Forrelation Is Extremally Hard

Uma Girish
Columbia University, New York, NY, USA

Rocco Servedio
Columbia University, New York, NY, USA

Abstract

The Forrelation problem is a central problem that demonstrates an exponential separation between quantum and classical capabilities. In this problem, given query access to n-bit Boolean functions f and g, the goal is to estimate the Forrelation function $\mathrm{forr}(f, g)$, which measures the correlation between g and the Fourier transform of f.

In this work we provide a new linear algebraic perspective on the Forrelation problem, as opposed to prior analytic approaches. We establish a connection between the Forrelation problem and *bent Boolean functions* and through this connection, analyze an *extremal* version of the Forrelation problem where the goal is to distinguish between extremal instances of Forrelation, namely (f, g) with $\mathrm{forr}(f, g) = 1$ and $\mathrm{forr}(f, g) = -1$.

We show that this problem can be solved with *one* quantum query and success probability *one*, yet requires $\tilde{\Omega}\left(2^{n/4}\right)$ classical randomized queries, even for algorithms with a one-third failure probability, highlighting the remarkable power of one exact quantum query. We also study a restricted variant of this problem where the inputs f, g are computable by small classical circuits and show classical hardness under cryptographic assumptions.

2012 ACM Subject Classification Theory of computation → Oracles and decision trees; Theory of computation → Complexity classes; Theory of computation → Quantum complexity theory

Keywords and phrases Forrelation, exact quantum, query complexity

Digital Object Identifier 10.4230/LIPIcs.ITCS.2026.72

Related Version *Full Version*: https://arxiv.org/abs/2508.02514

1 Introduction

Understanding the relative power of quantum versus classical computation is one of the major goals in complexity theory. Following the seminal work of Shor [38], it is widely believed that quantum computation is exponentially more powerful than classical computation; however, since we are unable to prove classical lower bounds for strong models of computation, there are relatively few settings in which such a separation can be unconditionally established. Query complexity is an important example of such a setting where we can unconditionally prove exponential quantum speedups.

In query complexity, we typically consider a Boolean function $f : \{0,1\}^n \to \{\pm 1\}$ and the objective is to compute some property of f by querying the values $f(x)$ on as few inputs $x \in \{0,1\}^n$ as possible. In the classical setting, the algorithm can adaptively and probabilistically choose inputs to query, and the goal is to solve the problem with high success probability, say at least $2/3$. In the quantum setting, the standard way to model a quantum query is by means of the unitary operator O_f which maps $|x\rangle$ to $|x\rangle\, f(x)$ for all $x \in \{0,1\}^n$ and as before, the goal is to compute some property of f with high success probability, while minimizing the number of calls to the unitary O_f. Numerous works [21, 40, 10, 1] have demonstrated properties of f that are exponentially easier to compute with quantum queries as opposed to classical queries. For instance, a version of periodicity testing [19] can be solved with $\mathrm{poly}(n)$ quantum queries, while requiring $2^{\Omega(n)}$ classical randomized queries;

this algorithm is a key subroutine in Shor's factoring algorithm [38], and the classical query lower bound helps explain some of the difficulty in finding efficient classical algorithms for factoring.

Understanding the strongest possible separation between quantum and classical computation has long been a topic of great interest. The overarching motivation here is to find a problem that is as easy as possible for quantum algorithms and as hard as possible for classical algorithms. In this context, a new problem called the *Forrelation problem* has emerged as a central concept.

In the Forrelation problem, given query access to Boolean functions f, g, the goal is to estimate the value of the Forrelation function $\mathrm{forr}(f,g) \in [-1,1]$, which captures the correlation between g and $\widehat{f}$, the Fourier transform of f. The Forrelation function has been used to establish numerous results about the power of quantum computation. The first such result is due to Aaronson [1], who showed that the Forrelation problem can be solved with high probability using just *one* quantum query, yet randomized algorithms require $2^{\Omega(n)}$ queries. This was quantitatively strengthened by [2] who defined variants of the Forrelation problem that are now known to demonstrate the largest possible separation between quantum and randomized query complexity for partial functions [2, 41, 37, 9, 14]. The Forrelation problem has since been used to prove many other results, including the celebrated oracle separation of BQP and PH [35]. Variants of this problem have been used to prove quantum lower bounds, such as a construction of a classical oracle relative to which $\mathsf{P} = \mathsf{NP}$ but $\mathsf{BQP} \neq \mathsf{QCMA}$ [4], as well as the existence of various quantum cryptography primitives [31, 32], separations between adaptive and non-adaptive quantum algorithms [27], and various separations between quantum and classical communication complexity [25, 26, 8].

However, all these works share one common limitation – they only establish classical hardness for estimating the Forrelation function *up to a constant less than one*. The best-known result in this context is due to Aaronson and Ambainis [2], who show that distinguishing between $\mathrm{forr}(f,g) \geq 2/\pi$ and $\mathrm{forr}(f,g) \leq -2/\pi$ requires $2^{\Omega(n)}$ classical randomized queries. The factor of $2/\pi$ arises from their analytic approach, which involves sampling Gaussian random variables and rounding them to $\{\pm 1\}$. All existing techniques use this framework and run into the same $2/\pi$ barrier. This naturally leads us to ask: just how hard is it to approximate the Forrelation function in the extremal case? In particular, Aaronson and Ambainis [2] ask the following question (see open question #4 in the discussion section of their paper): if we want a 1 versus $2^{\Omega(n)}$ separation between quantum and classical query complexity, how small can the error of the quantum algorithm be? More precisely, we ask:

$$\textit{How hard is it to distinguish } \mathrm{forr}(f,g) = 1 \textit{ from } \mathrm{forr}(f,g) = -1\textit{?}$$

These two extreme cases capture the largest and smallest possible values of the Forrelation function, and hence this question captures the hardness of approximating Forrelation to any non-trivial factor.

To study this problem, we introduce a fundamentally new way of looking at the Forrelation problem. In contrast to previous analytic approaches, which rely on rounding high-dimensional Gaussian distributions, our approach uses only simple linear algebra over $\mathbb{F}_2^n$ and elementary probabilistic arguments. We establish a novel connection between the Forrelation problem and *bent* functions, a well-studied concept in the analysis of Boolean functions. Using this connection, we show that despite the strong promise on the inputs, the extremal Forrelation problem is classically hard. Our main theorem establishes a bounded-error randomized lower bound of $2^{\Omega(n)}$ for this problem. In contrast, there is a simple quantum algorithm [1] that solves this problem with one quantum query and success probability one.

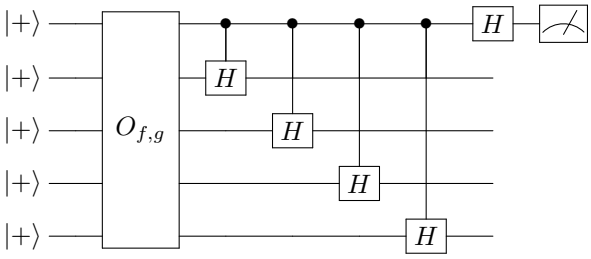

Figure 1 Quantum circuit for the Forrelation problem for $n = 5$. Here, $|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$ and $O_{f,g}$ is the oracle mapping basis states $|0\rangle|x\rangle$ to $|0\rangle|x\rangle f(x)$ and $|1\rangle|x\rangle$ to $|1\rangle|x\rangle g(x)$.

In the following section, we formally define the Forrelation problem, describe the history of the problem, and state our main results.

1.1 Forrelation Problem

To describe the Forrelation problem, we first need to introduce the concept of the Fourier transform and the Forrelation function. The Boolean Fourier transform, also known as the Walsh-Hadamard transform, is a central concept in Boolean function analysis which has applications to learning theory, social choice theory, circuit complexity, property testing, and quantum versus classical separations. It is defined as follows.

▶ **Definition 1** (Fourier Transform). *For $f : \mathbb{F}_2^n \to \mathbb{R}$, define $\widehat{f} : \mathbb{F}_2^n \to \mathbb{R}$ by*

$$\widehat{f}(y) := \frac{1}{2^n} \sum_{x \in \mathbb{F}_2^n} f(x)(-1)^{\langle x,y \rangle} \text{ for all } y \in \mathbb{F}_2^n.$$

We now define the Forrelation function, which has close connections with the Fourier transform of Boolean functions. The input to this function consists of the truth tables of two Boolean functions f and g and the output is the correlation between g and the Fourier transform of f.

▶ **Definition 2** (Forrelation Function). *The function* forr *is defined as follows. For Boolean functions $f, g : \mathbb{F}_2^n \to \{\pm 1\}$, define*

$$\begin{aligned} \mathrm{forr}(f,g) &:= \frac{1}{2^{n/2}} \sum_{y \in \mathbb{F}_2^n} \widehat{f}(y) g(y) \qquad (1) \\ &= \frac{1}{2^{3n/2}} \sum_{x,y \in \mathbb{F}_2^n} f(x) g(y) (-1)^{\langle x,y \rangle}. \end{aligned}$$

It is not too difficult to see that for any pair of Boolean functions (f, g), the value of $\mathrm{forr}(f, g)$ is between -1 and 1. Indeed, applying Cauchy-Schwarz on Equation (1) implies that $|\mathrm{forr}(f,g)| \leq 2^{-n/2} \sqrt{\sum_y \widehat{f}(y)^2} \sqrt{\sum_y g(y)^2}$. Since f and g are ± 1-valued functions, Parseval's theorem implies that $\sum_y \widehat{f}(y)^2 = \mathbf{E}_{\boldsymbol{x}}[f(\boldsymbol{x})^2] = \mathbf{E}_{\boldsymbol{y}}[g(\boldsymbol{y})^2] = 1$ and it follows that $|\mathrm{forr}(f,g)| \leq 1$.

The Forrelation function is also interesting from the perspective of quantum algorithms as it can be interpreted as the bias of a certain one-query quantum algorithm. More precisely, Aaronson [1] gave a simple quantum query algorithm that makes one call to $O_{f,g}$ and returns 1 with probability precisely $\frac{1}{2} + \frac{\mathrm{forr}(f,g)}{2}$ (see Figure 1 for an illustration of the algorithm.)

The intuition for this quantum algorithm comes from the ability of quantum circuits to implement the Fourier transform. We can view the Fourier transform as a unitary map that transforms the truth table of f into that of $\widehat{f}$ (up to a normalization factor of $2^{n/2}$). Additionally, this unitary map turns out to be a Hadamard matrix, i.e., a tensor product of n Hadamard gates. In contrast, it seems difficult to estimate the Forrelation of f and g using only classical queries to the truth tables of f and g. This motivates the definition of the Forrelation problem, where the goal is to estimate the Forrelation function up to a small additive error.

▶ **Definition 3** (Forrelation Problem). *Fix a parameter $0 \leq \varepsilon \leq 1$. Given query access to the truth tables of Boolean functions $f, g : \mathbb{F}_2^n \to \{\pm 1\}$ that are promised to satisfy either*

- *YES case:* $\mathrm{forr}(f, g) \geq \varepsilon$, *or*
- *NO case:* $\mathrm{forr}(f, g) \leq -\varepsilon$,

distinguish between the two cases.[1]

As mentioned before, there is a simple quantum algorithm that solves the Forrelation problem with one query, and the success probability of the algorithm is precisely $\frac{1}{2} + \frac{\varepsilon}{2}$ where ε is the underlying parameter. In particular, when $\varepsilon = 1$, the algorithm makes no error.

Limitations of Prior Works on Forrelation

Numerous works have established classical hardness of the Forrelation problem for ε bounded away from 1 [1, 2, 35, 9]. We will describe these results in more detail in Table 1, but all these works share a common limitation, which is that they only establish hardness for estimating the Forrelation *up to a global constant less than one*. The best-known constant is due to Aaronson and Ambainis [2], who showed that the Forrelation problem with $\varepsilon = 2/\pi - o(1)$ requires $2^{\Omega(n)}$ randomized queries.

1.2 Our Results

We consider the Forrelation problem with $\varepsilon = 1$, and call this the Extremal Forrelation Problem. Here, we are given query access to Boolean functions $f, g : \{0,1\}^n \to \{\pm 1\}$ that are promised to satisfy $|\mathrm{forr}(f, g)| = 1$, and we wish to tell whether $\mathrm{forr}(f, g) = 1$ or $\mathrm{forr}(f, g) = -1$. As mentioned before, the quantum algorithm for this problem follows immediately from the works of [1, 2]. When this algorithm is run on inputs to the Extremal Forrelation Problem it makes one query and solves the problem with success probability one. Our main theorem is that the classical randomized bounded-error query complexity of this problem is $2^{\Omega(n)}$.

▶ **Theorem 4.** *The Extremal Forrelation Problem, which is solvable with one quantum query and success probability one, requires $\tilde{\Omega}(2^{n/4})$ queries for any classical randomized query algorithm that succeeds with at least $2/3$ probability.*

We also analyze a variant of this problem where the oracles f, g are efficiently computable.

[1] In prior works, the goal of the Forrelation problem is usually to distinguish $\mathrm{forr}(f, g) \geq \varepsilon$ from $\mathrm{forr}(f, g) \leq \varepsilon/2$. Nevertheless, existing lower bounds also work for our variant of the problem; the proofs can be modified to produce two distributions supported on $\mathrm{forr}(f, g) \geq \varepsilon$ and $\mathrm{forr}(f, g) \leq -\varepsilon$, respectively, such that they are each indistinguishable from the uniform distribution over (f, g) for classical algorithms of small cost.

Table 1 Lower Bounds for the Forrelation Problem.

	Forrelation Problem	Classical Model	Classical Lower Bound
[1]	$\varepsilon = 0.05$	Randomized decision tree	$\tilde{\Omega}(2^{n/4})$ queries (depth)
[2]	$\varepsilon = 2/\pi$	Randomized decision tree	$\tilde{\Omega}(2^{n/2})$ queries
[35]	$\varepsilon = \Theta(1/n)$	Randomized AC0 circuits	$\exp\left(2^{\Omega(n/d)}\right)$ size
[9]	$\varepsilon = 1/2^{10}$	Randomized AC0 circuits	$\exp\left(2^{\Omega(n/d)}\right)$ size
[This work]	$\varepsilon = 1$	**Randomized decision tree**	$\tilde{\Omega}(2^{n/4})$ **queries**

▶ **Corollary 5.** *Suppose one-way functions exist against classical* $\mathrm{poly}(n)$*-time algorithms. Then, there is no* $\mathrm{poly}(n)$*-time randomized algorithm that solves the* Extremal Forrelation Problem *with probability at least* $2/3$*, even if the oracles* f, g *are computable by* $\mathrm{poly}(n)$*-sized classical circuits.*

We remark that the above result is proved in the black-box setting where the algorithm can only query the truth tables of f, g. We do not know if these results apply in the white-box setting where the algorithm is given an explicit description of a small circuit computing f, g.

We now mention a few consequences of our results.

Lower Bounds for the Forrelation Problem

Numerous works have studied lower bounds for the Forrelation problem. See Table 1 for a summary. The first classical lower bound for the Forrelation problem was established by Aaronson [1], who showed a $\tilde{\Omega}(2^{n/4})$ lower bound when $\varepsilon = 0.05$. This was improved to a $\tilde{\Omega}(2^{n/2})$ lower bound for $\varepsilon = 2/\pi - o(1)$ by Aaronson and Ambainis [2]. In their breakthrough result, Raz and Tal [35] proved lower bounds when $\varepsilon = \Theta(1/n)$. Although this choice of ε is smaller than in the previously mentioned works, the true strength of [35] lies in their classical lower bound which holds against a much more powerful model than classical query algorithms. Bansal and Sinha [9] strengthened this result by proving it in the regime of $\varepsilon = \Theta(1)$. It is worth emphasizing that a crucial aspect of this work – and a key reason they were able to resolve the conjecture of Aaronson and Ambainis on maximal separations – was their focus on lower bounds in the regime of $\varepsilon = \Theta(1)$ as opposed to $\varepsilon = \Theta(1/n)$. All of this underscores the difficulty and importance of proving lower bounds for the Forrelation problem, particularly as ε increases. In particular, as mentioned earlier, [2] have asked the following question (open question #4): how large ε can be for the Forrelation problem while remaining classical hard with $2^{\Omega(n)}$ queries? Our main theorem shows that even when ε is as large as can be (i.e., one), the Forrelation problem requires $2^{\Omega(n)}$ classical queries.

Efficient Oracle Separation

In [3], Aaronson and Chen ask, what happens if we consider quantum algorithms that can access an oracle, but we impose a constraint that the oracle has to be "physically realistic"? The motivation is to design a quantum advantage experiment by studying query complexity separations where the input oracles are implementable by small classical circuits. Motivated by this, we ask

How hard is it to estimate $\mathrm{forr}(f, g)$ *when* f, g *are computable by* $\mathrm{poly}(n)$*-sized circuits?*

Table 2 Speedups with Exact Quantum Computation.

	Exact Quantum	Classical Lower Bound	
	Queries	Queries	Success Probability
Deutsch-Jozsa [21]	1	$2^{\Omega(n)}$ O(1)	1 2/3
Simon's Problem [40, 13]	$O(n)$ (adaptive)	$2^{\Omega(n)}$	2/3
Welded Tree [17, 33]	$O(n^{2.5})$ (adaptive)	$2^{\Omega(n)}$	2/3
Order Finding (QFT) [34, 19]	$O(n)$ (parallel), can be made 1	$2^{\Omega(n)}$	2/3
Hidden Linear Structure (QFT) [20]	$O(n)$ (parallel), can be made 1	$2^{\Omega(n)}$	2/3
Extremal Forrelation Problem [this work]	1	$2^{\Omega(n)}$	2/3

As an easy corollary of our main result (Corollary 5), we show that if (classically secure) one-way functions exist, then there is no classical poly(n)-time algorithm for the Forrelation problem, even if f, g are computable by polynomial-sized circuits. As mentioned before, our result is a query complexity lower bound that holds in the black-box setting where the algorithm can only query the truth tables of f, g and does not have an explicit description of the circuits computing f, g.

Power of Exact Quantum Computation

Exact algorithms are algorithms that make no error, that is, they succeed with probability one. The power of exact quantum algorithms has been studied extensively in the past; see Table 2 for a summary. Numerous works have tried to understand the best possible separations between exact quantum and classical algorithms for total functions [6, 7, 15]. For partial functions, one of the earliest results is due to Deutsch–Jozsa [21], who showed that distinguishing between the constant function and a balanced function can be solved by an exact quantum algorithm with one query, but any zero-error randomized algorithm requires $2^{\Omega(n)}$ queries. However, the randomized query complexity drops to $O(1)$ if the algorithm is allowed to err with small probability. The first exponential speedup over bounded-error randomized algorithms is demonstrated by Simon's problem [40], which can be solved exactly [13] with $O(n)$ quantum queries, but requires $2^{\Omega(n)}$ classical queries in the bounded-error model. Since then, there have been other problems showing a similar separation using Discrete Fourier transforms, with the additional advantage that all the queries can be made in parallel [19, 34, 20]. While many of these works describe their quantum algorithms as making one query, in their models one can retrieve the truth table values at n different points with just one query; this corresponds to making $O(n)$ parallel queries in the standard model. In particular, they consider query access of the form $|x\rangle \to |f(x)\rangle$ where f has n-bit outputs, whereas the standard model only allows single-bit outputs. At first glance, it appears that such quantum algorithms need to make $\Omega(n)$ queries. However, using this, one can obtain a new Boolean function on $2n$-bit inputs with an *exact one-query* quantum algorithm such that every randomized algorithm requires $2^{\Omega(n)}$ queries[2]. The idea here is to

[2] We thank the anonymous reviewer for TQC 2025 pointing this out.

replace $f(x)$ by the Hadamard encoding of $f(x)$ and to use the Bernstein-Vazirani algorithm. Our main result (Theorem 4) achieves a similar separation, arguably for a simpler function and a simpler quantum protocol.

1.3 Outlook & Future Directions

We now highlight a number of open questions inspired by our work.

Optimal Separations

The bounded-error randomized query complexity of Extremal Forrelation Problem remains to be understood. The results of [2, 14] implies a $\tilde{O}(2^{n/2})$ upper bound, while we prove an $\tilde{\Omega}(2^{n/4})$ lower bound in Theorem 4. We conjecture that our lower bound can be improved to $\tilde{\Omega}(2^{n/2})$, matching the upper bound from [2, 14]. This would recover the separation from [2], with the additional advantage that the quantum algorithm succeeds with probability one.

▶ **Conjecture 6.** *The bounded-error randomized complexity of the Extremal Forrelation Problem is* $\tilde{\Omega}(2^{n/2})$.

We remark that Andrej Bogdanov and Yanbo Chen [12] have pointed out that for the particular $\mu^{\mathcal{H}}_{\text{yes}}$ and $\mu^{\mathcal{H}}_{\text{no}}$ distributions that we use to prove Theorem 4, there is a classical algorithm which makes $2^{n/4+o(n)}$ queries and with high probability distinguishes a random instance $(\boldsymbol{f}, \boldsymbol{g})$ drawn from $\mu^{\mathcal{H}}_{\text{yes}}$ (for which $\text{forr}(\boldsymbol{f}, \boldsymbol{g}) = 1$) from a random instance $(\boldsymbol{f}, \boldsymbol{g})$ drawn from $\mu^{\mathcal{H}}_{\text{no}}$ (for which $\text{forr}(\boldsymbol{f}, \boldsymbol{g}) = -1$). Hence, any proof of Conjecture 6 must use a different construction than the one used in our proof of Theorem 4.

More broadly, one can ask about optimal separations between quantum and classical query complexity. This question was studied by [2], who introduced a variant of Forrelation called k-Forrelation whose quantum query complexity is $\lceil k/2 \rceil$ and conjectured that its randomized query complexity is $\tilde{\Omega}(2^{n\cdot(1-1/k)})$. When $k = 2$, this problem is identical to the Forrelation problem as in Definition 3. They further conjectured that this is the best possible separation between bounded-error quantum and randomized query complexity. There have been a number of works on this topic [14, 41], culminating in the works of [37] and [9] that proved this conjecture. One can ask if a similar separation can be achieved by *exact* quantum algorithms. We conjecture that this can indeed be achieved.

▶ **Conjecture 7.** *For $k > 2$, the bounded-error randomized complexity of the extremal version of the k-Forrelation problem is* $\tilde{\Omega}(2^{n(1-1/k)})$.

Resolving this conjecture would shed light on the best possible separations between exact quantum and bounded-error randomized query complexity for partial functions.

Forrelation of Low-Degree Polynomials

Recently, there has been a lot of interest in studying the complexity of forrelation for instances in which f and g are degree-d $\mathbb{F}_2$-polynomials [24, 39]. It is not too difficult to define pairs of low-degree polynomials that have large Forrelation, for instance, in Definition 12, we can sample h to be a random degree-d $\mathbb{F}_2$-polynomial as opposed to a uniformly random Boolean function. This variant of Forrelation has been studied with the broad motivation of finding an efficient instantiation of an oracle which makes Forrelation classically hard. Towards this, researchers have tried to understand the computational complexity of estimating Forrelation for low-degree polynomials [24, 39].

We believe that understanding the query complexity of this version of Forrelation is independently interesting. In this setting, given query access to low-degree polynomials f, g, we wish to compute $\mathrm{forr}(f, g)$ using as few queries to the truth tables of f and g as possible. One trivial algorithm for this is to learn the polynomials using $\binom{n}{\leq d}$ queries each and then compute Forrelation offline. We conjecture that this algorithm is more or less optimal.

▶ **Conjecture 8.** *The* Extremal Forrelation Problem *where the inputs are restricted to degree-d $\mathbb{F}_2$-polynomials has randomized query complexity $n^{\Omega(d)}$.*

Proving this conjecture would give some evidence towards the hardness of computing forrelation even when the inputs are low-degree polynomials – it would indicate that computing the forrelation of low-degree polynomials is almost as hard as learning the polynomials.

Hardness Against Stronger Classical Models

One of the original motivations in [1] for introducing the Forrelation problem is that it was conjectured to be classically hard to compute, even for the relatively powerful model of AC^0 circuits. This conjecture was proved in the breakthrough result of Raz and Tal [35], resulting in an oracle separation between BQP and PH. We conjecture that a variant of the extremal Forrelation problem can be used to demonstrate a similar separation.

▶ **Conjecture 9.** *Depth-d AC^0 circuits require $\exp\left(2^{\Omega(n/d)}\right)$ size to solve the* Extremal Forrelation Problem.

We suspect that proving this conjecture would require studying a richer class of bent functions. We think this question is also interesting from the perspective of classical complexity theory, as it may lead to new techniques to prove AC^0 lower bounds.

1.4 Our Ideas

Before we go into our proof ideas, we first describe the existing approaches to proving Forrelation lower bounds and explain why they fail to work as ε tends to 1.

1.4.1 Why prior approaches don't work for $\varepsilon = 1$.

To prove randomized lower bounds on the Forrelation problem, one needs to produce hard distributions, i.e., distributions on the YES and NO instances of the problem that are classically hard to distinguish. For now, let us focus on generating YES instances. It is not too difficult to generate a pair of *real-valued* functions f, g with large Forrelation and $\mathbb{E}[f^2] = \mathbb{E}[g^2] = 1$; indeed, given any nonzero function f, we may define g to be proportional to $\widehat{f}$, with an appropriate normalization factor so as to satisfy $\mathrm{forr}(f, g) = 1$ and $\mathbb{E}[f^2] = \mathbb{E}[g^2] = 1$. The difficulty is in producing a pair of ± 1-valued functions with large Forrelation. Existing techniques to generate such pairs of functions follow essentially the same framework:

1. For each $x \in \mathbb{F}_2^n$, independently sample $\boldsymbol{f}(x)$ according to the Gaussian distribution with mean 0 and variance $\Theta(\varepsilon)$. For each $y \in \mathbb{F}_2^n$, define $\boldsymbol{g}(y) := 2^{n/2}\widehat{\boldsymbol{f}}(y)$.
2. For each $x, y \in \mathbb{F}_2^n$, independently round each $\boldsymbol{f}(x), \boldsymbol{g}(y)$ to ± 1.

It is not difficult to show that Item 1 generates a distribution on pairs of real-valued functions that with high probability have large Forrelation, namely at least $\Theta(\varepsilon)$. The goal of Item 2 is to modify these functions to produce ± 1-valued functions that continue to have large Forrelation, at least $\Theta(\varepsilon)$. This is where the choice of the rounding function and the parameter ε become crucial. The best parameters to date are obtained in [2], where Item 1

is as described above with $\varepsilon = 1$ and the rounding in Item 2 is done using the sign function, which maps non-negative reals to 1 and negative reals to -1. Using properties of the Gaussian distribution, [2] show that the resulting Boolean functions (namely $\mathrm{sign}(\boldsymbol{f}), \mathrm{sign}(\boldsymbol{g})$) have Forrelation at least $2/\pi$ in expectation.[3]

One might wonder if there is a different rounding procedure that produces a distribution on inputs with Forrelation very close to 1, but we suspect that this is not the case. The reason is that this framework is actually fairly oblivious to the underlying unitary matrix. In particular, one can consider the problem of estimating the *Rorrelation* function,

$$\mathrm{rorr}_U(f,g) := 2^{-n} \sum_{x,y \in \mathbb{F}_2^n} f(x)g(y)U_{x,y},$$

where U is any $2^n \times 2^n$ unitary matrix; note that when $U = H^{\otimes n}$ this is identical to the Forrelation function. The Rorrelation problem was introduced and studied by [41]. It turns out that the problem of estimating $\mathrm{rorr}_U(f,g)$ is classically hard for any unitary matrix whose entries are small (at most $1/2^{\Omega(n)}$ in magnitude), furthermore, this can be proved by using the above framework. In particular, if we sample a Haar random orthogonal matrix $\boldsymbol{U}$, all of its entries will be small in magnitude and the framework described above will establish classical hardness of estimating $\mathrm{rorr}_{\boldsymbol{U}}(f,g)$ up to a small global constant. On the other hand, the extremal version of the Rorrelation problem is vacuously easy for a Haar random orthogonal matrix $\boldsymbol{U}$. In more detail, in the full version of the paper, we prove the following:

▷ Claim 10. With extremely high probability over a Haar random orthogonal matrix $\boldsymbol{U}$,

$$\max_{f,g:\mathbb{F}_2^n \to \{\pm 1\}} |\mathrm{rorr}_{\boldsymbol{U}}(f,g)| < 0.99. \tag{2}$$

In other words, for a typical Haar random orthogonal matrix $\boldsymbol{U}$, there are *no* ± 1-valued functions for which $\mathrm{rorr}_{\boldsymbol{U}}(f,g) = \pm 1$ and thus, the extremal version of the Rorrelation problem becomes vacuously easy. There is something special about the Forrelation problem and the Hadamard matrix which makes it possible for there to exist even one pair of ± 1-valued functions (f,g) such that $|\mathrm{forr}(f,g)| = 1$.

It turns out that these extremal instances of the Forrelation problem arise from an important class of Boolean functions known as *bent* functions; we now describe this connection.

1.4.2 Connections to Bent Functions

Let us try to generate Boolean functions $f, g : \mathbb{F}_2^n \to \{\pm 1\}$ with as large Forrelation value as possible. To do this, let us first revisit the argument for why $\mathrm{forr}(f,g)$ cannot be larger than 1. Recall that

$$\mathrm{forr}(f,g) \triangleq 2^{-n/2} \sum_{y \in \mathbb{F}_2^n} \widehat{f}(y)g(y) \leq 2^{-n/2} \cdot \sqrt{\sum_y \widehat{f}(y)^2} \cdot \sqrt{\sum_y g(y)^2} = 1,$$

where the inequality is by Cauchy-Schwarz. For this to be tight, we need to set each $g(y)$ to be a multiple of $\widehat{f}(y)$. Due to the normalization factor, it turns out that we need to set $g(y) = 2^{n/2} \cdot \widehat{f}(y)$. In particular, since $g(y)$ is ± 1, it implies that each Fourier coefficient $\widehat{f}(y)$

[3] An intuition for the $2/\pi$ factor is as follows. A basic fact about the Gaussian distribution is that for nearly-uncorrelated Gaussians $\boldsymbol{f}(x)$ and $\boldsymbol{g}(y)$, we have $\mathbb{E}[\mathrm{sign}(\boldsymbol{f}(x)) \cdot \mathrm{sign}(\boldsymbol{g}(y))] \approx \frac{2}{\pi}\mathbb{E}[\boldsymbol{f}(x) \cdot \boldsymbol{g}(y)]$. Since the Forrelation function is a linear combination of terms of the form $\boldsymbol{f}(x) \cdot \boldsymbol{g}(y)$, we get that $\mathbb{E}[\mathrm{forr}(\mathrm{sign}(\boldsymbol{f}), \mathrm{sign}(\boldsymbol{g}))] \approx 2/\pi$.

is $\pm 2^{-n/2}$. Boolean functions with this property, namely that all of the Fourier coefficients have equal magnitude, are known as *bent* Boolean functions. These are precisely the functions that give rise to extremal instances of the Forrelation problem. In more detail, as shown above any pair of functions (f, g) whose Forrelation value is 1 must arise from a bent function f; conversely, it is easy to see that any bent function f gives rise to a Boolean function g (namely $g(y) := 2^{n/2} \cdot \widehat{f}(y)$) such that (f, g) has Forrelation 1.

Bent functions are extensively studied in Boolean function analysis. One important class of bent functions is the Maiorana-McFarland class of bent functions (see e.g. Section 6.1 of [16]). This family consists of all n-bit Boolean functions $(-1)^f$ where $f : \mathbb{F}_2^n \to \mathbb{F}_2$ is defined at $x \in \mathbb{F}_2^n$ by $f(x) := \langle x_1, x_2 \rangle + h(x_2)$ where n is even, x_1 and x_2 denote the first and second halves of x, and $h : \mathbb{F}_2^{n/2} \to \mathbb{F}_2$ is any Boolean function. Here, $\langle x_1, x_2 \rangle$ is the inner product function (mod 2). It is not too difficult to show that every function of this form is bent (see [22]; a proof of this is implicit in the proof of our Lemma 13); intuitively, the inner product function itself is a bent function and adding any Boolean function that only depends on the second half of x preserves the bent-ness. We will use the Maiorana-McFarland family to construct hard instances for our lower bound.

Related Works

There have been a few works [36, 18, 5] that observe the close connections between bent functions and the hidden shift problem (Simon's problem). Independently of our work, [23] observed the connection between bent functions and extremal instances of the Forrelation problem [4].

A related problem is property testing for the class of bent Boolean functions. Here, we are given query access to a Boolean function f and the objective is to tell whether f is bent or $\Omega(1)$-far from every bent function. This problem was studied by [11, 29]. In particular, [29] established an $\Omega(2^{n/4})$ lower bound on the bounded-error randomized query complexity of this problem and they used the Maiorana-McFarland family to construct hard distributions. In our problem, we are given truth table access to f and g that are promised to be bent and that satisfy either $g = \widehat{f} \cdot 2^{n/2}$ or $g = -\widehat{f} \cdot 2^{n/2}$ and we wish to tell these apart. As both f and g are promised to be bent and there is a high degree of correlation between f and g, this significantly complicates the problem and we need to introduce some new ideas in our proof.

1.4.3 Proof Sketch

The high-level structure of our argument is reminiscent of the lower bound argument of [29] (see Section 4 of [29]). For ease of notation, let $\chi : \mathbb{F}_2 \to \{\pm 1\}$ be the function mapping $x \to (-1)^x$. For a Boolean-valued function $f : \mathbb{F}_2^n \to \mathbb{F}_2$, let χ_f denote the function $(-1)^f$.

To explain the ideas underlying our proof, we instead consider the problem of distinguishing $\mathrm{forr}(f, g) = 1$ from $\mathrm{forr}(f, g) = 1/\sqrt{2^n}$. It is simpler to describe a pair of hard distributions for this variant of the problem, so we focus on it in the current sketch; however, the main section of our paper directly tackles the case of $\mathrm{forr}(f, g) = 1$ versus $\mathrm{forr}(f, g) = -1$. For the distributions we work with in the proof overview (Equations (3) and (4)), it is easier to pinpoint exactly where the classical hardness arises from, whereas the actual distributions we work with (Definition 12) have more complicated terms.

[4] We thank anonymous reviewers for TQC 2025 for pointing this out.

We will now define a distribution over pairs of functions (χ_f, χ_g) where $f, g : \mathbb{F}_2^n \to \mathbb{F}_2$ are Boolean functions. For the YES distribution, we will sample $\chi_{\boldsymbol{f}}$ to be a bent function. As described before, this determines a unique ± 1-valued function $\chi_{\boldsymbol{g}}$ (namely, $\chi_{\boldsymbol{g}} := 2^{n/2}\widehat{\chi_{\boldsymbol{f}}}$) such that $\mathrm{forr}(\chi_{\boldsymbol{f}}, \chi_{\boldsymbol{g}}) = 1$.[5] The process to generate $\boldsymbol{f}$ consists of two steps:

1. **Maiorana-McFarland family:** First, sample a random bent function from the Maiorana-McFarland family. We use this step to ensure that we only produce bent functions – this is essential to construct instances with $\mathrm{forr}(\chi_f, \chi_g) = 1$.
2. **Linear Transformation:** We then apply an invertible linear transformation on the input variables; doing this simply permutes the Fourier coefficients, and hence preserves bent-ness. We will show this step sufficiently masks the input and "hides" the inner-product structure, which effectively prevents classical algorithms from detecting structural properties by reading just a few coordinates.

In contrast, for the NO distribution, we will sample both f, g to be highly non-bent functions that mimic the Maiorana-McFarland family of bent functions. We now give a more formal description of this process. Sample $\boldsymbol{A} \in \mathbb{F}_2^{n \times n}$ to be a random invertible matrix, $\boldsymbol{h} : \mathbb{F}_2^{n/2} \to \mathbb{F}_2$ to be a uniformly random function, and let $\boldsymbol{B} = (\boldsymbol{A}^T)^{-1}$. For the YES distribution, define

$$\boldsymbol{f}(x) := \langle \boldsymbol{A}_1 x, \boldsymbol{A}_2 x \rangle + \boldsymbol{h}(\boldsymbol{A}_2 x) \quad \text{and} \quad \boldsymbol{g}(y) := \langle \boldsymbol{B}_1 y, \boldsymbol{B}_2 y \rangle + \boldsymbol{h}(\boldsymbol{B}_1 y), \tag{3}$$

where M_1, M_2 denote the upper and lower halves of a matrix M. For the NO distribution,

$$\boldsymbol{f}(x) := \boldsymbol{h}(\boldsymbol{A}_2 x) \quad \text{and} \quad \boldsymbol{g}(y) := \boldsymbol{h}(\boldsymbol{B}_1 y), \tag{4}$$

We then let $(\chi_{\boldsymbol{f}}, \chi_{\boldsymbol{g}})$ be the inputs to the Forrelation problem. It is not too difficult to show that $\boldsymbol{f}, \boldsymbol{g}$ in Equation (3) satisfy $\mathrm{forr}(\chi_{\boldsymbol{f}}, \chi_{\boldsymbol{g}}) = 1$ and similarly, in Equation (4) satisfy $\mathrm{forr}(\chi_{\boldsymbol{f}}, \chi_{\boldsymbol{g}}) = 1/\sqrt{2^n}$. The proof of this involves analyzing the Fourier transform of the Maiorana-McFarland family and applying a suitable change of variables; a more general version of this statement is proved in Lemma 13.

Classical Hardness

We will now give some intuition as to why these distributions are hard to distinguish by classical algorithms. Consider a randomized decision tree of depth ℓ that distinguishes these distributions with large advantage. By Yao's minmax principle, fixing the randomness of this algorithm, there exists a deterministic decision tree with the same properties. For now, let us assume that the tree is non-adaptive, that is, it fixes a set of ℓ points and this same set is queried along every root-to-leaf path. (This assumption turns out to be not so important, and the analysis for general adaptive algorithms closely follows the non-adaptive case.) Let $x_1, \ldots, x_k \in \mathbb{F}_2^n$ and $y_1, \ldots, y_{\ell-k} \in \mathbb{F}_2^n$ be the points queried by the non-adaptive decision tree. Here, x_i and y_j indicate points in the truth table of $\boldsymbol{f}$ and $\boldsymbol{g}$ respectively and $k \in \{0, \ldots, \ell\}$; in other words, the algorithm just queries $\boldsymbol{f}(x_1), \ldots, \boldsymbol{f}(x_k)$ and $\boldsymbol{g}(y_1), \ldots, \boldsymbol{g}(y_{\ell-k})$. We can assume without loss of generality that the x_i are distinct and similarly the y_j are distinct (there may be collisions between x_i and y_j). We will assume for the proof sketch that none of x_i, y_j are zero, since such queries are useless in distinguishing the YES and NO distributions (as the outcomes are the same for both distributions).

[5] Throughout the paper we use **bold font** to indicate random variables; note that in the current context both $\boldsymbol{f}$ and $\boldsymbol{g}$ are random variables, albeit (completely) correlated ones.

Let us look at the contribution of $\boldsymbol{h}(\circ)$ to each of $\boldsymbol{f}(x_1), \ldots, \boldsymbol{f}(x_k)$ and $\boldsymbol{g}(y_1), \ldots, \boldsymbol{g}(y_{\ell-k})$. Recalling the definition of the YES and NO distribution in Equations (3) and (4), it is easy to see that for both these distributions, the contribution of $\boldsymbol{h}(\circ)$ is given by

$$\boldsymbol{h}(\boldsymbol{A}_2 x_1), \ldots, \boldsymbol{h}(\boldsymbol{A}_2 x_k) \quad \text{and} \quad \boldsymbol{h}(\boldsymbol{B}_1 y_1), \ldots, \boldsymbol{h}(\boldsymbol{B}_1 y_{\ell-k}). \tag{5}$$

In particular, the sequence of points on which $\boldsymbol{h}(\circ)$ is implicitly queried is

$$\boldsymbol{A}_2 x_1, \ldots, \boldsymbol{A}_2 x_k \quad \text{and} \quad \boldsymbol{B}_1 y_1, \ldots, \boldsymbol{B}_1 y_{\ell-k}. \tag{6}$$

The main technical lemma of our work shows that if $\ell \leq 2^{cn}$, for a suitable absolute constant $c > 0$, then with high probability Equation (6) is a sequence of ℓ distinct points. (This is not necessarily the case if some of x_i, y_j are allowed to be zero, but as we argued before, such queries are useless and we can assume without loss of generality that $x_i, y_j \neq 0$.) A version of this is formalized in Lemma 17 and Lemma 18. Whenever this is the case, the sequence of outcomes in Equation (5) consists of uniform and independent random bits – indeed, $\boldsymbol{h}$ is a uniformly random function and hence its evaluations on distinct points are independent and uniform. This happens for both the YES and the NO distributions and as a result, any decision tree of depth $\ell \leq 2^{cn}$ cannot sufficiently distinguish these distributions. We now sketch the proof that Equation (6) is a distinct sequence with high probability.

Collision Probability Analysis

This is done in Section 3.1. We will analyze the probability that any two points in Equation (6) are equal and show that it is at most $2^{-\Omega(n)}$. We then apply a union bound over all pairs of points (there are at most ℓ^2 pairs) to conclude that with high probability, at least $1 - \ell^2 \cdot 2^{-\Omega(n)} \geq 2/3$, the sequence of points in Equation (6) is distinct. This along with the above paragraph would complete the proof.

Collisions within the $\boldsymbol{A}_2 x_i$ are significantly easier to analyze; the matrix $\boldsymbol{A}$ is distributed according to a uniformly random invertible matrix, and hence intuitively the lower half $\boldsymbol{A}_2$ has enough entropy to make collisions of the form $\boldsymbol{A}_2 x_i = \boldsymbol{A}_2 x_i'$ highly unlikely. Collisions within the $\boldsymbol{B}_1 y_j$ can be similarly controlled. We prove this in Lemma 18. Collisions between $\boldsymbol{A}_2 x_i$ and $\boldsymbol{B}_1 y_j$ are significantly harder to analyze, since $\boldsymbol{A}_2$ and $\boldsymbol{B}_1$ are correlated with each other due to the relationship $\boldsymbol{B} = (\boldsymbol{A}^T)^{-1}$, but this can nevertheless be shown.

The General Case: Forrelation 1 versus −1

We now describe the additional ideas required to prove the classical hardness of the EXTREMAL FORRELATION PROBLEM. In order to generate instances with $\mathrm{forr}(f, g) = 1$ and $\mathrm{forr}(f, g) = -1$, we are forced to sample f to be a bent function and we are forced to set g so that either $\chi_g = +\widehat{\chi_f}$ or $\chi_g = -\widehat{\chi_f}$. If we were to naively modify the aforementioned YES distribution to a NO distribution by letting $\chi_g = -\widehat{\chi_f}$, then the resulting YES and NO distributions would become easy to classically distinguish: querying the YES distribution on $x = 0$ and $y = 0$ would produce identical answers, namely $(-1)^{\boldsymbol{h}(0)}$ and $(-1)^{\boldsymbol{h}(0)}$ while querying the NO distribution on these points would produce different answers, namely $(-1)^{\boldsymbol{h}(0)}$ and $-(-1)^{\boldsymbol{h}(0)}$. In order to get around this issue, in Item 2 of our actual hard distributions we need to apply an *affine* transformation on the input variables, instead of a *linear* transformation. This has the effect of shifting the origin, which intuitively means that the classical algorithm "does not know" which point to query in order to see this correlated pair of answers. Remarkably, applying this random shift also simplifies the collision probability analysis in the case of collisions between $\boldsymbol{A}_2 x_i$ and $\boldsymbol{B}_1 y_j$. For more details, see Definition 12.

1.4.4 Organization

We describe our hard distributions in Section 2. We prove that these are indeed valid distributions (Lemma 13) and prove some some key properties about them (Lemma 14 and Corollary 16). In Section 3, we present the main technical lemma (Lemma 17) and prove the main theorem assuming this in Section 3.2. In Section 3.1, we prove the main lemma.

2 Hard Distributions for the Extremal Forrelation Problem

As described in the introduction, the hard instances of our problem will be based on the Maiorana-McFarland family (see e.g. Section 6.1 of [16]) of bent functions – this family consists of "inner-product-like" functions. To sample our hard instances, we will first sample an affine shift, followed by a random "inner-product-like" function under this affine shift. We describe this in more detail below.

2.1 Descriptions of Hard Distributions

Notation

For simplicity of notation, define the function $\chi : \mathbb{F}_2 \to \{\pm 1\}$ mapping x to $(-1)^x$. For any Boolean-valued function $f : \mathbb{F}_2^n \to \mathbb{F}_2$, let $\chi_f : \mathbb{F}_2^n \to \{\pm 1\}$ denote the function $\chi(f)(x) = (-1)^{f(x)}$. For matrices $A, B \in \mathbb{F}_2^{n \times n}$, let $A = \begin{bmatrix} A_1 \\ A_2 \end{bmatrix}$ and $B = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}$ where A_1, A_2 and B_1, B_2 are $n/2 \times n$ matrices representing the upper and lower halves of A and B respectively. We use $\langle x, y \rangle$ to denote the inner product (mod 2) between vectors $x, y \in \mathbb{F}_2^n$.

We now proceed to the description of the hard distributions. We first define a joint distribution on 4-tuples (A, B, a, b) where A, B are matrices and a, b are affine shifts (vectors) as follows.

▶ **Definition 11.** *Let $\boldsymbol{A} \sim \mathbb{F}_2^{n \times n}$ be a uniformly random matrix of full rank and let $\boldsymbol{B} := (\boldsymbol{A}^T)^{-1}$. Let $\boldsymbol{a} \sim \mathbb{F}_2^n$ be a uniformly vector and let $\boldsymbol{b} = \boldsymbol{B}^T \begin{bmatrix} \boldsymbol{B}_2 \\ \boldsymbol{B}_1 \end{bmatrix} \boldsymbol{a}$.*

Let $\mathcal{L}$ be the induced distribution on $(\boldsymbol{A}, \boldsymbol{B})$. We use $\mathcal{L}_{21}$, $\mathcal{L}_2$ and $\mathcal{L}_1$ to denote the induced distribution on $(\boldsymbol{A}_2, \boldsymbol{B}_1)$, $\boldsymbol{A}_2$ and $\boldsymbol{B}_1$ respectively. We now define the hard instances of the Extremal Forrelation Problem for a family of functions $\mathcal{H}$.

▶ **Definition 12** (Hard Instances of Extremal Forrelation Problem)**.** *Let $\mathcal{H}$ be any collection of boolean functions mapping $\mathbb{F}_2^{n/2}$ to $\{0, 1\}$. Sample $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{a}, \boldsymbol{b})$ as in Definition 11. Sample $\boldsymbol{h} : \mathbb{F}_2^{n/2} \to \{0, 1\}$ to be a uniformly random Boolean function in $\mathcal{H}$. Define Boolean functions $\boldsymbol{f}, \boldsymbol{g} : \mathbb{F}_2^n \to \mathbb{F}_2$ as follows.*

$$\boldsymbol{f}(x) := \langle \boldsymbol{A}_1 x, \boldsymbol{A}_2 x \rangle + \langle x, \boldsymbol{a} \rangle + \boldsymbol{h}(\boldsymbol{A}_2 x)$$
$$\boldsymbol{g}(y) := \langle \boldsymbol{B}_1 y, \boldsymbol{B}_2 y \rangle + \langle y, \boldsymbol{b} \rangle + \boldsymbol{h}(\boldsymbol{B}_1 y + \boldsymbol{B}_1 \boldsymbol{a}) + \langle \boldsymbol{B}_1 \boldsymbol{a}, \boldsymbol{B}_2 \boldsymbol{a} \rangle .$$

Let $\mu_{\text{yes}}^{\mathcal{H}}$ and $\mu_{\text{no}}^{\mathcal{H}}$ be the induced distributions on $(\chi_{\boldsymbol{f}}, \chi_{\boldsymbol{g}})$ and $(\chi_{\boldsymbol{f}}, -\chi_{\boldsymbol{g}})$ respectively for $h \sim \mathcal{H}$.

We will later instantiate $\mathcal{H}$ in various ways. The following lemma shows that regardless of $\mathcal{H}$, these are indeed valid distributions, that is, $\mu_{\text{yes}}^{\mathcal{H}}$ and $\mu_{\text{no}}^{\mathcal{H}}$ are indeed supported on the YES and NO instances of the Extremal Forrelation Problem.

▶ **Lemma 13.** *For (χ_f, χ_g) in the support of $\mu^{\mathcal{H}}_{\text{yes}}$ and $\mu^{\mathcal{H}}_{\text{no}}$, we have* $\text{forr}(f,g) = 1$ *and* $\text{forr}(f,g) = -1$ *respectively (regardless of $\mathcal{H}$).*

Proof of Lemma 13. Fix any A, B, a, b as in Definition 12 and let f, g be as specified in Definition 12. For any $y \in \mathbb{F}_2^n$, consider

$$\begin{aligned}\widehat{\chi_f}(y) &= \frac{1}{2^n}\sum_{z\in\mathbb{F}_2^n} \chi_f(z)\cdot\chi\left(\langle z, y\rangle\right)\\ &= \frac{1}{2^n}\sum_{z\in\mathbb{F}_2^n} \chi\left(f(z)+\langle z, y\rangle\right)\\ &= \frac{1}{2^n}\sum_{z\in\mathbb{F}_2^n} \chi\left(\langle A_1 z, A_2 z\rangle + \langle z, a\rangle + h(A_2 z) + \langle z, y\rangle\right).\end{aligned}$$

Since A is invertible we can do a change of variables $x \leftarrow Az$, which in particular gives us $x_1 \leftarrow A_1 z$ and $x_2 \leftarrow A_2 z$. This lets us rewrite the above as

$$\widehat{\chi_f}(y) = \frac{1}{2^n}\sum_{x\in\mathbb{F}_2^n} \chi\left(\langle x_1, x_2\rangle + \left\langle A^{-1}x, a\right\rangle + h(x_2) + \left\langle A^{-1}x, y\right\rangle\right).$$

We now observe that $\left\langle A^{-1}x, \circ\right\rangle = \langle x, B\circ\rangle$, since $(A^{-1})^T = B$. Substituting this above, we see that

$$\widehat{\chi_f}(y) = \frac{1}{2^n}\sum_{x\in\mathbb{F}_2^n} \chi\left(\langle x_1, x_2\rangle + \langle x, Ba\rangle + h(x_2) + \langle x, By\rangle\right).$$

We now express $\langle x, B\circ\rangle$ as $\langle x_1, B_1\circ\rangle + \langle x_2, B_2\circ\rangle$. Thus,

$$\widehat{\chi_f}(y) = \frac{1}{2^n}\sum_{x\in\mathbb{F}_2^n} \chi\left(\langle x_1, x_2\rangle + \langle x_1, B_1 a\rangle + \langle x_2, B_2 a\rangle + h(x_2) + \langle x_1, B_1 y\rangle + \langle x_2, B_2 y\rangle\right).$$

We now group the terms based on x_1.

$$\begin{aligned}\widehat{\chi_f}(y) &= \frac{1}{2^n}\sum_{x\in\mathbb{F}_2^n} \chi\left(\langle x_1, x_2 + B_1 a + B_1 y\rangle + \langle x_2, B_2 a + B_2 y\rangle + h(x_2)\right)\\ &= \frac{1}{2^n}\sum_{x_2\in\mathbb{F}_2^{n/2}}\sum_{x_1\in\mathbb{F}_2^{n/2}} \chi\left(\langle x_1, x_2 + B_1 a + B_1 y\rangle\right)\cdot\chi\left(\langle x_2, B_2 a + B_2 y\rangle + h(x_2)\right).\end{aligned}$$

Observe that the first term $\chi\left(\langle x_1, x_2 + B_1 a + B_1 y\rangle\right)$ when summed over all $x_1 \in \mathbb{F}_2^{n/2}$ is non-zero only if $x_2 + B_1 a + B_1 y = 0$ (equivalently, $x_2 = B_1 a + B_1 y$); and if it is non-zero, it equals $2^{n/2}$. Thus, we have

$$\widehat{\chi_f}(y) = \frac{1}{2^{n/2}}\chi\left(\langle B_1 a + B_1 y, B_2 a + B_2 y\rangle + h(B_1 a + B_1 y)\right).$$

We now expand the terms in the R.H.S. to obtain

$$\begin{aligned}\widehat{\chi_f}(y) &= \frac{1}{2^{n/2}}\chi\left(\langle B_1 y, B_2 y\rangle + \left\langle \begin{bmatrix}B_1\\ B_2\end{bmatrix} y, \begin{bmatrix}B_2\\ B_1\end{bmatrix} a\right\rangle + h(B_1 a + B_1 y) + \langle B_1 a, B_2 a\rangle\right)\\ &= \frac{1}{2^{n/2}}\chi\left(\langle B_1 y, B_2 y\rangle + \left\langle y, \begin{bmatrix}B_1^T & B_2^T\end{bmatrix}\begin{bmatrix}B_2\\ B_1\end{bmatrix} a\right\rangle + h(B_1 a + B_1 y) + \langle B_1 a, B_2 a\rangle\right).\end{aligned}$$

Recall that we have $b = B^T \begin{bmatrix} B_2 \\ B_1 \end{bmatrix} a = \begin{bmatrix} B_1^T & B_2^T \end{bmatrix} \begin{bmatrix} B_2 \\ B_1 \end{bmatrix} a$. Substituting this in the above equation, we have

$$\widehat{\chi_f}(y) = \frac{1}{2^{n/2}} \chi \left(\langle B_1 y, B_2 y \rangle + \langle y, b \rangle + h(B_1 a + B_1 y) \right) + \langle B_1 a, B_2 a \rangle .$$

Recalling the definition of $g(y)$ from Definition 12, we see that

$$\widehat{\chi_f}(y) \triangleq \frac{1}{2^{n/2}} \chi_g(y). \tag{7}$$

We now use the defining equation for the Forrelation function (Equation (1)) to get

$$\mathrm{forr}(\chi_f, \chi_g) \triangleq \frac{1}{2^{n/2}} \sum_{y \in \mathbb{F}_2^n} \widehat{\chi_f}(y) \chi_g(y).$$

Combining this and Equation (7), we see that $\mathrm{forr}(\chi_f, \chi_g) = \frac{1}{2^n} \sum_y \chi_g(y)^2 = 1$. It can be similarly shown that $\mathrm{forr}(\chi_f, -\chi_g) = \frac{1}{2^n} \sum_y (-1) \cdot \chi_g(y)^2 = -1$. This completes the proof of Lemma 13. ◀

2.2 Characterizing the Marginals of the Distributions

Recall that we used $\mathcal{L}$ to denote the induced distribution on $(\boldsymbol{A}, \boldsymbol{B})$, and $\mathcal{L}_2$ and $\mathcal{L}_1$ to denote the induced distribution on $\boldsymbol{A}_2$ and $\boldsymbol{B}_1$ respectively. In this section, we will characterize the marginal distributions $\mathcal{L}_1$ and $\mathcal{L}_2$, which turn out to be identical, as follows:

▶ **Lemma 14.** *Each of the distributions $\mathcal{L}_1$ and $\mathcal{L}_2$ is precisely the uniform distribution over all matrices in $\mathbb{F}_2^{n/2 \times n}$ that have full row-rank.*

We then show the following fact about the row-rank of a uniformly random rectangular matrix.

▶ **Fact 15.** *A uniformly random matrix in $\mathbb{F}_2^{n/2 \times n}$ has full row-rank with probability at least $1 - (n/2)2^{-n/2}$.*

As an immediate corollary of this and Lemma 14, we obtain the following.

▶ **Corollary 16.** *Each of the distributions $\mathcal{L}_1$ and $\mathcal{L}_2$ is $(n/2) \cdot 2^{-n/2}-$close to the uniform distribution over $\mathbb{F}_2^{n/2 \times n}$ in total variational distance.*

We now give the proofs of Lemma 14 and Fact 15.

Proof of Lemma 14. We will prove this for the distribution $\mathcal{L}_1$ and the argument for $\mathcal{L}_2$ is identical. Recall that $\boldsymbol{B}$ is sampled according to a uniformly random full-rank matrix. Fix any full row-rank matrix B_1. We will count the number of matrices B_2 such (B_1, B_2) has full rank. We will show that this number is precisely $(2^n - 2^{n/2}) \times (2^n - 2^{n/2+1}) \times \ldots \times (2^{n/2} - 2^{n-1})$. This would complete the proof.

To count the number of B_2, we first count the number of possibilities for each row of B_2. Firstly, each row of B_2 must not lie in the span of the previous rows of B_2 and the rows of B_1. The first row of B_2 is any vector not in the span of the rows of B_1 and thus has $2^n - 2^{n/2}$ possibilities. Having fixed this, the second row of B_2 can be any vector that is not in the span of the first row of B_2 and the rows of B_1, and thus has $2^n - 2^{n/2+1}$ possibilities. We repeat this argument and at the i-th step, we choose a vector that is not in the $n/2 + i - 1$-dimensional space spanned by the first $i - 1$ rows of B_2 and the $n/2$ rows of

B_1; this can be done in $2^n - 2^{n/2+i-1}$ ways. Doing this for all $n/2$ rows shows that the total number of possibilities for B_2 is precisely $\prod_{i=1}^{n/2}(2^n - 2^{n/2+i-1})$ and this completes the proof of Lemma 14. ◀

Proof of Fact 15. We perform a calculation identical to that in Lemma 14. By a similar argument, we can show that the number of matrices in $\mathbb{F}_2^{n/2\times n}$ with full row-rank is precisely

$$\prod_{i=1}^{n/2}(2^n - 2^{i-1}).$$

Thus, the probability that a uniformly random $n/2 \times n$ matrix is of full row-rank is precisely

$$\frac{\prod_{i=1}^{n/2}(2^n - 2^{i-1})}{2^{n^2/2}} = \prod_{i=1}^{n/2}\left(\frac{2^n - 2^{i-1}}{2^n}\right) \geq (1 - 2^{-n/2})^{n/2} \geq 1 - (n/2)2^{-n/2}.$$

This completes the proof of Fact 15. ◀

3 Classical Lower Bound

The main technical ingredient in the classical lower bound will be Lemma 17. We will describe this lemma and its proof in Section 3.1. We will then prove Theorem 4 in Section 3.2 assuming this.

3.1 Statement of Lemma 17 and its Proof

The main technical ingredient is the following.

▶ **Lemma 17.** *Let $x, y \in \{0,1\}^n$ be any two vectors. Then,*

$$\Pr_{\substack{(\boldsymbol{A}_2,\boldsymbol{B}_1)\sim\mathcal{L}_{21}\\ \boldsymbol{a}\sim\mathbb{F}_2^n}}[\boldsymbol{A}_2 x = \boldsymbol{B}_1 y + \boldsymbol{B}_1\boldsymbol{a}] = 2^{-n/2}.$$

In the above lemma, $(\boldsymbol{A}_2, \boldsymbol{B}_1) \sim \mathcal{L}_{21}$ and $\boldsymbol{a} \sim \mathbb{F}_2^n$ are sampled independently, just as in Definition 11. We will also require the following lemma.

▶ **Lemma 18.** *For $x \neq x' \in \mathbb{F}_2^n$ and $y \neq y' \in \mathbb{F}_2^n$, we have*

$$\Pr_{\boldsymbol{A}_2\sim\mathcal{L}_2}[\boldsymbol{A}_2 x = \boldsymbol{A}_2 x'] \leq (n/2+1)\cdot 2^{-n/2}$$

$$\Pr_{\boldsymbol{B}_1\sim\mathcal{L}_1}[\boldsymbol{B}_1 y = \boldsymbol{B}_1 y'] \leq (n/2+1)\cdot 2^{-n/2}.$$

We now prove these two lemmas.

Proof of Lemma 17. Let $\mathcal{E}$ be the event $\boldsymbol{A}_2 x = \boldsymbol{B}_1 y + \boldsymbol{B}_1\boldsymbol{a}$. This is equivalent to $\boldsymbol{B}_1\boldsymbol{a} = \boldsymbol{A}_2 x + \boldsymbol{B}_1 y$. Now, regardless of what x, y are, the vector $\boldsymbol{a}$ is distributed as a uniformly random vector in $\mathbb{F}_2^n$ that is *independent* of $\boldsymbol{A}, \boldsymbol{B}$, because $\boldsymbol{a}$ is sampled uniformly and independently of $\boldsymbol{A}, \boldsymbol{B}$. Furthermore, $\boldsymbol{B}_1$ has full row-rank. Therefore, fixing $\boldsymbol{A} = A$ and $\boldsymbol{B} = B$, we see that the probability over $\boldsymbol{a}$ that $B_1\boldsymbol{a}$ is equal to $A_2 x + B_1 y$ is precisely $2^{-n/2}$. This completes the proof. ◀

Proof of Lemma 18. We prove this lemma for $\boldsymbol{A}_2 \sim \mathcal{L}_2$ and the proof of $\boldsymbol{B}_1 \sim \mathcal{L}_1$ is identical. The event we wish to bound the probability of is $[\boldsymbol{A}_2(x - x') = 0]$. We use Corollary 16 to conclude that the total variational distance between $\mathcal{L}_2$ and the uniform distribution is at most $(n/2) \cdot 2^{-n/2}$. Let us now work with a uniformly random matrix $\boldsymbol{A}_2$. Since $(x - x') \neq 0$, the vector $\boldsymbol{A}_2(x - x')$ is a uniformly random vector in $\mathbb{F}_2^{n/2}$. Therefore, the probability that it is zero is at most $2^{-n/2}$. This completes the proof. ◀

We now complete the proof of Theorem 4 using these lemmas.

3.2 Proof of Theorem 4

Let $\mathcal{H}$ be any family of all $n/2$-variate boolean functions. Consider a classical randomized query protocol for the Extremal Forrelation Problem with D queries. Recall from Lemma 13 that $\mu_{\text{yes}}^{\mathcal{H}}$ and $\mu_{\text{no}}^{\mathcal{H}}$ are supported on the yes and no instances of the Extremal Forrelation Problem. Given a randomized query protocol with D queries for the Extremal Forrelation Problem that succeeds with at least $2/3$ probability, by Yao's principle, there exists a *deterministic* decision tree of depth D that distinguishes $\mu_{\text{yes}}^{\mathcal{H}}$ and $\mu_{\text{no}}^{\mathcal{H}}$ with advantage at least $1/3$.

Given such a deterministic decision tree of depth $D \leq 2^{n/4}/6$ and a root-to-leaf path $\mathcal{P}$ of length ℓ in the tree, each node in the path $\mathcal{P}$ corresponds to either a query of the form $f(x)$ (where the truth table of f is probed), or a query of the form $g(y)$ (where the truth table of g is probed). We assume without loss of generality that the query vectors for f are distinct and similarly the query vectors for g are distinct (there may be common query vectors which are given to both f and g). We will then use the following claims.

▷ Claim 19. Consider any deterministic decision tree of depth $D \leq 2^{n/4}/(6\sqrt{n})$, and fix any root-to-leaf path $\mathcal{P}$ of length $\ell \leq D$ in the tree. Let $x^{(1)}, \ldots, x^{(k)} \in \mathbb{F}_2^n$ and $y^{(1)}, \ldots, y^{(\ell-k)} \in \mathbb{F}_2^n$ be the sequence of vectors queried. Let $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{a}$ be distributed as in Definition 11, and consider the sequence of points

$$\boldsymbol{A}_2 x^{(1)}, \ldots, \boldsymbol{A}_2 x^{(k)} \quad \text{and} \quad \boldsymbol{B}_1 y^{(1)} + \boldsymbol{B}_1 \boldsymbol{a}, \ldots, \boldsymbol{B}_1 y^{(\ell-k)} + \boldsymbol{B}_1 \boldsymbol{a}. \tag{8}$$

Let $\mathcal{E}$ be the event that (8) is a sequence of ℓ *distinct* points. Then, $\Pr_{\mu_{\text{yes}}^{\mathcal{H}}}[\mathcal{E}], \Pr_{\mu_{\text{no}}^{\mathcal{H}}}[\mathcal{E}] \geq 9/10$.

▷ Claim 20. Let $\mathcal{H}$ be the family of all $n/2$-variate boolean functions. Under the same hypothesis as Claim 19, whenever $\mathcal{E}$ occurs, the probability of taking path $\mathcal{P}$ for the distributions $\mu_{\text{yes}}^{\mathcal{H}}|\mathcal{E}$ and $\mu_{\text{no}}^{\mathcal{H}}|\mathcal{E}$ is exactly $2^{-\ell}$.

Remark

Claim 20 is the *only* place where the properties of $\mathcal{H}$ come into play – every other part of the proof is *independent* of $\mathcal{H}$.

Once we have these claims, the proof follows quite easily. Let $\mathcal{H}$ be the family of all $n/2$-variate boolean functions. Let us look at the induced distributions $\mu_{\text{yes}}^{\text{leaf},\mathcal{H}}$ and $\mu_{\text{no}}^{\text{leaf},\mathcal{H}}$ on the leaves of the decision tree when the inputs are sampled according to $\mu_{\text{yes}}^{\mathcal{H}}$ and $\mu_{\text{no}}^{\mathcal{H}}$ respectively and let μ be the distribution on the leaves induced by a truly random walk down the tree. Claim 19 and Claim 20 imply that for any leaf, the probability that $\mu_{\text{yes}}^{\text{leaf},\mathcal{H}}$ and $\mu_{\text{no}}^{\text{leaf},\mathcal{H}}$ assign to that leaf are each at least $9/10$ times the probability assigned by μ. This implies that there exists distributions $\tilde{\mu}_{\text{yes}}^{\text{leaf},\mathcal{H}}, \tilde{\mu}_{\text{no}}^{\text{leaf},\mathcal{H}}$ on the leaves such that

$$\mu_{\text{yes}}^{\text{leaf},\mathcal{H}} = \tfrac{9}{10}\mu + \tfrac{1}{10}\tilde{\mu}_{\text{yes}}^{\text{leaf},\mathcal{H}},$$

$$\mu_{\text{no}}^{\text{leaf},\mathcal{H}} = \tfrac{9}{10}\mu + \tfrac{1}{10}\tilde{\mu}_{\text{no}}^{\text{leaf},\mathcal{H}}.$$

This implies that the total variational distance between $\mu_{\text{yes}}^{\text{leaf},\mathcal{H}}$ and $\mu_{\text{no}}^{\text{leaf},\mathcal{H}}$ is at most $1/10$ and hence, the output distributions of the decision tree on inputs sampled according to $\mu_{\text{yes}}^{\mathcal{H}}$ and $\mu_{\text{no}}^{\mathcal{H}}$ differ in total variational distance by at most $1/10$. This contradicts the assumption that the decision tree distinguishes these distributions with advantage at least $1/3$. This completes the proof of Theorem 4 assuming Claim 19 and Claim 20. We will now prove these claims.

3.3 Proof of Claim 19 and Claim 20

We now proceed to the proof of Claim 19, which is where we will use Lemma 17 and Lemma 18. We will then prove Claim 20 which relies primarily on the properties of $\mathcal{H}$.

Proof of Claim 19. We first analyze the probability of $\mathcal{E}$ when the distribution on inputs is $\mu_{\text{yes}}^{\mathcal{H}}$. The calculation for $\mu_{\text{no}}^{\mathcal{H}}$ is identical and is omitted. Note that (8) is a random sequence of points (where the randomness comes from $\boldsymbol{A}, \boldsymbol{B}$ and $\boldsymbol{a}$). As stated in Claim 20, let $\mathcal{E}$ be the event that this is a sequence of ℓ *distinct* points. We will now argue that $\mathcal{E}$ is a high-probability event.

First, let us bound the probability of collisions within the $\boldsymbol{A}_2 x^{(i)}$. Fix any $i \neq i' \in [k]$. We apply Lemma 18 to the vectors $x^{(i)} \neq x^{(i')}$. Lemma 18 implies that

$$\mathbf{Pr}[\boldsymbol{A}_1 x^{(i)} = \boldsymbol{A}_1 x^{(i')}] \leq (n/2+1) \cdot 2^{-n/2}.$$

We now apply a union bound over $i, i' \in [k]$. There are at most k^2 possibilities to union bound over. This implies that with probability at least $1 - k^2 \cdot (n/2+1) \cdot 2^{-n/2}$, the sequence of points

$$\boldsymbol{A}_2 x^{(1)}, \ldots, \boldsymbol{A}_2 x^{(k)}$$

is a sequence of k distinct points. We can similarly argue about collisions within the $\boldsymbol{B}_1 y^{(j)}$ to conclude that with probability at least $1 - (\ell-k)^2 \cdot (n/2+1) \cdot 2^{-n/2}$, the sequence of points

$$\boldsymbol{B}_1 y^{(1)} + \boldsymbol{B}_1 \boldsymbol{a}, \ldots, \boldsymbol{B}_1 y^{(\ell-k)} + \boldsymbol{B}_1 \boldsymbol{a}$$

is a sequence of $\ell - k$ distinct points. Finally, we argue about collisions between pairs of the form $\boldsymbol{A}_2 x^{(i)}$ and $\boldsymbol{B}_1 y^{(j)} + \boldsymbol{B}_1 \boldsymbol{a}$. Fix any $i \in [k]$ and $j \in [\ell - k]$. We apply Lemma 17 to the vectors $x^{(i)}$ and $y^{(j)}$. Lemma 17 implies that

$$\mathbf{Pr}[\boldsymbol{A}_2 x^{(i)} = \boldsymbol{B}_1 y^{(j)} + \boldsymbol{B}_1 \boldsymbol{a}] \leq 2^{-n/2}.$$

We now apply a union bound over (i, j). There are at most $k \cdot (\ell - k)$ possibilities to union bound over. This implies that with probability at least $1 - k \cdot (\ell-k) \cdot 2^{-n/2}$, there are no collisions among pairs of the form $\boldsymbol{A}_2 x^{(i)}$ and $\boldsymbol{B}_1 y^{(j)} + \boldsymbol{B}_1 \boldsymbol{a}$. So by a union bound, the total probability of the bad event $\neg\mathcal{E}$ is at most

$$k^2 \cdot (n/2+1) \cdot 2^{-n/2} + (\ell-k)^2 \cdot (n/2+1) \cdot 2^{-n/2} + k \cdot (\ell-k) \cdot 4 \cdot 2^{-n/2} \leq 3 \cdot D^2 \cdot n \cdot 2^{-n/2}.$$

for large enough n. Recall that we set $D \leq 2^{n/4}/(6\sqrt{n})$, so we get that

$$3 \cdot D^2 \cdot n \cdot 2^{-n/2} \leq 3 \cdot 2^{n/2} \cdot \tfrac{1}{36n} \cdot n \cdot 2^{-n/2} < \tfrac{1}{10}.$$

This shows that $\mathbf{Pr}_{\mu_{\text{yes}}^{\mathcal{H}}}[\mathcal{E}] \geq 9/10$. The argument for $\mu_{\text{no}}^{\mathcal{H}}$ is identical. This completes the proof of Claim 19. ◁

Proof of Claim 20. We first analyze the probability of receiving any particular sequence of outcomes when the distribution on inputs is $\mu^{\mathcal{H}}_{\text{yes}}$. The calculation for $\mu^{\mathcal{H}}_{\text{no}}$ is identical and is omitted.

Recall the $\mu^{\mathcal{H}}_{\text{yes}}$ distribution. This is obtained by sampling $\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{a}, \boldsymbol{b}$ as in Definition 12, sampling $\boldsymbol{h}$ uniformly at random from $\mathcal{H}$, and outputting $(\chi_{\boldsymbol{f}}, \chi_{\boldsymbol{g}})$ where $\boldsymbol{f}, \boldsymbol{g} : \mathbb{F}_2^n \to \mathbb{F}_2$ are defined as:

$$\begin{aligned}\boldsymbol{f}(x) &:= \langle \boldsymbol{A}_1 x, \boldsymbol{A}_2 x\rangle + \langle x, \boldsymbol{a}\rangle + \boldsymbol{h}(\boldsymbol{A}_2 x)\\ \boldsymbol{g}(y) &:= \langle \boldsymbol{B}_1 y, \boldsymbol{B}_2 y\rangle + \langle y, \boldsymbol{b}\rangle + \boldsymbol{h}(\boldsymbol{B}_1 y + \boldsymbol{B}_1 \boldsymbol{a}) + \langle \boldsymbol{B}_1 \boldsymbol{a}, \boldsymbol{B}_2 \boldsymbol{a}\rangle\,.\end{aligned}$$

Along the path $\mathcal{P}$, we have queried $\boldsymbol{f}(x^{(1)}), \ldots, \boldsymbol{f}(x^{(k)})$ and $\boldsymbol{g}(y^{(1)}), \ldots, \boldsymbol{g}(y^{(\ell-k)})$. Let us consider the contribution of $\boldsymbol{h}(\circ)$ to these query responses. This is given by evaluating $\boldsymbol{h}(\circ)$ on the following sequence of points

$$\boldsymbol{A}_2 x^{(1)}, \ldots, \boldsymbol{A}_2 x^{(k)} \quad \text{and} \quad \boldsymbol{B}_1 y^{(1)} + \boldsymbol{B}_1 \boldsymbol{a}, \ldots, \boldsymbol{B}_1 y^{(\ell-k)} + \boldsymbol{B}_1 \boldsymbol{a}$$

which is precisely the sequence given in Equation (8). We will now argue that when $\mathcal{E}$ happens, the probability of taking this path under $\mu^{\mathcal{H}}_{\text{yes}}$ (and similarly $\mu^{\mathcal{H}}_{\text{no}}$) is precisely $2^{-\ell}$. Let us compute the probability of taking the path $\mathcal{P}$ under $\mu^{\mathcal{H}}_{\text{yes}}$ conditioned on $\mathcal{E}$ happening. When $\mathcal{E}$ happens, we have that the sequence of points

$$\boldsymbol{A}_2 x^{(1)}, \ldots, \boldsymbol{A}_2 x^{(k)} \quad \text{and} \quad \boldsymbol{B}_1 y^{(1)} + \boldsymbol{B}_1 \boldsymbol{a}, \ldots, \boldsymbol{B}_1 y^{(\ell-k)} + \boldsymbol{B}_1 \boldsymbol{a}$$

is a sequence of ℓ distinct points. We now observe that the evaluations of the function $\boldsymbol{h}$ on these points are independent and uniformly random bits in $\{0,1\}$. In other words,

$$\boldsymbol{h}(\boldsymbol{A}_2 x^{(1)}), \ldots, \boldsymbol{h}(\boldsymbol{A}_2 x^{(k)}) \quad \text{and} \quad \boldsymbol{h}(\boldsymbol{B}_1 y^{(1)} + \boldsymbol{B}_1 \boldsymbol{a}), \ldots, \boldsymbol{h}(\boldsymbol{B}_1 y^{(\ell-k)} + \boldsymbol{B}_1 \boldsymbol{a})$$

is a sequence of uniformly random bits in $\{0,1\}$ when $\boldsymbol{h} \sim \mathcal{H}$. Thus, the probability of receiving any particular sequence of outcomes when querying the truth tables of $\boldsymbol{f}, \boldsymbol{g}$ at the ℓ vectors $x^{(1)}, \ldots, x^{(k)}$, $y^{(1)}, \ldots, y^{(\ell-k)}$ is either exactly $2^{-\ell}$. This completes the proof. ◁

3.4 Proof of Corollary 5

Proof of Corollary 5. Let $\mathcal{H}$ be the family of all boolean functions $\{h : \{0,1\}^{n/2} \to \{0,1\}\}$. It is well known [30, 28] that if one-way functions exist, then we can construct a family of pseudorandom functions $\mathcal{H}' := \{h_\lambda : \{0,1\}^{n/2} \to \{0,1\}\}_{\lambda \in \{0,1\}^{k(n)}}$ with $k(n) = \text{poly}(n)$ such that

- **Efficiency:** there is a poly(n)-sized classical circuit that computes $h_\lambda(x)$ given inputs $\lambda \in \{0,1\}^{k(n)}$ and $x \in \{0,1\}^{n/2}$.
- **Security:** Any classical polynomial-time algorithm $\mathcal{A}$ that queries the truth-table of an $n/2$-bit Boolean function cannot sufficiently distinguish a uniformly random function in $\mathcal{H}'$ from a truly uniformly random function, i.e.,

$$\left| \mathop{\mathbf{E}}_{\boldsymbol{\lambda} \sim \{0,1\}^{k(n)}} \left[\mathcal{A}^{h_{\boldsymbol{\lambda}}(x)}(1^n)\right] - \mathop{\mathbf{E}}_{\boldsymbol{h} \sim \mathcal{H}} \left[\mathcal{A}^{\boldsymbol{h}(x)}(1^n)\right] \right| \leq \text{negl}(n).$$

Let $\mu^{\mathcal{H}}_{\text{yes}}$ and $\mu^{\mathcal{H}}_{\text{no}}$ be the hard distributions as in Definition 12 defined with respect to $\mathcal{H}$ and similarly $\mu^{\mathcal{H}'}_{\text{yes}}$ and $\mu^{\mathcal{H}'}_{\text{no}}$ be defined with respect to $\mathcal{H}'$.

Implementability of f, g. For any pair (f, g) drawn from either $\mu_{\text{yes}}^{\mathcal{H}'}$ or $\mu_{\text{no}}^{\mathcal{H}'}$, the functions f, g are computable by polynomial-sized circuits. In more detail, for a fixed draw of A, B, a, b, λ, the circuit for f takes input x, first computes the inner products $\langle A_1 x, A_2 x\rangle$ and $\langle x, a\rangle$, then computes the vector $A_2 x$, and finally applies the circuit for h_λ on this vector and thus computes $f(x) = \langle A_1 x, A_2 x\rangle + \langle x, a\rangle + h_\lambda(A_2 x)$. All of these operations can be implemented by $\text{poly}(n)$-sized classical circuits and the circuit for g is analogous. We will now show that under the cryptographic assumption, there is no classical algorithm that distinguishes $\mu_{\text{yes}}^{\mathcal{H}'}$ and $\mu_{\text{no}}^{\mathcal{H}'}$ with $\text{poly}(n)$ queries.

Classical Indistinguishability of f, g. Let $\mathcal{A}'$ be any classical algorithm that makes at most $\text{poly}(n)$ queries to the truth tables of f and g. Theorem 4 shows that $\mathcal{A}'$ cannot distinguish $\mu_{\text{yes}}^{\mathcal{H}}$ and $\mu_{\text{no}}^{\mathcal{H}}$ with advantage more than $\text{negl}(n)$[6]. We will now show that under the cryptographic assumption, $\mathcal{A}'$ cannot distinguish $\mu_{\text{yes}}^{\mathcal{H}}$ and $\mu_{\text{yes}}^{\mathcal{H}'}$ with more than $\text{negl}(n)$ advantage. By the same argument, an analogous statement holds for the distributions $\mu_{\text{no}}^{\mathcal{H}}$ and $\mu_{\text{no}}^{\mathcal{H}'}$. Consequently, by the triangle inequality, we get that $\mathcal{A}'$ cannot distinguish $\mu_{\text{yes}}^{\mathcal{H}'}$ and $\mu_{\text{no}}^{\mathcal{H}'}$ with more than $\text{negl}(n)$ advantage and this completes the proof.

To see that $\mathcal{A}'$ cannot distinguish $\mu_{\text{yes}}^{\mathcal{H}}$ and $\mu_{\text{yes}}^{\mathcal{H}'}$ with more than $\text{negl}(n)$ advantage, we show that any such algorithm can be turned into a distinguisher for $\mathcal{H}$ and $\mathcal{H}'$. Indeed, consider the algorithm $\mathcal{A}$ that given query access to an unknown $n/2$-bit boolean function h, first samples $(\boldsymbol{A}, \boldsymbol{B}, \boldsymbol{a}, \boldsymbol{b})$ as in Definition 11 and runs $\mathcal{A}'$ on the pair of functions (f, g) as defined in the YES distribution in Definition 12. Similarly to the argument before, each query to f at the point x can be simulated by making a query to h at the point $A_2 x$ and computing $f(x) = \langle A_1 x, A_2 x\rangle + \langle x, a\rangle + h(A_2 x)$, and similarly for g. Thus, a distinguisher for $\mu_{\text{yes}}^{\mathcal{H}'}$ and $\mu_{\text{yes}}^{\mathcal{H}}$ can be turned into one for $\mathcal{H}'$ and $\mathcal{H}$ with the same number of queries and same advantage. This completes the proof. ◀

References

1 Scott Aaronson. BQP and the polynomial hierarchy. In *Proceedings of the 61st IEEE Annual Symposium on Foundations of Computer Science*, pages 141–150, 2010. doi:10.1145/1806689.1806711.

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

3 Scott Aaronson and Lijie Chen. Complexity-Theoretic Foundations of Quantum Supremacy Experiments. In *32nd Computational Complexity Conference (CCC 2017)*, pages 22:1–22:67, 2017. doi:10.4230/LIPIcs.CCC.2017.22.

4 Scott Aaronson, DeVon Ingram, and William Kretschmer. The acrobatics of BQP. In *Proceedings of the 37th Computational Complexity Conference*, pages 20:1–20:17, 2022. doi:10.4230/LIPIcs.CCC.2022.20.

5 Serge Adonsou, Peter Bruin, Maris Ozols, and Joppe Stokvis. Hidden shift problem for complex functions. Available at https://arxiv.org/pdf/2507.19440, 2025.

6 Andris Ambainis. Superlinear advantage for exact quantum algorithms. *SIAM J. Comput.*, 45:617–631, 2012.

[6] While the statement of Theorem 4 only refers to 1/3 advantage, the proof (Claim 19) shows that the advantage of a classical algorithm in solving the EXTREMAL FORRELATION PROBLEM scales as $2^{-n/2}$ times D^2 where D is the number of queries. In particular, for $\text{poly}(n)$-time algorithms (which make at most at most $\text{poly}(n)$ queries), this advantage is $\text{negl}(n)$.

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

8 Srinivasan Arunachalam and Uma Girish. Trade-Offs Between Entanglement and Communication. In *38th Computational Complexity Conference (CCC 2023)*, pages 25:1–25:23, 2023. doi:10.4230/LIPIcs.CCC.2023.25.

9 Nikhil Bansal and Makrand Sinha. k-forrelation optimally separates quantum and classical query complexity. In Samir Khuller and Virginia Vassilevska Williams, editors, *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 1303–1316, 2021. doi:10.1145/3406325.3451040.

10 Ethan Bernstein and Umesh Vazirani. Quantum complexity theory. *SIAM Journal on Computing*, 26(5):1411–1473, 1997. doi:10.1137/S0097539796300921.

11 Abhishek Bhrushundi, Sourav Chakraborty, and Raghav Kulkarni. Property testing bounds for linear and quadratic functions via parity decision trees. In *Computer Science - Theory and Applications - 9th International Computer Science Symposium in Russia*, pages 97–110, 2014. doi:10.1007/978-3-319-06686-8_8.

12 Andrej Bogdanov and Yanbo Chen, 2025. Personal communication.

13 Gilles Brassard and Peter Høyer. An exact quantum polynomial-time algorithm for simon's problem. In *Fifth Israel Symposium on Theory of Computing and Systems, ISTCS 1997*, pages 12–23, 1997. doi:10.1109/ISTCS.1997.595153.

14 Sergey Bravyi, David Gosset, Daniel Grier, and Luke Schaeffer. Classical algorithms for forrelation. Available as arXiv e-print at https://arxiv.org/abs/2102.06963, 2022. arXiv:2102.06963.

15 Harry Buhrman, Richard Cleve, Ronald de Wolf, and Christof Zalka. Bounds for small-error and zero-error quantum algorithms. In *Proceedings of 40th Annual Symposium on Foundations of Computer Science*, pages 358–368, 1999. doi:10.1109/SFFCS.1999.814607.

16 Claude Carlet and Sihem Mesnager. Four decades of research on bent functions. *Des. Codes Cryptogr.*, 78:78:5–78:50, 2016. doi:10.1007/S10623-015-0145-8.

17 Andrew M. Childs, Richard Cleve, Enrico Deotto, Edward Farhi, Sam Gutmann, and Daniel A. Spielman. Exponential algorithmic speedup by a quantum walk. In *Proceedings of the 35th Annual ACM Symposium on Theory of Computing (STOC 2003)*, pages 59–68, 2003. doi:10.1145/780542.780552.

18 Andrew M. Childs, Robin Kothari, Maris Ozols, and Martin Roetteler. Easy and hard functions for the Boolean hidden shift problem. In *8th Conference on the Theory of Quantum Computation, Communication and Cryptography, TQC 2013*, pages 50–79, 2013. doi:10.4230/LIPIcs.TQC.2013.50.

19 Richard Cleve. The query complexity of order-finding. *Information and Computation*, 192(2):162–171, 2004. doi:10.1016/J.IC.2004.04.001.

20 J. Niel de Beaudrap, Richard Cleve, and John Watrous. Sharp quantum vs. classical query complexity separations. Available as arXiv e-print at https://arxiv.org/abs/quant-ph/0011065, 2001. arXiv:quant-ph/0011065.

21 David Deutsch and Richard Jozsa. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences*, 439:553–558, 1992. URL: https://api.semanticscholar.org/CorpusID:121702767.

22 J.F. Dillon. *Elementary Hadamard Difference Sets*. PhD thesis, University of Maryland, College Park, 1974.

23 Suman Dutta, Subhamoy Maitra, and Chandra Sekhar Mukherjee. Following forrelation – quantum algorithms in exploring Boolean functions' spectra. *Advances in Mathematics of Communications*, 18(1):1–25, 2024. doi:10.3934/amc.2021067.

24 Alexandru Georghiu. Verifiable quantum advantage: old and new ideas, 2025. Link to recording of talk: https://youtu.be/7NqAcaSwKf8?si=2ZBkEkbiVaUMSzMJ.

25 Uma Girish, Ran Raz, and Avishay Tal. Quantum versus randomized communication complexity, with efficient players. *Comput. Complex.*, 31(2):17, 2022. doi:10.1007/S00037-022-00232-7.

26 Uma Girish, Ran Raz, and Wei Zhan. Lower Bounds for XOR of Forrelations. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2021)*, pages 52:1–52:14, 2021. doi:10.4230/LIPIcs.APPROX/RANDOM.2021.52.

27 Uma Girish, Makrand Sinha, Avishay Tal, and Kewen Wu. The power of adaptivity in quantum query algorithms. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 1488–1497, 2024. doi:10.1145/3618260.3649621.

28 Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *J. ACM*, 33(4):792–807, August 1986. doi:10.1145/6490.6503.

29 Elena Grigorescu, Karl Wimmer, and Ning Xie. Tight lower bounds for testing linear isomorphism. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 16th International Workshop, APPROX 2013, and 17th International Workshop, RANDOM 2013*, pages 559–574, 2013. doi:10.1007/978-3-642-40328-6_39.

30 Johan HÅstad, Russell Impagliazzo, Leonid A. Levin, and Michael Luby. A pseudorandom generator from any one-way function. *SIAM Journal on Computing*, 28(4):1364–1396, 1999. doi:10.1137/S0097539793244708.

31 William Kretschmer, Luowen Qian, Makrand Sinha, and Avishay Tal. Quantum cryptography in Algorithmica. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, pages 1589–1602, 2023. doi:10.1145/3564246.3585225.

32 William Kretschmer, Luowen Qian, and Avishay Tal. Quantum-computable one-way functions without one-way functions. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, pages 189–200, 2025. doi:10.1145/3717823.3718144.

33 Guanzhong Li, Lvzhou Li, and Jingquan Luo. Recovering the original simplicity: succinct and deterministic quantum algorithm for the welded tree problem. In *ACM-SIAM Symposium on Discrete Algorithms*, pages 2454–2480, 2023.

34 Michele Mosca and Christof Zalka. Exact quantum Fourier transforms and discrete logarithm algorithms. *International Journal of Quantum Information*, 02(01):91–100, 2004.

35 Ran Raz and Avishay Tal. Oracle separation of BQP and PH. *J. ACM*, 69(4):30:1–30:21, 2022. doi:10.1145/3530258.

36 Martin Rotteler. Quantum algorithms for highly non-linear Boolean functions. *Proceedings of the 2010 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 448–457, 2020. doi:10.1137/1.9781611973075.37.

37 Alexander A. Sherstov, Andrey A. Storozhenko, and Pei Wu. An optimal separation of randomized and quantum query complexity. *SIAM J. Comput.*, 52(2):525–567, 2023. doi:10.1137/22M1468943.

38 Peter W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997. doi:10.1137/S0097539795293172.

39 Noah Shutty. Simons institute summer cluster on quantum computing: Lightning talks, 2025. Link to recording of talk: https://www.youtube.com/live/7F5LBNGDRmk?si=NhPVNL25qGTtcngP&t=2651.

40 Daniel R. Simon. On the power of quantum computation. *SIAM J. Comput.*, 26(5):1474–1483, October 1997. doi:10.1137/S0097539796298637.

41 Avishay Tal. Towards optimal separations between quantum and randomized query complexities. In *Proceedings of the 61st IEEE Annual Symposium on Foundations of Computer Science*, pages 228–239, 2020. doi:10.1109/FOCS46700.2020.00030.