Electronic Colloquium on Computational Complexity, Report No. 166 (2020)

Lower Bounds for Monotone Arithmetic Circuits Via Communication Complexity

Arkadev Chattopadhyay* Rajit Datta† Partha Mukhopadhyay‡

November 9, 2020

Abstract

Valiant [Val80] showed that general arithmetic circuits with negation can be exponentially more powerful than monotone ones. We give the first qualitative improvement to this classical result: we construct a family of polynomials P_n in n variables, each of its monomials has positive coefficient, such that P_n can be computed by a polynomial-size *depth-three formula* but every monotone circuit computing it has size $2^{\Omega(n^{1/4}/\log(n))}$.

The polynomial P_n embeds the SINK $\circ$ XOR function devised recently by Chattopadhyay, Mande and Sherif [CMS20] to refute the Log-Approximate-Rank Conjecture in communication complexity. To prove our lower bound for P_n, we develop a general connection between corruption of combinatorial rectangles by any function $f \circ \text{XOR}$ and corruption of product polynomials by a certain polynomial P^f that is an arithmetic embedding of f. This connection should be of independent interest.

Using further ideas from communication complexity, we construct another family of set-multilinear polynomials $f_{n,m}$ such that both $F_{n,m} - \epsilon \cdot f_{n,m}$ and $F_{n,m} + \epsilon \cdot f_{n,m}$ have monotone circuit complexity $2^{\Omega(n/\log(n))}$ if $\epsilon \geq 2^{-\Omega(m)}$ and $F_{n,m} := \prod_{i=1}^{n} (x_{i,1} + \cdots + x_{i,m})$, with $m = O(n/\log n)$. The polynomials $f_{n,m}$ have 0/1 coefficients and are in VNP. Proving such lower bounds for monotone circuits has been advocated recently by Hrubeš [Hru20] as a first step towards proving lower bounds against *general cicuits* via his new approach.

*TIFR, Mumbai. Partially supported by a MATRICS grant of the Science and Engineering Research Board, DST, India. arkadev.c@tifr.res.in

†CMI, Chennai. Partially supported by a TCS Fellowship. rajit@cmi.ac.in

‡CMI, Chennai. partham@cmi.ac.in

ISSN 1433-8092

1 Introduction

The arithmetic analog of Cook's P vs. NP question is Valiant's VP vs. VNP question. Despite the latter being seemingly an easier question, progress on it has been frustratingly slow. One class of natural but restricted circuits, in both the Boolean and arithmetic world, is that of monotone circuits where a lot of progress has taken place. Exponential lower bounds on these circuits have been known for long, first in the arithmetic world due to the work of Shamir and Snir [SS77] and then the breakthrough work of Razborov [Raz85b] in Boolean complexity. A long line of work has been carried out both in the arithmetic [SS77, SS80, Val80, JS82, GS12, RY11, Yeh19, Sri19] and the Boolean world (see for example [Raz85b, Raz85a, Tar88, KW, RW92, RM99, HR00, Ros15, COS17, PR17, GKRS19]) to further strengthen these bounds and make progress. Despite these efforts, several problems remain open even for monotone complexity. In this paper, we focus on two such problems.

The first problem concerns the understanding of the relative powers of monotone and non-monotone computation. How much advantage do non-monotone circuits have, exploiting cancellation, to compute a target function or polynomial that itself is monotone? This was first answered in the arithmetic world by Valiant [Val80], forty years ago. Valiant showed that a certain monotone polynomial can be computed efficiently by general circuits, but monotone circuits need exponential size to compute it. In the Boolean world, Razborov [Raz85a] gave a super-polynomial separation between monotone and non-monotone computations by showing that the bipartite matching problem in P needs $n^{\Omega(\log n)}$-size monotone circuits. Later, Tardos [Tar88] proved an exponential separation using a different function. Raz and Wigderson [RW92] showed that matching needs exponential size monotone formulas.

Several researchers have investigated the following natural question that arises from these separation results: what is the weakest non-monotone model that can compute a monotone function which is hard even for the most powerful monotone model? It is known that matching can not only be computed in P, but it can also be computed efficiently, i.e. using polynomially many processors, in fast parallel time or depth of $O(\log n)^2$ by randomized algorithms again using cancellations. Yet, Razborov's lower bound puts it outside monotone P. This was the best known separation for long. Recently, Göös et. al. [GKRS19] improved this by finding another function that has such an efficient and parallel deterministic algorithm and showed that it even requires exponential size monotone circuits. Can one find hard functions for monotone circuits in even shallower depths of general circuits? An important result of Ajtai and Gurevich [AG87] showed that there are monotone functions that can be computed efficiently by constant-depth, non-monotone circuits, and yet need super-polynomial size to be computed by constant-depth monotone circuits. The recent work of Chen et.al. [COS17] significantly strengthens this by exhibiting a monotone function having efficient constant-depth circuits with negations but that still requires exponential size to be computed by monotone circuits of constant depth even though they are allowed to use gates computing a powerful monotone function like Majority. However, these developments still leave tantalizingly open the possibility of finding a monotone function computable by small size constant-depth circuits with negation that require *exponential size* to be computed by *unrestricted-depth* monotone circuits.

The arithmetic analog of this possibility remained unresolved as well. However, for *non-commutative* circuits, it was answered in the positive by Hrubeš and Yehudayoff [HY13] by exhibiting an exponential separation of constant-depth arithmetic formulas and monotone circuits. We provide the first such separation in the more natural and commonly studied setting

of *commutative* arithmetic circuits that we describe next.

We consider set-multilinear monomials over sets of variables $X_1, X_2, \ldots, X_n$ where $X_i := \{x_{i,1}, x_{i,2}, \ldots, x_{i,m}\}$, i.e. multilinear monomials that depend precisely on just one of the m variables from each of the n blocks[1]. In order to generate hard polynomials for monotone arithmetic circuits, we will use an idea of embedding a Boolean function f that is known to be hard in the 2-party setting of communication complexity. The general idea to do so, is to associate with each multilinear monomial κ a unique m-bit Boolean string x. The coefficient of κ in the polynomial would be just $f(x) \in \{0,1\}$. Applying this general framework, for each $f : \{0,1\}^m \to \{0,1\}$ one derives a unique monotone polynomial P^f with $0/1$ coefficients.

However, we want our target polynomial to be also easy to be computed by constant-depth arithmetic formulas, using the power of cancellations. This makes the design of associating a Boolean string to a monomial and the choice of f delicate. In particular, we will have to perturb a Boolean f at each point by a slight amount δ to create a real-valued f' that uniformly approximates f. We will then be able to argue that $P^{f'}$ retains the monotone hardness of P^f but just becomes easy to be computed with cancellations. Creating hard polynomials for arithmetic circuits in this way, as far as we know, was not done in any of the prior works.

More precisely, the set of such multilinear monomials can be identified with the set of all mappings from $[n]$ to $[m]$, the latter denoted by $\mathcal{F}_{n,m}$. For a mapping $\sigma \in \mathcal{F}_{n,m}$ (or a set-multilinear monomial κ) we define a parity vector $\vec{\oplus}(\sigma) \in \{0,1\}^m$ (or $\vec{\oplus}(\kappa)$) where the jth entry is $|\sigma^{-1}(j)| \pmod 2$. The parity vector $\vec{\oplus}(\kappa)$ of a monomial κ will be the unique m-bit Boolean string we associate with it. Now we describe the hard Boolean function f that we will be using: this will be the same function called SINK that was recently used in the refutation of the Log-Approximate-Rank Conjecture (LARC) in [CMS20]. Set $m = \binom{k}{2}$, so that each bit of a string x in $\{0,1\}^m$ is viewed as the assignment to an edge of a complete graph on k vertices $\{1, \ldots, k\}$. If a bit of x, corresponding to an edge (u, v) in K_k is set to 0, then the edge orients $u \leftarrow v$, otherwise the edge orients $u \to v$, when $u < v$. An input string x, can thus be interpreted as a tournament by orienting the edges of the complete graph K_k. A vertex in a tournament is called a sink vertex if its out-degree is 0. SINK evaluates to 1 on x if the tournament specified by x has a sink vertex. Otherwise, SINK evaluates to 0 on x.

Thus, a monomial κ is called a sink monomial if $\text{SINK}(\vec{\oplus}(\kappa)) = 1$. Let $P^{\text{Non-Sink}}_{n,m}$ ($P^{\text{Sink}}_{n,m}$) be the polynomial that is supported completely on non-sink (sink) monomials. It can be shown that each of $P^{\text{Non-Sink}}_{n,m}$ and $P^{\text{Sink}}_{n,m}$ is hard for monotone circuits, but we don't know whether any one of them is easy for general formulas. We thus look at a slight perturbation of $P^{\text{Non-Sink}}_{n,m}$: A polynomial $P_{n,m}$ is called a δ-non-sink polynomial if the coefficient of every monomial κ in $P_{n,m}$ lies in the interval $[0, \delta]$ if κ is a sink, otherwise (i.e. κ is not a sink) it lies in the interval $[1-\delta, 1]$. We are ready to state our result now:

Theorem 1.1.

1. *For every constant $0 < \delta < 1$, there exists a δ-non-sink polynomial $P_{\delta,n,m}$ that has a depth-three arithmetic formula of size $O_\delta(nm^4)$.*

2. *There exists a sufficiently small constant $\delta < 1$, such that every δ-non-sink polynomial $Q_{\delta,n,m}$ needs monotone circuits (with no depth restrictions) of size $2^{\Omega(\sqrt{m})}$ to be computed, when $4m \ln m \leq n$.*

[1]Many commonly studied polynomials like the Permanent and Determinant are set-multilinear.

Remark 1.1. *Let us recall that Valiant [Val80], Jerrum and Snir [JS82] gave exponential separations between VP and monotone VP. Applying modern depth reduction techniques [AV08, Koi12, Tav15, GKKS16] to either of their polynomials would result in them being computed by depth-3/depth-4 formulas of exponential size of* $2^{O(\sqrt{d}\log N)}$*, where the polynomials are* N*-variate and degree* d*. More precisely, for Valiant's family of polynomials the* n^{th} *polynomial* P_n *has number of variables* $N = O(n^2)$ *and degree* $d = O(n^2)$*. For Jerrum-Snir family of polynomials, the* n^{th} *polynomial* P_n *has number of variables* $N = O(n^2)$ *and degree* $d = O(n)$*.*

More generally, the use of parity vectors of monomials as described above, gives us a conceptually simple but novel way of constructing set-multilinear polynomials by embedding hard Boolean functions of the form $f \circ \text{XOR}$, where $f : \{0,1\}^m \to \{0,1\}$. These functions are called XOR functions and we make more use of them, via parity vectors of monomials, to deal with a different problem.

We consider the problem of making progress towards proving lower bounds for general arithmetic circuits. Efforts in this direction have remained stuck for a long time. Very recently, Hrubeš [Hru20] formulated a new interesting approach that reduces the task of proving lower bounds on the size of general arithmetic circuits to that of proving lower bounds on the monotone complexity of special classes of polynomials. A crucial aspect of the approach is that one, sort of, *plants* a conjectured hard polynomial scaled by a vanishingly small number ϵ inside an otherwise super-easy polynomial. One needs to prove that the new polynomial remains hard for monotone circuits to conclude that the conjectured hardness of the original polynomial against general circuits is true. There are no known monotone lower bounds against such polynomials and Hrubeš argues that new techniques in monotone complexity need to be developed to take on this challenge. Using our communication complexity based approach, we take a first step in this direction.

More precisely, Hrubeš showed that if a polynomial f_n is computed efficiently by a general circuit of size s, then there exists an $\epsilon_0 > 0$, such that for every $\epsilon \leq \epsilon_0$, the function $F_n + \epsilon \cdot f$ has efficient monotone circuits, where $F_n := (1 + \sum_i x_i)^d$ is the polynomial that contains all monomials of degree at most d. The difficulty in proving monotone lower bounds for such polynomials is the following: for $\epsilon = 0$, they become easy for monotone circuits and yet we need to show that for a tiny non-zero ϵ, the function is hard. Most monotone lower bounds in the literature are based on covering arguments, i.e. they are valid against all polynomials which are supported on the same hard set of monomials, paying no regard to the specific set of coefficients used in the target polynomial. Such arguments cannot obviously work against the kind of polynomials suggested by Hrubeš. Recently, Yehudayoff [Yeh19] and Srinivasan [Sri19] gave new arguments to prove monotone lower bounds that take into consideration the distribution of coefficients. In fact, covering arguments would not be enough to even prove our Theorem 1.1 as the support set of a δ-non-sink polynomial can be full and our argument does make essential use of the *distribution* of the coefficients of monomials. Still, it is not clear how to use any of these arguments in the context of Hrubeš' question.

We give a new argument based on the discrepancy method from communication complexity to make progress. To discuss this approach, let us consider the Boolean function $\text{MOD}_3 \circ \text{XOR}$, where $\text{MOD}_3(x) = 0$ if and only if $\sum_{i=1}^{n} x_i \equiv 0 \bmod 3$. It can be shown that this problem has small discrepancy w.r.t combinatorial rectangles and is therefore hard for Alice and Bob to solve in Yao's 2-party communication model. Our main insight is that embedding such a problem in a set-multilinear polynomial via the parity vectors of the monomials results in a monotone polynomial that is very hard for monotone circuits in the sense of Hrubeš' Theorem. More

precisely, we define

$$P_{n,m}^{\text{MOD}_3} := \sum_{\substack{\sigma:[n]\mapsto[m] \\ \text{MOD}_3(\vec{\oplus}(\sigma))=0}} \prod_{i=1}^{n} x_{i,\sigma(i)}.$$

Clearly this polynomial is in VNP. Let the full set-multilinear polynomial be defined as

$$F_{n,m} := \prod_{i=1}^{n} (x_{i,1} + \cdots + x_{i,m}).$$

Our main result is the following:

Theorem 1.2. *There exists a constant $\gamma > 0$ such that both the polynomials $F_{n,m} - \epsilon \cdot P_{n,m}^{MOD_3}$ and $F_{n,m} + \epsilon \cdot P_{n,m}^{MOD_3}$ have monotone complexity $2^{\Omega(m)}$, provided $4m \ln m \leq n$ and $\epsilon \geq 2^{-\gamma m}$.*

In summary, both Theorems 1.1 and 1.2 obtain monotone lower bounds of the kinds that were not obtained before our work. They use two powerful techniques from communication complexity: the corruption and the discrepancy method. That communication complexity methods and arithmetic complexity lower bound techniques should be related is not a complete surprise. A low cost communication protocol induces a decomposition of the communication matrix of the target function into a non-negative sum of few rectangles. A small monotone circuit results in the decomposition of the computed polynomial into a sum of non-negative product polynomials. In fact, several researchers have used this similarity to draw intuition from communication complexity to prove monotone lower bounds. However, there is an important difference (among others) that, we feel, has prevented direct usage of measures like corruption bounds in past work. In the arithmetic decomposition, the partition of the input variables used varies across the product polynomials used. In standard version of communication complexity, this does not happen. One chooses the most convenient possible partition to prove lower bounds. Raz and Yehudayoff [RY11] used sophisticated exponential sum estimates of [BGK06] to overcome this difficulty. We, on the other hand, use a simple but novel trick of using parity vectors to embed a hard XOR function in our target polynomial. Our general Corruption Transfer Lemma 4.1 and the proof of Theorem 1.2 are evidences of the broad applicability of this idea. We feel that this would also be further useful in proving lower bounds for circuits that are less restricted than monotone ones.

1.1 Our Ideas and Techniques

Separating General Depth-3 From Monotone Circuits

We are looking for polynomials (with positive coefficients) that are very hard for monotone circuits, yet are not only easy with cancellations but even remain easy in constant-depth. There are two important computations in which cancellations are known to help. First, is the computation of the determinant. This is not a monotone polynomial but one can effectively embed it into a such a polynomial. For example, Jerrum and Snir [JS82] used spanning tree polynomial to separate VP from monotone VP and it can be expressed as a determinant of linear forms [W70]. But determinant is unlikely to be easy for small depth-computations. Ben-Or [SY10] showed that depth three is capable of interpolating which does make crucial use of

cancellations. This yields small-size depth-3 circuits for computing elementary symmetric polynomials, making them a possible target for separating the power of constant-depth general formulas and monotone unrestricted-depth circuits. However, it is well known that elementary symmetric polynomials are actually easy for even monotone arithmetic branching programs (ABPs).

This forces us to seek alternative powers of cancellations: depth-2 circuits cannot be more powerful than their monotone counterparts. How do cancellations in random depth-three set-multilinear circuits take place? Consider the following $\Sigma\Pi\Sigma$ circuit:

$$\frac{1}{N}\sum_{i=1}^{N}\prod_{j=1}^{n}\left(b_1^i x_{j,1} + b_2^i x_{j,2} + \cdots + b_m^i x_{j,m}\right) \tag{1}$$

Here, $b^1, \ldots, b^N$ are randomly sampled points from $\{1,-1\}^m$. To talk about cancellations, let us consider the coefficient of a monomial $\kappa \coloneqq \prod_{j=1}^{n} x_{j,\sigma(j)}$, where $\sigma : [n] \to [m]$ is the map defining κ. The coefficient of κ in the polynomial computed by the random circuit is $(1/N)\cdot\sum_{i=1}^{N}\prod_{j=1}^{n} b^i_{\sigma(j)}$ which can be re-written as

$$\frac{1}{N}\sum_{i=1}^{N}\prod_{\ell=1}^{m}(b_\ell^i)^{\left|\sigma^{-1}(\ell)\right| \bmod 2} \tag{2}$$

Thus, the parity vector $\vec{\oplus}(\kappa) \coloneqq (|\sigma^{-1}(1)| \bmod 2, \ldots, |\sigma^{-1}(m)| \bmod 2)$ of monomial κ determines its coefficient. All monomials that are even, i.e. their parity vectors are all zeroes, will have coefficient exactly 1, i.e. there was no cancellation for them. For any odd monomial κ, the expected value of the coefficient is 0, i.e. we expect a lot of cancellations associated with κ taking place. To translate this phenomenon from random circuits to a fixed deterministic circuit, we choose the points $b^1, \ldots, b^N$ to form an ϵ-biased space in $\{0,1\}^m$. Armed with this insight, we want to craft a polynomial where the deterministic circuit is able to suppress the magnitude of the coefficients of a select group of monomials while keeping the rest of the magnitudes high. The group to be selected should be such that the polynomial becomes hard for monotone circuits.

The basic weakness of a monotone circuit of size s computing a multilinear polynomial P is well known i.e. such a P can be expressed as a sum of few *balanced product* polynomials. Each such product polynomial has a structure resembling that of a *combinatorial rectangle*, an object that appears commonly in the study of communication complexity. This similarity has been the source of intuition in past works in arithmetic complexity in general and monotone complexity in particular. But a direct correspondence had not been established, as far as we know, until now. We do so in this work and crucially use this correspondence to prove our monotone lower bounds.

Before we describe further details, we point out an interesting connection to the famous Log-Rank Conjecture in communication complexity and our problem. Consider any set multilinear polynomial P^f and a partition $\{\mathcal{X}_1, \mathcal{X}_2\}$ of the sets of its input variables. A natural matrix w.r.t the partition is one where each row corresponds to a set-multilinear monomial in variables from $\mathcal{X}_1$ and every column to one in variables from $\mathcal{X}_2$. Each entry of this matrix is the coefficient in the target polynomial of the monomial formed by the product of the corresponding row and column monomials. Observe that if P^f is computed by a syntactic depth-3 set-multilinear circuit of top fain-in s, then the matrix has rank at most s, as every product gate computes polynomial

whose corresponding matrix has rank 1. Let our polynomial P^f be the embedding of the Boolean function f by the parity vector scheme. This translates into saying that the communication matrix of $f \circ \text{XOR}$ has rank at most s. To prove our lower bound on the monotone circuit complexity of P^f, current techniques end up proving lower bounds on the number of product polynomials needed in any decomposition of P^f. Each product polynomial appearing in the sum comes with its own partition. But even if all these partitions were $\{\mathcal{X}_1, \mathcal{X}_2\}$, this would imply proving a lower bound on the number of combinatorial rectangles needed to non-negatively sum up to the communication matrix of $f \circ \text{XOR}$. A strong lower bound on the monotone complexity of P^f when s, the upper bound on the top fan-in of a depth-3 circuit computing P^f, is just polynomial in n, m would thus result in the refutation of the Log-Rank Conjecture (LRC) via the function $f \circ \text{XOR}$. No refutation of the LRC is known. Under these circumstances, we do the next best possible thing: we take recourse to the recent refutation of the approximate/randomized version of the LRC [CMS20], by embedding an *approximate* version of the f in our polynomial, where f is the same function SINK used in the refutation.

The set of points at which SINK outputs 1 is a small union of mutually disjoint sub-cubes. Roughly speaking, we observe that parity vectors in each sub-cube can be expressed by a single $\Sigma\Pi\Sigma$ circuit of the form in (1) by appropriately 'shifting' the parity vector. Thus, we express S by writing the polynomial as a sum of few (as many as the number of sub-cubes) such depth-3 circuits and then collapse the whole thing naturally to a single depth-3 circuit. The resulting polynomial has negative coefficients. We turn it monotone by subtracting it from a slightly scaled-up full product polynomial.

This polynomial is a δ-non-sink polynomial. As it has nearly full support, one needs to find an argument that uses the distribution of its coefficients to prove its hardness against balanced product polynomials. Our simple but key insight is to regard this polynomial's coefficients as the acceptance probabilities of the parity vectors of the respective monomials. In other words, just as SINK $\circ$ XOR was shown by [CMS20] to be hard to be pointwise δ-approximated by a small non-negative sum of combinatorial rectangles, we should in principle be able to say that a small sum of non-negative product polynomials cannot compute any δ-non-sink polynomial. Realizing this idea requires care. More interestingly in doing so, we develop a general transfer theorem that relates *rectangular corruption*, a very useful measure in communication complexity, under natural probability distributions on the input space of Boolean XOR functions to that of an analogous corruption-like measure on the set-multilinear monomial space. This simple but powerful correspondence is developed in Section 4. While our immediate use of this correspondence is to establish the lower bound for δ-non-sink polynomials to prove Theorem 1.1, we believe this to be of independent interest in monotone complexity.

ϵ-Sensitive Monotone Lower Bounds

Now we briefly explain the main ideas for proving Theorem 1.2. The crucial insight comes from the fact that the boolean function $\text{MOD}_3 \circ \text{XOR}$ has small *discrepancy* w.r.t the combinatorial rectangles. The notion of discrepancy was defined by Babai, Nisan and Szegedy [BNS92]. To exploit this, we define two measures W_0 and W_1 on the space of set-multilinear monomials as follows: W_0 puts uniform weights to the set of monomials κ such that the Hamming weight (wt) of $\vec{\oplus}(k)$ is a multiple of 3. The measure W_1 acts similarly on the set of monomials κ such that $\text{wt}(\vec{\oplus}(\kappa)) \equiv 1 \pmod 3$. Combining W_0 and W_1, we define the main measure W on any polynomial P as $W(P) = W_1(P) - W_0(P)$.

Using a (nearly)-equidistribution property of parity vectors, we show that the number of monomials κ such that $\mathrm{wt}(\vec{\oplus}\kappa) \equiv b \pmod 3$ are roughly same for $b \in \{1,2,3\}$. This immediately shows that the contribution of the measure W for the polynomial $P = F_{n,m} - \epsilon \cdot P_{n,m}^{\mathrm{MOD}_3}$ is approximately proportional to the contribution from $P_{n,m}^{\mathrm{MOD}_3}$. In particular, this shows that $W(P) \geq O(\epsilon)$.
Further, the equidistribution property and a simple exponential sum estimate help us in proving that the measure $W(a \cdot b)$ is exponentially small for any balanced product polynomial. Conceptually, this step is a transfer of small discrepancy of $\mathrm{MOD}_3 \circ \mathrm{XOR}$ function w.r.t the combinatorial rectangles to the product polynomials. Since the measure of $W(a \cdot b)$ is exponentially small, the sub-additive property of W shows that the number of product polynomials needed to account for $W(P)$ must be large (for a suitable range of values for ϵ). Finally, the lower bound follows from the structure theorem of monotone circuits which says that if P is computable by a polynomial size monotone circuit, then P can be written as a small sum of balanced product polynomials.

Organization

In Section 2, we recall basic facts about set-multilinear polynomials, monotone circuits, and relevant notions of corruption and discrepancy from communication complexity. In Section 3, we establish the key equidistribution property of parity vectors that makes possible the transfer of ideas from communication complexity to the arithmetic setting. In Section 4, we establish a general transfer theorem from rectangular corruption bounds to analogous bounds for product polynomials. Then, in Section 5.1, we exhibit a δ-non-sink polynomial that is computed efficiently by depth-3 circuits. We complement this in Section 5.2, applying the transfer theorem from Section 4 , to show that every δ-non-sink polynomial has exponentially large monotone complexity. Finally, in Section 6, we prove our ϵ-sensitive monotone lower bounds making progress along Hrubeš' approach.

2 Preliminaries

Notation.

Let $[n] = \{1, 2, \ldots, n\}$. For a polynomial $p \in \mathbb{R}[X]$ and a monomial κ, let $p[\kappa]$ be the coefficient of κ in p. For polynomials $p, q \in \mathbb{R}[X]$, we write $p \leq q$ if for each κ, we have that $p[\kappa] \leq q[\kappa]$. For a polynomial p, let $\mathrm{var}(p)$ denote the set of variables in p. For a vector $u \in \{0,1\}^n$, the notation $\mathrm{wt}(u)$ is used for the Hamming weight of u.

Set-multilinear Polynomials.

Let $X = \cup_{i=1}^{n} X_i$ be a set of variables where $X_i = \{x_{i,1}, x_{i,2}, \ldots, x_{i,m}\}$. A polynomial $p \in \mathbb{R}[X]$ is set-multilinear if each monomial in p respects the partition given by the set of variables $X_1, X_2, \ldots, X_n$. In other words, each monomial κ in p is of the form $x_{1,j_1} x_{2,j_2} \cdots x_{n,j_n}$. One can naturally associate a function (mapping) $\sigma : [n] \to [m]$ such that $\sigma(i) = j_i$. This association forms a bijection between the space of all functions from $[n]$ to $[m]$, denoted by $\mathcal{F}_{n,m}$ and the space of all such set-multilinear monomials. The cardinality of each set is easily seen to be m^n. Often we shall abuse notation to identify the monomial κ with the function it represents.

Parity Vectors of Set-Multilinear Monomials.

For a set-multilinear monomial κ with the associated function σ we define the parity vector $\vec{\oplus}(\kappa) \in \{0,1\}^m$ where the j^{th} entry is $\vec{\oplus}(\kappa)_j = |\sigma^{-1}(j)| \pmod 2$. For a set of parity vectors $S \subseteq \{0,1\}^m$ we shall denote

$$\mathcal{K}(S) = \{\kappa \mid \vec{\oplus}(\kappa) \in S\}.$$

Ordered Polynomial.

For a monomial of the form $\kappa = x_{i_1,j_1}x_{i_2,j_2}\cdots x_{i_n,j_n}$ we define the set $I(\kappa) = \{i_1, i_2, \ldots, i_n\}$. If a polynomial p has the same set $I(\kappa)$ for every monomial occurring in it with a non-zero coefficient, then we say that the polynomial is ordered and we write $I(p) = I(\kappa)$ for each κ. Clearly, the set-multilinear polynomials are ordered polynomials with $I(p) = \{1, 2, \ldots, n\}$.

Structure of Monotone Circuits.

The main structural result for monotone circuits that we use throughout, is the following theorem.

Theorem 2.1. *[Yeh19, Lemma 1] Let $n > 2$ and $p \in \mathbb{R}[X]$ be an ordered monotone polynomial with $I(p) = [n]$. Let C be a monotone circuit of size s that computes p. Then, we can write*

$$p = \sum_{t=1}^{s} a_t \cdot b_t$$

where a_t and b_t are monotone ordered polynomials with $\frac{n}{3} \leq |I(a_t)| \leq \frac{2n}{3}$ and $I(b_t) = I(a_t)\backslash[n]$. Moreover, $a_t \cdot b_t \leq p$ for each $1 \leq t \leq s$.

Such ordered product polynomials $a \cdot b$ with $\frac{n}{3} \leq |I(a)| \leq \frac{2n}{3}$ and $I(b) = [n] \setminus I(a)$ will be called balanced product polynomials.

Rectangular Corruption.

We recall here the concept of corruption measure from communication complexity that we make use of in this work. To do so, let us very briefly first recall the basic notions in the 2-party communication model of Yao. The joint input space of Alice and Bob is $\{0,1\}^m \times \{0,1\}^m$ with each player receiving an m-bit Boolean string, and they want to evaluate a Boolean function $F : \{0,1\}^m \times \{0,1\}^m \to \{0,1\}$. One defines a combinatorial rectangle R as a product set $A \times B$, for some $A, B \subseteq \{0,1\}^m$. Put another way, R is just a submatrix of the $2^m \times 2^m$ communication matrix M_F of the function F, that Alice and Bob want to compute. The rows of this matrix are indexed by possible inputs of Alice and the columns by the ones of Bob and $M_F(x,y) = F(x,y)$. To define it, consider a probability distribution λ on $\{0,1\}^m \times \{0,1\}^m$ that is almost balanced. Then, the intuition is that λ renders F hard, if rectangles (sub-matrices) of F (M_F) cannot even be *approximately monochromatic* unless they are small as measured by λ. We will use the following notion of corruption to measure approximate monochromaticity. This was implicitly defined by Razborov [Raz92].
Let $z \in \{0,1\}$. Then, the ϵ, *z-corruption* of F w.r.t. λ is denoted by $\mathrm{Corr}^z_{\lambda,\epsilon}(F)$, which is defined as follows:

$$\text{Corr}^z_{\lambda,\epsilon}(F) := \min_{R:\lambda(R\cap F^{-1}(z))\leq\epsilon\lambda(R)} \log\Big(\frac{1}{\lambda(R)}\Big).$$

The importance of the measure above lies by the fact that several lower bounds on the randomized communication complexity of functions, beginning with the famous one for Set-Disjointness, are proved by the simple relation: $R_\epsilon(F) \geq \Omega(\text{Corr}^z_{\lambda,\epsilon}(F))$, for each $z \in \{0,1\}$, where $R_\epsilon(F)$ is the ϵ-error randomized communication complexity of F.

Small Bias Spaces.

We recall the well-known notion of ϵ-biased spaces.

Definition 2.1 (ϵ-biased space)**.** *A multi-set $\mathcal{B} \subseteq \{-1,+1\}^m$ of size N is called an ϵ-biased space if for every subset $S \subseteq [m]$ we have*

$$|\frac{1}{N}\sum_{b\in\mathcal{B}}\prod_{i\in S} b_i| < \epsilon.$$

Recently, a breakthrough work of Ta-Shma [Ta-17] gives near optimal size explicit construction of ϵ-biased spaces.

Theorem 2.2 (Explicit construction of ϵ-bias spaces [Ta-17])**.** *There is a deterministic algorithm that for every $\epsilon > 0$ constructs an ϵ-biased space $\mathcal{B}_{m,\epsilon}$ of size $O(\frac{m}{\epsilon^{2+o(1)}})$. The algorithm runs in time* poly$(m, \frac{1}{\epsilon})$.

3 Equidistribution of Parity Vectors

In this section we record a few combinatorial results which will be used throughout the paper. The omitted proofs are presented in Appendix A. We establish an approximate equidistribution property of $\mathcal{F}_{n,m}$ which is the set of functions from $[n]$ to $[m]$. For a function $\sigma \in \mathcal{F}_{n,m}$ we define the parity vector $\vec{\oplus}(\sigma) \in \{0,1\}^m$ where the j^{th} entry is $\vec{\oplus}(\sigma)_j = |\sigma^{-1}(j)| \pmod 2$. For a vector $v \in \{0,1\}^m$ we define

$$\mathcal{F}^v_{n,m} = \{\sigma \mid \sigma \in \mathcal{F}_{n,m} \text{ such that } \vec{\oplus}(\sigma) = v\ \}.$$

We will show that $\mathcal{F}_{n,m}$ is partitioned into *approximately* equal size classes $\mathcal{F}^v_{n,m}$ (where $v \in \{0,1\}^m$) in the sense of Corollary 3.1. The following fact is easy to verify using a symmetry argument.

Fact 3.1. $|\mathcal{F}^v_{n,m}| = |\mathcal{F}^u_{n,m}|$ *if* wt(u) = wt(v).

The following results can be obtained via simple recurrences involving $|\mathcal{F}^v_{n,m}|$. The proofs are presented in Appendix A.

Claim 3.1. $|\mathcal{F}^v_{n,m}| \geq |\mathcal{F}^u_{n,m}|$ *when* $wt(v) = wt(u) - 2$.

When n is even we shall denote $\mathcal{F}^{(0,0,\ldots,0,0)}_{n,m}$ as $\mathcal{EF}_{n,m}$ and call it the set of even functions. Next we derive an upper bound on the number of even functions.

Lemma 3.1.

$$|\mathcal{EF}_{n,m}| \leq \frac{1}{2^{m-1}} \left(\sum_{i=0}^{\frac{m-2}{2}} \binom{m}{i} (m-2i)^n \right) \textit{ when } m \textit{ is even}$$
$$\leq \frac{1}{2^{m-1}} \left(\sum_{i=0}^{\frac{m-1}{2}} \binom{m}{i} (m-2i)^n \right) \textit{ when } m \textit{ is odd.}$$

From the above bounds we derive the main equidistribution property which will be used repeatedly.

Corollary 3.1 (Key Equidistribution Property)**.** *Let n be even. If $m \ln m \leq n$, then for every $v \in \{0,1\}^m$ we have,*

$$|\mathcal{F}^v_{n,m}| \leq \frac{4m^n}{2^m}.$$

Henceforth, in the rest of this paper, without loss of generality, we assume that n is even.

4 Corruption Transfer from Rectangles to Product Polynomials

Let $f : \{0,1\}^m \mapsto \{0,1\}$ be a boolean function and let $F = f \circ \text{XOR}$ be the boolean function defined on $\{0,1\}^{2m}$ as $F(x,y) = f(x \oplus y)$. Suppose F has high *corruption* with respect to rectangles i.e. $\text{Corr}^z_{\lambda,\nu}(F) \geq \log\left(\frac{1}{T}\right)$, or in other words:

$$\begin{aligned} \text{if} \quad & \lambda(R \cap F^{-1}(z)) \leq \nu\lambda(R) \\ \text{then} \quad & \lambda(R) \leq T \end{aligned} \tag{3}$$

where λ is the distribution on $\{0,1\}^{2m}$ defined as $\lambda(x,y) = \frac{1}{2^m}\mu(x \oplus y)$ using a distribution μ on $\{0,1\}^m$, and $0 \leq \nu \leq 1$ is some constant.
Using the distribution μ we define a measure on ordered polynomials. We first define W for an ordered monomial κ and then we extend it linearly to all ordered polynomials:

$$W(\kappa) = \mu(\vec{\oplus}(\kappa)) \cdot \frac{2^m}{m^n}.$$

Analogous to the boolean setting we define a concept of corruption for product polynomials. Formally we define

$$\text{MCorr}^z_{W,\gamma}(f) \coloneqq \min_{\substack{\alpha\cdot\beta \text{ : balanced} \\ \|\alpha\|_\infty, \|\beta\|_\infty \leq 1 \\ W(\alpha\cdot\beta \cap \mathcal{K}(f^{-1}(z))) \leq \gamma W(\alpha\cdot\beta)}} \log\left(\frac{1}{W(\alpha\cdot\beta)}\right).$$

Now we shall prove that the corruption of product polynomials is high.

Lemma 4.1 (Corruption Transfer)**.**

$$\textit{MCorr}^z_{W,\frac{\nu}{3}}(f) \geq \min\{\textit{Corr}^z_{\lambda,\nu}(f \circ \text{XOR}), m\} - \log_2 48.$$

In other words if a balanced product polynomial $H = \alpha \cdot \beta$ with $\|\alpha\|_\infty, \|\beta\|_\infty \leq 1$ satisfies

$$W(H \cap \mathcal{K}(f^{-1}(z))) \leq \frac{\nu}{3} W(H) \tag{4}$$

then we have

$$W(H) \leq 48 \cdot 2^{-\min\{Corr^z_{\lambda,\nu}(f \circ \mathrm{XOR}),m\}}.$$

Proof. Let $H = \alpha \cdot \beta$ be a product polynomial whose coefficients are at most 1. We define $\widetilde{\alpha}, \widetilde{\beta} \in \mathbb{R}^{2^m}$ as well by assigning for each $u \in \{0,1\}^m$

$$\widetilde{\alpha}[u] = \sum_{\kappa \mid \vec{\oplus}(\kappa)=u} \alpha[\kappa]$$

and

$$\widetilde{\beta}[u] = \sum_{\kappa \mid \vec{\oplus}(\kappa)=u} \beta[\kappa].$$

By definition of product polynomials we have

$$\begin{aligned}
\alpha \cdot \beta &= \left(\sum_{u \in \{0,1\}^m} \sum_{\kappa \mid \vec{\oplus}(\kappa)=u} \alpha[\kappa] \cdot \kappa \right) \cdot \left(\sum_{v \in \{0,1\}^m} \sum_{\kappa' \mid \vec{\oplus}(\kappa')=v} \beta[\kappa'] \cdot \kappa' \right) \\
&= \sum_{x \in \{0,1\}^m} \left(\sum_{u \in \{0,1\}^m} \left(\sum_{\kappa \,:\, \vec{\oplus}(\kappa)=u} \alpha[\kappa] \cdot \kappa \right) \cdot \left(\sum_{\kappa' \,:\, \vec{\oplus}(\kappa')=u \oplus x} \beta[\kappa'] \cdot \kappa' \right) \right).
\end{aligned}$$

Now applying W on both sides using linearity we have

$$\begin{aligned}
W(\alpha \cdot \beta) &= \sum_{x \in \{0,1\}^m} W(x) \cdot \left(\sum_{u \in \{0,1\}^m} \left(\sum_{\kappa \,:\, \vec{\oplus}(\kappa)=u} \alpha[\kappa] \right) \cdot \left(\sum_{\kappa' \,:\, \vec{\oplus}(\kappa')=u \oplus x} \beta[\kappa'] \right) \right) \\
&= \sum_{x \in \{0,1\}^m} W(x) \cdot \left(\sum_{u \in \{0,1\}^m} \widetilde{\alpha}[u] \cdot \widetilde{\beta}[u \oplus x] \right),
\end{aligned}$$

where by abuse of notation we denote $W(x) = W(\kappa)$ where κ is some monomial with $\vec{\oplus}(\kappa) = x$. For ease of writing we denote,

$$\widetilde{W}(\widetilde{\alpha}, \widetilde{\beta}) \coloneqq \sum_{x \in \{0,1\}^m} W(x) \cdot \left(\sum_{u \in \{0,1\}^m} \widetilde{\alpha}[u] \cdot \widetilde{\beta}[u \oplus x] \right) = W(\alpha \cdot \beta).$$

Similarly we have,

$$\widetilde{W_z}(\widetilde{\alpha}, \widetilde{\beta}) \coloneqq \sum_{\substack{x \in \{0,1\}^m \\ f(x)=z}} W(x) \cdot \left(\sum_{u \in \{0,1\}^m} \widetilde{\alpha}[u] \cdot \widetilde{\beta}[u \oplus x] \right) = W(\alpha \cdot \beta \cap \mathcal{K}(f^{-1}(z))).$$

Now we construct an optimization problem with $\gamma = \frac{\nu}{3}$

Program A

Variables:	$\widetilde{\alpha}[v], \widetilde{\beta}[v] \ : \ v \in \{0,1\}^m$
Objective:	$\max \widetilde{W}(\widetilde{\alpha}, \widetilde{\beta})$
Constraints:	$0 \leq \widetilde{\alpha}[v] \leq \lvert\mathcal{F}^v_{I(\alpha),m}\rvert$
	$0 \leq \widetilde{\beta}[v] \leq \lvert\mathcal{F}^v_{I(\beta),m}\rvert$
	$\widetilde{W_z}(\widetilde{\alpha}, \widetilde{\beta}) \leq \gamma \cdot \widetilde{W}(\widetilde{\alpha}, \widetilde{\beta})$

Let OPT1 be the optimum value of the optimization program A. Clearly, the Lemma will be proved by establishing the required upper bound on OPT1. This is the aim of the remaining part of the argument. First, we define a relaxation of the above optimization problem.

Program B

Variables:	$\widetilde{\alpha}[v], \widetilde{\beta}[v] \ : \ v \in \{0,1\}^m$
Objective:	$\max \widetilde{W}(\widetilde{\alpha}, \widetilde{\beta})$
Constraints:	$0 \leq \widetilde{\alpha}[v] \leq \dfrac{4m^{\lvert I(\alpha)\rvert}}{2^m}$
	$0 \leq \widetilde{\beta}[v] \leq \dfrac{4m^{\lvert I(\beta)\rvert}}{2^m}$
	$\widetilde{W_z}(\widetilde{\alpha}, \widetilde{\beta}) \leq \gamma \cdot \widetilde{W}(\widetilde{\alpha}, \widetilde{\beta})$

Let OPT2 be the optimum value of the optimization program B. Since we have $m \ln m \leq \frac{n}{4}$, Corollary 3.1 tell us that the second optimization problem is indeed a relaxation of the first[2], and hence OPT1 $\leq$ OPT2. The goal in the next few steps is to extract a combinatorial rectangle R from $\widetilde{\alpha}$ and $\widetilde{\beta}$ such that OPT2 is upper bounded by $O(\lambda(R))$. Then, applying the corruption bound on R we will get our desired bound on OPT2. To do this, it will be convenient to understand a simple structure of an optimal solution to Program B.

Let $(\widehat{\alpha}, \widehat{\beta})$ be an optimum solution to the optimization Program B. We obtain a linear program on the variables $\tilde{\alpha}$ from Program B by fixing the values $\widetilde{\beta} = \widehat{\beta}$. The constraints of this LP define a polytope and let θ be a corner point. Then, θ has the property that for at most one coordinate $\widetilde{u}$, that we call exceptional, we have $\theta[\widetilde{u}] \notin \{0, \frac{4m^{|I(\widetilde{\alpha})|}}{2^m}\}$. For every other coordinate $u \neq \widetilde{u}$ we have $\theta[u] \in \{0, \frac{4m^{|I(\alpha)|}}{2^m}\}$. Hence, there exists an optimum solution at one of the corner points, denoted by α^*. Clearly, $W(\alpha^*, \widehat{\beta}) = W(\widehat{\alpha}, \widehat{\beta})$. Again fixing $\widetilde{\alpha} = \alpha^*$ in Program B we get a linear program on the variables $\widetilde{\beta}$. We can get an optimum solution β^* at a corner point of the polytope defined by the constraints on the β variables. This gives us a corner point solution (α^*, β^*) (with exceptional coordinates u^*, v^*) which achieves the optimum.

Now for such a product polynomial we define a rectangle $R = A \times B$ where

$$A = \{u \mid \alpha^*[u] \neq 0\} \setminus \{u^*\}.$$

$$B = \{v \mid \beta^*[v] \neq 0\} \setminus \{v^*\}.$$

[2]This is the only place that we make use of the fact that H is a balanced product polynomial, i.e. $\frac{n}{3} \leq |I(\alpha)|, |I(\beta)| \leq \frac{2n}{3}$.

From the definition we have

$$\widetilde{W}(\alpha^*, \beta^*) = \sum_{x \in \{0,1\}^m} W(x) \cdot \left(\sum_{u \in \{0,1\}^m} \alpha^*[u] \cdot \beta^*[u \oplus x] \right).$$

Now interchanging the order of summation we have,

$$\widetilde{W}(\alpha^*, \beta^*) = \sum_{u \in A \cup \{u^*\}} \alpha^*[u] \sum_{\substack{x \in \{0,1\}^m \\ u \oplus x \in B \cup \{v^*\}}} \beta^*[u \oplus x] \cdot W(x).$$

Renaming $u \oplus x$ as v,

$$\widetilde{W}(\alpha^*, \beta^*) \leq \sum_{u \in A, v \in B} \alpha^*[u] \cdot \beta^*[v] \cdot W(u \oplus v) + \sum_{v \in \{0,1\}^m} \alpha^*[u^*] \cdot \beta^*[v] \cdot W(u^* \oplus v) + \sum_{u \in \{0,1\}^m} \alpha^*[u] \cdot \beta^*[v^*] \cdot W(u \oplus v^*).$$

For ease of notation we define $\widehat{W}(\alpha^*, \beta^*) = \sum_{u \in A, v \in B} \alpha^*[u] \cdot \beta^*[v] \cdot W(u \oplus v)$ and $W'(\alpha^*, \beta^*) = \widetilde{W}(\alpha^*, \beta^*) - \widehat{W}(\alpha^*, \beta^*)$.
We establish an upper bound on $\widetilde{W}(\alpha^*, \beta^*)$ in terms of $\lambda(R)$ via parts 1 and 2 of the following claims which gives upper bounds on $\widehat{W}(\alpha^*, \beta^*)$ and $W'(\alpha^*, \beta^*)$.

Claim 4.1.

1. $\widehat{W}(\alpha^*, \beta^*) = 16\lambda(R)$.
2. $W'(\alpha^*, \beta^*) \leq \frac{32}{2^m}$.
3. $\widetilde{W}_z(\alpha^*, \beta^*) \geq 16\lambda(R \cap F^{-1}(z))$.

First we complete the proof of Lemma 4.1 assuming the above claim. Combining parts 1 and 2 of Claim 4.1 we conclude that

$$\widetilde{W}(\alpha^*, \beta^*) \leq 16\lambda(R) + 32 \cdot 2^{-m}.$$

If $\lambda(R) \leq 2^{-m}$ then $\widetilde{W}(\alpha^*, \beta^*) \leq 48 \cdot 2^{-m}$ and then we are done. So we may assume $\lambda(R) > 2^{-m}$ and hence we obtain the upper bound

$$\widetilde{W}(\alpha^*, \beta^*) \leq 16\lambda(R) + 32 \cdot 2^{-m} \leq 16\lambda(R) + 32\lambda(R) = 48\lambda(R). \tag{5}$$

Finally we have,

$$16\lambda(R \cap F^{-1}(z)) \underset{\text{Claim 4.1Part 3}}{\leq} \widetilde{W}_z(\alpha^* \cdot \beta^*) \underset{\text{constraint}}{\leq} \gamma \cdot \widetilde{W}(\alpha^* \cdot \beta^*) \underset{\text{Equation 5}}{\leq} 48\gamma \cdot \lambda(R).$$

Since $\gamma \leq \frac{\nu}{3}$, we note that the rectangle R satisfies

$$\lambda(R \cap F^{-1}(z)) \leq \nu\lambda(R).$$

Therefore we have,

$$\text{OPT1} \leq \text{OPT2} \underset{\text{Equation 5}}{\leq} 48\lambda(R) \underset{\text{Equation 3}}{\leq} 48 \cdot 2^{-\min\{\text{Corr}^{z}_{\lambda,\nu}(F),m\}}.$$

This yields the bound for all balanced product polynomials satisfying Equation 4 and completes the proof of Lemma 4.1.
All that remains is to prove Claim 4.1 which we do next.

Proof of Claim 4.1. For the first part we have,

$$\begin{aligned}
\widehat{W}(\alpha^*, \beta^*) &= \sum_{u \in A, v \in B} \alpha^*[u] \cdot \beta^*[v] \cdot W(u \oplus v) \\
&= \sum_{u \in A, v \in B} \frac{4m^{|I(\alpha)|}}{2^m} \cdot \frac{4m^{|I(\beta)|}}{2^m} \cdot W(u \oplus v) \\
&= \sum_{u \in A, v \in B} \frac{16m^n}{2^{2m}} \cdot \mu(u \oplus v) \cdot \frac{2^m}{m^n} \\
&= \sum_{u \in A, v \in B} \frac{16}{2^m} \cdot \mu(u \oplus v) \\
&= \sum_{u \in A, v \in B} 16\lambda(u, v) \\
&= 16\lambda(R).
\end{aligned} \tag{6}$$

For the second part we have,

$$\begin{aligned}
W'(\alpha^*, \beta^*) &\leq \sum_{v \in \{0,1\}^m} \alpha^*[u^*] \cdot W(u^* \oplus v)\beta^*[v] \cdot + \sum_{u \in \{0,1\}^m} \alpha^*[u] \cdot \beta^*[v^*] \cdot W(u \oplus v^*) \\
&\leq \sum_{v \in \{0,1\}^m} \frac{4m^{|I(\alpha)|}}{2^m} \cdot \frac{4m^{|I(\beta)|}}{2^m} \cdot W(u^* \oplus v) + \sum_{u \in \{0,1\}^m} \frac{4m^{|I(\alpha)|}}{2^m} \cdot \frac{4m^{|I(\beta)|}}{2^m} \cdot W(u \oplus v^*) \\
&= \sum_{v \in \{0,1\}^m} 16\frac{1}{2^m}\mu(u^* \oplus v) + \sum_{u \in \{0,1\}^m} 16\frac{1}{2^m}\mu(u \oplus v^*) \\
&= \frac{32}{2^m}.
\end{aligned}$$

In the last equality we have used the fact that μ is a probability measure on $\{0,1\}^m$.
For the third part we have,

$$\begin{aligned}
\widetilde{W_z}(\alpha^*, \beta^*) &\geq \sum_{\substack{u \in A, v \in B \\ F(u \oplus v) = z}} \alpha^*[u] \cdot \beta^*[v] \cdot W(u \oplus v) \\
&= \sum_{\substack{u \in A, v \in B \\ F(u \oplus v) = z}} 16\lambda(u, v) \\
&= 16\lambda(R \cap F^{-1}(z)).
\end{aligned}$$

□

□

We give a simple reformulation of our corruption bound[3] for product polynomials which will be useful later.

Corollary 4.1. *For every balanced product polynomial $H = \alpha \cdot \beta$ with $\|\alpha\|_\infty, \|\beta\|_\infty \leq 1$ we have for every $z \in \{0,1\}$*

$$W(\alpha \cdot \beta \cap \mathcal{K}(f^{-1}(z))) \geq \frac{\nu}{3} W(\alpha \cdot \beta) - 48 \cdot 2^{-\min\{Corr^z_{\lambda,\nu}(f \circ XOR), m\}}. \tag{7}$$

Proof. From Lemma 4.1 we know that if the product polynomial satisfies

$$W(\alpha \cdot \beta \cap \mathcal{K}(f^{-1}(z)) \leq \frac{\nu}{3} W(\alpha \cdot \beta).$$

then $W(\alpha \cdot \beta) \leq 48 \cdot 2^{-\min\{\mathrm{Corr}^z_{\lambda,\nu}(F), m\}}$, where $F = f \circ \mathrm{XOR}$ and thus the right hand side of Equation 7 would be negative and hence (7) would be true. Otherwise the product polynomial satisfies

$$W(\alpha \cdot \beta \cap \mathcal{K}(f^{-1}(z))) > \frac{\nu}{3} W(\alpha \cdot \beta).$$

and hence (7) is again true. □

5 Exponential Separation Between Depth-3 Formulas and Monotone VP

In this section, we prove Theorem 1.1. We show the construction of a polynomial-size depth three circuit for a δ-non-sink polynomial which embeds an approximation of the boolean function SINK $\circ$ XOR. Next, we use the corruption transfer result from Section 4 to show that δ-non-sink polynomials are hard for monotone circuits.

5.1 The construction of depth-3 formula.

For the sake of the reader, we recall some concepts from the introduction. For a function $\sigma \in \mathcal{F}_{n,m}$ when $m = \binom{k}{2}$, the parity vector $\vec{\oplus}(\sigma)$ can be interpreted as a tournament by orienting the edges of K_k according to $\vec{\oplus}(\sigma)$. More precisely, we fix a bijection $\phi : [\binom{k}{2}] \mapsto E(K_k)$ and interpret $\vec{\oplus}(\sigma)_j$ giving an orientation to the edge $\phi(j) = (x,u)$ where $x < u$. If $\vec{\oplus}(\sigma)_j = 1$ then we give the orientation $x \to u$ otherwise we give the orientation $u \to x$. Let $T(\vec{\oplus}(\sigma))$ be the tournament on k vertices obtained using the above process. A function is said to have a sink if the tournament $T(\vec{\oplus}(\sigma))$ has a sink. If $T(\vec{\oplus}(\sigma))$ has a sink u then we have

$$\vec{\oplus}(\sigma) = (*,*,*, \underbrace{1,1,\ldots,1}_{\substack{\phi(j)=(x,u) \\ \text{with } x<u}}, \underbrace{0,0,\ldots,0}_{\substack{\phi(j)=(u,x) \\ \text{with } x>u}}, *,*,*)$$

where the coordinates marked by $*$ could be either 0 or 1. For a vertex u, we define $\mathrm{sink}(u) \coloneqq \{\sigma \mid \vec{\oplus}(\sigma) \text{ has a sink at } u\}$.

[3]This form of the corruption bound appears in Razborov's [Raz92] argument for Set-Disjointness.

Proof of Theorem 1.1 Part (1): Let $X_i = \{x_{i,j}\}_{j=1}^m$ be a set of variables and let $X = \cup_{i=1}^n X_i$. For a vertex $u \in K_k$, consider the vector

$$\boldsymbol{u} = (*,*,*,\underbrace{1,1,\ldots,1}_{\substack{\phi(j)=(x,u)\\ \text{with } x<u}},\underbrace{0,0,\ldots,0}_{\substack{\phi(j)=(u,x)\\ \text{with } x>u}},*,*,*).$$

For a vertex $u \in K_k$ and a vector $b^{(r)} \in \{+1,-1\}^m$, we define a polynomial

$$Q_{u,r} \coloneqq \prod_{j|\boldsymbol{u}_j=1} b_j^{(r)} \cdot \prod_{i=1}^{n} \Big(\sum_{j|\boldsymbol{u}_j \neq *} b_j^{(r)} x_{i,j} + \sum_{j|\boldsymbol{u}_j=*} x_{i,j} \Big).$$

For $0 < \epsilon < 1$, let $\mathcal{B}_{m,\frac{\epsilon}{k}}$ be the ϵ/k-biased space obtained from Theorem 2.2 of size $N = O(\frac{m}{(\epsilon/k)^{2+o(1)}})$.
Using the $\frac{\epsilon}{k}$-biased space $\mathcal{B}_{m,\frac{\epsilon}{k}} = \{b^{(1)}, b^{(2)}, \ldots, b^{(N)}\}$, we define the following polynomial Q_u

$$Q_u \coloneqq \frac{1}{N} \cdot \sum_{r=1}^{N} Q_{u,r}.$$

Any monomial in $Q_{u,r}$ has the form $\kappa = \prod_{i=1}^n x_{i,\sigma(i)}$ for some function $\sigma : [n] \mapsto [m]$ and further,

$$Q_{u,r}[\kappa] = \prod_{j|\boldsymbol{u}_j=0} (b_j^{(r)})^{|\sigma^{-1}(j)|} \cdot \prod_{j|\boldsymbol{u}_j=1} (b_j^{(r)})^{|\sigma^{-1}(j)|+1}.$$

Let us note that when $\vec{\oplus}(\kappa) \in \text{sink}(u)$, we have $Q_{u,r}[\kappa] = 1$ and for any other monomial its coefficient depends on at least one coordinate of $b^{(r)}$. Since $b^{(1)}, b^{(2)}, \ldots, b^{(N)}$ form an $\frac{\epsilon}{k}$-bias space, we have $|Q_u[\kappa]| < \frac{\epsilon}{k}$ for any κ with $\vec{\oplus}(\kappa) \notin \text{sink}(u)$. Now we define the polynomial

$$Q_{\epsilon,n,m} \coloneqq \sum_{u \in [k]} Q_u.$$

Let us note that $Q_{\epsilon,n,m}[\kappa] \in (1-\epsilon, 1+\epsilon)$ if $\vec{\oplus}(\kappa)$ has a sink and $|Q_{\epsilon,n,m}[\kappa]| < \epsilon$ otherwise. We define another polynomial

$$H_{\epsilon,n,m} \coloneqq \frac{1}{1+2\epsilon} \left[(1+\epsilon) Q_{\text{All}} - Q_{\epsilon,n,m} \right],$$

where $Q_{\text{All}} = \prod_{i=1}^n (\sum_{j=1}^m x_{i,j})$. We observe that $H_{\epsilon,n,m}[\kappa] \in (0, \frac{2\epsilon}{1+2\epsilon})$ if $\vec{\oplus}(\kappa)$ has a sink and $H_{\epsilon,n,m}[\kappa] \in (\frac{1}{1+2\epsilon}, 1)$ otherwise.
By definition, the polynomial $H_{\epsilon,n,m}$ has a depth-three formula of size $O(kNnm)$ which is $O(nm^{3.5+o(1)}/\epsilon^{2+o(1)})$. Finally, we define the following polynomial

$$P_{\delta,n,m} \coloneqq H_{\frac{\delta}{2(1-\delta)},n,m}.$$

Notice that $P_{\delta,n,m}[\kappa] \in (0,\delta)$ if $\vec{\oplus}(\kappa)$ has a sink, and $P_{\delta,n,m}[\kappa] \in (1-\delta, 1)$ otherwise. Note that $P_{\delta,n,m}$ is a δ-non-sink Polynomial. □

5.2 The Lower Bound.

In this section, we establish the following theorem which is a restatement of part 2 of Theorem 1.1.

Theorem 5.1 (Restatement of Theorem 1.1, part 2)**.** *There exists a sufficiently small constant $\delta < 1$ such that for every δ-non-sink polynomial $Q_{\delta,n,m}$, the monotone circuit complexity of $Q_{\delta,n,m}$ is $2^{\Omega(\sqrt{m})}$, when $4m \ln m \leq n$.*

Since in the previous section, we have given an example of a δ-non-sink polynomial that can be computed by depth-3 general formula of small size, we get our required separation as claimed by Theorem 1.1.
In order to prove our theorem, we will make use of a rectangular corruption bound established by Chattopadhyay, Mande and Sherif [CMS20] for the Boolean function SINK $\circ$ XOR. More precisely, consider the SINK function associated with the complete graph K_k, with $m = \binom{k}{2}$. Define the following distribution μ on the space of inputs $\{0,1\}^m$ to SINK: toss a fair coin b. If $b = 1$, sample a vertex $i \in [k]$ at random, and then sample at random $x \in \{0,1\}^m$ from all inputs that make i a sink. If $b = 0$, sample x at random from $\{0,1\}^m$. Let $\lambda : \{0,1\}^m \times \{0,1\}^m \to [0,1]$ be the probability distribution given by $\lambda(x,y) = \frac{1}{2^m} \cdot \mu(x \oplus y)$. We collect some simple facts together:

Fact 5.1.

1. $\forall z \in \mathit{SINK}^{-1}(0) : \mu(z) = \frac{1}{2^{m+1}}$.
2. $\left|\mathit{SINK}^{-1}(1)\right| = k2^{m-k+1} = O(\sqrt{m}2^{m-\sqrt{m}})$.

The main corruption bound that we use is as follows:

Theorem 5.2 (Lemma 6.2 in[4] [CMS20])**.** *Let $0 \leq \nu \leq 1/2$ be any constant. Then,*

$$\mathit{Corr}^1_{\lambda,\nu}\Big(\mathit{SINK} \circ \mathit{XOR}\Big) = \Omega(\sqrt{m}).$$

Now we prove Theorem 5.1.

Proof. Let Q by any δ-non-sink polynomial. Let C be a monotone circuit of size s computing Q. Then by the structure Theorem 2.1, we may write Q as a sum of s many balanced product polynomials

$$Q = \sum_{t=1}^{s} \alpha_t \cdot \beta_t.$$

The general idea of our argument is as follows: given the hard distribution μ on SINK considered by [CMS20], we define a measure W on monomials as prescribed by the Transfer Lemma 4.1 established in the previous section. Combining Theorem 5.2 with the Corruption Transfer Lemma, we immediately obtain that every product polynomial $\alpha_t \cdot \beta_t$ either measures very little

[4]Please note that [CMS20] use the symbol m to represent the number of vertices in the complete graph whose edges represent the variables to SINK. Their m corresponds to our k. Further, their ν corresponds to our λ, and their value of $4 \cdot \epsilon$ corresponds to our ν.

w.r.t W or its contribution to the sink monomials, measured w.r.t W, is a significant fraction of the total measure $W(\alpha_t \cdot \beta_t)$. However, we show that the sink monomials of Q, weighted by their coefficients, measure up to a tiny fraction of the total measure $W(Q)$. These two opposing facts can be reconciled only if the number of product polynomials, s, in the decomposition of Q is exponentially large[5].

Forthwith the details: set measure W on monomials according to our prescription, i.e. for any monomial κ, $W(\kappa) \coloneqq \mu(\oplus(\kappa)) \cdot \frac{2^m}{m^n}$, where μ is the hard probability distribution on the inputs of SINK.

Transferring the rectangular corruption bound of Theorem 5.2 to W via Lemma 4.1 and then using the bound of Corollary 4.1, we observe that

$$\begin{aligned} W(Q \cap \mathcal{K}(\text{SINK}^{-1}(1))) &= \sum_{t=1}^{s} W(\alpha_t \cdot \beta_t \cap \mathcal{K}(\text{SINK}^{-1}(1))) \\ &\geq \sum_{t=1}^{s} \frac{\nu}{3} W(\alpha_t \cdot \beta_t) - 48 \cdot s \cdot 2^{-\text{Corr}^1_{\lambda,\nu}(\text{SINK} \circ \text{XOR})}. \end{aligned}$$

Hence,

$$48 \cdot s \cdot 2^{-\text{Corr}^1_{\lambda,\nu}(\text{SINK} \circ \text{XOR})} \geq \frac{\nu}{3} W(Q) - W\Big(Q \cap \mathcal{K}(\text{SINK}^{-1}(1))\Big). \tag{8}$$

Now we would like to prove a lower bound on the right hand side. In order to do so we first prove two claims.

Claim 5.1.

$$W\Big(Q \cap \mathcal{K}(\textit{SINK}^{-1}(1))\Big) \leq 4\delta.$$

Proof.

$$\begin{aligned} W\Big(Q \cap \mathcal{K}(\text{SINK}^{-1}(1))\Big) &\leq \delta \cdot \sum_{\substack{\kappa \\ \text{SINK}(\vec{\oplus}(\kappa))=1}} W(\kappa) \\ &\leq \delta \cdot \sum_{\substack{\kappa \\ \text{SINK}(\vec{\oplus}(\kappa))=1}} \mu(\vec{\oplus}(\kappa)) \cdot \frac{2^m}{m^n} \\ &= \delta \cdot \sum_{\substack{x \in \{0,1\}^m \\ \text{SINK}(x)=1}} \sum_{\substack{\kappa \\ \vec{\oplus}(\kappa)=x}} \mu(\vec{\oplus}(\kappa)) \cdot \frac{2^m}{m^n} \\ &\leq \delta \cdot \sum_{\substack{x \in \{0,1\}^m \\ \text{SINK}(x)=1}} \frac{4m^n}{2^m} \cdot \mu(x) \cdot \frac{2^m}{m^n} \\ &\leq 4\delta \cdot \sum_{\substack{x \in \{0,1\}^m \\ \text{SINK}(x)=1}} \mu(x) \\ &\leq 4\delta. \end{aligned}$$

where the last inequality follows because μ is a probability measure on $\{0,1\}^m$.

□

[5]While this is the well-known idea behind the formulation of the notion of corruption in communication complexity, we are not aware of its prior use in arithmetic complexity.

Claim 5.2.

$$W(Q) \geq \frac{1-\delta}{3}$$

for large m.

Proof. Using the fact that the coefficient of every non-sink monomial in Q is in the interval $[1-\delta, 1]$, and substituting f for SINK we get,

$$\begin{aligned}
W(Q) &\geq (1-\delta) W\Big(Q \cap \mathcal{K}(\text{SINK}^{-1}(0))\Big) \\
&= (1-\delta) \cdot \sum_{\substack{\kappa \\ f(\vec{\oplus}(\kappa))=0}} W(\kappa) \\
&= (1-\delta) \cdot \sum_{\substack{x \in \{0,1\}^m \\ f(x)=0}} \sum_{\substack{\kappa \\ \vec{\oplus}(\kappa)=x}} \mu(\vec{\oplus}(\kappa)) \cdot \frac{2^m}{m^n} \\
&= (1-\delta) \cdot \sum_{\substack{x \in \{0,1\}^m \\ f(x)=0}} \sum_{\substack{\kappa \\ \vec{\oplus}(\kappa)=x}} \frac{1}{2^{m+1}} \cdot \frac{2^m}{m^n} \\
&= (1-\delta) \cdot \frac{1}{2} \frac{|\mathcal{K}(f^{-1}(0))|}{m^n} \\
&= (1-\delta) \cdot \frac{1}{2} \left(1 - \frac{|\mathcal{K}(f^{-1}(1))|}{m^n}\right).
\end{aligned}$$

Now we may bound the total number of sink monomials, using Fact 5.1 and Corollary 3.1, as

$$|\mathcal{K}(f^{-1}(1))| \leq c \cdot \sqrt{m} 2^{m-\sqrt{m}} \cdot \frac{4m^n}{2^m} \leq c\sqrt{m} \cdot \frac{4m^n}{2^{\sqrt{m}}}.$$

for some constant c. Thus we have

$$\left(1 - \frac{|\mathcal{K}(f^{-1}(1))|}{m^n}\right) \geq (1 - \frac{4c\sqrt{m}}{2^{\sqrt{m}}}) \geq \frac{2}{3},$$

for large m. Finally we conclude that

$$W(Q) \geq (1-\delta) \cdot \frac{1}{2} \left(1 - \frac{|\mathcal{K}(f^{-1}(1))|}{m^n}\right) \geq \frac{1-\delta}{3}.$$

for large m. □

Continuing from Equation 8 and applying Claims 5.1, 5.2 we have

$$48 \cdot s \cdot 2^{-\text{Corr}^1_{\lambda,\nu}(\text{SINK} \circ \text{XOR})} \geq \frac{\nu}{3} W(Q) - W\Big(Q \cap \mathcal{K}(\text{SINK}^{-1}(1))\Big) \geq \frac{\nu}{3} \frac{1-\delta}{3} - 4\delta.$$

Now since δ is small enough we may write $\frac{\nu}{3}\frac{1-\delta}{3} - 4\delta \geq \delta$ and hence

$$s \geq \frac{\delta}{48} \cdot 2^{\text{Corr}^1_{\lambda,\nu}(\text{SINK} \circ \text{XOR})} \geq 2^{\Omega(\sqrt{m})}.$$

□

6 ϵ-Sensitive Monotone Lower Bound for MOD3 $\circ$ MOD2 Polynomial

In this section, we prove Theorem 1.2. Consider the boolean function defined on $\{0,1\}^m$ as $f(x) = 0$ if $\text{wt}(x) \equiv 0 \pmod 3$ and $f(x) = 1$ otherwise. This is often called the MOD_3 Boolean function. We define a polynomial $P_{n,m}$ that remarkably remains hard even when it is added to the full set-multilinear polynomial after being multiplied by a tiny number ϵ. The question of proving such lower bounds against monotone circuits was raised in the recent work of Hrubeš[Hru20] where he gives an alternative approach for attacking general (non-monotone) circuits. More precisely, Hrubeš shows that if one could prove strong lower bounds for arbitrary small, but non-zero ϵ, then they imply comparable bounds for general circuits. Our lower bound works as long as $\epsilon \geq 2^{-\gamma n/\log n}$ for some constant γ.

Candidate polynomial.

First, define the following polynomial.

$$P_{n,m}^{\text{MOD}_3} := \sum_{\substack{\sigma:[n]\mapsto[m] \\ \text{MOD}_3(\vec{\oplus}(\sigma))=0}} \prod_{i=1}^{n} x_{i,\sigma(i)}.$$

It is trivial to see that there is a polynomial time algorithm that given a monomial decides whether the coefficient of the monomial is zero or one in $P_{n,m}^{\text{MOD}_3}$. Hence, by Valiant's criterion [Val79, Proposition 4], the polynomial $P_{n,m}^{\text{MOD}_3}$ is in VNP. We show a monotone circuit lower bound for

$$P = F_{n,m} - \epsilon \cdot P_{n,m}^{\text{MOD}_3}.$$

for some $\epsilon > 0$, where we define the full polynomial as $F_{n,m} = \prod_{i=1}^{n}(\sum_{j=1}^{m} x_{i,j})$. The lower bound proof for

$$P = F_{n,m} + \epsilon \cdot P_{n,m}^{\text{MOD}_3}.$$

is analogous. Since the polynomial $F_{n,m}$ can be computed by a polynomial-size monotone circuit, the lower bound result for $P = F_{n,m} + \epsilon \cdot P_{n,m}^{\text{MOD}_3}$ also shows that the polynomial $P_{n,m}^{\text{MOD}_3}$ needs exponential-size monotone circuit.
Now we are ready to prove the main result under the condition $4m \ln m \leq n$.

Proof of Theorem 1.2.
We define two measures W_0 and W_1. For a monomial κ of the form $\prod_{i=1}^{n} x_{i,\sigma(i)}$ we define

$$\begin{aligned} W_0(\kappa) &= \frac{1}{m^n} && \text{if } \text{wt}(\vec{\oplus}(\kappa)) \equiv 0 \pmod 3 \\ &= 0 && \text{otherwise.} \end{aligned}$$

and similarly,

$$\begin{aligned} W_1(\kappa) &= \frac{1}{m^n} && \text{if } \text{wt}(\vec{\oplus}(\kappa)) \equiv 1 \pmod 3 \\ &= 0 && \text{otherwise.} \end{aligned}$$

We linearly extend these measures to all polynomials and define our main measure

$$W(P) = W_1(P) - W_0(P).$$

Our main result will follow immediately from the following two claims: the main technical result of this section will show that the measure W is exponentially small on product polynomials.

Lemma 6.1 (Measure is small for balanced product polynomials)**.** *Let $a \cdot b$ be a balanced product polynomial whose coefficients are at most 1. Then*

$$|W(a \cdot b)| \leq \frac{64}{3}\left(\sqrt{\frac{3}{4}}\right)^m,$$

when $4m \ln m \leq n$.

However, the following claim shows that $W(P)$ is large.

Claim 6.1. *There exists a constant $\gamma_0 < 1$, such that if $\epsilon \geq 2^{-\gamma_0 \cdot m}$, then*

$$W(P) \geq \frac{\epsilon}{5},$$

when $4m \ln m \leq n$.

Given these two results, the structure theorem readily yields the main result as shown by the following short argument. Suppose that P has a monotone circuit of size s. Then by the structure Theorem 2.1, we can write

$$P = \sum_{t=1}^{s} a_t \cdot b_t$$

where a_t, b_t are balanced product polynomials.
Thus,

$$\frac{\epsilon}{5} \underset{\text{Claim 6.1}}{\leq} W(P) = \sum_{t=1}^{s} W(a_t \cdot b_t) \underset{\text{Lemma 6.1}}{\leq} s \cdot \frac{64}{3}\left(\sqrt{\frac{3}{4}}\right)^m.$$

Hence, $s \geq \frac{3\epsilon}{320}\left(\sqrt{\frac{4}{3}}\right)^m$. To get the required lower bound for s, we must have $\epsilon \geq 2^{-\gamma_1 m}$ for some $\gamma_1 > 0$. Claim 6.1 needs $\epsilon \geq 2^{-\gamma_0 m}$. Hence, choose $\epsilon \geq 2^{-\min\{\gamma_0, \gamma_1\} m}$. The result now immediately follows.
All that remains is to establish Lemma 6.1 and Claim 6.1. The latter follows by a short calculation from the former and so we first prove it below.

Proof of Claim 6.1. Recall the following notation

$$N_b = |\{\kappa \mid \text{wt}(\vec{\oplus}(\kappa)) \equiv b \pmod 3\}| \qquad \forall\, b \in \{0, 1, 2\}$$

We explicitly compute the measure for our target polynomial using linearity.

$$W(P) = W(F_{n,m}) - \epsilon \cdot W(P_{n,m}).$$

For the full polynomial one can easily compute $W(F_{n,m}) = \frac{N_1}{m^n} - \frac{N_0}{m^n}$ and clearly $W(P_{n,m}) = -W_0(P_{n,m}) = -\frac{N_0}{m^n}$. Thus, we have

$$W(P) = \frac{N_1}{m^n} - \frac{N_0}{m^n} + \epsilon \cdot \frac{N_0}{m^n}.$$

and hence $|W(P)| \geq \epsilon \cdot |\frac{N_0}{m^n}| - |\frac{N_0}{m^n} - \frac{N_1}{m^n}|$. Since $F_{n,m}$ is a product polynomial, using Lemma 6.1 we conclude that $W(F_{n,m}) = |\frac{N_1}{m^n} - \frac{N_0}{m^n}| \leq \frac{64}{3} \cdot \left(\sqrt{\frac{3}{4}}\right)^m$.

By a similar calculation one can show that $|\frac{N_0}{m^n} - \frac{N_2}{m^n}|, |\frac{N_1}{m^n} - \frac{N_2}{m^n}| \leq 64 \cdot \left(\sqrt{\frac{3}{4}}\right)^m$ and since $\frac{N_0}{m^n} + \frac{N_1}{m^n} + \frac{N_2}{m^n} = 1$ it must be the case that $|\frac{N_0}{m^n} - \frac{1}{3}| \leq 2^{-\Omega(m)}$ and hence we may write $\frac{N_0}{m^n} \geq \frac{1}{4}$. Finally we have

$$|W(P)| \geq \epsilon \cdot |\frac{N_0}{m^n}| - |\frac{N_0}{m^n} - \frac{N_1}{m^n}| \geq \frac{\epsilon}{5}$$

which is true if the parameter ϵ satisfies the following condition

$$\frac{\epsilon}{20} \geq \frac{64}{3}\left(\sqrt{\frac{3}{4}}\right)^m.$$

Thus we can choose the parameter γ_0 appropriately such that $\epsilon \geq 2^{-\gamma_0 n/\log n}$. □

All that remains to finish this section is to argue for the correctness of Lemma 6.1. We do so by an argument that is inspired by discrepancy estimation techniques used in communication complexity, especially of the kind that appeared in Ada et.al. [ACFN15].

Proof of Lemma 6.1. Given any product polynomial $a \cdot b$, we define vectors $A, B \in \mathbb{R}^{2^m}$ below, where $v \in \{0,1\}^m$ is an arbitrary parity vector.

$$A[v] \coloneqq \frac{|\{\kappa | \vec{\oplus}(\kappa) = v \text{ and } a[\kappa] \neq 0\}|}{m^{|I(a)|}}.$$

$$B[v] \coloneqq \frac{|\{\kappa | \vec{\oplus}(\kappa) = v \text{ and } b[\kappa] \neq 0\}|}{m^{|I(b)|}}.$$

We also define as

$$\vec{\oplus}(a) \coloneqq \{\vec{\oplus}(\kappa) \mid a[\kappa] \neq 0\},$$
$$\vec{\oplus}(b) \coloneqq \{\vec{\oplus}(\kappa) \mid b[\kappa] \neq 0\}.$$

Further define vectors $\alpha, \beta \in \mathbb{R}^{2^m}$ where $\alpha[v] \coloneqq 2^m \cdot A[v]$ (similarly $\beta[v] \coloneqq 2^m \cdot B[v]$). Assuming that n is even and using Corollary 3.1, we conclude that $\alpha[v], \beta[v] \leq 4$.
We write the measures for a product polynomial $a \cdot b$ as

$$W_0(a \cdot b) = \sum_{u,v \in \{0,1\}^m} A[u] \cdot B[v] \cdot \frac{1}{3}(1 + \omega^{|u \oplus v|} + \omega^{2|u \oplus v|}),$$

and

$$W_1(a \cdot b) = \sum_{u,v \in \{0,1\}^m} A[u] \cdot B[v] \cdot \frac{1}{3}(1 + \omega^{|u \oplus v|+2} + \omega^{2|u \oplus v|+1}),$$

where $|u \oplus v| = (\sum_i u_i + \sum_i v_i - 2\sum_i u_i v_i)$, and ω is the complex third root of unity. Note that we have made use of the fact that $(1 + \omega + \omega^2) = 0$.
We write

$$\begin{aligned}
3 \cdot |W(a \cdot b)| = | &\sum_{u,v \in \{0,1\}^m} A[u] \cdot B[v] \cdot \omega^{|u \oplus v|} + \sum_{u,v \in \{0,1\}^m} A[u] \cdot B[v] \cdot \omega^{2|u \oplus v|} \\
&- \sum_{u,v \in \{0,1\}^m} A[u] \cdot B[v] \cdot \omega^{|u \oplus v|+2} - \sum_{u,v \in \{0,1\}^m} A[u] \cdot B[v] \cdot \omega^{2|u \oplus v|+1}| \\
&\leq | \sum_{u,v \in \{0,1\}^m} A[u] \cdot B[v] \cdot \omega^{|u \oplus v|}| + | \sum_{u,v \in \{0,1\}^m} A[u] \cdot B[v] \cdot \omega^{2|u \oplus v|}| \\
&+ | \sum_{u,v \in \{0,1\}^m} A[u] \cdot B[v] \cdot \omega^{|u \oplus v|}| + | \sum_{u,v \in \{0,1\}^m} A[u] \cdot B[v] \cdot \omega^{2|u \oplus v|}|.
\end{aligned}$$

Now we proceed to bound each of the four terms separately.

$$\begin{aligned}
| \sum_{u,v \in \{0,1\}^m} A[u] \cdot B[v] \cdot \omega^{|u \oplus v|}|^2 &= | \sum_{u,v \in \{0,1\}^m} \frac{\alpha[u]}{2^m} \cdot \frac{\beta[v]}{2^m} \cdot \omega^{|u \oplus v|}|^2 \\
&= | \mathop{\mathbb{E}}_{u,v \in \{0,1\}^m} \alpha[u] \cdot \beta[v] \cdot \omega^{|u \oplus v|}|^2.
\end{aligned}$$

Claim 6.2.

$$| \mathop{\mathbb{E}}_{u,v \in \{0,1\}^m} \alpha[u] \cdot \beta[v] \cdot \omega^{|u \oplus v|}|^2 \leq 256 \cdot \left(\frac{3}{4}\right)^m.$$

Proof. We apply triangle inequality and then we remove $\alpha[u]$ using the fact that $\alpha[u] \leq 4$. Then we apply the Cauchy-Schwarz inequality,

$$\begin{aligned}
| \mathop{\mathbb{E}}_{u \in \{0,1\}^m} \alpha[u] \cdot \left(\mathop{\mathbb{E}}_{v \in \{0,1\}^m} \beta[v] \cdot \omega^{|u \oplus v|} \right) |^2 &\leq \left(\mathop{\mathbb{E}}_{u \in \{0,1\}^m} \alpha[u] \cdot | \mathop{\mathbb{E}}_{v \in \{0,1\}^m} \beta[v] \cdot \omega^{|u \oplus v|}| \right)^2 \\
&\leq 16 \cdot \left(\mathop{\mathbb{E}}_{u \in \{0,1\}^m} | \mathop{\mathbb{E}}_{v \in \{0,1\}^m} \beta[v] \cdot \omega^{|u \oplus v|}| \right)^2 \\
&\leq 16 \cdot \mathop{\mathbb{E}}_{u \in \{0,1\}^m} | \mathop{\mathbb{E}}_{v \in \{0,1\}^m} \beta[v] \cdot \omega^{|u \oplus v|}|^2.
\end{aligned}$$

Next, we write $|z|^2 = z \cdot \bar{z}$, rearrange terms and then apply triangle inequality,

$$\begin{aligned}
\mathop{\mathbb{E}}_{u \in \{0,1\}^m} | \mathop{\mathbb{E}}_{v \in \{0,1\}^m} \beta[v] \cdot \omega^{|u \oplus v|}|^2 &= \mathop{\mathbb{E}}_{u \in \{0,1\}^m} \left(\mathop{\mathbb{E}}_{v \in \{0,1\}^m} \beta[v] \cdot \omega^{|u \oplus v|} \right) \cdot \left(\mathop{\mathbb{E}}_{\tilde{v} \in \{0,1\}^m} \beta[\tilde{v}] \cdot \omega^{-|u \oplus \tilde{v}|} \right) \\
&= \mathop{\mathbb{E}}_{v \in \{0,1\}^m} \mathop{\mathbb{E}}_{\tilde{v} \in \{0,1\}^m} \beta[v]\beta[\tilde{v}] \cdot \left(\mathop{\mathbb{E}}_{u \in \{0,1\}^m} \omega^{|u \oplus v| - |u \oplus \tilde{v}|} \right) \\
&\leq \mathop{\mathbb{E}}_{v \in \{0,1\}^m} \mathop{\mathbb{E}}_{\tilde{v} \in \{0,1\}^m} \beta[v]\beta[\tilde{v}] \cdot | \mathop{\mathbb{E}}_{u \in \{0,1\}^m} \omega^{|u \oplus v| - |u \oplus \tilde{v}|}|.
\end{aligned}$$

Again we use $\beta[v] \leq 4$,

$$\begin{aligned}
\mathop{\mathbb{E}}_{v\in\{0,1\}^m}\mathop{\mathbb{E}}_{\tilde{v}\in\{0,1\}^m}\beta[v]\beta[\tilde{v}]\cdot\Big|\mathop{\mathbb{E}}_{u\in\{0,1\}^m}\omega^{|u\oplus v|-|u\oplus\tilde{v}|}\Big| &\leq 16\cdot\mathop{\mathbb{E}}_{v\in\{0,1\}^m}\mathop{\mathbb{E}}_{\tilde{v}\in\{0,1\}^m}\Big|\mathop{\mathbb{E}}_{u\in\{0,1\}^m}\omega^{|u\oplus v|-|u\oplus\tilde{v}|}\Big| \\
&= 16\cdot\mathop{\mathbb{E}}_{v,\tilde{v}\in\{0,1\}^m}\Big|\mathop{\mathbb{E}}_{u\in\{0,1\}^m}\omega^{|v|-|\tilde{v}|+2\langle u,\tilde{v}-v\rangle}\Big| \\
&= 16\cdot\mathop{\mathbb{E}}_{v,\tilde{v}\in\{0,1\}^m}\Big|\mathop{\mathbb{E}}_{u\in\{0,1\}^m}\omega^{2\langle u,\tilde{v}-v\rangle}\Big| \\
&= 16\cdot\sum_{k=0}^{m}\sum_{\substack{\hat{v}\in\{-1,0,1\}^m\\ \text{number of zeros in }\hat{v}=k}}\frac{2^k}{2^{2m}}\cdot\prod_i\Big|\mathop{\mathbb{E}}_{u_i\in\{0,1\}}\omega^{2u_i\cdot\hat{v}_i}\Big|.
\end{aligned}$$

A simple calculation shows that $\big|\mathop{\mathbb{E}}_{u_i\in\{0,1\}}\omega^{2u_i\cdot\hat{v}_i}\big| = \frac{1}{2}$, whenever $\hat{v}_i \in \{-1,1\}$. Plugging this back into the previous equation we get

$$\begin{aligned}
\sum_{k=0}^{m}\sum_{\substack{\hat{v}\in\{-1,0,1\}^m\\ \text{number of zeros in }\hat{v}=k}}\frac{2^k}{2^{2m}}\cdot\prod_i\Big|\mathop{\mathbb{E}}_{u_i\in\{0,1\}}\omega^{2u_i\cdot\hat{v}_i}\Big| &= \sum_{k=0}^{m}\binom{m}{k}\cdot 2^{m-k}\cdot\frac{2^k}{2^{2m}}\cdot\frac{1}{2^{m-k}} \\
&= \frac{1}{4^m}\sum_{k=0}^{m}\binom{m}{k}\cdot 2^k \\
&= \left(\frac{3}{4}\right)^m.
\end{aligned}$$

□

One can use a similar analysis to bound the other terms as well. Finally we conclude that $|W(a\cdot b)| \leq \frac{64}{3}\cdot\left(\sqrt{\frac{3}{4}}\right)^m$. □

References

[ACFN15] Anil Ada, Arkadev Chattopadhyay, Omar Fawzi, and Phuong Nguyen, *The NOF multiparty communication complexity of composed functions*, Comput. Complex. **24** (2015), no. 3, 645–694.

[AG87] Miklós Ajtai and Yuri Gurevich, *Monotone versus positive*, J. ACM **34** (1987), no. 4, 1004–1015.

[AV08] Manindra Agrawal and V. Vinay, *Arithmetic circuits: A chasm at depth four*, 49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA, IEEE Computer Society, 2008, pp. 67–75.

[BGK06] J. Bourgain, A. A. Glibichuk, and S. V. Konyagin, *Estimates for the number of sums and products and for exponential sums in fields of prime order*, Journal of London Mathematical Society **2** (2006), 380–398.

[BNS92] László Babai, Noam Nisan, and Mario Szegedy, *Multiparty protocols, pseudorandom generators for logspace, and time-space trade-offs*, J. Comput. Syst. Sci. **45** (1992), no. 2, 204–232, Preliminary version appeared in STOC 1989.

[CMS20] Arkadev Chattopadhyay, Nikhil Mande, and Suhail Sherif, *The log-approximate-rank conjecture is false*, J. ACM **67** (2020), no. 4, 23:1–23:28.

[COS17] Xi Chen, Igor Carboni Oliviera, and Rocco A. Servedio, *Addition is exponentially harder than counting for shallow monotone circuits*, STOC, 2017, pp. 665–677.

[GKKS16] Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi, *Arithmetic circuits: A chasm at depth 3*, SIAM J. Comput. **45** (2016), no. 3, 1064–1079.

[GKRS19] Mika Göös, Pritish Kamath, Robert Robere, and Dmitry Sokolov, *Adventures in monotone complexity and TFNP*, ITCS, 2019, pp. 38:1–38:19.

[GS12] S. B. Gashkov and I. S. Sergeev, *A method for deriving lower bounds for the complexity of monotone arithmetic circuits computing real polynomials*, Sbornik. Mathematics **203(10)** (2012).

[HR00] Danny Harnik and Ran Raz, *Higher lower bounds on monotone size*, Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing, May 21-23, 2000, Portland, OR, USA, ACM, 2000, pp. 378–387.

[Hru20] Pavel Hrubeš, *On ϵ-sensitive monotone computations*, Computational Complexity **29** (2020), no. 2, 6.

[HY13] Pavel Hrubeš and Amir Yehudayoff, *Formulas are exponentially stronger than monotone circuits in non-commutative setting*, Computational Complexity Conference, 2013, pp. 10–14.

[JS82] Mark Jerrum and Marc Snir, *Some exact complexity results for straight-line computations over semirings*, J. ACM **29** (1982), no. 3, 874–897.

[Koi12] Pascal Koiran, *Arithmetic circuits: The chasm at depth four gets wider*, Theor. Comput. Sci. **448** (2012), 56–65.

[KW] Mauricio Karchmer and Avi Wigderson, *Monotone circuits for connectivity require superlogarithmic depth*, SIAM J. Discrete Math.

[PR17] Toniann Pitassi and Robert Robere, *Strongly exponential lower bounds for monotone computation*, Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017, ACM, 2017, pp. 1246–1255.

[Raz85a] Alexander A. Razborov, *A lower bound on the monotone network complexity of the logical permanent*, Mathematicheskie Garnetki **37** (1985), no. 6, 887–900, Mathematical notes of the Academy of Sciences of the USSR, 37:6, 485–493.

[Raz85b] _____, *Lower bounds for the monotone complexity of some boolean functions*, Dokl. Ak. Nauk. SSSR **281** (1985), 354–357, (in Russian) English translation in *Sov. Math. Dokl.*

[Raz92] _____, *On the distributional complexity of disjointness*, Theor. Comput. Sci. **106** (1992), no. 2, 385–390, Preliminary version in ICALP 1990.

[RM99] Ran Raz and Pierre McKenzie, *Separation of the monotone NC hierarchy*, Combinatorica **19** (1999), no. 3, 403–435, Preliminary version in FOCS 1997.

[Ros15] Benjamin Rossman, *Correlation bounds against monotone nc^1*, 30th Conference on Computational Complexity, CCC 2015, June 17-19, 2015, Portland, Oregon, USA, LIPIcs, vol. 33, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015, pp. 392–411.

[RW92] Ran Raz and Avi Wigderson, *Monotone circuits for matching require linear depth*, J. ACM **39** (1992), no. 3, 736–744.

[RY11] Ran Raz and Amir Yehudayoff, *Multilinear formulas, maximal-partition discrepancy and mixed-sources extractors*, J. Comput. Syst. Sci. **77** (2011), no. 1, 167–190.

[Sri19] Srikanth Srinivasan, *Strongly exponential separation between monotone VP and monotone VNP*, Electron. Colloquium Comput. Complex. **26** (2019), 32.

[SS77] Eli Shamir and Marc Snir, *Lower bounds on the number of multiplications and additions in monotone computations*, Tech. report, IBM, 1977.

[SS80] _____, *On the depth complexity of formulas*, Theory of Computing Systems **13** (1980), no. 1, 301–322.

[SY10] Amir Shpilka and Amir Yehudayoff, *Arithmetic circuits: A survey of recent results and open questions*, Found. Trends Theor. Comput. Sci. **5** (2010), no. 3-4, 207–388.

[Ta-17] Amnon Ta-Shma, *Explicit, almost optimal, epsilon-balanced codes*, Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017 (Hamed Hatami, Pierre McKenzie, and Valerie King, eds.), ACM, 2017, pp. 238–251.

[Tar88] Eva Tardos, *The gap between monotone and non-monotone circuit complexity is exponential*, Combinatorica **8** (1988), 141–142.

[Tav15] Sébastien Tavenas, *Improved bounds for reduction to depth 4 and depth 3*, Inf. Comput. **240** (2015), 2–11.

[Val79] Leslie G. Valiant, *Completeness classes in algebra*, Proceedings of the 11h Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1979, Atlanta, Georgia, USA (Michael J. Fischer, Richard A. DeMillo, Nancy A. Lynch, Walter A. Burkhard, and Alfred V. Aho, eds.), ACM, 1979, pp. 249–261.

[Val80] _____, *Negation can be exponentially powerful*, Theor. Comput. Sci. **12** (1980), 303–314, Preliminary version in STOC 1979.

[W70] Moon J W, *Counting labelled trees*, Canadian Mathematical Congress, Montreal (1970).

[Yeh19] Amir Yehudayoff, *Separating monotone VP and VNP*, Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019 (Moses Charikar and Edith Cohen, eds.), ACM, 2019, pp. 425–429.

A The Omitted Proofs From Section 3

In this section, we present the omitted proofs from Section 3. First, we write a recurrence for $|\mathcal{F}^v_{n,m}|$. Using Fact 3.1, we may assume that the vector is of the form $v = (0, 0, 0, \ldots, 1, 1, 1)$. Then we have,

$$|\mathcal{F}^v_{n,m}| = \sum_{\substack{k=0 \\ k \text{ is even}}}^{n-\text{wt}(v)} \binom{n}{k} |\mathcal{F}^{v_{-1}}_{n-k,m-1}| \tag{9}$$

where v_{-1} is the vector obtained from v by deleting the first coordinate. If $v = (1, 1, 1, \ldots, 1, 1)$ then the recurrence is

$$|\mathcal{F}^v_{n,m}| = \sum_{\substack{k=1 \\ k \text{ is odd}}}^{n-\text{wt}(v)} \binom{n}{k} |\mathcal{F}^{v_{-1}}_{n-k,m-1}|.$$

Claim A.1 (Re-statement of Claim 3.1)**.** $|\mathcal{F}^v_{n,m}| \geq |\mathcal{F}^u_{n,m}|$ *when* $wt(v) = wt(u) - 2$.

Proof. The proof is by induction on m with base case at $m = 2$. Notice that for $m = 2$, $v = (0, 0)$ and $u = (1, 1)$. Hence

$$|\mathcal{F}^v_{n,m}| = \sum_{\substack{k=0 \\ k \text{ is even}}}^{n-\text{wt}(v)} \binom{n}{k} = 2^{n-1}.$$

Similarly,

$$|\mathcal{F}^u_{n,m}| = 2^{n-1}.$$

Writing the recurrence for $|\mathcal{F}^v_{n,m}|$ we have

$$|\mathcal{F}^v_{n,m}| = \sum_{\substack{k=0 \\ k \text{ is even}}}^{n-\text{wt}(v)} \binom{n}{k} |\mathcal{F}^{v_{-1}}_{n-k,m-1}|.$$

Applying the induction hypothesis for $m - 1$ on the vectors v_{-1} and u_{-1} we conclude that $|\mathcal{F}^{v_{-1}}_{n-k,m-1}| \geq |\mathcal{F}^{u_{-1}}_{n-k,m-1}|$ which shows that,

$$\begin{aligned}
|\mathcal{F}^v_{n,m}| &= \sum_{\substack{k=0 \\ k \text{ is even}}}^{n-\text{wt}(v)} \binom{n}{k} |\mathcal{F}^{v_{-1}}_{n-k,m-1}| \\
&\geq \sum_{\substack{k=0 \\ k \text{ is even}}}^{n-\text{wt}(u)} \binom{n}{k} |\mathcal{F}^{u_{-1}}_{n-k,m-1}| + \binom{n}{n-\text{wt}(v)} |\mathcal{F}^{v_{-1}}_{\text{wt}(v),m-1}| \\
&\geq \sum_{\substack{k=0 \\ k \text{ is even}}}^{n-\text{wt}(u)} \binom{n}{k} |\mathcal{F}^{u_{-1}}_{n-k,m-1}| \\
&= |\mathcal{F}^u_{n,m}|.
\end{aligned}$$

This completes the induction. □

When n is even we shall denote $\mathcal{F}_{n,m}^{(0,0,\ldots,0,0)}$ as $\mathcal{EF}_{n,m}$ and call it the set of even functions.

Lemma A.1 (Re-statement of Lemma 3.1)**.**

$$|\mathcal{EF}_{n,m}| \leq \frac{1}{2^{m-1}} \left(\sum_{i=0}^{\frac{m-2}{2}} \binom{m}{i} (m-2i)^n \right) \textit{ when } m \textit{ is even}$$

$$\leq \frac{1}{2^{m-1}} \left(\sum_{i=0}^{\frac{m-1}{2}} \binom{m}{i} (m-2i)^n \right) \textit{ when } m \textit{ is odd}$$

Proof. The proof is by induction on m and the base case is $m = 2$. We have

$$\mathcal{EF}_{n,2} = \sum_{\substack{k=0 \\ k \text{ is even}}}^{n} \binom{n}{k} = 2^{n-1}.$$

and hence the base case is established. We do the induction step when $m+1$ is odd (the even case when $m+1$ is even is similar). We write the recurrence and apply the induction hypothesis.

$$\begin{aligned}
|\mathcal{EF}_{n,m+1}| &= \sum_{\substack{k=0 \\ k \text{ is even}}}^{n} \binom{n}{k} |\mathcal{EF}_{n-k,m}| \\
&\leq \sum_{\substack{k=0 \\ k \text{ is even}}}^{n} \binom{n}{k} \frac{1}{2^{m-1}} \left(\sum_{i=0}^{\frac{m-2}{2}} \binom{m}{i} (m-2i)^{n-k} \right) \\
&\leq \frac{1}{2^{m-1}} \sum_{i=0}^{\frac{m-2}{2}} \binom{m}{i} \left(\sum_{\substack{k=0 \\ k \text{ is even}}}^{n} \binom{n}{k} (m-2i)^{n-k} \right) \\
&\leq \frac{1}{2^m} \left(\sum_{i=0}^{\frac{m-2}{2}} \binom{m}{i} \left((m+1-2i)^n + (m-1-2i)^n \right) \right) \\
&\leq \frac{1}{2^m} \left(\sum_{i=0}^{\frac{m-2}{2}} \binom{m}{i} \left((m+1-2i)^n + (m+1-2(i+1))^n \right) \right) \\
&\leq \frac{1}{2^m} \left(\binom{m+1}{0} (m+1)^n + \sum_{i=1}^{\frac{m}{2}} \left[\binom{m}{i-1} + \binom{m}{i} \right] \left((m+1-2i)^n \right) \right) \\
&= \frac{1}{2^m} \left(\binom{m+1}{0} (m+1)^n + \sum_{i=1}^{\frac{m}{2}} \binom{m+1}{i} \left((m+1-2i)^n \right) \right).
\end{aligned}$$

This completes the induction.

□

Corollary A.1 (Re-statement of Corollary 3.1)**.** *Let n be even. If $m \ln m \leq n$, then for every $v \in \{0,1\}^m$ we have,*

$$|\mathcal{F}^v_{n,m}| \leq \frac{4m^n}{2^m}.$$

Proof. We shall prove the bound for $|\mathcal{EF}_{n,m}|$ and then by lemma A.1 the bound will hold for $|\mathcal{F}^v_{n,m}|$ for all $v \in \{0,1\}^m$. From Lemma A.1 we know that

$$\begin{aligned}
|\mathcal{EF}_{n,m}| &\leq \frac{1}{2^{m-1}} \left(\sum_{i=0}^{\frac{m-2}{2}} \binom{m}{i} (m-2i)^n \right) \\
&\leq \frac{m^n}{2^{m-1}} \left(\sum_{i=0}^{\frac{m-2}{2}} \binom{m}{i} \left(\frac{m-2i}{m} \right)^n \right) \\
&\leq \frac{m^n}{2^{m-1}} \left(\sum_{i=0}^{\frac{m-2}{2}} m^i \left(1 - \frac{2i}{m} \right)^n \right) \\
&\leq \frac{m^n}{2^{m-1}} \left(\sum_{i=0}^{\frac{m-2}{2}} m^i e^{\frac{-2in}{m}} \right) \\
&\leq \frac{m^n}{2^{m-1}} \left(\sum_{i=0}^{\frac{m-2}{2}} \left(\frac{m}{e^{\frac{2n}{m}}} \right)^i \right).
\end{aligned}$$

Now by the choice of n and m we have $\frac{m}{e^{\frac{2n}{m}}} \leq \frac{1}{m}$ and hence we may bound

$$\sum_{i=0}^{\frac{m-2}{2}} \left(\frac{m}{e^{\frac{2n}{m}}} \right)^i \leq \sum_{i=0}^{\frac{m-2}{2}} \left(\frac{1}{m} \right)^i \leq \frac{m}{m-1} \leq 2.$$

Thus we conclude that when $m \ln m \leq n$ we have $|\mathcal{EF}_{n,m}| \leq \frac{4m^n}{2^m}$.

□

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il