Electronic Colloquium on Computational Complexity, Report No. 219 (2025)

Negations are powerful even in small depth

Bruno Cavalar * Théo Borém Fabris† Partha Mukhopadhyay‡
Srikanth Srinivasan§ Amir Yehudayoff¶

December 22, 2025

Abstract

We study the power of negation in the Boolean and algebraic settings and show the following results.

1. We construct a family of polynomials P_n in n variables, all of whose monomials have positive coefficients, such that P_n can be computed by a depth three circuit of polynomial size but any monotone circuit computing it has size $2^{\Omega(n)}$. This is the strongest possible separation result between monotone and non-monotone arithmetic computations and improves upon all earlier results, including the seminal work of Valiant (1980) and more recently by Chattopadhyay, Datta, and Mukhopadhyay (2021). We then boot-strap this result to prove strong monotone separations for polynomials of constant degree, which solves an open problem from the survey of Shpilka and Yehudayoff (2010).
2. By moving to the Boolean setting, we can prove superpolynomial monotone Boolean circuit lower bounds for specific Boolean functions, which imply that *all the powers* of certain monotone polynomials cannot be computed by polynomially sized monotone arithmetic circuits. This leads to a new kind of monotone vs. non-monotone separation in the arithmetic setting.
3. We then define a collection of problems with linear-algebraic nature, which are similar to span programs, and prove monotone Boolean circuit lower bounds for them. In particular, this gives the strongest known monotone lower bounds for functions in uniform (non-monotone) $\mathbf{NC}^2$. Our construction also leads to an explicit matroid that defines a monotone function that is difficult to compute, which solves an open problem by Jukna and Seiwert (2020) in the context of the relative powers of greedy and pure dynamic programming algorithms.

Our monotone arithmetic and Boolean circuit lower bounds are based on known techniques, such as reduction from monotone arithmetic complexity to multipartition communication complexity and the approximation method for proving lower bounds for monotone Boolean circuits, but we overcome several new challenges in order to obtain efficient upper bounds using low-depth circuits.

*Department of Computer Science, University of Oxford, United Kingdom. Supported by EPSRC project EP/Z534158/1 on "Integrated Approach to Computational Complexity: Structure, Self-Reference and Lower Bounds". Email: bruno.cavalar@cs.ox.ac.uk

†Department of Computer Science, University of Copenhagen, Denmark. Supported by the European Research Council (ERC) under grant agreement no. 101125652 (ALBA). Email: thfa@di.ku.dk

‡Chennai Mathematical Institute, India. Email: partham@cmi.ac.in

§Department of Computer Science, University of Copenhagen, Denmark. Supported by the European Research Council (ERC) under grant agreement no. 101125652 (ALBA). Email: srsr@di.ku.dk

¶Department of Computer Science, The University of Copenhagen, and The Technion-IIT. Supported by a DNRF Chair grant. Email: amir.yehudayoff@gmail.com

ISSN 1433-8092

Contents

1 Introduction

1.1 Background and results

Separation between arithmetic models. In his paper "A single negation is exponentially powerful", Valiant [Val80] proved that there are n-variate monotone polynomials in **VP** that require a monotone arithmetic circuit of size at least $2^{\Omega(n^{1/2})}$ and that every computation in **VP** can be performed with a single negation gate. Chattopadhyay, Datta and Mukhopadhyay [CDM21] recently strengthened the upper bound to depth-three circuits (also known as a $\Sigma\Pi\Sigma$ circuit); they proved a $2^{\tilde{\Omega}(n^{1/4})}$ lower bound on the monotone circuit complexity of a monotone polynomial that can be computed by a $\Sigma\Pi\Sigma$ circuit of polynomial size. Another recent work shows that the spanning tree polynomials, having n variables and defined over constant-degree expander graphs, have monotone arithmetic circuit complexity $2^{\Omega(n)}$ [CDGM22]. However, the best known non-monotone upper bound for the spanning tree polynomial is a polynomial-size algebraic branching program (ABP) [W70] via a determinant computation [Csa76, Ber84]. Our first main result is a strengthening of these results.

Theorem 1.1. There is an n-variate polynomial $P = P_n$ such that the following hold:

1. P can be computed by a $\Sigma\Pi\Sigma$ circuit of size $O(n^3)$.
2. Any monotone arithmetic circuit computing P has size at least $2^{\Omega(n)}$.

The family of polynomials $(P_n)_{n\geq 1}$ in Theorem 1.1 can be computed by a uniform sequence of circuits. That is, there is a Turing machine that gets n as input, runs in time $\mathrm{poly}(n)$ and outputs a circuit computing P_n. The polynomial P_n is described using an expander graph G with $n/2$ vertices. It is of the form

$$P_n(x) = \sum_{a\in\{0,1\}^{n/2}} \left(\sum_{\{u,v\}\in E(G)} a_u a_v - 1 \right)^2 \prod_{i=1}^{n/2} x_{i,a_i}. \tag{1}$$

We prove below that there is a small $\Sigma\Pi\Sigma$ circuit for P_n. Using the fact that G is an expander, we show that the monotone circuit complexity of P_n is high. This is based on a reduction from monotone arithmetic circuit lower bounds to a problem in communication complexity. Several recent results exploit the connection between monotone circuit lower bounds and communication complexity [RY11, HY13, Sri20, CDM21, CDGM22, CGM22].

All together, this leads to the strongest possible separation between monotone and non-monotone computation in the algebraic setting. A single negation gate which performs the subtraction of two monotone depth three circuits is exponentially powerful even compared to general monotone circuits.

A padding argument allows us to deduce new separation between monotone and non-monotone arithmetic circuits for constant degree polynomials. This allows us to solve Open Problem 9 in the survey of Shpilka and Yehudayoff [SY10]. Before this work, such a separation for polynomials of constant degree, was known only between monotone and general arithmetic *formulas* [HY09].

Theorem 1.2. For every constant $k \in \mathbb{N}$ and $n \in \mathbb{N}$, there is an n-variate polynomial $P = P_{n,k}$ of total degree k such that the following hold:

1. P can be computed by a $\Sigma\Pi\Sigma$ circuit of size $O(n(\log n)^2)$.

2. Any monotone arithmetic circuit computing P has size at least $(n/k)^{\Omega(k)}$.

Such a padding argument is only possible because we prove a *strongly* exponential lower bound in Theorem 1.1 (strengthening [CDM21]) and because of the simplicity of our upper bounds for the polynomial P_n defined above (it is not clear how to get such a separation from the strongly exponential lower bound of [CDGM22]).

Separation between arithmetic and Boolean models. One of the major challenges in the study of monotone arithmetic complexity is proving that there are explicit monotone polynomials P such that for every monotone polynomial Q, the monotone complexity of $P \cdot Q$ is high [HY21]. All of our proof techniques seem to fail for this problem. We take a step towards this problem. We identify an explicit monotone polynomial P such that P has small non-monotone arithmetic circuits but every power of P has high monotone complexity. As far as we know, this is the first separation of this kind.

We do this via a connection to *Boolean* circuit lower bounds. For every n-variate monotone polynomial P over $\mathbb{R}$, we can define a monotone Boolean function $f_P : \{0,1\}^n \to \{0,1\}$ via the correspondence

$$f_P(x) = 1 \iff P(x) > 0.$$

Our first observation is that a monotone circuit for P leads to a monotone circuit for f_P.

Observation 1.3. If P can be computed by a monotone arithmetic circuit of size s, then f_P can be computed by a monotone Boolean circuit of size s.

Sketch. The Boolean circuit for s is obtained by replacing $+$ gates by $\vee$ gates and $\times$ gates by $\wedge$ gates. ■

One advantage is that f_P corresponds not just to P but also to all the powers of P. So, a monotone lower bound for f_P leads to a monotone lower bound to all the powers of P. A second important advantage is that we know how to prove lower bounds for monotone Boolean complexity. Razborov [Raz85] and many works that followed [AB87, Tar88, HR00, CKR22, CGR+25] developed the approximation method for proving Boolean monotone circuit complexity lower bounds. In particular, the known lower bounds for the perfect bipartite matching function already imply that any power of the permanent polynomial is superpolynomially [Raz85] (and even exponentially [CGR+25]) hard for monotone arithmetic circuits.

We can extend this to show separations. To do this, we also need to upper bound the non-monotone circuit complexity of P or f_P. A central tool in proving arithmetic circuit upper bounds in through linear algebra and the determinant [Csa76, Val79, Ber84, MV97]. All together we get the following strong separation.

Theorem 1.4. There is a monotone n-variate polynomial $P = P_n$, over the reals, such that the following hold:

1. P can be computed by an arithmetic circuit of size polynomial in n and depth $O((\log n)^2)$.

2. Any monotone Boolean circuit computing the Boolean function f_P that corresponds to P has size at least $n^{\Omega((\log n)^{1/2})}$.

It follows that any monotone arithmetic circuit computing a power of P must be of superpolynomial size as well.

Separations between Boolean models. An adaptation of our construction to work over the field $\mathbb{F}_2$ leads to an improved separation between monotone and non-monotone circuits in the Boolean setting as well. We take a small detour to explain the context better. The results in [AB87, Tar88] show that there exists a monotone function in **P** that requires monotone circuits of size $2^{n^{1/6-o(1)}}$. Building on these results, de Rezende and Vinyals [dRV25] have recently shown that there exists a monotone function in **P** that requires monotone circuits of size $2^{n^{1/3-o(1)}}$. Very recently, Cavalar et al. [CGR+25] improved Razborov's result and showed that the matching function for bipartite graphs (computable in **RNC** [Lov79] and in **quasi-NC** [FGT19]) and another explicit monotone function computable in **L** require monotone circuits of size $2^{n^{1/6-o(1)}}$. Earlier works [GKRS18, GGKS20] also exhibited an explicit function in **NC** requiring monotone circuits of size $2^{n^{\varepsilon}}$, where $\varepsilon > 0$ is an unspecified constant.

In this paper, we obtain the following theorem, proving the strongest quantitative lower bounds for a function in **NC** (except for the $o(1)$ terms, it also matches the best known lower bound for a function in **P**).

Theorem 1.5. For every sufficient large $n \in \mathbb{N}$, there is a monotone Boolean function $f : \{0,1\}^n \to \{0,1\}$ such that the following hold:

1. f can be computed by a uniform Boolean circuit of size polynomial in n and depth $O(\log n)^2$, i.e., f is in uniform $\mathbf{NC}^2$
2. Any monotone Boolean circuit computing f has size at least $2^{n^{1/3-o(1)}}$.

The function f from Theorem 1.5 has an additional important property. As any monotone function is specified by its min-terms, it is interesting to compare the "complexity" of the min-terms to the monotone complexity of the function. Concretely, Jukna and Seiwert [JS20] studied monotone functions whose min-terms are the bases of a matroid. Their motivation was comparing the power of greedy algorithms and pure dynamic programming algorithms. In their terminology, the function f from Theorem 1.5 gives an explicit problem for which a greedy algorithm exactly solves the problem whereas the cost of approximately solving the problem using pure dynamic programming is high. They proved that such problems exists using counting arguments, but we provide the first explicit example of such a function, solving an open problem from their work (see [JS20, Problem 1]). While there have been other lower bounds recently for monotone Boolean functions defined via linear algebra [GKRS18, CGR+25], they do not seem to yield a lower bound for a matroid-based function as above.

1.2 Challenges and Proof Ideas

Arithmetic lower bounds. In order to prove Theorem 1.1, we follow an idea of Hrubeš and Yehudayoff [HY13] in the *non-commutative* setting,[1] which was in turn inspired by results from communication complexity [Raz92, FMP+15]. Let $P(x_{1,0}, x_{1,1}, \ldots, x_{n,0}, x_{n,1})$ be a (set-multilinear) polynomial such that each monomial in P has the form $x_{1,a_1} \cdots x_{n,a_n}$ for some $a \in \{0,1\}^n$. It is known [Nis91] that if P has a 'simple'[2] monotone non-commutative circuit of size s, then we can

[1] In this model, variables do not commute.

[2] The technical term for this is an Algebraic Branching Program (ABP). A similar but more complicated decomposition can be given for general circuits.

write

$$P = \sum_{i=1}^{s} Q_i \cdot R_i$$

where each Q_i depends on the first half of the variables ($x_{j,b}$ for $j \in [n/2]$) and R_i on the latter half, and the polynomials are all monotone. This is analogous to the setting of communication complexity, where we can use a small 2-party communication protocol for a function $f : \{0,1\}^{n/2} \times \{0,1\}^{n/2} \to \{0,1\}$ to decompose f into a small sum of Boolean (and hence non-negative) *rectangles*. The work of [HY13] shows how to exploit this connection to obtain a non-monotone versus monotone separation. The separating example is inspired by the *Unique-Disjointness* function. This function has a simple algebraic description via inner products which leads to a non-monotone upper bound, and the lower bound uses ideas from Razborov's lower bound [Raz92] on the communication complexity of this problem.

Our set-up is the standard *commutative* setting, which means that we get a similar decomposition to the one above, except that each Q_i depends on (an unknown subset of) half of the variables (or more precisely $x_{j,b}$ for some $j \in S_i$ where $|S_i| = n/2$). Now, the analogy is no longer with the standard model of communication complexity but with the model of *multipartition* communication complexity, where each rectangle in the decomposition is allowed to use a different partition of the inputs. This model and its variants have been investigated before in works of [ĎHJ04, Hay11, HY16]. The work of [Hay11, Juk15, HY16] in particular discovered a technique to 'lift' some structured lower bounds from standard communication complexity to the multipartition setting, by defining variants of the hard functions (for the standard communication model) using an expander graph. In particular, the lower bounds for the Unique-disjointness function are amenable to this lifting technique. The polynomial P_n defined in (1) is an algebraic variant of this lifted Boolean function (analogous to how the separating example of [HY13] is related to the Unique-disjointness function). The lower bound argument combines a *non-deterministic* communication complexity lower bound for Unique-Disjointness (this is due to Razborov but we use a version due to [KW15]). To lift the lower bound, we need to construct a hard distribution that allows to carry out a covering argument. This hard distribution uses some ideas from [Sri20].

To get a separation for constant-degree polynomials (hence resolving the open problem from [SY10]), we use a padding argument. The variables of P_n naturally come in n groups, each of size 2. We collect these groups into k buckets of size n/k each. There are now $2^{n/k}$ possible monomials in each group and we define a new polynomial Q_k treating these monomials as our variables. It easily follows that Q_k is at least as hard as P_n since a circuit for Q_k easily yields a circuit for P_n by replacing each variable of Q_k by the corresponding monomial. The converse, however, is not clear. Nevertheless, because our depth-3 non-monotone circuits for P_n are very simple, we are also able to modify this construction to get a non-monotone circuit for Q_k of small size and depth.

Arithmetic and Boolean upper bounds. To prove Theorems 1.4 and 1.5, we consider polynomials and Boolean functions corresponding to full-rank sets of vectors. Let M be an $r \times n$ matrix over a field $\mathbb{F}$. We are going to consider both the cases in which the field is a finite field and $\mathbb{R}$. For any subset $S \subseteq [n]$ of the columns of M, let $M[S]$ be the restriction of M to the columns in S. Let $f_M : \{0,1\}^n \to \{0,1\}$ be the monotone Boolean function associated to M defined as

$$f_M(x) = 1 \text{ iff the matrix } M[S] \text{ has full rank},$$

where $S = \{i \in [n] : x_i = 1\}$ is the set of columns of M that x indicates. Equivalently, the function $f_M(x)$ evaluates to 1 iff the linear system $y^\mathsf{T} M[S] = 0$ has a unique solution. This can be checked by a Boolean circuit of polynomial size and depth $O(\log n)^2$ [ABO99, Mul87].

When $\mathbb{F} = \mathbb{R}$, we also associate with M an n-variate multilinear polynomial P_M via

$$P_M(x) := \sum_{\substack{S \subseteq [n]: \\ |S| = r}} \det(M[S])^2 \cdot \prod_{i \in S} x_i. \tag{2}$$

For $x \in \{0,1\}^n$, it follows that $P_M(x) > 0$ iff $f_M(x) = 1$. The Cauchy–Binet formula allows to compute P_M efficiently with a non-monotone circuit (and in poly-log depth). This leads to our arithmetic non-monotone upper bounds.

It is tempting to try to derive our separations with the type of functions considered in [GKRS18, CGR+25]. These functions correspond to checking whether a certain system of linear equations is satisfiable. More concretely, there is an implicit matrix M over a field $\mathbb{F}$, and the input string is interpreted as encoding a subset S of the columns of M. The function accepts if $M[S]$ spans a fixed target vector $v \neq 0$. However, unlike the full-rank property which is tightly related to the determinant polynomial, it does not seem possible to connect such a span restriction with an efficiently constructible polynomial. Moreover, whereas a full-rank set of vectors is a basis of a vector space (and thus of a matroid), the same cannot be said of vectors spanning v. This is crucial for our solution of the Problem 1 of [JS20].

Finally, as we discuss in the *Open Problems* section, the function f_M can be computed by a rarely studied monotone model of computation which we name *monotone rank programs*. These are restrictions of *monotone span programs*, a widely studied monotone computational model [KW93, BGW99] that computes the functions from [GKRS18, CGR+25] and is equivalent to linear secret-sharing schemes [Bei11]. It is possible that monotone rank programs are much weaker than monotone span programs, just like *monotone dependency programs* (another restriction of monotone span programs) [PS96]. If that is true, then our work extends the exponential monotone circuit lower bound shown for monotone span programs [GKRS18] to an even simpler computational model.

Monotone Boolean lower bounds. We now discuss the monotone Boolean circuit lower bounds in this paper. Recall from above that there is a fixed matrix $r \times n$ M and we want to prove a lower bound for the Boolean function $f_M : \{0,1\}^n \to \{0,1\}$ which decides if a given subset of the columns of M is full-rank or not.

To prove such a lower bound, we turn to the approximation technique of Razborov [Raz85] which has seen many refinements over the years [AB87, Ros14, CKR22, CGR+25]. Based on these works, we abstract out the core of the approximation method as applied to monotone Boolean circuit lower bounds into a general and clean statement (Theorem 3.3) below. Informally, the statement says that to prove a monotone circuit lower bound for a Boolean function f, it suffices to devise two distributions $\mathcal{D}_0$ and $\mathcal{D}_1$, supported on the 0s and 1s of f respectively, that satisfy two conditions:

- (Spreadness of $\mathcal{D}_1$): The probability that any w variables are simultaneously set to 1 under $\mathcal{D}_1$ is at most q^{-w} (for as large a q and w as possible).

- (Robust Sunflowers for $\mathcal{D}_0$): Any w-DNF of size s^k contains a subset that is a *robust sunflower* with respect to $\mathcal{D}_0$. Informally, this is a subset of the terms that can be replaced by a single term without changing the DNF much on inputs from $\mathcal{D}_0$.

The lower bound obtained from these two conditions is roughly $(q/s)^w$. At a high-level, the spreadness of $\mathcal{D}_1$ says that it is supported on strings of low weight, and the sunflower property of $\mathcal{D}_0$ says that it is supported on strings of high weight. By the monotonicity of f, these weight are "opposite of what they should be"; the 1s of f should be higher than the 0s of f.

To use this technique for f_M, we need to define the distributions which in turn depend on the choice of M. The cleanest example is that of a random $r \times n$ matrix over $\mathbb{F}_2$ where $r \approx n^{\varepsilon}$. Standard calculations show that such a matrix satisfies two linear algebraic properties:

- P1: Most subsets of r columns are linearly independent.
- P0: *All* subsets of columns of cardinality $r' = \tilde{\Omega}(r)$ are linearly independent.

P1 allows us to easily obtain a spread $\mathcal{D}_1$ with as low a Hamming weight as possible: simply choose a random subset of size r and set these bits to 1. This is a spread distribution with spreadness parameter $q \approx (n/r)$.

Choosing the distribution $\mathcal{D}_0$ is slightly trickier: To sample a large number of columns that are linearly *dependent*, we choose a random vector $u \in \mathbb{F}_2^r$ and pick all the columns that are orthogonal to u. This distribution has *uniform* marginals (and thus relatively high Hamming weight) and is also r'-wise independent by property P0 above. This last property allows us to prove the robust sunflower property for $\mathcal{D}_0$ by replacing $\mathcal{D}_0$ by the uniform distribution, since low-width $w \approx \sqrt{r'}$ DNFs cannot distinguish between the two [Baz09, Tal17] except with error $\exp(-\tilde{\Omega}(\sqrt{r'}))$. Recent bounds on robust sunflower lemmas with respect to product distributions [ALWZ21, BCW21, Rao25] can now be used to argue the desired property with the parameter $s \approx \sqrt{r'}$. Optimizing the parameters leads to choosing r as $r \approx n^{2/3}$ which leads to a lower bound of $\approx \exp(n^{1/3})$. Interestingly, this is the first time Bazzi's theorem is employed to show monotone circuit lower bounds.

We note that Bazzi's theorem is the only overhead in our lower bound, otherwise we could have matched the best known monotone circut lower bound for an **NP** function ($\approx \exp(n^{1/2})$ [CKR22]). Nonetheless, our reduction to the robust sunflower lemma via Bazzi is more efficient than the reduction from matching-sunflowers via "blocky" families of [CGR$^+$25], which gives us a lower bound of higher order for an **NC** function (theirs being $\approx \exp(n^{1/6})$). On the other hand, their upper bound is better (the function is in **L**), which makes the results arguably incomparable.

To obtain explicit matrices with the above properties, we use constructions of binary linear codes with sufficiently high distances and dual distances. By translating properties P0 and P1 into the coding theory language, we can obtain P0 by proving that the dual distance of the code is at least r', and P1 by proving that its distance is sufficiently large.

Finally, to prove the separation between the arithmetic and Boolean models, we need to argue the above lower bound for a *real* matrix M. A natural strategy is to repeat the above argument, say, for random matrices where each column is an independent sign vector $v \in \{-1, 1\}^r$. However this leads to a problem with defining the distribution $\mathcal{D}_0$ as above, since the point thus sampled is unlikely to have a high Hamming weight (by standard anti-concentration arguments [LO43, Erd45], the chance that two random sign vectors are orthogonal is small), and thus $\mathcal{D}_0$ has marginals that are small.

We need to have many real vectors so that many of their subsets are linearly dependent. *How to achieve this?* The main idea is to choose a sparse matrix. We sample each column of M to be a random Boolean vector of low Hamming weight. This now gives a distribution $\mathcal{D}_0$ that has higher marginals and satisfy a weak form of bounded independence. Unfortunately, we are unable to apply this weak form in conjunction with the strongest robust sunflower lemmas. We are, however, able to use it along with the classical sunflower lemma [ER60] and thus prove a superpolynomial lower bound *à la* Razborov [Raz85] (instead of an exponential lower bound).

Hamming balls and subspaces. On the way to establishing the spreadness property for $\mathcal{D}_1$, we prove a technical result about intersection patterns of subspaces and the Boolean cube, which we believe is independently interesting. Let $\mathbb{F}$ be any field and V be a subspace of $\mathbb{F}^n$ of dimension d. It is a standard fact that $|V \cap \{0,1\}^n| \leq 2^d$ (see e.g. [AY24]). We extend this bound from the full Hamming cube to a Hamming ball of radius s. We prove that the number of points of the Boolean cube and Hamming weight at most s in V is at most $\binom{d}{\leq s}$. These bounds are all easily seen to be tight for a vector space V generated by d standard basis vectors.

1.3 Open problems

The power of monotone rank programs. A *monotone span program of size k over a field* $\mathbb{F}$ is a $\mathbb{F}$-matrix A with k rows, together with a labelling of the rows with an input variable from $\{x_1, \ldots, x_n\}$. The span program accepts a binary string $x \in \{0,1\}^n$ if the set A_x of rows whose labels are satisfied by x span the vector $e_1 = (1, 0, \ldots, 0)$ (any other vector can be chosen by a proper change of basis). There is a linear-size monotone span program computing the functions from [GKRS18, CGR+25]. Using standard tricks, it is not hard to compute f_M with a monotone span program as well.

Interestingly, the function f_M can also be computed by a seemingly simpler monotone model which we name *monotone rank programs*. Just like a monotone span program, a monotone rank program is a matrix A with a labelling of the rows. The difference is that a monotone rank program accepts an input x iff A_x is full-rank. Such models have been studied in the context of linear secret sharing schemes [NNP04] and have been called "non-redundant" as there are no dependencies in A_x for any minimally accepting x.

A related model called *monotone dependency programs* accepts x iff A_x is linearly dependent. This was shown to be exponentially weaker than monotone span programs in [PS96]. It remains an open problem to determine whether monotone rank programs are weaker than monotone span programs.

Monotone complexity of matching over planar graphs. There are two parts for proving a separation in computational complexity. We have two computational classes A and B, and we need to come up with a problem P that belongs to B but does not belongs to A. In our context, A is a monotone class and B is a non-monotone class. We have used techniques from communication complexity [Raz92, KW15] and approximation methods [Raz85] to prove lower bounds against A. To prove that P belongs to B, we have used either the power of depth-3 circuits with negations, or the power of linear algebra.

There is also a different path for proving separations. In his work [Val80], Valiant used the fact the number of perfect matchings in a planar graphs is given by the Pfaffian [Kas67] so we can reduce it to computing the determinant polynomial. Again, the upper bound comes from linear algebra. Valiant proved that (for some planar graphs) the Pfaffian can be computed by a non-monotone arithmetic circuit of polynomial size; more precisely, by an algebraic branching program of polynomial size. But we are interested in separating Boolean monotone circuits and arithmetic non-monotone circuits. We know that the upper bound holds, but we do not know if deciding the perfect matching problem is hard for planar graphs for monotone Boolean circuits.

Constant-depth separations. Theorem 1.1 shows a polynomial computable by depth-3 arithmetic circuits requiring maximal monotone arithmetic complexity, improving previous separations of this kind [CDM21]. In the Boolean setting, a line of works [AG87, Oko82, RW92, GS92, BGW99, COS17, GKRS18, CO23, CGR+25] has either directly or indirectly studied the relative power of constant-depth and monotone circuits, recently discovering a monotone function computable by constant-depth circuits which requires superpolynomial size monotone circuits [CGR+25]. It remains an open question whether there exists a monotone function computable by some constant-depth circuit model (in fact, even $O(\log n)$-depth) requiring exponential-size monotone circuits. Is there a matrix M such that f_M can exhibit this separation?

This type of question is also interesting from the point of view algebraic separations. Is there a polynomial P computable by constant-depth or even logarithmic-depth arithmetic circuits and such that f_P requires superpolynomial or even exponential size monotone Boolean circuits? This looks plausible in light of several nontrivial algorithms which have recently been implemented with constant-depth arithmetic circuits [AW24].

Other questions. There are also more specific questions about the techniques we used to obtain our results:

- Can we improve our lower bounds in Section 2 to $\Omega\left(\binom{n}{k}\right)$ for an n-variate polynomial of constant degree k?
- Can we make the bounds in Section 5 (Theorem 1.4) exponential? This requires proving a better robust sunflower lemma for distributions that are only weakly independent (in the sense of Lemma 5.11). One approach would be to show that our distribution fools low-width DNFs. Note that recent work has shown limitations on what kinds of distributions can fool DNFs [AGG+25].
- Can we make the random choice of matrices M_n in Sections 5 explicit? In Section 4, our construction uses explicit constructions of binary linear codes with high distances and dual distances.
- Does the lower bound in Section 5 also hold if each column (in M_n) is just $O(1)$-sparse?

1.4 Acknowledgements

PM, TBF and SS started their collaboration at the *Workshop on Algebraic Complexity Theory (WACT) 2025* in Bochum and would like to thank the organizers of this workshop for their hospitality. SS would also like to thank Rohit Gurjar and Roshan Raj for pointing him to polynomial identities related to matroids using the Cauchy-Binet formula; Radu Curticapean for a discussion related to subgraph polynomials; Duri Janett for references related to lifting theorems; and Susanna de Rezende for clarifications regarding [dRV25]. BC acknowledges support of EPSRC project EP/Z534158/1 on "Integrated Approach to Computational Complexity: Structure, Self-Reference and Lower Bounds".

Organization of the paper. The rest of the paper is organized as follows. The proofs of Theorem 1.1 and Theorem 1.2 are presented in Section 2. Section 3 contains the necessary framework for our lower bounds of Boolean monotone circuits. The proof of Theorem 1.4 and Theorem 1.5 are given in Section 4 and in Section 5 respectively.

2 Arithmetic separations

The goal of this section is to prove strong separations between monotone and non-monotone algebraic circuits.

2.1 Polynomials from graphs and their $\Sigma\Pi\Sigma$ upper bounds

Let G be a graph with vertex-set $[n]$. Denote by $E(G)$ the edge-set of G, and let $e(G) := |E(G)|$. Let X be the set of $2n$ variables $X := \{x_{1,0}, x_{1,1}, \dots, x_{n,0}, x_{n,1}\}$. Let P_G be the nonnegative polynomial defined by

$$P_G := \sum_{a\in\{0,1\}^n} \left(\sum_{\{u,v\}\in E(G)} a_u a_v - 1 \right)^2 \prod_{i=1}^{n} x_{i,a_i}.$$

For every divisor k of n, let $Q_{k,G}$ be the degree-k polynomial in the $k2^{n/k}$ variables $\{\, x_{i,a} \mid i \in [k], a \in \{0,1\}^{n/k}\}$, defined by

$$Q_{k,G} := \sum_{a\in\{0,1\}^n} \left(\sum_{\{u,v\}\in E(G)} a_u a_v - 1 \right)^2 \prod_{i=1}^{k} x_{i,(a_{(i-1)n/k+1},\dots,a_{in/k})}.$$

Note that $P_G = Q_{n,G}$. In the other direction, by applying the substitution

$$x_{i,a} \leftarrow \prod_{j\in[n/k]} x_{(i-1)n/k+j,a_j} \tag{3}$$

to the polynomial $Q_{k,G}$ we obtain the polynomial P_G. We start by proving some upper bounds for the polynomials $Q_{k,G}$.

Lemma 2.1. For all G, k as above, the polynomial $Q := Q_{k,G}$ has a $\Sigma\Pi\Sigma$ formula of size $O(e(G)^2 k2^{n/k})$.

Proof. For every $p \in \mathbb{N}$, let

$$Q_p := \sum_{a\in\{0,1\}^n} \left(\sum_{\{u,v\}\in E(G)} a_u a_v \right)^p \prod_{i=1}^{k} x_{i,(a_{(i-1)n/k+1},\dots,a_{in/k})}.$$

We can write Q as

$$\begin{aligned} Q &= \sum_{a\in\{0,1\}^n} \left(\left(\sum_{\{u,v\}\in E(G)} a_u a_v \right)^2 - 2 \left(\sum_{\{u,v\}\in E(G)} a_u a_v \right) + 1 \right) \prod_{i=1}^{k} x_{i,(a_{(i-1)n/k+1},\dots,a_{in/k})} \\ &= Q_2 - 2Q_1 + Q_0. \end{aligned}$$

First,

$$Q_0 = \prod_{i=1}^{k} \left(\sum_{a \in \{0,1\}^{n/k}} x_{i,a} \right).$$

Second,

$$\begin{aligned} Q_1 &= \sum_{\{u,v\} \in E(G)} \sum_{a \in \{0,1\}^n} a_u a_v \prod_{i=1}^{k} x_{i,(a_{(i-1)n/k+1},\ldots,a_{in/k})} \\ &= \sum_{\{u,v\} \in E(G)} Q_{1,uv} \end{aligned}$$

where

$$Q_{1,uv} := \sum_{a \in \{0,1\}^n} a_u a_v \prod_{i=1}^{k} x_{i,(a_{(i-1)n/k+1},\ldots,a_{in/k})}.$$

For every $u \in [n]$, let $(i_u, j_u) \in [k] \times [n/k]$ be the unique pair such that

$$u = (i_u - 1)n/k + j_u.$$

If $i_u \neq i_v$, then

$$\begin{aligned} Q_{1,uv} = &\left(\sum_{a \in \{0,1\}^{n/k}} a_{j_u} x_{i_u,a} \right) \cdot \left(\sum_{a \in \{0,1\}^{n/k}} a_{j_v} x_{i_v,a} \right) \\ &\cdot \prod_{i \in [k] \setminus \{i_u,i_v\}} \left(\sum_{a \in \{0,1\}^{n/k}} x_{i,a} \right). \end{aligned}$$

If $i_u = i_v$, then

$$Q_{1,uv} = \left(\sum_{a \in \{0,1\}^{n/k}} a_{j_u} a_{j_v} x_{i_u,a} \right) \cdot \prod_{i \in [k] \setminus \{i_u\}} \left(\sum_{a \in \{0,1\}^{n/k}} x_{i,a} \right).$$

Third,

$$Q_2 = \sum_{\{u,v\},\{r,s\} \in E(G)} Q_{2,uv,rs}$$

where

$$Q_{2,uv,rs} := \sum_{a \in \{0,1\}^n} a_u a_v a_r a_s \prod_{i=1}^{k} x_{i,(a_{(i-1)n/k+1},\ldots,a_{in/k})}.$$

Similarly to the cases for $Q_{1,uv}$ above, we can compute $Q_{2,uv,rs}$ using a $\Pi\Sigma$ monotone formula of size $O(k2^k)$. ■

Corollary 2.2. For all G as above, the polynomial P_G has a $\Sigma\Pi\Sigma$ formula of size $O(e(G)^2 n)$.

Proof. This follows from Lemma 2.1 using k equals n. ■

2.2 Monotone arithmetic complexity and communication complexity

Our monotone arithmetic circuit lower bound in based on communication complexity. This connection was utilized in several works [RY11, Juk15, Sri20]. We start by defining the *multipartition rectangle number* of a Boolean function and prove (for completeness) how it is related to the monotone arithmetic complexity of a polynomial.

Let X be a finite set. We say that $R \subseteq \mathcal{P}(X)$ is a *rectangle* if there is a partition $\{Y, Z\}$ of X such that

$$R = \{\, S \cup T \mid S \in \mathcal{Y}, T \in \mathcal{Z} \}$$

for some $\mathcal{Y} \subseteq \mathcal{P}(Y)$ and $\mathcal{Z} \subseteq \mathcal{P}(Z)$. We say that a rectangle is *balanced* if $|Y|, |Z| \in [1/3, 2/3) \cdot |X|$.

For every $a \in \{0,1\}^n$, let

$$\mathrm{supp}(a) := \{\, i \in [n] \mid a_i = 1\}$$

and for $S \subseteq \{0,1\}^n$, let

$$\mathrm{supp}(S) := \{\, \mathrm{supp}(a) \mid a \in S\} \subseteq \mathcal{P}([n]).$$

The *multipartition rectangle number* $\mathrm{mprn}(f)$ of $f : \{0,1\}^n \to \{0,1\}$ is the minimum number r such that there is a family $\{R_i\}_{i \in [r]}$ of balanced rectangles (with respect to $[n]$) such that

$$\mathrm{supp}(f^{-1}(1)) = \bigcup_{i \in [r]} R_i.$$

The connection between arithmetic complexity and rectangle numbers is established as follows. Let $X := \{x_1, \ldots, x_m\}$ and let $\mathcal{X} := \{X_1, \ldots, X_n\}$ be a partition of X to n parts. We say that a polynomial P is *$\mathcal{X}$-set-multilinear* if every monomial of P has exactly one element of X_i for every $i \in [n]$. We say that a pair (g, h) of polynomials is a *monotone pair* (with respect to $\mathcal{X}$) if there is a partition $\{Y, Z\}$ of X such that $\mathcal{X}$ is a refinement of $\{Y, Z\}$ and $g \in \mathbb{R}[Y]$ and $h \in \mathbb{R}[Z]$ and g and h are monotone polynomials. We say that a nonnegative pair (g, h) is *balanced* if $|\mathcal{X}_Y|, |\mathcal{X}_Z| \in [1/3, 2/3) \cdot |\mathcal{X}|$ where for every $W \subseteq X$,

$$\mathcal{X}_W := \{\, B \in \mathcal{X} \mid B \subseteq W\}.$$

In words, the number of parts of $\mathcal{X}$ in each of Y, Z is balanced.

The following structural lemma was proved in [RY11] for general (not necessarily set-multilinear) monotone circuits and in [Yeh19] for ordered polynomials. The proof for the set-multilinear case below is basically the same.

Lemma 2.3. If P is a monotone $\mathcal{X}$-set-multilinear polynomial computed by a monotone circuit of size s then there is a family of balanced monotone pairs $\{(g_i, h_i)\}_{i \in [s]}$ such that

$$P = \sum_{i \in [s]} g_i h_i,$$

and each product $g_i h_i$ only contains monomials from P.

Proof sketch. In a nutshell, the proof uses that monotone circuits for set-multilinear polynomials are syntactically set-multilinear, and the lemma follows by induction on the number of edges by locating a single product gate that computes a balanced pair. ■

Now, let X be the set of $2n$ variables $X := \{x_{1,0}, x_{1,1}, \ldots, x_{n,0}, x_{n,1}\}$ and $\mathcal{X} := \{X_1, \ldots, X_n\}$ be its partition with $X_i := \{x_{i,0}, x_{i,1}\}$. For any $\mathcal{X}$-set-multilinear monomial $m := \prod_{i=1}^n x_{i,a_i}$, let

$$\mathcal{A}(m) := \{\, i \in [n] \mid a_i = 1\}.$$

For a $\mathcal{X}$-set-multilinear polynomial P, let

$$\mathcal{A}(P) := \{\, \mathcal{A}(m) \mid m \in \mathrm{supp}(P)\} \subseteq \mathcal{P}([n]).$$

Lemma 2.4. Let $f : \{0,1\}^n \to \{0,1\}$ and let P be a monotone $\mathcal{X}$-set-multilinear polynomial. If $\mathrm{supp}(f^{-1}(1)) = \mathcal{A}(P)$, then

$$\mathrm{S}^+(P) \geq \mathrm{mprn}(f).$$

Proof. Let C be a monotone circuit of size s computing P. By Lemma 2.3, we know that there is a family $\{(g_i, h_i)\}_{i\in[s]}$ of balanced monotone pairs such that

$$P = \sum_{i\in[s]} g_i h_i.$$

This decomposition of P implies the following decomposition of its support

$$\mathrm{supp}(P) = \bigcup_{i\in[s]} \{\, m_1 m_2 \mid m_1 \in \mathrm{supp}(g_i), m_2 \in \mathrm{supp}(h_i)\}.$$

Thus, we get the following decomposition

$$\mathcal{A}(P) = \bigcup_{i\in[s]} \{\, \mathcal{A}(m_1) \cup \mathcal{A}(m_2) \mid m_1 \in \mathrm{supp}(g_i), m_2 \in \mathrm{supp}(h_i)\},$$

which, by hypothesis, implies

$$\mathrm{supp}(f^{-1}(1)) = \bigcup_{i\in[s]} \{\, \mathcal{A}(m_1) \cup \mathcal{A}(m_2) \mid m_1 \in \mathrm{supp}(g_i), m_2 \in \mathrm{supp}(h_i)\}.$$

The final observation is that for any balanced monotone pair (g, h), the set

$$R := \{\, \mathcal{A}(m_1) \cup \mathcal{A}(m_2) \mid m_1 \in \mathrm{supp}(g), m_2 \in \mathrm{supp}(h)\}$$

is a balanced rectangle of $[n]$. Together with the decomposition above, this concludes the proof of the lemma.

∎

2.3 A multipartition communication complexity lower bound

In this section, we prove the multipartition communication complexity lower bounds, which in turn implies the monotone circuit lower bound. The problem we consider is defined over a graph G with vertex-set $[n]$. For a subset U of the vertices of G, we denote by $G[U]$ the induced graph of G on U. Denote by $f_G : \{0,1\}^n \to \{0,1\}$ the function defined by

$$f_G(a) = 1 \iff e(G[\mathrm{supp}(a)]) \neq 1.$$

Note that the support of the polynomial P_G is the same as $f_G^{-1}(1)$.

Our lower bounds hold as long as G is an *expander graph*, since this is required by Lemma 9 from [Sri20], which we restated as Lemma 2.7. For concreteness, the definition of expander graph used by [Sri20] is that G is a d-regular graph with the second largest eigenvalue of its adjacency matrix at most $d^{0.75}$. Our main lemma is the following.

Lemma 2.5. If G is an expander graph with vertex-set $[n]$, then

$$\mathrm{mprn}(f_G) \geq 2^{\Omega(n)}.$$

The main ingredient in the lower bound is the following corruption-like lemma. It follows from a communication complexity perspective of a result by Kaibel and Weltge [KW15]; see Lemma 5.10 in Roughgarden's lecture notes [Rou15].

Lemma 2.6. For $\ell \in \mathbb{N}$, we define the 2ℓ-variate *different-from-1 disjointness function* $\mathrm{DISJ}_\ell^{\neq 1}$ as

$$\mathrm{DISJ}_\ell^{\neq 1}(b_1 \dots, b_\ell, c_1, \dots, c_\ell) = 1 \Leftrightarrow |\,\mathrm{supp}((b_1, \dots, b_\ell)) \cap \mathrm{supp}((c_1, \dots, c_\ell))| \neq 1.$$

Let $\ell \in [n]$ and $g := \mathrm{DISJ}_\ell^{\neq 1}$. For every $i \in [\ell]$, choose independently and uniformly at random an element $(b_i, b_{\ell+i})$ of the set $\{(0,0),(1,0),(0,1)\}$, and let $b := (b_1, \dots, b_{2\ell})$ be the corresponding random assignment over 2ℓ-variables. If R is a rectangle of $[2\ell]$ with partition $\{\{1, \dots, \ell\}, \{\ell + 1, \dots, 2\ell\}\}$ such that $R \subseteq \mathrm{supp}(g^{-1}(1))$, then

$$\Pr_b[b \in R] < 2^{-\ell/2}.$$

Proof. As the proof of Lemma 5.10 in the notes [Rou15] works for any unique-disjointness 1-rectangle, this lemma follows from the fact that any different-from-one disjointness 1-rectangle is also a unique-disjointness 1-rectangle. ∎

Next, we need to define a "hard distribution" on $f_G^{-1}(1)$ such that the measure of all balanced 1-rectangles contained in $f_G^{-1}(1)$ of is small. While the function f_G bears some similarity to the different-from-1 disjointness function, this does not immediately follow from the lemma above, because we do not a priori know the partition we need to deal with.

Consider the following algorithm, as defined in [Sri20], to sample a random matching from G: for $m \in [n]$ given as input,

1. Set $M \leftarrow \varnothing$.
2. For $i = 1, 2, \dots, m$, do the following:
   (a) Remove all vertices from G that are at distance at most 2 from any vertex in M. Let G_i be the resulting graph.
   (b) Choose a uniformly random edge e_i from $E(G_i)$, and add it to M.
3. Output M.

Now given a matching M, we use the following procedure to define a random input $a \in \{0,1\}^n$ to f_G:

1. For each $u \in [n] \setminus V(M)$, set $a_u = 0$.
2. For each $\{u, v\} \in M$, choose independently and uniformly at random an element (a_u, a_v) from the $\{(0,0), (1,0), (0,1)\}$.

Note that $e(G[\mathrm{supp}(a)]) = 0$ by our choice of a, implying that the procedure above defines a probability distribution over $f_G^{-1}(1)$. Hence, Lemma 2.5 immediately follows from the claim that for every balanced rectangle $R \subseteq f_G^{-1}(1)$,

$$\Pr_a[\mathrm{supp}(a) \in R] \leq 2^{-\Omega(n)}.$$

So it remains to prove this inequality.

The useful properties of the random input a rely on the following properties of the random matching M, which were stated in Lemma 9 from [Sri20].

Lemma 2.7. If G is an expander with vertex-set $[n]$, then there is a constant $\alpha > 0$ such that, for $m := \lceil \alpha n \rceil$ and for M being a random matching sampled by the above algorithm using m as input, the following hold:

1. M is an induced matching of cardinality m.
2. For every balanced partition $\{A, B\}$ of $[n]$,

$$\Pr_M[|M \cap E_G(A, B)| \leq \gamma m] \leq \exp(-\gamma m),$$

for an absolute constant $\gamma > 0$.

Let $\{A, B\}$ be any valid partition witnessing that R is a balanced rectangle of $[n]$. By Lemma 2.7, we know that M is an induced matching of G and, with probability at least $1 - 2^{-\Omega(n)}$, we have

$$|M \cap E_G(A, B)| \geq \gamma m =: s'.$$

For a fixed induced matching

$$E := \{\{u_1, v_1\}, \ldots, \{u_s, v_s\}\} \subseteq E(G)$$

with $s \geq s'$ and $u_i \in A$ and $v_i \in B$ for every $i \in [s]$, we condition on the event that the random matching M satisfies

$$M \cap E_G(A, B) = E.$$

We can now define the following random variables: for every $i \in [s]$, let

$$b_i := a_{u_i} \text{ and } b_{s+i} := a_{v_i},$$

where $a \in \{0,1\}^n$ is the random assignment associated to M. Let

$$b := (b_1, \ldots, b_{2s})$$

be the random $2s$-variable Boolean assignment obtained by the definition above. By the definition of the assignment a, we obtain that each pair (b_i, b_{s+i}) is chosen independently and uniformly at random from the set $\{(0,0),(1,0),(0,1)\}$. Let g be the $2s$-variate different-from-one disjointness function. We observe that

$$f_G(a) = 1 \iff g(b) = 1$$

for the assignments a and b defined above. Also note that b is a random assignment to g that satisfies the assumptions of Lemma 2.6, and, furthermore, $\mathrm{supp}(a) \in R$ implies that $\mathrm{supp}(b) \in R_E$, where

$$R_E := \{\, S \cup T \mid S \subseteq \{1,\dots,s\}, T \subseteq \{s+1,\dots,2s\}, (\{\, u_i \mid i \in S\} \cup \{\, v_{i-s} \mid i \in T\}) \in R\}.$$

As R is an rectangle with partition $\{A, B\}$ and we have $\{\, u_i \mid i \in S\} \subseteq A$ and $\{\, v_{i-s} \mid i \in T\} \subseteq B$, we get that R_E is a rectangle of $[2s]$ with partition $\{\{1,\dots,s\},\{s+1,\dots,2s\}\}$. Moreover, we have $R_E \subseteq \mathrm{supp}(g^{-1}(1))$, since any $S \cup T \in R_E$ satisfies

$$\{\, u_i \mid i \in S\} \cup \{\, v_{i-s} \mid i \in T\} \in R \subseteq \mathrm{supp}(f^{-1}(1)),$$

which implies that $g(1_S, 1_T) = 1$, where we have identified S and T with their corresponding indicator vectors.

Thus, R_E is a rectangle satisfying the hypothesis of Lemma 2.6, so we can apply this lemma to obtain that

$$\Pr_a[b \in R_E] \le 2^{-\Omega(s)} = 2^{-\Omega(n)},$$

which implies that

$$\begin{aligned}\Pr_{M,a}[a \in R \mid M \cap E_G(A,B) = E] &\le \Pr_{M,a}[b \in R_E \mid M \cap E_G(A,B) = E] \\ &= \Pr_a[b \in R_E] \le 2^{-\Omega(n)}.\end{aligned}$$

Therefore, we get

$$\begin{aligned}\Pr_{M,a}[a \in R] = &\Pr_{M,a}[a \in R, |M \cap E_G(A,B)| < s'] \\ &+ \sum_{E \subseteq E(G): |E| \ge s'} \Pr_{M,a}[a \in R \mid M \cap E_G(A,B) = E] \cdot \Pr_M[M \cap E_G(A,B) = E] \\ &\le 2^{-\Omega(n)}.\end{aligned}$$

2.4 Monotone arithmetic circuit lower bounds

Let us now prove our monotone arithmetic circuit lower bounds.

Corollary 2.8 (Theorem 1.1)**.** If G is an n-vertex expander graph, then $\mathrm{S}^+(P_G) \ge 2^{\Omega(n)}$.

Proof. We have that

$$\begin{aligned}\mathcal{A}(P_G) &= \left\{ \mathcal{A}\left(\prod_{i=1}^{n} x_{i,a_i}\right) \middle| \, a \in \{0,1\}^n, \sum_{\{u,v\} \in E(G)} a_u a_v \neq 1 \right\} \\ &= \{\, \mathrm{supp}(a) \mid a \in \{0,1\}^n, e(G[\mathrm{supp}(a)]) \neq 1\} \\ &= \mathrm{supp}(f_G^{-1}(1)),\end{aligned}$$

thus, by Lemmas 2.4 and 2.5, we obtain

$$\mathrm{S}^+(P_G) \geq \mathrm{mprn}(f_G) \geq 2^{\Omega(n)}.$$

■

Corollary 2.9 (Theorem 1.2)**.** Let $k \in \mathbb{N}$ and $n \in \mathbb{N}$. Let G be a graph with nk vertices. If G is an expander graph, then

$$\mathrm{S}^+(Q_{k,G}) \geq (m/k)^{\Omega(k)},$$

where $m := k2^n$ is the number of variables in the degree-k polynomial $Q_{k,G}$.

Proof. By a substitution (Equation 3), we can convert any monotone circuit of size s computing $Q_{k,G}$ into a monotone circuit of size $s + O(mn)$ computing P_G. Thus, by Corollary 2.8, we have $2^{\Omega(nk)} \leq s + O(mn)$, so

$$s \geq 2^{\Omega(nk)} = (2^n)^{\Omega(k)} = (m/k)^{\Omega(k)}.$$

Therefore, we obtain

$$\mathrm{S}^+(Q_{k,G}) \geq (m/k)^{\Omega(k)}.$$ ■

3 A criterion for monotone Boolean circuit lower bounds

The goal here is to describe a criterion for proving monotone Boolean circuit lower bounds for a function $f : \{0,1\}^n \to \{0,1\}$. The criterion is based on the existence of distributions $\mathcal{D}_0$ and $\mathcal{D}_1$ over $\{0,1\}^n$ satisfying some properties. We start with two important definitions.

Definition 3.1 (Sunflower)**.** For $S \subseteq [n]$, let t_S denote the function

$$t_S(x) := \bigwedge_{i \in S} x_i$$

where by convention $t_\varnothing \equiv 1$.

We say that a family of sets $\mathcal{S} \subseteq \mathcal{P}([n])$ is a $(\mathcal{D}, \varepsilon)$-*sunflower* if $|\mathcal{S}| \geq 2$ and

$$\Pr_{x \sim \mathcal{D}}\left[\exists S \in \mathcal{S} : t_{S \setminus K}(x) = 1\right] > 1 - \varepsilon, \tag{4}$$

where $K := \bigcap_{S \in \mathcal{S}} S$. The family $\mathcal{S}$ is called ℓ-uniform if $|S| = \ell$ for every $S \in \mathcal{S}$. Let $r(\mathcal{D}, \ell, \varepsilon)$ be the minimum integer r such that every ℓ-uniform family of sets of size at least r^ℓ contains a $(\mathcal{D}, \varepsilon)$-sunflower (if no such r exists then it is ∞). ◀

Definition 3.2 (Spread)**.** For an integer t and $q > 0$, a distribution $\mathcal{D}$ over $\{0,1\}^n$ is *t-wise q-spread* if, for every $A \subseteq [n]$ such that $|A| \leq t$,

$$\Pr_{x \sim \mathcal{D}}[t_A(x) = 1] \leq q^{-|A|}.$$

◀

We show that a monotone circuit lower bound for a function $f \colon \{0,1\}^n \to \{0,1\}$ follows immediately from a spreadness bound on a distribution $\mathcal{D}_1$ of accepting inputs, and a sunflower bound on a distribution $\mathcal{D}_0$ of rejecting inputs.

Theorem 3.3 (A lower bound criterion)**.** Let $n \in \mathbb{N}$ and let $f : \{0,1\}^n \to \{0,1\}$ be a monotone function. Let $t \in \mathbb{N}$ and $q \in (0,1)$. Let $\mathcal{D}_0, \mathcal{D}_1$ be distributions over $\{0,1\}^n$ such that $\mathcal{D}_1$ is t-wise q-spread, and let

$$\alpha := \min\left\{ \Pr_{x\sim\mathcal{D}_0}[f(x)=0],\ \Pr_{x\sim\mathcal{D}_1}[f(x)=1] \right\}.$$

Let $w \in \mathbb{N}$ such that $w \le t/2$, and define

$$r_w := \max_{\ell\in[2w]} r(\mathcal{D}_0, \ell, \alpha n^{-3w}).$$

If $8r_w \le q \le r_w n$, then any Boolean monotone circuit computing f has size at least

$$\left(\frac{c\alpha q}{r_w}\right)^w$$

where c is a universal positive constant.

We prove Theorem 3.3 using the approximation method of Razborov [Raz85], generalising the "tailored sunflower" approach of recent works [CKR22, BM25, CGR+25] which began with [Ros14]. Certain sunflower criterions have previously appeared in [Cav20, BM25]. Our presentation of the method is similar to [CGR+25], with the difference that we consider arbitrary distributions. The following is the main lemma.

Lemma 3.4. Let $n \in \mathbb{N}$ and let $f : \{0,1\}^n \to \{0,1\}$ be a monotone function. Let $t \in \mathbb{N}$ and $q \in (0,1)$. Let $\mathcal{D}_0, \mathcal{D}_1$ be distributions over $\{0,1\}^n$ such that $\mathcal{D}_1$ is t-wise q-spread and $\mathrm{supp}(\mathcal{D}_i) \subseteq f^{-1}(i)$ for every $i \in \{0,1\}$. Let $w \in \mathbb{N}$ such that $w \le t/2$, and define

$$r_w := \max_{\ell\in[2w]} r(\mathcal{D}_0, \ell, n^{-3w}).$$

If $8r_w \le q \le r_w n$, then any Boolean monotone circuit computing f has size at least

$$\left(\frac{cq}{r_w}\right)^w$$

where c is a universal positive constant.

Before proving our main lemma, let us use it to prove Theorem 3.3.

Proof of Theorem 3.3. For every $b \in \{0,1\}$, we have $\Pr_{x\sim\mathcal{D}_b}[f(x)=b] \ge \alpha > 0$. Let $\mathcal{D}_b^*$ be the distribution $\mathcal{D}_b$ conditioned on the event $f(x)=b$. First, the distribution $\mathcal{D}_1^*$ is t-wise (αq)-spread because for every $A \in \binom{[n]}{\le t}$,

$$\Pr_{x\sim\mathcal{D}_1^*}[t_A(x)=1] \le \frac{1}{\alpha}\cdot \Pr_{x\sim\mathcal{D}_1}[t_A(x)=1] \le (\alpha q)^{-|A|}.$$

Second, every $(\mathcal{D}_0, \alpha\varepsilon)$-sunflower $\mathcal{S}$ is a $(\mathcal{D}_0^*, \varepsilon)$-sunflower because

$$\Pr_{x\sim\mathcal{D}_0^*}\left[\forall S\in\mathcal{S}\ \ t_{S\setminus K}(x)=0\right] \le \frac{1}{\alpha}\cdot \Pr_{x\sim\mathcal{D}_0}\left[\forall S\in\mathcal{S}\ \ t_{S\setminus K}(x)=0\right] < \frac{\varepsilon\alpha}{\alpha} = \varepsilon,$$

so $r(\mathcal{D}_0^*, \ell, \varepsilon) \le r(\mathcal{D}_0, \ell, \alpha\varepsilon)$. The result now follows by Lemma 3.4. ∎

For the rest of this section, we prove Lemma 3.4. Let $\mathcal{D}_0, \mathcal{D}_1, q, t, w, \ell$ be as in the assumptions of the lemma. We denote by $\mathcal{D}$ the distribution

$$\mathcal{D} := (\mathcal{D}_0 + \mathcal{D}_1)/2.$$

Our goal is to approximate a given monotone circuit by a monotone DNF formula

$$F_{\mathcal{S}} := \bigvee_{S \in \mathcal{S}} t_S$$

for a given family of sets $\mathcal{S} \subseteq \mathcal{P}([n])$ satisfying some properties.

We say that $F_{\mathcal{S}}$ is r*-small* if, for every $\ell \in [n]$,

$$|\mathcal{S} \cap \binom{[n]}{\ell}| \leq r^\ell.$$

We say that $F_{\mathcal{S}}$ has *width* w if $|S| \leq w$ for every $S \in \mathcal{S}$. Finally, we say that $F_{\mathcal{S}}$ is a (w,r)-DNF if it is both r-small and has width w. The approximation method now proceeds in the following two steps.

Claim 3.5. There is a universal constant $c \in (0,1)$ such that if a monotone circuit of size at most $(cq/r_w)^w$ computes f then there is a (w, r_w)-DNF F such that

$$\Pr_{x \sim \mathcal{D}}[F(x) = f(x)] \geq 0.9.$$

Claim 3.6. For every $\delta \in (0, 1/2)$ and for every $(w, \delta q)$-DNF F, we have

$$\Pr_{x \sim \mathcal{D}}[F(x) = f(x)] \leq 1/2 + 2\delta.$$

Before proving the two claims, we prove the main lemma.

Proof of Lemma 3.4. The proof uses the two claims above. When $r_w = \infty$, the theorem trivially holds so assume that $r_w < \infty$. Assume (towards a contradiction) that f can be computed by a monotone circuit of size $\leq (cq/2r_w))^w$ where $c > 0$ is the constant from Claim 3.5. As $8r_w \leq q$ by assumption, Claim 3.5 implies that there is a $(w, q/8)$-DNF such that

$$\Pr_{x \sim \mathcal{D}}[F(x) = f(x)] \geq 0.9,$$

which contradicts Claim 3.6. ■

Proof of Claim 3.6. Let $F := F_S$ be a $(w, \delta q)$-DNF. If $F \equiv \mathbb{1}$ (i.e., $\varnothing \in \mathcal{S}$), the claim is true by definition of $\mathcal{D}$ and the fact that $\mathcal{D}_b$ is supported in $f^{-1}(b)$ for every $b \in \{0,1\}$. Otherwise, note that $F(x) = 1$ only if there exists $S \in \mathcal{S}$ such that $t_S(x) = 1$. As F is δq-small and has width w and $\mathcal{D}_1$ is t-wise q-spread with $w \leq t/2$, we have

$$\Pr_{x \sim \mathcal{D}_1}[F(x) = 1] \leq \sum_{k \in [w]} \sum_{S \in \mathcal{S} \cap \binom{[n]}{k}} \Pr_{x \sim \mathcal{D}_1}[t_S(x) = 1] \leq \sum_{k \in [w]} (\delta q)^k q^{-k} \leq 2\delta. \qquad ■$$

Proof of Claim 3.5. For convenience, let $r := r_w$ and let $\varepsilon := n^{-3w}$. Let C be a monotone circuit of size at most $(cq/r)^w$ computing f where $c > 0$ is to be determined. We will construct a (w, r)-DNF for C gate-by-gate, inductively, starting at the input gates until we reach the output gate. Every input variable is already a (w, r)-DNF. As we naively combine our inductively constructed (w, r)-DNFs, the number of terms might increase, potentially violating r-smallness. In order to maintain smallness of our DNF, we approximate the naive combination by the following procedure. The subsequent claim summarises the properties of the resulting DNF.

Algorithm 1 Plucking procedure $\mathsf{pluck}(\mathcal{S})$

1: **while** $\exists \ell \in [2w]\colon |\mathcal{S} \cap \binom{[n]}{\ell}| > r^\ell$ **do**
2: Let $\mathcal{S}' \subseteq \mathcal{S} \cap \binom{[n]}{\ell}$ be a $(\mathcal{D}_0, \varepsilon)$-sunflower with core K
3: Let $\mathcal{S} \leftarrow \{K\} \cup \{S \in \mathcal{S} : K \not\subseteq S\}$
4: **end while**

Note that Line 2 of the algorithm is always possible by the choice of r.

Claim 3.7. If $F_\mathcal{S}$ has width $2w$, then $F_{\mathsf{pluck}(\mathcal{S})}$ is a $(2w, r)$-DNF with $F_{\mathsf{pluck}(\mathcal{S})} \geq F_\mathcal{S}$ and

$$\Pr_{x\sim\mathcal{D}_0}[F_{\mathsf{pluck}(\mathcal{S})}(x) > F_\mathcal{S}(x)] \leq n^{-w}.$$

Proof. Since the core K is contained in all sets in $\mathcal{S}'$ and $|\mathcal{S}'| > 1$, the size of $\mathcal{S} \cap \binom{[n]}{\ell}$ is necessarily reduced at Line 3, and eventually we obtain an r-small family. Therefore, as $\mathcal{S}$ has width $2w$, the algorithm ends in at most $\sum_{i=0}^{2w} \binom{n}{i} \leq n^{2w}$ iterations. It follows that $F_{\mathsf{pluck}(\mathcal{S})}$ is a $(2w, r)$-DNF.

Because the core belongs to all sets in the sunflower, we get $F_{\mathsf{pluck}(\mathcal{S})} \geq F_\mathcal{S}$. It remains to bound the error on $\mathcal{D}_0$ incurred in Line 3. Such an error happens only if $t_K(x) = 1$ and, for all $S \in \mathcal{S}$, we have $t_S(x) = 0$. Thus, the error of a single iteration (sunflower plucking) can be bounded by

$$\Pr_{x\sim\mathcal{D}_0}[F_{\mathsf{pluck}(\mathcal{S})}(x) > F_\mathcal{S}(x)] \leq \Pr_{x\sim\mathcal{D}_0}[\forall S \in \mathcal{S},\ t_{S\setminus K}(x) = 0] \leq \Pr_{x\sim\mathcal{D}_0}[\forall S \in \mathcal{S}',\ t_{S\setminus K}(x) = 0] < \varepsilon,$$

because $\mathcal{S}'$ is a $(\mathcal{D}_0, \varepsilon)$-sunflower. As there are at most n^{2w} iterations and $\varepsilon = n^{-3w}$, the total error is at most n^{-w}. ■

As remarked above, every input gate is already a (w, r)-DNF. Going over the gates of the circuit one-by-one (according to the circuit-order), we will inductively construct a (w, r)-DNF for each gate of the circuit. Let g be a gate of the form $g := g_1 \circ g_2$ for a binary operation $\circ \in \{\vee, \wedge\}$. Suppose we have inductively constructed two (w, r)-DNFs $F_\mathcal{S}$ and $F_\mathcal{T}$ for g_1 and g_2 respectively. We will construct a DNF F_g such that

$$E_{1,g} := \Pr_{x\sim\mathcal{D}_1}[F_g(x) < (F_\mathcal{S} \circ F_\mathcal{T})(x)] \ \leq\ (2r/q)^w \tag{5}$$

and

$$E_{0,g} := \Pr_{x\sim\mathcal{D}_0}[F_g(x) > (F_\mathcal{S} \circ F_\mathcal{T})(x)] \ \leq\ (2r/q)^w. \tag{6}$$

If $\circ = \vee$, we approximate $F_\mathcal{S} \vee F_\mathcal{T}$ by letting $F_g := F_{\mathsf{pluck}(\mathcal{S}\cup\mathcal{T})}$. Note that, by Claim 3.7, we have that $F_{\mathsf{pluck}(\mathcal{S}\cup\mathcal{T})}$ is a (w, r)-DNF and

$$F_{\mathsf{pluck}(\mathcal{S}\cup\mathcal{T})} \geq F_{\mathcal{S}\cup\mathcal{T}} = F_\mathcal{S} \vee F_\mathcal{T}.$$

Thus plucking introduces no errors on $\mathcal{D}_1$, implying (5). Moreover, plucking incurs errors at most $n^{-w} \leq (2r/q)^w$ errors on $\mathcal{D}_0$ by Claim 3.7 and $q \leq rn$, which implies (6).

If $\circ = \wedge$, we approximate g by first taking

$$\mathcal{F} := \mathsf{pluck}\left(\{S \cup T : S \in \mathcal{S},\, T \in \mathcal{T}\}\right).$$

By Claim 3.7, $F_{\mathcal{F}}$ is r-small since $|S \cup T| \leq 2w$ for every $S \in \mathcal{S}, T \in \mathcal{T}$. As in the $\vee$ case, this creates no error on $\mathcal{D}_1$ and the error on $\mathcal{D}_0$ is at most $n^{-w} \leq (2r/q)^w$. We now define F_g be removing all sets of width larger than w from $\mathcal{F}$. This removal does not create an error on $\mathcal{D}_0$, since $F_g \leq F_{\mathcal{F}}$. Since $\mathcal{F}$ is r-small, the q-spreadness of $\mathcal{D}_1$ implies that

$$\Pr_{x\sim\mathcal{D}_1}[F_g(x) > F_{\mathcal{F}}(x)] \leq \sum_{\ell=w+1}^{2w} \sum_{S \in \mathcal{F} \cap \binom{[n]}{\ell}} \Pr_{x\sim\mathcal{D}_1}[t_S(x) = 1] \leq \sum_{\ell=w+1}^{t} r^{\ell} q^{-\ell} \leq (2r/q)^w,$$

where we used that $q \geq 2r$. We have thus shown (5) and (6).

Finally, let F be the (w, r)-DNF of the output gate. By the union bound and Equation (5) ,

$$\Pr_{x\sim\mathcal{D}_1}[C(x) > F(x)] \leq \sum_{g} E_{1,g} \leq (cq/r)^w (2r/q)^w \leq (2c)^w \leq 0.1,$$

for $c = 0.05$. A similar bound holds for $\Pr_{x\sim\mathcal{D}_0}[C(x) < F(x)]$. It follows that

$$\Pr_{x\sim\mathcal{D}}[F(x) \neq f(x)] = \frac{1}{2}\Pr_{x\sim\mathcal{D}_1}[C(x) > F(x)] + \frac{1}{2}\Pr_{x\sim\mathcal{D}_0}[C(x) < F(x)] \leq 0.1.$$

■

4 Boolean separation

Recall that, given a matrix M with n rows and m columns over a field $\mathbb{F}$, the Boolean function $f_M : \{0,1\}^m \to \{0,1\}$ accepts an input $S \subseteq [m]$ if and only if $M[S]$ has full rank. In this section, we only consider $\mathbb{F} := \mathbb{F}_2$ in order to prove Theorem 1.5.

4.1 Nonmonotone low-depth circuit upper bounds

Let us first prove the upper bound part of Theorem 1.2.

Lemma 4.1 (Upper bound)**.** For any matrix $M \in \mathbb{F}^{n\times m}$, the function f_M is computed by a Boolean circuit of size $\mathrm{poly}(m)$ and depth $O(\log m)^2$.

Proof. On an input 1_S corresponding to the indicator vector of a set $S \subseteq [m]$, we need to compute the rank of the matrix $M[S]$. This is known to be computed by a Boolean circuit of size $\mathrm{poly}(m)$ and depth $O(\log m)^2$ [ABO99, Mul87]. ■

4.2 Monotone Boolean circuit lower bounds via well-behaved codes

Let us start by defining a special kind of linear binary codes, which are explicitly constructed in Section 4.3. We will only use the abstract properties of these codes to define probability distributions that allow us to apply Theorem 3.3 and obtain lower bounds for monotone circuits computing functions related to these codes.

Definition 4.2. For $M \in \mathbb{F}^{n \times m}$, we define the *(linear) code $\mathcal{C}_M$ of M* as

$$\mathcal{C}_M := \{ M^\top w \mid w \in \mathbb{F}^n \} \subseteq \mathbb{F}^m,$$

and define the *dual code $\mathcal{D}_M$ of M* as

$$\mathcal{D}_M := \{ w \in \mathbb{F}^m \mid Mw = 0 \} \subseteq \mathbb{F}^m.$$

We say that a code $\mathcal{C} \subseteq \mathbb{F}^m$ is a *linear code* if there is a rank n matrix $G \in \mathbb{F}^{n \times m}$, called a *generator matrix* of $\mathcal{C}$, such that

$$\mathcal{C} = \mathcal{C}_G.$$

For any linear code $\mathcal{C} \subseteq \mathbb{F}^m$, we say that a rank $m - n$ matrix $H \in \mathbb{F}^{(m-n) \times n}$ is a *parity check matrix* of $\mathcal{C}$ if

$$\mathcal{C} = \mathcal{D}_H,$$

and, in this case, we define the *dual code $\mathcal{C}^*$ of $\mathcal{C}$* as

$$\mathcal{C}^* := \mathcal{C}_H,$$

which is independent of the choice of a parity check matrix H. We define the *distance* of a linear code $\mathcal{C} \subseteq \mathbb{F}^m$ as

$$d(\mathcal{C}) := \min_{x \in \mathcal{C}, x \neq 0} |\{ i \in [m] \mid x_i \neq 0\}|,$$

and its *dual distance* as $d^*(\mathcal{C}) := d(\mathcal{C}^*)$. For every $d, t \in \mathbb{R}_{\geq 0}$, we say that $\mathcal{C}$ is *(d,t)-well-behaved* if $d(\mathcal{C}) > d$ and $d^*(\mathcal{C}) > t$. ◀

The main theorem of this section is the following.

Theorem 4.3. Let $n \in \mathbb{N}$, and $m := m(n) \in \mathbb{N}$, and $d := d(n) \in \mathbb{N}$, and $t := t(n) \in \mathbb{N}$. Let $\mathcal{C} \subseteq \mathbb{F}^m$ be any (d,t)-well-behaved linear code with generator matrix $M \in \mathbb{F}^{n \times m}$. If $2n < d$, then the monotone complexity of f_M is at least

$$\Omega \left(\frac{d}{n\sqrt{t}} \right)^{\sqrt{t}/(b \log m)},$$

where $b \in \mathbb{R}$ is a sufficiently large positive constant.

In Section 4.3, we are going to use this theorem with

$$m \approx n^{3/2} \log n, \text{ and } d \approx m - n, \text{ and } t \approx n.$$

The rest of this section is devoted to prove this theorem. Henceforth we assume that $\mathcal{C}$ and M are as in the assumption of the theorem. Let us denote by $M_i \in \mathbb{F}^n$ the i-th column of M for every $i \in [m]$. We first prove a few properties related to the rank of submatrices of M. Some of them are standard facts from coding theory, but we add the proof here for completeness.

Lemma 4.4 (Theorem 10 from Chapter 1 of [MS77])**.** For every $S \subseteq [m]$ with $|S| \leq t$, the set of columns of M indexed by S is linearly independent.

Proof. Let S be a subset of $[m]$ such that the set of columns of M indexed by S is linearly dependent. Then there is a nonzero vector $v \in \mathbb{F}^m$ such that $\mathrm{supp}(v) \subseteq S$, and

$$\sum_{i \in [m]} v_i M_i = 0.$$

Thus, $Mv = 0$, and $v \in \mathcal{D}_M$. Note that $\mathcal{D}_M = \mathcal{C}^*$, since, for any matrix $H \in \mathbb{F}^{(m-n)\times m}$ such that $\mathcal{C} = \mathcal{D}_H$, we have that

$$\mathcal{C}^* = \mathcal{C}_H \text{ and } HM^\top = 0.$$

By the definition of the distance of $\mathcal{C}^*$ and the fact that $\mathcal{C}$ is (d,t)-well-behaved, we get that

$$|S| \geq |\mathrm{supp}(v)| \geq d^*(\mathcal{C}) > t. \qquad \blacksquare$$

Lemma 4.5. For any subspace $V \subseteq \mathbb{F}^n$ of dimension $D < n$, and for u being a uniform random column of M, we have

$$\Pr_u [u \in V] \leq 1 - d/m =: \Delta.$$

Proof. As V has dimension $D \leq n-1$, there is a nonzero vector $c \in \mathbb{F}^n$ such that, for every $v \in V$, we have $v^\top c = 0$. Let $C \subseteq [m]$ be the set of the columns of M that are in V, and let

$$w := M^\top c \in \mathcal{C}_M = \mathcal{C}.$$

Note that, for every $i \in C$, we have $w_i = M_i^\top c = 0$. Thus, by the definition of the distance of $\mathcal{C}$ and the fact that $\mathcal{C}$ is (d,t)-well-behaved, we get that

$$m - |C| \geq |\mathrm{supp}(w)| \geq d(\mathcal{C}) > d.$$

and, consequently,

$$\Pr_u [u \in V] \leq \frac{|C|}{m} \leq 1 - \frac{d}{m}. \qquad \blacksquare$$

Lemma 4.6. For any $N := N(m) \in \mathbb{N}$ with $N \leq m$, and for a random set S distributed uniformly in $\binom{[m]}{N}$, we have

$$\Pr_S [M[S] \text{ is not full rank}] \leq 2^n \Delta^N.$$

Proof. Let $\mathrm{span}(S) := \mathrm{span}(\{ M_i \mid i \in S\})$, where M_i is the column of M indexed by $i \in [m]$. Note that the rank of $M[S]$ is at most $n-1$ if and only if there is a subspace $V \subseteq \mathbb{F}^n$ of dimension $n-1$ such that $\mathrm{span}(S) \subseteq V$. Hence,

$$\begin{aligned}
\Pr_S [M[S] \text{ is not full rank}] &\leq \Pr_S [\exists V \subseteq \mathbb{F}_2^n \text{ s.t. } \dim(V) = n-1 \text{ and } \mathrm{span}(S) \subseteq V] \\
&\leq \sum_{V \subseteq \mathbb{F}_2^n, \dim(V) = n-1} \Pr_S [S \subseteq V].
\end{aligned}$$

By Proposition 1.7.2 in Stanley [Sta97], the number of $n-1$ dimensional subspaces of $\mathbb{F}^n$ is

$$\frac{(2^n - 1)(2^n - 2)\cdots(2^n - 2^{n-2})}{(2^{n-1} - 1)(2^{n-1} - 2)\cdots(2^{n-1} - 2^{n-2})} \leq 2^n - 1.$$

For any $(n-1)$-dimensional subspace V of $\mathbb{F}^n$, let $C_V \subseteq [m]$ be the set of columns of M that are contained in V. By Lemma 4.5, we know that $c := |C_V| \leq \Delta m$, which implies that

$$\Pr_S [S \subseteq V] \leq \Pr_S [S \subseteq C_V] = \frac{\binom{c}{N}}{\binom{m}{N}} \leq \left(\frac{c}{m}\right)^N \leq \Delta^N.$$

Therefore,

$$\Pr_S [M[S] \text{ is not full rank}] \leq 2^n \Delta^N.$$

■

Now our goal is to apply Theorem 3.3 to prove a lower bound for f_M. So we first define two probability distributions that, intuitively, are hard to distinguish for monotone Boolean circuits of small size.

Definition 4.7. We define the *distribution* $a \sim D_1$ by sampling a uniformly random $a \in \{0,1\}^m$ of Hamming weight

$$N := n\lceil m/d \rceil. \tag{7}$$

We define the *distribution* $a \sim D_0$ by sampling a uniformly random $u \in \mathbb{F}_2^n$ and, for every $j \in [m]$, we set $a_j := 1$ if $\langle M[j], u \rangle = 0$ and $a_j := 0$ otherwise, where the inner product is taken over $\mathbb{F}_2$. ◀

Note that, by the choice of N and the assumption $2n \leq d$, we have $N < m$ and

$$1 - 2^n(1 - d/m)^N \geq 1 - 2^n e^{-Nd/m} \geq 1 - 2^n 2^{-n/\ln 2} \geq 1/2.$$

Thus, by Lemma 4.6, we have

$$\Pr_{a \sim D_1} [f_M(a) = 1] \geq 1/2 =: \alpha.$$

Moreover, the point a sampled from D_0 is an element of $f_M^{-1}(0)$ as long as $u \neq 0$, so

$$\Pr_{a \sim D_0} [f_M(a) = 0] = 1 - \frac{1}{2^n} \geq \alpha.$$

Spreadness of D_1. To apply Theorem 3.3, we need to show that D_1 is "spread".

Lemma 4.8 (Spreadness of D_1)**.** The distribution D_1 is N-wise (m/N)-spread.

Proof. Let T be any subset of $[m]$ of size $k \leq N$. The number of N-sized sets of columns of M which contain the set of columns indexed by T is $\binom{m-k}{N-k}$. Thus,

$$\Pr_{a \sim D_1} \left[\bigwedge_{i \in T} a_i = 1 \right] \leq \frac{\binom{m-k}{N-k}}{\binom{m}{N}}.$$

Simplifying the above, we get

$$\Pr_{a \sim D_1} \left[\bigwedge_{i \in T} a_i = 1 \right] \leq \left(\frac{\prod_{i=0}^{k-1} (N-i)}{\prod_{i=0}^{k-1} (m-i)} \right) \leq \left(\frac{N}{m} \right)^k,$$

using the fact that $\frac{N-i}{m-i} \leq \frac{N}{m}$ for $i \leq k-1$. ■

Sunflower bound for D_0. The second step is to show a sunflower bound for (D_0, ε)-sunflowers. We first prove that D_0 is t-wise independent.

Lemma 4.9 (Independence of D_0)**.** The distribution D_0 has uniform marginals and is t-wise independent.

Proof. Fix any set $S := \{j_1, j_2, \ldots, j_t\}$ of $[m]$. Also fix a vector

$$b := (b_1, b_2, \ldots, b_t) \in \mathbb{F}_2^t.$$

Then

$$\Pr_{a \sim D_0}[a\restriction_S = b] = \Pr_{u \in \mathbb{F}_2^n}[(\langle u, M[j_1]\rangle = b_1) \wedge \ldots \wedge (\langle u, M[j_t]\rangle = b_t)] = \Pr_{u \in \mathbb{F}_2^n}\left[u^\top M[S] = b\right].$$

Using the rank-nullity theorem, the above probability is the same as

$$\Pr_{u \in \mathbb{F}_2^n}\left[u^\top M[S] = b\right] = \frac{|\ker(M[S])|}{2^n} = \frac{1}{2^t},$$

where the last equality follows from the fact that the rank of $M[S]$ is t (Lemma 4.4). As the columns of M are nonzero (as $t > 0$), we can prove that the marginals are uniform: for every $\ell \in [t]$,

$$\Pr_{u \in \mathbb{F}_2^n}[\langle u, M[j_\ell]\rangle = 0] = \frac{1}{2}.$$

■

Another important ingredient in our proof is Bazzi's theorem [Baz09], which has been further improved by work of Tal [Tal17]. This shows that small DNFs have almost the same behaviour on k-wise independent distributions as they have over truly uniform distributions.

Theorem 4.10 (Fooling DNFs with independence: Theorem 7.1 [Tal17])**.** Let F be a DNF with N terms and $\varepsilon \in (0, 1)$. Then there exists $T \leq O(\log N \cdot \log(N/\varepsilon))$ such that, for any distribution $\mathcal{D}$ that is T-wise independent with uniform marginals, we have

$$\left|\Pr_{a \sim \{0,1\}^n}[F(a) = 0] - \Pr_{a \sim \mathcal{D}}[F(a) = 0]\right| \leq \varepsilon.$$

Finally, to show our sunflower lemma, we combine Bazzi's theorem with the optimal bounds for "robust sunflowers" recently obtained (see, e.g., [Rao25, Lemma 2], [BCW21]). *Robust sunflowers* are $(\mathsf{Uniform}, \varepsilon)$-sunflowers, where $\mathsf{Uniform}$ denotes the uniform distribution. Recall the definition of "sunflower bound" $r(\cdot, \cdot, \cdot)$ from Definition 3.1.

Lemma 4.11 (Robust Sunflower Lemma: Lemma 2 [Rao25])**.** For every $k \in \mathbb{N}$ and $\varepsilon \leq 1/2$, we have $r(\mathsf{Uniform}, k, \varepsilon) = O(\log(k/\varepsilon))$.

We can now show a sunflower lemma for D_0. Recall that m is the number of input bits of the function f_M.

Lemma 4.12 (Sunflower lemma for t-wise independent distributions)**.** There exists a constant $b \geq 1$, such that, for $w := \sqrt{t}/(b \log m)$ and every $k \leq 2w$, we have

$$r(D_0, k, m^{-4w}) = O(w \log m).$$

Proof. Let $w := \sqrt{t}/(b \log m)$ for some constant b to be defined later, and set $\varepsilon := m^{-4w}$. We will show

$$r(D_0, k, \varepsilon) \le r(\mathsf{Uniform}, k, \varepsilon/2).$$

Indeed, suppose that $\mathcal{S}$ is an $\varepsilon/2$-robust sunflower with core K. Let $\mathcal{S}_K = \{ S \setminus K \mid S \in \mathcal{S}\}$. Note that $F_{\mathcal{S}_K}$ has width at most k and at most $\binom{m}{\le k} \le m^k$ terms.

We now wish to apply Theorem 4.10 on $F_{\mathcal{S}_K}$ with the t-wise independent distribution D_0 and approximation parameter $\varepsilon/2$. Theorem 4.10 can only be applied when

$$t \ge D \log(|\mathcal{S}_K|) \log(|\mathcal{S}_K|/(\varepsilon/2)),$$

for some constant $D > 0$ given by Theorem 4.10. Now note that

$$D \log(|\mathcal{S}_K|) \log(|\mathcal{S}_K|/(\varepsilon/2)) \le O(w^2(\log m)^2) = \frac{1}{b^2} O(t) \le t$$

for large enough b. Thus, we can apply Theorem 4.10 and obtain

$$\begin{aligned}
\Pr_{x \sim D_0}[\exists S \in \mathcal{S} : t_{S\setminus K}(x) = 1] &= \Pr_{x \sim D_0}[F_{\mathcal{S}_K}(x) = 1] \\
&\ge \Pr_{x \sim \{0,1\}^n}[F_{\mathcal{S}_K}(x) = 1] - \varepsilon/2 \\
&= \Pr_{x \sim \{0,1\}^n}[\exists S \in \mathcal{S} : t_{S\setminus K}(x) = 1] - \varepsilon/2 \\
&> 1 - \varepsilon.
\end{aligned}$$

Thus, the set $\mathcal{S}$ is a (D_0, ε)-sunflower. The result now follows from Lemma 4.11 by observing that

$$r(\mathsf{Uniform}, k, \varepsilon/2) \le O(\log(k/\varepsilon)) \le O(w \log m). \qquad \blacksquare$$

Wrapping up. We can now apply Theorem 3.3, finishing the proof. Note that the distributions D_1 and D_0 defined in Definition 4.7 satisfy the following properties:

1. We have $\Pr_{x\sim D_i}[f_M(x) = i] \ge \alpha$ for every $i \in \{0, 1\}$;
2. D_1 is N-wise (m/N)-spread (Lemma 4.8);
3. For $w := \sqrt{t}/(b \log m)$ where b is a sufficiently large universal constant (Lemma 4.12), we have $r(D_0, k, \alpha m^{-3w}) \le r(D_0, k, m^{-4w}) \le O(w \log m)$ for every $k \le 2w$ and sufficiently large n.

Therefore, by Theorem 3.3 and the choice of N (Equation 7), we obtain that the monotone complexity of f_M is at least

$$\Omega\left(\frac{m}{N w \log m}\right)^w \ge \Omega\left(\frac{d}{n w \log m}\right)^w = \Omega\left(\frac{d}{n\sqrt{t}}\right)^{\sqrt{t}/(b\log m)}.$$

This finishes the proof of Theorem 4.3.

4.3 Explicit constructions of well-behaved codes

We now construct an explicit well-behaved binary code $\mathcal{C}$ for which we can apply the argument from Section 4.2 to prove exponential lower bounds for monotone circuits computing a boolean function in uniform-$\mathbf{NC}^2$.

Lemma 4.13. Let $n, m, \ell \in \mathbb{N}$, and let $\mathbb{F}_q$ be a finite field with $q := 2^\ell$. Let $\mathcal{L} \subseteq \mathbb{F}_q^m$ be the linear code generated by a matrix $G \in \mathbb{F}_q^{n\times m}$. Then there is a linear binary code $\mathcal{C} \subseteq \mathbb{F}_2^s$ generated by a matrix $M \in \mathbb{F}_2^{r\times s}$ for $r := \ell n$ and $s := \ell m$ such that

$$d(\mathcal{L}) \leq d(\mathcal{C}) \leq \ell d(\mathcal{L}) \text{ and } d^*(\mathcal{L}) \leq d^*(\mathcal{C}) \leq \ell d^*(\mathcal{L}).$$

Furthermore, if each entry of the generator matrix G for $\mathcal{L}$ can be computed in time $\mathrm{poly}(n, m, \ell)$, then we can compute all the entries of the generator matrix M for $\mathcal{C}$ in time $\mathrm{poly}(n, m, \ell)$.

Proof. Let $\mathcal{B} := \{b_1, \dots, b_\ell\}$ be a basis of $\mathbb{F}_q$ as an ℓ-dimensional vector space over $\mathbb{F}_2$. Let $\phi\colon \mathbb{F}_q \to \mathbb{F}_2^\ell$ be the map from elements $v \in \mathbb{F}_q$ to their coordinate vector with respect to the basis $\mathcal{B}$: that is, for every $v \in \mathbb{F}_q$, let $\phi(v) \in \mathbb{F}_2^\ell$ be the unique vector such that

$$v = \sum_{i\in[\ell]} \phi(v)_i b_i.$$

Note that ϕ is a linear isomorphism between $\mathbb{F}_q$ and $\mathbb{F}_2^\ell$. Let $\gamma\colon \mathbb{F}_q^m \to \mathbb{F}_2^{\ell m}$ be the following map: for every $v_1, \dots, v_m \in \mathbb{F}_q$,

$$\gamma(v_1, \dots, v_m) := (\phi(v_1), \dots, \phi(v_m)).$$

Again note that γ is a linear isomorphism between $\mathbb{F}_q^m$ and $\mathbb{F}_2^{\ell m}$; in particular:

$$\gamma(0) = 0 \text{ and } \gamma \text{ is a linear bijection.}$$

Let

$$\mathcal{C} := \{\, \gamma(w) \mid w \in \mathcal{L}\},$$

and let $g_1, \dots, g_n \in \mathbb{F}^m$ be the rows of G. Note that, for every $w \in \mathbb{F}_q^k$,

$$G^\top w = \sum_{i\in[n]} w_i g_i = \sum_{i\in[n]} \sum_{j\in[\ell]} \phi(w_i)_j b_j g_i,$$

hence

$$\gamma(G^\top w) = \sum_{i\in[n]} \sum_{j\in[\ell]} \phi(w_i)_j \gamma(b_j g_i).$$

Let $M \in \mathbb{F}_2^{r\times s}$, for $r := \ell n$ and $s := \ell m$, be the matrix with the vectors $\gamma(b_j g_i)$ as rows, that is,

$$M := \sum_{i\in[n]} \sum_{j\in[\ell]} e_{(i-1)\ell+j} \gamma(b_j g_i)^\top = \begin{bmatrix} \gamma(b_1 g_1)^\top \\ \vdots \\ \gamma(b_n g_\ell)^\top \end{bmatrix}, \tag{8}$$

where $e_1, \dots, e_{n\ell}$ are the standard basis vectors of $\mathbb{F}_2^{n\ell}$. As ϕ is a linear isomorphism, we can prove that

$$\mathcal{C} = \{\, M^\top w \mid w \in \mathbb{F}_2^{n\ell}\}.$$

Now let us prove the claimed distance bounds for $\mathcal{C}$. First note that if $v \in \mathcal{C}$ satisfies $v = \gamma(w) = (\phi(w_1), \ldots, \phi(w_m))$ for some $w \in \mathcal{L} \subseteq \mathbb{F}_q^m$, then, as $\phi(x) = 0$ iff $x = 0$, we have

$$|\operatorname{supp}(w)| \leq |\operatorname{supp}(v)|,$$

which implies that $d(\mathcal{L}) \leq d(\mathcal{C})$, and $|\operatorname{supp}(v)| \leq \ell |\operatorname{supp}(w)|$, which implies that

$$d(\mathcal{C}) \leq \ell d(\mathcal{L}).$$

In order to prove the bound on the dual distance of $\mathcal{C}$, we will use the following characterization of the dual distance of codes:

Proposition 4.14 (Theorem 8 from Chapter 5 of [MS77])**.** We say that a set $S \subseteq \mathbb{F}_q^m$ is t-wise independent if, for a uniformly chosen $(x_1, \ldots, x_m) := x \sim \operatorname{Unif}(S)$ and for all $I \subseteq [m]$ with $|I| \leq t$, the variables $(x_i)_{i \in I}$ are independent and uniformly distributed over $\mathbb{F}_q$. Then a linear code $\mathcal{C} \subseteq \mathbb{F}_q^m$ is t-wise independent if and only if $t \leq d^*(\mathcal{C}) - 1$.

Now let us prove that $d^*(\mathcal{L}) \leq d^*(\mathcal{C})$. For any $t \leq d^*(\mathcal{L}) - 1$, we know that $\mathcal{L}$ is t-wise independent over $\mathbb{F}_q$ by Proposition 4.14. This implies that, for a uniformly chosen $(x_1, \ldots, x_m) := x \sim \operatorname{Unif}(\mathcal{L})$ and for $I \subseteq [m]$ with $|I| \leq t$, the variables $(x_i)_{i \in I}$ are independent and uniformly distributed over $\mathbb{F}_q$. Let

$$(y_1, \ldots, y_{\ell m}) := y \sim \operatorname{Unif}(\mathcal{C}).$$

As x_i for any $i \in I$ is uniformly over $\mathbb{F}_q$ and as ϕ is a bijection from $\mathbb{F}_q$ to $\mathbb{F}_2^\ell$, we get that $(y_{i,1}, \ldots, y_{i,\ell}) := \phi(x_i)$ are independent random variables and uniformly distributed over $\mathbb{F}_2$. Thus, any subset $J \subseteq I \times [\ell]$ of the random variables $(y_{i,p})_{i \in I, p \in [\ell]}$ with $|J| \leq t$ satisfies that $(y_j)_{j \in J}$ are independent random variables and uniformly distributed over $\mathbb{F}_2$. Hence, $\mathcal{C}$ is t-wise independent, and, by Proposition 4.14,

$$d^*(\mathcal{L}) \leq d^*(\mathcal{C}).$$

Now suppose that

$$d^*(\mathcal{C}) \geq \ell d^*(\mathcal{L}) + 1.$$

By Proposition 4.14, we get that $\mathcal{C}$ is $(\ell d^*(\mathcal{L}))$-wise independent. Again using the fact ϕ is a bijection from $\mathbb{F}_q$ to $\mathbb{F}_2^\ell$, we can prove that, for a uniformly chosen $(x_1, \ldots, x_m) := x \sim \operatorname{Unif}(\mathcal{L})$ and for $I \subseteq [m]$ with $|I| \leq d^*(\mathcal{L})$, the variables $(x_i)_{i \in I}$ are independent and uniformly distributed over $\mathbb{F}_q$. By Proposition 4.14, we obtain that $d^*(\mathcal{L}) + 1 \leq d^*(\mathcal{L})$, which is a contradiction. Therefore,

$$d^*(\mathcal{C}) \leq \ell d^*(\mathcal{L}).$$

Now let us argue about the explicitness of our construction. Note that:

- Element representations (via a basis for $\mathbb{F}_q$) and operations over the field $\mathbb{F}_q$ can be performed in time $\operatorname{poly}(\ell)$ [Sho88].
- For any given $(v_1, \ldots, v_m) := v \in \mathbb{F}_q^m$, the element $\gamma(v) \in \mathbb{F}_2^{\ell m}$ can be uniformly computed in time $\operatorname{poly}(m, \ell)$ as it is just the concatenation of $\phi(v_1), \ldots, \phi(v_m) \in \mathbb{F}_2^\ell$ and $\phi(v_i)$ is the representation of v_i as an element of $\mathbb{F}_q$.
- The generator matrix $M \in \mathbb{F}_2^{\ell n \times \ell m}$ (Equation 8) for $\mathcal{C} \subseteq \mathbb{F}_2^{\ell m}$ can be uniformly computed in time $\operatorname{poly}(n, m, \ell)$. ■

Corollary 4.15. For every $n, m \in \mathbb{N}$ with $m > n$ and for $\ell := \lceil \log_2 m \rceil$, there is a binary code $\mathcal{C} \subseteq \mathbb{F}_2^{\ell m}$ such that we can explicitly construct a generator matrix $M \in \mathbb{F}_2^{\ell n \times \ell m}$ for $\mathcal{C}$ and

$$m - n \leq d(\mathcal{C}) \leq 2\ell(m-n) \text{ and } n \leq d^*(\mathcal{C}) \leq 2\ell n.$$

Proof. For $q := 2^\ell$, let $\mathcal{L} \subseteq \mathbb{F}_q^m$ be a Reed-Solomon code over $\mathbb{F}_q$ with generator matrix $G \in \mathbb{F}_q^{n \times m}$. The following facts are standard results in coding theory (e.g., see Chapter 10 of [MS77]):

- Each entry of the matrix G can be constructed in time $\text{poly}(n, m, \ell)$.
- The distance of $\mathcal{L}$ is
$$d(\mathcal{L}) = m - n + 1.$$
- The dual code of $\mathcal{L}$ has distance
$$d^*(\mathcal{L}) = m - (m-n) + 1 = n + 1.$$

By Lemma 4.13, there is a binary code $\mathcal{C} \subseteq \mathbb{F}_2^{\ell m}$ such that we can explicitly construct a generator matrix $M \in \mathbb{F}_2^{\ell n \times \ell m}$ for $\mathcal{C}$, and

$$m - n \leq d(\mathcal{L}) \leq d(\mathcal{C}) \leq \ell d(\mathcal{L}) \leq 2\ell(m-n), \text{ and } n \leq d^*(\mathcal{L}) \leq d^*(\mathcal{C}) \leq \ell d^*(\mathcal{L}) \leq 2\ell n. \qquad \blacksquare$$

Theorem 4.16 (Theorem 1.5)**.** For every sufficiently large $n \in \mathbb{N}$, and for

$$m := \lceil n^{3/2} (\log n)^2 \rceil \text{ and } \ell := \lceil \log_2 m \rceil,$$

there is an ℓm-variate monotone Boolean function $f \colon \{0,1\}^{\ell m} \to \{0,1\}$ such that the following hold:

1. f can be computed by a uniform Boolean circuit of size polynomial in n and depth $O(\log \ell m)^2$.
2. Any monotone Boolean circuit computing f has size at least $2^{(\ell m)^{1/3 - o(1)}}$.

Proof. By Corollary 4.15, there is a binary code $\mathcal{C} \subseteq \mathbb{F}_2^{\ell m}$ such that we can explicitly construct a generator matrix $M \in \mathbb{F}_2^{\ell n \times \ell m}$ for $\mathcal{C}$ and

$$m - n \leq d(\mathcal{C}) \leq 2\ell(m-n) \text{ and } n \leq d^*(\mathcal{C}) \leq 2\ell n. \tag{9}$$

Let $f := f_M$ be the ℓm-variate Boolean function corresponding to M. By Lemma 4.1, the function f can be computed by uniform Boolean circuit of size $\text{poly}(\ell m)$ and depth $O(\log \ell m)^2$. Let

$$d := d(\mathcal{C}) - 1 \text{ and } t := d^*(\mathcal{C}) - 1.$$

Note that, for sufficiently large n,

$$d \geq 2n\ell.$$

By Theorem 4.3, we obtain that the monotone complexity $S^+(f)$ of f is at least

$$\Omega\left(\frac{d}{\ell n w \log(\ell m)}\right)^w,$$

for $w := \sqrt{t}/(b \log(\ell m))$. By Equation 9, we get that

$$\Omega(m^{1/3}/(\log m)^2) \leq w \leq O(n^{1/2}/(\log m)^{1/2}).$$

Therefore,

$$S^+(f) \geq \Omega\left(\frac{d}{\ell n w \log(\ell m)}\right)^w \geq \Omega\left(\frac{m}{n^{3/2}(\log m)^{3/2}}\right)^w \geq \Omega(\log m)^{w/2} \geq 2^{\Omega(m^{1/3} \log\log m/(\log m)^2)}. \qquad \blacksquare$$

5 Mixed separation

This section uses some ideas from Section 4 to prove a lower bound for f_M when M is a real matrix. We get weaker (but still superpolynomial) lower bounds for the size of monotone circuits.

5.1 Nonmonotone low-depth circuit upper bounds

Let us first prove the upper bound part of Theorem 1.4.

Lemma 5.1 (Upper bound)**.** For any matrix $A \in \mathbb{R}^{n\times m}$, the polynomial $P_A \in \mathbb{R}[x_1, \ldots, x_m]$, as defined in Equation 2, is computed by an arithmetic circuit of size poly(m) and depth $O(\log m)^2$.

Proof. Let

$$I_X := \sum_{i=1}^{m} x_i e_i e_i^\top \in \mathbb{R}[X]^{m\times m}$$

be the m-dimensional identity matrix with diagonal elements replaced by the variables $x_1, \ldots, x_m$. By the Cauchy-Binet formula, we have

$$\det((AI_X)A^T) = \sum_{\substack{S\subseteq[m]:\\|S|=n}} \det((AI_X)[[n], S]) \det(A^T[S, [n]]) = \sum_{\substack{S\subseteq[m]:\\|S|=n}} \det((AI_X)[[n], S]) \det(A[S]),$$

and, by the block structure of I_X and the multiplicative property of the determinant,

$$\det((AI_X)[[n], S]) = \det(A[[n], S]I_X[S, S]) = \det(A[[n], S]) \det(I_X[S, S]) = \det(A[S]) \prod_{i\in S} x_i.$$

Hence,

$$\det((AI_X)A^T) = \sum_{\substack{S\subseteq[m]:\\|S|=n}} \left(\det(A[S]) \prod_{i\in S} x_i\right) \det(A[S]) = P_A(x_1, \ldots, x_m).$$

By efficient computation of the determinant of a symbolic matrix [Ber84, MV97], we obtain that the polynomial $P_A = \det(AI_X A^T)$ can be computed by an arithmetic circuit of size poly(m) and depth $O((\log m)^2)$. ■

5.2 Choice of a well-behaved $\mathbb{R}$-matrix

Let $k := k(n) \in \mathbb{N}$ be a growing function, which will be specified later, such that $k \leq n^{0.1}$. Let

$$m := n^2 \qquad \text{and} \qquad s := 200k^2. \tag{10}$$

Let M_n be an $n \times m$ random matrix obtained by sampling each column $M_n[i]$ of M_n independently and uniformly at random from the set of vectors in $\{0, 1\}^n$ of Hamming weight at most s.

For technical reasons, it is nicer to sample the matrix using points of Hamming weight *at most* s, though the following lemma shows that this is not very different from sampling points of weight close to s.

Lemma 5.2. With probability at least $1 - 1/n$, every column of M_n has Hamming weight at least $s/2$.

Proof. Note that $s = 200k^2 = O(n^{0.2})$. We note that $\binom{n}{i} \leq n^i$ for $i \in [n]$, and use the bound on s to conclude

$$\binom{n}{\leq s} \geq \binom{n}{s} \geq \left(\frac{n}{s}\right)^s \geq n^{0.7s}.$$

Putting things together, the probability that any fixed column $M_n[j]$ has Hamming weight less than $s/2$ is at most

$$\sum_{i=0}^{s/2} \frac{\binom{n}{i}}{\binom{n}{\leq s}} \leq \frac{(s/2+1) \cdot n^{s/2}}{n^{0.7s}} \leq \frac{n}{n^{0.2s}}.$$

Union bounding over $m = n^2$ columns and using the fact that $s \geq 200$, the probability that there is a column of Hamming weight at most $s/2$ is at most $1/n$. ■

To construct our hard distributions, the following lemma will be important.

Lemma 5.3 (Full rank lemma)**.** If S is distributed uniformly in $\binom{[m]}{10n \log n}$ and independently of M_n, then

$$\Pr_{M_n}\left[\Pr_{S}[M_n[S] \text{ is full rank}] \geq 1/10\right] \geq 1/10.$$

Proof. The idea is to carry out an argument analogous to the solution to the coupon-collector's problem to show that for each $S \in \binom{[m]}{10n \log n}$, we have

$$\Pr_{M_n}[M_n[S] \text{ is full rank}] \geq \frac{4}{5}. \tag{11}$$

The claim of the lemma then follows from Equation 11 by the following averaging argument. Define the random variables

$$X := \Pr_{S \in \binom{[m]}{n}}[M_n[S] \text{ is full rank}], \text{ and } Y := 1 - X.$$

Thus

$$\begin{aligned}
\mathbb{E}_{M_n}[X] &= \mathbb{E}_{M_n}\left[\frac{1}{\binom{m}{n}} \sum_{S \subseteq [m]:|S|=n} \mathbb{1}_{\text{rank}(M_n[S])=n}\right] \\
&= \frac{1}{\binom{m}{n}} \sum_{S \subseteq [m]:|S|=n} \Pr_{M_n}[\text{rank}(M_n[S]) = n] \geq \frac{4}{5},
\end{aligned}$$

and $\mathbb{E}_{M_n}[Y] \leq \frac{1}{5}$. By Markov's inequality, we get

$$1 - \Pr_{M_n}\left[X \geq \frac{1}{10}\right] = \Pr_{M_n}\left[Y \geq \frac{9}{10}\right] \leq \frac{10\mathbb{E}_{M_n}[Y]}{9} \leq \frac{2}{9},$$

which implies that

$$\Pr_{M_n}\left[X \geq \frac{1}{10}\right] \geq \frac{7}{9} > \frac{1}{10}.$$

Now let us prove the probability bound in Equation 11. Assume, without loss of generality, that $S = \{1, \ldots, 10n \log n\}$. Assume that we choose an infinite sequence of vectors $\{v_i\}_{i \in \mathbb{N}}$ of vectors independently and uniformly at random $\{0,1\}^n_{\le s}$. For each $d \le n$, let X_d denote the smallest r such that $v_1, \dots, v_r$ span a subspace of dimension d. It suffices to show that $\mathbb{E}X_n \le 2n \log n$, because Equation 11 then follows by Markov's inequality, since we can interpret the definition of M_n as choosing the vectors $v_1, \cdots, v_{10n \log n}$ as M_n's columns. For $X_0 := 0$, we have that $X_n := \sum_{d=0}^{n-1} X_{d+1} - X_d$, which implies that

$$\mathbb{E}X_n = \sum_{d=0}^{n-1} \mathbb{E}[X_{d+1} - X_d].$$

To compute $\mathbb{E}[X_{d+1} - X_d]$, we condition on the value $X_d = r$ and the vectors $v_1, \dots, v_r$ which span a vector space V of dimension d. Let p_d denote the probability that a uniformly random $v \in \{0,1\}^n_{\le s}$ lies outside V. By the subspace lemma (Lemma 5.4) proved below, we have

$$1 - p_d \le \frac{\binom{d}{\le s}}{\binom{n}{\le s}} \le \frac{d+1}{n+1}$$

where the latter inequality follows from the following simple binomial estimate:

$$\binom{d}{\le s} = 1 + d + \sum_{i=2}^{s} \binom{d}{i} \le 1 + d + \sum_{i=2}^{s} \frac{d+1}{n+1} \cdot \binom{n}{i} = \frac{d+1}{n+1} \cdot \binom{n}{\le s}.$$

It follows that $X_{d+1} - X_d$ (conditioned on $v_1, \dots, v_r$) has a geometric distribution with success probability $p_d \ge (n-d)/(n+1)$ and thus has expectation at most $(n+1)/(n-d)$. Hence

$$\mathbb{E}X_n \le \sum_{d=0}^{n-1} \frac{n+1}{n-d} \le 2n \log n.$$

We have thus shown the desired upper bound on $\mathbb{E}X_n$, which completes the proof of the lemma. ■

The above proof used the following lemma, which show that a uniformly randomly chosen $v \in \{0,1\}^n_s$ cannot lie in any proper subspace of $\mathbb{R}^n$ with high probability.

Lemma 5.4 (Subspace lemma)**.** Let $\mathbb{F}$ be any field and let V be a subspace of $\mathbb{F}^n$ of dimension d, and $\{0,1\}^n_{\le s}$ be the set of all binary strings of Hamming weight at most s. Then,

$$|V \cap \{0,1\}^n_{\le s}| \le \binom{d}{\le s}.$$

We note that the above lemma is tight, as witnessed by a subspace V generated by any d standard basis vectors.

Proof. As V has dimension d, there is a basis B of V such that there is a set $R \in \binom{[n]}{d}$ indexing the elements of B as

$$B = \{\, b^{(r)} \mid r \in R\}$$

such that, for every $r \in R$ and $v \in B$,

$$\begin{cases} v_r = 1 & \text{if } v = b^{(r)}, \text{ and} \\ v_r = 0 & \text{otherwise.} \end{cases}$$

Note that we can find such a basis via Gaussian elimination over any arbitrary basis of V, as the set R above corresponds to the rows of the pivot element of the matrix in column echelon form. For every $v \in X := V \cap \{0,1\}^n_{\leq s}$, let $\alpha_v \in \mathbb{R}^B$ be the unique vector such that $v = \sum_{b \in B} \alpha_{v,b} b$. So, for every $r \in R$,

$$v_r = \sum_{b \in B} \alpha_{v,b} b_r = \alpha_{v,b^{(r)}},$$

which implies that $\operatorname{supp}(v) \cap R$ uniquely determines α_v (as $v_r \in \{0,1\}$), and, consequently, uniquely determines v. As $|\operatorname{supp}(v) \cap R| \leq |\operatorname{supp}(v)| \leq s$ for every $v \in X$, we get that

$$|X| \leq \left| \binom{|R|}{\leq s} \right| = \binom{d}{\leq s}. \qquad \blacksquare$$

We also need the following lemma to show that, with high probability, most columns in an arbitrary small tuple of columns of M_n have a small fraction of their support intersecting the union of the support of their preceding columns.

Definition 5.5. For $\tau := (i_1, \ldots, i_t)$ a tuple of distinct elements of $[m]$, we say that i_j is *c-contained w.r.t. τ* if the set $\operatorname{supp}(M_n[i_j])$ has at least c elements in common with $\bigcup_{p<j} \operatorname{supp}(M_n[i_p])$. ◀

Lemma 5.6. Let S_τ be the following (random) set

$$S_\tau := \{\, j \in [t] \mid i_j \text{ is } 10k\text{-contained w.r.t. } \tau\}.$$

Then, for any positive integer $t \leq n^{0.1}$, we have

$$\Pr_{M_n}[\exists\, \tau \text{ such that } |S_\tau| \geq t/2k] \leq \frac{1}{n}.$$

Proof. Let $r := \lceil t/2k \rceil$ and $\kappa := 10k$. By the union bound over the choices of $\tau \in [m]^t$ and $S \in \binom{[t]}{r}$, we get that

$$\Pr_{M_n}[\exists\, \tau \text{ such that } |S_\tau| \geq t/2k] \leq m^t \cdot \binom{t}{r} \max_{\tau, S} \Pr_{M_n}[S \subseteq S_\tau].$$

Now we are going to find an upper bound for $\Pr_{M_n}[S \subseteq S_\tau]$ for every $\tau =: (i_1, \ldots, i_t)$ and $S \in \binom{[t]}{r}$. Let $\{s_1, \ldots, s_r\} := S$ such that $s_1 < \ldots < s_r$, and, for every $j \in [r]$, let

$$T_j := \cup_{p<s_j} \operatorname{supp}(M_n[i_p]).$$

First note that $|T_j| \leq ts$, and

$$\begin{aligned}
\Pr_{M_n}[S \subseteq S_\tau] &\leq \Pr_{M_n}\left[\cap_{j=1}^r \{|\operatorname{supp}(M_n[i_{s_j}]) \cap T_j| \geq \kappa\}\right] \\
&= \prod_{j=1}^r \Pr_{M_n}\left[|\operatorname{supp}(M_n[i_{s_j}]) \cap T_j| \geq \kappa \,\middle|\, \cap_{i=1}^{j-1} \{|\operatorname{supp}(M_n[i_{s_i}]) \cap T_i| \geq \kappa\}\right] \\
&= \prod_{j=1}^r \Pr_{M_n}\left[|\operatorname{supp}(M_n[i_{s_j}]) \cap T_j| \geq \kappa\right],
\end{aligned}$$

where the last equality follows from the independence between choices for the columns of M_n. For every $j \in [r]$, we have, again by independence of columns,

$$\begin{aligned}\Pr_{M_n}\left[|\operatorname{supp}(M_n[i_{s_j}]) \cap T_j| \geq \kappa\right] &= \sum_{T\subseteq[n],\kappa\leq|T|\leq ts} \Pr_{M_n}\left[|\operatorname{supp}(M_n[i_{s_j}]) \cap T| \geq \kappa\right] \Pr_{M_n}\left[T_j = T\right] \\ &\leq \max_{T\subseteq[n],\kappa\leq|T|\leq ts} \Pr_{M_n}\left[|\operatorname{supp}(M_n[i_{s_j}]) \cap T| \geq \kappa\right].\end{aligned}$$

For every $T \subseteq [n]$ such that $\kappa \leq |T| \leq ts$, we get

$$\begin{aligned}\Pr_{M_n}\left[|\operatorname{supp}(M_n[i_{s_j}]) \cap T| \geq \kappa\right] &= \Pr_{M_n}\left[\exists R \in \binom{T}{\kappa} \text{ s.t. } R \subseteq \operatorname{supp}(M_n[i_{s_j}])\right] \\ &\leq \frac{\binom{|T|}{\kappa}\binom{n-\kappa}{\leq(s-\kappa)}}{\binom{n}{\leq s}} \leq \frac{\binom{|T|}{\kappa}\cdot(s+1)\cdot\binom{n-\kappa}{s-\kappa}}{\binom{n}{s}} \\ &\leq (s+1)(tse/\kappa)^{\kappa}(2s/n)^{\kappa} \\ &\leq n^{-0.5\kappa} = n^{-5k}.\end{aligned}$$

Therefore, we obtain

$$\Pr_{M_n}\left[S \subseteq S_\tau\right] \leq n^{-5kr} \leq n^{-2.5t}$$

and, as a consequence,

$$\Pr_{M_n}\left[\exists\ \tau \text{ such that } |S_\tau| \geq t/2k\right] \leq \binom{t}{r} \cdot m^t n^{-2.5t} = \binom{t}{r} \cdot n^{-0.5t} \leq 1/n.$$ ■

We now collect in the following definition the properties of a matrix that we need for the proof of our lower bound.

Definition 5.7. We say that a matrix $M \in \mathbb{R}^{n\times m}$ is *well-behaved* if the following properties hold:

1. Every column $M[i]$ has support size at least $s/2$,
2. $\Pr_{S\in\binom{[m]}{10n\log n}}\left[M[S] \text{ is full rank}\right] \geq 1/10$, and
3. for any $t \leq n^{0.1}$ and any tuple $\tau = (i_1, \ldots, i_t)$ of distinct elements from $[m]$, the number of $j \in [t]$ such that i_j is $10k$-contained w.r.t. τ is smaller than $t/2k$. In particular, if $t \leq k$, there are no $j \in [t]$ such that i_j is $10k$-contained w.r.t. τ.

◀

By Lemma 5.2, Lemma 5.3 and Lemma 5.6 (along with a union bound over all $t \leq n^{0.1}$), we know that, with positive probability, M_n is a well-behaved matrix for every sufficiently large n. For the rest of this section, we will denote by M a fixed well-behaved matrix.

5.3 Monotone Boolean circuit lower bounds via well-behaved matrices

In the remaining of this section, we show that the Boolean function f_M cannot be computed by a monotone Boolean circuit of small size. This is the lower bound part of Theorem 1.4. As in Section 4.2, we start by defining two probability distributions over the inputs of f_M.

Definition 5.8. We define the distribution $a \sim D_1$ by sampling a uniformly random $a \in \{0,1\}^m$ of Hamming weight $10n \log n$. We define the distribution $a \sim D_0$ by sampling a uniformly random $u \in \{-1,0,1\}^n$ and, for every $j \in [m]$, we set

$$a_j := 1 \text{ if } \langle M[j], u\rangle = 0, \text{ and}$$
$$a_j := 0 \text{ otherwise,}$$

where the inner product is taken over $\mathbb{R}$. ◀

Observation 5.9. Since the matrix M is well-behaved (see Definition 5.7), the distribution D_1 satisfies $\Pr_{a\sim D_1}[f(a) = 1] \geq 1/10$.

Further, as long as $u \neq 0$, the point a sampled from D_0 is an element of $f_M^{-1}(0)$. Thus we have

$$\Pr_{a\sim D_0}[f_M(a) = 0] = 1 - \frac{1}{3^n}.$$

Thus, to show a lower bound via the monotone circuit lower bound criterion (Theorem 3.3), as before it suffices to show that D_1 is spread and to show a bound for (D_0, ε)-sunflowers.

Lemma 5.10 (Spreadness of D_1)**.** The distribution D_1 is $n^{0.1}$-wise $(n/(10 \log n))$-spread.

Proof. Let T be a subset of $[m]$ of size $k \leq n^{0.1}$. The proof of this lemma is similar to the proof of Lemma 4.8. Note that, for $\ell := 10n \log n$ and $S \sim \mathrm{Unif}(\binom{m}{\ell})$,

$$\Pr_{a\sim D_1}\left[\bigwedge_{i\in T} a_i = 1\right] = \Pr_S[T \subseteq S] = \frac{\binom{m-k}{\ell-k}}{\binom{m}{\ell}} = \prod_{i=0}^{k-1}\frac{\ell-i}{m-i} \leq \left(\frac{\ell}{m}\right)^k,$$

where the last inequality follows from $\frac{\ell-i}{m-i} \leq \frac{\ell}{m}$ for $i \leq k-1$. As $m = n^2$, we have

$$\Pr_S[T \subseteq S] \leq \left(\frac{10n \log n}{n^2}\right)^k = (n/(10 \log n))^{-k}.$$

■

Sunflower bound. To show the sunflower bound, we first show a weak form of bounded independence for D_0.

Lemma 5.11 (Weak form of independence for D_0)**.** Assume that $t \leq n^{0.1}$. Fix any tuple $\tau := (i_1, \dots, i_t)$ of distinct elements in $[m]$ and $j \in [t]$ such that i_j is not $10k$-contained w.r.t. τ. Then,

$$\Pr_{a\sim D_0}\left[a_{i_j} = 1 \mid a_{i_1}, \dots, a_{i_{j-1}}\right] \geq \Omega(1/k).$$

Proof. Let $S_\ell := \operatorname{supp} M[i_\ell]$ for $\ell \in [j]$, and $R := S_j \setminus \bigcup_{\ell=1}^{j-1} S_\ell$. For every $T \subseteq [n]$, let

$$X_T := \sum_{\ell\in T} u_\ell,$$

where u_ℓ's are the random variables used to defined D_0. If i_j is not $10k$-contained w.r.t. τ, we have

$$|R| \geq |S_j| - 10k \geq (s/2) - 10k.$$

As $X_{[n]} = 0$, we have that

$$\begin{aligned}
&\Pr_{a\sim D_0}[a_{i_j} = 1 \mid a_{i_1}, \ldots, a_{i_{j-1}}] \\
&\quad = \sum_{p=-10k}^{10k} \Pr_{a\sim D_0}\left[X_{S_j\setminus R} = -p, X_R = p \mid a_{i_1}, \ldots, a_{i_{j-1}}\right] \\
&\quad = \sum_{p=-10k}^{10k} \Pr_{a\sim D_0}\left[X_R = p \mid X_{S_j\setminus R} = -p, a_{i_1}, \ldots, a_{i_{j-1}}\right] \Pr_{a\sim D_0}\left[X_{S_j\setminus R} = -p \mid a_{i_1}, \ldots, a_{i_{j-1}}\right] \\
&\quad = \sum_{p=-10k}^{10k} \Pr_{a\sim D_0}[X_R = p] \Pr_{a\sim D_0}\left[X_{S_j\setminus R} = -p \mid a_{i_1}, \ldots, a_{i_{j-1}}\right],
\end{aligned}$$

where the last equality follows from the independence between coordinates of u. Note that, for every $p \in [-10k, 10k]$ and $r := |R|$ and $Y_\ell \sim \text{Rademacher}(\ell)$ being a Rademacher random variable with ℓ independent samples for every $\ell \in \{0, \ldots, r\}$, we have

$$\begin{aligned}
\Pr_{a\sim D_0}[X_R = p] &= \sum_{U\subseteq R} \Pr_{a\sim D_0}\left[u^{-1}(0) = U, X_R = p\right] \\
&= \sum_{U\subseteq R} \Pr_{a\sim D_0}\left[X_{R\setminus U} = p \mid u^{-1}(0) = U\right] \Pr_{a\sim D_0}\left[u^{-1}(0) = U\right] \\
&\geq \sum_{U\subseteq R, ||U|-r/3|\leq \delta r/3} \Pr_{a\sim D_0}\left[Y_{r-|U|} = p\right] \Pr_{a\sim D_0}\left[u^{-1}(0) = U\right] \\
&\geq \left(\min_{w\in[r] \text{ s.t. } |w-r/3|\leq \delta r/3} \Pr_{a\sim D_0}[Y_{r-w} = p]\right) \left(\sum_{U\subseteq R, ||U|-r/3|\leq \delta r/3} \Pr_{a\sim D_0}\left[u^{-1}(0) = U\right]\right),
\end{aligned}$$

for any $\delta \in [0, 1]$. Note that $u^{-1}(0) \sim \text{Bin}(r, 1/3) =: B$; thus, by Chernoff's inequality [MU17],

$$\Pr_B[|B - \mathbb{E}B| > \delta\mathbb{E}B] \leq 2e^{-\delta^2\mathbb{E}B/3},$$

and, for $\delta := \sqrt{18/r}$,

$$\Pr_B[|B - r/3| > \delta r/3] \leq 2e^{-2} \leq 1/2.$$

Hence,

$$\sum_{U\subseteq R, ||U|-r/3|\leq \delta r/3} \Pr_{a\sim D_0}\left[u^{-1}(0) = U\right] \geq 1/2.$$

Now we just need to find good estimates for $\Pr_{a\sim D_0}[Y_v = p]$ for $v := r - w$ with $w \in [r]$ satisfying

$$|w - r/3| \leq \sqrt{2r}.$$

In order to obtain these estimates, we will use standard inequalities to deal with binomial coefficients. First note that

$$\Pr_{a\sim D_0}[Y_v = p] = \binom{v}{(v+p)/2} 2^{-v}.$$

By using that $1 - x \geq e^{-x/(1-x)}$ for every $x \leq 1$, we get that

$$\begin{aligned} \frac{\binom{v}{(v+p)/2}}{\binom{v}{v/2}} &= \prod_{i=0}^{p/2-1} \frac{v/2 - i}{v/2 + p/2 - i} \\ &\geq \left(\frac{v/2 - p/2}{v/2}\right)^{p/2} \\ &\geq \left(e^{-\frac{p}{v}\frac{1}{1-p/v}}\right)^{p/2} \\ &\geq e^{-\frac{p^2}{2(v-p)}}, \end{aligned}$$

and, using that $\binom{v}{v/2} = \Theta(2^v/\sqrt{v})$, we obtain

$$\begin{aligned} \Pr_{a\sim D_0}[Y_v = p] &\geq e^{-\frac{p^2}{2(v-p)}} \binom{v}{v/2} 2^{-v} \\ &\geq \Omega\left(\frac{e^{-\frac{p^2}{2(v-p)}}}{\sqrt{v}}\right). \end{aligned}$$

As $p \in [-10k, 10k]$ and

$$200k^2 = s \geq v \geq 2r/3 - \sqrt{18r} \geq r/3 \geq \frac{1}{3}((s/2) - 10k) = \frac{1}{3}(200k^2 - 10k) \geq 60k^2$$

for sufficiently large k, we get

$$\begin{aligned} \Pr_{a\sim D_0}[Y_v = p] &\geq \Omega\left(\frac{e^{-\frac{200k^2}{2\cdot(60k^2-10k)}}}{\sqrt{k^2}}\right) \\ &\geq \Omega\left(1/k\right). \end{aligned}$$

Therefore, we obtain

$$\Pr_{a\sim D_0}\left[a_{i_j} = 1 \mid a_{i_1}, \ldots, a_{i_{j-1}}\right] \geq \Omega\left(1/k\right) \cdot \frac{1}{2} = \Omega(1/k). \quad \blacksquare$$

Our main combinatorial tool is the classical sunflower lemma [ER60] (see also [Juk11, Section 6.1]). Recall that a sunflower is a collection of sets $S_1, \ldots, S_r$ such that the pairwise intersections $S_i \cap S_j$ are all the same. The improved bounds of [ALWZ21] and subsequent works will not make any substantial difference in our bounds.

Lemma 5.12 (Sunflower lemma)**.** If $\mathcal{S}$ is a family of sets of size at most $\ell \in [k]$ such that $|\mathcal{S}| \geq \ell!(r-1)^\ell$, then $\mathcal{S}$ contains a sunflower with r sets.

We now prove our D_0-sunflower bound. To make the bound cleaner and simpler to prove, we now set our choice of k. We set

$$k := \lceil (\log m)^{1/2} \rceil. \tag{12}$$

This is the only place where we need to set $k = n^{o(1)}$, owing to our use of the *classical* sunflower lemma.

Lemma 5.13 (D_0-sunflower lemma)**.** For every $\ell \in [k]$, we have $r(D_0, \ell, m^{-4k}) \leq 2k^{2\ell+1}\ell \log m$.

Proof. Let $\varepsilon := m^{-4k}$. Let $\mathcal{S}$ be a ℓ-uniform family of sets larger than $(2k^{2\ell}\ell \log(1/\varepsilon))^{\ell}$. By Lemma 5.12, there exists a sunflower $\mathcal{S}' \subseteq \mathcal{S}$ with

$$r := 2k^{2\ell} \log(1/\varepsilon)$$

sets with some core $K := \bigcap_{S \in \mathcal{S}'} S$. Note that

$$r = 8k^{2\ell+1} \log m \ll n^{0.1},$$

by our choice of $k = \sqrt{\log m}$ and $m = n^2$ (Equation (10)).

We now show that $\mathcal{S}'$ is a $(\mathcal{D}_0, \ell, \varepsilon)$-sunflower. Let $\mathcal{F} := \{S \setminus K : S \in \mathcal{S}'\}$. It suffices to show that, for $F := F_{\mathcal{F}}$, we have

$$\Pr_{a \sim \mathcal{D}_0}[F(a) = 0] \leq \varepsilon.$$

Let $F_1, F_2, \ldots, F_r$ be the sets of $\mathcal{F}$. Let

$$\tau := (i_1, i_2, \ldots, i_t)$$

be the sequence of all the indices corresponding to the variables appearing in the terms $F_1, F_2, \ldots, F_r$ in that order (inside each term we order the variables arbitrarily). Since $t \leq rk \leq n^{0.1}$ and by the choice of M (see Definition 5.7, Item (3)), there are at most $t/2k \leq r/2$ indices in τ which are $10k$-contained with respect to τ. In particular, these indices appear across at most $r/2$ 'corrupted' terms among $F_1, \ldots, F_r$. In the remaining terms, all of their indices are not $10k$-contained w.r.t. τ. Removing the corrupted terms yields a a subfamily $\mathcal{F}' \subseteq \mathcal{F}$ with $r' \geq r/2$ sets such that all their indices are not $10k$-contained w.r.t. τ. Let $F'_1, \ldots, F'_{r'}$ be the sets of $\mathcal{F}'$.

For any $j \in [r']$, we can apply Lemma 5.11 for each of F'_j's elements (recall that F'_j has at most ℓ literals), and then obtain

$$\Pr_{a \sim D_0}\left[F'_j(a) = 1 \mid F'_1(a) = 0, \ldots, F'_{j-1}(a) = 0\right] \geq \Omega(1/k)^{\ell}.$$

Therefore, we obtain

$$\Pr_{a \sim D_0}\left[F'_j(a) = 0 \mid F'_1(a) = 0, \ldots, F'_{j-1}(a) = 0\right] \leq 1 - \Omega(1/k)^{\ell},$$

and, as a consequence,

$$\Pr_{a \sim D_0}[F(a) = 0] \leq \Pr_{a \sim D_0}\left[F'(a) = 0\right] \leq \left(1 - \Omega(1/k)^{\ell}\right)^{r'} \leq \exp(-rk^{-2\ell}/2) = \varepsilon,$$

for sufficiently large k. ■

Wrapping up. We can now apply Theorem 3.3, finishing the proof.

Proof of Theorem 1.4. We have shown that there is a sequence of matrices $(M_n)_{n\in\mathbb{N}}$ with entries from $\mathbb{R}$ such that M_n is an $n \times m$ matrix where $m := n^2$ and M_n is *well-behaved* (Definition 5.7). We have also exhibited two distributions D_1, D_0 (Definition 5.8) supported over strings $\{0,1\}^m$ such that

1. $\Pr_{x\sim D_i}[f_M(x) = i] \geq \alpha$ for every $i \in \{0,1\}$, where $\alpha > 0$ is some constant (Definition 5.7 and Observation 5.9);
2. D_1 is $n^{0.1}$-wise $(n/(10\log n))$-spread (Lemma 5.10);
3. $r(D_0, \ell, m^{-4k}) \leq 8k^{2\ell+1}\log m$ for every $\ell \in [k]$ (Lemma 5.13).

Taking $w := k$ in Theorem 3.3 and noting that $\alpha m^{-3k} \leq m^{-4k}$, we obtain that there exists a constant $\beta > 0$ such that the monotone complexity of f_M is

$$\left(\frac{\beta n}{k^{2k+1}\log^2 n}\right)^k = m^{\Omega(\sqrt{\log m})}.$$

The nonmonotone circuit upper bound for f_M was proved in Lemma 5.1. ∎

References

[AB87] Noga Alon and Ravi B. Boppana. The monotone circuit complexity of Boolean functions. *Comb.*, 7(1):1–22, 1987. doi:10.1007/BF02579196.

[ABO99] Eric Allender, Robert Beals, and Mitsunori Ogihara. The complexity of matrix rank and feasible systems of linear equations. *Comput. Complex.*, 8(2):99–126, 1999. URL: https://doi.org/10.1007/s000370050023, doi:10.1007/S000370050023.

[AG87] Miklós Ajtai and Yuri Gurevich. Monotone versus positive. *Journal of the ACM*, 34(4):1004–1015, 1987. doi:10.1145/31846.31852.

[AGG+25] Yaroslav Alekseev, Mika Göös, Ziyi Guan, Gilbert Maystre, Artur Riazanov, Dmitry Sokolov, and Weiqiang Yuan. Generalised linial-nisan conjecture is false for dnfs. In Srikanth Srinivasan, editor, *40th Computational Complexity Conference, CCC 2025, August 5-8, 2025, Toronto, Canada*, volume 339 of *LIPIcs*, pages 29:1–29:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025. URL: https://doi.org/10.4230/LIPIcs.CCC.2025.29, doi:10.4230/LIPICS.CCC.2025.29.

[ALWZ21] Ryan Alweiss, Shachar Lovett, Kewen Wu, and Jiapeng Zhang. Improved bounds for the sunflower lemma. *Ann. of Math.*, 2(194(3)):795–815, 2021. doi:10.4007/annals.2021.194.3.5.

[AW24] Robert Andrews and Avi Wigderson. Constant-depth arithmetic circuits for linear algebra problems. In *65th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2024, Chicago, IL, USA, October 27-30, 2024*, pages 2367–2386. IEEE, 2024. doi:10.1109/FOCS61266.2024.00138.

[AY24] Daniel Avraham and Amir Yehudayoff. On Blocky Ranks of Matrices. *Comput. Complex.*, 33(1):2, 2024. URL: https://doi.org/10.1007/s00037-024-00248-1, doi:10.1007/S00037-024-00248-1.

[Baz09] Louay M. J. Bazzi. Polylogarithmic independence can fool DNF formulas. *SIAM J. Comput.*, 38(6):2220–2272, 2009. doi:10.1137/070691954.

[BCW21] Tolson Bell, Suchakree Chueluecha, and Lutz Warnke. Note on Sunflowers. *Discrete Mathematics*, 344(8):112328, 2021. Preprint: https://arxiv.org/abs/2009.09327. arXiv:arXiv:2009.09327, doi:10.1016/j.disc.2021.112328.

[Bei11] Amos Beimel. Secret-sharing schemes: A survey. In Yeow Meng Chee, Zhenbo Guo, San Ling, Fengjing Shao, Yuansheng Tang, Huaxiong Wang, and Chaoping Xing, editors, *Coding and Cryptology - Third International Workshop, IWCC 2011, Qingdao, China, May 30-June 3, 2011. Proceedings*, volume 6639 of *Lecture Notes in Computer Science*, pages 11–46. Springer, 2011. doi:10.1007/978-3-642-20901-7_2.

[Ber84] Stuart J. Berkowitz. On computing the determinant in small parallel time using a small number of processors. *Inf. Process. Lett.*, 18(3):147–150, 1984. doi:10.1016/0020-0190(84)90018-8.

[BGW99] László Babai, Anna Gál, and Avi Wigderson. Superpolynomial lower bounds for monotone span programs. *Comb.*, 19(3):301–319, 1999. URL: https://doi.org/10.1007/s004930050058, doi:10.1007/S004930050058.

[BM25] Jaroslaw Blasiok and Linus Meierhöfer. Hardness of clique approximation for monotone circuits. In Srikanth Srinivasan, editor, *40th Computational Complexity Conference, CCC 2025, August 5-8, 2025, Toronto, Canada*, volume 339 of *LIPIcs*, pages 4:1–4:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025. URL: https://doi.org/10.4230/LIPIcs.CCC.2025.4, doi:10.4230/LIPICS.CCC.2025.4.

[Cav20] Bruno Cavalar. Sunflower theorems in computational complexity. Master's dissertation, Instituto de Matemática e Estatística, University of São Paulo, 2020. doi:10.11606/D.45.2020.tde-25112020-162107.

[CDGM22] Arkadev Chattopadhyay, Rajit Datta, Utsab Ghosal, and Partha Mukhopadhyay. Monotone complexity of spanning tree polynomial re-visited. In Mark Braverman, editor, *13th Innovations in Theoretical Computer Science Conference, ITCS 2022, January 31 - February 3, 2022, Berkeley, CA, USA*, volume 215 of *LIPIcs*, pages 39:1–39:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. URL: https://doi.org/10.4230/LIPIcs.ITCS.2022.39, doi:10.4230/LIPICS.ITCS.2022.39.

[CDM21] Arkadev Chattopadhyay, Rajit Datta, and Partha Mukhopadhyay. Lower bounds for monotone arithmetic circuits via communication complexity. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 786–799. ACM, 2021. doi:10.1145/3406325.3451069.

[CGM22] Arkadev Chattopadhyay, Utsab Ghosal, and Partha Mukhopadhyay. Robustly separating the arithmetic monotone hierarchy via graph inner-product. In Anuj Dawar and Venkatesan Guruswami, editors, *42nd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2022, December 18-20, 2022, IIT Madras, Chennai, India*, volume 250 of *LIPIcs*, pages 12:1–12:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. URL: https://doi.org/10.4230/LIPIcs.FSTTCS.2022.12, doi:10.4230/LIPICS.FSTTCS.2022.12.

[CGR+25] Bruno Cavalar, Mika Göös, Artur Riazanov, Anastasia Sofronova, and Dmitry Sokolov. Monotone circuit complexity of matching. *CoRR*, abs/2507.16105, 2025. URL: https://doi.org/10.48550/arXiv.2507.16105, arXiv:2507.16105, doi:10.48550/ARXIV.2507.16105.

[CKR22] Bruno Pasqualotto Cavalar, Mrinal Kumar, and Benjamin Rossman. Monotone circuit lower bounds from robust sunflowers. *Algorithmica*, 84(12):3655–3685, 2022. URL: https://doi.org/10.1007/s00453-022-01000-3, doi:10.1007/S00453-022-01000-3.

[CO23] Bruno Cavalar and Igor Oliveira. Constant-depth circuits vs. monotone circuits. In *Proceedings of the 38h Computational Complexity Conference (CCC)*, volume 264 of *LIPIcs*, pages 29:1–29:37. Schloss Dagstuhl, 2023. doi:10.4230/LIPIcs.CCC.2023.29.

[COS17] Xi Chen, Igor Oliveira, and Rocco Servedio. Addition is exponentially harder than counting for shallow monotone circuits. In *Proceedings of 49th Symposium on Theory of Computing (STOC)*, pages 1232–1245. ACM, 2017. doi:10.1145/3055399.3055425.

[Csa76] László Csanky. Fast parallel matrix inversion algorithms. *SIAM Journal on Computing*, 5(4):618–623, 1976. doi:10.1137/0205042.

[ĎHJ04] Pavol Ďuriš, Juraj Hromkovič, and Stasys Jukna. On multi-partition communication complexity. *Information and Computation*, 194(1):49–75, 2004. doi:10.1016/j.ic.2004.05.002.

[dRV25] Susanna F. de Rezende and Marc Vinyals. Lifting with colourful sunflowers. In Rahul Santhanam, editor, *40th Computational Complexity Conference (CCC 2025)*, volume 339 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 36:1–36:19, Dagstuhl, Germany, 2025. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CCC.2025.36.

[ER60] P. Erdös and R. Rado. Intersection theorems for systems of sets. *J. Lond. Math. Soc.*, 35(1):85–90, 1960.

[Erd45] Paul Erdős. On a lemma of Littlewood and Offord. *Bull. Amer. Math. Soc.*, 51(12):898–902, 1945.

[FGT19] Stephen A. Fenner, Rohit Gurjar, and Thomas Thierauf. A deterministic parallel algorithm for bipartite perfect matching. *Communications of the ACM*, 62(3):109–115, 2019. doi:10.1145/3306208.

[FMP+15] Samuel Fiorini, Serge Massar, Sebastian Pokutta, Hans Raj Tiwary, and Ronald de Wolf. Exponential lower bounds for polytopes in combinatorial optimization. *J. ACM*, 62(2):17:1–17:23, 2015. doi:10.1145/2716307.

[GGKS20] Ankit Garg, Mika Göös, Pritish Kamath, and Dmitry Sokolov. Monotone circuit lower bounds from resolution. *Theory Comput.*, 16:1–30, 2020. URL: https://doi.org/10.4086/toc.2020.v016a013, doi:10.4086/TOC.2020.V016A013.

[GKRS18] Mika Göös, Pritish Kamath, Robert Robere, and Dmitry Sokolov. Adventures in monotone complexity and TFNP. *Electron. Colloquium Comput. Complex.*, TR18-163, 2018. URL: https://eccc.weizmann.ac.il/report/2018/163, arXiv:TR18-163.

[GS92] Michelangelo Grigni and Michael Sipser. Monotone complexity. In *Boolean Function Complexity*, London Mathematical Society Lecture Note Series, pages 57–75. Cambridge University Press, 1992. doi:10.1017/CBO9780511526633.006.

[Hay11] Thomas P. Hayes. Separating the k-party communication complexity hierarchy: an application of the Zarankiewicz problem. *Discrete Mathematics & Theoretical Computer Science*, 13(4):15–22, 2011. doi:10.46298/dmtcs.546.

[HR00] Danny Harnik and Ran Raz. Higher lower bounds on monotone size. In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, STOC '00, page 378–387, New York, NY, USA, 2000. Association for Computing Machinery. doi:10.1145/335305.335349.

[HY09] Pavel Hrubes and Amir Yehudayoff. Monotone separations for constant degree polynomials. *Inf. Process. Lett.*, 110(1):1–3, 2009. URL: https://doi.org/10.1016/j.ipl.2009.09.003, doi:10.1016/J.IPL.2009.09.003.

[HY13] Pavel Hrubes and Amir Yehudayoff. Formulas are exponentially stronger than monotone circuits in non-commutative setting. In *Proceedings of the 28th Conference on Computational Complexity, CCC 2013, K.lo Alto, California, USA, 5-7 June, 2013*, pages 10–14. IEEE Computer Society, 2013. doi:10.1109/CCC.2013.11.

[HY16] Pavel Hrubes and Amir Yehudayoff. On Isoperimetric Profiles and Computational Complexity. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, volume 55 of *LIPIcs*, pages 89:1–89:12. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. URL: https://doi.org/10.4230/LIPIcs.ICALP.2016.89, doi:10.4230/LIPICS.ICALP.2016.89.

[HY21] Pavel Hrubes and Amir Yehudayoff. Shadows of Newton Polytopes. In Valentine Kabanets, editor, *36th Computational Complexity Conference, CCC 2021, July 20-23, 2021, Toronto, Ontario, Canada (Virtual Conference)*, volume 200 of *LIPIcs*, pages 9:1–9:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. URL: https://doi.org/10.4230/LIPIcs.CCC.2021.9, doi:10.4230/LIPICS.CCC.2021.9.

[JS20] Stasys Jukna and Hannes Seiwert. Approximation limitations of pure dynamic programming. *SIAM J. Comput.*, 49(1):170–205, 2020. doi:10.1137/18M1196339.

[Juk11] Stasys Jukna. *Extremal Combinatorics - With Applications in Computer Science*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2011. doi:10.1007/978-3-642-17364-6.

[Juk15] Stasys Jukna. Lower bounds for tropical circuits and dynamic programs. *Theory Comput. Syst.*, 57(1):160–194, 2015. doi:10.1007/s00224-014-9574-4.

[Kas67] P. W. Kasteleyn. Graph theory and crystal physics. In Frank Harary, editor, *Graph Theory and Theoretical Physics*, pages 43–110. Academic Press, London, 1967. Chapter in edited volume.

[KW93] Mauricio Karchmer and Avi Wigderson. On span programs. In *Proceedings of the Eigth Annual Structure in Complexity Theory Conference, San Diego, CA, USA, May 18-21, 1993*, pages 102–111. IEEE Computer Society, 1993. doi:10.1109/SCT.1993.336536.

[KW15] Volker Kaibel and Stefan Weltge. A short proof that the extension complexity of the correlation polytope grows exponentially. *Discrete Comput. Geom.*, 53(2):397–401, 2015. doi:10.1007/s00454-014-9655-9.

[LO43] J. E. Littlewood and A. C. Offord. On the number of real roots of a random algebraic equation. iii. *Rec. Math. [Mat. Sbornik] N.S.*, 12(54):277–286, 1943.

[Lov79] László Lovász. On determinants, matchings and random algorithms. In L. Budach, editor, *Fundamentals of Computation Theory (FCT'79)*, pages 565–574, Berlin, 1979. Akademie-Verlag.

[MS77] F. J. MacWilliams and N. J. A. Sloane. *The theory of error-correcting codes. Parts I, II*, volume 16 of *North-Holland Math. Libr.* Elsevier (North-Holland), Amsterdam, 1977.

[MU17] Michael Mitzenmacher and Eli Upfal. *Probability and computing*. Cambridge University Press, Cambridge, second edition, 2017. Randomization and probabilistic techniques in algorithms and data analysis.

[Mul87] Ketan Mulmuley. A fast parallel algorithm to compute the rank of a matrix over an arbitrary field. *Comb.*, 7(1):101–104, 1987. doi:10.1007/BF02579205.

[MV97] Meena Mahajan and V. Vinay. Determinant: Combinatorics, algorithms, and complexity. *Chic. J. Theor. Comput. Sci.*, 1997, 1997. URL: http://cjtcs.cs.uchicago.edu/articles/1997/5/contents.html.

[Nis91] Noam Nisan. Lower bounds for non-commutative computation (extended abstract). In Cris Koutsougeras and Jeffrey Scott Vitter, editors, *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5-8, 1991, New Orleans, Louisiana, USA*, pages 410–418. ACM, 1991. doi:10.1145/103418.103462.

[NNP04] Ventzislav Nikov, Svetla Nikova, and Bart Preneel. On the size of monotone span programs. In Carlo Blundo and Stelvio Cimato, editors, *Security in Communication Networks, 4th International Conference, SCN 2004, Amalfi, Italy, September 8-10, 2004, Revised Selected Papers*, volume 3352 of *Lecture Notes in Computer Science*, pages 249–262. Springer, 2004. doi:10.1007/978-3-540-30598-9_18.

[Oko82] Elizaveta Okol'nishnikova. The effect of negations on the complexity of realization of monotone Boolean functions by formulas of bounded depth. *Metody Diskretnogo Analiza*, (38):74–80, 1982.

[PS96] Pavel Pudlák and Jirí Sgall. Algebraic models of computation and interpolation for algebraic proof systems. In Paul Beame and Samuel R. Buss, editors, *Proof Complexity and Feasible Arithmetics, Proceedings of a DIMACS Workshop, New Brunswick, New Jersey, USA, April 21-24, 1996*, volume 39 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 279–295. DIMACS/AMS, 1996. URL: https://doi.org/10.1090/dimacs/039/15, doi:10.1090/DIMACS/039/15.

[Rao25] Anup Rao. The story of sunflowers, 2025. URL: https://arxiv.org/abs/2509.14790, arXiv:2509.14790.

[Raz85] Alexander Razborov. Lower bounds on the monotone complexity of some Boolean functions. *Dokl. Akad. Nauk SSSR*, 281(4):798–801, 1985.

[Raz92] Alexander A. Razborov. On the Distributional Complexity of Disjointness. *Theor. Comput. Sci.*, 106(2):385–390, 1992. doi:10.1016/0304-3975(92)90260-M.

[Ros14] Benjamin Rossman. The monotone complexity of k-clique on random graphs. *SIAM Journal on Computing*, 43(1):256–279, 2014. doi:10.1137/110839059.

[Rou15] Tim Roughgarden. Communication complexity (for algorithm designers), 2015. URL: https://arxiv.org/abs/1509.06257.

[RW92] Ran Raz and Avi Wigderson. Monotone circuits for matching require linear depth. *Journal of the ACM*, 39(3):736–744, 1992. doi:10.1145/146637.146684.

[RY11] Ran Raz and Amir Yehudayoff. Multilinear formulas, maximal-partition discrepancy and mixed-sources extractors. *J. Comput. Syst. Sci.*, 77(1):167–190, 2011. doi:10.1016/j.jcss.2010.06.013.

[Sho88] Victor Shoup. New algorithms for finding irreducible polynomials over finite fields. In *29th Annual Symposium on Foundations of Computer Science, White Plains, New York, USA, 24-26 October 1988*, pages 283–290. IEEE Computer Society, 1988. doi:10.1109/SFCS.1988.21944.

[Sri20] Srikanth Srinivasan. Strongly exponential separation between monotone VP and monotone VNP. *ACM Trans. Comput. Theory*, 12(4):12, 2020. Id/No 23. doi:10.1145/3417758.

[Sta97] Richard P. Stanley. *Enumerative combinatorics. Vol. 1.*, volume 49 of *Camb. Stud. Adv. Math.* Cambridge: Cambridge University Press, 1997.

[SY10] Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Found. Trends Theor. Comput. Sci.*, 5(3-4):207–388, 2010. doi:10.1561/0400000039.

[Tal17] Avishay Tal. Tight bounds on the fourier spectrum of AC0. In Ryan O'Donnell, editor, *32nd Computational Complexity Conference, CCC 2017, July 6-9, 2017, Riga, Latvia*, volume 79 of *LIPIcs*, pages 15:1–15:31. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. URL: https://doi.org/10.4230/LIPIcs.CCC.2017.15, doi:10.4230/LIPICS.CCC.2017.15.

[Tar88] Éva Tardos. The gap between monotone and non-monotone circuit complexity is exponential. *Comb.*, 8(1):141–142, 1988. doi:10.1007/BF02122563.

[Val79] Leslie G. Valiant. Completeness classes in algebra. In Michael J. Fischer, Richard A. DeMillo, Nancy A. Lynch, Walter A. Burkhard, and Alfred V. Aho, editors, *Proceedings of the 11h Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1979, Atlanta, Georgia, USA*, pages 249–261. ACM, 1979. doi:10.1145/800135.804419.

[Val80] Leslie G. Valiant. Negation can be exponentially powerful. *Theor. Comput. Sci.*, 12:303–314, 1980. doi:10.1016/0304-3975(80)90060-2.

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

[Yeh19] Amir Yehudayoff. Separating monotone VP and VNP. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 425–429. ACM, 2019. doi:10.1145/3313276.3316311.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il