Low-Depth Arithmetic Circuit Lower Bounds: Bypassing Set-Multilinearization

Prashanth Amireddy ✉
Harvard University, Cambridge, MA, USA

Ankit Garg ✉
Microsoft Research, Bangalore, India

Neeraj Kayal ✉
Microsoft Research, Bangalore, India

Chandan Saha ✉
Indian Institute of Science, Bangalore, India

Bhargav Thankey ✉
Indian Institute of Science, Bangalore, India

Abstract

A recent breakthrough work of Limaye, Srinivasan and Tavenas [29] proved superpolynomial lower bounds for low-depth arithmetic circuits via a "hardness escalation" approach: they proved lower bounds for low-depth *set-multilinear* circuits and then lifted the bounds to low-depth general circuits. In this work, we prove superpolynomial lower bounds for low-depth circuits by bypassing the hardness escalation, i.e., the set-multilinearization, step. As set-multilinearization comes with an exponential blow-up in circuit size, our direct proof opens up the possibility of proving an exponential lower bound for low-depth homogeneous circuits by evading a crucial bottleneck. Our bounds hold for the iterated matrix multiplication and the Nisan-Wigderson design polynomials. We also define a subclass of unrestricted depth homogeneous formulas which we call *unique parse tree* (UPT) formulas, and prove superpolynomial lower bounds for these. This significantly generalizes the superpolynomial lower bounds for *regular* formulas [6, 19].

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory

Keywords and phrases arithmetic circuits, low-depth circuits, lower bounds, shifted partials

Digital Object Identifier 10.4230/LIPIcs.ICALP.2023.12

Category Track A: Algorithms, Complexity and Games

Related Version *Full Version*: `https://eccc.weizmann.ac.il/report/2022/151/`

Funding *Prashanth Amireddy*: Supported in part by a Simons Investigator Award and NSF Award CCF 2152413 to Madhu Sudan. A part of this work was done while the author was a research fellow at Microsoft Research, India.
Chandan Saha: Partially supported by a MATRICS grant of the Science and Engineering Research Board, DST, India.
Bhargav Thankey: Supported by the Prime Minister's Research Fellowship, India.

Acknowledgements We would like to thank the anonymous reviewers for their valuable feedback.

1 Introduction

Arithmetic circuits are a natural model for computing polynomials using the basic operations of addition and multiplication. One of the most fundamental questions about arithmetic circuits is about finding a family of explicit polynomials (if they exist) that cannot be computed by polynomial-sized arithmetic circuits. The existence of such explicit polynomials was conjectured by Valiant in 1979 [40] and is the famed VP vs VNP conjecture. Arithmetic circuit lower bounds are expected to be easier than Boolean circuit lower bounds. Among many reasons, one is due to the phenomenon of depth reduction. Arithmetic circuits can be converted into low-depth circuits preserving the output polynomial and not blowing up the size too much [1, 10, 22, 39, 41]. Due to this, strong enough lower bounds even for restrictive models of computation like depth-3 circuits or homogeneous depth-4 circuits can lead to superpolynomial arithmetic circuit lower bounds.

Arithmetic formulas are an important subclass of arithmetic circuits where the out-degree of every gate is at most 1. For constant-depth, formulas and circuits are polynomially related. Also, all our results deal with formulas. So we will only refer to formulas from here on. We consider (families of) polynomials having degree at most polynomial in n, the number of variables. One of the first results studying low-depth arithmetic formulas was that of [32], who proved lower bounds for homogeneous depth-3 formulas. Progress on homogeneous formula lower bound was stalled for a while, and then various lower bounds for homogeneous depth-4 formulas were proven in a series of works [6, 9, 13, 14, 19, 25, 26]. There was limited progress for higher-depth formulas, and lower bounds remained open even for depth-5 formulas. In a recent breakthrough work, [29] proved superpolynomial lower bounds for constant-depth arithmetic formulas. Their lower bounds are of the form $n^{\Omega(\log(n)^{c_\Delta})}$ for a constant $0 < c_\Delta < 1$ depending on the depth Δ of the formula. The following two open problems naturally emerge out of their work.

▶ **Open Problem 1.** *Prove superpolynomial lower bounds for* general *formulas or even homogeneous formulas. (A formula is homogeneous if every gate computes a homogeneous polynomial.)*

▶ **Open Problem 2.** *Prove* exponential *lower bounds for constant-depth arithmetic formulas. This is interesting even for homogeneous depth-*5 *formulas.*

Towards answering Open Problems 1.1 and 1.2, let us examine the lower bound proof in [29] at a high level. Their proof has two main steps: First, they reduce the problem of proving lower bounds for low-depth formulas to the problem of proving lower bounds for low-depth *set-multilinear* formulas; set-multilinear formulas are special homogeneous formulas with an underlying partition of the variables into subsets. [29] calls such reductions "hardness escalation". Second, they use an interesting adaptation of the rank of the partial derivatives matrix measure [31] to prove a lower bound for low-depth set-multilinear formulas. They call this measure *relative rank* (relrk). The effectiveness of the relrk measure crucially depends on a certain "imbalance" between the sizes of the sets used to define set-multilinear polynomials. The proof in [29] raises two natural questions:

Question 1: Can we bypass the hardness escalation, i.e., the set-multilinearization, step?
Question 2: Can we design a measure that exploits some weakness of homogeneous (but not necessarily set-multilinear) formulas directly?

Motivations for studying Question 1. Set-multilinear circuits form a natural circuit class as most interesting polynomial families, such as the determinant, permanent, iterated matrix multiplication, etc., are set-multilinear. However, set-multilinearization comes with an exponential blow up in size – a homogeneous, depth-Δ formula computing a set-multilinear polynomial of degree d can be converted to a set-multilinear formula of depth Δ and size $d^{O(d)} \cdot s$ (see [29]). So, an exponential lower bound for low-depth set-multilinear formulas does not imply an exponential lower bound for low-depth homogeneous formulas since we are restricted to work with $d \leq \frac{\log n}{\log \log n}$. Indeed, it is possible to strengthen and refine the argument in [29] to get an *exponential* lower bound for low-depth set-multilinear formulas (see [2]). An approach that evades the hardness escalation step, which is a critical bottleneck, and directly works with homogeneous formulas has the potential to avoid the $d^{O(d)}$ loss and give an exponential lower bound for low-depth homogeneous formulas. For instance, the direct arguments in [14,26] yield exponential lower bounds for homogeneous depth-4 formulas. If we go via the hardness escalation approach, we get a quasi-polynomial lower bound for the same model. Besides, a direct argument can also be used to prove lower bounds for polynomials that do not have a non-trivial set-multilinear component, see the full version of this article [2] for more details. The hardness escalation approach of [29] can not yield such a lower bound. Furthermore, it is conceivable that a direct argument can also be used to obtain functional lower bounds for low-depth formulas which might be useful in proof complexity.

Motivations for studying Question 2. Typical measures used for proving lower bounds for arithmetic circuits include the partial derivatives measure (PD) [32,38], the rank of the partial derivatives matrix measure (a.k.a. evaluation dimension) [31,34,36], the shifted partials measure (SP) and its variants [9,14,19], the affine projections of partials measure (APP) [7,15], etc. All these measures are defined for *any* polynomial, which is not necessarily set-multilinear. Whereas the relrk measure used in [29], although very effective, is defined for set-multilinear polynomials. Measures such as PD, SP, and APP have the geometrically appealing property that they are invariant under the application of invertible linear transformations on the variables. Since low-depth formulas, as well as low-depth homogeneous formulas, are closed under linear transformations, it is natural to look for measures that do not blow up much on applying linear transformations. Another important motivation for studying Question 2 is to learn low-depth homogeneous formulas. While the "hardness escalation" paradigm of reducing to the set-multilinear case works for proving lower bounds, it is not clear how to exploit it to design learning algorithms for low-depth formulas. Lower bounds for arithmetic circuits are intimately connected to learning [5,7,18,42]. Hence if we have a lower bound measure that directly exploits the weakness of low-depth homogeneous formulas, it opens up the possibility of new learning algorithms for such models.

1.1 Our results

We answer Questions 1 and 2 by giving a direct lower bound for low-depth homogeneous formulas via the SP measure which was used in the series of works on homogeneous depth-4 exponential lower bounds. While our proof also yields lower bounds only in the low-degree setting, the hope is that it could potentially lead to a stronger lower bound in the future.

Consider the *shifted partials* measure: $\mathsf{SP}_{k,\ell}(f) := \dim\langle \mathbf{x}^\ell \cdot \boldsymbol{\partial}^k(f)\rangle$, where f is a polynomial. That is, $\mathsf{SP}_{k,\ell}(f)$ is the dimension of the space spanned by the polynomials obtained by multiplying degree ℓ monomials to partial derivatives of f of order k. Also, for convenience, let us denote by $M(n,k) := \binom{n+k-1}{k}$ the number of monomials of degree k in n variables. Then note that for a homogeneous polynomial f of degree d, $\mathsf{SP}_{k,\ell}(f) \leq \min\{M(n,k)M(n,\ell), M(n,d-k+\ell)\}$.

We show that for polynomials computed by low-depth homogeneous formulas, the shifted partials measure with an appropriate setting of k and ℓ is substantially smaller than the above upper bound. At the same time, we exhibit explicit "hard" polynomials for which the shifted partials measure is close to the above bound, hence yielding a lower bound.

▶ **Theorem 3** (Lower bound for low-depth homogeneous formulas via shifted partials). *Let C be a homogeneous formula of size s and product-depth Δ that computes a polynomial of degree d in n variables. Then for appropriate values of k and ℓ,*

$$\mathsf{SP}_{k,\ell}(C) \leq \frac{s\, 2^{O(d)}}{n^{\Omega(d^{2^{1-\Delta}})}} \min\{M(n,k)M(n,\ell), M(n,d-k+\ell)\}.$$

At the same time, there are homogeneous polynomials f of degree d in n variables (e.g., an appropriate projection of iterated matrix multiplication polynomial, Nisan-Wigderson design polynomial, etc.) such that

$$\mathsf{SP}_{k,\ell}(f) \geq 2^{-O(d)} \min\{M(n,k)M(n,\ell), M(n,d-k+\ell)\}.$$

This gives a lower bound of $\frac{n^{\Omega(d^{2^{1-\Delta}})}}{2^{O(d)}}$ on the size of homogeneous product-depth Δ formulas for f.

▶ Remark 4.

1. At the end of this section, we briefly remark why it is surprising that we are able to obtain the above lower bound using shifted partials. We also show that the lower bound can be derived using the *affine projections of partials* (APP) measure (Lemma 19).
2. The above lower bound is slightly better than the bound of [29]. Instead of the $d^{O(d)}$ loss incurred due to converting homogeneous to set-multilinear formulas, our analysis incurs a $2^{O(d)}$ loss; in fact, this loss can be brought down to $2^{O(k)}$, but we ignore this distinction as we set $k = \Theta(d)$ in the analysis. So, for example, for homogeneous product-depth 2 formulas, our superpolynomial lower bound continues to hold for a higher degree ($\log^2(n)$ vs $(\log(n)/\log\log(n))^2$ in [29]). While the improvement may be insignificant, this hints at something interesting going on with the direct approach (see Section 1.2).

Lower bounds for general-depth arithmetic formulas are expected to be easier than arithmetic circuit lower bounds. However, despite several approaches and attempts (e.g. via tensor rank lower bounds [35]), we still do not have superpolynomial arithmetic formula lower bounds. There has been some success though in proving lower bounds for some natural restricted models (apart from the depth restrictions considered above). For example, [19] considered the model of *regular* arithmetic formulas. These are formulas which consist of alternating layers of addition (+) and multiplication (×) gates such that the fanin of all gates in any fixed layer is the same. This is a natural model and the best-known formulas for many interesting polynomial families like determinant, permanent, iterated matrix multiplication, etc. are all regular. [19] proved a superpolynomial lower bound on the size of regular formulas for an explicit polynomial and later [6] proved a tight lower bound for the iterated matrix multiplication polynomial.

We prove superpolynomial lower bounds for a more general model.[1] Consider a model of homogeneous arithmetic formulas consisting of alternating layers of addition (+) and multiplication (×) gates such that the fanin of all addition gates can be arbitrary but fanin of product gates in any fixed layer is the same. We call these *product-regular.* We prove super-polynomial lower bounds for homogeneous product-regular formulas. Previously we did not know of lower bounds for even a much simpler model where the fanins of all the product gates are fixed to 2.

In fact, we prove lower bounds for an even more general model which we call *Unique Parse Tree* (UPT) formulas. A parse tree of a formula is a tree where for every + gate, one picks exactly one child and for every product gate, we pick all the children. Then we "short circuit" all the addition gates. Parse trees capture the way monomials are generated in a formula. We say that a formula is UPT if all its parse trees are isomorphic. A product-regular formula is clearly UPT. In the theorem below, $IMM_{n,\log n}$ is the iterated multiplication polynomial of degree $\log n$.

[1] The model in [6, 19] allowed slight non-homogeneity with the formal degree upper bounded by a small constant times the actual degree. However, we only work with homogeneous formulas.

▶ **Theorem 5.** *Any UPT formula computing $IMM_{n,\log(n)}$ has size at least $n^{\Omega(\log\log(n))}$. A similar lower bound holds for the Nisan-Wigderson design polynomial.*

▶ Remark 6.

1. While homogeneous product-regular formulas are restricted to compute polynomials with only certain degrees (e.g., higher product-depth cannot compute prime degrees), homogeneous UPT formulas do not suffer from this restriction.
2. While this result (which is obtained using the SP and the APP measures) could possibly also be obtained by defining a similar model in the set-multilinear world, proving a lower bound there and then transporting it back to the homogeneous world, our framework has fewer number of moving parts and hence makes it easier to derive such results.

Challenges to using the SP measure. Let us remark briefly why it is surprising that we are able to prove low-depth lower bounds via shifted partials. [8,37] showed that the PD measure of the polynomial $(x_1^2 + \cdots + x_n^2)^{\frac{d}{2}}$ is the maximum possible when the order of derivatives, k, is at most $\frac{d}{2}$. Notice that $(x_1^2 + \cdots + x_n^2)^{\frac{d}{2}}$ can be computed by a homogeneous depth-4 formula of size $O(nd)$. So, it is not possible to prove super-polynomial lower bounds for low-depth homogeneous formulas using the PD measure as it is. One may ask if the SP measure also has a similar limitation. Some of the finer separation results in [23,24] indicate that the SP measure (and some of its variants) can be fairly large for homogeneous depth-4 and depth-5 formulas for the choices of k used in prior work. Also, the exponential lower bounds for homogeneous depth-4 circuits in [14,26] use random restrictions along with a variant of the SP measure. It is not clear how to leverage random restrictions for even homogeneous depth-5 circuits – this is also pointed out in [29]. Fortunately, [23,24] do not rule out the possibility of using SP for all choices of parameters, like, say, $k \approx \frac{d}{2}$, to prove lower bounds for low-depth homogeneous formulas. But, the original intuition from algebraic geometry that led to the development of the SP measure (see [9] Section 2.1) breaks down completely when k is so large (see [2]). Despite these apparent hurdles, and to our surprise, we overcome these challenges and are able to use SP with $k \approx \frac{d}{2}$ to prove super-polynomial lower bounds for low-depth homogeneous formulas. To the best of our knowledge, no previous work uses SP with this high a value of k.

1.2 Techniques and proof overview

In this section, we explain the proof idea and compare it with that in [29]. A lot of lower bounds in arithmetic complexity follow the following outline.

Step 1: Depth reduction. One first shows that if $f(\mathbf{x})$ is computed by a *small* circuit from some restricted subclass of circuits, then there is a corresponding subclass of depth-4 circuits such that $f(\mathbf{x})$ is also computed by a *relatively small* circuit from this subclass[2]. The resulting subclass is of the form: $f(\mathbf{x}) = \sum_{i=1}^{s} \prod_{j=1}^{t_i} Q_{i,j}$. Usually there are simple restrictions on the degrees of $Q_{i,j}$'s. For example, they could be upper bounded by some number.

[2] Some major results in the area such as [29,33] did not originally proceed via a depth reduction but instead analysed formulas directly. These results can however be restated as first doing a depth reduction and then applying the appropriate measure.

Step 2: Employing a suitable set of linear maps. Let $\mathbb{F}[\mathbf{x}]^{=d}$ be the space of homogeneous polynomials of degree d, W be a suitable vector space, and $\mathrm{Lin}(\mathbb{F}[\mathbf{x}]^{=d}, W)$ be the space of linear maps from $\mathbb{F}[\mathbf{x}]^{=d}$ to W. We choose a suitable set of linear maps $\mathcal{L} \subseteq \mathrm{Lin}(\mathbb{F}[\mathbf{x}]^{=d}, W)$ that define a complexity measure $\mu_{\mathcal{L}}(f) := \dim(\mathcal{L}(f))$, where $\mathcal{L}(f) := \langle\{L(f) \ : \ L \in \mathcal{L}\}\rangle$.

We would like to choose $\mathcal{L}$ so that it identifies some weakness of the terms $\prod_{j=1}^{t} Q_j$ in the depth-4 circuit. That is, $\mu_{\mathcal{L}}\left(\prod_{j=1}^{t} Q_j\right)$ should be much smaller than $\mu_{\mathcal{L}}(f)$ for a generic f. For e.g., if Q_j's are all linear polynomials, we can choose $\mathcal{L}$ to be the partial derivatives of order k, $\boldsymbol{\partial}^k$. Then, $\mu_{\mathcal{L}}\left(\prod_{j=1}^{t} Q_j\right) \le \binom{t}{k} \ll \binom{n+k-1}{k}$ which is the value for a generic f (for $k \le t/2$). This is the basis of the homogeneous depth-3 formula lower bound in [32].

For proving lower bounds for bounded bottom fan-in depth-4 circuits (i.e., when degree of Q_j's is upper bounded by some number), [9, 13] introduced the SP measure and used the linear maps $\mathcal{L} = \mathbf{x}^{\ell} \cdot \boldsymbol{\partial}^k$. The main insight in their proof was that if we apply a partial derivative of order k on $\prod_{j=1}^{t} Q_j$ and use the product rule, then at least $t-k$ of the Q_j's remain untouched. This structure can then be exploited by the shifts to get a lower bound. This intuition however completely breaks down for $k \ge t$ (see [2]). Due to this, progress remain stalled for higher depth arithmetic circuit lower bounds via SP.

In a major breakthrough, [29] gets around the above obstacle by working with set-multilinear circuits which entails working with polynomials over d sets of variables $(\mathbf{x}_1, \ldots, \mathbf{x}_d)$, $|\mathbf{x}_i| = n$. Let us use the shorthand $\mathbf{x}_S = (\mathbf{x}_i)_{i \in S}$. The products they deal with are of the form $\prod_{j=1}^{t} Q_j(\mathbf{x}_{S_j})$, where $S_1, S_2, \ldots, S_t$ form a partition of $[d]$. The set of linear maps they use are $\mathcal{L} = \Pi \circ \partial_{\mathbf{x}_A}$ for a subset $A \subseteq [d]$. Here, Π is a map that sets $n - n_0$ variables in each of the variable sets in $\mathbf{x}_{[d]\setminus A}$ to 0. They observe (for the appropriate choice of n_0) that $\mu_{\mathcal{L}}\left(\prod_{j=1}^{t} Q_j(\mathbf{x}_{S_j})\right) \le \frac{n^{|A|}}{2^{\frac{1}{2}\sum_{j=1}^{t} \mathrm{imbalance}_j}}$.

Here, $\mathrm{imbalance}_j = ||A \cap S_j| \log(n) - |S_j \setminus A| \log(n_0)|$. For the appropriate choice of n_0, a generic set-multilinear f satisfies $\mu_{\mathcal{L}}(f) = n^{|A|}$, so that lower bound (on the number of summands) obtained is exponential in the total imbalance $\sum_{j=1}^{t} \mathrm{imbalance}_j$. [29] observe that this quantity is *somewhat large* for the depth-4 circuits that they consider.

The core of the above derivatives-based argument allows us to unravel some structure in partial derivatives of order k applied on $\prod_{j=1}^{t} Q_j$ for values of $k \gg t$. We use this to derive a structure for the partial derivative space of a product $\prod_{j=1}^{t} Q_j(\mathbf{x})$. Consider a partial derivative operator of order k indexed by a multiset α of size k. Using the chain rule,

$$\partial_\alpha \prod_{j=1}^{t} Q_j = \sum_{\alpha_1, \ldots, \alpha_t : \sum_{i=1}^{t} \alpha_i = \alpha} c^{\alpha}_{\alpha_1, \ldots, \alpha_t} \prod_{j=1}^{t} \partial_{\alpha_j} Q_j$$

for appropriate constants $c^{\alpha}_{\alpha_1, \ldots, \alpha_t}$'s. In the product $\prod_{j=1}^{t} \partial_{\alpha_j} Q_j$, we can try to club terms into two groups depending on if the size of $|\alpha_j|$ is small or large. It turns out that the right threshold for $|\alpha_j|$ is $k \deg(Q_j)/d$ (i.e., if we divide the order of the derivatives proportional to the degrees of the terms). Let $S := \{j : |\alpha_j| \le k \deg(Q_j)/d\}$. Define $k_0 := \sum_{j \in S} |\alpha_j|$ and $\ell_0 := \sum_{j \in \overline{S}} (\deg(Q_j) - |\alpha_j|)$. Notice that we can write the product $\prod_{j=1}^{t} \partial_{\alpha_j} Q_j$ as $P \prod_{j \in S} \partial_{\alpha_j} Q_j$, for a degree ℓ_0 polynomial P. Hence, $\partial_\alpha \prod_{j=1}^{t} Q_j$ is a sum of terms of this form. While it is not immediate (due to the condition on α_j's in S), with a bit more work, one can combine the product of partials into a single partial.

What can we say about k_0 and ℓ_0? It turns out that the quantity that comes up in the calculations is $k_0 + \frac{k}{d-k}\ell_0$ and it satisfies $k_0 + \frac{k}{d-k}\ell_0 \le k$. Note that k_0 is between 0 and k, and ℓ_0 between 0 and $d-k$. So the normalization brings ℓ_0 to the right "scale".

It turns out we can give a better bound in terms of a quantity we call *residue* defined as

$$\mathsf{residue}_k(d_1,\ldots,d_t) := \frac{1}{2}\cdot \min_{k_1,\ldots,k_t\in\mathbb{Z}} \sum_{j=1}^{t}\left|k_j - \frac{k}{d}\cdot d_j\right|.$$

and having the property that:

▶ **Proposition 7.** *Let k_0 and ℓ_0 be defined as above. Then, $k_0 + \frac{k}{d-k}\ell_0 \le k - \mathsf{residue}_k(d_1,\ldots,d_t)$, where $d_j = deg(Q_j)$.*

We want to spread the derivatives equally among all terms but cannot due to *integrality issues*. The residue captures this quantitatively and as described below, is what gives us our lower bounds. While the proof in [29] also relies on an integrality issue, there it originates from an imbalance between the sizes of the variable sets involved in a set-multilinear partition (as the map Π sets some variables in certain sets to 0). In contrast, we show that the integrality issue arising directly from the derivatives can be leveraged without involving set-multilinearity. In this sense, our approach is *conceptually* direct and simpler. Combined with the above discussion, we get the following structural lemma about the derivative space of $\prod_{j=1}^{t} Q_j$.

▶ **Lemma 8.**

$$\left\langle \boldsymbol{\partial}^k\left(Q_1\cdots Q_t\right)\right\rangle \subseteq \sum_{\substack{S\subseteq[t],\ k_0\in[0..k],\ \ell_0\in[0..(d-k)],\\ k_0+\frac{k}{d-k}\cdot\ell_0\ \le\ k-\mathsf{residue}_k(d_1,\ldots,d_t)}} \left\langle \mathbf{x}^{\ell_0}\cdot\boldsymbol{\partial}^{k_0}\left(\prod_{j\in S} Q_j\right)\right\rangle.$$

Now we have the choice to utilize the above structure using an additional set of linear maps. Both shifts and projections give similar lower bounds, so let us explain shifts here. Note that there is an intriguing possibility of getting even better lower bounds (in terms of dependence on d) using other sets of linear maps! From the above structural result, we have

$$\left\langle \mathbf{x}^{\ell}\cdot\boldsymbol{\partial}^k\left(Q_1\cdots Q_t\right)\right\rangle \subseteq \sum_{\substack{S\subseteq[t],\ k_0\in[0..k],\ \ell_0\in[0..(d-k)],\\ k_0+\frac{k}{d-k}\cdot\ell_0\ \le\ k-\mathsf{residue}_k(d_1,\ldots,d_t)}} \left\langle \mathbf{x}^{\ell+\ell_0}\cdot\boldsymbol{\partial}^{k_0}\left(\prod_{j\in S} Q_j\right)\right\rangle.$$

Thus we can upper bound,

$$\begin{aligned}\mathsf{SP}_{k,\ell}((Q_1\cdots Q_t)) &\le 2^t\cdot d^2\cdot \max_{\substack{k_0,\ell_0\ge 0\\ k_0+\frac{k}{d-k}\cdot\ell_0\ \le\ k-\mathsf{residue}_k(d_1,\ldots,d_t)}} M(n,k_0)\cdot M(n,\ell_0+\ell)\\ &\le 2^t\cdot d^2\frac{2^{O(d)}}{n^{\mathsf{residue}_k(d_1,\ldots,d_t)}}\ \min\{M(n,k)M(n,\ell), M(n,d-k+\ell)\},\end{aligned}$$

where the second inequality follows from elementary calculations.

Now to upper bound the shifted partial dimension of polynomials computed by low-depth formulas, we give a decomposition for such formulas into sums of products of polynomials (Lemma 16) where the degree sequences are carefully chosen so that that the residues can be simultaneously lower bounded for all the terms (Lemma 17). While in a different context, these calculations do bear similarity with related calculations in [29].

Step 3: Lower bounding $\dim(\mathcal{L}(f))$ for an explicit f. As a last step, one shows that for some explicit candidate hard polynomial $\dim(\mathcal{L}(f))$ is large and thereby obtains a lower bound. This is another step where bypassing set-multilinearity helps as one is not constrained to pick a set-multilinear hard polynomial. Indeed, using a straightforward analysis we show that the APP measure is high for an explicit non-set-multilinear polynomial (see Remark 23). We also show that the measures are high for more standard polynomial families such as the iterated matrix multiplication polynomials and the Nisan-Wigderson design polynomials.

Application to UPT formulas. We observe here that for the subclass of homogeneous formulas that we call UPT formulas, one can do a depth-reduction to obtain a depth-4 formula in which all the summands have the same factorization pattern (i.e. the sequence of degrees of the factors in all the summands is that same) - see Lemma 30. We further observe (Lemma 31) that for any fixed sequence of degrees, there exists a suitable value of the parameter k such that the residue is *sufficiently large*. This gives us the superpolynomial lower bound for UPT formulas as stated in Theorem 5.

Despite the conceptual directness and simplicity of our approach, in bypassing set-multilinearity, some of the calculations in the analysis become evidently more involved than that in [29]. This is primarily due to the delicate choice of parameters in ratios involving binomial coefficients; this is also the case in several prior exponential lower bound proofs using SP and its variants [14, 16, 26]. Nevertheless, we think that by circumventing a critical bottleneck, the analysis opens up the possibility of an exponential lower bound for low-depth arithmetic circuits. Some of the ideas may indeed yield stronger bounds in the future.

Organization. After describing preliminaries in Section 2, we present a structural theorem about the derivative space of a product of homogeneous polynomials in Section 3. This result is then directly used to upper bound both the SP and APP measures of a product of polynomials. Using this result and a decomposition result for low-depth formulas, we obtain lower bounds for low-depth formulas in Section 4. Finally, we prove lower bounds for UPT formulas in Section 5.

2 Preliminaries

In this section, we give the essential notations and definitions necessary to follow the article.

Let a, b, c be real numbers. Then we define the sets $[a..b] := \{x \in \mathbb{Z} : x \in [a, b]\}$ and $[a] := [1..a]$. For a constant $c \geq 1$ and $b \geq 0$, we say $a \approx_c b$ if $a \in [b/c, b]$. We write $a \approx b$ if $a \approx_c b$ for some (unspecified) constant c. All logarithms have base 2 unless specified otherwise. We denote the fractional part of a by $\{a\} := a - \lfloor a \rfloor$ and the nearest integer of a by $\lfloor a \rceil$. The following quantity will be crucially used in the proofs of our lower bounds. Here we think of $d_1, \ldots, d_t$ as degrees of certain homogeneous polynomials, d as the degree of the product of those polynomials, and k is the order of partial derivatives used for the complexity measures.

▶ **Definition 9** (residue). *For non-negative integers $d_1, \ldots, d_t$ such that $d := \sum_{i=1}^{t} d_i \geq 1$ and $k \in [0..(d-1)]$, we define* $\mathsf{residue}_k(d_1, \ldots, d_t) := \frac{1}{2} \cdot \min_{k_1,\ldots,k_t \in \mathbb{Z}} \sum_{i=1}^{t} \left| k_i - \frac{k}{d} \cdot d_i \right|$.

The factor of half has been included in the definition just to make the statements of some of the lemmas in our analysis simple. It is easy to show that $\mathsf{residue}_k(d_1, \ldots, d_t) \leq \frac{k}{2}$. The minimum is attained when for all $i \in [t]$, $k_i = \left\lfloor \frac{k}{d} \cdot d_i \right\rceil$. When we use residue in the analysis of complexity measures, we would also have the following additional constraints that $k_i \geq 0$ and $k_i \leq d_i$, $k_1 + \cdots + k_n = k$, where k shall be the order of derivatives. As the value of residue can not decrease when we impose these constraints, we omit them.

Let n and n_0 be positive integers. Define variable sets $\mathbf{x} := \{x_1, \ldots, x_n\}$ and $\mathbf{z} := \{z_1, \ldots, z_{n_0}\}$. For a monic monomial m and a $P \in \mathbb{F}[\mathbf{x}]$, we define $\partial_m P \in \mathbb{F}[\mathbf{x}]$ to be the polynomial obtained by successively taking partial derivatives with respect to all the variables of m (counted with their multiplicities). For an integer $\ell \geq 0$, $\mathbf{x}^\ell := \{{x_1}^{e_1} \cdots {x_n}^{e_n} : e_1, \ldots, e_n \in \mathbb{Z}_{\geq 0} \text{ and } \sum_{i\in[n]} e_i = \ell\}$. For an integer $k \geq 0$ and $P \in \mathbb{F}[\mathbf{x}]$, $\boldsymbol{\partial}^k P := \left\{\partial_m P : m \in \mathbf{x}^k\right\}$. For a $P \in \mathbb{F}[\mathbf{x}]$, a map $L : \mathbf{x} \to \langle \mathbf{z} \rangle$, and $\mathcal{S} \subseteq \mathbb{F}[\mathbf{x}]$, $\pi_L(P) \in \mathbb{F}[\mathbf{z}]$ and $\pi_L(\mathcal{S}) \subseteq \mathbb{F}[\mathbf{z}]$ are defined as $\pi_L(P) := P(L(x_1), \ldots, L(x_n))$ and $\pi_L(\mathcal{S}) := \{\pi_L(P) : P \in \mathcal{S}\}$, respectively.

For $\mathcal{S}, \mathcal{T} \subseteq \mathbb{F}[\mathbf{x}]$, $\mathcal{S} \cdot \mathcal{T} := \{P \cdot Q : P \in \mathcal{S} \text{ and } Q \in \mathcal{T}\}$ and $\mathcal{S} + \mathcal{T} := \{P + Q : P \in \mathcal{S} \text{ and } Q \in \mathcal{T}\}$. For a $\mathcal{S} \subseteq \mathbb{F}[\mathbf{x}]$, we define its *span* as $\langle \mathcal{S} \rangle \subseteq \mathbb{F}[\mathbf{x}]$ to be the set of all polynomials which can be expressed as $\mathbb{F}$-linear combinations of elements in $\mathcal{S}$. For a $\mathcal{S} \subseteq \mathbb{F}[\mathbf{x}]$, its *dimension*, denoted by $\dim \mathcal{S}$, refers to the maximum number of *linearly independent* polynomials in $\mathcal{S}$. We can now define the complexity measures for polynomials that we use to prove our lower bounds: the *shifted partials* (SP) measure and the *affine projections of partials* (APP) measure.

▶ **Definition 10** (SP and APP measures)**.** *For a polynomial $P \in \mathbb{F}[\mathbf{x}]$, non-negative integers k, ℓ, and $n_0 \in [n]$, we define $\mathsf{SP}_{k,\ell}(P) := \dim \left\langle \mathbf{x}^\ell \cdot \boldsymbol{\partial}^k P \right\rangle$ and $\mathsf{APP}_{k,n_0}(P) := \max\limits_{L:\mathbf{x}\to\langle\mathbf{z}\rangle} \dim \left\langle \pi_L\left(\boldsymbol{\partial}^k P\right)\right\rangle$.*

SP and APP are *sub-additive.* APP is related to the *skewed partials* and relrk measures used in [15] and [29], respectively. For a comparison, see [2].

Next, we define a subclass of homogeneous formulas which we call *UPT formulas*[3].

▶ **Definition 11.** *A homogeneous formula C is said to be a* unique-parse-tree formula *if all of its parse trees are isomorphic to each other as directed graphs.*

For a UPT formula C, we define its *canonical parse tree* to be some fixed tree among all the parse trees (this is a binary tree without loss of generality). For a detailed definition of (canonical) parse tree, we refer the reader to the full version of this article [2].

Iterated Matrix Multiplication. The iterated matrix multiplication, $IMM_{n,d}$ is a polynomial in $N = d{\cdot}n^2$ variables defined as the $(1,1)$-th entry of the matrix product of d many $n \times n$ matrices whose entries are distinct variables. To prove a lower bound for IMM, we analyze the SP and APP for a projection of IMM, $P_{\mathbf{w}}$ that was introduced in [29].

▶ **Definition 12** (Word polynomial $P_{\mathbf{w}}$ [29])**.** *Given a word $\mathbf{w} = (w_1, \ldots, w_d) \in \mathbb{Z}^d$, let $\mathbf{x}(\mathbf{w})$ be a tuple of d pairwise disjoint sets of variables $(\mathbf{x}_1(\mathbf{w}), \ldots, \mathbf{x}_d(\mathbf{w}))$ with $|\mathbf{x}_i(\mathbf{w})| = 2^{|w_i|}$ for all $i \in [d]$. $\mathbf{x}_i(\mathbf{w})$ will be called negative if $w_i < 0$ and positive otherwise. As the set sizes are powers of 2, we can map the variables in a set $\mathbf{x}_i(\mathbf{w})$ to Boolean strings of length $|w_i|$. Let $\sigma : \mathbf{x} \to \{0,1\}^*$ be such a mapping.*[4] *We extend the definition of σ from variables to set-multilinear monomials as follows: Let $X = x_1 \cdots x_r$ be a set-multilinear monomial where $x_i \in \mathbf{x}_{\phi(i)}(\mathbf{w})$ and $\phi : [r] \to [d]$ be an increasing function. Then, we define a Boolean string $\sigma(X) := \sigma(x_1) \circ \cdots \circ \sigma(x_r)$, where $\circ$ denotes the concatenation of bits. Let $\mathcal{M}_+(\mathbf{w})$ and $\mathcal{M}_-(\mathbf{w})$ denote the set of all (monic) set-multilinear monomials over all the positive sets*

[3] Our definition for UPT formulas is more general than the model considered in a recent paper by Limaye, Srinivasan and Tavenas [30] as we do not impose set-multilinearity.

[4] Note that σ may map a variable from $\mathbf{x}_i(\mathbf{w})$ and a variable from $\mathbf{x}_j(\mathbf{w})$ to the same string if $i \neq j$.

and all the negative sets, respectively. For two Boolean strings a, b, we say $a \sim b$ if a is a prefix of b or vice versa. For a word $\mathbf{w}$, the corresponding word polynomial $P_{\mathbf{w}}$ is defined as $P_{\mathbf{w}} := \sum_{\substack{m_+ \in \mathcal{M}_+(\mathbf{w}),\ m_- \in \mathcal{M}_-(\mathbf{w}) \\ \sigma(m_+) \sim \sigma(m_-)}} m_+ \cdot m_-$.

We will make use of the following lemma from [29] which shows that computing IMM is at least as hard as computing $P_{\mathbf{w}}$. For this, we recall the notion of *unbiased*-ness of $\mathbf{w} = (w_1, \ldots, w_d)$ from [29] – we say that $\mathbf{w}$ is *h-unbiased* if $\max_{i \in [d]} |w_1 + \cdots + w_i| \le h$.

▶ **Lemma 13** (Lemma 7 in [29]). *Let $\mathbf{w} \in [-h..h]^d$ be h-*unbiased*. If for some $n \ge 2^h$, $IMM_{n,d}$ has a formula C of product-depth*[5] *Δ and size s, then $P_{\mathbf{w}}$ has a formula C' of product-depth at most Δ and size at most s. Moreover, if C is homogeneous, then so is C' and if C is UPT, then so is C' with the same canonical parse tree.*[6]

Nisan-Wigderson design polynomial. For a prime power q and $d \in \mathbb{N}$, let $\mathbf{x} = \{x_{1,1}, \ldots, x_{1,q},$
$\ldots, x_{d,1}, \ldots, x_{d,q}\}$. For any $k \in [d]$, the Nisan-Wigderson design polynomial on qd variables, denoted by $NW_{q,d,k}$ or simply NW, is defined as follows:

$$NW_{q,d,k} = \sum_{\substack{h(z) \in \mathbb{F}_q[z]: \\ \deg(h) < k}} \prod_{i \in [d]} x_{i,h(i)}.$$

The IMM and the NW polynomials, and their variants, have been extensively used to prove various circuit lower bounds [3, 4, 11, 14, 16, 19–21, 23, 26, 27, 29, 32].

3 Structure of the space of partials of a product

In this section, we bound the partial derivative space of a product of homogeneous polynomials. In the following lemma, we show that the space of k-th order partial derivatives of a product of polynomials is contained in a sum of shifted partial spaces with shift ℓ_0 and order of derivatives k_0 such that $k_0 + \frac{k}{d-k} \cdot \ell_0$ is "small". Using this lemma, we upper bound the SP and APP measures of a product of homogeneous polynomials. These bounds are then used in Sections 4 and 5 for proving lower bounds for low-depth homogeneous formulas and UPT formulas respectively. Missing proofs from this section can be found in the full version of this article [2],

▶ **Lemma 14** (Upper bounding the partials of a product). *Let n and t be positive integers and $Q_1, \ldots, Q_t$ be non-constant, homogeneous polynomials in $\mathbb{F}[\mathbf{x}]$ with degrees $d_1, \ldots, d_t$ respectively. Let $d := \deg(Q_1 \cdots Q_t) = \sum_{i=1}^{t} d_i$ and $k < d$ be a non-negative integer. Then,*

$$\left\langle \boldsymbol{\partial}^k (Q_1 \cdots Q_t) \right\rangle \subseteq \sum_{\substack{S \subseteq [t],\ k_0 \in [0..k],\ \ell_0 \in [0..(d-k)], \\ k_0 + \frac{k}{d-k} \cdot \ell_0 \ \le\ k - \mathsf{residue}_k(d_1, \ldots, d_t)}} \left\langle \mathbf{x}^{\ell_0} \cdot \boldsymbol{\partial}^{k_0} \left(\prod_{i \in S} Q_i \right) \right\rangle.$$

We now use the above lemma to upper bound the shifted partials and affine projections of partials measures of a product of polynomials.

---

[5] The product-depth of a formula is the maximum number of product gates on any path from the root to a leaf in the formula.

[6] Although the lemma in [29] is stated for set-multilinear circuits, it also applies to homogeneous formulas and UPT formulas (albeit with a mild blow-up in size) by the same argument.

▶ **Lemma 15** (Upper bounding SP and APP of a product). *Let $Q = Q_1 \cdots Q_t$ be a homogeneous polynomial in $\mathbb{F}[x_1, \ldots, x_n]$ of degree $d = d_1 + \cdots + d_t \geq 1$, where Q_i is homogeneous and $d_i := \deg(Q_i)$ for $i \in [t]$. Then, for any non-negative integers $k < d$, $\ell \geq 0$, and $n_0 \leq n$,*

1.
$$\mathsf{SP}_{k,\ell}(Q) \leq 2^t \cdot d^2 \cdot \max_{\substack{k_0,\ell_0 \geq 0 \\ k_0 + \frac{k}{d-k}\cdot \ell_0 \ \leq \ k - \mathsf{residue}_k(d_1,\ldots,d_t)}} M(n, k_0) \cdot M(n, \ell_0 + \ell),$$

2.
$$\mathsf{APP}_{k,n_0}(Q) \leq 2^t \cdot d^2 \cdot \max_{\substack{k_0,\ell_0 \geq 0 \\ k_0 + \frac{k}{d-k}\cdot \ell_0 \ \leq \ k - \mathsf{residue}_k(d_1,\ldots,d_t)}} M(n, k_0) \cdot M(n_0, \ell_0).$$

4 Lower bound for low-depth homogeneous formulas

In this section, we present a superpolynomial lower bound for "low-depth" homogeneous formulas computing the IMM and NW polynomials. We begin by proving a structural result for homogeneous formulas. Missing proofs from this section can be found in the full version of this article [2].

4.1 Decomposition of low-depth formulas

We show that any homogeneous formula can be decomposed as a sum of products of homogeneous polynomials of lower degrees, where the number of summands is bounded by the number of gates in the original formula. The decomposition lemma given below bears some resemblance to a decomposition of homogeneous formulas in [12]. In the decomposition in [12], the degrees of the factors of every summand roughly form a geometric sequence, and hence each summand is a product of a "large" number of factors. Here we show that each summand has "many" low-degree factors. While the lower bound argument in [29] does not explicitly make use of such a decomposition, their inductive argument can be formulated as a depth-reduction or decomposition lemma (with slightly different thresholds for the degrees).

▶ **Lemma 16** (Decomposition of low-depth formulas). *Suppose C is a homogeneous formula of product-depth $\Delta \geq 1$ computing a homogeneous polynomial in $\mathbb{F}[x_1, \ldots, x_n]$ of degree* at least $d > 0$. *Then, there exist homogeneous polynomials $\{Q_{i,j}\}_{i,j}$ in $\mathbb{F}[x_1, \ldots, x_n]$ such that*

1. $C = \sum_{i=1}^{s} Q_{i,1} \cdots Q_{i,t_i}$, *for some $s \leq \mathrm{size}(C)$, and*
2. *for all $i \in [s]$, either*

$$|\{j \in [t_i] : \deg(Q_{i,j}) = 1\}| \geq d^{2^{1-\Delta}} \text{ , or}$$

$$\left|\left\{j \in [t_i] : \deg(Q_{i,j}) \approx_2 d^{2^{1-\delta}}\right\}\right| \geq d^{2^{1-\delta}} - 1 \text{ , for some } \delta \in [2..\Delta].$$

4.2 Low-depth formulas have high residue

The following lemma gives us a value for the order of derivatives k with respect to which low-depth formulas yield high $\mathsf{residue}$. Its proof uses Lemma 16.

▶ **Lemma 17** (Low-depth formulas have high residue). *Suppose C is a homogeneous formula of product-depth $\Delta \geq 1$ computing a polynomial in $\mathbb{F}[x_1, \ldots, x_n]$ of degree d, where $d^{2^{1-\Delta}} = \omega(1)$. Then, there exist homogeneous polynomials $\{Q_{i,j}\}_{i,j}$ in $\mathbb{F}[x_1, \ldots, x_n]$ such that $C = \sum_{i=1}^{s} Q_{i,1} \cdots Q_{i,t_i}$, for some $s \leq \mathrm{size}(C)$. Fixing an arbitrary $i \in [s]$, let $t := t_i$ and define $d_j := \deg(Q_{i,j})$ for $j \in [t]$. Then, $\mathsf{residue}_k(d_1, \ldots, d_t) \geq \Omega\left(d^{2^{1-\Delta}}\right)$, where $k := \left\lfloor \frac{\alpha \cdot d}{1+\alpha} \right\rfloor$, $\alpha := \sum_{\nu=0}^{\Delta-1} \frac{(-1)^\nu}{\tau^{2^\nu - 1}}$, and $\tau := \left\lfloor d^{2^{1-\Delta}} \right\rfloor$.*

4.3 High residue implies lower bounds

For a "random" homogeneous degree-d polynomial in $\mathbb{F}[x_1, \ldots, x_n]$, if the shift ℓ is not too large, we expect the SP measure to be close to the maximum number of operators used to construct the shifted partials space, i.e., $M(n,k) \cdot M(n,\ell)$. Explicit examples of such polynomials are given in Section 4.4. In the lemma below, we derive a lower bound corresponding to the decompositions established above. The main step is to show that the SP measure of a high-residue-decomposition is small.

▶ **Lemma 18** (High residue implies lower bounds). *Let $P = \sum_{i=1}^{s} Q_{i,1} \cdots Q_{i,t_i}$ be a homogeneous n-variate polynomial of degree d where $\{Q_{i,j}\}_{i,j}$ are homogeneous and $\mathsf{SP}_{k,\ell}(P) \geq 2^{-O(d)} \cdot M(n,k) \cdot M(n,\ell)$ for some $1 \leq k < \frac{d}{2}, n_0 \leq n$ and $\ell = \left\lfloor \frac{n \cdot d}{n_0} \right\rfloor$ such that $d \leq n_0 \approx 2(d-k) \cdot \left(\frac{n}{k}\right)^{\frac{k}{d-k}}$. If there is a $\gamma > 0$ such that for all $i \in [s]$, $\mathsf{residue}_k(\deg(Q_{i,1}), \ldots, \deg(Q_{i,t_i})) \geq \gamma$, then $s \geq 2^{-O(d)} \left(\frac{n}{d}\right)^{\Omega(\gamma)}$.*

We state an analogous lemma with APP instead of SP.

▶ **Lemma 19** (High residue implies lower bounds, using APP). *Let $P = \sum_{i=1}^{s} Q_{i,1} \cdots Q_{i,t_i}$ be a homogeneous n-variate polynomial of degree d where $\{Q_{i,j}\}_{i,j}$ are homogeneous and $\mathsf{APP}_{k,n_0}(P) \geq 2^{-O(d)} \cdot M(n,k)$ for some $1 \leq k < \frac{d}{2}, n_0 \leq n$ such that $d \leq n_0 \approx 2(d-k). \left(\frac{n}{k}\right)^{\frac{k}{d-k}}$. If there is a $\gamma > 0$ such that for all $i \in [s]$, $\mathsf{residue}_k(\deg(Q_{i,1}), \ldots, \deg(Q_{i,t_i})) \geq \gamma$, then $s \geq 2^{-O(d)} \cdot \left(\frac{n}{d}\right)^{\Omega(\gamma)}$.*

▶ Remark 20. In the above lemmas, although our lower bound appears as $2^{-O(d)} \cdot n^{\Omega(\gamma)}$, similar calculations actually give a lower bound of $2^{-O(k)} \cdot n^{\Omega(\gamma)}$ for any choice of k and an appropriate choice of ℓ (or n_0 in the case of APP). We do not differentiate between the two, as for our applications (i.e., low-depth circuits and UPT formulas), the value of k we choose is $\Theta(d)$. Moreover, we observe that the factor of $2^{-O(k)}$ in our lower bounds is likely unavoidable for any choice of k and ℓ (or n_0 in the case of APP) *using our current estimates for the complexity measures.* We refer the reader to the full version of this article [2] for more details.

4.4 The hard polynomials

We shall prove our lower bound for the word polynomial $P_{\mathbf{w}}$ introduced in [29] as well as for the Nisan-Wigderson design polynomial. In order to do this, we show that the SP and APP measures of $P_{\mathbf{w}}$ and the SP measure of NW are large for suitable choices of k, ℓ and n_0.

▶ **Lemma 21** ($P_{\mathbf{w}}$ as a hard polynomial)**.** *For integers h, d such that $h > 100$ and any $k \in \left[\frac{d}{30}, \frac{d}{2}\right]$, there exists an h-unbiased word $\mathbf{w} \in [-h..h]^d$, integers $n_0 \leq n$, $\ell = \left\lfloor \frac{n \cdot d}{n_0} \right\rfloor$ such that $n_0 \approx 2(d-k) \cdot \left(\frac{n}{k}\right)^{\frac{k}{d-k}}$ and the following bounds hold: $\mathsf{SP}_{k,\ell}(P_{\mathbf{w}}) \geq 2^{-O(d)} \cdot M(n,k) \cdot M(n,\ell)$ and $\mathsf{APP}_{k,n_0}(P_{\mathbf{w}}) \geq 2^{-O(d)} \cdot M(n,k)$. Here n refers to the number of variables in $P_{\mathbf{w}}$, i.e., $n = \sum_{i \in [d]} 2^{|w_i|}$.*

The following lemma shows that the SP measure of the Nisan-Wigderson design polynomial is "large" for k as high as $\Theta(d)$, if ℓ is chosen suitably.

▶ **Lemma 22** (NW as a hard polynomial)**.** *For $n, d \in \mathbb{N}$ such that $120 \leq d \leq \frac{1}{150}\left(\frac{\log n}{\log \log n}\right)^2$, let q be the largest prime number between $\left\lfloor \frac{n}{2d} \right\rfloor$ and $\left\lfloor \frac{n}{d} \right\rfloor$. For parameters $k \in \left[\frac{d}{30}, \frac{d}{2} - \frac{\sqrt{d}}{8}\right]$ and $\ell = \left\lfloor \frac{qd^2}{n_0} \right\rfloor$, where $n_0 = 2(d-k) \cdot \left(\frac{qd}{k}\right)^{\frac{k}{d-k}}$,$\mathsf{SP}_{k,\ell}(NW_{q,d,k}) \geq 2^{-O(d)} \cdot M(qd,k) \cdot M(qd,\ell)$.*

▶ Remark 23. An advantage of directly analysing the complexity measures for homogeneous formulas instead of for set-multilinear formulas is that our hard polynomial need not be set multilinear. In the full version of this article [2], we describe an explicit non set-multilinear polynomial P (in VNP) with a large APP measure; the construction is similar to a polynomial in [7]. The proof that APP of P is large is considerably simpler than the proofs of the above lemmas.

4.5 Putting everything together: the low-depth lower bound

▶ **Theorem 24** (Low-depth homogeneous formula lower bound for IMM)**.** *For any d, n, Δ such that $n = \omega(d)$, any homogeneous formula of product-depth at most Δ computing $IMM_{n,d}$ over any field $\mathbb{F}$ has size at least $2^{-O(d)} \cdot n^{\Omega\left(d^{2^{1-\Delta}}\right)}$. In particular, when $d = O(\log n)$, we get a lower bound of $n^{\Omega\left(d^{2^{1-\Delta}}\right)}$.*

▶ **Theorem 25** (Low-depth homogeneous formula lower bound for NW)**.** *Let n, d, Δ be positive integers. If $\Delta = 1$, let $d = n^{1-\epsilon}$ for any constant $\epsilon > 0$ and $k = \left\lfloor \frac{d-1}{2} \right\rfloor$. Otherwise, let $d \leq \frac{1}{150}\left(\frac{\log n}{\log \log n}\right)^2$, let $\tau = \left\lfloor d^{2^{1-\Delta}} \right\rfloor$, $\alpha = \sum_{\nu=0}^{\Delta-1} \frac{(-1)^\nu}{\tau^{2^\nu - 1}}$, and $k = \left\lfloor \frac{\alpha \cdot d}{1+\alpha} \right\rfloor$. In both cases, let q be the largest prime between $\left\lfloor \frac{n}{2d} \right\rfloor$ and $\left\lfloor \frac{n}{d} \right\rfloor$. Then, any homogeneous formula of product-depth at most Δ computing $NW_{q,d,k}$ over any field $\mathbb{F}$ has size at least $2^{-O(d)} \cdot n^{\Omega\left(d^{2^{1-\Delta}}\right)}$. In particular, when $d = O(\log n)$, we get a lower bound of $n^{\Omega\left(d^{2^{1-\Delta}}\right)}$.*

▶ Remark 26. Notice that in the above theorem, as k depends on the product-depth Δ, the polynomial $NW_{q,d,k}$ may be different for different values of Δ. However, much like in [19], there is a way to "stitch" all the different NW polynomials for different values of Δ into a single polynomial P such that any homogeneous formula of product-depth Δ computing P has size at least $2^{-O(d)} n^{\Omega\left(d^{2^{1-\Delta}}\right)}$. See Theorem 34 for more details.

In [29], the authors showed how to convert a circuit of product-depth Δ computing a homogeneous polynomial to a homogeneous formula of product-depth 2Δ without much increase in the size. Combining Lemma 11 from [29] with Theorems 24 and 25, we get:

▶ **Corollary 27** (Low-depth circuit lower bound for IMM). *For any positive integers d, n, Δ such that $n = \omega(d)$, any circuit of product-depth at most Δ computing $IMM_{n,d}$ over any field $\mathbb{F}$ with characteristic 0 or more than d has size at least $2^{-O(d)} \cdot n^{\Omega\left(\frac{d^{2^{1-2\Delta}}}{\Delta}\right)}$.*

In particular, when $d = O(\log n)$, we get a lower bound of $n^{\Omega\left(\frac{d^{2^{1-2\Delta}}}{\Delta}\right)}$.

▶ **Corollary 28** (Low-depth circuit lower bound for NW). *Let n, d, Δ be positive integers. If $\Delta = 1$, let $d = n^{1-\epsilon}$ for any constant $\epsilon > 0$ and $k = \left\lfloor \frac{d-1}{2} \right\rfloor$. Otherwise, let $d \leq \frac{1}{150}\left(\frac{\log n}{\log\log n}\right)^2$, let $\tau = \left\lfloor d^{2^{1-\Delta}} \right\rfloor$, $\alpha = \sum_{\nu=0}^{\Delta-1} \frac{(-1)^\nu}{\tau^{2^\nu - 1}}$, and $k = \left\lfloor \frac{\alpha \cdot d}{1+\alpha} \right\rfloor$. In both cases, let q be the largest prime number between $\left\lfloor \frac{n}{2d} \right\rfloor$ and $\left\lfloor \frac{n}{d} \right\rfloor$. Then, any circuit of product-depth at most Δ computing $NW_{q,d,k}$ over any field $\mathbb{F}$ of characteristic 0 or more than d has size at least $2^{-O(d)} \cdot n^{\Omega\left(\frac{d^{2^{1-2\Delta}}}{\Delta}\right)}$.*

In particular, when $d = O(\log n)$, we get a lower bound of $n^{\Omega\left(\frac{d^{2^{1-2\Delta}}}{\Delta}\right)}$.

We note that our lower bounds quantitatively improve on the original homogeneous formula lower bound of [29] in terms of the dependence on the degree. While [29] gives a lower bound of $d^{O(-d)} \cdot n^{\Omega\left(d^{1/2^\Delta - 1}\right)}$ (as the conversion from homogeneous to set-multilinear formulas increases the size by a factor of $d^{O(d)}$), our lower bound is $2^{-O(d)} \cdot n^{\Omega\left(d^{2^{1-\Delta}}\right)}$. Thus, we get slight improvement both in the multiplicative factor (from $d^{O(d)}$ to $2^{O(d)}$) and in the exponent of n (from $d^{\frac{1}{2^\Delta - 1}}$ to $d^{\frac{1}{2^{(\Delta-1)}}}$). We point out what these improvements mean for smaller depths: For $\Delta = 2$, our lower bound for homogeneous formulas computing IMM is superpolynomial as long as $d \leq \epsilon \cdot \log^2 n$ for a small enough positive constant ϵ, whereas the lower bound in [29] does not work beyond $d = O\left(\left(\frac{\log n}{\log\log n}\right)^2\right)$. In particular, we obtain a lower bound of $n^{\Omega(\log n)}$ for the size of homogeneous depth-5 formulas computing $IMM_{n,d}$ when $d = \Theta(\log^2 n)$. Finally, for $\Delta = 3$ and $d \leq \epsilon \cdot \log^{4/3} n$, we get a lower bound of $n^{\Omega\left(d^{1/4}\right)}$, as compared to $n^{\Omega\left(d^{1/7}\right)}$ from [29].

5 Lower bound for unique-parse-tree formulas

In this section, we show that UPT formulas computing IMM must have a "large" size. We begin by giving a decomposition for such formulas. Missing proofs from this section can be found in the full version of this article [2].

5.1 Decomposition of UPT formulas

In order to upper bound the SP (or APP) measure of a UPT formula, we need certain results about binary trees and UPT formulas. For a given canonical parse tree $\mathcal{T}$ with d leaves, we define its *degree sequence* $(d_1, \ldots, d_t)$ using the function Deg-seq described in Algorithm 1.

We prove the following lemma in the full version of this article [2]. The idea here is to "break" the tree at various nodes so that the successive sizes of the smaller trees are far from each other.

Algorithm 1 Degree sequence of a right-heavy binary tree.

function $\textsc{Deg-seq}(\mathcal{T})$
    $v_0 \leftarrow$ root node of $\mathcal{T}$.
    if v_0 is a leaf **then**
        return (1).
    end if
    $d \leftarrow \text{leaves}(v_0)$, $i \leftarrow 0$.
    while v_i is not a leaf **do**
        $v_{i+1} \leftarrow$ right child of v_i, $i \leftarrow i+1$.
    end while
    $v \leftarrow v_j$ corresponding to the largest index j such that $\text{leaves}(v_j) > \frac{d}{3}$.
    $d_1 \leftarrow d - \text{leaves}(v)$.
    return $(d_1, \textsc{Deg-seq}(\mathcal{T}_v))$.
end function

▶ **Lemma 29.** *For a given canonical parse tree $\mathcal{T}$ with $d \geq 1$ leaves, let $(d_1, \ldots, d_t) := \textsc{Deg-seq}(\mathcal{T})$, where the function* $\textsc{Deg-seq}$ *is given in Algorithm 1. Also let $e_i := d - \sum_{j=1}^{i} d_j$ for $i \in [t]$ and $e_0 := d$. Then, for all $i \in [t-1]$, $e_i \in \left(\frac{e_{i-1}}{3}, \frac{2 \cdot e_{i-1}}{3}\right]$. Additionally, $d_t = 1$, $e_t = 0$, and $\log_3 d + 1 \leq t \leq \log_{3/2} d + 1$.*

As mentioned in Section 4.1, it was shown in [12] that a homogeneous formula can be expressed as a "small" sum of products of homogeneous polynomials such that in each summand, the degrees of the factors roughly form a geometric sequence. We observe that this result can be strengthened for UPT formulas; in particular, we show that for UPT formulas, the "degree sequences" of all the summands are identical.

▶ **Lemma 30** (Log-product decomposition of UPT formulas). *Let $f \in \mathbb{F}[\mathbf{x}]$ be a homogeneous polynomial of degree $d \geq 1$ computed by a UPT formula C with canonical parse tree $\mathcal{T}(C)$. Let $(d_1, \ldots, d_t) := \textsc{Deg-seq}(\mathcal{T}(C))$. Then there exist an integer $s \leq \text{size}(C)$ and homogeneous polynomials $\{Q_{i,j}\}_{i,j}$ where $\deg(Q_{i,j}) = d_j$ for $i \in [s]$, $j \in [t]$, such that*

$$f = \sum_{i=1}^{s} Q_{i,1} \cdots Q_{i,t}.$$

5.2 UPT formulas have high residue

Now we show that there exists a value of k that has high residue with respect to the degrees of the factors given by the above log-product lemma.

▶ **Lemma 31** (High residue for a degree sequence). *For any given canonical parse tree $\mathcal{T}$ with $d \geq 1$ leaves, let $(d_1, \ldots, d_t) := \textsc{Deg-seq}(\mathcal{T})$ and $k := \textsc{Upt-K}(d_1, \ldots, d_t)$ where the function* $\textsc{Upt-K}$ *is described in Algorithm 2. Then*

$$\text{residue}_k(d_1, \ldots, d_t) \geq \frac{\log_3 d - 10}{216}.$$

Algorithm 2 The value of k for a given sequence of degrees.

function UPT-K$(d_1, \ldots, d_t)$
  /* Returns k which shall be the order of derivatives for the SP and APP measures. */
  $d = d_1 + \cdots + d_t$.
  for $i \in [0..t]$ **do**
    $e_i \leftarrow d - \sum_{j=1}^{i} d_j$.
  end for
  $m \leftarrow \left\lfloor \frac{\log_3 d - 1}{3} \right\rfloor$.
  /* Defining a function $\mathcal{J} : [3m] \to [t-2]$. */
  for $i \in [3m]$ **do**
    $\mathcal{J}(i) \leftarrow \min \left\{ j \in [0..t] : e_j \leq 3^i \right\}$.
  end for
  $(a_1, \ldots, a_m) \leftarrow$ *undefined*.
  for $i \in [m]$ **do**
    $j \leftarrow \mathcal{J}(3i)$.
    $b_0 \leftarrow \left(\sum_{p=1}^{i-1} \frac{a_p}{3^{3p}} \right) \cdot d_{j+1}$.
    /* b_1 defined below is not used in the algorithm but will be useful in the analysis. */
    $b_1 \leftarrow \left(\sum_{p=1}^{i-1} \frac{a_p}{3^{3p}} + \frac{1}{3^{3i}} \right) \cdot d_{j+1}$.
    if $\{b_0\} \in \left[\frac{1}{18}, \frac{17}{18} \right]$ **then**
      $a_i \leftarrow 0$.
    else
      $a_i \leftarrow 1$.
    end if
  end for
  $\alpha \leftarrow \sum_{p=1}^{m} \frac{a_p}{3^{3p}}$
  $k \leftarrow \lfloor \alpha \cdot d \rfloor$
  return k.
end function

5.3 Putting everything together: the UPT formula lower bound

In this section, we state our lower bounds for UPT formulas.

▶ **Theorem 32** (UPT formula lower bound for IMM). *For $n \in \mathbb{N}$ and $d \leq \epsilon \cdot \log n \cdot \log \log n$, where $\epsilon > 0$ is a small enough constant, any UPT formula computing $IMM_{n,d}$ over any field $\mathbb{F}$ has size $n^{\Omega(\log d)}$.*

▶ Remark 33. The above theorem can also be derived by using the complexity measure studied in [29] along with the observation that the *unbounded-depth* set-multilinearization due to [35] (which increases the size by a factor of $2^{O(d)}$) preserves parse trees.

We also get an analogous theorem for a polynomial related to the NW polynomial.

▶ **Theorem 34.** *Let $n \in \mathbb{N}$, $d \leq \epsilon \cdot \log n \cdot \log \log n$, where $\epsilon > 0$ is a small enough constant, and q be the largest prime number between $\left\lfloor \frac{n}{2d} \right\rfloor$ and $\left\lfloor \frac{n}{d} \right\rfloor$. Then, any UPT formula computing $P = \sum_{i=\lfloor d/30 \rfloor}^{\lceil d/2 \rceil} y_i \cdot NW_{q,d,i}$ (where the y variables are distinct from the $\mathbf{x}$ variables), over any field $\mathbb{F}$ has size $n^{\Omega(\log d)}$.*

6 Conclusion

Recently, [29] made remarkable progress on arithmetic circuit lower bounds by giving the first super-polynomial lower bound for low-depth formulas. They achieve this by a hardness escalation approach via set-multilinearization. But, set-multilinearization is an inherently expensive process that seems to restrict us from obtaining an exponential lower bound for even homogeneous low-depth formulas. In this work, we take the vital first step of sidestepping set-multilinearization and showing a super-polynomial lower bound for low-depth formulas via a direct approach. A direct approach does not seem to incur an *inherent* exponential loss. So, it might be possible to prove stronger lower bounds for low-depth homogeneous formulas or other related models using this approach or an adaptation of it.

Problem 1. Prove exponential lower bounds for low-depth homogeneous arithmetic formulas. Prove exponential lower bounds for low-depth, *multi-r-ic* formulas.

A formula is said to be multi-r-ic, if the formal degree of every gate with respect to every variable is at most r [17, 21]. The UPT formula lower bound proved in this work is for formulas computing polynomials of degree at most $O(\log n \cdot \log\log n)$. It would be interesting to increase the range of degrees for which our bound works. In the non-commutative setting, exponential lower bounds are known for formulas with exponentially many parse trees [28].

Problem 2. Prove an $n^{\Omega(\log d)}$ lower bound for UPT formulas for $d = n^{O(1)}$. Prove a superpolynomial lower bound for formulas with "many" parse trees.

Our work also raises the prospect of learning low-depth homogeneous formulas given black-box access using the "learning from lower bounds" paradigm proposed in [7, 18].

Problem 3. Obtain learning algorithms for random low-depth homogeneous formulas.

To upper bound SP or APP of a homogeneous formula C, we first show in Section 3 that the space of partial derivatives of C has some structure and then exploit this structure using shifts or affine projections. There might be a better way to exploit this structure, say by going modulo an appropriately chosen ideal or using random restrictions along with shifts as done in [14, 26]. Exploring this possibility is also an interesting direction for future work.

References

1 Manindra Agrawal and V. Vinay. Arithmetic Circuits: A Chasm at Depth Four. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, October 25-28, 2008, Philadelphia, PA, USA*, pages 67–75. IEEE Computer Society, 2008.

2 Prashanth Amireddy, Ankit Garg, Neeraj Kayal, Chandan Saha, and Bhargav Thankey. Low-depth arithmetic circuit lower bounds via shifted partials. *Electron. Colloquium Comput. Complex.*, TR22-151, 2022. URL: `https://eccc.weizmann.ac.il/report/2022/151`.

3 Suryajith Chillara, Nutan Limaye, and Srikanth Srinivasan. Small-depth multilinear formula lower bounds for iterated matrix multiplication with applications. *SIAM J. Comput.*, 48(1):70–92, 2019. Conference version appeared in the proceedings of STACS 2018. `doi:10.1137/18M1191567`.

4 Michael A. Forbes, Mrinal Kumar, and Ramprasad Saptharishi. Functional lower bounds for arithmetic circuits and connections to boolean circuit complexity. In *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, pages 33:1–33:19, 2016.

5 Lance Fortnow and Adam R. Klivans. Efficient learning algorithms yield circuit lower bounds. *J. Comput. Syst. Sci.*, 75(1):27–36, 2009.

6 Hervé Fournier, Nutan Limaye, Guillaume Malod, and Srikanth Srinivasan. Lower Bounds for Depth-4 Formulas Computing Iterated Matrix Multiplication. *SIAM J. Comput.*, 44(5):1173–1201, 2015. Conference version appeared in the proceedings of STOC 2014.

7 Ankit Garg, Neeraj Kayal, and Chandan Saha. Learning sums of powers of low-degree polynomials in the non-degenerate case. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 889–899. IEEE, 2020.

8 Fulvio Gesmundo and Joseph M. Landsberg. Explicit polynomial sequences with maximal spaces of partial derivatives and a question of k. mulmuley. *Theory Comput.*, 15:1–24, 2019. `doi:10.4086/toc.2019.v015a003`.

9 Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Approaching the Chasm at Depth Four. *J. ACM*, 61(6):33:1–33:16, 2014. Conference version appeared in the proceedings of CCC 2013.

10 Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Arithmetic Circuits: A Chasm at Depth 3. *SIAM J. Comput.*, 45(3):1064–1079, 2016. Conference version appeared in the proceedings of FOCS 2013.

11 Nikhil Gupta, Chandan Saha, and Bhargav Thankey. A super-quadratic lower bound for depth four arithmetic circuits. In Shubhangi Saraf, editor, *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. `doi:10.4230/LIPIcs.CCC.2020.23`.

12 Pavel Hrubes and Amir Yehudayoff. Homogeneous formulas and symmetric polynomials. *Comput. Complex.*, 20(3):559–578, 2011. `doi:10.1007/s00037-011-0007-3`.

13 Neeraj Kayal. An exponential lower bound for the sum of powers of bounded degree polynomials. *Electronic Colloquium on Computational Complexity (ECCC)*, 19:81, 2012. URL: `http://eccc.hpi-web.de/report/2012/081`.

14 Neeraj Kayal, Nutan Limaye, Chandan Saha, and Srikanth Srinivasan. An Exponential Lower Bound for Homogeneous Depth Four Arithmetic Formulas. *SIAM J. Comput.*, 46(1):307–335, 2017. Conference version appeared in the proceedings of FOCS 2014.

15 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):2:1–2:27, 2020. Conference version appeared in the proceedings of STACS 2016.

16 Neeraj Kayal and Chandan Saha. Lower Bounds for Depth-Three Arithmetic Circuits with small bottom fanin. *Computational Complexity*, 25(2):419–454, 2016. Conference version appeared in the proceedings of CCC 2015.

17 Neeraj Kayal and Chandan Saha. Multi-k-ic depth three circuit lower bound. *Theory Comput. Syst.*, 61(4):1237–1251, 2017. The conference version appeared in the proceedings of STACS, 2015. `doi:10.1007/s00224-016-9742-9`.

18 Neeraj Kayal and Chandan Saha. Reconstruction of non-degenerate homogeneous depth three circuits. 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 413–424. ACM, 2019.

19 Neeraj Kayal, Chandan Saha, and Ramprasad Saptharishi. A super-polynomial lower bound for regular arithmetic formulas. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 146–153, 2014.

20 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, July 11-15, 2016, Rome, Italy*, pages 33:1–33:15, 2016.

21 Neeraj Kayal, Chandan Saha, and Sébastien Tavenas. On the size of homogeneous and of depth four formulas with low individual degree. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 626–632, 2016.

22 Pascal Koiran. Arithmetic circuits: The chasm at depth four gets wider. *Theor. Comput. Sci.*, 448:56–65, 2012.

23 Mrinal Kumar and Ramprasad Saptharishi. The computational power of depth five arithmetic circuits. *SIAM J. Comput.*, 48(1):144–180, 2019. doi:10.1137/18M1173319.

24 Mrinal Kumar and Shubhangi Saraf. The limits of depth reduction for arithmetic formulas: it's all about the top fan-in. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 136–145, 2014.

25 Mrinal Kumar and Shubhangi Saraf. Superpolynomial lower bounds for general homogeneous depth 4 arithmetic circuits. In *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part I*, pages 751–762, 2014.

26 Mrinal Kumar and Shubhangi Saraf. On the Power of Homogeneous Depth 4 Arithmetic Circuits. *SIAM J. Comput.*, 46(1):336–387, 2017. Conference version appeared in the proceedings of FOCS 2014.

27 Deepanshu Kush and Shubhangi Saraf. Improved low-depth set-multilinear circuit lower bounds. In Shachar Lovett, editor, *37th Computational Complexity Conference, CCC 2022, July 20-23, 2022, Philadelphia, PA, USA*, volume 234 of *LIPIcs*, pages 38:1–38:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.CCC.2022.38.

28 Guillaume Lagarde, Nutan Limaye, and Srikanth Srinivasan. Lower Bounds and PIT for Non-commutative Arithmetic Circuits with Restricted Parse Trees. *Computational Complexity*, 28(3):471–542, 2019.

29 Nutan Limaye, Srikanth Srinivasan, and Sébastien Tavenas. Superpolynomial Lower Bounds Against Low-Depth Algebraic Circuits. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 804–814. IEEE, 2021. A full version of the paper can be found at https://eccc.weizmann.ac.il/report/2021/081.

30 Nutan Limaye, Srikanth Srinivasan, and Sébastien Tavenas. On the partial derivative method applied to lopsided set-multilinear polynomials. In Shachar Lovett, editor, *37th Computational Complexity Conference, CCC 2022, July 20-23, 2022, Philadelphia, PA, USA*, volume 234 of *LIPIcs*, pages 32:1–32:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.CCC.2022.32.

31 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.

32 Noam Nisan and Avi Wigderson. Lower Bounds on Arithmetic Circuits Via Partial Derivatives. *Computational Complexity*, 6(3):217–234, 1997. Conference version appeared in the proceedings of FOCS 1995.

33 Ran Raz. On the Complexity of Matrix Product. *SIAM J. Comput.*, 32(5):1356–1369, 2003. Conference version appeared in the proceedings of STOC 2002.

34 Ran Raz. Separation of Multilinear Circuit and Formula Size. *Theory of Computing*, 2(6):121–135, 2006. Conference version appeared in the proceedings of FOCS 2004.

35 Ran Raz. Tensor-Rank and Lower Bounds for Arithmetic Formulas. *J. ACM*, 60(6):40:1–40:15, 2013. Conference version appeared in the proceedings of STOC 2010.

36 Ran Raz and Amir Yehudayoff. Lower Bounds and Separations for Constant Depth Multilinear Circuits. *Computational Complexity*, 18(2):171–207, 2009. Conference version appeared in the proceedings of CCC 2008.

37 Bruce Reznick. Sums of even powers of real linear forms. *Memoirs of the AMS*, 96:463, 1992.

38 Amir Shpilka and Avi Wigderson. Depth-3 arithmetic circuits over fields of characteristic zero. *Computational Complexity*, 10(1):1–27, 2001. Conference version appeared in the proceedings of CCC 1999.

39 Sébastien Tavenas. Improved bounds for reduction to depth 4 and depth 3. *Inf. Comput.*, 240:2–11, 2015. Conference version appeared in the proceedings of MFCS 2013.

40 Leslie G. Valiant. Completeness Classes in Algebra. In *Proceedings of the 11h Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1979, Atlanta, Georgia, USA*, pages 249–261, 1979.

41 Leslie G. Valiant, Sven Skyum, S. Berkowitz, and Charles Rackoff. Fast Parallel Computation of Polynomials Using Few Processors. *SIAM J. Comput.*, 12(4):641–644, 1983.

42 Ilya Volkovich. A Guide to Learning Arithmetic Circuits. In *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*, pages 1540–1561, 2016.