symmetric functions, the polynomial method is essentially tight: $\widetilde{\deg}$ and Q are polynomially related.

Next, for any function on the k-slice,

$$\mathrm{D}(f) \;\leq\; \frac{3n}{\min\{k,n-k\}}\,\mathrm{C}(f)\,\mathrm{bs}(f),$$

which, together with known relationships between $\mathrm{C}(f)$, $\mathrm{bs}(f)$ and $\mathrm{Q}(f)$, implies $\mathrm{D}(f) = O\big(\frac{n}{\min\{k,n-k\}}\,\mathrm{Q}(f)^6\big)$.

This bound is not comparable to the result of [7]. It is stronger for slices near the middle ($k \approx n/2$) and weaker for very small or very large slices.

We then continue the line of work of sculpting [2], but in the reverse direction, i.e., instead of removing inputs from a total function, we will start with a partial function and add inputs to its domain to obtain a total one. This method is known as a *completion* and has been studied in the communication complexity model [18, 17]. However, as far as we know, in the query-complexity model, a systematic study of *completions* has been limited in the literature.

We ask how much different complexity measures can increase under this transformation. More specifically, given a partial Boolean function f, a total Boolean function F is a *completion* of f if $F(x) = f(x)$ for all $x \in \mathrm{Dom}(f)$, while values on inputs outside the promise are assigned arbitrarily. Naturally, a partial function may admit many distinct completions. The *completion complexity* of f with respect to any query complexity measure M is then defined as $\overline{M}(f) := \min M(F)$, where the minimum is over all possible completions of f.

As our third result, we show that completions exactly characterize superpolynomial quantum speedups.

▶ **Result 3 (Completions characterize superpolynomial quantum speedups).** *For any* partial *function f, and for any*[4] *query complexity measure M, deterministic query complexity* $\mathrm{D}(f)$ *is polynomially related to $M(f)$ if and only if the measure M is* extendable*, that is,*

$$\mathrm{D}(f) =_{\mathrm{poly}} M(f) \textit{ if and only if } \overline{M}(f) =_{\mathrm{poly}} M(f).$$

[4] Any measure M which for total functions is polynomially related to D [6].

Specifically, a partial function extends to a total function while preserving a polynomial relationship in any complexity measure, if and only if no superpolynomial speedup over classical query complexity is possible for that measure.

We can use this result to establish quantum non-speedup in the following way: For any (possible partial) function f, $\mathrm{D}(f) \geq \mathrm{Q}(f) \geq \widetilde{\deg}(f)$. If a partial function f has approximate degree T, then by definition for every input $x \in \mathrm{Dom}(f)$, the approximating polynomial p outputs a value $p(x)$ with $|p(x) - f(x)| \leq \epsilon$. However, for the inputs $x \notin \mathrm{Dom}(f)$, the polynomial can obtain any value in $[0, 1]$. Thus, if we can construct a *completed* polynomial P of degree $\mathrm{poly}(T)$ such that $|P(x) - f(x)| \leq \epsilon'$ for every $x \in \mathrm{Dom}(f)$ and $P(x)$ is ϵ' close to either 0 or 1 for all $x \notin \mathrm{Dom}(f)$, then $\mathrm{D}(f) = O(\mathrm{poly}(\widetilde{\deg}(f)))$. See Figure 1 for an illustration of this concept.

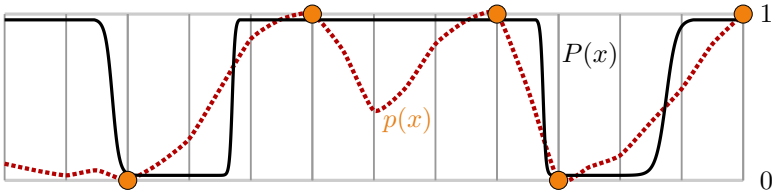

Figure 1 The orange points represent the original inputs in the domain, and the red dotted polynomial $p(x)$ is a polynomial that approximates the function f on these points. The solid black polynomial $P(x)$ is a completion that provides an approximation across the entire Boolean hypercube.

Using the completion framework, we show that superpolynomial quantum speedups exist only when the complexity of identifying the domain is superpolynomial with respect to the approximate degree.

▶ **Result 4 (No superpolynomial speedup for functions with efficiently verifiable domains).** *Let $X \subseteq \{0,1\}^n$ be a domain whose membership can be checked by an algorithm in T queries. Then, for* all *functions $f : X \to \{0,1\}$ with approximate degree $\Omega(\mathrm{poly}(T))$,*

$$\mathrm{D}(f) =_{\mathrm{poly}} \widetilde{\deg}(f).$$

Next, we demonstrate the usefulness of completions by proving quantum non-speedup results for various function families. We consider various smoothness properties of polynomials, the main property being Lipschitz continuity which has been studied in non-smooth and nonconvex optimization, stochastic convex optimization [24], and geometric analysis and optimal transport [26]. Furthermore, we consider functions f with low influence which as well as functions which are defined on the generalized p-biased hypercube. Functions on the p-biased hypercube have been studied for their relationships to the Erdős–Rényi random graph model in [35, 28], as well as with regards to the effects of the p-biased hypercube on noise stability [25] and linearity testing [22]. These p-biased functions have also been studied with respect to their relationships to juntas and low degree functions [15, 16], have been used to establish a hypercontractive inequality for general p and prove an analog of the invariance principle of Mossel, O'Donnell and Oleszkiewicz for the p-biased hypercube [27, 21].

▶ **Result 5 (No superpolynomial speedup for functions with low influence and sparsity).** *Let $f : \mathrm{Dom}(f) \to \{-1, 1\}$ have an approximating polynomial $p(x)$ such that for all $i \in [n]$, the influence $\mathrm{Inf}_i(p)$ and sparsity $\mathrm{spar}_i(p)$ are low. Then, $\mathrm{D}(f) =_{\mathrm{poly}} \mathrm{Q}(f)$.*

Furthermore, we generalize the results to the p-biased hypercube.

Finally, we propose a general completion technique concerning finding a perturbation vector Δ for an approximating polynomial $p(x)$ for a partial function f, so that adding this perturbation to the Fourier coefficients of $p(x)$ results in a polynomial $p_\Delta(x)$ such that $|p_\Delta(x)| \geq \frac{1}{\widetilde{\text{poly}}(\deg(f))}$ for all $x \in \{-1,1\}^n$, resulting in an approximating polynomial for a total function. Furthermore, we show via a reduction from 3-SAT that the problem of finding such a perturbation in general is NP-complete.

▶ **Result 6.** *Given multilinear polynomials $p, q_1, \ldots, q_n : \{-1,1\}^n \to \mathbb{R}$, and constants $\epsilon, B > 0$, deciding whether there exists a vector $\Delta \in \mathbb{R}^n$ such that for all $x \in \{-1,1\}^n$,*

$$\left| p(x) + \sum_{j=1}^{n} \Delta_j q_j(x) \right| \geq \epsilon \text{ and } ||\Delta||_\infty \leq B$$

is NP*-complete.*

1.2 Our techniques

Promise measures

To obtain the requirements for speedup using the two variations of promise measures, we follow the steps of the proof that for total functions f, $\mathrm{D}(f) =_{\text{poly}} \mathrm{Q}(f)$. For a survey of the proofs, see [10]. First, we define the alternative promise versions of $\mathrm{bs}(f)$ and $\mathrm{C}(f)$, $\mathrm{bs}'(f)$ and $\mathrm{C}'(f)$. As $\mathrm{bs}'(f)$ was already described alongside Result 1, let us describe $\mathrm{C}'(f)$. Standardly, a certificate c for a promise function certifies that each input in the promise which agrees with c has the same output. The alternative $\mathrm{C}'(f)$ is defined such that no input which agrees with c is outside of $\mathrm{Dom}(f)$.

Transferring the relationships from the total to our promise versions of the measures relies on the role of inputs outside of the promise. Specifically, many steps use the assumption that changing the input, for example by flipping multiple indices, keeps it in the promise, which is not necessarily true. For example, consider the proof that the size of a minimal block is bounded by sensitivity. If we attempt to show a promise-version of such a statement, meaning that the size of a minimal block B such that $x, x^B \in \mathrm{Dom}(f)$ is bounded by its sensitivity, we find that removing an index i from B might take it out of the promise, meaning $x^{B\setminus\{i\}} \notin \mathrm{Dom}(f)$ and therefore i might not be a sensitive index. This is underscored by the fact that there are partial functions for which $\mathrm{s}(f) = 0$, such as functions on the slice. On the other hand, using the same argument, we bound the size of a block which certifies that takes an output outside of the promise or changes its output by $\mathrm{s}'(f)$ ($\mathrm{bs}'(f)$ restricted to blocks of size 1). The other proofs follow similarly, meaning that one version of the measure makes the proof impossible, while the other does not.

Efficiently identifying domains is sufficient

When trying to think of explicit constructions for a completion, the naïve idea is to just set all inputs not in the promise of a partial function to have an output of 0. To characterize the complexity in using this naïve completion, we examine the difficulty of determining, for any partial function $f : X \to \{0,1\}$, whether a given input $x \in X$ or if $x \notin X$. This is equivalent to the condition that there exists a low degree approximation of the indicator function $\mathbb{1}_X$ on X, denoted by $\widetilde{\mathbb{1}}_X$. We take the degree d $\epsilon-$approximating polynomial $p(x)$ for $f(x)$, and the low degree ($\leq \text{poly}(d)$) $\epsilon'-$approximator $q(x) = \widetilde{\mathbb{1}}_X$ and define the new polynomial $r(x) = q(x)p(x)$. This new polynomial approximates the naïve completion F of f everywhere

on the Boolean hypercube to error at most $\epsilon + \epsilon'$ by the triangle inequality. Since $\widetilde{\mathbb{1}}_X$ is defined for a total function, this implies that being able to efficiently (classically) distinguish between X and $\{0,1\}^n/X$ is sufficient to show that the naïve completion is enough to prove that no superpolynomial speedup exists for the given partial function f.

Characterizing the natural completion

A natural completion one can think of is based on the sign of an approximating polynomial. If we are given a partial function $f : \mathrm{Dom}(f) \to \{-1, 1\}$ which has a degree d approximating polynomial $p(x)$, then $p(x)$ will still output a value on inputs $x \notin \mathrm{Dom}(f)$. If we take any $x \notin \mathrm{Dom}(f)$ and evaluate $p(x)$, then as long as $p(x)$ outputs a value that is sufficiently bounded away from 0, by standard boosting techniques we can use $p(x)$ as the degree d approximating polynomial for the completion $F(x) = \mathrm{sign}(p(x))$ (i.e. $F(x) = 1$ if $p(x) > 0$ and $F(x) = -1$ if $p(x) < 0$). The problem is that we are only guaranteed this holds for $x \in \mathrm{Dom}(f)$. For $x \notin \mathrm{Dom}(f)$, we could have $|p(x)| \leq \epsilon$ where $\epsilon \ll \mathrm{poly}(d)^{-1}$.

We would like to characterize the notion of this *natural* completion more precisely and give a condition which captures when we can guarantee that $|p(x)| \geq \frac{1}{\mathrm{poly}(n)}$ on inputs outside of the domain. To this end, we consider global smoothness properties of polynomials, namely the Lipschitz continuity of an approximating polynomial for f. We give the optimal Lipschitz continuity condition for an approximating polynomial $p(x)$ in order to ensure that $|p(x)| \geq \frac{1}{\mathrm{poly}(\widetilde{\deg}(f))}$ so that we can boost $p(x)$ to the canonical $\frac{1}{3}$ error approximating polynomial for f while only adding a polynomial blow-up in the degree of $p(x)$. We then give conditions on the influence and sparsity of partial functions f that ensure this global smoothness condition for an approximating polynomial for f, and generalize this condition to partial functions defined on the p-biased hypercube. Lastly, we discuss other ways one may obtain such completions by perturbing the approximating polynomial.

1.3 Discussion and further work

In this work we introduced two new frameworks for proving non-superpolynomial quantum speedups and analyzed the relationship between partial functions and S_n. Furthermore, we showed several applications of the completion technique. We also introduced a new technique using the smoothness of an approximating polynomial to prove that no superpolynomial speedup exists. We list some open problems that would further the usefulness of our techniques.

- Completion complexity characterizes the possibility of superpolynomial speedups in comparison to deterministic query complexity. Is there a stronger characterization when comparing quantum to randomized query complexity?
- Since the completion of an approximate degree captures the gap between deterministic and approximate degree – and consequently, between deterministic and quantum complexity – but not all functions exhibit quantum speedups, it follows that not all polynomials are extendable. What intrinsic properties of a polynomial determine whether it admits such a completion?
- For what classes of functions is a low degree approximation of the domain indicator function achievable? An exact characterization would quantify precisely when the naïve completion is sufficient to prove no superpolynomial speedup.

- Theorem 23 directly implies that partial functions on domains with a large promise (i.e., low number of undefined inputs) do not attain superpolynomial speedup. Can one define a measure of *connectedness* on the Boolean hypercube and show that on highly-connected domains one cannot obtain superpolynomial speedup?
- Are there more fine-grained versions of the promise measures discussed in Section 3 which allow us to show the non-existence of a superpolynomial speedup?

2 Preliminaries

For a positive integer n, let $[n] = \{1, 2, \ldots, n\}$. For $x \in \{0,1\}^n$ and $S \subseteq [n]$, we write x_S for the restriction of x to the coordinates in S. For $i \in [n]$, let x^i denote the string obtained from x by flipping its ith bit; more generally, for a set[5] $\mathrm{B} \subseteq [n]$, let x^{B} denote the string obtained by flipping all bits in B.

Let $\mathrm{Dom} \subseteq \{0,1\}^n$ and consider a function $f : \mathrm{Dom} \to \{0,1\}$. Alternatively, we can express the same function $f : \{0,1\}^n \to \{0,1,*\}$ with $f(x) = *$ if and only if $x \notin \mathrm{Dom}$. We will switch between these two representations depending on the context. We also view a function f either as $f : \mathrm{Dom}(f) \subseteq \{0,1\}^n \to \{0,1\}$ or as $f : \mathrm{Dom}(f) \subseteq \{\pm 1\}^n \to \{-1,1\}$. These two viewpoints are equivalent. The complement of a domain is denoted $\mathrm{Dom}(f)^c = \{x \in \{0,1\}^n : x \notin \mathrm{Dom}(f)\}$.

Given a string $x \in \{0,1\}^n$, we use $\mathrm{wt}(x)$ to denote its Hamming weight.

2.1 Query complexity

We use the standard query complexity model and measures, such as $\mathrm{D}(f), \mathrm{R}(f), \mathrm{Q}(f), \widetilde{\deg}(f)$. For a survey of these measures, see [10].

2.2 Definition of measures for partial functions

Due to the nature of partial functions, which are undefined on certain inputs, one must be careful when defining complexity measures. Indeed, when it comes to partial functions, even for standard combinatorial functions, there is no *canonical choice* of complexity measure, and there are often multiple definitions in the literature [13, 4]. Ideally, we would want these measures to be *meaningful* and to be related to some algorithmic measures, just like in the case of total functions. For example, some combinatorial measures, such as approximate degree, adversary bound etc., of total functions lower bound quantum query complexity $\mathrm{Q}(f)$. Therefore, a useful definition of these measures in the case of partial functions should also lower bound $\mathrm{Q}(f)$. Here, we state the definitions we will use throughout the paper, and show that they satisfy the requisite properties. Note that various measures for partial functions were already considered in the literature; see [4, 13, 11] for reference.

▶ **Definition 7** (Approximate degree). *For a (possibly partial) Boolean function f, its approximate degree measure, denoted by $\widetilde{\deg}_\epsilon(f)$, is the minimum degree of a real polynomial p that approximates f in the sense that*

$$|p(x) - f(x)| \leq \epsilon \quad \text{for all } x \in \mathrm{Dom}(f).$$

Additionally, $p(x)$ must satisfy

$$p(x) \in [0,1] \quad \text{for all } x \in \{0,1\}^n.$$

[5] We will use terms set and block interchangeably, as is common in query complexity literature.

As it is common in the literature, we will use the shorthand $\widetilde{\deg}$ to denote $\widetilde{\deg}_{\frac{1}{3}}$. Note that for $\widetilde{\deg}(f)$, we assert that the polynomial $p(x)$ is bounded for all $x \in \{0,1\}^n$ (as opposed to only on $\mathrm{Dom}(f)$), which provides a stronger lower bound of $\mathrm{Q}(f)$.

In a seminal result, [6] showed that one can lower bound quantum query complexity by the approximate degree, which leads to the following chain of inequalities.

▶ **Lemma 8** ([6]). *For every Boolean function f, we have,* $\mathrm{D}(f) \geq \mathrm{R}(f) \geq \mathrm{Q}(f) = \Omega\left(\widetilde{\deg}(f)\right)$.

The above results were initially proved for total functions, but as mentioned in [4], one can define these measures for partial functions, and the results also hold for them.

▶ **Definition 9** (Partial assignment). *A partial assignment α is an element of $\{0,1,*\}^n$. We say that α is consistent with an input $x \in \{0,1\}^n$ if for all $i \in [n]$, either $\alpha_i = x_i$ or $\alpha_i = *$.* Support *of a partial assignment α, denoted as* $\mathrm{Support}(\alpha)$*, is the set $\{i : \alpha_i \neq *\}$.*

▶ **Definition 10** (Certificate, [13]). *Let $f : \{0,1\}^n \to \{0,1,*\}$, and $f(x) = b \in \{0,1\}$. Let α be a partial assignment that is consistent with x. We say that α is*

- *A b-certificate (for x) if for any x' consistent with α, $f(x') = b$.*
- *A $\{b,*\}$-certificate (for x) if for any x' consistent with α, $f(x') \in \{b,*\}$.*

Moreover, we define

- $\mathrm{C}^b(f,x)$ *as the size of the smallest b-certificate of x and*
- $\mathrm{C}^{\{b,*\}}(f,x)$ *as the size of the smallest $\{b,*\}$-certificate of x.*

Essentially, a b-certificate certifies that we are always within the promise and output b, while a $\{b,*\}$-certificate certifies that *if* we are in the promise, we output b. We define the following measures.

▶ **Definition 11** (Certificate measure, [13]). *Given $f : \{0,1\}^n \to \{0,1,*\}$ and $b \in \{0,1\}$, we have that:*

$$\begin{aligned}
\mathrm{C}^b(f) &= \max_{x \in f^{-1}(b)} \mathrm{C}^b(f,x) \\
\mathrm{C}'(f) &= \max\{\mathrm{C}^0(f), \mathrm{C}^1(f)\} \\
\mathrm{C}^{\{b,*\}}(f) &= \max_{x \in f^{-1}(b)} \mathrm{C}^{\{b,*\}}(f,x) \\
\mathrm{C}(f) &= \max\{\mathrm{C}^{\{0,*\}}(f), \mathrm{C}^{\{1,*\}}(f)\}.
\end{aligned}$$

While there are multiple ways to define the certificate measure for partial function, the definition of $\mathrm{C}(f)$ above appears the most natural as it lower-bounds deterministic query complexity.[6] By definition, we have that $\mathrm{C}(f) \leq \mathrm{C}'(f)$ for all f and for any total function f, we have, $\mathrm{C}(f) = \mathrm{C}'(f)$. One should note that, $\mathrm{C}(f)$ could be significantly smaller than $\mathrm{C}'(f)$. For example, the function f defined on the 1-slice, which asks whether the location $i \in [n]$ such that $x_i = 1$ is less than $\frac{n}{2}$, maximally separates these measures with $\mathrm{C}(f) = O(1)$ and $\mathrm{C}'(f) = \Omega(n)$ [13].

[6] Although the notation $\mathrm{C}(f)$ has been canonically used in the literature for total Boolean functions to represent what we denote here as C', we follow the notational convention established in [13] for the certificate complexity of partial functions.

▶ **Definition 12** (Block notation). *Let $x \in \{0,1\}^n$ and $B \subseteq [n]$. We let x^B denote the string obtained from x by flipping the bits in B. Equivalently,*

$$(x^B)_i = \begin{cases} x_i, & i \notin B, \\ 1 - x_i, & i \in B. \end{cases}$$

We refer to such a set B as a block. *If B is a block of size one (say $B = \{j\}$), then we simply write x^j (instead of $x^{\{j\}}$).*

Following the ideas of [13], we define two promise versions of sensitivity and block sensitivity with the main difference being how we process undefined inputs.

▶ **Definition 13** (Sensitivity and block sensitivity). *Given $f : \{0,1\}^n \to \{0,1,*\}$ and $x \in \mathrm{Dom}(f)$, let:*

- $\mathrm{s}(f,x)$ *be number of variables $i \in [n]$ such that $f(x^i) = 1 - f(x)$.*
- $\mathrm{s}'(f,x)$ *be number of variables $i \in [n]$ such that $f(x^i) \in \{1 - f(x), *\}$.*
- $\mathrm{bs}(f,x)$ *be the number of disjoint blocks $B \subseteq [n]$ such that $f(x^B) = 1 - f(x)$.*
- $\mathrm{bs}'(f,x)$ *be the number of disjoint blocks $B \subseteq [n]$ such that $f(x^B) \in \{1 - f(x), *\}$.*

Then we define,

$$\mathrm{s}(f) := \max_{x \in \mathrm{Dom}(f)} \mathrm{s}(f,x)$$
$$\mathrm{s}'(f) := \max_{x \in \mathrm{Dom}(f)} \mathrm{s}'(f,x)$$
$$\mathrm{bs}(f) := \max_{x \in \mathrm{Dom}(f)} \mathrm{bs}(f,x)$$
$$\mathrm{bs}'(f) := \max_{x \in \mathrm{Dom}(f)} \mathrm{bs}'(f,x).$$

Note that the maximum is taken only over points in the domain, that is, over all those x such that $f(x) \neq *$.

By definition, for any total function f, $\mathrm{s}'(f) = \mathrm{s}(f)$ and $\mathrm{bs}'(f) = \mathrm{bs}(f)$. However, these equalities need not hold for partial functions. The dictator function on the 1-slice maximally separates $\mathrm{s}(f)$ and $\mathrm{bs}(f)$ from $\mathrm{s}'(f)$ and $\mathrm{bs}'(f)$, respectively.

▶ **Observation 14.** *By Definitions 11 and 13, for all f, b, x, we get the following relations:*

$$\mathrm{s}'(f,x) \leq \mathrm{bs}'(f,x) \leq \mathrm{C}^{f(x)}(f,x)$$
$$\mathrm{s}(f,x) \leq \mathrm{bs}(f,x) \leq \mathrm{C}^{\{f(x),*\}}(f,x)$$
$$\mathrm{s}(f,x) \leq \mathrm{s}'(f,x)$$
$$\mathrm{bs}(f,x) \leq \mathrm{bs}'(f,x).$$

▶ **Definition 15** ($\mathcal{A}$-sensitive blocks). *Let $\mathcal{A} \subsetneq \{0,1,*\}$. We say that a block B is $\mathcal{A}$-sensitive (for an input x) if $f(x) \notin \mathcal{A}$ and $f(x^B) \in \mathcal{A}$. In case that $\mathcal{A} = \{a\}$, we simply say a-sensitive.*

Recall that we say a function f' is a completion of f if f' is total and $f(x) = f'(x)$ for all $x \in \mathrm{Dom}(f)$.

▶ **Definition 16** (Critical Block Sensitivity [20]). *Let $f : \{0,1\}^n \to \{0,1,*\}$. The critical block sensitivity of f is*

$$\mathrm{cbs}(f) = \min_{f'} \max_{x \in \mathrm{Dom}(f)} \mathrm{bs}(f', x)$$

where f' is a completion of f.

Furthermore, using critical block sensitivity $\mathrm{cbs}(f)$, we define critical certificate,

▶ **Definition 17** (Critical Certificate [4]). *Let $f : \{0,1\}^n \to \{0,1,*\}$. Let $\mathcal{F} = \{f'$ such that f' is a completion of f and $\mathrm{bs}(f') = \mathrm{cbs}(f)\}$. The critical certificate complexity of f is,*

$$\mathrm{C}_{\mathrm{cri}}(f) = \min_{f' \in \mathcal{F}} \max_{x \in \mathrm{Dom}(f)} \mathrm{C}(f', x).$$

By definition, for any total function f, we have, $\mathrm{cbs}(f) = \mathrm{bs}(f)$ and $\mathrm{C}_{\mathrm{cri}}(f) = \mathrm{C}(f)$. Lastly, the following relationship between $\mathrm{bs}(f), \mathrm{cbs}(f)$ and $\widetilde{\deg}(f)$ holds for any Boolean function f.

▶ **Lemma 18** ([4]). *For every partial Boolean function f,*

$$\mathrm{bs}(f) \leq \mathrm{cbs}(f) = O(\widetilde{\deg}(f)^2).$$

It should be noted that the proof that $\mathrm{bs}(f) = O(\widetilde{\deg}(f)^2)$ follows by constructing a polynomial which turns the blocks into variables and via symmetrization argues that the approximate degree must be large. In particular, this means that we can not replace $\mathrm{bs}(f)$ with $\mathrm{bs}'(f)$ in the above lemma.

3 Promise measures

A central question in quantum query complexity is whether superpolynomial quantum speedups require a nontrivial promise. This has been formalized in several settings: symmetric and permutation-symmetric partial functions admit at most polynomial quantum advantages, and similar no-speedup results are known for functions on structured domains such as slices [1, 12, 7]. This section is motivated by the goal of identifying a more local obstruction that does not rely on global symmetry. The key point is that for partial functions, local perturbations may leave the domain rather than reach an opposite-labeled input. By analyzing the promise-aware versions of the usual combinatorial measures we defined in Section 2.2 and studying when they still satisfy total-function-type relations, this section aims to identify conditions for superpolynomial quantum speedups.

▶ **Lemma 19.** *For any partial function $f : \mathrm{Dom}(f) \to \{0,1\}$, we have,*

$$\mathrm{C}(f) \leq \mathrm{bs}(f)\mathrm{s}'(f).$$

Note that in the above lemma, we showed that $|B_i| \leq s'(f)$. For a total function f, it also holds that $|B_i| \leq \mathrm{s}(f)$. This does not transfer to partial functions. Moreover, $\{1 - f(x), *\}$-sensitive blocks B^* can neither be bounded by s or by s'. For example, consider the constant 0 function whose domain is $\{0,1\}^n \setminus \{0^n\}$.

▶ **Lemma 20.** *For any partial function f the following inequalities hold:*

1. $\mathrm{D}(f) \leq \min\{\mathrm{C}^{\{1,*\}}(f), \mathrm{C}^{\{0,*\}}(f)\} \cdot \mathrm{bs}'(f)$
2. $\mathrm{D}(f) \leq \min\{\mathrm{C}^1(f), \mathrm{C}^0(f)\} \cdot \mathrm{bs}(f)$
3. $\mathrm{D}(f) \leq \mathrm{C}_{\mathrm{cri}}(f) \cdot \mathrm{cbs}(f)$.

Alternatively, we may bound $\mathrm{D}(f)$ using $\mathrm{C}(f)$ and $\mathrm{C}'(f)$.

▶ **Lemma 21.** *For any partial function $f : \mathrm{Dom}(f) \to \{0,1\}$,*

$$\mathrm{D}(f) \leq \min\{\mathrm{C}^1(f)\mathrm{C}^{\{0,*\}}(f), \mathrm{C}^0(f)\mathrm{C}^{\{1,*\}}(f)\}.$$

Next, we find that $\mathrm{cbs}(f)$ sits between $\mathrm{bs}(f)$ and $\mathrm{bs}'(f)$.

▶ **Lemma 22.** $\mathrm{bs}(f) \leq \mathrm{cbs}(f) \leq \mathrm{bs}'(f)$.

Lastly, we prove a statement which shows how a polynomial relationship between various measures discussed in this section implies the impossibility of a quantum speedup.

▶ **Theorem 23.** *Let f be a partial function such that* at least *one of the following conditions is satisfied:*

1. $\mathrm{cbs}(f) =_{\mathrm{poly}} \mathrm{bs}'(f)$
2. $\mathrm{cbs}(f) =_{\mathrm{poly}} \mathrm{C}_{\mathrm{cri}}(f)$
3. $\min\{\mathrm{C}^1(f), \mathrm{C}^0(f)\} \leq_{\mathrm{poly}} \mathrm{cbs}(f)$.
4. $\mathrm{C}(f) =_{\mathrm{poly}} \mathrm{C}'(f)$ *and* $\mathrm{bs}(f) =_{\mathrm{poly}} \mathrm{s}'(f)$.

Then, $\mathrm{D}(f) =_{\mathrm{poly}} \mathrm{Q}(f)$.

Theorem 23 implies that several structured domains cannot attain a speedup. For example, Item 1 implies that $\mathrm{bs}(f) =_{\mathrm{poly}} \mathrm{bs}'(f)$ leads to no superpolynomial quantum speedups. It also implies that for a domain to attain speedup, there must exist at least one input x with many disjoint blocks B such that $x^B \notin \mathrm{Dom}(f)$. Alternatively, if the graph induced by adjacent inputs in the domain is "well-connected" such that the degree of each node is closer to n than the block sensitivity of the function, it cannot attain speedup.

Item 2 of Theorem 23 is consistent with Result 3: if a partial function admits a completion that does not increase a relevant parameter, then the corresponding deterministic and quantum query complexities are polynomially related.

As a simple example, fix some input $x^* \in \{0,1\}^n$ and consider a partial Boolean function f with domain $\mathrm{Dom}(f) = \{x : \forall i \in [n],\ x_i \leq x^*_i\}$. On this domain, define a completion g by

$$g(x) = f(x \wedge x^*), \text{ where } (x \wedge x^*)_i = x_i \wedge x^*_i.$$

This completion preserves the relevant measures: every minimal sensitive block and every certificate depends only on coordinates i such that $x^*_i = 1$. In particular,

$$\max_{x \in \mathrm{Dom}(f)} \mathrm{bs}(g, x) = \mathrm{cbs}(f) = \mathrm{bs}(f), \qquad \text{and} \qquad \mathrm{C}_{\mathrm{cri}}(f) = \mathrm{C}(f).$$

This argument extends to domains of the form $\bigcup_i d(x^*_i)$, provided that

$$\max_{i,j} \mathrm{wt}(x^*_i - x^*_j) = O(1),$$

as in this case, the coordinates on which the various x^*_i differ can be queried to identify the subdomain containing the input.

Interestingly, the conclusion of Theorem 23 does not follow from $\mathrm{C}(f) =_{\mathrm{poly}} \mathrm{C}'(f)$, meaning the second condition in Item 4 is necessary. This is due to the Deutsch-Jozsa algorithm which solves the problem of whether some input $x \in \{0,1\}^n$ is constant (i.e., $|x| = 0$ or n) or perfectly balanced ($|x| = n/2$) [14]. In this case, all three of $\mathrm{D}(f), \mathrm{C}(f)$ and $\mathrm{C}'(f)$ are $O(n)$, while $\mathrm{Q}(f) = 1$.

4 Functions with S_n symmetry

In this section, we look at functions that have some invariance under the symmetric group S_n. In the first part, we consider functions having the highest order of S_n symmetry (Definition 24). In the second part, we will consider functions defined on a single orbit of S_n, also known as the Boolean slice. However, in this case, the function may be arbitrarily defined on the slice.

4.1 Symmetric functions

In this part, we will study functions that are symmetric as defined below.

▶ **Definition 24** (Symmetric function). *Let* $\mathrm{Dom}(f) \subseteq \{0,1\}^n$. *We say that a function* $f : \mathrm{Dom}(f) \to \{0,1\}$ *is symmetric under* S_n *(or simply symmetric) if for all* $\sigma \in S_n$ *and for all* $x \in \{0,1\}^n$*:*

1. $x \in \mathrm{Dom}(f)$ *if and only if* $\sigma(x) \in \mathrm{Dom}(f)$.
2. $f(x) = f(\sigma(x))$.

Note that for a symmetric function f, its evaluation at a point x depends only on $\mathrm{wt}(x)$.

Our main focus will be on characterizing various measures such as D and Q for partial symmetric functions. This problem has already been considered for R and Q in [12], proving that there is no superpolynomial speedup between the two quantities. Furthermore, [31] analyzed these quantities and proved that Q can be written as a certain maximization problem, whereas our expression is free from such optimization and has a much simpler proof technique. Specifically, we give a characterization of various measures (up-to a polynomial factor) using GAP, a parameter for symmetric functions we define below. Additionally, we also consider the relationship between $\widetilde{\deg}$ and D, of which the former has been characterized for total Boolean functions by [30], and the latter is trivial; for total symmetric functions as it is $\Theta(n)$ for non-constant functions [10]. To the best of our knowledge, no earlier works attempt to analyze these measures for partial symmetric functions. Furthermore, our results exactly characterize when symmetric functions attain a superpolynomial separation between Q and D (or $\widetilde{\deg}$ and D). As an added result, our proof also shows that $\widetilde{\deg}$ and Q are polynomially related for every such function, hence the polynomial method is tight for partial symmetric functions. Let us formally define GAP.

▶ **Definition 25** (Gap of a symmetric function). *For any partial symmetric function* $f : \mathrm{Dom}(f) \to \{0,1\}$*, define* $\mathrm{GAP}(f)$ *as*

$$\mathrm{GAP}(f) = \min_{x,y:f(x)\neq f(y)} d_H(x,y) = \min_{x,y:f(x)\neq f(y)} |\mathrm{wt}(x) - \mathrm{wt}(y)|.$$

This equality need not be true for an arbitrary partial function, but holds for symmetric functions. First, consider the following standard fact.

▶ **Fact 26** ([32]). *Let* $p : \mathbb{R} \to \mathbb{R}$ *be a real polynomial. Let* $n \in \mathbb{N}$ *be a positive integer, and suppose that* $p(x) \in [0,1]$ *for all* $x \in [0,n]$*. Then*

$$\deg(p) \geq \sqrt{\frac{nc}{1+c}}, \qquad \text{where } c = \max_{x\in[0,n]} |p'(x)|\,.$$

We will use the standard symmetrization of a polynomial. Given a polynomial $p : \{0,1\}^n \to \mathbb{R}$, define its symmetrization $p_0 : \{0,1,\ldots,n\} \to \mathbb{R}$ by

$$p_0(t) \;=\; \mathbb{E}_{x:\mathrm{wt}(x)=t}\,[\,p(x)\,].$$

That is, $p_0(t)$ is the average value of $p(x)$ over all inputs of Hamming weight t. In this case, we say that p_0 is the symmetrized version of p.

Now, let us characterize $\widetilde{\deg}$, Q and R using GAP.

▶ **Lemma 27.** *Let* f *be a symmetric function. Then,*
$\widetilde{\deg}(f) =_{\mathrm{poly}} \mathrm{Q}(f) =_{\mathrm{poly}} \mathrm{R}(f) =_{\mathrm{poly}} \frac{n}{\mathrm{GAP}(f)}$.

Lastly, we evaluate D using GAP.

▶ **Lemma 28.** *Let f be a symmetric function. Then,* $\mathrm{D}(f) = n - \mathrm{GAP}(f) + 1$.

Combining Lemma 27 and Lemma 28 we can completely characterize D vs Q behavior for partial symmetric functions. For example, the Deutsch-Jozsa speedup [14] corresponds to $\mathrm{GAP}(f) = \frac{n}{2}$, giving $\mathrm{Q}(f) = \Theta(1)$ while $\mathrm{D}(f) = \Theta(n)$.

4.2 Functions defined on a slice

In this section, we will consider partial functions whose domain is a single slice of the Boolean cube. The deterministic and quantum query complexities of such functions were already considered by [7], who showed that $\mathrm{D}(f) = O(\mathrm{Q}(f)^{18})$. Here, we prove an alternate bound on $\mathrm{D}(f)$. Our main motivation is to give a fine-grained bound based on the slice's location. As such, our result is not comparable to that of [7]. Their result is tighter for slices near the edges, whereas our bound is tighter for slices near the middle section. In particular, we give a tighter bound for roughly balanced domains.

▶ **Definition 29** (Balanced blocks)**.** *We say that a block $B \subseteq [n]$ is balanced for $x \in \{0,1\}^n$ if*

$$|\{i \in [n] : i \in B \text{ and } x_i = 1\}| = |\{i \in [n] : i \in B \text{ and } x_i = 0\}|.$$

▶ Remark 30. For a function f defined on a slice, $x \in \mathrm{Dom}(f)$ if and only if $x^B \in \mathrm{Dom}(f)$ for every balanced block B. Hence, the block sensitivity $\mathrm{bs}(f)$ (Definition 13) can be restricted to only balanced blocks.

▶ **Lemma 31.** *Let $f : \mathrm{Dom}(f) \to \{0,1\}$ such that $\mathrm{Dom}(f) = \{x \in \{0,1\}^n : \mathrm{wt}(x) = k\}$ for some fixed $k \in [n]$. Then,* $\mathrm{D}(f) \leq \frac{3n}{\min\{k,n-k\}}\mathrm{C}(f)\mathrm{bs}(f)$.

It has been shown that $\mathrm{C}(f) \leq 3\mathrm{bs}(f)^2$ [7]. Therefore, combined with Lemmas 8 and 18, we have $\mathrm{D}(f) = O(\frac{n}{\min\{k,n-k\}}\mathrm{Q}(f)^6)$.

5 Quantum speedups through completions

Completion complexity

For a partial function $f : \mathrm{Dom}(f) \to \{0,1\}$, a total function $F : \{0,1\}^n \to \{0,1\}$ is a *completion* of f if $F(x) = f(x)$ for all $x \in \mathrm{Dom}(f)$. We denote the set of all completions of f as $\widehat{C}(f)$.

▶ **Definition 32** (Completion complexity of f with respect to query complexity measure M)**.** *Let f be a partial Boolean function and M be a query complexity measure. Then, the completion complexity of f (with respect to M), denoted as $\overline{M}(f)$ is*

$$\overline{M}(f) = \min_{F \in \widehat{C}(f)} M(F).$$

For a partial function f we use $F_M(f)$ to denote the completion that achieves the above minimum[7]. When f is clear from context, we drop it and simply write F_M. We say that F_M is an optimal completion of f with respect to M.

[7] If there are multiple completions that achieve the minimum, choose any one of them.

▶ **Definition 33** (Polynomially tight measures). *We say that measures $M_1, M_2, \ldots, M_l$ are polynomially tight if for all total functions F, we have, $M_1(F) =_{\text{poly}} M_2(F) =_{\text{poly}} \cdots =_{\text{poly}} M_l(F)$.*

Consider any partial function f and let $\mathcal{A}$ be the optimal deterministic algorithm. Since $\mathcal{A}$ produces a fixed output for each $x \notin \text{Dom}(f)$, we can define a completion $F_D(x) = \mathcal{A}(x)$. Thus:

▶ **Observation 34.** *For any partial function f, $D(f) = D(F_D)$.*

Combining everything, we get the following lemma: a measure M is extendable without a blow-up for f if and only if it is polynomially related to deterministic query complexity.

▶ **Lemma 35.** *For any* partial function $f : \text{Dom}(f) \to \{0,1\}$*, and for any complexity measure M which is polynomially related to* D *for* total functions*,*

$$\text{D}(f) =_{\text{poly}} M(f) \text{ iff } \overline{M}(f) =_{\text{poly}} M(f).$$

In particular, this lemma gives us a way to establish superpolynomial speedup between D and *an arbitrary measure M* by showing that the measure M is not completable to a total function without a superpolynomial blow-up.

References

1. Scott Aaronson and Andris Ambainis. The need for structure in quantum speedups. *Theory of Computing*, 10(6):133–166, 2014. doi:10.4086/toc.2014.v010a006.
2. Scott Aaronson and Shalev Ben-David. Sculpting Quantum Speedups. In *31st Conference on Computational Complexity (CCC 2016)*, volume 50 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 26:1–26:28. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.CCC.2016.26.
3. Scott Aaronson, Shalev Ben-David, Robin Kothari, Shravas Rao, and Avishay Tal. Degree vs. approximate degree and quantum implications of huang's sensitivity theorem. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1330–1342, 2021. doi:10.1145/3406325.3451047.
4. Anurag Anshu, Shalev Ben-David, and Srijita Kundu. On Query-To-Communication Lifting for Adversary Bounds. In *36th Computational Complexity Conference (CCC 2021)*, volume 200 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 30:1–30:39. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.CCC.2021.30.
5. Nikhil Balaji, Samir Datta, Raghav Kulkarni, and Supartha Podder. Graph properties in node-query setting: Effect of breaking symmetry. In Piotr Faliszewski, Anca Muscholl, and Rolf Niedermeier, editors, *41st International Symposium on Mathematical Foundations of Computer Science, MFCS 2016, Kraków, Poland, August 22-26, 2016*, volume 58 of *LIPIcs*, pages 17:1–17:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.MFCS.2016.17.
6. Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald De Wolf. Quantum lower bounds by polynomials. *Journal of the ACM (JACM)*, 48(4):778–797, 2001. doi:10.1145/502090.502097.
7. Shalev Ben-David. The structure of promises in quantum speedups. In *11th Conference on the Theory of Quantum Computation, Communication and Cryptography*, 2016.
8. Shalev Ben-David, Andrew M Childs, András Gilyén, William Kretschmer, Supartha Podder, and Daochen Wang. Symmetries, graph properties, and quantum speedups. *SIAM Journal on Computing*, 53(6):FOCS20–368, 2024.
9. Shalev Ben-David and Srijita Kundu. Separations in query complexity for total search problems. *arXiv preprint arXiv:2410.16245*, 2024. doi:10.48550/arXiv.2410.16245.

10 Harry Buhrman and Ronald de Wolf. Complexity measures and decision tree complexity: a survey. *Theoretical Computer Science*, 288(1):21–43, 2002. Complexity and Logic. doi:10.1016/S0304-3975(01)00144-X.

11 Mark Bun and Justin Thaler. Approximate degree in classical and quantum computing. *Coding for Interactive Communication: A Survey*, 15(3-4):229–423, 2022. doi:10.1561/0400000107.

12 André Chailloux. A note on the quantum query complexity of permutation symmetric functions. In *ITCS 2019-10th Annual Innovations in Theoretical Computer Science*, 2019.

13 Sourav Chakraborty, Anna Gál, Sophie Laplante, Rajat Mittal, and Anupa Sunny. Certificate Games. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference (ITCS 2023)*, volume 251 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 32:1–32:24, Dagstuhl, Germany, 2023. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2023.32.

14 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(1907):553–558, December 1992. doi:10.1098/rspa.1992.0167.

15 Irit Dinur, Yuval Filmus, and Prahladh Harsha. Analyzing boolean functions on the biased hypercube via higher-dimensional agreement tests. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '19, pages 2124–2133, USA, 2019. Society for Industrial and Applied Mathematics.

16 Irit Dinur, Yuval Filmus, and Prahladh Harsha. Sparse juntas on the biased hypercube. *TheoretiCS*, Volume 3, July 2024. doi:10.46298/theoretics.24.18.

17 Dmytro Gavinsky. Unambiguous parity-query complexity. *Random Structures & Algorithms*, 66(3):e70010, 2025. doi:10.1002/RSA.70010.

18 Mika Göös, Nathaniel Harms, Artur Riazanov, Anastasia Sofronova, Dmitry Sokolov, and Weiqiang Yuan. Pseudodeterministic communication complexity. *arXiv preprint arXiv:2511.04794*, 2025. doi:10.48550/arXiv.2511.04794.

19 Hao Huang. Induced subgraphs of hypercubes and a proof of the sensitivity conjecture. *Ann. Math.*, 190(3):949, November 2019.

20 Trinh Huynh and Jakob Nordstrom. On the virtue of succinct proofs: Amplifying communication complexity hardness to time-space trade-offs in proof complexity. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 233–248, 2012.

21 Peter Keevash, Noam Lifshitz, Eoin Long, and Dor Minzer. Hypercontractivity for global functions and sharp thresholds. *J. Amer. Math. Soc.*, July 2023.

22 Subhash Khot and Kunal Mittal. Biased Linearity Testing in the 1% Regime. In Srikanth Srinivasan, editor, *40th Computational Complexity Conference (CCC 2025)*, volume 339 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 10:1–10:23. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.CCC.2025.10.

23 Raghav Kulkarni and Supartha Podder. Quantum query complexity of subgraph isomorphism and homomorphism. In *33rd Symposium on Theoretical Aspects of Computer Science*, 2016.

24 Jeffrey Larson, Matt Menickelly, and Baoyu Zhou. Manifold sampling for optimizing nonsmooth nonconvex compositions. *SIAM Journal on Optimization*, 31(4):2638–2664, January 2021. doi:10.1137/20m1378089.

25 Noam Lifshitz and Dor Minzer. Noise sensitivity on the p -biased hypercube. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1205–1226, 2019. doi:10.1109/FOCS.2019.00075.

26 Valentino Magnani. Contact equations, lipschitz extensions and isoperimetric inequalities, 2009. arXiv:0711.5003.

27 Elchanan Mossel, Ryan O'Donnell, and Krzysztof Oleszkiewicz. Noise stability of functions with low influences: Invariance and optimality. *Ann. Math.*, 171(1):295–341, March 2010. doi:10.4007/annals.2010.171.295.

28 Ansh Nagda and Prasad Raghavendra. On optimal distinguishers for planted clique, 2025. doi:10.48550/arXiv.2505.01990.

29 Noam Nisan. Crew prams and decision trees. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 327–335, 1989. doi:10.1145/73007.73038.

30 Ramamohan Paturi. On the degree of polynomials that approximate symmetric boolean functions (preliminary version). In *Proceedings of the twenty-fourth annual ACM symposium on Theory of computing*, pages 468–474, 1992. doi:10.1145/129712.129758.

31 Supartha Podder, Penghui Yao, and Zekun Ye. On the fine-grained query complexity of symmetric functions. *computational complexity*, 34(1):1–77, 2025.

32 Ben-David Shalev. Polynomials, part 1: Symmetrization, 2020. URL: https://cs.uwaterloo.ca/~s4bendav/CS860S20.html.

33 Peter W Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM review*, 41(2):303–332, 1999. doi:10.1137/S0036144598347011.

34 Daniel R Simon. On the power of quantum computation. *SIAM journal on computing*, 26(5):1474–1483, 1997. doi:10.1137/S0097539796298637.

35 Perla Sousi and Sam Thomas. Cutoff for random walk on dynamical Erdős–Rényi graph. *Annales de l'Institut Henri Poincaré, Probabilités et Statistiques*, 56(4), November 2020. doi:10.1214/20-aihp1057.

36 Takashi Yamakawa and Mark Zhandry. Verifiable quantum advantage without structure. *Journal of the ACM*, 71(3):1–50, 2024.