Electronic Colloquium on Computational Complexity, Report No. 157 (2023)

One-Way Communication Complexity of Partial XOR Functions

Vladimir V. Podolskii[1] and Dmitrii Sluch[2]

[1] *Tufts University*
[2] *Nebius Israel*

Abstract

Boolean function $F(x, y)$ for $x, y \in \{0,1\}^n$ is an XOR function if $F(x, y) = f(x \oplus y)$ for some function f on n input bits, where $\oplus$ is a bit-wise XOR. XOR functions are relevant in communication complexity, partially for allowing Fourier analytic technique. For total XOR functions it is known that deterministic communication complexity of F is closely related to parity decision tree complexity of f. Montanaro and Osbourne (2009) observed that one-sided communication complexity $\mathrm{D}^{\rightarrow}_{\mathrm{cc}}(F)$ of F is exactly equal to nonadaptive parity decision tree complexity $\mathrm{NADT}^{\oplus}(f)$ of f. Hatami et al. (2018) showed that unrestricted communication complexity of F is polynomially related to parity decision tree complexity of f.

We initiate the studies of a similar connection for partial functions. We show that in case of one-sided communication complexity whether these measures are equal, depends on the number of undefined inputs of f. More precisely, if $\mathrm{D}^{\rightarrow}_{\mathrm{cc}}(F) = t$ and f is undefined on at most $O\left(\frac{2^{n-t}}{\sqrt{n-t}}\right)$, then $\mathrm{NADT}^{\oplus}(f) = t$. We provide improved bounds on the number of undefined inputs for $t = 1, 2$. On the other end of the spectrum, we observe that measures are equal for any partial function f satisfying $\mathrm{NADT}^{\oplus}(f) \geq n - 1$.

We show that the restriction on the number of undefined inputs in these results is unavoidable. That is, for a wide range of values of $\mathrm{D}^{\rightarrow}_{\mathrm{cc}}(F)$ and $\mathrm{NADT}^{\oplus}(f)$ (from constant to $n - 2$) we provide partial functions (with more than $\Omega\left(\frac{2^{n-t}}{\sqrt{n-t}}\right)$ undefined inputs) for which $\mathrm{D}^{\rightarrow}_{\mathrm{cc}}(F) < \mathrm{NADT}^{\oplus}(f)$. In particular, we provide a function with an exponential gap between the two measures. Our separation results translate to the case of two-sided communication complexity as well, in particular showing that the result of Hatami et al. (2018) cannot be generalized to partial functions.

Previous results for total functions heavily rely on Boolean Fourier analysis and thus, the technique does not translate to partial functions. For the proofs of our results we build a linear algebraic framework instead. Separation results are proved through the reduction to covering codes.

1 Introduction

In communication complexity model two players, Alice and Bob, are computing some fixed function $F\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ on a given input (x, y). However, Alice knows only

ISSN 1433-8092

x and Bob knows only y. The main object of studies in communication complexity is the amount of communication $\mathrm{D_{cc}}(F)$ needed between Alice and Bob to compute the function.

Function F is an XOR-function if for all $x, y \in \{0,1\}^n$ we have $F(x,y) = f(x \oplus y)$ for some $f\colon \{0,1\}^n \to \{0,1\}$, where $x \oplus y$ is a bit-wise XOR of Boolean vectors x and y. XOR-functions are important in communication complexity [28, 19, 26, 27, 3, 13, 15, 1, 24, 22, 5, 2, 8, 11, 9], on one hand, since there are important XOR-functions defined based on Hamming distance between x and y, and on the other hand, since the structure of XOR-functions allows for Fourier analytic techniques. In particular, this connection suggests an approach for resolving Log-rank Conjecture for XOR-functions [28, 13].

In recent years there was considerable progress in the characterization of communication complexity of a XOR-function F in terms of the complexity of f in parity decision tree model. In this model the goal is to compute a fixed function f on an unknown input $x \in \{0,1\}^n$ and in one step we are allowed to query XOR of any subset of input bits. We want to minimize the number of queries that is enough to compute f on any input x. The complexity of f in this model is denoted by $\mathrm{DT}^{\oplus}(f)$. It was shown by Hatami et al. [13] that for any total f we have $\mathrm{D_{cc}}(F) = \mathrm{poly}(\mathrm{DT}^{\oplus}(f))$.

Even stronger connection holds for one-way communication complexity case. In this setting only very restricted form of communication is allowed: Alice sends Bob a message based on x and Bob has to compute the output based on this message and y. We denote the complexity of F in this model by $\mathrm{D}^{\rightarrow}_{\mathrm{cc}}(F)$. The relevant model of decision trees is the model of non-adaptive parity decision trees. In this model we still want to compute some function f on an unknown input and we still can query XORs of any subsets of input bits, but now all queries should be provided at once (in other words, each query cannot depend on the answers to the previous queries). The complexity of f in this model is denoted by $\mathrm{NADT}^{\oplus}(f)$. It follows from the results of Montanaro, Osbourne [19] and Gopalan et al. [10] that for any total XOR-function $F(x,y) = f(x \oplus y)$ we have $\mathrm{D}^{\rightarrow}_{\mathrm{cc}}(F) = \mathrm{NADT}^{\oplus}(f)$.

These results on the connection between communication complexity and parity decision trees can be viewed as lifting results. This type of results have seen substantial progress in recent years (see [21]). The usual structure of a lifting result is that we start with a function f that is hard in some weak computational model (for example, decision tree type model), compose it with some gadget function g to obtain $f \circ g$ (each variable of f is substituted by a copy of g defined on fresh variables) and show that $f \circ g$ is hard in a stronger computational model (for example, communication complexity type model). The results on XOR-functions can be viewed as lifting results for $g = \mathrm{XOR}$.

The results on the connection between communication complexity of XOR-functions and parity decision trees discussed above are proved only for total functions f for the reason that the proofs heavily rely on Fourier techniques. However, in communication complexity and decision tree complexity it is often relevant to consider a more general case of partial functions, and many lifting theorems apply to this type of functions as well, see e.g. [7, 17, 4, 23]. In particular, there are some lifting results for partial functions for gadgets that are stronger than XOR: Mande et al. [18] proved such a result for one-way case for inner product gadget (inner product is XOR applied to ANDs of pairs of variables) and Loff, Mukhopadhyay [17] proved a result on lifting with equality gadget for general case (note that equality for inputs of length 1 is practically XOR function). In [17] a conjecture is mentioned that for partial XOR-functions $\mathrm{D_{cc}}(F)$ is approximately equal to $\mathrm{DT}^{\oplus}(f)$ as well.

Our results. In this paper we initiate the studies of the connection between communication complexity for the case of partial XOR functions and parity decision trees. It turns out that for one-way case whether they are equal depends on the number of inputs on which the function is undefined: if the number of undefined inputs is small, then the complexity measures are equal and if it is too large, they are not equal.

More specifically, we show that for $t = \mathrm{D}_{\mathrm{cc}}^{\rightarrow}(F)$ the equality $\mathrm{D}_{\mathrm{cc}}^{\rightarrow}(F) = \mathrm{NADT}^{\oplus}(f)$ holds if f is undefined on at most $O\left(\frac{2^{n-t}}{\sqrt{n-t}}\right)$ inputs. We prove a stronger bound on the number of undefined inputs for small values of t. More specifically, for $t = 1$ we show that the equality $\mathrm{D}_{\mathrm{cc}}^{\rightarrow}(F) = \mathrm{NADT}^{\oplus}(f)$ is true for all partial f. For $t = 2$ we show that the equality is true for at most $2^{n-3} - 1$ undefined inputs. On the other end of the spectrum we show that for any partial function if $\mathrm{NADT}^{\oplus}(f) \geq n - 1$, then $\mathrm{D}_{\mathrm{cc}}^{\rightarrow}(F) = \mathrm{NADT}^{\oplus}(f)$.

On the other hand, we provide a family of partial function for which $\mathrm{D}_{\mathrm{cc}}^{\rightarrow}(F) < \mathrm{NADT}^{\oplus}(f)$[1]. More specifically, we show that for any constant $0 < c < 1$ there is a function f with $\mathrm{NADT}^{\oplus}(f) = cn$ and $\mathrm{D}_{\mathrm{cc}}^{\rightarrow}(F) \leq c'n$ for some $c' < c$. The number of undefined inputs for the function is $O\left(\frac{2^{dn}}{\sqrt{n}}\right)$ if $c > 1/2$, 2^{n-1} if $c = 1/2$ and $2^n - O\left(\frac{2^{dn}}{\sqrt{n}}\right)$ if $c < 1/2$, where $0 < d < 1$ is some constant (depending of c).

We provide a function f for which $\mathrm{NADT}^{\oplus}(f) = \sqrt{n \log n}$ and $\mathrm{D}_{\mathrm{cc}}^{\rightarrow}(F) \leq O(\log n)$, the number of undefined inputs for f is $2^n - 2^{\Theta(\sqrt{n} \log^{3/2} n)}$. Thus, we provide an exponential gap between the two measures.

The largest value of $\mathrm{NADT}^{\oplus}$ for which we provide a separation is $n-2$, this complements the result that starting with $\mathrm{NADT}^{\oplus}(f) = n-1$ the measures are equal. The smallest values of measures for which we provide a separation are $\mathrm{D}_{\mathrm{cc}}^{\rightarrow}(F) = 7$ and $\mathrm{NADT}^{\oplus}(f) = 8$.

All our separation results translate to the setting of two-sided communication complexity vs. parity decision trees. In particular, we provide a partial function f with exponential gap between $\mathrm{D}_{\mathrm{cc}}(F)$ and $\mathrm{DT}^{\oplus}(f)$, which refutes the conjecture mentioned in [17].

The techniques behind the results on the connections between communication complexity of XOR-functions and parity decision tree complexity for total functions heavily rely on Fourier analysis. However, it is not clear how to translate this technique to partial functions. To prove our results we instead translate Fourier-based approach of [19, 10] into linear algebraic language. We design a framework to capture the notion of one-sided communication complexity of partial XOR-functions and use this framework to establish both equality of $\mathrm{D}_{\mathrm{cc}}^{\rightarrow}(F)$ and $\mathrm{NADT}^{\oplus}(f)$ for the small number of undefined points and the separation results. Within our framework we prove separation results by a reduction to covering codes.

The rest of the paper is organized as follows. In Section 2 we provide necessary preliminary information and introduce the notations. In Section 3 we introduce our linear-algebraic framework. In Section 4 we prove main results on the equality of complexity measures. In Section 5 we prove separation results. In Section 6 we provide the results for $\mathrm{D}_{\mathrm{cc}}^{\rightarrow}(F) = 1$ and $\mathrm{D}_{\mathrm{cc}}^{\rightarrow}(F) = 2$. Some of the technical proofs are presented in Appendix.

[1] Note that the gap in the other direction is impossible: it is easy to see that $\mathrm{D}_{\mathrm{cc}}^{\rightarrow}(F) \leq \mathrm{NADT}^{\oplus}(f)$ for all f (see Lemma 3 below). Similar inequality (with an extra factor of 2) holds for general communication complexity and parity decision tree complexity.

2 Preliminaries

2.1 Boolean cube

A Boolean cube is a graph on the set $\{0,1\}^n$ of Boolean strings of length n. We connect two vertices with an edge if they differ in a single bit only. The set $\{0,1\}^n$ can also be thought of as the vector space $\mathbb{F}_2^n$, with the bitwise XOR as the group operation. An inner product over this space can be defined as

$$\langle x, y\rangle = \bigoplus_i x_i \wedge y_i.$$

We define $\operatorname{dist}(x,y)$ between $x \in \{0,1\}^n$ and $y \in \{0,1\}^n$ to be the Hamming distance (that is defined as the number of coordinates at which x and y differ). We denote by $V(n,r)$ the size of a Hamming ball in $\{0,1\}^n$ of radius r.

2.2 Isoperimetric inequalities

We will need the vertex isoperimetric inequality for a Boolean cube known as Harper's theorem. To state it we first define Hales order.

Definition 1 (Hales Order)**.** Consider two subsets $x, y \subseteq [m]$. We define $x \prec y$ if $|x| < |y|$ or $|x| = |y|$ and the smallest element of symmetric difference of x and y belongs to x. In other words, there exists an i such that $i \in x, i \notin y$, and i is the smallest element in which x and y differ. Here is an example of Hales order for $m = 4$:

$$\varnothing, 1, 2, 3, 4, 12, 13, 14, 23, 24, 34, 123, 124, 134, 234, 1234.$$

We can induce Hales order on the set $\{0,1\}^m$ by identifying subsets of $[m]$ with their charqcteristic vectors.

Theorem 2 (Harper's theorem [12, Theorem 4.2])**.** *Let $A \subseteq \{0,1\}^m$ be a subset of vertices of m-dimensional Boolean cube and denote $a = |A|$. Define I_a^m to be the set of the first a elements of $\{0,1\}^m$ in Hales order. Then $|\Gamma A| \geq |\Gamma I_a^m|$.*

2.3 Communication Complexity and Decision Trees

Throughout this paper, f denotes a partial function $\{0,1\}^n \to \{0,1,\perp\}$, we let $\operatorname{Dom}(f) = f^{-1}(\{0,1\})$. We define an XOR-function $F : \{0,1\}^n \times \{0,1\}^n \to \{0,1,\perp\}$ as

$$F(x,y) = f(x \oplus y).$$

In communication complexity model two players, Alice and Bob, are computing some fixed function $F\colon \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ on a given input (x,y). However, Alice knows only x and Bob knows only y. The main subject of studies in communication complexity is the amount of communication $\mathrm{D_{cc}}(F)$ needed between Alice and Bob to compute the function. Formal definition of the model can be found in [16].

We will be mostly interested in one-way communication model. This is a substantially restricted setting, in which only Alice is permitted to send bits to Bob. Formally, one-way communication complexity $D_{cc}^{\rightarrow}(F)$ is defined to be the smallest integer t, allowing for a protocol where Alice knowing her input x sends t bits to Bob, which together with Bob's input y enable Bob to calculate the value of F.

The bits communicated by Alice depend only on x, that is Alice's message to Bob is $h(x)$ for some fixed total function $h\colon \{0,1\}^n \to \{0,1\}^t$. Bob computes the output $F(x,y)$ based on $h(x)$ and his input y. That is, Bob outputs $\varphi(h(x),y)$ for some fixed total function $\varphi\colon \{0,1\}^t \times \{0,1\}^n \to \{0,1\}$. If (x,y) is within the domain of F, then the equality $\varphi(h(x),y) = F(x,y)$ must be true.

The notion of parity decision tree complexity is a generalization of the well-known decision tree complexity model. In this model, to evaluate a function f for a given input x the protocol queries the parities of some subsets of the bits in x. The cost of the protocol is the maximum over all inputs number of queries protocol makes and our goal is to minimize it.

We consider the non-adaptive parity decision tree complexity $\mathrm{NADT}^{\oplus}(f)$. This version differs from its adaptive counterpart in that all the queries should be fixed at once. In other words, each next query should not depend on the answers to previous queries. Next we give more formal definition of $\mathrm{NADT}^{\oplus}(f)$.

The protocol of complexity p is defined by n-bit strings $s_1,\dots,s_p$ and a total function $l\colon \{0,1\}^p \to \{0,1\}$. On input x the protocol queries the values of

$$\langle s_1, x\rangle, \dots, \langle s_p, x\rangle$$

and outputs

$$l(\langle s_1, x\rangle, \dots, \langle s_p, x\rangle).$$

The protocol computes partial function f, if for any $x \in \mathrm{Dom}(f)$ we have

$$l(\langle s_1, x\rangle, \dots, \langle s_p, x\rangle) = f(x).$$

Throughout the paper $t, h, \varphi, p, s_1, \dots, s_p, l$ have the same meaning as defined above.

It is easy to see that there is a simple relation between $\mathrm{NADT}^{\oplus}(f)$ and $\mathrm{D}_{\mathrm{cc}}^{\rightarrow}(F)$.

Lemma 3. *For any f we have $\mathrm{D}_{\mathrm{cc}}^{\rightarrow}(F) \le \mathrm{NADT}^{\oplus}(f)$.*

Proof. Alice and Bob can compute $F(x,y)$ by a simple simulation of $\mathrm{NADT}^{\oplus}$ protocol for f. The idea is that they privately calculate the parities of their respective inputs according to $\mathrm{NADT}^{\oplus}$ protocol, then Alice sends the computed values to Bob, who XORs them with his own parities, and then computes the value of F.

More formally, assume that $\mathrm{NADT}^{\oplus}(f) = p$ and the corresponding protocol is given by $s_1,\dots,s_p \in \{0,1\}^n$ and a function l, that is

$$\forall x \in \mathrm{Dom}(f), f(x) = l(\langle s_1, x\rangle, \dots, \langle s_p, x\rangle).$$

For $i \in [p]$, we let

$$h_i(x) := \langle s_i, x\rangle.$$

For the communication protocol of complexity p we let

$$h(x) = (h_1(x), \ldots, h_p(x)),$$

$$\varphi(a, y) := l(a_1 \oplus \langle s_1, y\rangle, \ldots, a_p \oplus \langle s_p, y\rangle).$$

Then for any (x, y) such that $x \oplus y \in \mathrm{Dom}(f)$ we have

$$\begin{aligned}\varphi(h(x), y) = l(h_1(x) \oplus \langle s_1, y\rangle, \ldots, h_p(x) \oplus \langle s_p, y\rangle) = \\ l(\langle s_1, x\rangle \oplus \langle s_1, y\rangle, \ldots, \langle s_p, x\rangle \oplus \langle s_p, y\rangle) = \\ l(\langle s_1, x \oplus y\rangle, \ldots, \langle s_p, x \oplus y\rangle) = f(x \oplus y) = F(x, y).\end{aligned}$$

We constructed a p-bit communication protocol for F, and thus

$$\mathrm{D}^{\rightarrow}_{\mathrm{cc}}(F) \leq p = \mathrm{NADT}^{\oplus}(f).$$

□

In this paper we are mainly interested in whether the inequality in the opposite direction is true.

2.4 Covering Codes

Definition 4. A subset $\mathcal{C} \subseteq \{0,1\}^n$ is a (n, K, R) covering code if $|\mathcal{C}| \leq K$ and for any $x \in \{0,1\}^n$ there is $c \in \mathcal{C}$ such that $\mathrm{dist}(x, c) \leq R$. In other words, all point in $\{0,1\}^n$ are covered by balls of radius R with centers in $\mathcal{C}$.

The following general bounds on K are known for covering codes.

Theorem 5 ([6, Theorem 12.1.2])**.** *For any (n, K, R) covering code we have*

$$\log K \geq n - \log V(n, R).$$

For any n and any $R \leq n$ there is a (n, K, R) covering code with

$$\log K \leq n - \log V(n, R) + \log n.$$

We will use the following well known fact.

Theorem 6 ([6, Section 2.6])**.** *If $n = 2^m - 1$ for some m, then Boolean cube $\{0,1\}^n$ can be splitted into disjoint balls of radius 1.*

This construction is known as a Hamming error correcting code. Note that it is a $(n = 2^m - 1, \frac{2^n}{n+1}, 1)$ covering code.

Definition 7. For two covering codes $\mathcal{C}_1$ and $\mathcal{C}_2$ their direct sum is

$$\mathcal{C}_1 \oplus \mathcal{C}_2 = \{(c_1, c_2) \mid c_1 \in \mathcal{C}_1, c_2 \in \mathcal{C}_2\}.$$

Lemma 8 ([6, Theorem 12.1.2])**.** *If $\mathcal{C}_1$ is a (n_1, K_1, R_1) covering code and $\mathcal{C}_2$ is a (n_2, K_2, R_2) covering code, then $\mathcal{C}_1 \oplus \mathcal{C}_2$ has parameters $(n_1 + n_2, K_1K_2, R_1 + R_2)$.*

We need the following bounds on the sizes of Hamming balls (see, e.g. [14, Appendix A]).

Lemma 9. *For any n and $k \le n$ we have*

$$\left(\frac{n}{k}\right)^k \le V(n,k) \le \left(\frac{en}{k}\right)^k.$$

Lemma 10. *For any constant $0 < c < 1$ we have*

$$\binom{n}{cn} = O\left(\frac{1}{\sqrt{n}} 2^{H(c)n}\right).$$

For any constant $0 < c < 1/2$ we have

$$V(n, cn) = O\left(\frac{1}{\sqrt{n}} 2^{H(c)n}\right),$$

where H is an entropy function.

Lemma 11 ([25, Section 5.4])**.**

$$V\left(n, \frac{n}{2} - \Theta(\sqrt{n \log n})\right) = \frac{2^n}{poly(n)}.$$

For entropy function $H(x)$ we will use the following simple fact.

Lemma 12. *For any constant $c \in (0,1)$ and for any $\alpha_n \xrightarrow[n\to\infty]{} 0$ we have*

$$H(c + \alpha_n) = H(c) + O(\alpha_n),$$

where the constant in O-notation might depend on c, but not on n.

This is true since the derivative of H is upper bounded by a constant in any small enough neighborhood of c.

3 Linear-algebraic framework

3.1 Description of $\mathrm{D}^{\to}_{\mathrm{cc}}(F)$ in terms of a graph

Recall that in one-way communication protocol of complexity t for $F(x,y) = f(x \oplus y)$ Alice on input $x \in \{0,1\}^n$ first sends to Bob $h(x)$ for some fixed $h\colon \{0,1\}^n \to \{0,1\}^t$. After that Bob computes the output $\varphi(h(x), y)$, where $y \in \{0,1\}^n$ is Bob's input and $\varphi\colon \{0,1\}^t \times \{0,1\}^n \to \{0,1\}$.

Let's consider the partition $\mathcal{H} = \{H_a \mid a \in \{0,1\}^t\}$, where for any $a \in \{0,1\}^t$

$$H_a = h^{-1}(a).$$

We refer to $\mathcal{H}$ as *h-induced partition.* A class H_a of this partition is the set of inputs for which Alice sends Bob the same message.

Consider any two arbitrary inputs $x, y \in \{0,1\}^n$ and consider vector $\Delta = x \oplus y$ as a *shift* between x and y in the sense that $y = x \oplus \Delta$ (and vise versa). That is, y is obtained from x by a shift by Δ. We say that $\Delta \in \{0,1\}^n$ is a *good shift* if there is a pair $x, y \in \{0,1\}^n$ such that $x \oplus y = \Delta$ and $h(x) = h(y)$, or equivalently, such that x and y belong to the same class of $\mathcal{H}$. Note that f does not necessarily need to be defined on inputs x and y. However, it turns out that on the domain of f the value of f is invariant under good shifts.

Lemma 13. *Assume that Δ is a good shift. Consider any $v, u \in \mathrm{Dom}(f)$ such that $v \oplus u = \Delta$. Then, $f(v) = f(u)$.*

Proof. Since Δ is good, there are x and y such that $h(x) = h(y)$ and $x \oplus y = \Delta$. Then

$$f(v) = \varphi(h(x), x \oplus v) = \varphi(h(y), x \oplus v) = f(v \oplus x \oplus y) = f(v \oplus \Delta) = f(u).$$

□

This leads us to the following notion.

Definition 14. Consider a graph with vertices $\{0,1\}^n$ and edges drawn between vertices x and y if $x \oplus y$ is a good shift. We call this graph a *total h-induced graph.* Now remove vertices where the function f is undefined. We refer to the resulting graph as a *partial h-induced graph.*

There is an alternative way of thinking about total h-induced graph. Consider a graph in which we connect two vertices if the value of h on these vertices is the same. Clearly, it is a subgraph of the total h-induced graph. Now consider a shift of this graph, that is, a graph in which we shifted all vertices by some fixed vector. This graph is also a subset of the total h-induced graph. By considering all possible shifts and taking the union of all graphs we will get the total h-induced graph.

By transitivity, if h, φ form a valid communication protocol then f assigns identical values to each connected component in partial h-induced graph. The converse is also true.

Theorem 15. *For a function $h : \{0,1\}^n \to \{0,1\}^t$ there is a function $\varphi : \{0,1\}^t \times \{0,1\}^n \to \{0,1\}$ such that h, φ form a valid communication protocol if and only if f assigns the same value to each connected component in the partial h-induced graph.*

Proof. As discussed above, if h, φ form a valid communication protocol, then f assigns the same value to each connected component of the partial h-induced graph.

It remains to prove the converse statement. We assume that f assigns the same value to each connected component and we need to show that there is such φ that

$$\forall (x, y) \in \mathrm{Dom}(F), \quad F(x, y) = \varphi(h(x), y).$$

We define φ as follows. For each α and y, consider x' such that $h(x') = \alpha$ and $(x', y) \in \mathrm{Dom}(F)$. If there is no such x' we define $\varphi(\alpha, y)$ arbitrarily. If there is such an x', let

$$\varphi(\alpha, y) := F(x', y).$$

Now we show that the resulting protocol computes $F(x, y)$ correctly for any (x, y). Consider arbitrary $(x, y) \in \mathrm{Dom}(F)$. Consider x' chosen for $\alpha = h(x)$ and y (it exists, since clearly x itself satisfies all the necessary conditions).

Thus, we have

$$\varphi(h(x), y) = F(x', y).$$

It remains to prove that

$$F(x', y) = F(x, y)$$

or equivalently,

$$f(x' \oplus y) = f(x \oplus y).$$

For XOR of these two inputs of f we have

$$(x' \oplus y) \oplus (x \oplus y) = x' \oplus x.$$

Since $h(x) = h(x')$, we have that $x' \oplus x$ is a good shift. And since

$$(x, y), (x', y) \in \mathrm{Dom}(F),$$

we have

$$x \oplus y, x' \oplus y \in \mathrm{Dom}(f).$$

We have that vertices $x \oplus y$ and $x' \oplus y$ are connected in the partial h-induced graph and by Lemma 13 f assigns the same value to them. Hence, the function φ, together with h, forms a communication protocol for F. □

3.2 Description of $\mathrm{NADT}^{\oplus}(f)$ in terms of cosets

We consider the vertices of the Boolean cube as a vector space $\mathbb{F}_2^n$. We show that a $\mathrm{NADT}^{\oplus}$ protocol corresponds to a linear subspace such that f is constant on each of its cosets.

Theorem 16. *A p-bit $\mathrm{NADT}^{\oplus}$ protocol exists if and only if there exists an $n-p$ dimensional subspace such that for each coset of that subspace, f assigns the same value to all inputs of the coset where f is defined.*

Proof. Suppose $s_1, \ldots, s_p, l$ form a valid $\mathrm{NADT}^{\oplus}$ protocol for f. We construct a matrix S with rows $s_1, \ldots, s_p$. If some of the rows are linearly dependent, we add rows arbitrarily to make the rank of S equal to p. When S is multiplied on the right by some vector x, we obtain all inner products of x with vectors $s_1, \ldots, s_p$ (and possibly other bits if we added rows).

Consider the vector subspace $\{x|Sx = 0\}$. This is an $n - p$ dimensional space. For all points in the same coset of this subspace, the values of the inner products $\langle s_1, x\rangle, \ldots, \langle s_p, x\rangle$ are the same, so is the value of $l(\langle s_1, x\rangle, \ldots, \langle s_p, x\rangle)$. For all points where f is defined and lying in the same coset, the value of f must be equal to the value of l and thus the same for all points in the coset.

In the reverse direction, let $\langle e_1, \ldots, e_{n-p}\rangle$ be an $n - p$ dimensional subspace such that for each its coset f is constant on all points on which it is defined. We can represent this subspace in the form $\{x|Sx = 0\}$ for some matrix S of size $p \times n$.

Vectors x and y are in the same coset of $\langle e_1, \ldots, e_{n-p}\rangle$ iff $Sx = Sy$. Thus, to compute $f(x)$ it is enough to compute the inner product of x with the rows of S. □

Corollary 17. *If there exists an $n-p$ dimensional subspace L, such that any subgraph G of the partial h-induced graph, such that G is induced by a coset of L, is connected then* $\mathrm{NADT}^{\oplus}(f) \leq p$.

Proof. By Theorem 15 f is constant on each coset. By Theorem 16 it follows that $\mathrm{NADT}^{\oplus}(f) \leq p$. □

4 Equality between $\mathrm{D}^{\rightarrow}_{\mathrm{cc}}(F)$ and $\mathrm{NADT}^{\oplus}(f)$

In this section we will show that if $\mathrm{D}^{\rightarrow}_{\mathrm{cc}}(F) = t$ and the number of undefined inputs is small, then $\mathrm{NADT}^{\oplus}(f) = t$ as well. More specifically, we prove the following theorem.

Theorem 18. *If for the function f we have $\mathrm{D}^{\rightarrow}_{\mathrm{cc}}(F) = t$ and f is undefined on less than $\binom{n-t+1}{\lfloor\frac{n-t}{2}\rfloor-1}$ inputs, then* $\mathrm{NADT}^{\oplus}(f) = t$.

By Lemma 10 we have that $\binom{n-t+1}{\lfloor\frac{n-t+1}{2}\rfloor} = O(\frac{2^{n-t}}{\sqrt{n-t}})$ and since $\lfloor\frac{n-t}{2}\rfloor - 1$ differs from $\lfloor\frac{n-t+1}{2}\rfloor$ by only a constant, it is easy to see that the same estimate applies to $\binom{n-t+1}{\lfloor\frac{n-t}{2}\rfloor-1}$ as well. Thus, the number of undefined inputs is $O(\frac{2^{n-t}}{\sqrt{n-t}})$.

The rest of the section is devoted to the proof of Theorem 18. The idea of the proof is as follows. Consider h-induced partition $\mathcal{H}$ corresponding to the communication protocol of complexity t. We show that either the partition $\mathcal{H}$ corresponds to the cosets of an $n-t$ dimensional subspace, which allows us to construct an $\mathrm{NADT}^{\oplus}$ protocol, or there exist *many* good shifts. The structure of these good shifts imposes restrictions on f that again allow us to construct an $\mathrm{NADT}^{\oplus}$ protocol.

We start with a simple case.

Lemma 19. *If the partition $\mathcal{H}$ corresponds to cosets of an $n-t$ dimensional subspace L, then* $\mathrm{NADT}^{\oplus}(f) \leq t$.

Proof. Since the partition $\mathcal{H}$ corresponds to the cosets of L, we have that for any inputs x and y, if $h(x) = h(y)$, then $x \oplus y \in L$ and vice versa. In other words, all good shifts are in L and any shift in L is good. Thus, connected components of the total h-induced graph are cosets of L and are fully connected. By Corollary 17 we have that $\mathrm{NADT}^{\oplus}(f) \leq t$. □

The structure of the proof for the other case is the following. We show that the total h-induced graph is structured into connected components, each of which is a coset of a k-dimensional subspace for $k \geq n-t$. We show that there is a bijective graph homomorphism of the k-dimensional Boolean cube onto each component. Furthermore, each vertex in the total h-induced graph has a degree of at least $\frac{2^n}{2^t} - 1$. We show that if we remove fewer than $\binom{n-t+1}{\lfloor\frac{n-t}{2}\rfloor-1}$ vertices, each coset still contains one connected component. By the way of contradiction, suppose this is not the case and some coset contains more than one connected component. We consider the smallest of these components, denote the set of its nodes by S. We show that the number of neighboring vertices of S in the total h-induced graph (excluding S itself) is not less than $\binom{n-t+1}{\lfloor\frac{n-t}{2}\rfloor-1}$. This implies that after removing the undefined inputs of f S cannot not be separated from other nodes in the coset. To show this we treat separately cases of large and small $|S|$. For small $|S|$ we use the fact that vertices have high degree. For large $|S|$ we use the vertex-isoperimetric inequality for the Boolean cube.

Lemma 20. *Given a partition $\mathcal{H}$ whose classes do not correspond to cosets of an $n-t$-dimensional subspace, let D be the set of good shifts. Then D contains a minimum of $n-t+1$ linearly independent vectors.*

Proof. Suppose there are at most $n-t$ linearly independent good shifts $e_1, \dots, e_{n-t}$. Consider a linear subspace spanned over these shifts and add some vectors to it to make it exactly $n-t$ dimensional if needed. Notate resulting subspace L. As classes of $\mathcal{H}$ do not correspond to the cosets of L and there are 2^{n-t} of both classes and cosets there exist two elements belonging to the same class and different cosets. Their XOR is a good shift linearly independent with $e_1, \dots, e_{n-t}$. We got a contradiction implying the lemma. □

Lemma 21. *Consider D as the set of all good shifts and $\langle e_1, \dots, e_k \rangle$ as the largest linearly independent subset of D. Then the total h-induced graph has the following properties.*

- *Cosets of the subspace $\langle e_1, \dots, e_k \rangle$ are connected components of the total h-induced graph.*
- *There is a bijective graph homomorphism of k-dimensional Boolean cube into each coset.*

Proof. It is easy to see that all vertices in any coset are connected to each other. Let's show that no edges exist between vertices of different cosets. Assume by contradiction that there is an edge between vertices v and u from different cosets. Note that $u \oplus v \notin \langle e_1, \dots, e_k \rangle$. Thus, vectors $e_1, \dots, e_k, u \oplus v$ form a linearly independent system of size $k+1$, which is a contradiction.

Now, let's construct a homomorphism q from the Boolean cube $\{0,1\}^k$ into the coset $v + \langle e_1, \dots, e_k \rangle$ for an arbitrary vertex v. Consider a matrix B that has vectors $e_1, \dots, e_k$ as its columns and let $q(x) = v \oplus Bx$. The image of q is within the coset $v + \langle e_1, \dots, e_k \rangle$, as columns of B belong to the subspace $\langle e_1, \dots, e_k \rangle$. The mapping is bijective on $v + \langle e_1, \dots, e_k \rangle$, as B's columns are linearly independent. Finally, consider a pair of vertices x, y adjacent in a Boolean cube. Since the vertices are adjacent, they only differ in a single bit i. Thus,

$$q(x) \oplus q(y) = (v \oplus Bx) \oplus (v \oplus By) = B(x \oplus y) = e_i.$$

Since $e_i \in D$, an edge exists between $q(x)$ and $q(y)$, implying that q is a graph homomorphism. □

Lemma 22. *In the total h-induced graph, the degree of any vertex is not less than $\frac{2^n}{2^t} - 1$.*

Proof. Let's consider the largest class in the h-induced partition $\mathcal{H}$. Since the number of classes is at most 2^t, the largest class contains at least $\frac{2^n}{2^t}$ elements. Fix an element of the class and compute its XOR with all elements in the same class $\mathcal{H}$. We have $\frac{2^n}{2^t}$ XORs in total, $\frac{2^n}{2^t} - 1$ of which are non-zero. Since each XOR is computed between elements in the same class, these XORs are good shifts. For all vertices in the h-induced graph for each good shift we draw an edge from the vertex corresponding to this shift. Therefore, the degree of any vertex is at least $\frac{2^n}{2^t} - 1$. □

Lemma 23. *If A is a subset of k-dimensional Boolean cube satisfying $V\left(m, \lfloor\frac{m-1}{2}\rfloor - 2\right) \leq |A| \leq 2^{k-1}$ then $|\Gamma' A| \geq \binom{m}{\lfloor\frac{m-1}{2}\rfloor - 1}$.*

The proof of the lemma is moved to Appendix A. Finally, we are ready to prove Theorem 18.

Proof of Theorem 18. By Lemma 20, the partition $\mathcal{H}$ either corresponds to cosets of an $n-t$ dimensional subspace (and then by Lemma 19 we have $\mathrm{NADT}^{\oplus}(f) \leq t$), or the set of good shifts D contains at least $n-t+1$ linearly independent vectors. Let $\langle e_1, \ldots, e_k\rangle$, where $k \geq n-t+1$, be the largest subset of linearly independent vectors in D. Consider the cosets of the subspace $\langle e_1, \ldots, e_k\rangle$. We will show that if we remove fewer than $\binom{n-t+1}{\lfloor\frac{n-t}{2}\rfloor - 1}$ vertices from the total h-induced graph, each coset will contain no more than one connected component. Assume by contradiction that after removing the vertices, some coset splits into several connected components. Let A be the smallest of these components. If there are at most $V(n-t+1, \lfloor\frac{n-t}{2}\rfloor - 2) - 1$ vertices in A, consider a vertex a in A. Given the degree of a is at least $2^{n-t} - 1$, a has at least

$$2^{n-t} - V\left(n-t+1, \left\lfloor\frac{n-t}{2}\right\rfloor - 2\right) \geq V\left(n-t+1, \left\lfloor\frac{n-t}{2}\right\rfloor\right) - V\left(n-t+1, \left\lfloor\frac{n-t}{2}\right\rfloor - 2\right)$$
$$\geq \binom{n-t+1}{\lfloor\frac{n-t}{2}\rfloor - 1}$$

neighbors outside A.

On the other hand, suppose A has at least $V(n-t+1, \lfloor\frac{n-t}{2}\rfloor - 2)$ vertices. Since A is the smallest connected component in its coset it also follows that A has no more than 2^{k-1} vertices. By Lemma 23 we have $|\Gamma' A| \geq \binom{n-t+1}{\lfloor\frac{n-t}{2}\rfloor - 1}$, which is more than the number of removed vertices, a contradiction. Thus, cosets cannot be split into several components and by Corollary 17 we have $\mathrm{NADT}^{\oplus}(f) \leq n-k \leq t-1$, which is a contradiction. □

4.1 Large Values of $\mathrm{D}^{\rightarrow}_{\mathrm{cc}}(F)$ and $\mathrm{NADT}^{\oplus}(f)$

On the other end of the spectrum, we show that if $\mathrm{NADT}^{\oplus}(f)$ is really large, then it is equal for all partial functions.

Theorem 24. *For any partial function $f\colon \{0,1\}^n \to \{0,1,\perp\}$, if $\mathrm{NADT}^{\oplus}(f) \geq n-1$, then $\mathrm{D}^{\rightarrow}_{\mathrm{cc}}(F) = \mathrm{NADT}^{\oplus}(f)$.*

Proof. First consider the case $\mathrm{NADT}^{\oplus}(f) = n$ and assume that $\mathrm{D}^{\rightarrow}_{\mathrm{cc}}(F) \leq n-1$. Consider the corresponding function h. One of its equivalence classes H is of size at least 2, denote two of its elements by u and v. We have that $\Delta = u \oplus v$ is a good shift. Thus, for any x if $f(x)$ and $f(x \oplus \Delta)$ are defined, then $f(x) = f(x \oplus \Delta)$. But this exactly means that there is a 1-dimensional space such that f is constant on each of its cosets. Thus, $\mathrm{NADT}^{\oplus}(f) \leq n-1$, which is a contradiction.

Now consider the case $\mathrm{NADT}^{\oplus}(f) = n-1$ and again assume that $\mathrm{D}^{\rightarrow}_{\mathrm{cc}}(F) \leq n-2$. Consider the corresponding function h. Now one of its equivalence classes H is of size at least 4. Consider any three points u, v, w in this class. Then the vectors $u \oplus v$, $u \oplus w$ and

$v \oplus w$ are good shifts. Note that they together with 0-vector form a 2-dimensional linear subspace of good shifts. As a result, f is a constant on every coset of this subspace and $\mathrm{NADT}^{\oplus}(f) \leq n-2$, which is a contradiction. □

5 Separations between $\mathrm{D}^{\rightarrow}_{\mathrm{cc}}(F)$ and $\mathrm{NADT}^{\oplus}(f)$

In this section we show that if the number of undefined inputs is large, there is a gap between $\mathrm{D}^{\rightarrow}_{\mathrm{cc}}(F)$ and $\mathrm{NADT}^{\oplus}(f)$. That is, we aim to come up with a function f such that $\mathrm{D}^{\rightarrow}_{\mathrm{cc}}(F)$ is small and $\mathrm{NADT}^{\oplus}(f)$ is large.

The key idea in our construction is that in h-induced graph for the intended communication protocol the edges connect only vertices with small Hamming distance between them. Then, if the function f has 0-inputs and 1-inputs far away from each other, they are not connected and h corresponds to a valid protocol. We will ensure that at the same time f has large $\mathrm{NADT}^{\oplus}$ complexity.

We start with the construction of the functions, then investigate their $\mathrm{NADT}^{\oplus}$ complexity and then prove upper bounds on $\mathrm{D}^{\rightarrow}_{\mathrm{cc}}$ complexity of the corresponding XOR functions. The latter part is through the reduction to covering codes.

Definition 25. For a parameter k define $f_k\colon \{0,1\}^n \to \{0,1,\perp\}$ in the following way.

$$f_k(x) = \begin{cases} 0 & \text{for } |x| \leq k, \\ \perp & \text{for } k+1 \leq |x| \leq n-1, \\ 1 & \text{for } |x| = n. \end{cases}$$

We denote the corresponding XOR function by F_k.

Note, that the number of undefined inputs in f_k is $V(n, n-k-1) - 1$.

It turns out that f_k has reasonably large $\mathrm{NADT}^{\oplus}$ and $\mathrm{DT}^{\oplus}$ complexities.

Theorem 26. $\mathrm{NADT}^{\oplus}(f_k) = \mathrm{DT}^{\oplus}(f_k) = k+1$.

Proof. Since $\mathrm{DT}^{\oplus}(f) \leq \mathrm{NADT}^{\oplus}(f)$ for any f, it is enough to prove that $\mathrm{NADT}^{\oplus}(f_k) \leq k+1$ and $\mathrm{DT}^{\oplus}(f_k) \geq k+1$

For the upper bound, observe that it is enough to query variables $x_1, \ldots, x_{k+1}$. If all of them are equal to 1, we output 1, otherwise we output 0. It is easy to see that this protocol computes f_k correctly.

For the lower bound suppose, for the sake of contradiction, that an adaptive parity decision tree exists that can compute the function f with k or fewer queries. Consider the branch corresponding to the input $e = (1, \ldots, 1)$. Let's assume that the decision tree queried the parities $\langle s_i, x\rangle$ for $s_1, \ldots, s_k$. The answers to the queries are equal to $\langle s_1, e\rangle, \ldots, \langle s_k, e\rangle$. Consider a matrix $B \subseteq \mathbb{F}^{k\times n}$ consisting of rows $s_1, \ldots, s_k$.

Denote $a = Be$. In particular, we have that a lies in the subspace generated by columns of B. Since the rank of B is at most k (the matrix has k rows), there is a subset of at most k columns generating this subspace. In particular, there is $x \in \{0,1\}^n$ with $|x| \leq k$, such that $a = Bx$. That is, $Be = Bx$ and the protocol behaves the same way on e and x, which is a contradiction, since $f_k(e) = 1$ and $f_k(x) = 0$. □

Remark 27. Since f_k has large (adaptive) parity decision tree complexity and for any F we have $\mathrm{D}^{\rightarrow}_{\mathrm{cc}}(F) \geq \mathrm{D}_{\mathrm{cc}}(F)$, all separations provided by functions f_k translate into the same separations between $\mathrm{DT}^{\oplus}$ and D_{cc}.

Next, we proceed to the upper bound on the $\mathrm{D}^{\rightarrow}_{\mathrm{cc}}(F_k)$.

Theorem 28. *Suppose for some n, k and t there is a $(n, 2^t, R)$ covering code $\mathcal{C}$ for $R = \left\lfloor \frac{n-k-1}{2} \right\rfloor$. Then, $\mathrm{D}^{\rightarrow}_{\mathrm{cc}}(F_k) \leq t$.*

Proof. Split the points of $\{0,1\}^n$ into balls with radius R with centers in the points of $\mathcal{C}$ (if some point belongs to several balls, attribute it to one of them arbitrarily). This results in a partition of the cube into 2^t subsets with the diameter of each subset at most $n-k-1$. Consider a function h with this $\mathcal{H}$-partition.

Edges in h-induced graph connects only vertices at distance at most $n-k-1$. Since, distance between 0-inputs and 1-inputs of f_k is at least $n-k$, 0-inputs and 1-inputs belong to disjoint connected components. By Theorem 15 we have $\mathrm{D}^{\rightarrow}_{\mathrm{cc}}(F_k) \leq t$. □

Theorem 29. *For any n and k we have*

$$\mathrm{D}^{\rightarrow}_{\mathrm{cc}}(F_k) \leq n - \log V(n, R) + \log n$$

for $R = \left\lfloor \frac{n-k-1}{2} \right\rfloor$.

Proof. By Theorem 5 there exists a $(n, 2^t, R)$ covering code for

$$\log 2^t = t \leq n - \log V(n, R) + \log n.$$

The corollary follows from Theorem 28. □

From this we can get a separation for a wide range of parameters.

Corollary 30. *Suppose $k = cn$ for some constant $0 < c < 1$. Then $\mathrm{NADT}^{\oplus}(f_k) = cn + 1$ and*

$$\mathrm{D}^{\rightarrow}_{\mathrm{cc}}(F_k) \leq \left(1 - H\left(\frac{1-c}{2}\right)\right) n - O(\log n).$$

In particular, $\mathrm{D}^{\rightarrow}_{\mathrm{cc}}(F_k) < \mathrm{NADT}^{\oplus}(f_k)$. The number of undefined inputs for f_k is $2^n - O(\frac{2^{H(c)n}}{\sqrt{n}})$ if $c < 1/2$, $(1+o(1))2^{n-1}$ if $c = 1/2$ and $O(\frac{2^{H(1-c)n}}{\sqrt{n}})$ if $c > 1/2$.

Proof. The equality for $\mathrm{NADT}^{\oplus}$ is proved in Theorem 26.

For communication complexity bound we apply Theorem 29. We have $R = \left\lfloor \frac{(1-c)n-1}{2} \right\rfloor = \frac{(1-c)n}{2} + O(1)$ and by Lemmas 10 and 12 we have

$$\log V(n, R) = H\left(\frac{1-c}{2}\right) n - O(\log n).$$

By Theorem 29 we have

$$\mathrm{D}^{\rightarrow}_{\mathrm{cc}}(F_k) \leq n - \log V(n, R) + \log n = \left(1 - H\left(\frac{1-c}{2}\right)\right) n + O(\log n).$$

To show that $\mathrm{D}^{\rightarrow}_{\mathrm{cc}}(F_k) < \mathrm{NADT}^{\oplus}(f_k)$ we need to compare $k = cn$ with the bound on communication complexity. It is easy to see that

$$1 - H\left(\frac{1-c}{2}\right) < c$$

for all $0 < c < 1$ (the left hand-side and the right hand-side are equal for $c = 0$ and $c = 1$ and the left hand-side is concave in c).

The bounds on the number of undefined inputs follow easily from Lemma 10. □

The largest gap we can get is the following.

Corollary 31. *For $k = \Theta(\sqrt{n \log n})$ we have that $\mathrm{NADT}^{\oplus}(f_k) = \Theta(\sqrt{n \log n})$ and $\mathrm{D}^{\rightarrow}_{\mathrm{cc}}(F_k) = O(\log n)$. The number of undefined inputs for f_k is $2^n - 2^{\Theta(\sqrt{n} \log^{3/2} n)}$.*

Proof. For $k = \Theta(\sqrt{n \log n})$ we have $R = \frac{n}{2} - \Theta(\sqrt{n \log n})$ in Theorem 29. By Lemma 11 we have $V(n,R) = \Theta\left(\frac{2^n}{\mathrm{poly}(n)}\right)$ and as a result $\mathrm{D}^{\rightarrow}_{\mathrm{cc}}(F_k) = O(\log n)$.

For the number of undefined inputs, we apply Lemma 9:

$$\left(\frac{n}{k}\right)^k \leq V(n,k) \leq \left(\frac{en}{k}\right)^k.$$

For $k = \Theta(\sqrt{n \log n})$ it is easy to see that both sides are $2^{\Theta(\sqrt{n} \log^{3/2} n)}$. From this the estimate on the number of undefined inputs follows. □

The largest value of $\mathrm{NADT}^{\oplus}$ for which we get separation is $n-2$.

Theorem 32. *$\mathrm{D}^{\rightarrow}_{\mathrm{cc}}(F_{n-3}) \leq n - \Theta(\log n)$, whereas $\mathrm{NADT}^{\oplus}(f_{n-3}) = n-2$. The number of undefined inputs for f_{n-3} is $\frac{n(n+1)}{2}$.*

Proof. We have already proved equality for $\mathrm{NADT}^{\oplus}(f_{n-3})$ and it remains to bound $\mathrm{D}^{\rightarrow}_{\mathrm{cc}}(F_{n-3})$.

For this we use Theorem 28. Note that in our case $R = \lfloor \frac{n-k-1}{2} \rfloor = 1$.

If $n = 2^m - 1$ for some integer m, then we can just use Theorem 6. Each ball of radius 1 is of size $n+1$ and thus in total we have $2^n/(n+1)$ balls. As a result,

$$\mathrm{D}^{\rightarrow}_{\mathrm{cc}}(F_{n-3}) \leq \log \frac{2^n}{n+1} = n - \log(n+1).$$

For general n consider maximal integer m such that $2^m - 1 \leq n$. Denote $n_1 = 2^m - 1$ and $n_2 = n - n_1$. Consider Hamming code $\mathcal{C}_1$ on $\{0,1\}^{n_1}$ and consider the code $\mathcal{C}_2 = \{0,1\}^{n_2}$. The latter code has parameters $(n_2, 2^{n_2}, 0)$. By Lemma 8 we have that $\mathcal{C}_1 \oplus \mathcal{C}_2$ has parameters $(n, \frac{2^{n_1}}{n_1+1} \cdot 2^{n_2}, 1)$. Since n_1 is at least half of n, we have

$$\mathrm{D}^{\rightarrow}_{\mathrm{cc}}(F_{n-3}) \leq \log\left(\frac{2^{n_1}}{n_1+1} \cdot 2^{n_2}\right) = n - \Theta(\log n).$$

The undefined inputs of f_{n-3} are just inputs $x \in \{0,1\}^n$ with weight $n-1$ and $n-2$. It is easy to see that there are $\frac{n(n+1)}{2}$ of them. □

The smallest value of $\mathrm{D}_{\mathrm{cc}}^{\rightarrow}$ for which we get a separation is 7.

Theorem 33. *For any $n \geq 32$ we have $\mathrm{D}_{\mathrm{cc}}^{\rightarrow}(F_7) \leq 7$, whereas $\mathrm{NADT}^{\oplus}(f_7) = 8$.*

Proof. Again, we already found $\mathrm{NADT}^{\oplus}(f_7)$.

For the bound on $\mathrm{D}_{\mathrm{cc}}^{\rightarrow}$ we start with Reed-Muller code $\mathcal{RM}(1,5)$ [6, Chapter 9]. This code has parameters $(2^5, 2^6, 12)$ (as a covering code), that is, it has 32 input bits, the number of covering balls is 2^6 and their radius is $R = 12$. In terms of Theorem 28 we have $R = \frac{32-7-1}{2}$ and thus the code gives us the protocol for F_7 of size $\log 2^6 = 6$ on $n = 32$ inputs (that is, for the particular case of $n = 32$ we have an even better upper bound on communication complexity).

For general $n \geq 32$ denote $n_1 = 32$ and $n_2 = n - n_1$. Let $\mathcal{C}_1$ be Reed-Muller code introduced above and $\mathcal{C}_2$ consist of two vectors: all zeros and all ones. The code $\mathcal{C}_2$ has parameters $(n_2, 2, \lfloor \frac{n_2}{2} \rfloor)$. Then $\mathcal{C}_1 \oplus \mathcal{C}_2$ has parameters $(n, 2^7, \lfloor \frac{n_2}{2} \rfloor + 12)$. Note that its radius R can be bounded as

$$R = \left\lfloor \frac{n_2}{2} \right\rfloor + 12 \leq \frac{n_2}{2} + 12 = \frac{n}{2} + 12 - \frac{32}{2} = \frac{n-7-1}{2}.$$

Thus, the code gives a protocol for F_7 of size 7. □

6 The Case of Small Communication Complexity

6.1 Case $\mathrm{D}_{\mathrm{cc}}^{\rightarrow}(F) = 1$

A function h is called *balanced* if all classes in the h-induced partition are of equal size. We say that h is *balanced on a subset* when its restriction to the inputs in this subset is balanced. We analyze two distinct scenarios separately: when h is balanced and when it is not.

For the scenario where h is unbalanced, we will demonstrate that all shifts are good, leading to the conclusion that f is a constant function. Conversely, when h is balanced, we identify a specific $n-1$-dimensional subspace on which h is unbalanced. We then show that every shift in this subspace is good. This observation gives us that the function value of f depends solely on whether x belongs to this identified subspace and this can be checked with a single parity query.

Lemma 34. *Assume F satisfies $\mathrm{D}_{\mathrm{cc}}^{\rightarrow}(F) = 1$. If h is unbalanced, then every shift is good.*

Proof. Consider arbitrary shift Δ. Consider the cosets corresponding to the subspace $\langle \Delta \rangle$. The h-induced partition consists of two classes, since they are not equal, one class contains more than 2^{n-1} elements. Applying the Pigeonhole principle we get that some coset of the subspace $\langle \Delta \rangle$ contains two elements with the same h value. Given that a coset has only two points and those differ by shift Δ, we conclude that Δ is indeed a good shift. □

Lemma 35. *Assume F satisfies $\mathrm{D}_{\mathrm{cc}}^{\rightarrow}(F) = 1$. If h is unbalanced on a given subspace, then every shift in this subspace is good.*

Proof. The proof is completely analogous to the proof of Lemma 34. Indeed, since h is unbalanced on the subspace, for any shift Δ in the subspace there are x and y such that $h(x) = h(y)$ and $x \oplus y = \Delta$. Thus, Δ is a good shift.

$$h'(x) = h(Bx).$$

□

Lemma 36. *For a balanced function h, there is an $n-1$-dimensional subspace over which h is unbalanced.*

Proof. The proof is based on Fourier analysis. For the completeness of the proof, we provide basic definitions in Appendix B.

Consider Fourier decomposition of h. Since h is balanced and thus not constant, there must be a non-zero coefficient $\hat{h}(S)$ in its Fourier decomposition associated with a non-empty subset S. We show that h is unbalanced on the $n-1$-dimensional linear subspace $X = \{x|\chi_S(x) = 1\}$. Assume, for the sake of contradiction, that h is balanced on X. The Fourier coefficient $\hat{h}(S)$ can be computed as follows:

$$\hat{h}(S) = \frac{1}{2^n}\sum_x (-1)^{h(x)}\chi_S(x) =$$

$$\frac{1}{2^n}\Big(|\{h(x) = 0, x \in X\}| - |\{h(x) = 1, x \in X\}| - |\{h(x) = 0, x \notin X\}| + |\{h(x) = 1, x \notin X\}|\Big).$$

Denote the quantity $|\{h(x) = 0, x \in X\}|$ as a. As h is balanced on X, it follows that $|\{h(x) = 1, x \in X\}| = a$. The set X contains 2^{n-1} elements so $a = 2^{n-2}$. Given that h is balanced across $\{0,1\}^n$, both the sets $\{h(x) = 0, x \in \{0,1\}^n\}$ and $\{h(x) = 1, x \in \{0,1\}^n\}$ each have 2^{n-1} elements. Therefore:

$$|\{h(x) = 0, x \notin X\}| = |\{h(x) = 0, x \in \{0,1\}^n\}| - |\{h(x) = 0, x \in X\}| = 2^{n-2},$$

$$|\{h(x) = 1, x \notin X\}| = |\{h(x) = 1, x \in \{0,1\}^n\}| - |\{h(x) = 1, x \in X\}| = 2^{n-2}.$$

We can see that $\hat{h}(S) = 0$ which leads us to the required contradiction. □

Theorem 37. *Suppose F satisfies $\mathrm{D}^{\rightarrow}_{\mathrm{cc}}(F) = 1$. It then follows that $\mathrm{NADT}^{\oplus}(f) = 1$.*

Proof. Consider the total h-induced graph. For any unbalanced h by Lemma 34 we get that all shifts are good, so the graph is complete. It can't be split into connectivity components by vertex removal, therefore the partial h-induced graph has a single connectivity component. By Corollary 17 we have $\mathrm{NADT}^{\oplus}(f) = 0$.

For a balanced function, we use Lemma 36 to choose an $n-1$-dimensional subspace U, on which h is unbalanced. By Lemma 35, all the shifts in U are good. Select two arbitrary vertices x and y, from the same coset of U. Vertices x and y are connected in the total h-induced graph because their XOR belongs to U. Therefore cosets of U are cliques and they will remain connected in a partial h-induced graph. By Corollary 17 we conclude that $\mathrm{NADT}^{\oplus}(f) = 1$. □

6.2 Case $\mathrm{D}^{\rightarrow}_{\mathrm{cc}}(F) = 2$

We handle cases when h is unbalanced and balanced separately. In the first case, we observe that the XOR of two bad shifts results in a good shift. We then use a known result on the bound on sumset cardinality to show that the good shifts either contain a coset of a $n-1$-dimensional subspace or there exists *large enough* number of such shifts. Either of these cases implies a certain structure on the total h-induced graph, which allows us to get the desired lower bound. When h is balanced, we again consider the subspace on which it is unbalanced and analogously to the prior scenario, we deduce specific structure on the subspace allowing us to conclude the proof.

Lemma 38. *Assume F satisfies $\mathrm{D}^{\rightarrow}_{\mathrm{cc}}(F) = 2$ and the function h is unbalanced. Then the XOR of two bad shifts is a good shift.*

Proof. Assume Δ_1 and Δ_2 are bad shifts. Consider the cosets of the subspace $\langle \Delta_1, \Delta_2 \rangle$. There are a total of 2^{n-2} such cosets. As the function h is unbalanced, the h-induced partition has a class, denoted as H_1, which contains strictly more than 2^{n-2} elements. By the Pigeonhole principle, there exists a coset of $\langle \Delta_1, \Delta_2 \rangle$ that contains two elements, namely x and y, both of which belong to H_1. As $h(x) = h(y)$, the XOR of x and y produces a good shift. Additionally, x and y lay in the same coset, thus the shift $x \oplus y$ is a member of $\langle \Delta_1, \Delta_2 \rangle$. Within the subspace $\langle \Delta_1, \Delta_2 \rangle$, there are only three distinct non-zero shifts: Δ_1, Δ_2, and $\Delta_1 \oplus \Delta_2$. Given that both Δ_1 and Δ_2 are bad shifts, the only possible good shift among them is $\Delta_1 \oplus \Delta_2$. □

Theorem 39. *Let A and B be non-empty subsets of $\{0,1\}^n$. Define the sumset of A and B as $A + B = \{a + b | a \in A, b \in B\}$. Assume that A is not contained in a coset of any proper subspace of $\{0,1\}^n$. Then*

$$|A + B| \geq \min\{|A| + |B| - 2^{n-3}, 3 \cdot 2^{n-2}\}.$$

The proof of this theorem is moved to Appendix C.

Lemma 40. *Assume that $\mathrm{D}^{\rightarrow}_{\mathrm{cc}}(F) = 2$ and h is unbalanced. Then either there exists at least $5 \cdot 2^{n-3} - 1$ good shifts (not counting 0), or the set of good shifts contains a coset of an $n-1$-dimensional subspace.*

Proof. Let B be the set of bad shifts and $\overline{B}$ be the set of good shifts, these are complementary so $|B| + |\overline{B}| = 2^n$. There are two cases to consider: either B is a subset of a coset of a proper subspace or it is not. In the first case, let Q be a subspace and q be a vector in $\{0,1\}^n$ such that $B \subseteq Q + q$. We extend the coset $Q + q$ to a coset $Q' + q$ of some $n-1$-dimensional subspace Q'. Observe that since B is fully contained in $Q' + q$, another coset of Q' it is fully contained in $\overline{B}$.

In the second case, first observe that by Lemma 38 the sum of bad shifts is a good shift, thus we have $B + B \subseteq \overline{B}$. By Theorem 39 we have

$$|\overline{B}| \geq |B + B| \geq \min\{2|B| - 2^{n-3}, 3 \cdot 2^{n-2}\}.$$

We also know that $|B| + |\overline{B}| = 2^n$. As a result, either

$$|B| + 2|B| - 2^{n-3} \leq 2^n,$$

or

$$|\overline{B}| \geq 3 \cdot 2^{n-2}.$$

It is easy to see that in both cases

$$|\overline{B}| \geq 5 \cdot 2^{n-3}.$$

If we exclude the zero shift, we have at least $5 \cdot 2^{n-3} - 1$ good shifts. □

Lemma 41. *Assume F satisfies $\mathrm{D}^{\rightarrow}_{\mathrm{cc}}(F) = 2$. If h is unbalanced, then one of the following two conditions is true for the total h-induced graph:*

- *Total h-induced graph consists of two cliques, each being a coset of an $n-1$-dimensional subspace.*
- *Total h-induced graph is 2^{n-2}-vertex connected.*

Proof. We consider three cases.

Case 1: In this case, we assume that the set of good shifts contains a subspace Q of dimension $n-1$. Take two arbitrary points x and y from the same coset $Q+q$, where q is a specific vector in $\{0,1\}^n$. Then, x and y can be expressed as $x = x' \oplus q$ and $y = y' \oplus q$ for $x', y' \in Q$. Consequently, $x \oplus y = x' \oplus y' \in Q$. This shows that any two points in the coset of Q are connected by an edge in the total h-induced graph, forming cliques.

Case 2: Assume that the set of good shifts contains an $n-1$-dimensional coset $Q+q$, where Q is an $n-1$-dimensional subspace and q is a vector not in Q. Consider two arbitrary points x and y from different cosets of Q. Without loss of generality, let $x \in Q$ and $y \in Q+q$. There exists $y' \in Q$ such that $y = y' \oplus q$. Then, $x \oplus y = (x \oplus y') \oplus q \in Q+q$. Thus, an edge exists between x and y in the total h-induced graph, and, as a result, the graph contains a complete bipartite graph with parts being the cosets of Q. To make this graph disconnected one has to delete the whole part, thus the graph is 2^{n-1}-connected.

Case 3: Assume the set of good shifts satisfies neither of the first two conditions. Then, by Lemma 40, there must be at least $5 \cdot 2^{n-3} - 1$ good shifts. Take any two arbitrary non-neighboring vertices x and y; the sizes of their neighbor sets are at least $5 \cdot 2^{n-3} - 1$. Given that the total number of vertices excluding x and y is $2^n - 2$, the intersection of these neighbor sets must contain at least 2^{n-2} vertices. Hence, removing fewer than 2^{n-2} vertices cannot disconnect the graph. □

Lemma 42. *Assume F satisfies $\mathrm{D}^{\rightarrow}_{\mathrm{cc}}(F) = 2$. If h is unbalanced on a subspace Q of dimension $n-1$, then one of the following conditions must hold:*

- *The total h-induced graph consists of four distinct cliques, each of which corresponds to a coset of an $n-2$-dimensional subspace that is itself a subspace of Q.*
- *The subgraphs of the complete h-induced graph on the vertices of cosets of Q, are at least 2^{n-3}-vertex connected.*

Proof. For the proof we just apply Lemma 41 on the subspace Q. Formally, let B be a matrix whose columns form a basis for Q. We define a new function $h' : x \mapsto h(Bx)$ (x is of length $n-1$). Applying Lemma 41, we conclude that the total h'-induced graph either consists of cliques corresponding to cosets of an $n-2$-dimensional subspace Q' or that graph is 2^{n-3}-vertex connected.

To relate h' back to h, we consider a vector q not in Q and define two graph embeddings $\psi_1 : x \mapsto Bx$ and $\psi_2 : x \mapsto Bx \oplus q$ of the total h'-induced graph into the total h-induced graph. The images of these mappings are Q and $Q + q$. To see that they are indeed graph embeddings we notice that if x and y are connected in the total h'-induced graph, $x \oplus y$ is a good shift for h', so $B(x \oplus y)$ is a good shift for h, which implies that images of x and y under ψ_1 as well as images of x and y under ψ_2 are indeed connected in h-induced graph. The bound on vertex connectivity of cosets follows from these embeddings. Note that these mappings are also affine transformations that only differ by a shift. Therefore, the image of cosets in $\{0,1\}^{n-1}$ over these mappings will result in cosets of the same space in $\{0,1\}^n$, which finishes the proof. □

Theorem 43. *If function f is undefined on fewer than 2^{n-3} inputs and $\mathrm{D}^{\to}_{\mathrm{cc}}(F) = 2$, then $NADT(f) = 2$.*

Proof. We have two main cases to consider, depending on whether h is balanced or unbalanced. If h is unbalanced, we apply Lemma 41. As a result, either the h-induced graph consists of cliques corresponding to $n-1$-dimensional cosets, or the h-induced graph is 2^{n-2}-vertex connected. In the first case, by Corollary 17, we conclude that $\mathrm{NADT}^{\oplus}(f) \leq 1$, which is a contradiction. In the second case the graph is 2^{n-2}-vertex connected and again by Corollary 17 we find that $\mathrm{NADT}^{\oplus}(f) = 0$ because the function f is undefined on fewer than 2^{n-2} inputs, making it impossible to disconnect the graph by removing vertices.

If h is balanced, we use Lemma 36 to find a subspace Q where h becomes unbalanced. Then by Lemma 42 the graph will split either into four fully connected cosets, or into two 2^{n-3} vertex-connected cosets. As f in undefined in less than 2^{n-3} points we again use Corollary 17 and conclude that $\mathrm{NADT}^{\oplus}(f) \leq 2$. □

References

[1] Anurag Anshu, Naresh Goud Boddu, and Dave Touchette. Quantum log-approximate-rank conjecture is also false. In David Zuckerman, editor, *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 982–994. IEEE Computer Society, 2019. doi:10.1109/FOCS.2019.00063.

[2] Arkadev Chattopadhyay, Ankit Garg, and Suhail Sherif. Towards stronger counterexamples to the log-approximate-rank conjecture. In Mikolaj Bojanczyk and Chandra Chekuri, editors, *41st IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2021, December 15-17, 2021, Virtual Conference*, volume 213 of *LIPIcs*, pages 13:1–13:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.FSTTCS.2021.13.

[3] Arkadev Chattopadhyay and Nikhil S. Mande. A lifting theorem with applications to symmetric functions. In Satya V. Lokam and R. Ramanujam, editors, *37th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer

Science, FSTTCS 2017, December 11-15, 2017, Kanpur, India, volume 93 of *LIPIcs*, pages 23:1–23:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. doi:10.4230/LIPIcs.FSTTCS.2017.23.

[4] Arkadev Chattopadhyay, Nikhil S. Mande, Swagato Sanyal, and Suhail Sherif. Lifting to parity decision trees via stifling. In Yael Tauman Kalai, editor, *14th Innovations in Theoretical Computer Science Conference, ITCS 2023, January 10-13, 2023, MIT, Cambridge, Massachusetts, USA*, volume 251 of *LIPIcs*, pages 33:1–33:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.ITCS.2023.33.

[5] Arkadev Chattopadhyay, Nikhil S. Mande, and Suhail Sherif. The log-approximate-rank conjecture is false. *J. ACM*, 67(4):23:1–23:28, 2020. doi:10.1145/3396695.

[6] Gérard D. Cohen, Iiro S. Honkala, Simon Litsyn, and Antoine Lobstein. *Covering Codes*, volume 54 of *North-Holland mathematical library*. North-Holland, 2005.

[7] Susanna F. de Rezende, Or Meir, Jakob Nordström, Toniann Pitassi, Robert Robere, and Marc Vinyals. Lifting with simple gadgets and applications to circuit and proof complexity. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 24–30. IEEE, 2020. doi:10.1109/FOCS46700.2020.00011.

[8] 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.

[9] Uma Girish, Makrand Sinha, Avishay Tal, and Kewen Wu. Fourier growth of communication protocols for XOR functions. *CoRR*, abs/2307.13926, 2023. arXiv:2307.13926, doi:10.48550/arXiv.2307.13926.

[10] Parikshit Gopalan, Ryan O'Donnell, Rocco A. Servedio, Amir Shpilka, and Karl Wimmer. Testing fourier dimensionality and sparsity. *SIAM J. Comput.*, 40(4):1075–1100, jul 2011. doi:10.1137/100785429.

[11] Lianna Hambardzumyan, Hamed Hatami, and Pooya Hatami. Dimension-free bounds and structural results in communication complexity. *Israel Journal of Mathematics*, 253:555–616, 2023. doi:10.1007/s11856-022-2365-8.

[12] L. H. Harper. *Global Methods for Combinatorial Isoperimetric Problems*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2004. doi:10.1017/CBO9780511616679.

[13] Hamed Hatami, Kaave Hosseini, and Shachar Lovett. Structure of protocols for XOR functions. *SIAM J. Comput.*, 47(1):208–217, 2018. doi:10.1137/17M1136869.

[14] Stasys Jukna. *Boolean Function Complexity - Advances and Frontiers*, volume 27 of *Algorithms and combinatorics*. Springer, 2012. doi:10.1007/978-3-642-24508-4.

[15] Sampath Kannan, Elchanan Mossel, Swagato Sanyal, and Grigory Yaroslavtsev. Linear sketching over f_2. In Rocco A. Servedio, editor, *33rd Computational Complexity Conference, CCC 2018, June 22-24, 2018, San Diego, CA, USA*, volume 102 of *LIPIcs*, pages 8:1–8:37. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.CCC.2018.8.

[16] Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, 1996. doi:10.1017/CBO9780511574948.

[17] Bruno Loff and Sagnik Mukhopadhyay. Lifting theorems for equality. In Rolf Niedermeier and Christophe Paul, editors, *36th International Symposium on Theoretical Aspects of Computer Science, STACS 2019, March 13-16, 2019, Berlin, Germany*, volume 126 of *LIPIcs*, pages 50:1–50:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.STACS.2019.50.

[18] Nikhil S. Mande, Swagato Sanyal, and Suhail Sherif. One-way communication complexity and non-adaptive decision trees. In Petra Berenbrink and Benjamin Monmege, editors, *39th International Symposium on Theoretical Aspects of Computer Science, STACS 2022, March 15-18, 2022, Marseille, France (Virtual Conference)*, volume 219 of *LIPIcs*, pages 49:1–49:24. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.STACS.2022.49.

[19] Ashley Montanaro and Tobias Osborne. On the communication complexity of XOR functions. *CoRR*, abs/0909.3392, 2009. URL: http://arxiv.org/abs/0909.3392, arXiv:0909.3392.

[20] Ryan O'Donnell. *Analysis of Boolean Functions*. Cambridge University Press, 2014. doi:10.1017/CBO9781139814782.

[21] A. Rao and A. Yehudayoff. *Communication Complexity: and Applications*. Cambridge University Press, 2020. URL: https://books.google.com/books?id=emw8PgAACAAJ.

[22] Swagato Sanyal. Fourier sparsity and dimension. *Theory Comput.*, 15:1–13, 2019. doi:10.4086/toc.2019.v015a011.

[23] 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.

[24] Makrand Sinha and Ronald de Wolf. Exponential separation between quantum communication and logarithm of approximate rank. In David Zuckerman, editor, *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 966–981. IEEE Computer Society, 2019. doi:10.1109/FOCS.2019.00062.

[25] J.H. Spencer and L. Florescu. *Asymptopia*. Student mathematical library. American Mathematical Society, 2104. URL: https://books.google.com/books?id=uBMLugEACAAJ.

[26] Hing Yin Tsang, Chung Hoi Wong, Ning Xie, and Shengyu Zhang. Fourier sparsity, spectral norm, and the log-rank conjecture. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 658–667. IEEE Computer Society, 2013. doi:10.1109/FOCS.2013.76.

[27] Shengyu Zhang. Efficient quantum protocols for XOR functions. In Chandra Chekuri, editor, *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2014, Portland, Oregon, USA, January 5-7, 2014*, pages 1878–1885. SIAM, 2014. doi:10.1137/1.9781611973402.136.

[28] Zhiqiang Zhang and Yaoyun Shi. On the parity complexity measures of boolean functions. *Theor. Comput. Sci.*, 411(26-28):2612–2618, 2010. doi:10.1016/j.tcs.2010.03.027.

A Isoperimetric Inequalities

This section is devoted to proving Lemma 23.

Lemma 44. *For any subset $A \subseteq \{0,1\}^m$ of m-dimensional Boolean cube vertices, it holds that $|\Gamma' A| \geq |\Gamma' I^m_{|A|}|$.*

Proof. In the case $|A| = 1$, A and $I^M_{|A|}$ are just sets of single element and equality between $|\Gamma' A|$ and $|\Gamma' I^m_{|A|}|$ is obvious. Otherwise the set $I^m_{|A|}$ doesn't have isolated vertices. Thus, all the vertices in $I^m_{|A|}$ are neighbors of $I^m_{|A|}$ and $|\Gamma' I^m_{|A|}| = |\Gamma I^m_{|A|}| - |A|$. Meanwhile $|\Gamma' A| \geq |\Gamma A| - |A|$. Therefore Theorem 2 implies

$$|\Gamma' A| \geq |\Gamma A| - |A| \geq |\Gamma I^m_{|A|}| - |A| = |\Gamma' I^m_{|A|}|.$$

□

Lemma 45. *For a satisfying $V(m,r) \leq a \leq V\left(m, \left\lfloor \frac{m-1}{2} \right\rfloor\right)$ the following holds:*

$$|\Gamma' I^m_a| \geq |\Gamma' I^m_{V(m,r)}| = \binom{m}{r+1}.$$

Proof. Let r' be the maximum integer for which $V(m,r') \leq a$. Note that $r' \geq r$. If $a = V(m,r')$, the lemma is trivial. Otherwise, the inequality $a \leq V\left(\left\lfloor \frac{m-1}{2} \right\rfloor, m\right)$ implies that $r' \leq \left\lfloor \frac{m-1}{2} \right\rfloor - 1$.

The set I^m_a contains elements with Hamming weight up to r' and possibly some with weight $r'+1$. Let $B = I^m_a \setminus I^m_{V(m,r')}$ be the elements of I^m_a with Hamming weight $r'+1$. Define

$$B^+ = \{x \in \Gamma' B : |x| = r'+2\}.$$

Elements of B doesn't belong to $\Gamma' I^m_a$, since they belong to I^m_a, meanwhile elements of B^+ belong to $\Gamma' I^m_a$, since they are neighbors of elements from B and doesn't belong $I^m_{V(m,r')}$. Therefore,

$$|\Gamma' I^m_a| = \binom{m}{r'+1} - |B| + |B^+|.$$

To prove that $|B^+| \geq |B|$, let's consider a bipartite subgraph G of m-dimensional Boolean cube. The left part contains vertices with Hamming weight $r'+1$, and the right part contains vertices with Hamming weight $r'+2$. Here, B is a subset of the left part, and B^+ is the set of neighbors of B in G. Note that the degree of any vertex in the left part is

$$\deg_L = m - (r'+1) \geq m - \lfloor \frac{m-1}{2} \rfloor = \lceil \frac{m-1}{2} \rceil + 1,$$

while the degree of any vertex in the right part is

$$\deg_R = r' + 2 \leq \lfloor \frac{m-1}{2} \rfloor + 1.$$

Given that edges from B connect exclusively to vertices in B^+, it follows that $|B| \deg_L \leq |B^+| \deg_R$, which implies $|B^+| \geq |B|$. Consequently,

$$|\Gamma' I_a^m| \geq |I^m_{V(m,r')}| = \binom{m}{r'+1} \geq \binom{m}{r+1}.$$

□

Remark 46. A similar idea applies for a larger then $V\left(m, \lfloor \frac{m-1}{2} \rfloor\right)$. In that case $|B^+| \geq \frac{\deg_L}{\deg_R}|B| = \frac{m-(r'+1)}{r'+2}|B|$, therefore $|\Gamma' I_a^m| \geq \frac{m-(r'+1)}{r'+2}|\Gamma' I^m_{V(m,r')}|$. Note that here, unlike in previous case, r' must be the largest integer satisfying $V(m,r') \leq a$.

Lemma 47. *For $M \geq m$ and $a \leq 2^m$ it holds that $|\Gamma' I_a^m| \leq |\Gamma' I_a^M|$.*

Proof. The proof goes by induction on M. The base case for $M = m$ is trivial. Assuming the lemma holds for M, we aim to prove it for $M+1$. For this we construct a subset $A \subseteq \{0,1\}^M$ with $|A| = a$ and $|\Gamma' A| \leq |\Gamma' I_a^{M+1}|$. Here, the first Γ' refers to the M-dimensional Boolean cube, while the second Γ' refers to the $(M+1)$-dimensional Boolean cube.

We consider the 'slices' of the set I_a^{M+1} along its last coordinate:

$$A_0 = \{(x_1, \dots, x_M) : x \in I_a^{M+1}, x_{M+1} = 0\},$$

$$A_1 = \{(x_1, \dots, x_M) : x \in I_a^{M+1}, x_{M+1} = 1\}.$$

Denote by r the maximum number such that all the elements with Hamming weight at most r belong to I_a^{M+1}. The set A_0 contains all the elements with Hamming weight r and maybe some elements with Hamming weight $r+1$, while the set A_1 contains all the elements with Hamming weight $r-1$ and maybe some elements with Hamming weight r. Three cases arise based on the dimension M: either $2r+1 < M$, $2r+1 = M$ or $2r = M$. As $a \leq 2^m \leq 2^M$ it's impossible for r to take larger values. The third case is trivial, here a is just equal to 2^M and boundary is empty.

In the first case, we define

$$A = A_0 \sqcup \neg A_1,$$

where

$$\neg A_1 = \{(1-x_1, \dots, 1-x_M) : x \in A_1\}.$$

This union is indeed disjoint because the first set has elements with Hamming weight not above $r+1$, while the second has elements with weight at most $M-r$. Next, we notice that the cardinality of the boundary of A does not exceed that of I_a^{M+1}. Indeed, if a vertex belongs to $\Gamma' A$ it either belongs to $\Gamma' A_0$ or to $\Gamma' \neg A_1$ or to both. That is, $|\Gamma' A| \leq |\Gamma' A_0| + |\Gamma' \neg A_1|$. As we get $\neg A_1$ from A_1 with graph automorphism, $|\Gamma' \neg A_1| = |\Gamma' A_1|$. If vertex v belongs $\Gamma' A_0$, then vertex $(v,0)$ belongs $\Gamma' I_a^{m'+1}$ and similarly if v belongs $\Gamma' A_1$, then $(v,1)$ belongs $\Gamma' I_a^{m'+1}$. Therefore,

$$|\Gamma' I_a^m| \leq |\Gamma' I_a^M| \leq |\Gamma' A| \leq |\Gamma' I_a^{M+1}|.$$

In the second case, we adjust the construction of A because otherwise points from A_0 and $\neg A_1$ may overlap. The set A contains all vertices with Hamming weight at most r and at least $M-r+1$, and is filled up to cardinality a with vertices having Hamming weight $r+1 = M-r$. In this configuration, $\Gamma' A$ contains vertices of Hamming weight $r+1$ that are not in A. But the number of such elements doesn't exceed the number of elements with weight $r+1$, which doesn't belong to A_0 and all these elements lay in $\Gamma' A_0$, hence:

$$|\Gamma' I_a^m| \leq |\Gamma' I_a^M| \leq |\Gamma' A| \leq |\Gamma' A_0| \leq |\Gamma' I_a^{M+1}|.$$

This finishes the proof of the induction step and the lemma. □

Lemma 48. *For all M there exists such r that*

$$V\left(M-1, \left\lfloor \frac{M-2}{2} \right\rfloor - 2\right) \leq V(M,r) \leq V\left(M-1, \left\lfloor \frac{M-2}{2} \right\rfloor\right).$$

Proof. We select r to be the smallest number such that $V\left(M-1, \left\lfloor \frac{M-2}{2} \right\rfloor - 2\right) \leq V(M,r)$. Clearly, $r \leq \left\lfloor \frac{M-2}{2} \right\rfloor - 2$. For such r the following holds:

$$V\left(M-1, \left\lfloor \frac{M-2}{2} \right\rfloor - 2\right) \leq V(M,r) \leq V\left(M-1, \left\lfloor \frac{M-2}{2} \right\rfloor - 2\right) + \binom{M}{r}.$$

From here, we can further bound $\binom{M}{r}$ as follows:

$$\binom{M}{r} = \frac{M}{M-r}\binom{M-1}{r} \leq 2\binom{M-1}{r}.$$

Thus,

$$V(M,r) \leq V\left(M-1, \left\lfloor \frac{M-2}{2} \right\rfloor - 2\right) + 2\binom{M-1}{r} \leq V\left(M-1, \left\lfloor \frac{M-2}{2} \right\rfloor\right).$$

The last inequality holds since $r \leq \left\lfloor \frac{M-2}{2} \right\rfloor - 2$. □

Proof of Lemma 23. First, we consider the case when $|A| \leq V\left(k, \left\lfloor \frac{k-1}{2} \right\rfloor\right)$. Here we let $M = k$ and $a = |A|$. By Lemma 44 we have $|\Gamma' A| \geq |\Gamma' I_a^M|$. We will iteratively decrease M and a until $M = m$ and $a = V(m, \lfloor \frac{m-1}{2} \rfloor - 2)$ in a way that the boundary of the set I_a^M does not increase. When the algorithm finishes, the set I_a^M is a Hamming ball and its boundary contains all the elements with weight $\lfloor \frac{m-1}{2} \rfloor - 1$ and thus is of size $\binom{m}{\lfloor \frac{m-1}{2} \rfloor - 1}$. The size of the boundary of an initial set is at least as large.

We decrease the variables in the following way. While M is larger then m, if $a \leq V(M-1, \lfloor \frac{M-2}{2} \rfloor)$ we simply apply Lemma 47 to decrease M by one, otherwise we first set a to be $V(M, r)$, where r is selected by Lemma 48, the boundary won't increase after these assignment by Lemma 45 and then we again apply Lemma 47 to decrease M. On all steps of the algorithm, a doesn't exceed $V(M, \lfloor \frac{M-1}{2} \rfloor)$ which allows us to use these lemmas. When M reaches m it holds that $V(m, \lfloor \frac{m-1}{2} \rfloor - 2) \leq a \leq V(m, \lfloor \frac{m-1}{2} \rfloor)$ and we make a to be precisely equal to $V(m, \lfloor \frac{m-1}{2} \rfloor - 2)$ by applying Lemma 45 once again.

There exists a remaining case if initially $V(k, \lfloor \frac{k-1}{2} \rfloor) \leq |A| \leq 2^{k-1}$. It is only possible if k is even. In that case we use Remark 46 with $r' = \lfloor \frac{k-1}{2} \rfloor$ to conclude that

$$|\Gamma' A| \geq \frac{k}{k+2} \binom{k}{\frac{k}{2}} = \binom{k}{\frac{k}{2} - 1} = |\Gamma' I^k_{V(k, k/2-2)}|.$$

As $\left(k, \lfloor \frac{k-1}{2} \rfloor - 2\right) \leq V(k, \frac{k}{2} - 2) \leq V\left(k, \lfloor \frac{k-1}{2} \rfloor\right)$ the statement of the lemma follows from the first case. □

B Fourier Analysis

Here we provide the basic definitions from Fourier analysis. Functions that map $\{0,1\}^n \to \mathbb{R}$ form a 2^n-dimensional vector space under the operation of addition (indeed we can represent the function as a 2^n-dimensional vector of values for each of n-bit binary strings). For this space, we introduce an inner product:

$$\langle \psi, \theta \rangle = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} \psi(x)\theta(x).$$

Let's consider the parity functions, which are expressed as

$$\chi_S(x) = (-1)^{\sum_{i \in S} x_i},$$

with $S \subseteq [n]$. These functions form an orthonormal basis with respect to our previously defined inner product. As a direct consequence, any function ψ of the form $\{0,1\}^n \to \mathbb{R}$ can be uniquely represented as

$$\psi(x) = \sum_{S \subseteq [n]} \hat{\psi}(S)\chi_S(x).$$

The terms $\hat{\psi}(S)$ in the above expansion are known as Fourier coefficients. They can be computed in the following way:

$$\hat{\psi}(S) = \langle \psi, \chi_S \rangle = \frac{1}{2^n} \sum_{x \in \{0,1\}^n} \psi(x)\chi_S(x).$$

Indeed,

$$\langle \psi, \chi_S \rangle = \langle \sum_{T \subseteq [n]} \hat{\psi}(T)\chi_T, \chi_S \rangle = \sum_{T \subseteq [n]} \langle \hat{\psi}(S)\chi_T, \chi_S \rangle = \hat{\psi}(S),$$

where the last equality follows from the orthonormality property.

The Parseval theorem states that

$$\frac{1}{2^n} \sum_{x \in \{0,1\}^n} \psi(x)^2 = \langle \psi, \psi \rangle = \sum_{S \subseteq [n]} \hat{\psi}(S)^2.$$

For Fourier analysis involving Boolean functions, the typical convention is to consider function outputs in the set $\{-1, 1\}$ as opposed to $\{0, 1\}$. In this case, Parseval theorem implies

$$\sum_{S \subseteq [n]} \hat{\psi}(S)^2 = 1.$$

When analyzing the Fourier coefficients of a binary function ψ with the domain $\{0, 1\}$, we analyze the function $(-1)^{\psi(x)}$ rather than ψ directly. For an in-depth discussion on Fourier analysis, refer to [20].

C Bounds on Cardinality of Set Sum

In this section, we prove Theorem 39. The core idea of the proof is in the procedure that iteratively moves elements from the set B to the set A while shifting them by some vector. This operation doesn't change the sum of the sizes of the sets and is done in a manner that ensures the sumset cardinality at any given iteration is at most the cardinality of the sumset from the previous iteration. We provide a lower bound on the size of the sumsets when algorithm finishes and argue that initial subset's size is at least as large.

Lemma 49. *Let A and B be non-empty subsets of $\{0, 1\}^n$. If B is not contained in any coset of a proper subspace of $\{0, 1\}^n$ and A isn't equal to $\{0, 1\}^n$, then $|A + B| > |A|$.*

Proof. In case $|B| > |A|$ the statement of the lemma is obvious because sets are nonempty. From now on we assume that $|A| \geq |B|$.

For the sake of contradiction, assume that $|A + B| = |A|$. Now, let's define the set B' as B shifted by an element b from B, i.e., $B' = B + b$. The zero element is contained in B' and the size of the sumset $A + B'$ equals that of $A + B$, which in turn is $|A|$. Indeed, adding the element b to each element in the sumset results in a bijection between $A + B'$ and $A + B$. Now, as $A \subseteq A + B'$ (since the zero element is in A') and the sizes of the two sets are equal, we deduce that $A + B' = A$. Consequently, for every element $b' \in B'$, $b' + A = A$.

Let's define a set Q as the set of all elements q in $\{0, 1\}^n$ such that $q + A = A$. This set Q satisfies the properties of a subspace in $\{0, 1\}^n$. Indeed, for any two elements q_1 and q_2 in Q, their sum when added to A remains A, i.e., $q_1 + q_2 + A = q_1 + A = A$. However, Q is not equal to $\{0, 1\}^n$. To illustrate this, for a given element a_0 in A, when q varies over $\{0, 1\}^n$, the summation $q + a_0$ ranges over all elements in $\{0, 1\}^n$, which inevitably includes elements outside of A. Since every shifted set $b' + A$ with $b' \in B'$ is A, we have $B' \subseteq Q$. This implies that B is contained in the coset defined by $b + Q$, leading to a contradiction, which finishes the proof of the lemma. □

Lemma 50. *Let A and B be non-empty subsets of $\{0, 1\}^n$. Assume that A is not contained in any coset of a proper subspace of $\{0, 1\}^n$. Let Q be the smallest subspace such that B is contained in a coset of Q. Then either $|A + B| > |A|$ or A satisfies the following condition: for each coset of Q, either all vectors from that coset belong to A or none do.*

Proof. Let us consider the cosets of Q. For each coset, we select an arbitrary vector q_i from that coset. Assume B is contained in the coset $Q + \tilde{q}$. We define $A_i = A \cap (Q + q_i)$, that is, A_i consists of the vectors from A that are in the coset $Q + q_i$. We first prove that for distinct A_i and A_j, their respective sum-sets $A_i + B$ and $A_j + B$ do not intersect. Consider arbitrary vectors $a_1 \in A_i, a_2 \in A_j$, and $b_1, b_2 \in B$. Notice that $a_1 + b_1 = q_i + \tilde{q} + (a_1 + q_i) + (b_1 + \tilde{q})$ and $a_2 + b_2 = q_j + \tilde{q} + (a_2 + q_j) + (b_2 + \tilde{q})$. As vectors $(a_1 + q_i), (a_2 + q_j), (b_1 + \tilde{q}), (b_2 + \tilde{q})$, belong to Q and vectors q_i and q_j are from different cosets of Q, it follows that $a_1 + b_1$ and $a_2 + b_2$ must belong to different cosets, ensuring that $(A_i + B) \cap (A_j + B) = \varnothing$.

Consequently, the sum-set $A + B$ can be partitioned as follows:

$$A + B = \bigsqcup_i (A_i + B).$$

We further note that $|A_i + B| = |A_i + q_i + B + \tilde{q}|$. Indeed XORing each element with $q_i + \tilde{q}$ establishes a bijection between these two sets. Since both $A_i + q_i$ and $B + \tilde{q}$ are contained in the subspace Q, and given that Q is the smallest subspace containing a coset of B, Lemma 49 can be applied unless $A_i + q_i$ is a empty or equal to Q. This results in $|A_i + B| > |A_i|$, unless A_i is empty or contains all the vectors from corresponding coset. Combining this result with our partition of $A + B$ completes the proof. □

Now we provide the main algorithm (see Algorithm 1).

Algorithm 1 Algorithm for Lemma 51

Input: A_0, B_0.

```
$i \leftarrow 0$
$Q_0 \leftarrow$ smallest subspace such that $B_0$ is contained in a coset of $Q_0$
while $\exists q \in \{0,1\}^n : A_i \cap Q_i + q \neq 0, A_i \cap Q_i + q \neq Q_i + q$ do
    $b' \leftarrow$ select arbitrary $b'$ in $B_i$
    $\tilde{B} \leftarrow B_i + b'$
    $a' \leftarrow$ select any $a'$ such that $a' + \tilde{B} \nsubseteq A_i$        ▷ We can find such $a'$ by Lemma 50
    $B' \leftarrow \{b \in \tilde{B} | a' + b \notin A_i\}$
    $A_{i+1} \leftarrow A_i \cup (a' + B')$
    $B_{i+1} \leftarrow \tilde{B} \setminus B'$
    $Q_{i+1} \leftarrow$ smallest subspace such that $B_{i+1}$ is contained in a coset of $Q_{i+1}$
    $i \leftarrow i + 1$
end while
```

Lemma 51. *Let A_0, B_0, A_i, B_i be as given in Algorithm 1. The size of the setsum $A_0 + B_0$ is at least as large as that of $A_i + B_i$ at any iteration i of the algorithm, and sizes of sets A_0, B_0, A_i, B_i satisfy $|A_0| + |B_0| = |A_i| + |B_i|$ at each iteration.*

Proof. We start by observing that $|A_i + B_i| = |A_i + \tilde{B}|$. This equality holds because $A_i + \tilde{B} = A_i + B_i + b'$, and XORing with b' establishes a bijection between $A_i + B_i$ and $A_i + \tilde{B}$. The loop's condition assures us that there exists a coset of Q_i such that its intersection with A is neither empty nor consists of all vectors of the coset. Given that $\tilde{B}$ is simply B translated by a vector b, Q_i is also the smallest subspace, coset of which contains $\tilde{B}$. Therefore, we can apply Lemma 50 to conclude that $|A_i + \tilde{B}| > |A_i|$. This allows us to choose a vector a' such that $a' + \tilde{B}$ is not a subset of A_i. By the definition of a', B' is non-empty. Now we construct the sets A_{i+1} and B_{i+1}. They have the following properties: First, $|A_{i+1}| = |A_i| + |B'|$. This is true because $A_i \cap (a' + B') = \varnothing$, which follows directly from the choice of B'. The cardinality of $\tilde{B} \setminus B'$ is $|B_i| - |B'|$. Consequently, $|A_{i+1}| + |B_{i+1}| = |A_i| + |B_i|$.

Next, $A_{i+1} + B_{i+1} \subseteq A_i + \tilde{B}$. The set $A_i + B_{i+1}$ is obviously contained in $A_i + \tilde{B}$. It remains to show that $(a' + B') + B_{i+1} = (a' + B') + (\tilde{B} \setminus B')$ is also contained in $A_i + \tilde{B}$. To demonstrate this, consider an arbitrary $a \in (a' + B')$ and $b \in \tilde{B} \setminus B'$. Then $a = a' + b'$ for some $b' \in B'$. Because b is not in B', $a' + b$ is an element of A_i. Therefore, $(a' + b) + b'$ belongs to $A_i + \tilde{B}$. By induction, we conclude that $|A_i| + |B_i| = |A_0| + |B_0|$ and $|A_i + B_i| \leq |A_0 + B_0|$. □

It remains to ptove the lower bound of the sumset size $|A_i + B_i|$ for the termination step of the algorithm. Initially, we construct $\tilde{B}$ to always include the element 0 to ensure that B_i is never empty throughout the algorithm. Indeed, if B_i were empty at some iteration i, it would imply that $B' = \tilde{B}$ in the previous iteration $i - 1$, which contradicts the fact that $a' + 0 \in A_i$ and therefore $0 \notin B'$. Consequently, $|A_i + B_i| \geq |A_i|$.

The algorithm halts when the condition specified in line 3 is not met. Specifically, given that Q_i is the smallest subspace such that B_i is contained in a coset of Q_i, for all cosets of Q_i, the intersection of A_i with that coset is either empty or contains the entire coset. It follows that $|B_i| \leq |Q_i| = 2^{\dim Q_i}$, yielding

$$|A_0 + B_0| \geq |A_i + B_i| \geq |A_i| \geq |A_0| + |B_0| - |B_i| \geq |A_0| + |B_0| - 2^{\dim Q_i}.$$

If the dimension of Q_i is at most 2^{n-3}, we obtain the desired bound. Next we consider the case when $\dim Q_i \geq 2^{n-2}$. We use the fact that $A_i \supseteq A_0$. When $\dim Q_i = n$ or $\dim Q_i = n-1$, it's straightforward to see that A_i would span the entire $\{0,1\}^n$ space. In the first case it follows since A_0 is non-empty and in the second case it follows because A_0 is not contained in neither Q_i, nor $\overline{Q_i}$. Next, consider the case $\dim Q_i = n - 2$. In this case, Q_i has four distinct cosets. Since A_0 is not contained in any proper subspace of $\{0,1\}^n$, it must contain elements in at least three of these cosets. Therefore, for these three cosets, A_i would contain all the elements, leading to a size of $3 \times 2^{n-2}$ at the minimum. This concludes the proof of Theorem 39.