Electronic Colloquium on Computational Complexity, Report No. 81 (2021)

Superpolynomial Lower Bounds Against Low-Depth Algebraic Circuits

Nutan Limaye *[1], Srikanth Srinivasan[2], and Sébastien Tavenas †[3]

[1]IIT Bombay
[2]Aarhus University
[3]Univ. Grenoble Alpes, Univ. Savoie Mont Blanc, CNRS, LAMA

Abstract

An Algebraic Circuit for a polynomial $P \in \mathbb{F}[x_1, \dots, x_N]$ is a computational model for constructing the polynomial P using only additions and multiplications. It is a *syntactic* model of computation, as opposed to the Boolean Circuit model, and hence lower bounds for this model are widely expected to be easier to prove than lower bounds for Boolean circuits. Despite this, we do not have superpolynomial lower bounds against general algebraic circuits of depth 3 (except over constant-sized finite fields) and depth 4 (over any field), while constant-depth Boolean circuit lower bounds have been known since the early 1980s.

In this paper, we prove the *first superpolynomial lower bounds against general algebraic circuits of all constant depths* over all fields of characteristic 0 (or large). We also prove the first lower bounds against *homogeneous* algebraic circuits of constant depth over any field.

Our approach is surprisingly simple. We first prove superpolynomial lower bounds for constant-depth *Set-Multilinear* circuits. While strong lower bounds were already known against such circuits, most previous lower bounds were of the form $f(d) \cdot \mathrm{poly}(N)$, where d denotes the degree of the polynomial. In analogy with Parameterized complexity, we call this an *FPT* lower bound. We extend a well-known technique of Nisan and Wigderson (FOCS 1995) to prove *non-FPT* lower bounds against constant-depth set-multilinear circuits computing the Iterated Matrix Multiplication polynomial $\mathrm{IMM}_{n,d}$ (which computes a fixed entry of the product of d $n \times n$ matrices). More precisely, we prove that any set-multilinear circuit of depth Δ computing $\mathrm{IMM}_{n,d}$ must have size at least $n^{d^{\exp(-O(\Delta))}}$. This result holds over any field.

We then show how to convert any constant-depth algebraic circuit of size s to a *constant-depth* set-multilinear circuit with a blow-up in size that is exponential in d but only polynomial in s over fields of characteristic 0. (For depths greater than 3, previous results of this form increased the depth of the resulting circuit to $\Omega(\log s)$.) This implies our constant-depth circuit lower bounds for d that is a slow-growing function of n.

Finally, we observe that our superpolynomial lower bound for constant-depth circuits implies the first deterministic sub-exponential time algorithm for solving the Polynomial Identity Testing (PIT) problem for all small depth circuits using the known connection between algebraic hardness and randomness.

Acknowledgement

The authors are grateful to Ramprasad Saptharishi for spending time to discuss the proof with us.

*Funded by SERB project File no. MTR/2017/000909.)

†Funded by the project BIRCA from the IDEX of Univ. Grenoble Alpes (Contract 183459).

ISSN 1433-8092

1 Introduction

Background on Algebraic Circuits. Let $P(x_1, \ldots, x_N)$ be a multivariate polynomial over a field $\mathbb{F}$. An *Algebraic Circuit* for $P(x_1, \ldots, x_N)$ is simply a circuit for constructing P using the input variables and constants from $\mathbb{F}$, by combining them iteratively using additions and multiplications. This construction may be visualized as a DAG, with leaves that are labelled by variables from $\{x_1, \ldots, x_N\}$ or field elements and internal nodes that either compute products or linear combinations of their inputs.[1] A special output node (or gate) represents the polynomial P. In the particular case where the DAG is a tree, such a circuit is called an *Algebraic Formula*.[2] The *size* of this construction is the number of nodes in the DAG. We also consider the *product-depth* of the circuit, which is the maximum number of product gates on a root-to-leaf path.[3]

We think of such an algebraic circuit as a computational model, solving the computational task of evaluating P at a given input $(x_1, \ldots, x_N) \in \mathbb{F}^N$. The efficiency of the model is measured by its size, which closely approximates the number of operations performed in the computation. As the circuit is required to construct the formal polynomial P, it is a *syntactic* model of computation, as opposed to the Boolean circuit model, which is only required to model certain input-output behaviours. As a consequence, the problem of proving algebraic circuit lower bounds is widely considered to be easier than its Boolean counterpart. Indeed, it is known that separating VP from VNP, the algebraic analog of the P vs. NP problem, is a prerequisite to solving the latter problem (in the non-uniform setting) [6].

As a result, proving lower bounds for algebraic circuits has been the focus of a large body of research (see, e.g. [7, 52, 45] for nice introductions to this area). Unfortunately, however, we are far from resolving the big questions. For instance, we do not even have superpolynomial lower bounds against general algebraic circuits of product-depth 1, which are also called $\Sigma\Pi\Sigma$ formulas (as they are linear combinations of products of linear combinations of the input variables), over fields of large size, and no superpolynomial lower bounds against general algebraic circuits of product-depth more than 1 (e.g. $\Sigma\Pi\Sigma\Pi$ formulas). Note that, in contrast, we have had strong constant-depth *Boolean* circuit lower bounds since the early 1980s [1, 16, 20, 43, 53].

In this paper, we prove the first superpolynomial lower bounds for algebraic circuits of constant product-depth. Our lower bounds hold over all fields of characteristic 0 (or large enough as a function of the degree parameter).

Theorem 1 (Main Result). *Let N, d, Δ be growing parameters with $d = o(\log N)$. Assume $\mathbb{F}$ has characteristic 0 or greater than d. There is an explicit polynomial $P_{N,d}(x_1, \ldots, x_N)$ that has no algebraic circuits of product-depth Δ and size at most $N^{d^{\exp(-O(\Delta))}}$.*

Moreover, the polynomial $P_{N,d}$ is a well-known polynomial that is easy to describe. Assume n and d are such that $N = dn^2$. The polynomial $P_{N,d}$ is the Iterated Matrix Multiplication polynomial $\mathrm{IMM}_{n,d}$ on $N = dn^2$ variables, defined as follows. The underlying variables are partitioned into d sets $X_1, \ldots, X_d$ of size n^2, each of which is visualized as an $n \times n$ matrix with distinct variable entries. Then $\mathrm{IMM}_{n,d}$ is defined to be the polynomial that is the $(1,1)$th entry of the product matrix $X_1 \cdot X_2 \cdots X_d$.

[1] More precisely, any internal node v with children $u_1, \ldots, u_r$ is labelled either × or +. In the former case, the nodes computes the product of its inputs. In the latter case, it computes a linear combination of the inputs, where the coefficients of the linear combination are field elements labelling the edges between the u_is and v.

[2] Another natural way to define it is that it is just a (possibly nested) algebraic expression made up of variables, constants, additions and multiplications.

[3] One can also consider the *depth* of the formula, which is the maximum length of a root-to-leaf path. The product-depth is, w.l.o.g., equal to depth up to a factor of two. It is sometimes easier to state results for algebraic circuits in terms of product-depth, and this is true for our results as well.

The Approach: 'Hardness Escalation'. While lower bounds for general algebraic circuits have been hard to prove, we do have several beautiful results for restricted kinds of algebraic circuits, such as *Homogeneous, Multilinear,* and *Set Multilinear* circuits. As these will be useful in the sequel, we review some of these definitions below.

Recall that a multilinear polynomial $P(x_1, \ldots, x_N)$ is one in which each variable x_i has degree at most 1, and a homogeneous polynomial is one that is a linear combination of monomials of the same total degree. If the underlying variable set is partitioned into d variable sets $X_1, \ldots, X_d$, then P is said to be *set-multilinear* with respect to this variable partition if P is a linear combination of monomials that contain one variable from each variable set among $X_1, \ldots, X_d$; note that a set-multilinear polynomial is both multilinear and homogeneous (of degree d). For example, the $n \times n$ Determinant is a set-multilinear polynomial w.r.t the variable partition corresponding to the rows of the underlying matrix, and the polynomial $\mathrm{IMM}_{n,d}$ defined above is set-multilinear w.r.t. the partition into matrices $X_1, \ldots, X_d$.

Given a set-multilinear polynomial P (w.r.t. variable partition $X_1, \ldots, X_d$), it is natural to look at algebraic circuits computing P that themselves have the same structure. In particular, an algebraic circuit is said to be set-multilinear if each internal gate computes a set-multilinear polynomial in a subset of $X_1, \ldots, X_d$. Similarly, a multilinear or homogeneous circuit is one where each internal node computes a multilinear or homogeneous polynomial respectively. For each such restricted type of circuit, we have non-trivial lower bounds on the sizes of circuit computing explicit polynomials (also restricted in the same way) [36, 39, 57, 13, 27, 33]. An important result of Nisan and Wigderson [36] proved lower bounds against small-depth set-multilinear and homogeneous circuits computing $\mathrm{IMM}_{n,d}$. Building upon this, Raz [39] showed superpolynomial lower bounds on the size of any (unbounded depth) multilinear *formula* computing the $n \times n$ Determinant and Permanent.

It is natural to ask if we can use these lower bounds against restricted kinds of circuits to prove lower bounds against more general algebraic circuits. Such 'hardness escalation'[4] results have appeared in many areas in computational complexity (see, e.g. [2, 42]), including Algebraic complexity theory. Strassen [54] and Raz [40] both observed (in different settings) that lower bounds for small-depth circuits computing low-degree polynomials imply lower bounds for larger depth circuits. More recently, Raz [41] showed that if a homogeneous or set-multilinear polynomial of degree d has an algebraic formula of size s, then it also has a *homogeneous* or *set-multilinear* formula of size $\mathrm{poly}(s) \cdot (\log s)^{O(d)}$ respectively. In particular, for a homogeneous (resp. set-multilinear) polynomial P of degree $d = O(\log N / \log\log N)$, it follows that P has a formula of size $\mathrm{poly}(N)$ if and only if P has a homogeneous (resp. set-multilinear) formula of size $\mathrm{poly}(N)$.[5]

The latter result implies that if we could prove homogeneous or set-multilinear formula lower bounds of the form $N^{\omega_d(1)}$ (i.e. the exponent goes to infinity with d) for a polynomial P with N variables and degree d, then we would have superpolynomial general algebraic formula lower bounds. In particular, this would imply lower bounds for constant-depth algebraic circuits, as any constant-depth algebraic circuit can be converted to an algebraic formula with only polynomial blow-up.

Unfortunately, known lower bounds do not yield such lower bounds. In the homogeneous case, we have strong lower bounds against certain formulas of product-depth at most 2 [36, 27, 33], but this falls short of proving anything for general formulas as Raz's 'homogenization' result does not preserve the product-depth of the formula (in fact, known results for homogeneous formulas stop yielding lower bounds exactly in the regime where they would yield implications

[4] This terminology appeared in a result of Beame, Huynh and Pitassi [2] on proof complexity. The authors of that paper attribute the term to Rahul Santhanam.

[5] Raz's result is slightly stronger for homogeneous formulas, but we ignore this point here.

for general circuits). In the set-multilinear, and more generally multilinear case, we do have lower bounds against formulas of large depth [36, 39, 57], but all such lower bounds are of the form $f(d) \cdot \mathrm{poly}(N)$ where $f(d)$ is a superpolynomial (and subexponential) function of d (see Appendix A). With analogy to *Parameterized Complexity Theory* [12], we call such bounds *FPT bounds.* Our motivating question is if we can prove strong *non-FPT lower bounds* against restricted types of circuits or formulas in a setting where we can use them for lower bounds for general algebraic circuits or formulas. We show that this is indeed possible.

Our results. Our main lower bound result is a strong non-FPT lower bound against small-depth set-multilinear circuits, considerably strengthening known results in this direction.

We prove our lower bounds for the $\mathrm{IMM}_{n,d}$ polynomial on $N = dn^2$ variables as defined above. This polynomial has a simple divide-and-conquer-based set-multilinear formula of size $n^{O(\log d)}$, and more generally for every $\Delta \leqslant \log d$, a set-multilinear formula of product-depth Δ and size $n^{O(\Delta d^{1/\Delta})}$. Even relaxing the set-multilinearity constraint, no considerably better upper bound is known. This is despite much work on this problem and close connections to important algorithmic problems such as Graph Reachability [56, 44]. It is reasonable to conjecture that this simple upper bound is tight up to the constant in the exponent.

This was proved for homogeneous $\Sigma\Pi\Sigma$ circuits by Nisan and Wigderson [36]. For product-depth $\Delta > 1$, they proved an FPT lower bound of $\exp(\Omega(d^{1/\Delta})) \cdot \mathrm{poly}(n)$ in the set-multilinear case. More recently, building on work of Kayal [25] and Gupta, Kamath, Kayal and Saptharishi [17], Fournier, Limaye, Malod and Srinivasan [15] showed that any set-multilinear $\Sigma\Pi\Sigma\Pi$ circuit for $\mathrm{IMM}_{n,d}$ must have size $n^{\Omega(\sqrt{d})}$, again showing the tightness of the naive upper bound. This was extended to homogeneous $\Sigma\Pi\Sigma\Pi$ circuits by Kayal, Limaye, Saha and Srinivasan [26] and Kumar and Saraf [33]. Kayal, Nair and Saha [28] extended the $\Sigma\Pi\Sigma$ lower bound of [36] to the more general multilinear setting, while Kayal, Saha and Tavenas [30] strengthened the result of [15] to the multilinear setting. Note that all these results show non-FPT lower bounds against special cases of product-depth 2 circuits.

However, as far as we know, no superpolynomial non-FPT lower bounds are known for any product-depths greater than 2 (or even for general product-depth 2, which is $\Sigma\Pi\Sigma\Pi\Sigma$), even under the set-multilinearity restriction. We show such lower bounds for all constant product-depths, and in fact, product-depths that are asymptotically smaller than $\log\log d$.

Theorem 2 (Lower bound for set-multilinear circuits)**.** *Assume $d \leqslant (\log n)/100$. For any product-depth $\Delta \geqslant 1$, any set-multilinear circuit C computing $\mathrm{IMM}_{n,d}$ of product-depth at most Δ must have size at least $n^{d^{\exp(-O(\Delta))}}$. In the particular case that $\Delta = 2$, the size of C must be at least $n^{\Omega(\sqrt{d})}$.*

Note that in the case of $\Delta = 2$, our bounds match the best-known (divide-and-conquer) upper bound for computing $\mathrm{IMM}_{n,d}$.

With these stronger non-FPT lower bounds for set-multilinear circuits in place, we are able to derive lower bounds for stronger families of algebraic circuits via hardness escalation arguments.

Firstly, we show (Lemma 11) that any homogeneous circuit of product-depth Δ and size s computing a set-multilinear polynomial P of degree d can be converted to a set-multilinear circuit *with the same product-depth* for P of size $s \cdot d^{O(d)}$. Putting this together with Theorem 2, we get the first superpolynomial lower bounds (FPT or non-FPT) for homogeneous circuits of product-depth greater than 2 (and even $\Sigma\Pi\Sigma\Pi\Sigma$ homogeneous circuits over large fields).

Corollary 3 (Lower bound for homogeneous circuits)**.** *Assume $d \leqslant (\log n)/100$. For any product-depth $\Delta \geqslant 1$, any homogeneous circuit C computing $\mathrm{IMM}_{n,d}$ of product-depth at most*

Δ must have size at least $n^{d^{\exp(-O(\Delta))}}$. In the particular case that $\Delta = 2$, the size of C must be at least $n^{\Omega(\sqrt{d})}$.

Both Theorem 2 and Corollary 3 hold over any field $\mathbb{F}$. Note that our improved non-FPT bounds are crucial for deriving the above result from Theorem 2. The previous best lower bound of $\exp(\Omega(\sqrt{d}))$ due to Nisan and Wigderson [36] does not suffice for this.

Next, we show (Lemma 10) that any (possibly non-homogeneous) algebraic circuit of product-depth Δ and size s computing a homogeneous polynomial P of degree d can be converted to a homogeneous circuit for P of product-depth 2Δ and size $\mathrm{poly}(s) \cdot d^{O(d)}$. This conversion assumes that the underlying field has characteristic 0 or greater than d. This implies the main theorem Theorem 1. More precisely, we get the following.

Corollary 4 (Lower bound for constant-depth circuits)**.** *Assume $d \leqslant (\log n)/100$ and $\mathrm{char}(\mathbb{F}) = 0$ or greater than d. For any product-depth $\Delta \geqslant 1$, any algebraic circuit C computing $\mathrm{IMM}_{n,d}$ of product-depth at most Δ must have size at least $n^{d^{\exp(-O(\Delta))}}$. In the particular case that $\Delta = 1$, the size of C must be at least $n^{\Omega(\sqrt{d})}$.*

In the case $\Delta = 1$, our bound is actually tight, by a beautiful upper bound due to Gupta, Kamath, Kayal and Saptharishi [18].

In comparison, in the particular case of depth-3 circuits, the best lower bound known for an explicit polynomial was a quadratic lower bound by Shpilka and Wigderson [51] which was then improved to an almost cubic lower bound in [29]. In the case of depth-4 circuits, Gupta Saha and Thankey [19] recently got a $\tilde{\Omega}(N^{2.5})$ lower bound improving the previous bound from [49]. To our knowledge, for depth $\Delta = 5$ or larger, the best lower bound known is $\Omega(\Delta N^{1+1/\Delta})$ which has been found by Shoup and Smolensky [50] and Raz [40].

Finally, we note that our superpolynomial lower bound (Theorem 1) implies a deterministic sub-exponential time algorithm for *Polynomial Identity Testing* (PIT) of constant-depth circuits.

Kabanets and Impagliazzo [23] established a formal connection between the two most important problems in algebraic complexity theory, namely, the problem of proving superpolynomial lower bounds for algebraic circuits and that of designing efficient deterministic PIT algorithms. Specifically, using the *Hardness versus Randomness* framework of Nisan and Wigderson [35] they showed that superpolynomial lower bounds for general algebraic circuits imply deterministic sub-exponential time algorithms for general PIT.

Recent results have tried to extend this *algebraic hardness vs. randomness* framework in several different ways [14, 11, 32]. Specifically, Dvir, Shpilka, and Yehudayoff [14] proved that the hardness of constant-depth circuits implies deterministic PIT for constant depth circuits. In a recent follow up paper, Chou, Kumar and Solomon [11] refined this result and improved the dependence on the degree of the polynomial.

We observe that this result from [11] combined with our lower bound from Theorem 1 gives the first sub-exponential time deterministic PIT for constant depth circuits. Specifically, we get the following.

Corollary 5. *Let $\varepsilon > 0$ be a real number. Let C be an algebraic circuit of size s, depth $\Delta = o(\log\log\log n)$ computing a polynomial on n variables, then there is a deterministic algorithm that can check whether the polynomial computed by C is identically zero or not in time $(s^{\Delta+1} \cdot n)^{O(n^{2\varepsilon})}$.*

As the general PIT problem is a well-known hard problem, several special cases of the problem have been considered. More specifically, constant-depth circuits have gained a lot of attention in the literature. See for instance [24, 46, 3, 37, 38] and references therein.

In spite of years of efforts, the problem continues to be notoriously open. Even today, no polynomial time deterministic algorithm is known for even product-depth 1 circuits. For $\Sigma\Pi\Sigma$ circuits, the best known upper bound is due to Seshadri and Saxena [47] which gives a $n^{O(k)}$ time deterministic algorithm, where k is the fan-in of the top Σ gate. This result gives polynomial upper bound for bounded top fan-in, but for the general case of unbounded top fan-in, this does not do better than a brute-force algorithm. Here, we obtain the first sub-exponential time deterministic algorithm for general $\Sigma\Pi\Sigma$ circuits, and more generally for circuits of any constant depth.

Our Techniques. Our lower bound techniques are simple adaptations of the *Partial Derivative method* from the paper of Nisan and Wigderson [36]. In particular, we show that this method, when applied to set-multilinear polynomials in the setting where the variables are *partitioned into sets of various sizes*, can prove considerably stronger lower bounds than previously known.

Interestingly, we do not use the *Shifted Partial Derivative method* that has proven useful in proving many previous lower bounds for circuits of product-depth greater than 1 [25, 17, 15, 26, 27, 33, 30, 29]. We leave as open the question of whether augmenting our methods with 'shifts' can prove stronger lower bounds.

Our 'set-multilinearization' argument is elementary, but does not seem to appear anywhere in the literature (however, see [10, Theorem 5.10] for a special case of this argument for $\Sigma\Pi\Sigma\Pi$ circuits). Our 'homogenization' argument uses a generalization of classical Newton Identities to derive homogeneous $\Sigma\Pi\Sigma\Pi$ formulas for certain interesting 'weighted' symmetric polynomials. In the case of $\Sigma\Pi\Sigma$ circuits, it follows from the work of Shpilka and Wigderson [51], as observed in [18, Section 5.2 of the journal version] and in Saptharishi's survey [45, Lemma 23.6].

Other non-FPT bounds. Apart from the above-mentioned work, non-FPT lower bounds have also been proved in some other models of algebraic computation.

A setting where many strong lower bounds are known for algebraic problems is that of *Monotone* circuits. Here, the underlying field is the reals and the given polynomial $P \in \mathbb{R}[x_1, \ldots, x_N]$ has non-negative coefficients. A monotone circuit for P is an algebraic circuit that does not use any negative field constants. Exponential lower bounds against monotone circuits have been known since the work of Jerrum and Snir [22]. It is also known by work of Shamir and Snir [48] that any monotone algebraic formula for $\mathrm{IMM}_{n,d}$ must have size $n^{\Omega(\log d)}$. A similar lower bound for an even simpler polynomial was proved by Hrubeš and Yehudayoff [21]. Unfortunately, these results do not seem to imply general formula or circuit lower bounds, as it is not clear how to efficiently convert a general algebraic circuit or formula to a monotone one: in fact, there is strong indication that this might be impossible [55, 9, 8].

Another setting where non-FPT lower bounds are known is in that of *Non-commutative* computation. Here, we assume that the underlying variables $x_1, \ldots, x_N$ do not commute. This implies that upper bounds get harder, and lower bounds easier. Nisan [34] showed exponential lower bounds for algebraic formulas and more generally *Algebraic Branching Programs* and his results imply, in particular, non-FPT lower bounds for these models.

Organization. We start with some preliminaries and then present a special case of our argument in Section 4, which already implies explicit lower bounds for homogeneous $\Sigma\Pi\Sigma\Pi\Sigma$ circuits and general $\Sigma\Pi\Sigma$ circuits, both of which are well-known open questions in their own right [36, 51, 29, 4, 31]. We then present the proof of Theorem 2 and the ensuing corollaries.

2 Preliminaries

We will consider the set of words on an alphabet $A \subseteq \mathbb{Z}\backslash\{0\}$. Let $w = (w_1, \ldots, w_d) \in A^d$. For a subset $S \subseteq [d]$, let w_S denote $\sum_{i\in S} w_i$. We define $\mathcal{P}_w = \{i \mid w_i \geqslant 0\}$ and $\mathcal{N}_w = \{i \mid w_i < 0\}$, i.e., the positive and negative indices of w respectively.

We say w is *b-unbiased* if $|w_I| \leqslant b$ for each interval $I \subseteq [d]$. Note that if w is b-unbiased, then, in particular, $|w_i| \leqslant b$ for each $i \in [d]$.

Given w, we denote by $\overline{X}(w)$ a tuple of d sets of variables $(X(w_1), \ldots, X(w_d))$ where $|X(w_i)| = 2^{|w_i|}$. We denote by $\mathbb{F}_{\text{sm}}[\mathcal{T}]$ the set of set-multilinear polynomials over the tuple of sets of variables $\mathcal{T}$.

2.1 The complexity measure

Let $\mathcal{M}_w^{\mathcal{P}}$ and $\mathcal{M}_w^{\mathcal{N}}$ denote the sets of the set-multilinear monomials over only the positive and only the negative variable sets. Let $f \in \mathbb{F}_{\text{sm}}[\overline{X}(w)]$, we define $M_w(f)$ as the matrix of size $|\mathcal{M}_w^{\mathcal{P}}| \times |\mathcal{M}_w^{\mathcal{N}}|$, where the rows are indexed by $\mathcal{M}_w^{\mathcal{P}}$ and the columns by $\mathcal{M}_w^{\mathcal{N}}$ and where the coefficient at the entry (m_1, m_2) is the coefficient of the monomial $m_1 m_2$ in f.

We associate with the space $\mathbb{F}_{\text{sm}}[\overline{X}(w)]$ the standard rank-based complexity measure relrk_w (short for "relative rank") defined as follows. Let $f \in \mathbb{F}_{\text{sm}}[\overline{X}(w)]$ and define

$$\text{relrk}_w(f) = \frac{\text{rank}(M_w(f))}{\sqrt{|\mathcal{M}_w^{\mathcal{P}}| \cdot |\mathcal{M}_w^{\mathcal{N}}|}} = \frac{\text{rank}(M_w(f))}{2^{\frac{1}{2}\sum_{i\in[d]}|w_i|}} \leqslant 1.$$

We use the following properties of relrk_w.

Claim 6. 1. *(Imbalance) Say $f \in \mathbb{F}_{sm}[\overline{X}(w)]$. Then,* $\text{relrk}_w(f) \leqslant 2^{-|w_{[d]}|/2}$.

2. *(Sub-additivity) Say $f, g \in \mathbb{F}_{sm}[\overline{X}(w)]$. Then* $\text{relrk}_w(f + g) \leqslant \text{relrk}_w(f) + \text{relrk}_w(g)$.

3. *(Multiplicativity) Say $f = f_1 \cdot f_2 \cdot \ldots \cdot f_t$ and assume that for each $i \in [t]$, $f_i \in \mathbb{F}_{sm}[\overline{X}(w_{|S_i})]$, where $(S_1, \ldots, S_t)$ is a partition of $[d]$ and for each $i \in [t]$, $w_{|S_i}$ stands for the sub-word of w indexed by S_i. Then*

$$\text{relrk}_w(f) = \text{relrk}_w(f_1 \cdot f_2 \cdot \ldots \cdot f_t) = \prod_{i\in[t]} \text{relrk}_{w_{|S_i}}(f_i).$$

Proof. We have $|\mathcal{M}_w^{\mathcal{P}}| = 2^{\sum_{i\in\mathcal{P}_w} w_i}$ and $|\mathcal{M}_w^{\mathcal{N}}| = 2^{-\sum_{i\in\mathcal{N}_w} w_i}$. So $2^{|w_{[d]}|}$ is just the ratio of the larger dimension of $M_w(f)$ by the smaller one. As the rank of a matrix is bounded by the minimum between its number of rows and its number of columns, it implies the first inequality of the claim.

The subadditivity property directly follows from the facts that $M_w(f+g) = M_w(f) + M_w(g)$ and that the rank of a matrix is subadditive.

The multiplicative argument is standard too. As the product is set-multilinear, it implies that the matrix $M_w(f_1 \cdot \ldots \cdot f_t)$ is the matrix $M_w(f_1) \otimes \ldots \otimes M_w(f_t)$ where the symbol $\otimes$ stands for the Kronecker product. Finally the rank is known to be multiplicative with respect to the Kronecker product. So,

$$\text{relrk}_w(f_1 \cdot f_2 \cdot \ldots \cdot f_t) = \frac{\text{rank}(M_w(f_1 \cdot \ldots \cdot f_t))}{2^{\frac{1}{2}\sum_{j\in[d]}|w_j|}} = \prod_{i\in[t]} \frac{\text{rank}(M_w(f_i))}{2^{\frac{1}{2}\sum_{j\in S_i}|w_j|}} = \prod_{i\in[t]} \text{relrk}_{w_{|S_i}}(f_i).$$

□

2.2 Word Polynomials and Iterated Matrix Multiplication polynomial

Let $w \in A^d$ be any word. For any such word, we define a polynomial P_w. Say $X(w) = (X_1, \ldots, X_d)$ and since each X_i has size $2^{|w_i|}$, we assume that the variables of X_i are labelled by strings in $\{0,1\}^{|w_i|}$.

Given any monomial $m \in \mathbb{F}_{\mathrm{sm}}[\overline{X}(w)]$, let m_+ denote the corresponding "positive" monomial from $\mathcal{M}_w^{\mathcal{P}}$ and m_- the corresponding "negative" monomial from $\mathcal{M}_w^{\mathcal{N}}$. As each variable of $\overline{X}(w)$ is labelled by a Boolean string and each set-multilinear monomial over any subset of $\overline{X}(w)$ is associated with a string of variables, we can associate any such monomial m' with a Boolean string $\sigma(m')$. More precisely, if $j_1 < \cdots < j_t$ and $m_1 = x_{\sigma_1}^{(j_1)} x_{\sigma_2}^{(j_2)} \cdots x_{\sigma_t}^{(j_t)}$ with $x_{\sigma_i}^{(j_i)} \in X_{j_i}$ and $\sigma_i \in \{0,1\}^{|w_{j_i}|}$ for each $i \in [t]$, then $\sigma(m')$ is defined to be $\sigma_1 \cdots \sigma_t$. If w is b-unbiased, the difference of length of the strings $\sigma(m_+)$ and $\sigma(m_-)$ is at most b. We will write $\sigma(m_+) \sim \sigma(m_-)$ when the shorter one is a prefix of the other one.

The polynomial P_w is defined as follows

$$P_w = \sum_{m \in \mathbb{F}[\overline{X}(w)],\ \ \sigma(m_+) \sim \sigma(m_-)} m.$$

Clearly, the matrices $M_w(P_w)$ are full-rank (i.e. have rank equal to either the number of rows or the number of columns, whichever is smaller). So, $\mathrm{relrk}_w(P_w) = 2^{-|w_{[d]}|/2} \geqslant 2^{-b/2}$.

We observe that P_w can be obtained as a *set-multilinear restriction* of $\mathrm{IMM}_{n,d}$ for an appropriate choice of n. Formally, we show the following.

Lemma 7. *Let $w \in A^d$ be any word which is b-unbiased. If there is a set-multilinear circuit computing $\mathrm{IMM}_{2^b,d}$ of size s and product-depth Δ, then there is also a set-multilinear circuit of size s and product-depth Δ computing a polynomial $P_w \in \mathbb{F}_{sm}[\overline{X}(w)]$ such that $\mathrm{relrk}_w(P_w) \geqslant 2^{-b/2}$.*

The proof of the lemma is presented in Section 8.

3 Set-multilinearization of small depth circuits

In the next sections we will show superpolynomial lower bounds for small-degree polynomials against set-multilinear formulas of various product-depths. We want to extend these lower bounds to the general setting (i.e., without the set-multilinearity constraint).

In [41], Raz showed that if there is a fanin-2 formula of size s and product-depth Δ that computes a set-multilinear polynomial over the disjoint sets $(X_1, \ldots, X_d)$, then there exists also a fanin-2 set-multilinear formula of size $O((\Delta+2)^d s)$ and product-depth Δ that computes the same polynomial. However the fanin-2 constraint is an issue when we want to deal with constant depth circuits.

We show here that we can get a similar result for circuits with arbitrary fanins at the cost of a size blow-up of $d^{O(d)} \mathrm{poly}(s)$ and an increase of the depth by a factor of at most 2.

Proposition 8. *Let s, N, d, Δ be growing parameters with $s \geqslant Nd$. Assume that $\mathrm{char}(\mathbb{F}) = 0$ or $\mathrm{char}(\mathbb{F}) > d$. If C is a circuit of size at most s and product-depth at most Δ computing a set-multilinear polynomial P over the sets of variables $(X_1, \ldots, X_d)$ (with $|X_i| \leqslant N$), then there is a set-multilinear circuit of size $d^{O(d)} \mathrm{poly}(s)$ and product-depth at most 2Δ computing P.*

Similar to Raz's approach, we start by homogenizing the circuit and then we set-multilinearize it. In particular the previous proposition is just the composition of Lemmas 10 and 11.

Non-homogeneous to homogeneous circuits. In this section, we state lemmas that convert non-homogeneous formulas of small product-depth Δ to homogeneous formulas of product-depth 2Δ with a relatively small size blow-up.

Let us begin by recalling how to do it in the case of product-depth 1. A general $\Sigma\Pi\Sigma$ circuit of size s yields a formula of the following kind

$$F = \sum_{i=1}^{s} \prod_{j=1}^{s} \ell_{i,j}$$

where each $\ell_{i,j}$ is an affine linear polynomial in the underlying variables. Note that the individual summands of the expression may compute polynomials of degree s, which is possibly much larger than d. The main observation is that, assuming that the underlying field $\mathbb{F}$ has characteristic 0 (or large enough), the homogeneous degree-d part of each summand can be computed by a homogeneous $\Sigma\Pi\Sigma\Pi\Sigma$ formula of size $\mathrm{poly}(s) \cdot \exp(O(\sqrt{d}))$. Replacing each of these terms with such a formula, we see then that the same polynomial can also be computed by a homogeneous $\Sigma\Pi\Sigma\Pi\Sigma$ formula of size $\mathrm{poly}(s) \cdot \exp(O(\sqrt{d}))$.

The main observation is also easy to prove. Consider any summand $T_i = \ell_{i,1} \cdots \ell_{i,s}$. It suffices to prove the observation in the case that each $\ell_{i,j}$ has a non-zero constant term c_j (it is easy to reduce to this case). In this case, we can write

$$T_i = \left(\prod_{j=1}^{s} c_i \right) \cdot \prod_{j=1}^{s} (1 + \ell'_{i,j})$$

where each $\ell'_{i,j}$ is a homogeneous linear polynomial. It then follows that the degree-d homogeneous part $T_{i,d}$ of T_i can be written as a linear projection applied to the *Elementary Symmetric Polynomial* E_s^d of degree d in s variables. More precisely, we have

$$T_{i,d} = \left(\prod_{i=1}^{s} c_i \right) \cdot E_s^d(\ell'_{i,1}, \ldots, \ell'_{i,s}).$$

Shpilka and Wigderson [51, Theorem 5.3] proved that, over fields of characteristic 0 the polynomial E_s^d has a homogeneous[6] $\Sigma\Pi\Sigma\Pi$ circuit of size $\mathrm{poly}(s) \cdot \exp(O(\sqrt{d}))$. Using this with the above expression, we get the following result.

Lemma 9 ([18] Lemma 5.6 in the journal version)**.** *Let s, N be growing parameters. Assume that* $\mathrm{char}(\mathbb{F}) = 0$ *or* $\mathrm{char}(\mathbb{F}) > d$. *Fix any $\Sigma\Pi\Sigma$ circuit F of size at most s computing a homogeneous polynomial $P(x_1, \ldots, x_N)$ of degree d. Then, P can also be computed by a* homogeneous $\Sigma\Pi\Sigma\Pi\Sigma$ *circuit F' of size at most* $\mathrm{poly}(s) \cdot \exp(O(\sqrt{d}))$.

We show a generalization of the above lemma for larger depths, with a worse dependence on the degree parameter d. The next lemma is proved in a similar way by using a generalization of Newton identities, known as the *Binet-Minc identity.*

Lemma 10. *Let s, N, d, Δ be growing parameters with $s \geqslant N$. Assume that* $\mathrm{char}(\mathbb{F}) = 0$ *or* $\mathrm{char}(\mathbb{F}) > d$. *If C is a circuit of size at most s and product-depth at most Δ computing a homogeneous polynomial $P(x_1, \ldots, x_N)$ of degree d, then, P can also be computed by a* homogeneous *circuit $\tilde{C}$ of size at most* $\mathrm{poly}(s) \cdot d^{O(d)}$ *and product-depth at most* 2Δ.

We will prove Lemma 10 in Section 7.

[6] In fact they claim the result for general depth-4 circuits, but it was already noticed in [18] that the formula they get with this approach is homogeneous. In fact in [18], they also show that the product gates can be replaced by exponentiation gates, but we do not need it here.

Homogeneous to set-multilinear circuits We also want to convert homogeneous circuits into set-multilinear ones without increasing the product-depth and with a relatively small size blow-up.

Lemma 11. *Let s, N, d, Δ be growing parameters with $s \geqslant Nd$. If C is a homogeneous circuit of size at most s and product-depth at most Δ computing a set-multilinear polynomial P over the sets of variables $(X_1, \ldots, X_d)$ (with $|X_i| \leqslant N$), then there is a set-multilinear circuit $\tilde{C}$ of size at most $(d!)s$ and product-depth at most Δ computing P.*

Proof. Let us describe our new circuit $\tilde{C}$. For any gate α of degree d_α from C, we create $\binom{d}{d_\alpha}$ gates α_S in $\tilde{C}$ (we index these gates by the subsets $S \subseteq [d]$ such that $|S| = d_\alpha$). Now we want to link these gates such that for every gate α in C and any $S \subseteq [d]$ with $|S| = d_\alpha$, the product-depth of α_S is the same than the one of α and the polynomial computed by α_S is the projection of the polynomial computed by α to the set-multilinear part associated to S:

$$\alpha_S = \sum_{m \text{ set-multilinear over } (X_i)_{i \in S}} ([m]\alpha)\ m$$

where $([m]\alpha)$ is the coefficient of the monomial m in α.

Let us do it by induction on the structure of the graph.

If α is a leaf, it is labelled either by a constant or by a variable. When $d_\alpha = 0$, there is nothing to change. Otherwise $d_\alpha = 1$. In C the leaf α is labelled by a variable x which belongs to an X_i. We just need to label the gates by $\alpha_{\{i\}} = x$ and $\alpha_{\{j\}} = 0$ for $j \neq i$.

If $\alpha = c^1\alpha^1 + \ldots + c^p\alpha^p$ is a sum gate (where the c^i are constants in $\mathbb{F}$), we just need to compute the linear combination part by part. For any $S \subseteq [d]$ with $|S| = d_\alpha$:

$$\alpha_S = c^1\alpha^1_S + \ldots + c^p\alpha^p_S.$$

Finally if $\alpha = \alpha^1 \cdot \ldots \cdot \alpha^p$ is a product gate, we need to extract all the decompositions. Let $S \subseteq [d]$ with $|S| = d_\alpha$:

$$\alpha_S = \sum_{\substack{(S_1,\ldots,S_p) \text{ partition of } S \\ \text{with } \forall i, |S_i| = d_{\alpha^i}}} \alpha^1_{S_1} \cdot \ldots \cdot \alpha^p_{S_p}.$$

The size of the sum is $\binom{d_\alpha}{d_{\alpha^1},\ldots,d_{\alpha^p}}$.

Hence each leaf and sum gate α in C creates $\binom{d}{d_\alpha} \leqslant d!$ new gates in $\tilde{C}$. Each multiplication gate α in C creates $\binom{d}{d_\alpha} \leqslant d!$ sum gates and $\binom{d}{d_\alpha}\binom{d_\alpha}{d_{\alpha^1},\ldots,d_{\alpha^p}} \leqslant d!$ new product gates. So the number of gates of $\tilde{C}$ is bounded by $2d!$ times the number of gates of C. Notice that we can avoid the factor 2 since we do not need to keep the sum gates which come from a product gate, we can inject them into the sum gates of the next layer of the circuit. Furthermore, the product depth of the gate α_S in $\tilde{C}$ is the same than the one of the gate α in C. □

4 Lower bounds for depth-three circuits

We prove in this section the case $\Delta = 2$ of Theorem 2 and Corollary 3 and the case $\Delta = 1$ of Corollary 4. By Proposition 8 and Lemma 11, it is sufficient to get a sufficiently large lower bound for set-multilinear depth-5 circuits.

Lemma 12. *Let $n, d \in \mathbb{N}\backslash\{0\}$ with $n \geqslant 16^{\sqrt{d}+1}$. Any set-multilinear circuit C of product-depth 2 computing $\mathrm{IMM}_{n,d}$ has size at least $n^{\Omega(\sqrt{d})}$.*

Proof of the case $\Delta = 2$ of Theorem 2 and Corollary 3 and $\Delta = 1$ of Corollary 4. For Theorem 2, the result directly follows Lemma 12. In the case of Corollary 4 (resp. Corollary 3) using Proposition 8 (resp. Lemma 11), we can transform the circuit C into a depth-5 set-multilinear one of size at most $d^{O(d)}\mathrm{poly}(s)$. By Lemma 12, it implies that $d^{O(d)}\mathrm{poly}(s) \geqslant n^{\Omega(\sqrt{d})}$. By the assumption $d \leqslant (\log n)/100$, we get the desired lower bound for s. □

Proof of Lemma 12. Recall that any circuit of constant depth can be converted to a formula with only polynomial blow-up. So it suffices to show the following.

Claim 13. *Let $d \geqslant 16$ and $k > 2\sqrt{d}$ be an integer. Let w be any word of length d on the alphabet $\{-k, \lfloor k - k/\sqrt{d} \rfloor\}$. Then any set-multilinear formula C of product depth 2 and of size s satisfies*

$$\mathrm{relrk}_w(C) \leqslant s \cdot 2^{-\frac{k\sqrt{d}}{8}}.$$

So, by fixing $k = \lfloor \frac{\log_2 n}{2} \rfloor$, we have $k > 2\sqrt{d}$. We can construct by induction a word w on the alphabet $\{-k, \lfloor k - k/\sqrt{d} \rfloor\}$ which is $2k$-unbiased. Indeed, if $|w_{[i]}| \leqslant 0$, we choose $w_{i+1} = \lfloor k - k/\sqrt{d} \rfloor$, otherwise we set $w_{i+1} = -k$. By Lemma 7 and Claim 13, we get the lower bound:

$$s \geqslant 2^{\frac{k\sqrt{d}}{8}} 2^{-k} \geqslant 2^{(\frac{\log_2 n}{2} - 1)\frac{\sqrt{d}}{8} - \frac{\log_2 n}{2}} \geqslant n^{\frac{\sqrt{d}}{16}} 2^{-\frac{\log_2 n}{32} - \frac{\log_2 n}{2}} \geqslant n^{\frac{\sqrt{d}}{16} - \frac{17}{32}}.$$

for the polynomial $\mathrm{IMM}_{2^{2k},d}$ against set-multilinear circuits of product-depth 2.

Proof of Claim 13. We know C is a product-depth 2 formula, so we can define $C = C_1 + \ldots + C_t$ where each C_i is of the form $\prod\sum\prod\sum$ and has size s_i. We say that C_i is of type 1 if some factor of C_i has degree $\geqslant \sqrt{d}/2$, otherwise it is of type 2.

- If C_i is of type 1, then $C_i = C_{i,1} \cdot \ldots \cdot C_{i,t_i}$. Upto reordering, we can assume that $C_{i,1}$ is a sum of products of linear forms of degree at least $\sqrt{d}/2$. Notice that if L is a linear form on variables $X(w_i)$, we have $\mathrm{relrk}(L) \leqslant 2^{-|w_i|/2} \leqslant 2^{-(k-k/\sqrt{d}-1)/2}$. In particular, by the multiplicativity and sub-additivity of relrk_w (Claim 6),

$$\mathrm{relrk}_w(C_i) \leqslant \mathrm{relrk}_w(C_{i,1}) \leqslant s_i 2^{-\frac{k\sqrt{d}-k-\sqrt{d}}{2\sqrt{d}}\deg(C_{i,1})} \leqslant s_i 2^{-\frac{k\sqrt{d}-k-\sqrt{d}}{4}} \leqslant s_i 2^{-\frac{k\sqrt{d}}{8}}.$$

- If C_i is of type 2, then $C_i = C_{i,1} \cdot \ldots \cdot C_{i,t_i}$ where each factor C_{ij} has degree $< \sqrt{d}/2$. Each C_{ij} is a set-multilinear formula over a subset $(X(w_p) : p \in S_j)$ for some $S_j \subseteq [d]$, where $S_1, \ldots, S_{t_i}$ partition $[d]$. Let $w^{i1}, \ldots, w^{it_i}$ be the corresponding decomposition of w. That is, $w^{ij} = w_{|S_j}$. Recall that for a word w^{ij} we defined in the preliminaries $w^{ij}_{S_j}$ as the sum of its entries.

  Let $j \in [t_i]$. Let a_{ij} be the number of positive indices in w^{ij}. If $2a_{ij} \leqslant \deg(C_{i,j})$, then

$$w^{ij}_{S_j} \leqslant \frac{\deg(C_{i,j})}{2} \times (-k) + \frac{\deg(C_{i,j})}{2} \times (k - \frac{k}{\sqrt{d}}) = -\frac{k}{2\sqrt{d}}\deg(C_{i,j}).$$

Otherwise, we have

$$\begin{aligned}
\left|w_{S_j}^{ij}\right| &= \left|a_{ij}\left\lfloor k - \frac{k}{\sqrt{d}}\right\rfloor - (\deg(C_{i,j}) - a_{ij})k\right| \\
&= \left|a_{ij}k - a_{ij}\left\lceil \frac{k}{\sqrt{d}}\right\rceil - \deg(C_{i,j})k + a_{ij}k\right| \\
&> \left(2a_{ij} - \deg(C_{i,j}) - \frac{a_{ij}}{\sqrt{d}}\right)k - a_{ij} \\
&> \frac{k}{2} - \frac{k}{4} = \frac{k}{2}\cdot\frac{1}{2} && \text{as } 2a_{ij} - \deg(C_{i,j}) \geqslant 1 \text{ and } a_{ij} < \sqrt{d}/2 \\
&> \frac{k\deg(C_{i,j})}{2\sqrt{d}} && \text{as } \deg(C_{i,j}) < \sqrt{d}/2.
\end{aligned}$$

So in both cases, $\left|w_{S_j}^{ij}\right| \geqslant \frac{k\deg(C_{i,j})}{2\sqrt{d}}$.

In particular,

$$\mathrm{relrk}_w(C_i) \leqslant \prod_{j=1}^{t_i} 2^{-\frac{1}{2}|w_{S_j}^{ij}|} \leqslant 2^{-\frac{1}{2}\sum\frac{k\deg(C_{i,j})}{2\sqrt{d}}} = 2^{-\frac{kd}{4\sqrt{d}}} \leqslant s_i 2^{-\frac{k\sqrt{d}}{8}}.$$

The result of the claim directly follows from the subadditivity of the measure. □

□

5 Lower bounds for small-depth circuits

We prove in this section the general case of Theorem 2 and Corollaries 3 and 4. By Proposition 8, it is sufficient to get a sufficiently large lower bound for set-multilinear circuits of small depth.

Lemma 14. *Let $n, d, \Delta \in \mathbb{N}\backslash\{0\}$ with $n \geqslant 4^{10d+1}$. Any set-multilinear circuit C of product-depth Δ computing $\mathrm{IMM}_{n,d}$ has size at least*

$$n^{\Omega\left(\frac{d^{1/(2^\Delta-1)}}{\Delta}\right)}.$$

Proof of Theorem 2, Corollary 3 and Corollary 4. By Proposition 8 or Lemma 11, we can transform the circuit C into a depth-$\tilde{\Delta}$ set-multilinear one of size at most $d^{O(d)}\mathrm{poly}(s)$. Moreover the product-depth is unaffected, $\tilde{\Delta} = \Delta$ during this transformation in the case of Theorem 2 and Corollary 3 and it is multiplied by 2: $\tilde{\Delta} = 2\Delta$ in the case of Corollary 4. By Lemma 14, it implies that $d^{O(d)}\mathrm{poly}(s) \geqslant n^{\Omega\left(d^{1/(2^{\tilde{\Delta}}-1)}/\tilde{\Delta}\right)}$. If $\tilde{\Delta} \geqslant \frac{1}{2}\log_2\log_2 d$, then $n^{d^{\exp(-O(d))}} = n^{(1/\log d)^{\Omega(1)}} < n$ and so the results are trivial. Otherwise, $d^{1/(2^{\tilde{\Delta}}-1)} > \log d$, and by the assumption $d \leqslant (\log n)/100$, we get that $n^{d^{2^{-\tilde{\Delta}}}/\tilde{\Delta}} \geqslant n^{2^{\sqrt{\log d}}/\log\log d} \geqslant d^{\omega(d)}$. It implies the desired lower bound for s. □

Proof of Lemma 14. Let us assume first the following claim:

Claim 15. *Let $k \geqslant 10d$. Let w be any word of length d such the entries of w are $\lfloor \alpha k\rfloor$ and $-k$ where $\alpha = 1/\sqrt{2}$. Then for any $\Delta \geqslant 1$, any set-multilinear formula C of product depth Δ of size at most s satisfies*

$$\mathrm{relrk}_w(C) \leqslant s \cdot 2^{-\frac{kd^{1/(2^\Delta-1)}}{20}}.$$

By fixing $k = \lfloor \frac{\log_2 n}{2} \rfloor$, we have $k \geqslant 10d$. As in the proof of Lemma 12, we can fix a word w of length d over the alphabet $\{\lfloor \alpha k \rfloor, -k\}$ such that w is $2k$-unbiased. By Lemma 7, $\mathrm{relrk}_w(P_w) \geqslant 2^{-k}$ for suitable set-multilinear polynomial P_w of degree d which is a set-multilinear projection of $\mathrm{IMM}_{2^{2k},d}$. If C is a set-multilinear circuit of size s and product-depth Δ computing $\mathrm{IMM}_{n,d}$, then by expanding it, we can transform it to a set-multilinear formula of size at most $s^{2\Delta}$ for the same polynomial. By Lemma 7 and Claim 15, we get the lower bound

$$s^{2\Delta} \geqslant 2^{-k} 2^{\frac{kd^{1/(2^\Delta-1)}}{20}} \geqslant \left(\frac{\sqrt{n}}{2}\right)^{\frac{d^{1/(2^\Delta-1)}}{20}} n^{-\frac{1}{2}}.$$

Proof of Claim 15. We do the proof by induction on Δ.

If $\Delta = 1$, then $C = C_1 + \ldots + C_t$ where each C_i is a product of linear forms. So for all i,

$$\mathrm{relrk}_w(C_i) = \prod_{j=1}^{d} 2^{-\frac{1}{2}|w_j|} \leqslant 2^{-\frac{kd}{4}}.$$

By subadditivity of relrk_w,

$$\mathrm{relrk}_w(C) \leqslant s2^{-\frac{kd}{4}} \leqslant s2^{-\frac{kd}{20}}.$$

Assume the claim is proved for all formulas of product-depth $\leqslant \Delta$. Let C be a formula of product-depth $(\Delta + 1)$.

Let $C = C_1 + \ldots + C_t$. Each C_i of size s_i is said to be of type 1 if one of its factors has degree at least $T_\Delta = d^{(2^\Delta-1)/(2^{\Delta+1}-1)}$, otherwise it is of type 2.

- If C_i is of type 1, then $C_i = C_{i,1} \cdot \ldots \cdot C_{i,t_i}$. Upto reordering, we can assume that $C_{i,1}$ is a product-depth-Δ formula of degree at least T_Δ. Assume it is of size $s_{i,1}$. By induction,

$$\mathrm{relrk}_w(C_i) \leqslant \mathrm{relrk}_w(C_{i,1}) \leqslant s_{i,1} 2^{-\frac{kT_\Delta^{1/(2^\Delta-1)}}{20}} \leqslant s_i 2^{-\frac{kd^{1/(2^{\Delta+1}-1)}}{20}}.$$

- If C_i is of type 2, then $C_i = C_{i,1} \cdot \ldots \cdot C_{i,t_i}$ where each factor C_{ij} has degree $< T_\Delta$. In particular $t_i > \frac{d}{T_\Delta}$. As the circuit is set-multilinear, $(S_1, \ldots, S_{t_i})$ form a partition of S where each $C_{i,j}$ is set-multilinear with respect to $(X_l)_{l \in S_j}$ and C_i is set-multilinear with respect to $(X_l)_{l \in S}$. Let $w^{i1}, \ldots, w^{it_i}$ be the corresponding decomposition.

  Let $j \in [t_i]$. Let a_{ij} be the number of positive indices in w^{ij}. We have

$$\begin{aligned} \left| w^{ij}_{S_j} \right| &= |a_{ij}\lfloor \alpha k \rfloor - (\deg(C_{i,j}) - a_{ij})k| \\ &\geqslant |a_{ij}\alpha k - (\deg(C_{i,j}) - a_{ij})k| - |a_{ij}\alpha k - a_{ij}\lfloor \alpha k \rfloor| \\ &\geqslant |a_{ij}\alpha - (\deg(C_{i,j}) - a_{ij})|\, k - a_{ij} \end{aligned}$$

  We use here a result on diophantine approximation.

  Claim 16. *Let $a, b \in \mathbb{Z}$. Then*

$$|a\alpha - b| \geqslant \frac{1}{4|a\alpha| + 2}.$$

  Proof. If $|b| \geqslant |a\alpha| + 1$, then the result is immediate. Otherwise, we can notice that

$$|a\alpha - b| \cdot |a\alpha + b| = \left| \frac{a^2}{2} - b^2 \right| \geqslant \frac{1}{2}.$$

And so,

$$|a\alpha - b| \geqslant \frac{1}{2|a\alpha| + 2|b|} \geqslant \frac{1}{4|a\alpha| + 2}.$$

□

Now we can come back to the bound on $|w^{ij}_{S_j}|$:

$$\left|w^{ij}_{S_j}\right| \geqslant \frac{k}{4a_{ij}\alpha + 2} - a_{ij} \geqslant \frac{k}{5T_\Delta} - T_\Delta \geqslant \frac{k}{10T_\Delta}.$$

The last inequality follows from the fact that $k \geqslant 10d \geqslant 10T_\Delta^2$. So,

$$\mathrm{relrk}_w(C_i) = \prod_{j=1}^{t_i} \mathrm{relrk}_{w^{ij}}(C_{i,j}) \leqslant \prod_{j=1}^{t_i} 2^{-\frac{1}{2}|w^{ij}_{S_j}|} \leqslant 2^{-\frac{kt_i}{20T_\Delta}} \leqslant 2^{-\frac{kd}{20T_\Delta^2}} \leqslant 2^{-\frac{kd^{1/(2^{\Delta+1}-1)}}{20}}.$$

The final result directly follows from the subadditivity of the mesure. □

□

6 PIT for small-depth circuits

In this section we consider the Polynomial Identity Testing (PIT) question for small-depth circuits. We observe that the PIT for small-depth circuits can be solved in deterministic subexponential time. We derive this as a corollary of our lower bound from Section 5 and the following result of Chou, Kumar and Solomon [11].

Lemma 17 ([11] Theorem 2.3)**.** *Let $\Lambda \geqslant 6$ be an integer and $\varepsilon > 0$ be a real number and let M, m be any integer parameters such that $m = M^\varepsilon$. Let f be an explicit*[7] *multilinear polynomial on m variables of degree $d = O(\log^2 m/\log^2 \log m)$, which cannot be computed by circuits of depth*[8] *Λ and size $\mathrm{poly}(m)$. Then, there is a deterministic algorithm, which given as circuit C of size s, depth $\Lambda - 5$, and degree D on M variables, runs in time $(s \cdot M \cdot D)^{O(m^2)}$ and determines if the polynomial computed by C is identically zero or not.*

From the above statement along with Corollary 4, Corollary 5 easily follows:

Proof of Corollary 5. Let us define $m = n^\varepsilon$. We would like to apply Lemma 17 with $f = \mathrm{IMM}_{\nu,\delta}$ where $\delta = \frac{\log m}{\log \log m}$ and $\nu = \sqrt{\frac{m}{\delta}}$. In particular, $\mathrm{IMM}_{\nu,\delta}$ is m-variate. Moreover, as $\frac{\log \nu}{100} \geqslant \omega(\delta)$, and as

$$\delta^{\exp(-O(\Delta))} \geqslant 2^{\frac{\log \delta}{(\log \log m)^{o(1)}}} \geqslant \omega(1),$$

Corollary 4 implies that $\mathrm{IMM}_{\nu,\delta}$ does not have circuits of depth $\Delta + 5$ and size $\nu^{O(1)} \geqslant m^{O(1)}$. So Lemma 17 directly implies a deterministic PIT algorithm with running time $(snd)^{O(n^{2\varepsilon})}$ against algebraic circuits of size s, depth Δ, degree d, and with n variables.

As a circuit of depth Δ and size s computes a polynomial of degree at most s^Δ, the claimed upper bound on the running time follows. □

[7] Here, explicit means that the polynomial can be evaluated at a given point in polynomial time.

[8] Here the parameter depth refers to the exact depth of the circuit and not the product-depth. I.e. if the circuit has product depth Δ then it has depth $\Lambda = 2\Delta + 1$.

7 Proof of the homogenization transformation

We give below a stronger statement of Lemma 10 that is more amenable to induction.

Lemma 18. *Let s, N, d, Δ be growing parameters with $s \geqslant N$. Assume that* $\mathrm{char}(\mathbb{F}) = 0$ *or* $\mathrm{char}(\mathbb{F}) > d$. *Fix any circuit C of size at most s and product-depth at most Δ. Assume C has m output gates which compute polynomials $P_1, \ldots, P_m$. There is a* homogeneous *circuit $\tilde{C}$ with $m\cdot(d+1)$ output gates that compute polynomials $P_j^{(i)}$ ($j \in [m], i \in \{0, \ldots, d\}$) where $P_j^{(i)}$ denotes the degree-i homogeneous component of P_j. Further, the size of $\tilde{C}$ is at most* $\mathrm{poly}(s)\cdot d^{O(d)}$ *and its product-depth is at most 2Δ.*

The proof of Lemma 9 (case $\Delta = 1$) is based on the construction of a homogeneous $\Sigma\Pi\Sigma\Pi$ formula for the Elementary Symmetric Polynomial of degree d. This construction, due to Shpilka and Wigderson [51], depends on the classical Newton identities relating different families of symmetric polynomials with each other. The lemma above is proved by using a generalization of these identities, known as the *Binet-Minc identity.*

To state Lemma 19, we will need the notion of the *weighted degree* of a polynomial. Assume that we are working over $\mathbb{F}[x_1, \ldots, x_N]$ and we have a 'weight function' $\varphi : \{x_1, \ldots, x_N\} \to [d]$ which assigns to each variable x_i an integer weight in $[d]$. The weighted degree of a monomial $\prod_{i=1}^N x_i^{e_i}$ w.r.t. φ is defined in the natural way to be $\sum_{i=1}^N e_i\varphi(x_i)$. The weighted degree of a polynomial P, the weighted degree-d part of P etc. are defined analogously. A formula in the variables $x_1, \ldots, x_N$ is weighted-degree homogeneous if each nodes in the formula computes a homogeneous weighted polynomial (of some degree).

We need the following technical lemma about 'extracting' the component of a fixed weighted degree from a $\Pi\Sigma$ expression.

Lemma 19. *Assume that* $\mathrm{char}(\mathbb{F}) = 0$ *or* $\mathrm{char}(\mathbb{F}) > d$. *Let s, d be growing parameters. Let $Y = (y_{i,j})_{i\in[d],j\in[s]}$ be a matrix of variables and with weight function $\varphi : \{y_{i,j} \mid i \in [d], j \in [s]\} \to [d]$, such that $\varphi(y_{i,j}) = i$ for each i, j. Assume $T = \prod_{j=1}^s (c_j + y_{1,j} + \cdots + y_{d,j})$. Then, the homogeneous weighted degree-d part $T^{(d)}$ of the polynomial T can be computed by a weighted-degree homogeneous $\Sigma\Pi\Sigma\Pi$ formula of size at most* $\mathrm{poly}(s)\cdot d^{O(d)}$.

We first show how to use the above lemma to prove Lemma 18.

Proof of Lemma 18. We prove the lemma by induction on Δ. We will aim for a size bound of $(s+1)^c \cdot (d+1)^{cd}$ for a large enough constant $c > 0$.

The case $\Delta = 0$ is trivial as the polynomials $P_1, \ldots, P_m$ are just linear polynomials.

Now consider $\Delta > 0$. Let C be a circuit of product depth Δ and size at most s. We first apply the induction hypothesis to the subcircuit D of C containing all the gates of product-depth at most $\Delta - 1$. We consider all the $t \leqslant s$ gates $g_1, \ldots, g_t$ of D to be output gates. Applying the induction hypothesis to D yields a circuit $\tilde{D}$ of size at most $s_1 = s^c(d+1)^{cd}$ with $t\cdot(d+1)$ output gates $\tilde{g}_{j,i}$ ($j \in [t], i \in \{0, \ldots, d\}$) and of product-depth $2\Delta - 2$.

Let the output gates of C be $h_1, \ldots, h_m$ computing polynomials $P_1, \ldots, P_m$. Assume that the subcircuits corresponding to $h_1, \ldots, h_r$ have product-depth Δ and $h_{r+1}, \ldots, h_m$ have product-depth less than Δ. Without loss of generality, we assume $h_{r+1}, \ldots, h_m$ are $g_1, \ldots, g_{m-r}$.

Fix any $u \in [r]$. We have

$$P_u = \sum_{j=0}^{s_u} \alpha_{u,j} \prod_{k=1}^{t_{u,j}} P_{u,j,k} \tag{1}$$

where $\alpha_{u,j} \in \mathbb{F}$, $s_u, t_{u,j} \leqslant s$ and each $P_{u,j,k}$ is computed by a gate of product-depth less than Δ in C.

Note that for any $i \in [d]$, the degree-i component $P_u^{(i)}$ equals the degree-i component of

$$\sum_{j=0}^{s_u} \alpha_{u,j} \underbrace{\prod_{k=1}^{t_{u,j}} \sum_{\ell=0}^{i} P_{u,j,k}^{(\ell)}}_{P_{u,j}^{(i)}}. \qquad (2)$$

This is because $P_{u,j,k}$ and $\sum_{\ell=0}^{i} P_{u,j,k}^{(\ell)}$ differ only on components of degree greater than i.

Consider the polynomial $P_{u,j}^{(i)}$ on the right hand side of (2). We note that Lemma 19 can be used to 'extract' the homogeneous degree-i component of $P_{u,j}^{(i)}$ using a homogeneous circuit. Putting these circuits will yield the desired circuit $\tilde{C}$.

More precisely, fix $j \in [s_u]$ and define the polynomial $T_{u,i,j} = \prod_{k=1}^{t_{u,j}} (c_k + y_{1,k}^{(u,j)} + \cdots + y_{i,k}^{(u,j)})$ where $c_k \in \mathbb{F}$ is the constant term $P_{u,j,k}^{(0)}$. We define a weight function $\varphi_{i,j} : \{y_{\ell,k}^{(u,j)} \mid \ell \in [i], k \in [t_{u,j}]\} \to [i]$ where each $y_{\ell,k}^{(u,j)}$ has weight ℓ. By Lemma 19, for any $i \in [d]$, the weighted degree-i component of $T_{u,i,j}$ has a weighted homogeneous $\Sigma\Pi\Sigma\Pi$ formula $F_{u,j}^{(i)}$ of size $(s+1)^{c'}(d+1)^{c'd}$ for a large enough constant $c' \geqslant 1$. Let $F_u^{(i)}$ denote the formula which computes the linear combination $\sum_{j=0}^{s_u} \alpha_{u,j} F_{u,j}^{(i)}$. Let $F_u^{(0)}$ be a leaf computing the constant term of P_u.

To construct $\tilde{C}$, we start with the circuit $\tilde{D}$ and add the formulas $F_u^{(i)}$ $(u \in [r], i \in \{0, \ldots, d\})$ with the inputs rewired so that $y_{\ell,k}^{(u,j)}$ is replaced by the gate computing $P_{u,j,k}^{(\ell)}$ in $\tilde{D}$. The output gates of $\tilde{C}$ are the output gates of these new formulas along with the gates $\tilde{g}_{j,i}$ $(j \in [m-r], i \in \{0, \ldots, d\})$ which compute the homogeneous components of $P_{r+1}, \ldots, P_m$.

The size of the circuit $\tilde{C}$ can be bounded by

$$s_1 + (s+1)^{c'+2}(d+1)^{2c'd} = s^c(d+1)^{cd} + (s+1)^{c'+2}(d+1)^{2c'd} \leqslant (s+1)^c(d+1)^{cd}$$

for a large enough choice of constant c. To get $\tilde{C}$, we increase the product-depth of $\tilde{D}$ of at most two. So $\tilde{C}$ has product-depth at most 2Δ. □

It remains to prove Lemma 19. We start with the Binet-Minc identity which will play a crucial role in the proof.

Lemma 20 (Binet-Minc identity [5], Theorem 1.2)**.** *Let $Z = (z_{i,j})_{k \times s}$ be a matrix of indeterminates with $k \leqslant s$. Define the rectangular permanent*

$$\mathrm{rPer}_{k,s}(Z) = \sum_{J \subseteq [s] : |J| = k} \; \sum_{\substack{\sigma : [k] \to J \\ \text{a bijection}}} \prod_{i=1}^{k} z_{i,\sigma(i)}.$$

Let $\mathcal{P}_k$ denote the set of all (unordered) partitions of $[k]$ into non-empty subsets. Then, we have

$$\mathrm{rPer}_{k,s}(Z) = \sum_{\mathcal{I} \in \mathcal{P}_k} (-1)^{k-|\mathcal{I}|} \prod_{I \in \mathcal{I}} (|I| - 1)! \cdot P_I$$

where P_I denotes the polynomial $\sum_{j=1}^{s} \prod_{i \in I} z_{i,j}$.

The reader familiar with the Newton Identities will also note that they can be recovered from the Binet-Minc identity by identifying all the variables in the same column, for each column $j \in [s]$.

The Binet-Minc identity has the following consequence, which will be useful for us.

Corollary 21. *Fix any k, s with $k \leqslant s$. The polynomial $\mathrm{rPer}_{k,s}(Z)$ has a homogeneous $\Sigma\Pi\Sigma\Pi$ formula of size at most $k^{O(k)} \cdot s$. Further, the formula is also weighted-degree homogeneous w.r.t. any weight function φ that assigns the same weight to all the variables in each row.*

With the above corollary in hand, we can prove Lemma 19.

Proof of Lemma 19. We start by proving the lemma in the special case that $c_j = 1$ for each $j \in [s]$. Finally, we will show how to reduce to this case.

Let WE_s^d be the sum of all multilinear monomials over Y that contain at most one variable per column and have weighted degree exactly d. As $c_j = 1$ for each $j \in [s]$, we observe that

$$T^{(d)} = WE_s^d(\ell'_{i,j} : i \in [d], j \in [s]). \tag{3}$$

From the above observation, it follows that it suffices to show that WE_s^d has a $\Sigma\Pi\Sigma\Pi$ weighted-degree homogeneous formula of the required size.

Given a monomial m of WE_s^d, we define the *type* of m to be the tuple $\tau = (t_1, \ldots, t_d)$ where t_i denotes the number of variables of weight exactly i dividing m. Note that if a monomial m of WE_s^d has type $\tau = (t_1, \ldots, t_d)$, then $\sum_{i=1}^d it_i = d$ and $\sum_i t_i \leqslant s$. We call such a τ a *valid* type.

It follows that we have the formula

$$WE_s^d(Y) = \sum_{\tau=(t_1,\ldots,t_d) \text{ valid}} WE_{s,\tau}^d \tag{4}$$

where $WE_{s,\tau}^d$ is defined to be the sum of exactly those monomials of WE_s^d of type τ. We show that each $WE_{s,\tau}^d$ has a small weighted-degree homogeneous formula of the required kind.

Fix a valid $\tau = (t_1, \ldots, t_d)$. We note that a formula for this polynomial follows easily from the formula for the rectangular permanent above. More precisely, let $k = \sum_i t_i$ and let Z be the $k \times s$ matrix where the first t_1 rows are copies of the first row of Y, the next t_2 rows are copies of the second row of Y, and so on. Consider the polynomial $\mathrm{rPer}_{k,s}(Z)$. We note that each monomial of the polynomial contains exactly t_1 variables from the first row of Y, t_2 variables from the second row of Y etc., and further the variables all come from distinct columns of Y. In other words, the monomials of this polynomial are exactly the monomials of $WE_{s,\tau}^d$. Further, by symmetry, each such monomial appears exactly $\prod_{i=1}^d (t_i!)$ times. Hence, we have

$$WE_{s,\tau}^d = \frac{1}{\prod_{i=1}^d (t_i!)} \mathrm{rPer}_{k,s}(Z).$$

Using Corollary 21, we get a weighted-degree homogeneous $\Sigma\Pi\Sigma\Pi$ formula for $WE_{s,\tau}^d$ of size $\mathrm{poly}(s) \cdot k^{O(k)} = \mathrm{poly}(s) \cdot d^{O(d)}$. Plugging this into (4), we get a weighted-degree homogeneous $\Sigma\Pi\Sigma\Pi$ formula for WE_s^d of size $\mathrm{poly}(s) \cdot k^{O(k)} = \mathrm{poly}(s) \cdot d^{O(d)}$. The statement of the lemma now follows in the case that each constant term c_j is 1.

It remains to prove the lemma for the case when the constant terms c_j are arbitrary. To see this, note that if there are more than d many j such that $c_j = 0$, then the weighted-degree d component $T^{(d)}$ of T is the zero polynomial. Hence, we assume that there are $t \leqslant d$ many j such that $c_j = 0$. Without loss of generality, say that these are $c_1, \ldots, c_t$. Hence, we have

$$T = \sum_{i_1,\ldots,i_t \in [d]} y_{i_1,1} \cdots y_{i_t,t} \cdot \prod_{j>t} (c_j + \sum_{i=1}^d y_{i,j}) = \sum_{i_1,\ldots,i_t \in [d]} y_{i_1,1} \cdots y_{i_t,t} \cdot \prod_{j>t} c_j \cdot \underbrace{\prod_{j>t} (1 + \sum_{i=1}^d \frac{y_{i,j}}{c_j})}_{T_{i_1,\ldots,i_t}}. \tag{5}$$

It follows from what we proved above that for each $(i_1, \ldots, i_t)$, the weighted-degree $d' := (d - \sum_{p \in [t]} i_p)$ component of $T_{i_1,\ldots,i_t}$ has weighted-degree homogeneous $\Sigma\Pi\Sigma\Pi$ formula of size $\mathrm{poly}(s) \cdot (d')^{O(d')} = \mathrm{poly}(s) \cdot d^{O(d)}$. Note that for $\ell \leqslant t$, each $y_{i_\ell,\ell}$ is trivially a weighted-degree homogeneous $\Sigma\Pi$ formula of size $O(1)$. Using distributivity and the fact that each $y_{i_\ell,\ell}$ is weighted-degree homogeneous, we get a similar expression for each summand on the right hand side of (5). As the sum has size at most d^d, we get a weighted-degree homogeneous $\Sigma\Pi\Sigma\Pi$ formula of size $\mathrm{poly}(s) \cdot d^{O(d)}$ for $T^{(d)}$. This proves the lemma. □

8 Proof of Lemma 7

We start by noting that for every w which does not have too much bias, there is a polynomial $P_w \in \mathbb{F}_{\mathrm{sm}}[\overline{X}(w)]$ that has large rank w.r.t. w and has a small set-multilinear *Algebraic Branching Program* (ABP). We start by recalling the definition of such an ABP.

A set-multilinear ABP over the variables in $\overline{X}(w)$ is a layered directed acyclic graph with $d+1$ layers labelled $0, \ldots, d$. The 0th and dth layer contain a single vertex each (they are the source and sink vertices of the DAG). All edges go from the $(i-1)$th layer to the ith layer for some $i \in [d]$, and each such edge is labelled by a homogeneous linear polynomial in the variables from $X(w_i)$. The polynomial computed by the ABP is defined to be the sum, over all source to sink paths ρ, of the products of the edge-labels seen along ρ. This is clearly a polynomial of the space $\mathbb{F}_{\mathrm{sm}}[\overline{X}(w)]$.

Lemma 22. *Let $w \in A^d$ be any word that is b-unbiased. Then, there is a set-multilinear ABP of width 2^b that computes a polynomial $P_w \in \mathbb{F}_{sm}[\overline{X}(w)]$ such that* $\mathrm{relrk}_w(P_w) \geqslant 2^{-b/2}$.

Proof Sketch. We start by recalling the description of the polynomial P_w. Say $\overline{X}(w) = (X_1, \ldots, X_d)$ and since each X_i has size $2^{|w_i|}$, we assume that the variables of X_i are labelled by strings in $\{0,1\}^{|w_i|}$.

Given any monomial $m \in \mathbb{F}_{\mathrm{sm}}[\overline{X}(w)]$, let m_+ denote the corresponding "positive" monomial from $\mathcal{M}_w^{\mathcal{P}}$ and m_- the corresponding "negative" monomial from $\mathcal{M}_w^{\mathcal{N}}$. As each variable of $\overline{X}(w)$ is labelled by a Boolean string and each monomial of $\mathcal{M}_w^{\mathcal{N}}$ and of $\mathcal{M}_w^{\mathcal{P}}$ is associated with a string of variables, we can associate any monomial m' with a Boolean string $\sigma(m')$. As w is b-unbiased, the difference of length of the strings $\sigma(m_+)$ and $\sigma(m_-)$ is at most b. We will write $\sigma(m_+) \sim \sigma(m_-)$ when the shorter one is a prefix of the other one.

The polynomial P_w is defined as follows

$$P_w = \sum_{m \in \mathbb{F}[\overline{X}(w)],\ \ \sigma(m_+) \sim \sigma(m_-)} m.$$

Clearly, the matrices $M_w(P_w)$ are full-rank. So, $\mathrm{relrk}_w(P_w) = 2^{-|w_{[d]}|/2} \geqslant 2^{-b/2}$.

We now show how to construct an ABP for P_w.

- At each layer $i \in \{0, \ldots, d\}$, the ABP has exactly $2^{|w_{[i]}|} \leqslant 2^b$ vertices. For the partial monomial m seen so far, the ABP is intuitively keeping track of either the last few bits of $\sigma(m_+)$ or the last few bits of $\sigma(m_-)$.

  For example, assume that $w_{[i]} \geqslant 0$. Then, for any monomial m in variable sets $X_1, \ldots, X_i$, the string $\sigma(m_+)$ has length exactly $w_{[i]}$ more than that of $\sigma(m_-)$. Assuming that $\sigma(m_+)$ agrees with $\sigma(m_-)$ on all but its last $w_{[i]}$ bits, i.e. $\sigma(m_+) = \sigma(m_-)\tau$ for $\tau \in \{0,1\}^{w_{[i]}}$, the vertex of the ABP keeps track of the strings τ.

More formally, for each $\tau \in \{0,1\}^{|w_{[i]}|}$, we have a vertex v_τ in the ith layer of the ABP, where the polynomial P_{v_τ} computed from the source node to v_τ is the sum over all monomials m over $X_1, \ldots, X_i$ such that $\sigma(m_+) = \sigma(m_-)\tau$ (resp. $\sigma(m_-)\tau = \sigma(m_+)$) if $w_{[i]} \geqslant 0$ (resp. $w_{[i]} < 0$).

- Given vertices u_τ on layer $i+1$, one can see that we have $P_{u_\tau} = \sum_{v_\rho} P_{v_\rho} \cdot L_\rho$ for a suitable linear polynomial L_ρ in $\mathbb{F}[X_{i+1}]$ where the sum runs over all vertices v_ρ in the ith layer. More precisely, we have

$$L_\rho = \begin{cases} 0 & \text{if } \mathrm{sgn}(w_{[i+1]}) = \mathrm{sgn}(w_{[i]}),\ |w_{[i+1]}| \geqslant |w_{[i]}|, \text{ and } \rho \text{ not a prefix of } \tau, \\ x_{\rho'} & \text{if } \mathrm{sgn}(w_{[i+1]}) = \mathrm{sgn}(w_{[i]}),\ |w_{[i+1]}| \geqslant |w_{[i]}|, \text{ and } \tau = \rho\rho', \\ 0 & \text{if } \mathrm{sgn}(w_{[i+1]}) = \mathrm{sgn}(w_{[i]}),\ |w_{[i+1]}| < |w_{[i]}|, \text{ and } \tau \text{ not a suffix of } \rho, \\ x_{\tau'} & \text{if } \mathrm{sgn}(w_{[i+1]}) = \mathrm{sgn}(w_{[i]}),\ |w_{[i+1]}| < |w_{[i]}|, \text{ and } \rho = \tau'\tau, \\ x_{\rho\tau} & \text{if } \mathrm{sgn}(w_{[i+1]}) \neq \mathrm{sgn}(w_{[i]}). \end{cases}$$

- Finally, identifying all the vertices on layer d gives us an ABP computing the polynomial P_w.

□

Proof of Lemma 7 now follows from the above lemma.

Proof of Lemma 7. By Lemma 22, we know that there is a width 2^b set-multilinear ABP computing a polynomial P_w such that $\mathrm{relrk}_w(P_w) \geqslant 2^{-b/2}$. It is a standard fact (and easy to see) that since the polynomial P_w is computed by a set-multilinear ABP of width at most 2^b, it is a *set-multilinear restriction* of $\mathrm{IMM}_{2^b,d} = \mathrm{IMM}_{n,d}$ in the following sense. There are maps $\rho_p : X_p \to X(w_p)$, such that upon applying these linear substitutions to all the variables in $\mathrm{IMM}_{n,d}$ yields the polynomial P_w.

By applying this linear substitution to the circuit computing $\mathrm{IMM}_{n,d}$, we directly get a circuit computing P_w. □

References

[1] Miklós Ajtai. Σ_1^1-formulae on finite structures. *Annals of Pure and Applied Logic*, 24(1):1 – 48, 1983.

[2] Paul Beame, Trinh Huynh, and Toniann Pitassi. Hardness amplification in proof complexity. In Leonard J. Schulman, editor, *Proceedings of the 42nd ACM Symposium on Theory of Computing, STOC 2010*, pages 87–96. ACM, 2010.

[3] Malte Beecken, Johannes Mittmann, and Nitin Saxena. Algebraic independence and blackbox identity testing. *Inf. Comput.*, 222:2–19, 2013.

[4] Suman K. Bera and Amit Chakrabarti. A depth-five lower bound for iterated matrix multiplication. In *Conference on Computational Complexity*, volume 33 of *LIPIcs*, pages 183–197. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.

[5] Jairo Bochi, Godofredo Iommi, and Mario Ponce. An ergodic theorem for permanents of oblong matrices. *arXiv*, 11 2014.

[6] Peter Bürgisser. Cook's versus valiant's hypothesis. *Theoretical Computer Science*, 235(1):71–88, 2000.

[7] Peter Bürgisser, Michael Clausen, and Mohammad Amin Shokrollahi. *Algebraic complexity theory*, volume 315 of *Grundlehren der mathematischen Wissenschaften*. Springer, 1997.

[8] Arkadev Chattopadhyay, Rajit Datta, and Partha Mukhopadhyay. Lower bounds for monotone arithmetic circuits via communication complexity. *Electron. Colloquium Comput. Complex. (To appear in CCC 2021)*, 27:166, 2020.

[9] Arkadev Chattopadhyay, Rajit Datta, and Partha Mukhopadhyay. Negations provide strongly exponential savings. *Electron. Colloquium Comput. Complex.*, 27:191, 2020.

[10] Suryajith Chillara, Mrinal Kumar, Ramprasad Saptharishi, and V. Vinay. The chasm at depth four, and tensor rank : Old results, new insights. *CoRR*, abs/1606.04200, 2016.

[11] Chi-Ning Chou, Mrinal Kumar, and Noam Solomon. Some closure results for polynomial factorization and applications. *CoRR*, abs/1803.05933, 2018.

[12] Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Springer Publishing Company, Incorporated, 2013.

[13] Zeev Dvir, Guillaume Malod, Sylvain Perifel, and Amir Yehudayoff. Separating multilinear branching programs and formulas. In *proceedings of Symposium on Theory of Computing (STOC)*, pages 615–624, 2012.

[14] Zeev Dvir, Amir Shpilka, and Amir Yehudayoff. Hardness-randomness tradeoffs for bounded depth arithmetic circuits. *SIAM J. Comput.*, 39(4):1279–1293, 2009.

[15] Hervé Fournier, Nutan Limaye, Guillaume Malod, and Srikanth Srinivasan. Lower bounds for depth-4 formulas computing iterated matrix multiplication. *SIAM Journal on Computing*, 44(5):1173–1201, 2015.

[16] Merrick Furst, James B. Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical systems theory*, 17(1):13–27, Dec 1984.

[17] Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Approaching the chasm at depth four. *J. ACM*, 61(6):33:1–33:16, December 2014.

[18] Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Arithmetic circuits: A chasm at depth 3. *SIAM Journal of Computing*, 45(3):1064–1079, 2016.

[19] Nikhil Gupta, Chandan Saha, and Bhargav Thankey. A super-quadratic lower bound for depth four arithmetic circuits. In *35th Computational Complexity Conference, CCC 2020, July 28-31, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 169 of *LIPIcs*, pages 23:1–23:31. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[20] J Hastad. Almost optimal lower bounds for small depth circuits. In *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing*, STOC '86, page 6–20, New York, NY, USA, 1986. Association for Computing Machinery.

[21] Pavel Hrubeš and Amir Yehudayoff. Monotone separations for constant degree polynomials. *Information Processing Letters*, 110(1):1–3, 2009.

[22] Mark Jerrum and Marc Snir. Some exact complexity results for straight-line computations over semirings. *Journal of the ACM*, 29(3):874–897, 1982.

[23] Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Comput. Complex.*, 13(1-2):1–46, 2004.

[24] Zohar Shay Karnin, Partha Mukhopadhyay, Amir Shpilka, and Ilya Volkovich. Deterministic identity testing of depth-4 multilinear circuits with bounded top fan-in. *SIAM J. Comput.*, 42(6):2114–2131, 2013.

[25] Neeraj Kayal. An exponential lower bound for the sum of powers of bounded degree polynomials. *Electron. Colloquium Comput. Complex.*, 19:81, 2012.

[26] Neeraj Kayal, Nutan Limaye, Chandan Saha, and Srikanth Srinivasan. Super-polynomial lower bounds for depth-4 homogeneous arithmetic formulas. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, USA*, pages 119–127. ACM, 2014.

[27] Neeraj Kayal, Nutan Limaye, Chandan Saha, and Srikanth Srinivasan. An exponential lower bound for homogeneous depth four arithmetic formulas. *SIAM Journal on Computing*, 46(1):307–335, 2017.

[28] Neeraj Kayal, Vineet Nair, and Chandan Saha. Separation between read-once oblivious algebraic branching programs (roabps) and multilinear depth-three circuits. *ACM Trans. Comput. Theory*, 12(1), February 2020.

[29] Neeraj Kayal, Chandan Saha, and Sébastien Tavenas. An Almost Cubic Lower Bound for Depth Three Arithmetic Circuits. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, volume 55 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 33:1–33:15, 2016.

[30] Neeraj Kayal, Chandan Saha, and Sébastien Tavenas. On the size of homogeneous and of depth-four formulas with low individual degree. *Theory Of Computing*, 14(16):1–46, 2018.

[31] Mrinal Kumar and Ramprasad Saptharishi. The computational power of depth five arithmetic circuits. *SIAM J. Comput.*, 48(1):144–180, 2019.

[32] Mrinal Kumar and Ramprasad Saptharishi. Hardness-randomness tradeoffs for algebraic computation. *Bull. EATCS*, 129, 2019.

[33] Mrinal Kumar and Shubhangi Saraf. On the power of homogeneous depth 4 arithmetic circuits. *SIAM Journal on Computing*, 46(1):336–387, 2017.

[34] Noam Nisan. Lower bounds for non-commutative computation. In *Proceedings of the twenty-third annual ACM symposium on Theory of computing*, pages 410–418, 1991.

[35] Noam Nisan and Avi Wigderson. Hardness vs randomness. *J. Comput. Syst. Sci.*, 49(2):149–167, 1994.

[36] Noam Nisan and Avi Wigderson. Lower bounds on arithmetic circuits via partial derivatives. *Computational Complexity*, 6(3):217–234, 1997.

[37] Shir Peleg and Amir Shpilka. A generalized sylvester-gallai type theorem for quadratic polynomials. In Shubhangi Saraf, editor, *35th Computational Complexity Conference, CCC 2020, July 28-31, 2020, Saarbrücken, Germany (Virtual Conference)*, volume 169 of *LIPIcs*, pages 8:1–8:33. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[38] Shir Peleg and Amir Shpilka. Polynomial time deterministic identity testingalgorithm for $\Sigma^{[3]}\Pi\Sigma\Pi^{[2]}$ circuits via edelstein-kelly type theorem for quadratic polynomials. *CoRR (to appear in STOC 2021)*, abs/2006.08263, 2020.

[39] Ran Raz. Multi-linear formulas for permanent and determinant are of super-polynomial size. *J. ACM*, 56(2):8:1–8:17, 2009.

[40] Ran Raz. Elusive functions and lower bounds for arithmetic circuits. *Theory Comput.*, 6(1):135–177, 2010.

[41] Ran Raz. Tensor-rank and lower bounds for arithmetic formulas. *Journal of the ACM*, 60(6):40:1–40:15, 2013.

[42] Ran Raz and Pierre McKenzie. Separation of the monotone NC hierarchy. *Comb.*, 19(3):403–435, 1999.

[43] Alexander A. Razborov. Lower bounds on the size of constant-depth networks over a complete basis with logical addition. *Mathematicheskie Zametki*, 41(2):598–607, 1986.

[44] Benjamin Rossman. Formulas versus circuits for small distance connectivity. *SIAM J. Comput.*, 47(5):1986–2028, 2018.

[45] Ramprasad Saptharishi. A survey of lower bounds in arithmetic circuit complexity. Github survey, 2015.

[46] Shubhangi Saraf and Ilya Volkovich. Black-box identity testing of depth-4 multilinear circuits. *Comb.*, 38(5):1205–1238, 2018.

[47] Nitin Saxena and Comandur Seshadhri. Blackbox identity testing for bounded top-fanin depth-3 circuits: The field doesn't matter. *SIAM J. Comput.*, 41(5):1285–1298, 2012.

[48] Eli Shamir and Marc Snir. Lower bounds on the number of multiplications and the number of additions in monotone computations. *IBM Thomas J. Watson Research Division*, 1977.

[49] Abhijat Sharma. An improved lower bound for depth four arithmetic circuits. *Master's thesis, Indian Institute of Science, Bangalore, India.*, 2017.

[50] Victor Shoup and Roman Smolensky. Lower bounds for polynomial evaluation and interpolation problems. *Computational Complexity*, 6(4):301–311, 1996.

[51] Amir Shpilka and Avi Wigderson. Depth-3 arithmetic circuits over fields of characteristic zero. *Computational Complexity*, 10(1):1–27, 2001.

[52] Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science*, 5:207–388, March 2010.

[53] Roman Smolensky. Algebraic methods in the theory of lower bounds for boolean circuit complexity. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, STOC '87, page 77–82, New York, NY, USA, 1987. Association for Computing Machinery.

[54] Volker Strassen. Vermeidung von divisionen. *Journal für die reine und angewandte Mathematik*, 264:184–202, 1973.

[55] Leslie G. Valiant. Negation can be exponentially powerful. *Theoretical Computer Science*, 12(3):303–314, 1980.

[56] Avi Wigderson. The complexity of graph connectivity. In Ivan M. Havel and Václav Koubek, editors, *Mathematical Foundations of Computer Science 1992*, pages 112–132, Berlin, Heidelberg, 1992. Springer Berlin Heidelberg.

[57] Ran Raz Amir Yehudayoff. Lower bounds and separations for constant depth multilinear circuits. *Computational Complexity*, 18(2):171–207, 2009.

A Why previous results give FPT bounds

We sketch why previous lower bounds for set-multilinear circuits of large (say constant) depth do not yield non-FPT bounds.

Nisan and Wigderson's technique. We start with the result of Nisan and Wigderson [36] which yields non-FPT bounds for set-multilinear circuits of product-depth 1 computing $\mathrm{IMM}_{n,d}$ but an FPT bound of $\exp(\Omega(d^{1/\Delta}))$ for product-depth $\Delta > 1$. Assume d is even. Denote by $\mathbb{F}_{\mathrm{sm}}[(X_1, \ldots, X_d)]$ the space of set-multilinear polynomials w.r.t. the partition $(X_1, \ldots, X_d)$. A $\Sigma\Pi\Sigma$ set-multilinear formula for such a polynomial is an expression of the form

$$F(X) = \sum_{i=1}^{s} \prod_{j=1}^{d} \ell_{i,j}(X_j) \tag{6}$$

where each $\ell_{i,j}$ is a homogeneous linear polynomial in the variables X_j. We want to show that $\mathrm{IMM}_{n,d} \in \mathbb{F}_{\mathrm{sm}}[(X_1, \ldots, X_d)]$ cannot be computed by a small $\Sigma\Pi\Sigma$ set-multilinear formula.

To do this, we associate with each polynomial a matrix obtained as follows. Assume that we partition the set $[d]$ into two sets $\mathcal{P}$ and $\mathcal{N}$ respectively, and let $\mathcal{M}^{\mathcal{P}}$ and $\mathcal{M}^{\mathcal{N}}$ denote the sets of multilinear monomials over the variable partitions $(X_i : i \in \mathcal{P})$ and $(X_j : j \in \mathcal{N})$ respectively. Note that any set-multilinear monomial m over $(X_1, \ldots, X_d)$ can be written uniquely as $m_1 \cdot m_2$ where $m_1 \in \mathcal{M}^{\mathcal{P}}$ and $m_2 \in \mathcal{M}^{\mathcal{N}}$. We associate with any set-multilinear polynomial P the matrix M_P with rows labelled by $m_1 \in \mathcal{M}^{\mathcal{P}}$ and columns labelled by $m_2 \in \mathcal{M}^{\mathcal{N}}$, where the (m_1, m_2)th entry of M_f is the coefficient of $m_1 \cdot m_2$ in P. We use the rank of M_P (denoted simply $\mathrm{rank}(P)$) to measure the complexity of P.

This is useful because of the following observation. Consider any summand $\prod_{j=1}^{d} \ell_{i,j}(X_j)$ on the right hand side of (6). It is easy to check that the matrix associated with the corresponding polynomial has rank at most 1. As rank is sub-additive, it follows that $\mathrm{rank}(M_F) \leqslant s$ for a formula F with at most s such summands. On the other hand, note that as $|X_i| = n$ for $i \in \{1, d\}$ and $|X_i| = n^2$ for $i \in [d]\backslash\{1, d\}$, and setting $\mathcal{P}$ and $\mathcal{N}$ to be the sets of even and odd numbers in $[d]$ respectively, then we see that M_f is a matrix of dimensions $n^{d-1} \times n^{d-1}$. On the other hand, one can easily check that for $P = \mathrm{IMM}_{n,d}$, the matrix M_P is a permutation matrix and thus has full rank. Our observation above then implies that any $\Sigma\Pi\Sigma$ formula for $\mathrm{IMM}_{n,d}$ must have size at least n^{d-1}, which is $N^{\Omega(d)}$ as long as $d \leqslant N^{O(1)}$. This yields a strong (and in fact optimal: $\mathrm{IMM}_{n,d}$ is the sum of exactly n^{d-1} monomials) non-FPT lower bound against product-depth 1 set-multilinear formulas.

Unfortunately, this method as it is does not work for larger product depths. Consider the following "Product of Inner Products" polynomial, which is an example due to Nisan and Wigderson. Assume that each $X_i = \{x_{i,1}, \ldots, x_{i,n}\}$ for $i \in \{1, d\}$ and $X_i = \{x_{i,1}, \ldots, x_{i,n^2}\}$ for

$i \notin \{1, d\}$. Define

$$\mathrm{PIP}(X_1, \ldots, X_d) = \left(\sum_{k=1}^{n} x_{1,k} x_{d,k}\right) \cdot \prod_{j=1}^{d/2-1} \left(\sum_{k=1}^{n^2} x_{2j,k} x_{2j+1,k}\right).$$

The above is a formula of product-depth 2 and it can be checked that, for $\mathcal{P}$ and $\mathcal{N}$ being the set of odd and even numbers respectively, M_{PIP} is also a permutation matrix, and hence full-rank.

To get around this, Nisan and Wigderson combined the product-depth 1 lower bound with *random restrictions.* More precisely, we choose a (random) set $I \subseteq [d]$ and set all the variables in the set $\bigcup_{j \notin I} X_j$ to constants. Restricting a formula F this way ensures that we get a set-multilinear formula F' w.r.t. the variable partition $(X_i : i \in I)$. In the example of the PIP polynomial above, it can be seen that if I is chosen randomly, with probability $1 - \exp(-\Omega(d))$ we have $\ell = \Omega(d)$ many terms in the products that become *linear* polynomials, which turns out to imply that the rank of the corresponding formula F' is at most $n^{|I|-\ell}$. A similar fact can be proved for formulas of any product-depth Δ, and this can be used to prove that for any formula F of product-depth Δ and size $\exp(O(\Delta d^{1/\Delta}))$, there is a restriction under which the rank of F is small. On the other hand, it is possible to show that under such a family of restrictions, the polynomial $\mathrm{IMM}_{n,d}$ retains its structure and remains full rank. This implies a size lower bound of $\exp(\Omega(\Delta d^{1/\Delta}))$ for computing this polynomial.

Note that the lower bound obtained above is an FPT lower bound. Unfortunately, this limitation is inherent to this technique, as it is easy to show that this method outlined above cannot prove a lower bound greater than $\exp(O(d)) \cdot \mathrm{poly}(N)$. This is because there are essentially only 2^d distinct restrictions (one for each $I \subseteq [d]$).[9] It is possible to construct, for each such restriction, a single "PIP-type" polynomial that is full-rank even after this restriction. A suitable linear combination of these polynomials yields a formula of product-depth 2 which remains full-rank after any restriction. Hence, to prove a non-FPT lower bound even for product-depth 2, a new idea is necessary.

Shifted Partial Derivatives. More recently, non-FPT lower bounds are also proved [15] against $\Sigma\Pi\Sigma\Pi$ set-multilinear formulas, which are a special case of product-depth 2. In fact, it is known that any such formula computing $\mathrm{IMM}_{n,d}$ must have size $n^{\Omega(\sqrt{d})}$, which is tight. This method is based on an extension of the Partial Derivative technique, called the *Shifted Partial Derivative* technique, due to Kayal [25]. Kayal defines a new complexity measure and shows that this measure is small even for products of low-degree polynomials. This implies a lower bound against set-multilinear[10] $\Sigma\Pi\Sigma\Pi^{[t]}$ formulas, which are sums of products of polynomials of degree at most t, for small t.

To obtain a lower bound against $\Sigma\Pi\Sigma\Pi$ set-multilinear formulas, we again apply a random restriction that sets each variable to 0 with high probability. This ensures that each product gate that involves many variables is set to 0 with high probability, and hence that the formula restricts to a $\Sigma\Pi\Sigma\Pi^{[t]}$ formula with high probability. At this point, the previous lower bound idea applies.

Unfortunately, it is unclear how to use this idea to prove even a lower bound against $\Sigma\Pi\Sigma\Pi\Sigma$ formulas, as these formulas are resistant to the random restriction idea (a generic sum gate does not vanish under random restrictions except with negligible probability).

[9] Strictly speaking, this is an undercount, as we also have the choice of the underlying constants. However, one can show that the constants do not significantly affect the argument.

[10] More generally, this technique also works for homogeneous formulas.

Raz's technique. Raz [39] generalized Nisan and Wigderson's results in a different direction by showing lower bounds for *multilinear* (not just set-multilinear) formulas. The heart of Raz's lower bound (and also followups [57, 13]) is to consider multilinear polynomials on a set of variables X of n variables which is partitioned into two sets Y and Z of size $n/2$ each. Any multilinear monomial m over X factors uniquely as $m_1 \cdot m_2$ where m_1 and m_2 are multilinear monomials over Y and Z respectively. Similar to the set-multilinear case above, we define the matrix M'_P (for a multilinear polynomial $P \in \mathbb{F}[X]$) to be the $2^{n/2} \times 2^{n/2}$ matrix whose (m_1, m_2)th entry is the coefficient of $m = m_1 \cdot m_2$ in P.

The rank of M'_P is used as a measure of the complexity of f. In order to prove lower bounds, this matrix has to be of large rank, in fact, at least $2^{n/2-o(n)}$. However, it can be easily checked that if f is a polynomial of degree at most d, then the rank of M'_f is at most $\binom{n/2}{\leqslant d} = 2^{o(n)}$ if $d = o(n)$. So this method cannot prove lower bounds in this regime.

However, we can prove lower bounds for polynomials of degree $d = o(n)$ by setting most variables to constants in the underlying field. In this situation, we again have reduced to the case when the number of variables $n_1 = O(d)$. However, in this situation, we can only hope to prove a lower bound of the form $f(n_1) = f'(d)$, which is an FPT lower bound.