Low-depth Arithmetic Circuit Lower Bounds: Bypassing Set-multilinearization

Prashanth Amireddy*
Harvard University
pamireddy@g.harvard.edu

Ankit Garg
Microsoft Research India
garga@microsoft.com

Neeraj Kayal
Microsoft Research India
neeraka@microsoft.com

Chandan Saha†
Indian Institute of Science
chandan@iisc.ac.in

Bhargav Thankey‡
Indian Institute of Science
thankeyd@iisc.ac.in

Abstract

A recent breakthrough work of Limaye, Srinivasan and Tavenas [LST21] 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 [KSS14, FLMS15].

*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.

†Partially supported by a MATRICS grant of the Science and Engineering Research Board, DST, India.

‡Supported by the Prime Minister's Research Fellowship, India.

Contents

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 [Val79] 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 [VSBR83, AV08, Koi12, Tav15, GKKS16]. 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 [NW97], 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 [Kay12, GKKS14, KSS14, FLMS15, KS14b, KLSS17, KS17b]. There was limited progress for higher-depth formulas, and lower bounds remained open even for depth-5 formulas. In a recent breakthrough work, [LST21] 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 1.1. Prove superpolynomial lower bounds for *general* formulas or even homogeneous formulas. (A formula is homogeneous if every gate computes a homogeneous polynomial.)

Open Problem 1.2. 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 [LST21] 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 (see Section B.3.2). [LST21] calls such reductions 'hardness escalation'. Second, they use an interesting adaptation of the rank of the partial derivatives matrix measure [Nis91] 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 [LST21] 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 [LST21]). 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 [LST21] to get an *exponential* lower bound for low-depth set-multilinear formulas (Appendix F). 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 [KLSS17, KS17b] 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 Remark 4.8 for more details. The hardness escalation approach of [LST21] can not yield a lower bound for such polynomials. 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) [NW97, SW01], the rank of the partial derivatives matrix measure (a.k.a. evaluation dimension) [Nis91, Raz06, RY09], the shifted partials measure (SP) and its variants [GKKS14, KSS14, KLSS17], the affine projections of partials measure (APP) [GKS20, KNS20], etc. All these measures are defined for *any* polynomial, which is not necessarily set-multilinear. Whereas the relrk measure used in [LST21], 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 [FK09, Vol16, KS19a, GKS20]. 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 exponen-

tial 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: $\mathsf{SP}_{k,\ell}(f) := \dim\langle \mathbf{x}^{\ell} \cdot \partial^{k}(f)\rangle$, 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 1.3 (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 1.4.

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 4.4).

2. The above lower bound is slightly better than the bound of [LST21]. 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 [LST21]). 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 [Raz13]), 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, [KSS14] considered the model of *regular* arithmetic formulas. These are formulas which consist of alternating layers of addition ($+$) and multiplication ($\times$) 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. [KSS14]

proved a superpolynomial lower bound on the size of regular formulas for an explicit polynomial and later [FLMS15] 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 ($\times$) 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 $\log n$.

Theorem 1.5. *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 1.6.

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. For example, see Figure 1a for a UPT formula of product-depth 2 that computes a degree 3 polynomial.

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. [GL19, Rez92] 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 [KS14a, KS19b] 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 [KLSS17, KS17b] 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 [LST21]. Fortunately, [KS14a, KS19b] 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 [GKKS14] Section 2.1) breaks down completely when k is so large (see Appendix G). Despite these apparent hurdles, and to our surprise, we overcome these challenges and are able to use SP with

[1] The model in [KSS14, FLMS15] 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.

$k \approx \frac{d}{2}$ 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 [LST21]. 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.

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, $\mu_{\mathcal{L}}\left(\prod_{j=1}^{t} Q_j\right) \leq \binom{t}{k} \ll \binom{n+k-1}{k}$ which is the value for a generic f (for $k \leq t/2$). This is the basis of the homogeneous depth-3 formula lower bound in [NW97].

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), [GKKS14, Kay12] 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 $k \geq t$ (see Appendix G). Due to this, progress remain stalled for higher depth arithmetic circuit lower bounds via SP.

In a major breakthrough, [LST21] 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 $\mu_{\mathcal{L}}\left(\prod_{j=1}^{t} Q_j(\mathbf{x}_{S_j})\right) \leq \frac{n^{|A|}}{2^{\frac{1}{2}\sum_{j=1}^{t} \mathrm{imbalance}_j}}$.

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 $\sum_{j=1}^{t} \mathrm{imbalance}_j$. [LST21] observe that this quantity is *somewhat large* for the depth-4 circuits that they consider.

[2] Some major results in the area such as [Raz03, LST21] 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.

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 $S := \{j : |\alpha_j| \leq k \deg(Q_j)/d\}$. 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 $k_0 + \frac{k}{d-k}\ell_0 \leq k$. 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 1.7. Let k_0 and ℓ_0 be defined as above. Then, $k_0 + \frac{k}{d-k}\ell_0 \leq k - \mathsf{residue}_k(d_1, \ldots, d_t)$, 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 [LST21] 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 1.8.

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

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

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

Thus we can upper bound,

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

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 4.1) where the degree sequences are carefully chosen so that that the residues can be simultaneously lower bounded for all the terms (Lemma 4.2). While in a different context, these calculations do bear similarity with related calculations in [LST21].

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 4.8). 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 5.2. We further observe (Lemma 5.3) 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 1.5.

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 [LST21]. 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 [KS17b, KLSS17, KS16]. 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. For an exhaustive set of notations and definitions, see Appendix B.

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 2.1 (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 2.2 (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 $\mathsf{APP}_{k,n_0}(P) := \max_{L:\mathbf{x} \to \langle \mathbf{z} \rangle} \dim \left\langle \pi_L\left(\boldsymbol{\partial}^k P\right) \right\rangle$.

SP and APP are *sub-additive*. APP is related to the *skewed partials* and relrk measures used in [KNS20] and [LST21], respectively. For a comparison, see Remark B.7 and Section B.5 in Appendix B.

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

Definition 2.3. 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 defer to Appendix B.

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 [LST21].

Definition 2.4 (Word polynomial $P_{\mathbf{w}}$ [LST21]). 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 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 [LST21] 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 [LST21] – we say that $\mathbf{w}$ is *h-unbiased* if $\max_{i \in [d]} |w_1 + \cdots + w_i| \leq h$.

Lemma 2.5 (Lemma 7 in [LST21]). Let $\mathbf{w} \in [-h..h]^d$ be *h-unbiased*. If for some $n \geq 2^h$, $IMM_{n,d}$ 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, de-

[3] Our definition for UPT formulas is more general than the model considered in a recent paper by Limaye, Srinivasan and Tavenas [LST22] 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 $i \neq j$.

[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 [LST21] 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.

noted 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 [NW97, KSS14, KLSS17, KS17b, KS16, KST16a, KST16b, FKS16, CLS19, KS19b, GST20, LST21, KS22].

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 Appendix C.

Lemma 3.1 (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,

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

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

Lemma 3.2 (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 $d = d_1 + \cdots + d_t \ge 1$, where Q_i is homogeneous and $d_i := \deg(Q_i)$ for $i \in [t]$. Then, for any non-negative integers $k < d$, $\ell \ge 0$, and $n_0 \le n$,

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

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

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 Appendix D.

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 [HY11]. In the decomposition in [HY11], 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 [LST21] 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 4.1 (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 $s \leq \text{size}(C)$, and
2. for all $i \in [s]$, either
$$\left|\{j \in [t_i] : \deg(Q_{i,j}) = 1\}\right| \geq d^{2^{1-\Delta}}, \text{ or}$$
$$\left|\left\{j \in [t_i] : \deg(Q_{i,j}) \approx_2 d^{2^{1-\delta}}\right\}\right| \geq d^{2^{1-\delta}} - 1 \text{ , for some } \delta \in [2..\Delta].$$

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 residue. Its proof uses Lemma 4.1.

Lemma 4.2 (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 $s \leq \text{size}(C)$. Fixing an arbitrary $i \in [s]$, let $t := t_i$ and define $d_j := \deg(Q_{i,j})$ for $j \in [t]$. Then, $\text{residue}_k(d_1, \ldots, d_t) \geq \Omega\left(d^{2^{1-\Delta}}\right)$, 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., $M(n,k) \cdot M(n,\ell)$. In the lemma below, we derive a bound for such polynomials. Explicit examples of such polynomials are given in Section 4.4.

Lemma 4.3 (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 4.4 (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 $d \leq n_0 \approx 2(d-k) \cdot \left(\frac{n}{k}\right)^{\frac{k}{d-k}}$. 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 $s \geq 2^{-O(d)} \cdot \left(\frac{n}{d}\right)^{\Omega(\gamma)}$.

Remark 4.5. 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 discussion in Section D.10 for more details.

4.4 The hard polynomials

We shall prove our lower bound for the word polynomial $P_{\mathbf{w}}$ introduced in [LST21] 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 4.6 ($P_{\mathbf{w}}$ 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 4.7 (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 $\lfloor \frac{n}{2d} \rfloor$ and $\lfloor \frac{n}{d} \rfloor$. 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 4.8. 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 Appendix D.7, we describe an explicit non set-multilinear polynomial P (in VNP) with a large APP measure; the construction is similar to a polynomial in [GKS20]. 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 4.9 (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 4.10 (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 4.11. 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 [KSS14], 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 5.6 for more details.

In [LST21], 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 [LST21] with Theorems 4.9 and 4.10, we get:

Corollary 4.12 (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 4.13 (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 [LST21] in terms of the dependence on the degree. While [LST21] 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 [LST21] 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 [LST21].

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 Appendix E.

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 2. Informally, we consider a sequence[7] of subtrees of $\mathcal{T}$ (i.e., each element in this sequence is a subtree of the previous element) and associate d_i's to the number of leaves between consecutive subtrees in that sequence: in particular we have $\sum_i d_i = d$.

We prove the following lemma in Section E.1. 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.

Lemma 5.1. For a given canonical parse tree $\mathcal{T}$ with $d \geq 1$ leaves, let $(d_1, \ldots, d_t) := \text{DEG-SEQ}(\mathcal{T})$, where the function DEG-SEQ is given in Algorithm 2. 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 [HY11] 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.

[7] the precise way of constructing this sequence is deferred to Algorithm 2.

Lemma 5.2 (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 5.3 (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 Upt-K is described in Algorithm 3. Then

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

5.3 Putting everything together: the UPT formula lower bound

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

Theorem 5.4 (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 5.5. The above theorem can also be derived by using the complexity measure studied in [LST21] along with the observation that the *unbounded-depth* set-multilinearization due to [Raz13] (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 5.6. *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 $\lfloor \frac{n}{2d} \rfloor$ and $\lfloor \frac{n}{d} \rfloor$. 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, [LST21] 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 [KS17a, KST16b]. 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 [LLS19].

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 [GKS20, KS19a].

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 [KLSS17, KS17b]. Exploring this possibility is also an interesting direction for future work.

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

References

[AKV18] Noga Alon, Mrinal Kumar, and Ben Lee Volk. Unbalancing Sets and an Almost Quadratic Lower Bound for Syntactically Multilinear Arithmetic Circuits. In Rocco A. Servedio, editor, *33rd Computational Complexity Conference, CCC 2018, June 22-24, 2018, San Diego, CA, USA*, volume 102 of *LIPIcs*, pages 11:1–11:16. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2018. 22

[AV08] 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. 1

[BDS22] C. S. Bhargav, Sagnik Dutta, and Nitin Saxena. Improved lower bound, and proof barrier, for constant depth algebraic circuits. In Stefan Szeider, Robert Ganian, and Alexandra Silva, editors, *47th International Symposium on Mathematical Foundations of*

Computer Science, MFCS 2022, August 22-26, 2022, Vienna, Austria, volume 241 of *LIPIcs*, pages 18:1–18:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. 22, 23

[BLS16] Nikhil Balaji, Nutan Limaye, and Srikanth Srinivasan. An almost cubic lower bound for ΣΠΣ circuits computing a polynomial in VP. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:143, 2016. 22

[BS83] Walter Baur and Volker Strassen. The Complexity of Partial Derivatives. *Theor. Comput. Sci.*, 22:317–330, 1983. 21

[CELS18] Suryajith Chillara, Christian Engels, Nutan Limaye, and Srikanth Srinivasan. A Near-Optimal Depth-Hierarchy Theorem for Small-Depth Multilinear Circuits. In Mikkel Thorup, editor, *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*, pages 934–945. IEEE Computer Society, 2018. 22, 64

[CKSV22] Prerona Chatterjee, Mrinal Kumar, Adrian She, and Ben Lee Volk. Quadratic lower bounds for algebraic branching programs and formulas. *Comput. Complex.*, 31(2):8, 2022. Conference version appeared in the proceedings of CCC 2020. 21

[CLS19] 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. 10, 22, 33, 64

[DMPY12] Zeev Dvir, Guillaume Malod, Sylvain Perifel, and Amir Yehudayoff. Separating multilinear branching programs and formulas. In Howard J. Karloff and Toniann Pitassi, editors, *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 615–624. ACM, 2012. 22

[FK09] Lance Fortnow and Adam R. Klivans. Efficient learning algorithms yield circuit lower bounds. *J. Comput. Syst. Sci.*, 75(1):27–36, 2009. 2

[FKS16] 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. 10, 33

[FLMS15] 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. 1, 4, 22, 23

[GKKS14] 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. 1, 2, 4, 5, 22, 26, 66

[GKKS16] 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. 1

[GKS20] 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. 2, 13, 16, 26, 27, 66

[GL19] 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. 4

[GST20] 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. 10, 22, 33

[HY11] Pavel Hrubes and Amir Yehudayoff. Homogeneous formulas and symmetric polynomials. *Comput. Complex.*, 20(3):559–578, 2011. 11, 14

[Kal85] K. Kalorkoti. A Lower Bound for the Formula Size of Rational Functions. *SIAM J. Comput.*, 14(3):678–687, 1985. 21

[Kay12] Neeraj Kayal. An exponential lower bound for the sum of powers of bounded degree polynomials. *Electronic Colloquium on Computational Complexity (ECCC)*, 19:81, 2012. 1, 5, 22, 26, 33

[KLSS17] 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. 1, 2, 4, 7, 10, 16, 22, 33

[KNS20] 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. 2, 9, 26, 27, 34

[Koi12] Pascal Koiran. Arithmetic circuits: The chasm at depth four gets wider. *Theor. Comput. Sci.*, 448:56–65, 2012. 1

[KS14a] 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. 4

[KS14b] 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. 1

[KS16] 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. 7, 10, 33

[KS17a] 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. 16

[KS17b] 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. 1, 2, 4, 7, 10, 16, 22, 33

[KS19a] 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. 2, 16, 27

[KS19b] Mrinal Kumar and Ramprasad Saptharishi. The computational power of depth five arithmetic circuits. *SIAM J. Comput.*, 48(1):144–180, 2019. 4, 10, 33

[KS22] 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. 10, 22, 33

[KSS14] 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. 1, 2, 3, 4, 10, 13, 22, 23, 29, 33, 66

[KST16a] 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. 10, 22, 33

[KST16b] 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. 10, 16, 33

[LLS19] 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. 16, 23, 29

[LMP19] Guillaume Lagarde, Guillaume Malod, and Sylvain Perifel. Non-commutative computations: lower bounds and polynomial identity testing. *Chicago Journal of Theoretical Computer Science*, (2):1–19, 2019. 23

[LST21] 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. 1, 2, 3, 4, 5, 6, 7, 9, 10, 11, 12, 13, 14, 15, 22, 27, 28, 32, 33, 53, 64, 65, 66

[LST22] 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. 9, 29

[Nis91] Noam Nisan. Lower Bounds for Non-Commutative Computation (Extended Abstract). In Cris Koutsougeras and Jeffrey Scott Vitter, editors, *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, May 5-8, 1991, New Orleans, Louisiana, USA*, pages 410–418. ACM, 1991. 1, 2

[NW97] 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. 1, 2, 5, 10, 22, 33

[Raz03] 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. 5

[Raz06] 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. 2, 22

[Raz10] Ran Raz. Elusive Functions and Lower Bounds for Arithmetic Circuits. *Theory of Computing*, 6(1):135–177, 2010. Conference version appeared in the proceedings of STOC 2008. 22

[Raz13] 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. 3, 15, 28

[Rez92] Bruce Reznick. Sums of even powers of real linear forms. *Memoirs of the AMS*, 96:463, 1992. 4

[RSY08] Ran Raz, Amir Shpilka, and Amir Yehudayoff. A Lower Bound for the Size of Syntactically Multilinear Arithmetic Circuits. *SIAM J. Comput.*, 38(4):1624–1647, 2008. Conference version appeared in the proceedings of FOCS 2007. 22

[RY08] Ran Raz and Amir Yehudayoff. Balancing Syntactically Multilinear Arithmetic Circuits. *Computational Complexity*, 17(4):515–535, 2008. 22

[RY09] 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. 2, 22, 64

[SS97] Victor Shoup and Roman Smolensky. Lower Bounds for Polynomial Evaluation and Interpolation Problems. *Computational Complexity*, 6(4):301–311, 1997. Conference version appeared in the proceedings of FOCS 1991. 22

[Str73a] Volker Strassen. Die berechnungskomplexiät von elementarysymmetrischen funktionen und von iterpolationskoeffizienten. *Numerische Mathematik*, 20:238–251, 1973. 21

[Str73b] Volker Strassen. Vermeidung von divisionen. *The Journal für die Reine und Angewandte Mathematik*, 264:182–202, 1973. 28

[SW01] 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. 2, 22

[Tav15] 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. 1

[TLS22] Sébastien Tavenas, Nutan Limaye, and Srikanth Srinivasan. Set-multilinear and non-commutative formula lower bounds for iterated matrix multiplication. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 416–425. ACM, 2022. 22

[Val79] 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. 1

[Vol16] 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. 2

[VSBR83] 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. 1

[Yau16] Morris Yau. Almost cubic bound for depth three circuits in VP. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:187, 2016. 22

A Other known lower bounds

In this section, we give a brief account of some of the known arithmetic circuit lower bounds that are related to our work.

General circuits and formulas. Not much is known about lower bounds for general arithmetic circuits and formulas computing explicit polynomials. Baur and Strassen [BS83, Str73a] proved that any arithmetic circuit computing the power symmetric polynomial ($\mathrm{PSym}_{n,d} := x_1^d + \cdots + x_n^d$) or the elementary symmetric polynomial ($\mathrm{ESym}_{n,d} := \sum_{S\subseteq[n]:\ |S|=d} \prod_{i\in S} x_i$) must have size $\Omega(n \log d)$. There has been no improvement on this bound and their result continues to be the best-known lower bound for general arithmetic circuits.

The best-known lower bound for general arithmetic formulas is quadratic. [Kal85] proved that any arithmetic formula computing the polynomial $\sum_{i,j\in[n]} x_i^j y_j$ must have size $\Omega(n^2)$. Recently, [CKSV22] proved an $\Omega(n^2)$ lower bound for arithmetic formulas computing $\mathrm{ESym}_{n,0.1n}$.

They also showed that any 'layered' Algebraic Branching Program (ABP) computing $\mathrm{PSym}_{n,n}$ has size $\Omega(n^2)$. ABPs are algebraic analogues of (Boolean) branching programs and, as a model of computation, are known to be at least as powerful as formulas. Because of the apparent difficulty of proving lower bounds for general models of computation, restricted classes of circuits like multilinear, homogeneous, and low-depth circuits have received a lot of attention in the last few decades. We now discuss a few results for these models.

Multilinear and set-multilinear circuits. A circuit or formula is said to be multilinear if every gate in it computes a multilinear polynomial. [RSY08] showed a lower bound of $\Omega(n^{4/3}/\log^2 n)$ for syntactically multilinear circuits. This lower bound was improved to an $\Omega(n^2/\log^2 n)$ bound in [AKV18]. Unlike the case of general circuits and formulas, a super-polynomial separation is known between multilinear circuits and formulas. [Raz06, RY08] proved that there exists a polynomial computable by polynomial-size multilinear circuits but can only be computed by multilinear formulas of size $n^{\Omega(\log n)}$. [DMPY12] showed a similar lower bound but for a polynomial computable by a polynomial-size multilinear ABP.

Exponential lower bounds are known for low-depth multilinear circuits. A lower bound of $2^{n^{\Omega(1/\Delta)}}$ for multilinear circuits of product-depth $\Delta = o(\log n/\log\log n)$ computing the $n \times n$ permanent and determinant was shown in [RY09]. [CLS19] proved a lower bound of $2^{\Omega(\Delta d^{1/\Delta})}$ for multilinear circuits of product-depth at most $\Delta \leq \log d$ computing $IMM_{2,d}$. A quasi-polynomial separation between product-depth Δ multilinear circuits and product-depth $\Delta+1$ multilinear circuits was proved in [RY09] and improved to an exponential separation in [CELS18].

Notice that the lower bounds mentioned in the previous paragraph are of the form $n^{O(1)} \cdot f(d)$ where $f(d)$ is a superpolynomial but sub-exponential function of the degree. Borrowing terminology from parameterised complexity, [LST21] calls such lower bounds FPT lower bounds. As pointed out in [LST21], it is unclear if FPT bounds can be used to prove lower bounds for low-depth circuits. [LST21] and later [BDS22] prove a non-FPT lower bound of $n^{d^{\exp(-\Delta)}}$ for set-multilinear formulas of product-depth Δ computing $IMM_{n,d}$ when $d = O(\log n)$. These lower bounds are then used to prove super-polynomial lower bounds for low-depth circuits. In [TLS22], a non-FPT lower bound of $(\log n)^{\Omega(\Delta d^{1/\Delta})}$ is proved for set-multilinear formulas of product-depth $\Delta = O(\log d)$ computing $IMM_{n,d}$. They also prove a lower bound of $(\log n)^{\Omega(\log d)}$ for any set-multilinear circuit computing $IMM_{n,d}$. Recently, [KS22] proved a lower bound of $n^{\Omega\left(n^{1/\Delta}/\Delta\right)}$ for set-multilinear formulas of product-depth Δ computing the Nisan-Wigderson design polynomial.

Homogeneous and low-depth circuits. [SS97, Raz10] proved a lower bound of $\Omega(\Delta n^{1+1/\Delta})$ for depth Δ circuits with multiple output gates. In a classic work [NW97], Nisan and Wigderson showed that any homogeneous depth 3 circuit computing $\mathrm{ESym}_{n,d}$ has size $n^{\Omega(d)}$. A series of papers [Kay12, GKKS14, KSS14, FLMS15, KLSS17, KS17b] resulted in an $n^{\Omega(\sqrt{d})}$ lower bound for homogeneous depth 4 circuits computing the Nisan-Wigderson design polynomial and $IMM_{n,d}$.

[SW01] proved a quadratic lower bound for depth 3 circuits computing elementary symmetric polynomials of degree $\Omega(n)$. This was improved to an almost cubic lower bound in [KST16a] for a polynomial in VNP. Subsequently, [BLS16, Yau16] proved similar lower bounds for polynomials in VP. [GST20] obtained a lower bound of $\widetilde{\Omega}(n^{2.5})$ for depth 4 circuits computing the Nisan-Wigderson design polynomial. As mentioned before, in a recent breakthrough work [LST21], Limaye, Srinivasan, and Tavenas proved a lower bound of $n^{\Omega(d^{1/(2^\Delta-1)}/\Delta)}$ for product-depth Δ cir-

cuits computing $IMM_{n,d}$, $d = O(\log n)$. [BDS22] improved this to a lower bound of $n^{\Omega(d^{1/\phi^{2\Delta}}/\Delta)}$ where $\phi = (\sqrt{5}+1)/2 \approx 1.618$.

Circuits with a few parse trees. Circuits with a bounded number of parse trees have been studied before in the non-commutative setting [LMP19, LLS19]. [LLS19] proved a lower bound of $n^{d^{1/4}}$ for non-commutative circuits having at most $2^{d^{1/4}}$ parse trees. This was an improvement on [LMP19], which proved a lower bound of $2^{\Omega(n)}$ for non-commutative UPT circuits computing the $n \times n$ permanent and determinant.

The UPT formulas that we study in this work are also related to regular formulas considered in [KSS14, FLMS15]. A formula is said to be regular if it has alternating levels of addition and multiplication gates and all gates at the same level have the same fan-in. Recall that we call a formula product-regular if the fan-ins of the addition gates in a formula are arbitrary, but the multiplication gates at the same level are restricted to having the same fan-ins. It is easy to see that UPT formulas are a generalization of homogeneous product-regular formulas. [KSS14] obtains a lower bound of $n^{\Omega(\log n)}$ for regular formulas computing a polynomial in VNP. [FLMS15] later obtained a lower bound of $n^{\Omega(\log d)}$ for regular formulas computing $IMM_{n,d}$.

B Full Preliminaries

In this section, we describe the algebraic models we are interested in, and the complexity measures and polynomials used for proving lower bounds for those models. We begin by establishing standard notations and terminology.

B.1 Notations

Basics. We will attempt to stick to the following usage of symbols: C, D for circuits; P, Q for polynomials; i, j for indices; d for the degree of a polynomial; s for the size of a formula or a sum-of-products decomposition; k for the order of derivatives and ℓ for the order of shifts; t for the number of polynomials; m or X for monic monomials; α, τ as some real parameters; μ, κ, σ for maps; $\mathcal{S}, \mathcal{T}$ for spaces (sets) of polynomials; and $\mathcal{P}, \mathcal{T}$ for binary trees.

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 define the integer part of a as $\lfloor a \rfloor := \max\{n \in \mathbb{Z} : n \leq a\}$, the ceiling of a as $\lceil a \rceil := \min\{n \in \mathbb{Z} : n \geq a\}$, the fractional part of a as $\{a\} := a - \lfloor a \rfloor$, the nearest integer of a by $\lfloor a \rceil$ (i.e., if $\{a\} \leq 1/2$, $\lfloor a \rceil = \lfloor a \rfloor$ and otherwise $\lfloor a \rceil = \lceil a \rceil$), and the absolute value of a by $|a|$.

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 B.1 (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 $\mathsf{residue}_k(d_1,\ldots,d_t)$ is a real number (but not necessarily an integer) that is at most $\frac{k}{2}$. 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. However, imposing these constraints does not alter the value of residue by much, so we omit them.

Sets and functions. When some sets $S_1,\ldots,S_t$ are pair-wise disjoint, we write their union as $S_1\sqcup\cdots\sqcup S_t$. For a function $\mu : S\to T$ and a subset $A\subseteq S$, we define the multiset $\mu(A) := \{\mu(x) : x\in A\}$. Clearly $|\mu(A)| = |A|$[8]. We denote the power set of a set S by 2^S. We say that a function $\mu : S\to T$ extends $\kappa : A\to T$ if $A\subseteq S$ and for all $x\in A$, $\mu(x) = \kappa(x)$. Let $\mu : S\to T$ and $\kappa : A\to T$ be functions such that $S\cap A = \emptyset$. Then $\mu\sqcup\kappa : S\sqcup A\to T$ is defined by setting $(\mu\sqcup\kappa)(x) = \mu(x)$ for all $x\in S$ and $(\mu\sqcup\kappa)(x) = \kappa(x)$ for all $x\in A$.

Binomial coefficients. For non-negative integers a,b, we shall denote the quantity $\binom{a+b-1}{b}$ by $M(a,b)$. Note that $M(a,b)$ is the number of (monic) monomials of degree b over a many variables.

The following lemma is useful when dealing with binomial coefficients.

Lemma B.2 (Approximations for $M(a,b)$). For positive integers $a\geq b\geq c$ and d, we have

1. $(a/b)^b \leq M(a,b) \leq (6a/b)^b$,
2. $(a/2b)^c \leq \frac{M(a,b+c)}{M(a,b)} \leq (2a/b)^c$,
3. $\frac{M(c,d)}{M(b,d)} \geq (c/b)^d$.

Proof. 1.

$$M(a,b) = \frac{(a+b-1)\cdots(a)}{b!} \geq \frac{a^b}{b^b}\text{, and}$$

$$\begin{aligned} M(a,b) &\leq \frac{(a+b-1)^b}{\left(\frac{b}{e}\right)^b} && \text{(using } b!\geq (b/e)^b\text{)} \\ &\leq \frac{(2a)^b\cdot e^b}{b^b} \leq \left(\frac{6a}{b}\right)^b. \end{aligned}$$

[8] For a multiset B, $|B|$ denotes its size, i.e. the number of elements in B counted with their respective multiplicities.

2.

$$\frac{M(a,b+c)}{M(a,b)} = \left(\frac{a+b+c-1}{b+c}\right)\cdots\left(\frac{a+b}{b+1}\right).$$

The bounds follow from the fact that each of the above c many fractions lies between $\frac{a}{2b}$ and $\frac{2a}{b}$.

3.

$$\frac{M(c,d)}{M(b,d)} = \left(\frac{c+d-1}{b+d-1}\right)\cdots\left(\frac{c}{b}\right).$$

The lower bound follows from the fact that each of the above d many fractions is at least $\frac{c}{b}$. □

Polynomials, derivatives, and affine projections. Let n and n_0 be positive integers. Define variable sets $\mathbf{x} := \{x_1, \ldots, x_n\}$ and $\mathbf{z} := \{z_1, \ldots, z_{n_0}\}$, where $x_1, \ldots, x_n$ and $z_1, \ldots, z_{n_0}$ are distinct variables. Then $\mathbb{F}[\mathbf{x}]$ denotes the set of all multivariate polynomials in $\mathbf{x}$-variables over the field $\mathbb{F}$. The degree of a polynomial $P \in \mathbb{F}[\mathbf{x}]$ is denoted by $\deg(P)$.

There is a natural one-to-one correspondence between monic monomials over $\mathbf{x}$ and multisets over $\mathbf{x}$ obtained by associating the monomial $m = \prod_{i\in[n]} x_i^{e_i}$ with the multiset X containing e_i many copies of x_i for all $i \in [n]$. For a monic monomial m, its corresponding multiset X, and a $P \in \mathbb{F}[\mathbf{x}]$, we define $\partial_m P = \partial_X P \in \mathbb{F}[\mathbf{x}]$ to be the polynomial obtained by successively taking partial derivatives with respect to all the elements of X (the order of elements does not matter). For a function $\mu : A \to \mathbf{x}$, we may simply denote $\partial_{\mu(A)} P$ by $\partial_\mu P$ if the domain of μ is clear from the context. We will use the following facts about partial derivatives.

Proposition B.3 (Sum and product rules of derivatives). Let k be a positive integer and $P, Q, Q_1, \ldots, Q_t \in \mathbb{F}[\mathbf{x}]$. Suppose μ is a function from $[k]$ to $\mathbf{x}$. Then

1. $\partial_\mu(P+Q) = \partial_\mu P + \partial_\mu Q$,
2. $\partial_\mu(Q_1 \cdots Q_t) = \sum_{\kappa:[t]\to 2^{[k]} \text{ s.t. } \bigsqcup_{i\in[t]}\kappa(i)=[k]} \partial_{\mu(\kappa(1))} Q_1 \cdots \partial_{\mu(\kappa(t))} Q_t$.

The product rule above can be obtained by repeatedly using the fact that $\partial_{x_j}(P \cdot Q) = P \cdot \partial_{x_j} Q + Q \cdot \partial_{x_j} P$ for appropriate polynomials P, Q and index j.

For a non-negative integer ℓ, we define

$$\mathbf{x}^\ell := \left\{ {x_1}^{e_1} \cdots {x_n}^{e_n} : e_1, \ldots, e_n \in \mathbb{Z}_{\geq 0} \text{ and } e_1 + \cdots + e_n = \ell \right\}.$$

For a non-negative integer k and $P \in \mathbb{F}[\mathbf{x}]$, we define

$$\boldsymbol{\partial}^k P := \left\{ \partial_m P : m \in \mathbf{x}^k \right\}.$$

For $P \in \mathbb{F}[\mathbf{x}]$, a map $L : \mathbf{x} \to \langle \mathbf{z} \rangle$ and $\mathcal{S} \subseteq \mathbb{F}[\mathbf{x}]$, we define $\pi_L(P) \in \mathbb{F}[\mathbf{z}]$ and $\pi_L(\mathcal{S}) \subseteq \mathbb{F}[\mathbf{z}]$ as

$$\pi_L(P) := P(L(x_1), \ldots, L(x_n)) \text{ and}$$

$$\pi_L(\mathcal{S}) := \{\pi_L(P) : P \in \mathcal{S}\}.$$

We now present some elementary notions regarding polynomials that are needed to formulate our complexity measures.

Spaces of polynomials. For $\mathcal{S}, \mathcal{T} \subseteq \mathbb{F}[\mathbf{x}]$, we define

$$\mathcal{S} \cdot \mathcal{T} := \{P \cdot Q : P \in \mathcal{S} \text{ and } Q \in \mathcal{T}\} \text{ and } \mathcal{S} + \mathcal{T} := \{P + Q : P \in \mathcal{S} \text{ and } Q \in \mathcal{T}\}.$$

For a set of polynomials $\mathcal{S} \subseteq \mathbb{F}[\mathbf{x}]$, 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 a linear combination of some elements in $\mathcal{S}$. That is,

$$\langle \mathcal{S} \rangle := \{P \in \mathbb{F}[\mathbf{x}] : \exists t \geq 0, a_1, \ldots, a_t \in \mathbb{F}, \text{ and } P_1, \ldots, P_t \in \mathcal{S} \text{ such that } P = a_1 \cdot P_1 + \cdots + a_t \cdot P_t\}.$$

For a set of polynomials $\mathcal{S} \subseteq \mathbb{F}[\mathbf{x}]$, its *dimension*, denoted by $\dim \mathcal{S}$, refers to the maximum number of *linearly independent* polynomials in $\mathcal{S}$. It is easy to observe the following relations.

Proposition B.4. For any two sets of polynomials $\mathcal{S}, \mathcal{T} \subseteq \mathbb{F}[\mathbf{x}]$,

1. $0 \in \langle \mathcal{S} \rangle$,
2. $\dim \langle \mathcal{S} \rangle \leq |\mathcal{S}|$,
3. If $\mathcal{S} \subseteq \mathcal{T}$, then $\dim \langle \mathcal{S} \rangle \leq \dim \langle \mathcal{T} \rangle$,
4. $\langle \mathcal{S} + \mathcal{T} \rangle \subseteq \langle \mathcal{S} \rangle + \langle \mathcal{T} \rangle$ and $\dim \langle \mathcal{S} + \mathcal{T} \rangle \leq \dim \langle \mathcal{S} \rangle + \dim \langle \mathcal{T} \rangle$,
5. $\dim \langle \mathcal{S} \cdot \mathcal{T} \rangle \leq \dim \langle \mathcal{S} \rangle \cdot \dim \langle \mathcal{T} \rangle$.

B.2 Complexity measures

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. We remark here that both these measures (with different parameters) have been used in the literature prior to our work – for example, the shifted partials measure in [GKKS14, Kay12] and the affine projections of partials in [GKS20, KNS20].

Definition B.5 (Complexity measures). For a polynomial $P \in \mathbb{F}[\mathbf{x}]$, non-negative integers k, ℓ and $n_0 \in [n]$, we define

- $\mathsf{SP}_{k,\ell}(P) := \dim \left\langle \mathbf{x}^{\ell} \cdot \boldsymbol{\partial}^{k} P \right\rangle$,
- $\mathsf{APP}_{k,n_0}(P) := \max_{L:\mathbf{x} \to \langle \mathbf{z} \rangle} \dim \left\langle \pi_L \left(\boldsymbol{\partial}^{k} P \right) \right\rangle$.

Both the above measures are *sub-additive*; this can be argued using Proposition B.4.

Proposition B.6 (Sub-additivity of the measures). For two polynomials $P, Q \in \mathbb{F}[\mathbf{x}]$, field constants $c_1, c_2 \in \mathbb{F}$, and any parameters k, ℓ, n_0,

1. $\mathsf{SP}_{k,\ell}(c_1 \cdot P + c_2 \cdot Q) \leq \mathsf{SP}_{k,\ell}(P) + \mathsf{SP}_{k,\ell}(Q)$,
2. $\mathsf{APP}_{k,n_0}(c_1 \cdot P + c_2 \cdot Q) \leq \mathsf{APP}_{k,n_0}(P) + \mathsf{APP}_{k,n_0}(Q)$.

Remark B.7. The lower bounds that we prove in this work can also be obtained using the skewed partials measure (SkewP) [KNS20], which is a special case of APP. [KNS20] used the SkewP measure to prove an optimal "non-FPT"[9] lower bound of $n^{\Omega(d)}$ for multilinear depth-3 circuits computing $IMM_{n,d}$. However, we use the more general APP measure for several reasons: Firstly, APP has the geometrically appealing feature that it is invariant under the application of invertible linear transformations on the variables. Secondly, there are models for which APP gives lower bounds but SkewP does not (see Section B.5). The third reason is that for reconstruction of circuits using the recently proposed learning from lower bounds framework [KS19a, GKS20], APP might give weaker non-degeneracy conditions than SkewP. Thus using APP, we might be able to learn more circuits from a circuit class than we can learn using the SkewP measure.

Also, there is a close connection between APP and the relative rank (relrk) measure used in [LST21]: Both of them are variants of evalDim with the added feature of 'imbalance'. It is natural to wonder to what extent the imbalance is required. The relrk measure works with an imbalance between the sizes of the sets involved in a set-multilinear partition. An imbalance or skew between the sizes of variable sets also appears in APP, albeit at a *gross level*: APP uses two sets – one for taking derivatives, the other for affine projections – and there is an imbalance between the sizes of these two sets. Drawing analogy with evalDim, one may also view these two sets as the variables used for evaluations ($\mathbf{y}$) and the remaining variables ($\mathbf{z}$). It turns out that (for set-multilinear polynomials) the "finer" imbalance used in the relrk measure implies an imbalance – at a gross level – between $\mathbf{y}$ and $\mathbf{z}$.[10] One may naturally ask if an imbalance at a gross level, like in APP and its precursor SkewP, is sufficient to prove lower bounds for low-depth circuits.

B.3 Algebraic circuits

In this section, we describe the relevant algebraic models of computation – homogeneous circuits and unique-parse-tree formulas.

B.3.1 Algebraic circuits and formulas

An *algebraic/arithmetic circuit* C is a directed acyclic graph (DAG) whose source nodes (called input gates) are labelled by variables (say, from a set $\mathbf{x}$) or constants from an underlying field $\mathbb{F}$, all other nodes are addition ($+$) or multiplication ($\times$) gates, and edges (called wires) are labeled by field constants. Each gate g in C naturally computes a polynomial in $\mathbb{F}[\mathbf{x}]$ and the polynomial computed by the (unique) sink node (called the output gate) is said to be the polynomial computed by C. If the underlying DAG is a directed rooted tree, the circuit is said to be a *formula*.

Whether C refers to a circuit or the polynomial computed by it is understood from the context. For example, $\text{size}(C)$ refers to the number of gates in the circuit C, whereas $\deg(C)$ refers to the degree of the polynomial computed by C. The *depth* of a circuit is the maximum number of addition and multiplication gates in any path in the underlying DAG, and the *product-depth* is the maximum number of multiplication gates in any path.

Sub-circuits and substitutions. For a gate g in a circuit C, the circuit between[11] the input gates and

[9] Borrowing terminology from [LST21].

[10] [LST21] talks about the (relative) rank of the partial derivatives matrix. The rank of this matrix is evalDim with respect to an appropriate set $\mathbf{y}$.

[11] i.e., the sub-graph of C induced by the nodes that lie on any directed path from a source node to g.

g is called the sub-circuit at g and is denoted by C_g. We will denote by $C_{g \leftarrow y}$ the circuit obtained by replacing g with y. A sub-circuit of a formula is called a *sub-formula*.

A circuit of 'low-depth' can be converted to a formula of the same depth without much blow-up in size. Thus, lower bounds against low-depth formulas also give lower bounds against low-depth circuits. In fact, it is shown in [LST21] that to prove lower bounds against low-depth formulas computing 'low' degree polynomials, it suffices to prove lower bounds against low-depth *homogeneous formulas* (provided the characteristic of the underlying field is large enough). In [LST21], the authors prove lower bounds for low-depth *set-multilinear* formulas which are even restrictive models. We now define these models. Unlike [LST21], we show lower bounds on homogeneous formulas directly without converting them to set-multilinear formulas.

B.3.2 Homogeneous circuits

A polynomial P is said to be *homogeneous* if all its monomials (if any) have the same degree i.e., all its monomials have the same number of variables, counted with repetition. We consider the zero polynomial to be homogeneous. A circuit C is said to be *homogeneous* if each gate g in C computes a homogeneous polynomial. A homogeneous formula is defined analogously. Furthermore, we may assume without loss of generality that all the input gates in a homogeneous circuit or formula are labeled by variables.

If there exists a partition of variables as $\mathbf{x} = \mathbf{x}_1 \sqcup \cdots \sqcup \mathbf{x}_d$ such that all the monomials in a polynomial $P \in \mathbb{F}[\mathbf{x}]$ have exactly one variable from each of the variable sets, then P is said to be *set-multilinear* with respect to the partition $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$. We shall denote the set of all set-multilinear polynomials over $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ by $\mathbb{F}_{sm}[\mathbf{x}_1, \ldots, \mathbf{x}_d]$. A circuit is said to be *set-multilinear* (with respect to $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$) if each gate in it computes a set-multilinear polynomial with respect to a subset of $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$. Observe that a set-multilinear circuit is also a homogeneous circuit.

An arithmetic circuit of size s computing a homogeneous polynomial of degree d can be converted into a homogeneous circuit of size $\mathrm{poly}(s, d)$ computing the same polynomial [Str73b]. An arithmetic formula computing a homogeneous polynomial can also be homogenized; however, the size of the resulting homogeneous formula is $d^{O(d)}\mathrm{poly}(s)$ [Raz13]. Notice that when $d = O(\log s / \log\log s)$ the homogeneous formula has size $\mathrm{poly}(s)$. Unfortunately, the homogenization process in [Raz13] does not preserve the depth of the formula. It can convert a formula of even constant depth to a homogeneous formula of depth as large as $O(\log s)$. Recently [LST21] showed that a product-depth Δ formula of size s computing a homogeneous, degree d polynomial over a field of characteristic 0 or more than d can be converted into an equivalent homogeneous formula of product-depth 2Δ and size $2^{O(\sqrt{d})}\mathrm{poly}(s)$.

Irrespective of the above-mentioned homogenisation results, homogeneous circuits and formulas are a natural model of computation, and also many polynomials for which lower bounds are known are homogeneous. Thus, in this article, we focus on homogeneous circuits and formulas. Further, we work in the regime of 'low' depth circuits and formulas computing 'low' degree polynomials. Since a depth Δ circuit of size s can be converted into an equivalent formula of size $s^{O(\Delta)}$, we shall work with homogeneous formulas (as opposed to homogeneous circuits) for the remainder of the article.

B.3.3 Unique-parse-tree (UPT) formulas

Next, we formalize certain notions about rooted trees and define a subclass of homogeneous formulas which we call *UPT formulas*[12]. For this, we define parse trees of a homogeneous formula; they capture the structure of multiplication gates in the formula.

Definition B.8 (Parse trees of a homogeneous formula). Given a homogeneous formula C computing a degree-d polynomial, we obtain a parse tree $\mathcal{P}$ of C as follows: Let $\widetilde{C}$ be the formula obtained by arbitrarily removing all but one sub-formula feeding into each addition gate. Viewing $\widetilde{C}$ as a rooted directed tree (with edges directed away from the root) with internal nodes being multiplication gates, leaves being variables, and ignoring the addition gates (by bypassing them) as well as the field constants labelling the edges, we get a parse tree $\mathcal{P}$. We also discard the labelling of all the (multiplication) gates and the input gates in $\mathcal{P}$.

Clearly, there are only a finite number of parse trees corresponding to a given formula C and they all have exactly d many leaves[13] as C is homogeneous. For an empty formula (i.e., 0), the empty tree is a parse tree. Substituting some variables to 0 does not affect UPT-ness of a formula (the same way as it does not affect homogeneity).

Definition B.9 (UPT formula). A homogeneous formula C is said to be a *unique-parse-tree (UPT) formula* if all of its parse trees are isomorphic to each other as directed graphs.

It is easy to observe that any sub-formula of a UPT formula is also a UPT formula. Moreover, without increasing the size by much, any UPT formula can be converted into a UPT formula in which all the multiplication gates have fain-in exactly 2 – in other words, all the parse trees are *binary trees*. Henceforth, we will work with this additional structure for UPT formulas.

Formulas with a bounded number of parse trees have been studied before [LLS19] in the non-commutative setting. [LLS19] proved an exponential lower bound for non-commutative circuits having at most exponentially many parse trees. While the lower bound that we prove in this work is only against formulas containing one parse tree, it is in the much more powerful commutative setting. UPT formulas are also related to regular formulas considered in [KSS14]. A formula is said to be regular if it has alternating levels of addition and multiplication gates, and all gates at the same level have the same fan-in. It is easy to see that UPT formulas capture homogeneous formulas wherein the addition gates at the same level can have different fan-ins, and only the multiplication gates at the same level are restricted to having the same fan-ins. Hence UPT formulas are a generalisation of homogeneous regular formulas.

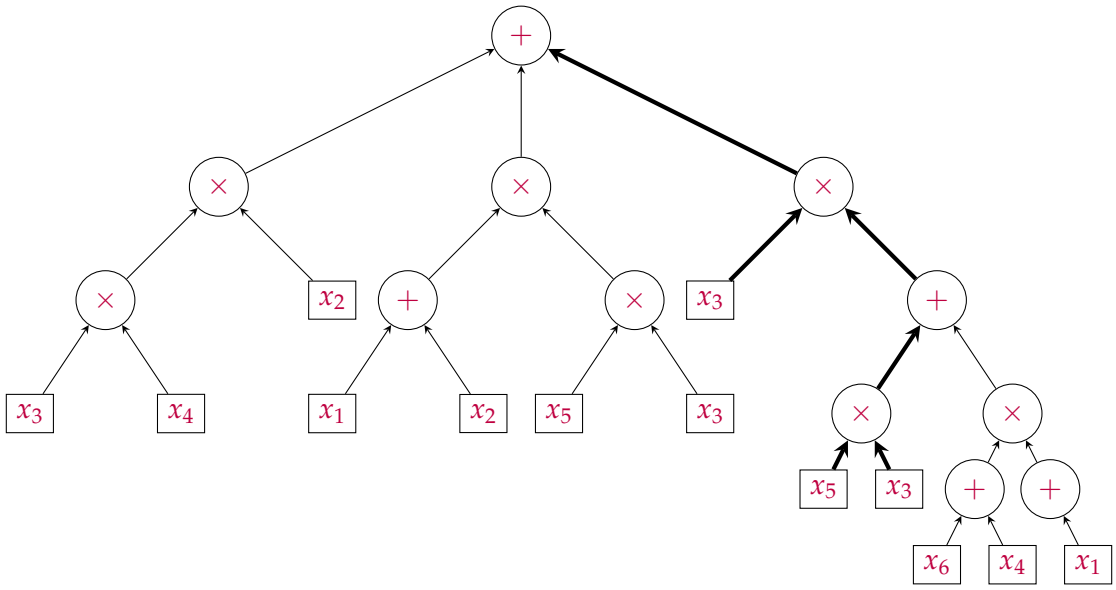
Figure 1 gives an example of a UPT formula and two of its parse trees, which can be seen to be isomorphic to each other.

Binary trees, isomorphism, and canonical trees. Unless stated otherwise, every binary tree $\mathcal{T}$ which we consider in this article will be a rooted, directed (away from the root) binary tree in which all the internal nodes have a *left child* and a *right child*. For a node v of $\mathcal{T}$, $\mathcal{T}_v$ denotes the subtree rooted at v and $\text{leaves}(v)$ denotes the number of leaves in $\mathcal{T}_v$.

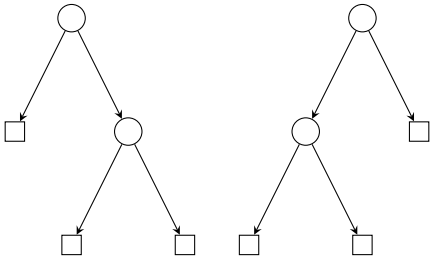
A binary tree $\mathcal{T}$ is said to be *right-heavy* if for all internal nodes v with left child v_L and right child v_R, we have $\text{leaves}(v_L) \leq \text{leaves}(v_R)$. Two binary trees $\mathcal{T}$ and $\widetilde{\mathcal{T}}$ are said to be *equal* or

[12] Our definition for UPT formulas is more general than the model considered in a recent paper by Limaye, Srinivasan and Tavenas [LST22] as we do not impose set-multilinearity.

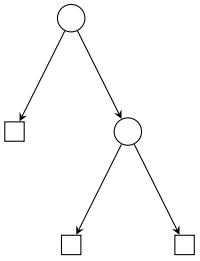
[13] unless C computes the 0 polynomial



(a) A UPT formula C – the dark edges produce a parse tree $\mathcal{P}_1$



(b) Two (isomorphic) parse trees of C: $\mathcal{P}_1$ and $\mathcal{P}_2$



(c) The canonical parse tree $\mathcal{T}(C)$

Figure 1: A UPT formula, its parse trees and canonical parse tree

identical if there is a bijection between their nodes preserving the *(parent, left-child)* and *(parent, right-child)* relations. They are said to be *isomorphic* if there exists a bijection preserving the *(parent, child)* relations.

For any given binary tree $\mathcal{T}$, we define its canonical tree $\text{can}(\mathcal{T})$ using the function in Algorithm 1. For every node v in $\mathcal{T}$, that function swaps the subtrees rooted at its left and right children if the former has more leaves than the latter. The only properties of that function we need are mentioned in the following proposition. These properties can be verified to be true for the parse trees of the formula in Figure 1.

Algorithm 1 Canonical tree of a binary tree

1. **function** $\text{can}(\mathcal{T})$
2. **if** $\mathcal{T}$ is an empty tree **then return** $\mathcal{T}$.
3. **end if**
4. $v \leftarrow$ root node of $\mathcal{T}$.

/* If v has no left (or right) child, v_L (resp. v_R) defined below is treated as an empty node and the corresponding subtree is considered empty. */

5. $v_L \leftarrow$ left child of v.
6. $v_R \leftarrow$ right child of v.

/* Recursively "canonizing" the left and right subtrees. */

7. $\mathcal{T}_{v_L} \leftarrow \text{can}(\mathcal{T}_{v_L})$.
8. $\mathcal{T}_{v_R} \leftarrow \text{can}(\mathcal{T}_{v_R})$.

/* *encoding* is any fixed 1-1 map from binary trees to positive integers. */

9. **if** $\text{leaves}(v_L) > \text{leaves}(v_R)$ or $(\text{leaves}(v_L) = \text{leaves}(v_R)$ and $\text{encoding}(\mathcal{T}_{v_L}) > \text{encoding}(\mathcal{T}_{v_R}))$ **then**

/* The left and right subtrees at the root are swapped. */

10. $\text{swap}(\mathcal{T}_{v_L}, \mathcal{T}_{v_R})$.
11. **end if**
12. **return** $\mathcal{T}$
13. **end function**

Proposition B.10 (Isomorphism means same canonical tree). For any binary trees $\mathcal{T}$ and $\widetilde{\mathcal{T}}$,

1. $\text{can}(\mathcal{T})$ is right-heavy and is isomorphic to $\mathcal{T}$.
2. $\mathcal{T}$ and $\widetilde{\mathcal{T}}$ are isomorphic if and only if $\text{can}(\mathcal{T}) = \text{can}(\widetilde{\mathcal{T}})$. Hence, $\text{can}(\text{can}(\mathcal{T})) = \text{can}(\mathcal{T})$.
3. Let ϕ denote an isomorphism between $\mathcal{T}$ and $\text{can}(\mathcal{T})$. Then for a node v in $\mathcal{T}$, $\text{can}(\mathcal{T}_v) = \text{can}(\mathcal{T})_{\phi(v)}$.

Proof. The lemma is trivially true for empty trees, so we will assume that $\mathcal{T}$ has at least one leaf.

1. In Algorithm 1, we swap the left and right subtrees of a node when the left-subtree has more leaves than the right-subtree and do not swap if the right-subtree has more nodes than the left-subtree. Because of this, it follows from a simple inductive argument that $\text{can}(\mathcal{T})$ is right-heavy and is isomorphic to $\mathcal{T}$.

2. Suppose $\text{can}(\mathcal{T}) = \text{can}(\widetilde{\mathcal{T}})$. Using Item 1, we get that $\mathcal{T}$ and $\widetilde{\mathcal{T}}$ are isomorphic as both of them are isomorphic to $\text{can}(\mathcal{T})$. Conversely, suppose that $\mathcal{T}$ is isomorphic to $\widetilde{\mathcal{T}}$ via a bijection ϕ. We shall use induction on the number of leaves of $\mathcal{T}$ and $\widetilde{\mathcal{T}}$; if both of them have just one leaf, then we trivially have that $\text{can}(\mathcal{T}) = \text{can}(\widetilde{\mathcal{T}})$. Denoting the left and right children of the root v of $\mathcal{T}$ by v_L and v_R, we get that $\mathcal{T}_{v_L}$ and $\widetilde{\mathcal{T}}_{\phi(v_L)}$ are isomorphic under ϕ and so are $\mathcal{T}_{v_R}$ and $\widetilde{\mathcal{T}}_{\phi(v_R)}$. As these trees have fewer leaves than $\mathcal{T}$ and $\widetilde{\mathcal{T}}$, we have $\text{can}(\mathcal{T}_{v_L}) = \text{can}(\widetilde{\mathcal{T}}_{\phi(v_L)})$ and $\text{can}(\mathcal{T}_{v_R}) = \text{can}(\widetilde{\mathcal{T}}_{\phi(v_R)})$ by induction. Hence, after line 10 in the execution of Algorithm 1 on inputs $\mathcal{T}$ and $\widetilde{\mathcal{T}}$, either $\text{can}(\mathcal{T}_{v_L})$ and $\text{can}(\widetilde{\mathcal{T}}_{\phi(v_L)})$ are both the left child of v and $\phi(v)$ respectively, or both are the right child. The same is true for $\text{can}(\mathcal{T}_{v_R})$ and $\text{can}(\widetilde{\mathcal{T}}_{\phi(v_R)})$. Thus $\text{can}(\mathcal{T}) = \text{can}(\widetilde{\mathcal{T}})$.

3. Note that ϕ also induces an isomorphism from $\mathcal{T}_v$ to $\text{can}(\mathcal{T})_{\phi(v)}$. Hence from Item 2, $\text{can}(\mathcal{T}_v) = \text{can}(\text{can}(\mathcal{T})_{\phi(v)})$. By our design of the function can(), note that all subtrees of a canonical tree are themselves canonical trees i.e., there exists a binary tree $\mathcal{T}'$ such that $\text{can}(\mathcal{T}') = \text{can}(\mathcal{T})_{\phi(v)}$. Thus, $\text{can}(\mathcal{T}_v) = \text{can}(\text{can}(\mathcal{T})_{\phi(v)}) = \text{can}(\text{can}(\mathcal{T}')) = \text{can}(\mathcal{T}') = \text{can}(\mathcal{T})_{\phi(v)}$.

□

Definition B.11 (Canonical parse tree). For a UPT formula C, we define its canonical parse tree as $\mathcal{T}(C) := \text{can}(\mathcal{P})$, where $\mathcal{P}$ is an arbitrary parse tree of C. The canonical parse tree is a binary tree and is well-defined as all parse trees of a UPT formula are isomorphic.

B.4 Polynomial families

The polynomials for which we prove the formula lower bounds are the Iterated Matrix Multiplication (IMM) polynomial and the Nisan-Wigderson design polynomial (NW).

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 such that the variables in each matrix and across different matrices are all distinct.

To prove a lower bound for IMM, we analyze the shifted partials (and APP) for a different, but related polynomial $P_{\mathbf{w}}$ that was introduced in [LST21]. This polynomial is parameterized by a word $\mathbf{w} = (w_1, \ldots, w_d)$, a sequence of integers. It was shown in Lemma 22 of [LST21] that $P_{\mathbf{w}}$ is a projection of IMM. Thus a lower bound for formulas computing $P_{\mathbf{w}}$ also gives a lower bound for formulas computing IMM. Both these polynomials can be seen to be homogeneous (in fact, they are set-multilinear), so they can indeed be computed by homogeneous as well as UPT formulas.

Definition B.12 (Word polynomial $P_{\mathbf{w}}$ [LST21]). 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 $i \in [d]$. We call a variable set $\mathbf{x}_i(\mathbf{w})$ 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.[14] 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 – in other words, the variables in X are ordered based on the index of

[14] 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 $i \neq j$.

the corresponding variable sets. 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 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. The word polynomial $P_{\mathbf{w}} \in \mathbb{F}_{sm}[\mathbf{x}(\mathbf{w})]$ for a word $\mathbf{w}$ is defined as

$$P_{\mathbf{w}} := \sum_{\substack{m_+ \in \mathcal{M}_+(\mathbf{w}),\, m_- \in \mathcal{M}_-(\mathbf{w}) \\ \sigma(m_+) \sim \sigma(m_-)}} m_+ \cdot m_-.$$

Notice that if $\sum_{w_i \geq 0} |w_i| \leq \sum_{w_i < 0} |w_i|$, then for any $m_+ \in \mathcal{M}_+$ and $m_- \in \mathcal{M}_-$, $\sigma(m_+)$ will be a prefix of $\sigma(m_-)$ and if $\sum_{i \in [d]: w_i \geq 0} |w_i| > \sum_{i \in [d]: w_i < 0} |w_i|$, then for any $m_+ \in \mathcal{M}_+$ and $m_- \in \mathcal{M}_-$, $\sigma(m_-)$ will be a prefix of $\sigma(m_+)$. We will make use of the following lemma from [LST21] which shows that IMM is at least as hard as $P_{\mathbf{w}}$. For this, we recall the notion of *unbiased*-ness of $\mathbf{w} = (w_1, \ldots, w_d)$ from [LST21] – we say that $\mathbf{w}$ is *h-unbiased* if $\max_{i \in [d]} |w_1 + \cdots + w_i| \leq h$.

Lemma B.13 (Lemma 7 in [LST21])**.** Let $\mathbf{w} \in [-h..h]^d$ be *h-unbiased*. If for some $n \geq 2^h$, $IMM_{n,d}$ has a formula C of product-depth 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, i.e., $\mathcal{T}(C') = \mathcal{T}(C)$.[15]

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 [NW97, KSS14, KLSS17, KS17b, KS16, KST16a, KST16b, FKS16, CLS19, KS19b, GST20, LST21, KS22].

B.5 APP vs skewed partials

In this section, we show that there are some lower bounds that can be proved using APP but not with the skewed partials measure (SkewP). Consider circuits of the form $C = Q_1^{e_1} + \cdots + Q_s^{e_s}$, where $Q_1, \ldots, Q_s$ are arbitrary polynomials of degree at most $d \leq \frac{n}{2e}$. How large an s do we need for C to compute the monomial $P := x_1 \cdots x_n$? [Kay12] introduced the SP measure to show that $s = 2^{\Omega\left(\frac{n}{d}\right)}$. Here, we prove the same using the APP measure. However, this lower bound cannot be proved using SkewP. Let $\mathbb{F}$ be a field of size greater than n. [16]

[15] Although the lemma in [LST21] 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.

[16] This restriction on $\mathbb{F}$ is not required. If $|\mathbb{F}| < n$, we take any extension $\mathbb{K}$ of size more than n, consider C and P over $\mathbb{K}$, and prove the lower bound on s. Observe that the bound will continue to hold over $\mathbb{F}$.

Analysing $\mathsf{SkewP}(P)$**.** The SkewP measure is defined in [KNS20]. If $\mathbf{x} = \mathbf{y} \uplus \mathbf{z}$, then

$$\mathsf{SkewP}_{\mathbf{y},k}(P) := \dim \left\langle \left[\frac{\partial P}{\partial m}\right]_{\mathbf{y}=0} : m \text{ is a monomial of degree } k \text{ in } \mathbf{y} \right\rangle.$$

Observe that $\mathsf{SkewP}_{\mathbf{y},k}(P) \leq 1$ for all $\mathbf{y} \subseteq \mathbf{x}$ and k. Hence, we cannot hope to get a lower bound on s using skewed partials.

Analysing APP **of** C **and** P**.** Let $k = \left\lfloor \frac{n}{2ed} \right\rfloor$ and $n_0 = k + 1$. We begin by proving an upper bound on $\mathsf{APP}_{n_0,k}(C)$.

Claim B.14. $\mathsf{APP}_{n_0,k}(C) \leq s \cdot \binom{n_0+dk-k}{n_0}$.

Proof. Observe that for all $i \in [s]$,

$$\boldsymbol{\partial}^k \left(Q_i^{e_i}\right) \subseteq \left\{ \mathbf{x}^{\leq d(k-1)} Q_i^{\max\{e_i-k,0\}} \right\}.$$[17]

Thus, $\mathsf{APP}_{k,n_0}\left(Q_i^{e_i}\right) \leq \binom{n_0+dk-k}{n_0}$. The claim follows from the sub-additivity of APP. □

We now compute $\mathsf{APP}_{k,n_0}(P)$.

Claim B.15. $\mathsf{APP}_{k,n_0}(P) = \binom{n}{k}$.

Proof. Fix distinct $\alpha_1, \ldots, \alpha_n \in \mathbb{F}$ and let $\mathbf{z} = \{z_1, \ldots, z_{n_0}\}$ be a fresh set of variables. Let L map x_i to $\ell_i(\mathbf{z}) := z_1 + \alpha_i \cdot z_2 + \alpha_i^2 \cdot z_3 + \cdots + \alpha_i^{n_0-1} z_{n_0}$ for all $i \in [n]$. Observe that any $n_0 = k+1$ many linear forms in $\{\ell_1, \ldots, \ell_n\}$ are linearly independent. Now,

$$\boldsymbol{\partial}^k(P) = \left\{ \prod_{i \in S} x_i : S \subseteq [n], |S| = n - k \right\}.$$

We now argue that the polynomials of the set $\pi_L\left(\boldsymbol{\partial}^k(P)\right) = \{\pi_L(\prod_{i\in S} x_i) : S \subseteq [n], |S| = n-k\} = \{\prod_{i\in S} \ell_i : S \subseteq [n], |S| = n-k\}$ are linearly independent; this would prove the claim.

Consider any $\mathbb{F}$-linear combination

$$\sum_{|S|=n-k} \beta_S \cdot \prod_{i \in S} \ell_i = 0$$

and fix an arbitrary S' of size $n-k$. Observe that for every $S \neq S'$, at least one of the linear forms $\{\ell_j : j \notin S'\}$ divides $\prod_{i\in S} \ell_i$. Thus, if I is the ideal generated by $\{\ell_j : j \notin S'\}$, then

$$\beta_{S'} \prod_{i \in S'} \ell_i = 0 \mod I.$$

Since I is an ideal generated by linear forms, $\mathbb{F}[\mathbf{z}]/I$ is a polynomial ring. This implies that $\beta_{S'} = 0$ or $\prod_{i\in S'} \ell_i = 0 \mod I$. The latter is not true: It implies that there exists an $i \in S'$ such that $\ell_i = 0 \mod I$, i.e., ℓ_i is an $\mathbb{F}$-linear combination of $\{\ell_j : j \notin S'\}$. However, as $|[n] \setminus S'| = k$, this contradicts the fact that every $k+1$ linear forms in $\{\ell_1, \ldots, \ell_n\}$ are linearly independent. Thus, $\beta_{S'} = 0$. Repeating this argument for all $S' \subset [n]$ such that $|S'| = n-k$, we get that the elements of $\{\prod_{i\in S} \ell_i : S \subseteq [n], |S| = n-k\}$ are linearly independent. □

[17] Here $\mathbf{x}^{\leq d(k-1)}$ denotes the set of all monomials of degree at most $d(k-1)$.

From the above two claims, and using $k = \left\lfloor \frac{n}{2ed} \right\rfloor$, $n_0 = k+1$, we get,

$$s \geq \frac{\binom{n}{k}}{\binom{n_0+dk-k}{n_0}} \geq \frac{\left(\frac{n}{k}\right)^k}{\left(\frac{e(n_0+kd-k))}{n_0}\right)^{n_0}} \geq \frac{\left(\frac{n}{k}\right)^k}{\left(\frac{e(kd+1)}{k+1}\right)^{k+1}}$$

$$\geq \frac{\left(\frac{n}{k}\right)^k}{n^{O(1)}\left(\frac{n/2}{n/2ed}\right)^k} \geq \frac{1}{n^{O(1)}}\left(\frac{n}{ked}\right)^k \geq \frac{2^k}{n^{O(1)}} \geq 2^{\Omega\left(\frac{n}{d}\right)}.$$

C Proofs from Section 3

C.1 Proof of Lemma 3.1

We first give some intuition about the proof. Let m be any multilinear monomial (i.e., no variable appears more than once) of degree k, and X be the corresponding set of variables. Then, by the product rule $\partial_X(Q_1 \cdots Q_t)$ can be expressed as the sum

$$\sum_{\substack{(X_1,\ldots,X_t):\\ X_1 \sqcup \cdots \sqcup X_t = X}} \partial_{X_1} Q_1 \cdots \partial_{X_t} Q_t. \tag{1}$$

Note that since the sizes of X_i's should sum to k and the degrees of Q_i's should sum to d in each term of the above summation, some factors are differentiated 'a lot' while the others are differentiated only 'a little'. More specifically, if $|X_i| > \frac{k}{d} \cdot d_i$, we use the fact that $\partial_{X_i} Q_i \in \left\langle \mathbf{x}^{d_i - |X_i|} \right\rangle$ and otherwise we use $\partial_{X_i} Q_i \in \left\langle \boldsymbol{\partial}^{|X_i|} Q_i \right\rangle$ to conclude that

$$\partial_{X_1} Q_1 \cdots \partial_{X_t} Q_t \in \left\langle \mathbf{x}^{\sum_{i \in \overline{S}} \ell_{0,i}} \cdot \prod_{i \in S} \boldsymbol{\partial}^{k_{0,i}} Q_i \right\rangle,$$

where $S := \left\{ i \in [t] : |X_i| \leq \frac{k}{d} \cdot d_i \right\}$, $\overline{S} = [t] \setminus S$, $\ell_{0,i} = d_i - |X_i|$ for all $i \in \overline{S}$, and $k_{0,i} = |X_i|$ for all $i \in S$. By the nature of our construction, we can show that $k_0 + \frac{k}{d-k} \cdot \ell_0 \leq k - \mathsf{residue}_k(d_1, \ldots, d_t)$, where $k_0 := \sum_{i \in S} k_{0,i}$ and $\ell_0 := \sum_{i \in \overline{S}} \ell_{0,i}$ (see the calculations at the end of the proof). Now suppose it holds that $\prod_{i \in S} \boldsymbol{\partial}^{k_{0,i}} Q_i = \boldsymbol{\partial}^{k_0} \prod_{i \in S} Q_i$. In such a case, we would get the space required in the R.H.S. of the lemma statement, and we would be done. However, this assumption need not be true if $|S| \geq 1$. To get around this issue, we employ an inductive argument on the size of S (see Claim C.2). For this argument, it will be helpful to combine certain terms in the sum (1) depending on the set of factors that are differentiated a 'lot' (see Claim C.1). We now present the proof in full detail. Since, in general, the variables in m need not be distinct, it will be convenient to think of degree k monomials over $\mathbf{x}$ as maps from $[k]$ to $\mathbf{x}$.

For a function $\mu : P \to \mathbf{x}$ and any $P' \subseteq P$, recall that $\mu(P')$ refers to the multiset of images of the elements of P' under μ. Thus $|\mu(P')| = |P'|$. Let $\mathcal{V}$ be the set of polynomials on the R.H.S., i.e.,

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

We now argue that for an arbitrary total function $\mu : [k] \to \mathbf{x}$, $\partial_{\mu([k])} \left(\prod_{i \in [t]} Q_i \right) \in \mathcal{V}$; the lemma then follows immediately. We use the following identity which is a direct consequence of the product rule for derivatives (Proposition B.3):

$$\partial_{\mu([k])} \left(\prod_{i \in [t]} Q_i \right) = \sum_{\substack{\kappa : [t] \to 2^{[k]} \text{ s.t.} \\ \sqcup_{i \in [t]} \kappa_i = [k]}} \prod_{i \in [t]} \partial_{\mu(\kappa_i)} Q_i.$$

In fact, the product rule yields something general: for any $P \subseteq [k]$, function $\mu : P \to \mathbf{x}$, and $S \subseteq [t]$,

$$\partial_{\mu(P)} \left(\prod_{i \in S} Q_i \right) = \sum_{\substack{\kappa : S \to 2^{P} \text{ s.t.} \\ \sqcup_{i \in S} \kappa_i = P}} \prod_{i \in S} \partial_{\mu(\kappa_i)} Q_i. \tag{2}$$

In the above identities we have used κ_i as a shorthand for $\kappa(i)$; we shall also do so for the rest of this section.

For an arbitrary $S \subseteq [t]$, recall that we denote $\overline{S} = [t] \setminus S$. Let $\widetilde{\kappa} : \overline{S} \to 2^{[k]}$ be such that $|\widetilde{\kappa}_i| > \frac{k}{d} \cdot d_i$ for all $i \in \overline{S}$. Then we define a polynomial $R_{S,\widetilde{\kappa}}$ as

$$R_{S,\widetilde{\kappa}} := \sum_{\substack{\kappa : [t] \to 2^{[k]} \text{ s.t.} \\ \kappa \text{ extends } \widetilde{\kappa} \\ \sqcup_{i \in [t]} \kappa_i = [k] \\ \forall i \in S,\ |\kappa_i| \leq \frac{k}{d} \cdot d_i}} \prod_{i \in [t]} \partial_{\mu(\kappa_i)} Q_i. \tag{3}$$

The idea is to express any k-th order partial derivative of the product $Q_1 \cdots Q_t$ in terms of $R_{S,\widetilde{\kappa}}$. Indeed we have the following claim; its proof (which uses the product rule for derivatives) can be found in Section C.1.1.

Claim C.1.

$$\partial_{\mu([k])} \left(\prod_{i \in [t]} Q_i \right) = \sum_{S \subseteq [t]} \sum_{\substack{\widetilde{\kappa} : \overline{S} \to 2^{[k]} \text{ s.t.} \\ \forall i \in \overline{S}, |\widetilde{\kappa}_i| > \frac{k}{d} \cdot d_i}} R_{S,\widetilde{\kappa}}.$$

Hence, to show that $\partial_{\mu([k])} (Q_1 \cdots Q_t) \in \mathcal{V}$, it suffices to argue that the polynomials $R_{S,\widetilde{\kappa}}$ are in $\mathcal{V}$. We show this by induction on the size of S. In the base case of $|S| = 0$, there does not exist any function $\kappa : [t] \to 2^{[k]}$ that extends $\widetilde{\kappa}$ such that $\left\{ i \in [t] : |\kappa_i| \leq \frac{k}{d} \cdot d_i \right\} = S$ and $\sqcup_{i \in [t]} \kappa_i = [k]$. This is so because $|\kappa_i| = |\widetilde{\kappa}_i| > \frac{k}{d} \cdot d_i$ for all $i \in [t]$ implies that $\sum_{i \in [t]} |\kappa_i| > \sum_{i \in [t]} \frac{k}{d} \cdot d_i = k$, and hence $\sqcup_{i \in [t]} \kappa_i \neq [k]$. So by definition, $R_{S,\widetilde{\kappa}} = 0 \in \mathcal{V}$.

Suppose that $R_{T,\kappa'} \in \mathcal{V}$ for all $T \subseteq [n]$ such that $|T| < |S|$. Let $\widetilde{\kappa} : \overline{S} \to 2^{[k]}$ be any function such that $|\widetilde{\kappa}_i| > \frac{k}{d} \cdot d_i$ for all $i \in \overline{S}$, and let $\kappa : [t] \to 2^{[k]}$ be a function that extends $\widetilde{\kappa}$ such that

$$\sqcup_{i \in [t]} \kappa_i = [k] \text{ and } \left\{ i \in [t] : |\kappa_i| \leq \frac{k}{d} \cdot d_i \right\} = S. \tag{4}$$

Denoting $\sqcup_{i\in\overline{S}}\kappa_i$ by $P_{\overline{S}}$ and $\sqcup_{i\in S}\kappa_i$ by P_S,

$$\begin{aligned}\partial_{\mu(\sqcup_{i\in S}\kappa_i)}\prod_{i\in S}Q_i &= \partial_{\mu(P_S)}\prod_{i\in S}Q_i \\ &= \sum_{\substack{\kappa':S\to 2^{P_S} \text{ s.t.} \\ \sqcup_{i\in S}\kappa'_i = P_S}}\prod_{i\in S}\partial_{\mu(\kappa'_i)}Q_i. \qquad \text{(from Equation (2))}\end{aligned}$$

For $U_{S,\kappa}\in\mathbb{F}[\mathbf{x}]$ defined as $U_{S,\kappa} := \left(\partial_{\mu(P_S)}\prod_{i\in S}Q_i\right)\cdot\prod_{i\in\overline{S}}\partial_{\mu(\kappa_i)}Q_i$, we have the following claim. It is proved in Section C.1.2.

Claim C.2.

$$R_{S,\widetilde{\kappa}} = U_{S,\kappa} - \sum_{\substack{T\subsetneq S \text{ and } \kappa'':S\setminus T\to 2^{P_S} \text{ s.t.} \\ \forall i\in S\setminus T, |\kappa''_i|>\frac{k}{d}\cdot d_i}} R_{T,\kappa''\sqcup\widetilde{\kappa}}.$$

When $T\subsetneq S$, by the induction hypothesis, all the terms $R_{T,\kappa''\sqcup\widetilde{\kappa}}$ in the above expression are in $\mathcal{V}$. Therefore, to conclude that $R_{S,\widetilde{\kappa}}\in\mathcal{V}$, it suffices to show that $U_{S,\kappa}\in\mathcal{V}$. From its definition, note that $U_{S,\kappa}\in\left\langle\partial^{k_0}\left(\prod_{i\in S}Q_i\right)\cdot\mathbf{x}^{\ell_0}\right\rangle$ where $k_0 := |\mu(P_S)| = |P_S| = \sum_{i\in S}|\kappa_i|$ and $\ell_0 := \sum_{i\in\overline{S}}\deg(\partial_{\mu(\kappa_i)}Q_i) = \sum_{i\in\overline{S}}(d_i-|\kappa_i|)$. Also,

$$\begin{aligned}k-k_0-\frac{k}{d-k}\cdot\ell_0 &= k-\sum_{i\in S}|\kappa_i| - \frac{k}{d-k}\cdot\sum_{i\in\overline{S}}(d_i-|\kappa_i|) \\ &= \sum_{i\in\overline{S}}|\kappa_i| - \frac{k}{d-k}\cdot\sum_{i\in\overline{S}}(d_i-|\kappa_i|) \quad \text{(as from (4), } \kappa_1,\dots,\kappa_t \text{ form a partition of } [k]) \\ &= \sum_{i\in\overline{S}}|\kappa_i| - \frac{k}{d-k}\cdot(d_i-|\kappa_i|) \\ &= \sum_{i\in\overline{S}}\frac{d}{d-k}\cdot\left(|\kappa_i|-\frac{k}{d}\cdot d_i\right) \\ &\geq \sum_{i\in\overline{S}}|\kappa_i| - \frac{k}{d}.d_i \qquad \text{(using } d\geq d-k \text{ and } |\kappa_i|>\tfrac{k}{d}\cdot d_i \text{ iff } i\in\overline{S}) \\ &= \frac{1}{2}\left(\sum_{i\in\overline{S}}|\kappa_i|-\frac{k}{d}\cdot d_i\right)+\frac{1}{2}\left(\sum_{i\in S}|\kappa_i|-\frac{k}{d}\cdot d_i\right) \\ &\quad +\frac{1}{2}\left(\sum_{i\in\overline{S}}|\kappa_i|-\frac{k}{d}\cdot d_i\right)-\frac{1}{2}\left(\sum_{i\in S}|\kappa_i|-\frac{k}{d}\cdot d_i\right) \\ &= \frac{1}{2}\left(\sum_{i\in[t]}|\kappa_i|-\frac{k}{d}\cdot d_i\right)+\frac{1}{2}\left(\sum_{i\in\overline{S}}|\kappa_i|-\frac{k}{d}\cdot d_i\right)-\frac{1}{2}\left(\sum_{i\in S}|\kappa_i|-\frac{k}{d}\cdot d_i\right)\end{aligned}$$

$$= \frac{1}{2}\left(k - \frac{k}{d}\cdot d\right) + \frac{1}{2}\left(\sum_{i\in\overline{S}}|\kappa_i| - \frac{k}{d}\cdot d_i\right) - \frac{1}{2}\left(\sum_{i\in S}|\kappa_i| - \frac{k}{d}\cdot d_i\right)$$

(since $|\kappa_i|$'s sum to k and d_i's sum to d)

$$= \frac{1}{2}\cdot\sum_{i\in[t]}\left||\kappa_i| - \frac{k}{d}\cdot d_i\right| \qquad \text{(from (4))}$$

$$\geq \mathsf{residue}_k(d_1,\ldots,d_t). \qquad \text{(from definition of residue)}$$

Hence, $U_{S,\kappa} \in \left\langle \mathbf{x}^{\ell_0}\cdot\boldsymbol{\partial}^{k_0}\left(\prod_{i\in S}Q_i\right)\right\rangle \subseteq \mathcal{V}$ as $k_0 + \frac{k}{d-k}\cdot\ell_0 \leq k - \mathsf{residue}_k(d_1,\ldots,d_t)$. □

C.1.1 Proof of Claim C.1

$$\partial_{\mu([k])}\left(\prod_{i\in[t]}Q_i\right) = \sum_{\substack{\kappa:[t]\to 2^{[k]}\text{ s.t.}\\ \sqcup_{i\in[t]}\kappa_i=[k]}}\prod_{i\in[t]}\partial_{\mu(\kappa_i)}Q_i$$

$$= \sum_{S\subseteq[t]}\sum_{\substack{\kappa:[t]\to 2^{[k]}\text{ s.t.}\\ \sqcup_{i\in[t]}\kappa_i=[k]\\ \left\{i\in[t]:|\kappa_i|\leq\frac{k}{d}.d_i\right\}=S}}\prod_{i\in[t]}\partial_{\mu(\kappa_i)}Q_i$$

$$= \sum_{S\subseteq[t]}\sum_{\substack{\widetilde{\kappa}:\overline{S}\to 2^{[k]}\text{ s.t.}\\ \forall i\in\overline{S},|\widetilde{\kappa}_i|>\frac{k}{d}.d_i}}\sum_{\substack{\kappa':S\to 2^{[k]}\text{ s.t.}\\ \kappa=\kappa'\sqcup\widetilde{\kappa}\\ \sqcup_{i\in[t]}\kappa_i=[k]\\ \forall i\in S,|\kappa'_i|\leq\frac{k}{d}.d_i}}\prod_{i\in[t]}\partial_{\mu(\kappa_i)}Q_i$$

$$= \sum_{S\subseteq[t]}\sum_{\substack{\widetilde{\kappa}:\overline{S}\to 2^{[k]}\text{ s.t.}\\ \forall i\in\overline{S},|\widetilde{\kappa}_i|>\frac{k}{d}.d_i}}\sum_{\substack{\kappa:[t]\to 2^{[k]}\text{ s.t.}\\ \kappa\text{ extends }\widetilde{\kappa}\\ \sqcup_{i\in[t]}\kappa_i=[k]\\ \forall i\in S,|\kappa_i|\leq\frac{k}{d}.d_i}}\prod_{i\in[t]}\partial_{\mu(\kappa_i)}Q_i$$

$$= \sum_{S\subseteq[t]}\sum_{\substack{\widetilde{\kappa}:\overline{S}\to 2^{[k]}\text{ s.t.}\\ \forall i\in\overline{S},|\widetilde{\kappa}_i|>\frac{k}{d}.d_i}} R_{S,\widetilde{\kappa}}. \qquad \text{(by the definition of } R_{S,\widetilde{\kappa}} \text{ in (3))}$$

□

C.1.2 Proof of Claim C.2

$$U_{S,\kappa} = \left(\partial_{\mu(P_S)}\prod_{i\in S}Q_i\right)\cdot\prod_{i\in\overline{S}}\partial_{\mu(\kappa_i)}Q_i$$

$$= \sum_{\substack{\kappa':S\to 2^{P_S}\text{ s.t.}\\ \sqcup_{i\in S}\kappa'_i=P_S}}\prod_{i\in S}\partial_{\mu(\kappa'_i)}Q_i\cdot\prod_{i\in\overline{S}}\partial_{\mu(\kappa_i)}Q_i$$

$$= \sum_{T\subseteq S}\sum_{\substack{\kappa':S\to 2^{P_S}\text{ s.t.}\\ \sqcup_{i\in S}\kappa'_i=P_S\\ \left\{i\in S:|\kappa'_i|\leq\frac{k}{d}.d_i\right\}=T}}\prod_{i\in S}\partial_{\mu(\kappa'_i)}Q_i\cdot\prod_{i\in\overline{S}}\partial_{\mu(\kappa_i)}Q_i \qquad \text{(reordering the summation based on } T\text{)}$$

$$= \sum_{T \subseteq S} \sum_{\substack{\kappa'':S\setminus T \to 2^{P_S} \text{ and } \kappa''':T\to 2^{P_S} \text{ s.t.} \\ \kappa' = \kappa'' \sqcup \kappa''' \\ \sqcup_{i\in S}\kappa'_i = P_S \\ \forall i \in S\setminus T, |\kappa''_i| > \frac{k}{d}\cdot d_i \\ \forall i \in T, |\kappa'''_i| \le \frac{k}{d}\cdot d_i}} \prod_{i\in S} \partial_{\mu(\kappa'_i)} Q_i \cdot \prod_{i\in \overline{S}} \partial_{\mu(\kappa_i)} Q_i$$

$$= \sum_{T\subseteq S} \sum_{\substack{\kappa'':S\setminus T\to 2^{P_S} \text{ s.t.} \\ \forall i\in S\setminus T, |\kappa''_i| > \frac{k}{d}\cdot d_i}} \sum_{\substack{\kappa''':T\to 2^{P_S} \text{ s.t.} \\ \kappa^* = \kappa''\sqcup\kappa'''\sqcup\widetilde{\kappa} \\ \forall i\in T, |\kappa^*_i| \le \frac{k}{d}\cdot d_i \\ \sqcup_{i\in T}\kappa'''_i \sqcup \sqcup_{i\in S\setminus T}\kappa''_i = P_S}} \prod_{i\in[t]} \partial_{\mu(\kappa^*_i)} Q_i$$

(as κ^* extends $\kappa' = \kappa'' \sqcup \kappa'''$ and κ extends $\widetilde{\kappa}$)

$$= \sum_{T\subseteq S} \sum_{\substack{\kappa'':S\setminus T\to 2^{P_S} \text{ s.t.} \\ \forall i\in S\setminus T, |\kappa''_i| > \frac{k}{d}\cdot d_i}} \sum_{\substack{\kappa^*:[t]\to 2^{[k]} \text{ s.t.} \\ \kappa^* = \kappa''\sqcup\kappa'''\sqcup\widetilde{\kappa} \\ \forall i\in T, |\kappa^*_i| \le \frac{k}{d}\cdot d_i \\ \sqcup_{i\in[t]}\kappa^*_i = [k]}} \prod_{i\in[t]} \partial_{\mu(\kappa^*_i)} Q_i$$

(because $\sqcup_{i\in[t]}\kappa^*_i = \left(\sqcup_{i\in S\setminus T}\kappa''_i \sqcup \sqcup_{i\in T}\kappa'''_i\right) \sqcup \sqcup_{i\in\overline{S}}\widetilde{\kappa}_i = P_S \sqcup \sqcup_{i\in\overline{S}}\kappa_i = \sqcup_{i\in[t]}\kappa_i = [k]$ from (4))

$$= \sum_{T\subseteq S} \sum_{\substack{\kappa'':S\setminus T\to 2^{P_S} \text{ s.t.} \\ \forall i\in S\setminus T, |\kappa''_i| > \frac{k}{d}\cdot d_i}} R_{T,\kappa''\sqcup\widetilde{\kappa}}$$

($R_{T,\kappa''\sqcup\widetilde{\kappa}}$ is well-defined because κ^* extends $\kappa''\sqcup\widetilde{\kappa}$ and (4))

$$= R_{S,\widetilde{\kappa}} + \sum_{\substack{T\subsetneq S \text{ and } \kappa'':S\setminus T\to 2^{P_S} \text{ s.t.} \\ \forall i\in S\setminus T, |\kappa''_i| > \frac{k}{d}.d_i}} R_{T,\kappa''\sqcup\widetilde{\kappa}}. \qquad \text{(separating out the case } T = S\text{)}$$

□

C.2 Proof of Lemma 3.2

We will first upper bound the shifted partials measure. From Lemma 3.1, we know that

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

Hence,

$$\left\langle \mathbf{x}^{\ell}\cdot\partial^k \left(Q_1 \cdots Q_t\right) \right\rangle \subseteq \sum_{\substack{S\subseteq[t];\, k_0,\ell_0\ge 0 \\ k_0 + \frac{k}{d-k}\cdot\ell_0 \le k - \mathsf{residue}_k(d_1,\ldots,d_t)}} \left\langle \mathbf{x}^{\ell_0+\ell}\cdot\partial^{k_0}\left(\prod_{i\in S} Q_i\right)\right\rangle. \tag{5}$$

For a fixed $S \subseteq [t]$ and k_0, ℓ_0, since $\left\langle \mathbf{x}^{\ell_0+\ell} \cdot \boldsymbol{\partial}^{k_0} \left(\prod_{i \in S} Q_i \right) \right\rangle \subseteq \left\langle \mathbf{x}^{\ell_0+\ell} \right\rangle \cdot \left\langle \boldsymbol{\partial}^{k_0} \cdot \left(\prod_{i \in S} Q_i \right) \right\rangle$, and $\dim \left\langle \mathbf{x}^{\ell_0+\ell} \right\rangle \leq \left| \mathbf{x}^{\ell_0+\ell} \right| = M(n, \ell_0+\ell)$ and $\dim \left\langle \boldsymbol{\partial}^{k_0} \left(\prod_{i \in S} Q_i \right) \right\rangle \leq \left| \boldsymbol{\partial}^{k_0} \left(\prod_{i \in S} Q_i \right) \right| \leq \left| \mathbf{x}^{k_0} \right| = M(n, k_0)$, we have,

$$\dim \left\langle \mathbf{x}^{\ell_0+\ell} \cdot \boldsymbol{\partial}^{k_0} \left(\prod_{i \in S} Q_i \right) \right\rangle \leq \dim \left\langle \mathbf{x}^{\ell_0+\ell} \right\rangle \cdot \dim \left\langle \boldsymbol{\partial}^{k_0} \left(\prod_{i \in S} Q_i \right) \right\rangle \leq M(n, \ell_0+\ell) \cdot M(n, k_0).$$

Adding up the above upper bound over all the $2^t \cdot d^2$ possible combinations of $S \subseteq [t]$, $k_0 \in [0..k]$, and $\ell_0 \in [0..(d-k)]$ in (5), we get,

$$\mathsf{SP}_{k,\ell}(Q) = \dim \left\langle \mathbf{x}^{\ell} \cdot \boldsymbol{\partial}^{k} \left(Q_1 \cdots Q_t \right) \right\rangle \leq 2^t \cdot d^2 \cdot \max_{\substack{k_0, \ell_0 \geq 0 \\ k_0 + \frac{k}{d-k} \cdot \ell_0 \leq k - \mathsf{residue}_k(d_1, \ldots, d_t)}} M(n, k_0) \cdot M(n, \ell_0 + \ell).$$

The details for an upper bound on APP are similar.

$$\begin{aligned}
\mathsf{APP}_{k,n_0}(Q) &= \max_{L: \mathbf{x} \to \langle \mathbf{z} \rangle} \dim \left\langle \pi_L \left(\boldsymbol{\partial}^k (Q_1 \cdots Q_t) \right) \right\rangle \\
&\leq \max_{L: \mathbf{x} \to \langle \mathbf{z} \rangle} \dim \left\langle \pi_L \left(\sum_{\substack{S \subseteq [t];\ 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)}} \left\langle \mathbf{x}^{\ell_0} \cdot \boldsymbol{\partial}^{k_0} \left(\prod_{i \in S} Q_i \right) \right\rangle \right) \right\rangle \\
&\qquad \text{(from Lemma 3.1)} \\
&\leq \max_{L: \mathbf{x} \to \langle \mathbf{z} \rangle} \dim \left\langle \sum_{\substack{S \subseteq [t];\ 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)}} \left\langle \pi_L \left(\mathbf{x}^{\ell_0} \cdot \boldsymbol{\partial}^{k_0} \left(\prod_{i \in S} Q_i \right) \right) \right\rangle \right\rangle \\
&\qquad \text{(as } \pi_L \text{ distributes over addition)} \\
&\leq \max_{L: \mathbf{x} \to \langle \mathbf{z} \rangle} \sum_{\substack{S \subseteq [t];\ 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)}} \dim \left\langle \pi_L \left(\mathbf{x}^{\ell_0} \right) \cdot \pi_L \left(\boldsymbol{\partial}^{k_0} \left(\prod_{i \in S} Q_i \right) \right) \right\rangle \\
&\qquad \text{(using Proposition B.4 (Item 4) and } \pi_L \text{ distributes over multiplication)} \\
&\leq \max_{L: \mathbf{x} \to \langle \mathbf{z} \rangle} \sum_{\substack{S \subseteq [t];\ 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)}} \dim \left\langle \pi_L \left(\mathbf{x}^{\ell_0} \right) \right\rangle \cdot \dim \left\langle \pi_L \left(\boldsymbol{\partial}^{k_0} \left(\prod_{i \in S} Q_i \right) \right) \right\rangle \\
&\qquad \text{(from Proposition B.4 (Item 5))} \\
&\leq \max_{L: \mathbf{x} \to \langle \mathbf{z} \rangle} \sum_{\substack{S \subseteq [t];\ 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)}} \left| \pi_L \left(\mathbf{x}^{\ell_0} \right) \right| \cdot \left| \pi_L \left(\boldsymbol{\partial}^{k_0} \left(\prod_{i \in S} Q_i \right) \right) \right| \\
&\leq \max_{L: \mathbf{x} \to \langle \mathbf{z} \rangle} \sum_{\substack{S \subseteq [t];\ 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)}} \left| \mathbf{z}^{\ell_0} \right| \cdot \left| \pi_L \left(\boldsymbol{\partial}^{k_0} \left(\prod_{i \in S} Q_i \right) \right) \right| \\
&\qquad \text{(as } L \text{ is a map from } \mathbf{x} \text{ to } \langle \mathbf{z} \rangle, \pi_L(m) \in \mathbf{z}^{\ell_0} \text{ for any monomial } m \text{ over } \mathbf{x} \text{ of degree } \ell_0)
\end{aligned}$$

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

□

D Proofs from Section 4

D.1 Proof of Lemma 4.1

The decomposition is constructed inductively – at addition gates, we simply add the decompositions of the smaller sub-formulas, whereas the multiplication gates need to be handled more carefully. Consider a multiplication gate $Q_1 \times \cdots \times Q_t$. If all the factors ($Q_i$'s) have 'low' degrees, we use this expression directly to construct the decomposition. Otherwise, we go deeper into a factor which has a 'large' degree, but do not expand the other factors. The thresholds to decide whether a factor is of 'low' degree may appear arbitrary (and are indeed so) for this lemma, but we fix them to be $d^{2^{1-\delta}}$ for $\delta \in [2..\Delta]$ as these give us the desired lower bounds.

Without loss of generality, we may assume that C has alternate layers of addition and multiplication gates. Further, we can assume that the degrees of the polynomials computed by all the multiplication gates that feed into an addition gate are the same as the degree of the polynomial computed by that addition gate. This is so because disconnecting all the multiplication gates that compute polynomials of other degrees does not affect the polynomial computed by the addition gate. Also, for brevity, we will ignore the edge weights in C, i.e., we assume that all the field constants on the edges are equal to 1. As scaling with constant factors does not affect the homogeneity of polynomials, this is a valid assumption. Let

$$C = \sum_{i=1}^{u} C_i \text{ , and for } i \in [u],\ C_i = \prod_{j=1}^{u_i} C_{i,j},$$

where u and $\{u_i\}_i$ are integers and $\{C_{i,j}\}_{i,j}$ are (homogeneous) sub-formulas of C of product-depth $\Delta - 1$. The proof of this lemma is by induction on the product-depth. For $\Delta = 1$, for all $i \in [u]$, we have $u_i \geq d$ and for all $j \in [d]$, $\deg(C_{i,j}) = 1$, so both the conditions in the lemma statement are met for $Q_{i,j} := C_{i,j}$.

Suppose that the lemma is true for all homogeneous formulas of product-depth at most $\Delta - 1$, $\Delta \geq 2$. For a formula C with product-depth Δ, we consider the term $C_{i,1} \cdots C_{i,u_i}$ for an arbitrary $i \in [u]$ and analyze the following two cases.

Case 1: There exists some $j^* \in [u_i]$ such that $\deg(C_{i,j^*}) \geq \sqrt{d}$. As the product-depth of C_{i,j^*} is at most $\Delta - 1$, we have the following expression for the polynomial computed by C_{i,j^*} from the induction hypothesis:

$$C_{i,j^*} = \sum_{\tilde{i}=1}^{\tilde{s}_i} \widetilde{Q}_{i,\tilde{i},1} \cdots \widetilde{Q}_{i,\tilde{i},t_{\tilde{i}}}\,, \tag{6}$$

where

$$\tilde{s}_i \leq \mathrm{size}(C_{i,j^*}) \leq \mathrm{size}(C_i), \tag{7}$$

and $\left\{\widetilde{Q}_{i,\tilde{i},\tilde{j}}\right\}_{i,\tilde{i},\tilde{j}}$ are homogeneous polynomials such that for all $\tilde{i} \in \tilde{s}_i$, either

$$\left|\left\{\tilde{j} \in [\tilde{t}_{\tilde{i}}] : \deg(\widetilde{Q}_{i,\tilde{i},\tilde{j}}) = 1\right\}\right| \geq \sqrt{d}^{2^{1-(\Delta-1)}} = d^{2^{1-\Delta}}, \text{or} \tag{8}$$

$$\left|\left\{\tilde{j} \in [\tilde{t}_{\tilde{i}}] : \deg(\widetilde{Q}_{i,\tilde{i},\tilde{j}}) \approx_2 \sqrt{d}^{2^{1-\delta}}\right\}\right| \geq \sqrt{d}^{2^{1-\delta}} - 1 \text{ , for some } \delta \in [2..(\Delta-1)]. \tag{9}$$

Note that since $\sqrt{d}^{2^{1-\delta}} = d^{2^{1-(\delta+1)}}$, (9) is equivalent to

$$\left|\left\{\tilde{j} \in [\tilde{t}_{\tilde{i}}] : \deg(\widetilde{Q}_{i,\tilde{i},\tilde{j}}) \approx_2 d^{2^{1-\delta}}\right\}\right| \geq d^{2^{1-\delta}} - 1 \text{ , for some } \delta \in [3..\Delta]. \tag{10}$$

Indeed, when $\Delta = 2$, the above scenario never arises and the number of linear factors is 'large', i.e., (8) holds. Denoting $\prod_{j \in [u_i] \setminus \{j^*\}} C_{i,j}$ by D_{i,j^*} and using (6), we have

$$C_i = C_{i,1} \cdots C_{i,u_i} = C_{i,j^*}.D_{i,j^*} = \sum_{\tilde{i}=1}^{\tilde{s}_i} \widetilde{Q}_{i,\tilde{i},1} \cdots \widetilde{Q}_{i,\tilde{i},\tilde{t}_{\tilde{i}}} \cdot D_{i,j^*}. \tag{11}$$

Thus, we are able to decompose the sub-formula C_i as a sum of at most size(C_i) many products.

Case 2: For all $j \in [u_i]$, $\deg(C_{i,j}) < \sqrt{d}$. Consider the polynomials computed by $C_{i,1}, \ldots, C_{i,u_i}$. Suppose there exists $j_1 \neq j_2 \in [u_i]$ such that $\deg(C_{i,j_1}) < \frac{\sqrt{d}}{2}$ and $\deg(C_{i,j_2}) < \frac{\sqrt{d}}{2}$. Then $\deg(C_{i,j_1} \cdot C_{i,j_2}) < \sqrt{d}$. By repeatedly combining such low degree factors, we can express $C_i = C_{i,1} \cdots C_{i,u_i}$ as

$$C_i = D_{i,1} \cdots D_{i,v_i}, \tag{12}$$

where $\{D_{i,j}\}_{i,j}$ are homogeneous polynomials such that for all $j \in [v_i]$, we have $\deg(D_{i,j}) < \sqrt{d}$ and there exists at most one index $j^* \in [v_i]$ such that $\deg(D_{i,j^*}) < \frac{\sqrt{d}}{2}$. In other words, for at least $v_i - 1$ indices $j \in [v_i]$, $\deg(D_{i,j}) \approx_2 \sqrt{d}$. Using the fact that C is a homogeneous formula,

$$d \leq \deg(C) = \deg(C_i) = \sum_{j=1}^{v_i} \deg(D_{i,j}) \leq v_i \cdot \sqrt{d}.$$

Therefore, the number of indices $j \in [v_i]$ such that $\deg(D_{i,j}) \approx_2 \sqrt{d}$ is at least $v_i - 1 \geq \sqrt{d} - 1$. In other words,

$$\left|\left\{j \in [v_i] : \deg(D_{i,j}) \approx_2 d^{2^{1-\delta}}\right\}\right| \geq d^{2^{1-\delta}} - 1 \text{ , for } \delta = 2. \tag{13}$$

Now, expressing C_i for each $i \in [u]$ using (11) if i falls under Case 1, and using (12) if i falls under Case 2, we get

$$C = \sum_{i=1}^{u} C_i = \sum_{i=1}^{s} Q_{i,1} \cdots Q_{i,t_i},$$

for polynomials $\{Q_{i,j}\}_{i,j}$ that are defined appropriately based on $\left\{\widetilde{Q}_{i,\tilde{i},\tilde{j}}\right\}_{i,\tilde{i},\tilde{j}}$ and $\{D_{i,j}\}_{i,j}$. Using (7) and (12), we get that the number of terms is $s \leq \sum_{i=1}^{u} \text{size}(C_i) \leq \text{size}(C)$. Item 2 in the lemma statement directly follows from (8), (13), or (10). □

D.2 Proof of Lemma 4.2

We will show that the decomposition proven in Lemma 4.1 itself satisfies the required property. We first establish a range for the value of k (and α) given in the lemma statement. We have $\alpha \leq 1$ and

$$\alpha \geq \sum_{\nu=0}^{1} \frac{(-1)^\nu}{\tau^{2^\nu - 1}} = 1 - \frac{1}{\tau} = 1 - \frac{1}{\lfloor d^{2^{1-\Delta}} \rfloor} \geq \frac{1}{2}.$$

Hence, $k \in \left[\left\lfloor \frac{d}{3} \right\rfloor, \frac{d}{2}\right] \subseteq \left[\frac{d}{4}, \frac{d}{2}\right]$ because $d = \omega(1)$. As C computes a polynomial of degree $d \geq \tau^{2^{\Delta-1}}$, we can apply Item 2 of Lemma 4.1 to C using $\tau^{2^{\Delta-1}}$ (rather than d) as the threshold. Thus, we have that at least one of the following two cases will hold.

Case 1: $\left|\{j \in [t] : d_j = 1\}\right| \geq \left(\tau^{2^{\Delta-1}}\right)^{2^{1-\Delta}} = \tau$. Then,

$$\begin{aligned}
\mathsf{residue}_k(d_1, \ldots, d_t) &= \frac{1}{2} \cdot \min_{k_1,\ldots,k_t \in \mathbb{Z}} \sum_{j \in [t]} \left| k_j - \frac{k}{d} \cdot d_j \right| \\
&\geq \frac{1}{2} \cdot \sum_{j \in [t]} \min\left\{ \left\{ \frac{k}{d} \cdot d_j \right\}, 1 - \left\{ \frac{k}{d} \cdot d_j \right\} \right\} \\
&\geq \frac{1}{2} \cdot \sum_{j \in [t] : d_j = 1} \min\left\{ \left\{ \frac{k}{d} \cdot d_j \right\}, 1 - \left\{ \frac{k}{d} \cdot d_j \right\} \right\} \\
&\geq \frac{1}{2} \cdot \left|\{j \in [t] : d_j = 1\}\right| \cdot \min\left\{ \left\{ \frac{k}{d} \right\}, 1 - \left\{ \frac{k}{d} \right\} \right\} \\
&\geq \tau/8. && \text{(as } k/d \in [1/4, 1/2]\text{)}
\end{aligned}$$

Case 2: $\left|\left\{j \in [t] : d_j \approx_2 \left(\tau^{2^{\Delta-1}}\right)^{2^{1-\delta}}\right\}\right| \geq \left(\tau^{2^{\Delta-1}}\right)^{2^{1-\delta}} - 1$ for some $\delta \in [2..\Delta]$ (this case cannot occur when $\Delta < 2$). Equivalently, there exists a $\delta \in [0..(\Delta-2)]$ such that

$$\left|\left\{j \in [t] : d_j \approx_2 \tau^{2^\delta}\right\}\right| \geq \tau^{2^\delta} - 1.$$

Let $k_1, \ldots, k_t$ be arbitrary non-negative integers such that $k_j \leq d_j$ for all $j \in [t]$. Then for any $j \in [t]$ such that $d_j \approx_2 \tau^{2^\delta}$, we have

$$\begin{aligned}
\tau^{2^\delta - 1} \cdot \left| k_j - \frac{k \cdot d_j}{d} \right| &= \tau^{2^\delta - 1} \cdot \left| k_j - \frac{d_j}{d} \cdot \left\lfloor \frac{\alpha \cdot d}{1+\alpha} \right\rfloor \right| \\
&\geq \tau^{2^\delta - 1} \cdot \left(\left| k_j - \frac{d_j}{d} \cdot \frac{\alpha \cdot d}{1+\alpha} \right| - \frac{d_j}{d} \cdot \left\{ \frac{\alpha \cdot d}{1+\alpha} \right\} \right) \\
&\geq \tau^{2^\delta - 1} \cdot \left| k_j - \frac{\alpha \cdot d_j}{1+\alpha} \right| - \tau^{2^\delta - 1} \cdot \frac{d_j}{d} \\
&\geq \tau^{2^\delta - 1} \cdot \left| k_j - \frac{\alpha \cdot d_j}{1+\alpha} \right| - \frac{\tau^{2^\delta - 1} \cdot \tau^{2^\delta}}{d} && \text{(since } d_j \approx_2 \tau^{2^\delta}\text{)}
\end{aligned}$$

$$
\begin{aligned}
&\geq \tau^{2^\delta-1}\cdot\left|k_j-\frac{\alpha\cdot d_j}{1+\alpha}\right|-\frac{\left(d^{2^{1-\Delta}}\right)^{2^{\delta+1}-1}}{d}\\
&\geq \tau^{2^\delta-1}\cdot\left|k_j-\frac{\alpha\cdot d_j}{1+\alpha}\right|-\frac{1}{d^{2^{1-\Delta}}} && (\text{as } \delta\leq\Delta-2)\\
&= \tau^{2^\delta-1}\cdot\left|k_j-\frac{\alpha\cdot d_j}{1+\alpha}\right|-o(1)\\
& && (\text{if } d^{2^{1-\Delta}}=O(1), \text{ then the lemma is not interesting})\\
&\geq \frac{1}{2}\cdot\tau^{2^\delta-1}\cdot\left|k_j-\alpha\cdot(d_j-k_j)\right|-o(1) && (\text{as } \alpha\leq 1) \quad (14)
\end{aligned}
$$

We use the following claim which is proved in Section D.2.1 below. For $j\in[t]$, let $m_j := d_j-k_j$, note that m_j is a non-negative integer.

Claim D.1. $\eta := \tau^{2^\delta-1}\cdot\left|k_j-\alpha\cdot m_j\right|\geq\Omega(1)$.

Let $k_1,\ldots,k_t\in\mathbb{Z}$ be the such that $\sum_{i=1}^{t}\left|k_i-\frac{k}{d}\cdot d_i\right|$ is minimised. Hence,

$$
\begin{aligned}
\mathsf{residue}_k(d_1,\ldots,d_t) &\geq \frac{1}{2}\sum_{j\in[t]:d_j\approx_2\tau^{2^\delta}}\left|k_j-\frac{k}{d}\cdot d_j\right|\\
&\geq \frac{1}{2}\cdot\left|\left\{j\in[t]:d_j\approx_2\tau^{2^\delta}\right\}\right|\cdot\min_{j\in[t]:d_j\approx_2\tau^{2^\delta}}\left|k_j-\frac{k}{d}\cdot d_j\right|\\
&\geq \Omega\left(\tau^{2^\delta}\right)\cdot\min_{j\in[t]:d_j\approx_2\tau^{2^\delta}}\left|k_j-\frac{k}{d}.d_j\right|\\
&= \Omega\left(\tau\right)\cdot\min_{j\in[t]:d_j\approx_2\tau^{2^\delta}}\tau^{2^\delta-1}\cdot\left|k_j-\frac{k}{d}\cdot d_j\right|\\
&\geq \Omega(\tau)\cdot\left(\frac{\eta}{2}-o(1)\right) && (\text{using (14)})\\
&\geq \Omega\left(\tau\right).
\end{aligned}
$$

Therefore, $\mathsf{residue}_k(d_1,\ldots,d_t)\geq\Omega(\tau)\geq\Omega\left(\frac{\tau+1}{2}\right)\geq\Omega\left(\tau+1\right)\geq\Omega\left(d^{2^{1-\Delta}}\right)$. □

D.2.1 Proof of Claim D.1

We prove the claim by analysing the following three sub-cases.

Case (i): $m_j<k_j$. Then, $\eta\geq\left|k_j-\alpha\cdot m_j\right|=k_j-\alpha\cdot m_j\geq k_j-m_j\geq 1$.

Now, let $\alpha_1:=\sum_{\nu=0}^{\delta}\frac{(-1)^\nu}{\tau^{2^\nu-1}}$, $\alpha_2:=\frac{(-1)^{\delta+1}}{\tau^{2^{\delta+1}-1}}$ and $\alpha_3:=\sum_{\nu=\delta+2}^{\Delta-1}\frac{(-1)^\nu}{\tau^{2^\nu-1}}$. Then, let $\alpha_4:=\tau^{2^\delta-1}\cdot\left(k_j-m_j\cdot\alpha_1\right)$. Noting that $\alpha=\alpha_1+\alpha_2+\alpha_3$ we have,

$$
\eta=\tau^{2^\delta-1}\cdot\left|k_j-\alpha\cdot m_j\right|=\left|\tau^{2^\delta-1}\cdot k_j-\tau^{2^\delta-1}\cdot m_j\cdot(\alpha_1+\alpha_2+\alpha_3)\right| \quad (\text{as } \alpha=\alpha_1+\alpha_2+\alpha_3 \text{ by definition})
$$

$$
\begin{aligned}
&\geq \left|\alpha_4 - \tau^{2^\delta-1}\cdot m_j\cdot\frac{(-1)^{\delta+1}}{\tau^{2^{\delta+1}-1}} - \tau^{2^\delta-1}\cdot m_j\cdot\alpha_3\right| \\
&\geq \left|\alpha_4 - \tau^{2^\delta-1}\cdot m_j\cdot\frac{(-1)^{\delta+1}}{\tau^{2^{\delta+1}-1}}\right| - \left|\tau^{2^\delta-1}\cdot m_j\cdot\alpha_3\right| \\
&\geq \left||\alpha_4| - \frac{m_j}{\tau^{2^\delta}}\right| - \left|\tau^{2^\delta-1}\cdot m_j\cdot\alpha_3\right| \\
&= \left||\alpha_4| - \frac{m_j}{\tau^{2^\delta}}\right| - \left|\sum_{\nu=\delta+2}^{\Delta-1}\cdot\frac{(-1)^\nu\cdot\tau^{2^\delta-1}\cdot m_j}{\tau^{2^\nu-1}}\right| \\
&\geq \left||\alpha_4| - \frac{m_j}{\tau^{2^\delta}}\right| - \left|\frac{\tau^{2^\delta-1}\cdot m_j}{\tau^{2^{\delta+2}-1}}\right| \\
&\qquad\qquad\qquad\qquad\text{(taking only the leading term of the summation)} \\
&\geq \left||\alpha_4| - \frac{m_j}{\tau^{2^\delta}}\right| - \left|\frac{\tau^{2^\delta-1}\cdot\tau^{2^\delta}}{\tau^{2^{\delta+2}-1}}\right| \qquad \text{(since } m_j \leq d_j \approx_2 \tau^{2^\delta}\text{)} \\
&\geq \left||\alpha_4| - \frac{m_j}{\tau^{2^\delta}}\right| - \frac{1}{\tau^2} \\
&= \left||\alpha_4| - \frac{m_j}{\tau^{2^\delta}}\right| - o(1).
\end{aligned}
$$

Notice that, as $m_j \leq d_j \approx_2 \tau^{2^\delta}$, $\frac{m_j}{\tau^{2^\delta}} \leq 1$.

Case (ii): $k_j \leq m_j \leq 6\cdot k_j$. Note that

$$
m_j = \frac{6\cdot m_j + m_j}{7} \leq \frac{6\cdot m_j + 6\cdot k_j}{7} = \frac{6}{7}\cdot d_j \leq \frac{6}{7}\cdot\tau^{2^\delta}, \text{ and}
$$

$$
m_j \geq \frac{m_j + k_j}{2} = \frac{d_j}{2} \geq \frac{1}{4}\cdot\tau^{2^\delta}.
$$

Thus $\frac{m_j}{\tau^{2^\delta}} \in \left[\frac{1}{4}, \frac{6}{7}\right]$. On the other hand, $\alpha_4 = \tau^{2^\delta-1}\cdot k_j - \tau^{2^\delta-1}\cdot m_j\cdot\alpha_1$ is an integer since the denominators of all the terms in α_1 divide $\tau^{2^\delta-1}$. Therefore $\frac{m_j}{\tau^{2^\delta}}$ is at least $\min\{1/4, 3/4, 6/7, 1/7\} = 1/7$ distance from any integer, and from $|\alpha_4|$ in particular. That is, $\left||\alpha_4| - \frac{m_j}{\tau^{2^\delta}}\right| \geq 1/7$ and $\eta \geq \left||\alpha_4| - \frac{m_j}{\tau^{2^\delta}}\right| - o(1) \geq \Omega(1)$.

Case (iii): $m_j > 6\cdot k_j$. Then,

$$
\begin{aligned}
-k_j + m_j\cdot\alpha_1 &= -k_j + m_j\cdot\left(\sum_{\nu=0}^{\delta}\frac{(-1)^\nu}{\tau^{2^\nu-1}}\right) \\
&= -k_j + m_j - m_j\cdot\left(\sum_{\nu=1}^{\delta}\frac{(-1)^{\nu-1}}{\tau^{2^\nu-1}}\right) \\
&\geq -k_j + m_j - \frac{m_j}{\tau} \\
&\geq \frac{m_j}{2} - k_j \qquad\qquad \text{(as } \tau = \omega(1)\text{)}
\end{aligned}
$$

$$\geq \frac{m_j}{2} - \frac{m_j}{6} = \frac{m_j}{3} \geq \frac{2}{7} \cdot (m_j + k_j) = \frac{2 \cdot d_j}{7} \geq \frac{\tau}{7} \geq 2.$$

Hence, $\left| |\alpha_4| - \frac{m_j}{\tau^{2^\delta}} \right| = \left| \tau^{2^\delta - 1} \cdot (-k_j + m_j \cdot \alpha_1) - \frac{m_j}{\tau^{2^\delta}} \right| \geq 2 - 1 = 1$ and $\eta \geq \Omega(1)$. □

D.3 Proof of Lemma 4.3

Using Lemma 3.2 (Item 1) and the fact that SP is sub-additive (Proposition B.6), we get

$$\mathsf{SP}_{k,\ell}(P) \leq \sum_{i=1}^{s} \mathsf{SP}_{k,\ell}(Q_{i,1} \cdots Q_{i,t_i}) \leq s \cdot 2^t \cdot d^2 \cdot \max_{\substack{k_0, \ell_0 \geq 0 \\ k_0 + \frac{k}{d-k} \cdot \ell_0 \leq k - \gamma}} M(n, k_0) \cdot M(n, \ell + \ell_0),$$

where $t := \max_i t_i$ is at most d. On the other hand, by our assumption we have $\mathsf{SP}_{k,\ell}(P) \geq 2^{-O(d)} \cdot M(n,k) \cdot M(n,\ell)$. Putting these two together, we get for some integers $k_0 \in [0..k]$, $\ell_0 \in [0..(d-k)]$ satisfying

$$k_0 + \frac{k}{d-k} \cdot \ell_0 \leq k - \gamma, \tag{15}$$

that,

$$\begin{aligned} s &\geq 2^{-O(d)} \cdot 2^{-t} \cdot d^{-2} \cdot \frac{M(n,k) \cdot M(n,\ell)}{M(n,k_0) \cdot M(n, \ell + \ell_0)} && (16) \\ &\geq 2^{-O(d)} \cdot \frac{M(n,k)}{M(n,k_0) \cdot (2n/\ell)^{\ell_0}} \\ &\qquad \text{(Lemma B.2 (Item 2) is applicable as } n_0 \geq d \text{ implies that } n \geq \left\lfloor \tfrac{nd}{n_0} \right\rfloor = \ell \text{; also } n_0 \leq n \text{ implies} \\ &\qquad\qquad \ell = \left\lfloor \tfrac{nd}{n_0} \right\rfloor \geq d \geq \ell_0) \\ &\geq 2^{-O(d)} \cdot \frac{M(n,k)}{M(n,k_0) \cdot (n_0/\ell_0)^{\ell_0}} && (17) \\ &\qquad \text{(because } 2n/\ell \leq 4n_0/\ell_0 \text{ as } \ell_0 \leq d \text{ and absorbing } 4^{\ell_0} \text{ in } 2^{-O(d)}) \\ &\geq 2^{-O(d)} \cdot \frac{(n/k)^k}{(6n/k_0)^{k_0} \cdot (n_0/\ell_0)^{\ell_0}} \\ &\qquad \text{(assuming } k_0, \ell_0 \neq 0 \text{ and using Lemma B.2 (Item 1) as } n \geq n_0 \geq d \geq \max\{k_0, \ell_0\}\text{; the analysis is} \\ &\qquad\qquad \text{easier if any of } k_0, \ell_0 \text{ is 0)} \\ &\geq 2^{-O(d)} \cdot \frac{(n/k)^k}{(n/k_0)^{k_0} \cdot (n_0/\ell_0)^{\ell_0}} && \text{(since } k_0 = O(d)) \\ &\geq 2^{-O(d)} \cdot \frac{(n/k)^k}{(n/k_0)^{k_0} \cdot \left(\frac{2(d-k)}{\ell_0} \cdot (n/k)^{\frac{k}{d-k}}\right)^{\ell_0}} && \text{(substituting } n_0) \end{aligned}$$

$$
\begin{aligned}
&= 2^{-O(d)} \cdot \frac{(n/k)^k}{(n/k_0)^{k_0} \cdot (\frac{d-k}{\ell_0})^{\ell_0} \cdot (n/k)^{\frac{k \cdot \ell_0}{d-k}}} && \text{(as } \ell_0 = O(d)\text{)} \\
&= 2^{-O(d)} \cdot \frac{(n/k)^k}{(n/k)^{k_0} \cdot (k/k_0)^{k_0} \cdot (\frac{d-k}{\ell_0})^{\ell_0} \cdot (n/k)^{\frac{k\ell_0}{d-k}}} && \text{(as } n/k_0 = (n/k) \cdot (k/k_0)\text{)} \\
&= 2^{-O(d)} \cdot \frac{(n/k)^{k-k_0-\frac{k}{d-k}\cdot\ell_0}}{(k/k_0)^{k_0} \cdot (\frac{d-k}{\ell_0})^{\ell_0}} \\
&\geq 2^{-O(d)} \cdot \frac{(n/d)^{\gamma}}{(d/k_0)^{k_0} \cdot (\frac{d}{\ell_0})^{\ell_0}} && \text{(using } k < d \text{ and (15))} \\
&= 2^{-O(d)} \cdot \left(\frac{n}{d}\right)^{\gamma} \cdot \left(\frac{k_0}{d}\right)^{k_0} \cdot \left(\frac{\ell_0}{d}\right)^{\ell_0} \\
&\geq 2^{-O(d)} \cdot \left(\frac{n}{d}\right)^{\gamma} \cdot (e^{-1/e})^d \cdot (e^{-1/e})^d && \text{(using } x^x \geq e^{-1/e} \text{ for } x > 0\text{)} \\
&\geq 2^{-O(d)} \cdot \left(\frac{n}{d}\right)^{\Omega(\gamma)}.
\end{aligned}
$$

□

D.4 Proof of Lemma 4.4

Using Lemma 3.2 (Item 2) and the fact that APP is sub-additive (Proposition B.6), we get

$$
\mathsf{APP}_{k,n_0}(P) \leq \sum_{i=1}^{s} \mathsf{APP}_{k,n_0}(Q_{i,1} \cdots Q_{i,t_i}) \leq s \cdot 2^t \cdot d^2 \cdot \max_{\substack{k_0,\ell_0 \geq 0 \\ k_0 + \frac{k}{d-k}\cdot\ell_0 \leq k-\gamma}} M(n,k_0) \cdot M(n_0,\ell_0).
$$

On the other hand, we have $\mathsf{APP}_{k,n_0}(P) \geq 2^{-O(d)} \cdot M(n,k)$. Putting these two together, we get for some integers $k_0, \ell_0 \geq 0$ satisfying

$$
k_0 + \frac{k}{d-k} \cdot \ell_0 \leq k - \gamma,
$$

that,

$$
s \geq 2^{-O(d)} \cdot 2^{-t} \cdot d^{-2} \cdot \frac{M(n,k)}{M(n,k_0) \cdot M(n_0,\ell_0)} \geq 2^{-O(d)} \cdot \left(\frac{n}{d}\right)^{\Omega(\gamma)}.
$$

(Using Lemma B.2, absorbing 6^{ℓ_0} in $2^{-O(d)}$, and borrowing calculations beginning from (17))

□

D.5 Proof of Lemma 4.6

We construct the word $\mathbf{w}$ as follows. Let $h' = \frac{h \cdot k}{d-k} \in \left[\frac{h}{29}, h\right]$. The word $\mathbf{w}$ shall consist of the following elements (the ordering of these elements shall be fixed shortly): $h, \ldots, h$ (k times), $-\lfloor h' \rfloor, \ldots, -\lfloor h' \rfloor$ (k_1 times), $-\lceil h' \rceil, \ldots, -\lceil h' \rceil$ (k_2 times), where $k_1 := (d-k)\lceil h' \rceil - kh$ and

$k_2 := d - k - k_1$. We note that $k_1, k_2 \in \mathbb{Z}_{\geq 0}$ and $k + k_1 + k_2 = d$. Assuming $\lfloor h' \rfloor = \lceil h' \rceil - 1$ (even if $h' \in \mathbb{Z}$, the calculations are similar), the total sum of the weights is

$$\sum_{i \in [d]} w_i = kh - k_1 \lfloor h' \rfloor - k_2 \lceil h' \rceil = kh - k_1(\lceil h' \rceil - 1) - k_2 \lceil h' \rceil = kh - k_1 \lceil h' \rceil + k_1 - k_2 \lceil h' \rceil$$
$$= kh - k_1 \lceil h' \rceil + (d-k) \lceil h' \rceil - kh - k_2 \lceil h' \rceil = 0. \tag{18}$$

Now we fix the ordering of the above weights. For $i = 1$ to d in this order, if the sum $\sum_{j \in [i-1]} w_j$ is non-negative (for example, this happens for $i = 1$), set w_i to be an arbitrary negative weight that is available, otherwise set it to be the positive weight h (if available).

If the above procedure never runs out of positive or negative weights at any step $i \in [d]$, then for all $i \in [d]$, $|w_1 + \cdots + w_i| \leq h$. In other words, $\mathbf{w}$ is *h-unbiased*. Now suppose the procedure runs out of negative weights at an index $i \in [d]$. This means that the sum $\sum_{j \in [i-1]} w_j$ is non-negative but there are no negative weights available among the unused weights. But then, the total sum of the weights would be equal to $\sum_{j \in [i-1]} w_j$ plus the sum of unused weights, which is greater than 0, contradicting (18). We get a similar contradiction if there are insufficient positive weights at any point. For the rest of the proof, we fix $\mathbf{w}$ to be the above word. Then,

$$k \cdot 2^h \leq n \leq d \cdot 2^h, \text{ so } 2^h \approx_{30} \binom{n}{k}. \tag{19}$$

Denoting the variables of $P_{\mathbf{w}}$ by $\mathbf{x} = \mathbf{y} \sqcup \mathbf{z}$, where $\mathbf{y}$ are the positive variables and $\mathbf{z}$ are the negative variables, we take

$$n_0 := |\mathbf{z}| \approx_2 (d-k) \cdot 2^{\lceil h' \rceil}.$$

Note that $n_0 \approx_2 (d-k) \cdot 2^{\lceil h' \rceil} \approx 2(d-k) \cdot 2^{h'} = 2(d-k) \cdot 2^{\frac{hk}{d-k}} \leq 2k \cdot 2^h = 2(n - n_0)$ where the last inequality follows from the fact that $\frac{d-k}{k} \cdot 2^{\frac{hk}{d-k}}$ is an increasing function of k when $k \in \left[\frac{d}{30}, \frac{d}{2}\right]$ and $h > 100$. That is, $n_0 \leq 2n/3$ and $n_0 \approx 2(d-k) \cdot \left(\frac{n}{k}\right)^{\frac{k}{d-k}}$ by (19) and $k \leq \frac{d}{2}$. Also, $n_0 \geq \frac{d-k}{2} \cdot 2^{\lceil h' \rceil} \geq \frac{d-k}{2} \cdot 2^{h'} \geq \frac{d-k}{2} \cdot 2^{\frac{h}{29}} \geq \frac{d-k}{2} \cdot 2^3 \geq 2d$ as $h > 100$ and $k \leq \frac{d}{2}$. Define a map $L : \mathbf{x} \to \langle \mathbf{z} \rangle$ as follows:

$$L(x) = \begin{cases} 0, \text{ if } x \in \mathbf{y}, \\ x, \text{ if } x \in \mathbf{z}. \end{cases}$$

We can lower bound the APP measure by using L and considering only the derivatives with respect to the set-multilinear monomials over all the positive sets, i.e., $\mathcal{M}_+(\mathbf{w})$. By the definition of the polynomial $P_{\mathbf{w}}$ and because $\sum_{i \in [d]} w_i = 0$, for every $m_+ \in \mathcal{M}_+$, there exists a unique $m_- \in \mathcal{M}_-$ such that $m_+ \cdot m_-$ is a monomial in $P_{\mathbf{w}}$ and vice versa.[18] Hence the set of all derivatives of $P_{\mathbf{w}}$ with respect to monomials in $\mathcal{M}_+$ is exactly $\mathcal{M}_-$, yielding

$$\partial^k (P_{\mathbf{w}}) \supseteq \mathcal{M}_-(\mathbf{w}). \tag{20}$$

Using the fact that $\sum_{i \in [d]} w_i = 0$ and (19), the size of $\mathcal{M}_-(\mathbf{w})$ is

$$|\mathcal{M}_-(\mathbf{w})| = 2^{\sum_{i \in [d] : w_i < 0} |w_i|} = 2^{hk} \geq 2^{-O(k)} \cdot \left(\frac{n}{k}\right)^k \geq 2^{-O(d)} \cdot M(n,k). \tag{21}$$

[18] Recall the definition of $P_{\mathbf{w}}$ from Section B. Because $|\mathcal{M}_+| = |\mathcal{M}_-|$, the bit representations of m_+ and m_- are the same. However, they can have different degrees.

The last bound follows from Lemma B.2 (Item 1), as $n \geq n_0 \geq d \geq k$. As the substitution π_L does not affect negative variables, thus,

$$\mathsf{APP}_{k,n_0}(P_{\mathbf{w}}) \geq \dim\left\langle \pi_L\left(\partial^k\left(P_{\mathbf{w}}\right)\right)\right\rangle \geq \dim\left\langle \pi_L(\mathcal{M}_-(\mathbf{w}))\right\rangle = \dim\left\langle \mathcal{M}_-(\mathbf{w})\right\rangle \geq 2^{-O(d)}\cdot M(n,k).$$

We now analyze the shifted partials of the same polynomial with $\ell := \left\lfloor \frac{n\cdot d}{n_0}\right\rfloor$. Recall that $n_0 \leq 2n/3$.

$$\begin{aligned}
\mathsf{SP}_{k,\ell}(P_{\mathbf{w}}) &\geq \dim\left\langle \mathbf{x}^\ell \cdot \partial^k\left(P_{\mathbf{w}}\right)\right\rangle \\
&\geq \dim\left\langle \mathbf{y}^\ell \cdot \mathcal{M}_-(\mathbf{w})\right\rangle && \text{(as } \mathbf{x} \supseteq \mathbf{y} \text{ and (20))} \\
&\geq \left|\mathbf{y}^\ell \cdot \mathcal{M}_-(\mathbf{w})\right| \\
&= \left|\mathbf{y}^\ell\right| \cdot \left|\mathcal{M}_-(\mathbf{w})\right| && \text{(since } \mathcal{M}_-(\mathbf{w}) \subseteq \mathbf{z}^{d-k} \text{ and } \mathbf{y}\cap\mathbf{z} = \Phi) \\
&= M(n-n_0,\ell)\cdot 2^{-O(d)}\cdot M(n,k) && \text{(using } |\mathbf{y}| = |\mathbf{x}| - |\mathbf{z}| \text{ and (21))} \\
&\geq M(n,\ell)\cdot\left(1-\frac{n_0}{n}\right)^{\ell}\cdot 2^{-O(d)}\cdot M(n,k) && \text{(using Lemma B.2 (Item 3))} \\
&\geq M(n,\ell)\cdot\left(1-\frac{n_0}{n}\right)^{\frac{n}{n_0}\cdot d}\cdot 2^{-O(d)}\cdot M(n,k) \\
&\geq 2^{-O(d)}\cdot M(n,k)\cdot M(n,\ell). && \text{(since } (1-x)^{1/x} \geq 1/3\sqrt{3} \text{ for } x := n_0/n \leq 2/3)
\end{aligned}$$

□

D.6 Proof of Lemma 4.7

We begin by obtaining bounds on the value of ℓ.

Claim D.2. $n_0 = o(qd)$, $d^2 = o(\ell)$ and $\ell = o(qd)$.

Proof.

$$\begin{aligned}
n_0 &= 2(d-k)\left(\frac{qd}{k}\right)^{\frac{k}{d-k}} \\
&\leq 2(d-k)\left(\frac{qd}{k}\right)^{\frac{d/2-\sqrt{d}/8}{d/2+\sqrt{d}/8}} && \left(\text{because } k = o(qd) \text{ and } k \leq \frac{d}{2} - \frac{\sqrt{d}}{8}\right) \\
&= 2(d-k)\cdot\frac{qd}{k}\cdot\left(\frac{k}{qd}\right)^{\frac{2}{4\sqrt{d}+1}} \\
&\leq 2d\cdot qd\cdot\frac{1}{(qd)^{\frac{1}{2.5\sqrt{d}}}} \\
&\leq 2d\cdot qd\cdot\frac{1}{2^{\frac{\log qd}{2.5\sqrt{d}}}}.
\end{aligned}$$

As $d \leq \frac{1}{150}\left(\frac{\log n}{\log\log n}\right)^2$ and $qd \geq \frac{n}{4}$, $\frac{\log qd}{2.5\sqrt{d}} \geq \frac{12\log\log n}{3} = 4\log\log n$. As $\log d^2 \leq 4\log\log n - \omega(1)$, $\frac{\log qd}{2.5\sqrt{d}} = \log d^2 + \omega(1)$ and $2^{\frac{\log qd}{2.5\sqrt{d}}} = \omega(d^2)$. Thus, $n_0 = o(q) = o(qd)$.[19] Now, $\ell \geq \frac{qd^2}{n_0} - 1 \geq \frac{qd^2}{o(q)} - 1 = \omega(d^2)$. Thus, $d^2 = o(\ell)$. Also,

$$\begin{aligned}
\ell &\leq \frac{qd^2}{n_0} \\
&= \frac{qd^2}{2(d-k)}\left(\frac{k}{qd}\right)^{\frac{k}{d-k}} \\
&\leq \frac{qd^2}{2(d-k)}\left(\frac{k}{qd}\right)^{\frac{1}{29}} && \left(\text{because } k = o(qd) \text{ and } \frac{d}{30} \leq k\right) \\
&\leq k\cdot(qd)^{\frac{28}{29}} && \left(\text{as } k \leq \frac{d}{2}\right) \\
&= o(n) && \left(\text{because } k \leq \log^2 n \text{ and } qd \leq n\right).
\end{aligned}$$

□

Let

$$S = \left\{ \prod_{i\in[k+1\ldots d]} x_{i,h(i)} : h \in \mathbb{F}_q[z], \deg(h) < k \right\}$$

and

$$T = \left\{ m : \exists \text{ monomials } m_1,\ m_2,\ \deg(m_1) = \ell, m_2 \in S \text{ and } m = m_1 m_2 \right\}.$$

Observe that $T \subseteq \left\langle \mathbf{x}^\ell \boldsymbol{\partial}^k NW_{q,d,k} \right\rangle$ and so, $\mathsf{SP}_{k,\ell}(NW_{q,d,k}) \geq |T|$. We obtain a lower bound on $|T|$. For $h \in \mathbb{F}_q[z]$ such that $\deg(h) < k$, let

$$T_h = \left\{ m_1 \prod_{i\in[k+1\ldots d]} x_{i,h(i)} : \deg(m_1) = \ell \right\}.$$

Then, $T = \cup_{\substack{h(z)\in\mathbb{F}_q[z]:\\ \deg(h)<k}} T_h$. Thus, from the inclusion-exclusion principle,

$$|T| \geq \sum_{\substack{h\in\mathbb{F}_q[z]:\\ \deg(h)<k}} |T_h| - \sum_{\substack{h_1\neq h_2\in\mathbb{F}_q[z]:\\ \deg(h_1),\deg(h_2)<k}} |T_{h_1} \cap T_{h_2}|. \tag{22}$$

Lower bound on $\sum_h |T_h|$. Fix an $h \in \mathbb{F}_q[z]$ such that $\deg(h) < k$. Then, since for monomials $m_1 \neq m_2$, $m_1 \cdot \prod_{i\in[k+1\ldots d]} x_{i,h(i)} \neq m_2 \cdot \prod_{i\in[k+1\ldots d]} x_{i,h(i)}$, $|T_h| = \binom{qd+\ell-1}{qd-1}$. Hence,

$$\sum_{\substack{h\in\mathbb{F}_q[z]:\\ \deg(h)<k}} |T_h| = |S|^k \cdot \binom{qd+\ell-1}{qd-1} = q^k \cdot \binom{qd+\ell-1}{qd-1}. \tag{23}$$

[19] In this proof, we need $n_0 = o(q)$. However, we require $n_0 = o(n)$, in Section D.7 and so we have mentioned $n_0 = o(qd)$ in the statement of the claim.

Upper bound on $\sum_{h_1 \neq h_2} |T_{h_1} \cap T_{h_2}|$. For $h_1, h_2 \in \mathbb{F}[z]$ such that $\deg(h_1), \deg(h_2) < k$, we say that $|h_1 \cap h_2| = r$ if $|\{h_1(k+1), \ldots, h_1(d)\} \cap \{h_2(k+1), \ldots, h_2(d)\}| = r$. Now

$$\sum_{h_1 \neq h_2} |T_{h_1} \cap T_{h_2}| = \sum_{r=0}^{k-1} \sum_{\substack{h_1 \neq h_2: \\ |h_1 \cap h_2| = r}} |T_{h_1} \cap T_{h_2}|. \tag{24}$$

Fix h_1 and h_2 such that $|h_1 \cap h_2| = r$. Let $m_1 = \prod_{i \in [k+1..d]} x_{i,h_1(i)}$ and $m_2 = \prod_{i \in [k+1..d]} x_{i,h_2(i)}$. A monomial $m \in T_{h_1} \cap T_{h_2}$ if and only if there exist degree ℓ monomials m_1' and m_2' such that $m = m_1' m_1 = m_2' m_2$. Thus $\frac{m_2}{\gcd(m_1, m_2)}$ must divide m_1'. As $|h_1 \cap h_2| = r$, $\gcd(m_1, m_2)$ has degree r, and so $\frac{m_2}{\gcd(m_1, m_2)}$ has degree $d - k - r$. Hence the number of possible monomials m_1', and thus the number of possible monomials m is at most $\binom{qd+\ell-d+k+r-1}{qd-1}$. Now, the number of possible polynomials h_1 and h_2 such that $|h_1 \cap h_2| = r$ is at most $\binom{d-k}{r} q^{k-r} q^k = q^{2k-r} \binom{d-k}{r}$.[20] Hence,

$$\sum_{\substack{h_1 \neq h_2: \\ |h_1 \cap h_2| = r}} |T_{h_1} \cap T_{h_2}| \leq q^{2k-r} \cdot \binom{d-k}{r} \binom{qd+\ell-d+k+r-1}{qd-1}. \tag{25}$$

Claim D.3. For $r \in [0..k-1]$, let $\chi(r) = q^{2k-r} \cdot \binom{d-k}{r} \binom{qd+\ell-d+k+r-1}{qd-1}$. Then $\chi(0) \geq \chi(r)$ for all $r \in [k-1]$.

Proof. We shall show that for all $r \in [0..k-2]$, $\frac{\chi(r+1)}{\chi(r)} < 1$; this will prove the claim. Fix any $r \in [0..k-2]$.

$$\begin{aligned}
\frac{\chi(r+1)}{\chi(r)} &= \frac{q^{2k-r-1} \cdot \binom{d-k}{r+1} \binom{qd+\ell-d+k+r}{qd-1}}{q^{2k-r} \cdot \binom{d-k}{r} \binom{qd+\ell-d+k+r-1}{qd-1}} \\
&= \frac{1}{q} \cdot \frac{\frac{(d-k)!}{(r+1)!(d-k-r-1)!}}{\frac{(d-k)!}{r!(d-k-r)!}} \cdot \frac{\frac{(qd+\ell-d+k+r)!}{(qd-1)!(\ell-d+k+r+1)!}}{\frac{(qd+\ell-d+k+r-1)!}{(qd-1)!(\ell-d+k+r)!}} \\
&= \frac{1}{q} \cdot \frac{d-k-r}{r+1} \cdot \frac{qd+\ell-d+k+r}{\ell-d+k+r+1} \\
&\leq \frac{d}{q} \cdot \frac{(1+o(1))qd}{(1-o(1))\ell} && \text{(by Claim D.2)} \\
&= (1+o(1))\frac{d^2}{\ell} \\
&= o(1), && \text{(by Claim D.2)}
\end{aligned}$$

where the second to last inequality follows from $k, r \geq 0$, $\ell, d, k, r = o(n)$ and $d, k, r = o(\ell)$ (Claim D.2), and the last equality from the fact that $d^2 = o(\ell)$ (Claim D.2). □

From Equations (24), (25), and the above claim, we get

$$\sum_{h_1 \neq h_2} |T_{h_1} \cap T_{h_2}| \leq k \cdot q^{2k} \cdot \binom{qd+\ell-d+k-1}{qd-1}. \tag{26}$$

[20] This is so because $|h_1 \cap h_2| = r$ implies that $h_1 - h_2 = (z-\alpha_1) \cdots (z-\alpha_r) \cdot g(z)$, where $\alpha_1, \ldots, \alpha_r$ are distinct elements in $[k+1..d]$ and $g(z)$ is a polynomial of degree at most $d-k$.

Thus from Equations (22), (23), and (26),

$$|T| \geq q^k \cdot \binom{qd+\ell-1}{qd-1} - k \cdot q^{2k} \cdot \binom{qd+\ell-d+k-1}{qd-1}$$
$$= q^k \cdot \binom{qd+\ell-1}{qd-1} \left(1 - \frac{k \cdot q^{2k} \cdot \binom{qd+\ell-d+k-1}{qd-1}}{q^k \cdot \binom{qd+\ell-1}{qd-1}}\right). \tag{27}$$

Claim D.4. $\frac{k \cdot q^{2k} \cdot \binom{qd+\ell-d+k-1}{qd-1}}{q^k \cdot \binom{qd+\ell-1}{qd-1}} \leq \frac{1}{2}$.

Proof.

$$\frac{k \cdot q^{2k} \cdot \binom{qd+\ell-d+k-1}{qd-1}}{q^k \cdot \binom{qd+\ell-1}{qd-1}} = k \cdot q^k \cdot \frac{\frac{(qd+\ell-d+k-1)!}{(qd-1)!(\ell-d+k)!}}{\frac{(qd+\ell-1)!}{(qd-1)!\ell!}}$$
$$= k \cdot q^k \cdot \frac{\ell \cdot (\ell-1) \cdot (\ell-2) \cdots (\ell-d+k+1)}{(qd+\ell-1) \cdot (qd+\ell-2) \cdot (qd+\ell-3) \cdots (qd+\ell-d+k)}$$
$$= k \cdot q^k \cdot \frac{1}{\left(\frac{qd-1}{\ell}+1\right) \cdot \left(\frac{qd-1}{\ell-1}+1\right) \cdot \left(\frac{qd-1}{\ell-2}+1\right) \cdots \left(\frac{qd-1}{\ell-d+k+1}+1\right)}$$
$$\leq k \cdot q^k \cdot \frac{1}{\left(\frac{qd-1}{\ell}\right)^{d-k}}$$
$$\leq k \cdot q^k \cdot \left(\frac{\ell}{qd}\right)^{d-k} e^{\frac{2(d-k)}{qd}} \qquad \text{(as } 1-x \geq e^{-2x} \text{ for } x \in [0,1/2]\text{)}$$

$$= (1+o(1)) \cdot k \cdot q^k \cdot \left(\frac{d}{n_0}\right)^{d-k} \qquad \text{(as } d-k = o(qd) \text{ and } \ell \leq \frac{qd^2}{n_0}\text{)}$$
$$= (1+o(1)) \cdot k \cdot q^k \cdot \left(\frac{d}{2(d-k)}\right)^{d-k} \cdot \left(\frac{k}{qd}\right)^k$$
$$\leq (1+o(1)) \cdot k \cdot \left(\frac{1}{2}\right)^k \qquad \text{(as } k \leq \frac{d}{2}\text{)}$$
$$\leq \frac{1}{2},$$

when $d \geq 120$. □

Thus, from Equation (27) and $k = \Theta(d)$, we get

$$|T| \geq \frac{1}{2} \cdot q^k \cdot \binom{qd+\ell-1}{qd-1} \geq 2^{-O(d)} \cdot \left(\frac{qd}{k}\right)^k \cdot \binom{qd+\ell-1}{qd-1} \geq 2^{-O(d)} \cdot \binom{qd+k-1}{qd-1} \cdot \binom{qd+\ell-1}{qd-1},$$

where the last inequality follows from Lemma B.2. Recall that $\mathsf{SP}_{k,\ell}(NW_{q,d,k}) \geq |T|$. Hence, $\mathsf{SP}_{k,\ell}(NW_{q,d,k}) \geq 2^{-O(d)} \cdot M(qd,k) \cdot M(qd,\ell)$.

D.7 A non-set-multilinear hard polynomial

For any $n, d, \Delta \in \mathbb{N}$ such that $120 \leq d \leq \frac{1}{150}\left(\frac{\log n}{\log\log n}\right)^2$, define $k = \left\lfloor \frac{\alpha \cdot d}{1+\alpha} \right\rfloor$, where $\alpha := \sum_{\nu=0}^{\Delta-1} \frac{(-1)^\nu}{\tau^{2^\nu - 1}}$ and $\tau := \left\lfloor d^{2^{1-\Delta}} \right\rfloor$. Then, let $n_0 = \left\lfloor 2(d-k) \cdot \left(\frac{n}{k}\right)^{\frac{k}{d-k}} \right\rfloor$ ($n_0 \leq n$, see Section D.7), $n_1 = n - n_0$, $\mathbf{y} = \{x_1, \ldots, x_{n_1}\}$ and $\mathbf{z} = \{x_{n_1+1}, \ldots, x_n\}$. Let $\mathcal{M}_y$ be the set of all (monic) monomials of degree k in $\mathbf{y}$ variables and $\mathcal{M}_z$ be the set of all (monic) monomials in $\mathbf{z}$ variables of degree $d-k$; it can be verified that $|\mathcal{M}_y| \leq |\mathcal{M}_z|$. Fix any one-to-one function $\sigma : \mathcal{M}_y \to \mathcal{M}_z$. Then, it is easy to see that for $P_\sigma := \sum_{m \in \mathcal{M}_y} m \cdot \sigma(m)$, $\mathsf{APP}_{k,n_0}(P) = M(n_1, k)$. While P_σ defined above might have a non-trivial set-multilinear component, it can be modified to ensure that there are no multilinear monomials in it. Notice that to prove a lower bound for such a polynomial, we must analyse the measure of a homogeneous formula computing it directly; we can not hope to get a lower bound by going via set-multilinearity as is done in [LST21].

Lemma D.5 (Non-set-multilinear hard polynomial). $\mathsf{APP}(P_\sigma) \geq 2^{-O(k)} M(n,k)$.

Proof. Using an analysis similar to the one in the proof of Claim D.2, it can be shown that $n_0 = o(n)$. Also, from the proof of Lemma 4.2, $k \in \left[\frac{d}{4}, \frac{d}{2}\right]$. Let us assume that $\Delta \geq 2$; the case of $\Delta = 1$ is simple and can be handled separately. Then, as $\frac{k}{d-k} \leq \alpha \leq 1 - \frac{1}{2\tau} \leq 1 - \frac{1}{2\sqrt{d}}$, $k \leq d - \frac{\sqrt{d}}{2} - k + \frac{k}{2\sqrt{d}}$, and hence $k \leq \frac{d}{2} - \frac{\sqrt{d}}{4} + \frac{k}{4\sqrt{d}} \leq \frac{d}{2} - \frac{\sqrt{d}}{4} + \frac{d}{8\sqrt{d}} = \frac{d}{2} - \frac{\sqrt{d}}{8}$.

Claim D.6. $|\mathcal{M}_y| \leq |\mathcal{M}_z|$.

Proof. $|\mathcal{M}_y| = \binom{n_1+k-1}{n_1-1}$ and $\mathcal{M}_z = \binom{n_0+d-k-1}{n_0-1}$. Thus,

$$\begin{aligned}
\frac{|\mathcal{M}_z|}{|\mathcal{M}_y|} &= \frac{(n_0+d-k-1)(n_0+d-k-2)\cdots n_0}{(n_1+k-1)(n_1+k-2)\cdots n_1} \cdot \frac{k!}{(d-k)!} \\
&\geq \frac{k!}{(d-k)!} \cdot \frac{n_0^{d-k}}{n_1^k} \cdot \frac{1}{\left(1+\frac{k-1}{n_1}\right)^k} \\
&\geq (1-o(1)) \cdot \frac{k!}{(d-k)!} \cdot ((2-o(1))(d-k))^{d-k} \left(\frac{n}{k}\right)^k \cdot \frac{1}{n^k} \\
&\qquad \text{(replacing } n_0 \text{ by its value and as } n_0 = o(n), n_1 = \Theta(n) = \omega(k^2)\text{)} \\
&\geq (1.9)^{d-k} \frac{(1-o(1))\sqrt{2\pi k}\left(\frac{k}{e}\right)^k}{(1+o(1))\sqrt{2\pi(d-k)}\left(\frac{d-k}{e}\right)^{d-k}} \cdot \frac{(d-k)^{d-k}}{k^k} \\
&\qquad \text{(using Sterling's approximation)} \\
&\geq (1.8)^{d-k} \cdot e^{d-2k} \cdot \sqrt{\frac{k}{d-k}} \\
&\geq 1,
\end{aligned}$$

for $d \geq 120$. □

Claim D.7. $M(n_1,k) \geq 2^{-O(k)} M(n,k)$.

Proof. As $n_1 = \Theta(n) \geq k$, from Lemma B.2, we get

$$M(n_1,k) \geq \left(\frac{n_1}{k}\right)^k = \left(\frac{n}{k}\right)^k \left(1-\frac{n_0}{n}\right)^k \geq \left(\frac{n}{k}\right)^k \cdot e^{-\frac{2kn_0}{n}} \geq 2^{-O(k)} \cdot \left(\frac{6n}{k}\right)^k \geq 2^{-O(k)} \cdot M(n,k),$$

where the third to last inequality follows from $n_0 = o(n)$ and the last inequality follows from $n \geq k$ and Lemma B.2. □

□

D.8 Proof of Theorem 4.9

We can assume that $d^{2^{1-\Delta}} = \omega(1)$ and $h := \lfloor \log n \rfloor > 100$, as otherwise the lower bound is trivial. Suppose $IMM_{n,d}$ has a homogeneous formula C of product-depth at most Δ. Consider the polynomial $P_{\mathbf{w}}$, given by Lemma 4.6, by setting $k := \left\lfloor \frac{\alpha \cdot d}{1+\alpha} \right\rfloor$, where $\alpha := \sum_{\nu=0}^{\Delta-1} \frac{(-1)^\nu}{\tau^{2^\nu}-1}$ and $\tau := \left\lfloor d^{2^{1-\Delta}} \right\rfloor$; these parameters are the same as those in Lemma 4.2. It is easy to show that $k \in \left[\frac{d}{4}, \frac{d}{2}\right]$. As $\mathbf{w}$ is *h-unbiased*, by Lemma B.13 there exists a homogeneous formula C' of product-depth at most Δ computing $P_{\mathbf{w}}$ such that $\text{size}(C') \leq \text{size}(C)$. Hence, by Lemma 4.2, there exist homogeneous polynomials $\{Q_{i,j}\}_{i,j}$ such that $P_{\mathbf{w}} = \sum_{i\in[s]} Q_{i,1} \cdots Q_{i,t_i}$, $s \leq \text{size}(C')$ and $\text{residue}_k(\deg(Q_{i,1}), \ldots, \deg(Q_{i,t_i})) \geq \Omega\left(d^{2^{1-\Delta}}\right)$ for $i \in [s]$. Denoting the number of variables in $P_{\mathbf{w}}$ by $\widetilde{n}$, Lemma 4.6 guarantees that $n_0 \leq 2(d-k) \cdot \left(\frac{\widetilde{n}}{k}\right)^{\frac{k}{d-k}}$, $\ell = \left\lfloor \frac{\widetilde{n} \cdot d}{n_0} \right\rfloor$ and $\mathsf{SP}_{k,\ell}(P_{\mathbf{w}}) \geq 2^{-O(d)} \cdot M(\widetilde{n},k) \cdot M(\widetilde{n},\ell)$. Therefore, we can apply Lemma 4.3 to the same polynomial $P_{\mathbf{w}}$ which gives that $s \geq 2^{-O(d)} \cdot \left(\frac{\widetilde{n}}{d}\right)^{\Omega\left(d^{2^{1-\Delta}}\right)}$. Hence, $\text{size}(C) \geq \text{size}(C') \geq s \geq 2^{-O(d)} \cdot n^{\Omega\left(d^{2^{1-\Delta}}\right)}$, since $\widetilde{n} \geq 2^h \geq n/2 = \omega(d)$. □

D.9 Proof of Theorem 4.10

We analyse the cases $\Delta = 1$ and $\Delta \geq 2$ separately.

$\underline{\Delta = 1.}$ Let C be a homogeneous formula of product-depth 1 computing $NW_{q,d,k}$. Then, $C = \sum_{i\in[s]} \prod_{j\in[d]} Q_{i,j}$, where $Q_{i,j}$ are linear forms. Observe that for any $i \in [k]$, $\partial^k\left(\prod_{j\in[d]} Q_{i,j}\right) \subseteq \left\langle \prod_{j\in[d]\setminus S} C_{i,j} : |S| = k \right\rangle$. Thus, $\dim\left\langle \partial^k\left(\prod_{j\in[d]} Q_{i,j}\right)\right\rangle \leq \binom{d}{k}$. As $\partial^k C \subseteq \sum_{i\in[s]} \left\langle \partial^k\left(\prod_{j\in[d]} Q_{i,j}\right)\right\rangle$, $\dim\langle \partial^k C\rangle \leq s \cdot \binom{d}{k}$.

On the other hand, $\dim\langle \partial^k(NW_{q,d,k})\rangle = \binom{d}{k} \cdot q^k$: For every $S \subseteq [d], |S| = k$,

$$T_S := \left\{ \prod_{i\in[d]\setminus S} x_{i,h(i)} : h \in \mathbb{F}[z], \deg(h) < k \right\} \subseteq \partial^k(NW_{q,d,k}).$$

Now, for $h_1 \neq h_2 \in \mathbb{F}[z], \deg(h_1), \deg(h_2) < k$, there exists an $i \in [d] \setminus S$ such that $h_1(i) \neq h_2(i)$ because $|[d] \setminus S| = d - k \geq k+1$. Thus, $\prod_{i\in[d]\setminus S} x_{i,h_1(i)} \neq \prod_{i\in[d]\setminus S} x_{i,h_2(i)}$, and $|T_S| = q^k$. Also, for

$S \neq S' \subseteq [d]$, $|S| = |S'| = k$, T_S and $T_{S'}$ are disjoint. Hence, $\dim \left\langle \partial^k(NW_{q,d,k}) \right\rangle \geq \binom{d}{k} \cdot q^k$.[21] Thus, $s \geq q^k = \left(\frac{n}{d}\right)^{O(d)}$ as $k = \Theta(d)$ and $qd = \Theta(n)$. Because $d \leq n^{1-\epsilon}$, this means that $s \geq n^{O(d)}$.

$\underline{\Delta \geq 2.}$ We can assume that $d^{2^{1-\Delta}} = \omega(1)$, as otherwise the given bound is trivial. Let C be a homogeneous formula of product-depth at most Δ computing $NW_{q,d,k}$; C is a formula in qd variables. By Lemma 4.2, there exist homogeneous polynomials $\left\{Q_{i,j}\right\}_{i,j}$ such that $NW_{q,d,k} = \sum_{i \in [s]} Q_{i,1} \cdots Q_{i,t_i}$, $s \leq \text{size}(C)$, and $\text{residue}_k\left(\deg(Q_{i,1}), \ldots, \deg(Q_{i,t_i})\right) \geq \Omega\left(d^{2^{1-\Delta}}\right)$ for $i \in [s]$. From the proof of Lemma 4.2, $k \in \left[\frac{d}{4}, \frac{d}{2}\right]$. In fact, as $\frac{k}{d-k} \leq \alpha \leq 1 - \frac{1}{2\tau} \leq 1 - \frac{1}{2\sqrt{d}}$, $k \leq \frac{d}{2} - \frac{\sqrt{d}}{8}$. Thus, Lemma 4.7 guarantees that for $n_0 = 2(d-k) \cdot \left(\frac{qd}{k}\right)^{\frac{k}{d-k}}$ and $\ell = \left\lfloor \frac{qd^2}{n_0} \right\rfloor$, $\mathsf{SP}_{k,\ell}\left(NW_{q,k,d}\right) \geq 2^{-O(d)} \cdot M(qd,k) \cdot M(qd,\ell)$. Also, it follows from the proof of Lemma 4.7 (see Claim D.2) that for $n_0 \leq qd$. So, applying Lemma 4.3 to $NW_{k,d,q}$ we get that $s \geq 2^{-O(d)} \cdot \left(\frac{qd}{d}\right)^{\Omega\left(d^{2^{1-\Delta}}\right)} = 2^{-O(d)} \cdot n^{\Omega\left(d^{2^{1-\Delta}}\right)}$ as $qd \geq \frac{n}{4}$ and $d = o(n)$. Hence, $\text{size}(C) \geq 2^{-O(d)} \cdot n^{\Omega\left(d^{2^{1-\Delta}}\right)}$. □

D.10 Can we avoid the $2^{-O(k)}$ factor loss in our lower bounds?

In this section, we will work with the shifted partials measure, but the conclusion also holds for APP. For any homogeneous polynomial $P \in \mathbb{F}[x_1, \ldots, x_n]$ of degree d and any choice of parameters $k < d$ and ℓ, note that

$$\mathsf{SP}(P) \leq \min\left\{M(n,k) \cdot M(n,\ell), M(n,\ell+d-k)\right\}.$$

Hence, the best possible lower bound *with our current estimates for the measure* (from Lemma 3.2), for fixed values of n, d, k, γ, is at most

$$\Lambda := \max_{\ell} \left\{ \frac{\min\left\{M(n,k) \cdot M(n,\ell), M(n,\ell+d-k)\right\}}{\max\limits_{\substack{k_0,\ell_0 \geq 0: \\ k_0 + \frac{k}{d-k} \cdot \ell_0 \leq k - \gamma}} M(n,k_0) \cdot M(n,\ell+\ell_0)} \right\}. \tag{28}$$

We claim that the above quantity is at most $2^{-\Omega(k)} \cdot n^{O(\gamma)}$ as long as $d \leq n^{1/4}$. We will assume that $k \leq d/2$ as the other case simply reduces to this case. Also, we assume that $\gamma = o(k)$ as otherwise the multiplicative factor of $2^{-\Omega(k)}$ is irrelevant. The maximum of the R.H.S. in (28) is attained for the choice of ℓ when $M(n,k) \cdot M(n,\ell)$ and $M(n,\ell+d-k)$ are the closest. Notice that $\frac{M(n,k) \cdot M(n,\ell)}{M(n,\ell+d-k)}$ increases along with ℓ. Also, as $k \leq d/2$, there are values of ℓ for which the ratio is less than 1 and greater than 1. Thus we can fix ℓ (for a given k) such that $M(n,k) \cdot M(n,\ell) = M(n,\ell+d-k)$ (we use an exact equality here for brevity). Then,

$$\Lambda \leq \frac{M(n,k) \cdot M(n,\ell)}{\max\limits_{\substack{k_0,\ell_0 \geq 0: \\ k_0 + \frac{k}{d-k} \cdot \ell_0 \leq k - \gamma}} M(n,k_0) \cdot M(n,\ell+\ell_0)}$$

[21] In fact, it can be shown that this is an equality.

$$= \min_{\substack{k_0,\ell_0\geq 0:\\ k_0+\frac{k}{d-k}\cdot\ell_0\leq k-\gamma}} \frac{M(n,k)\cdot M(n,\ell)}{M(n,k_0)\cdot M(n,\ell+\ell_0)} \tag{29}$$

We have two cases depending on how large the value of ℓ is.

Case 1: $\ell \geq d^2$.

$$\begin{aligned}
\frac{M(n,k)\cdot M(n,\ell)}{M(n,k_0)\cdot M(n,\ell+\ell_0)} &\leq \frac{M(n,k)}{M(n,k_0)\cdot\left(1+\frac{n-1}{\ell+\ell_0}\right)^{\ell_0}} \\
&\leq \frac{4\cdot M(n,k)}{M(n,k_0)\cdot\left(1+\frac{n-1}{\ell+1}\right)^{\ell_0}} \quad (\ell_0 \leq d \text{ and } \ell \geq d^2 \text{ implies that } \left(\tfrac{\ell+\ell_0}{\ell+1}\right)^{\ell_0} \leq 4) \\
&\leq \frac{4\cdot M(n,k)}{M(n,k_0)\cdot M(n,k)^{\ell_0/(d-k)}} \\
&\qquad (\text{using } \left(1+\tfrac{n-1}{\ell+1}\right)^{d-k} \geq \tfrac{M(n,\ell+d-k)}{M(n,\ell)} \text{ and } M(n,k)\cdot M(n,\ell) = M(n,\ell+d-k) \\
&\leq \frac{4e\sqrt{2\pi k_0}\cdot(1+o(1))^k\cdot(en/k)^{k-\frac{k}{d-k}\cdot\ell_0}}{(en/k_0)^{k_0}}. \\
&\qquad\qquad (\text{using Sterling's approximation and } k_0 = o(n))
\end{aligned}$$

Case 2: $\ell \leq d^2 \leq \sqrt{n}$. In this regime, ignoring some polynomial factors, $M(n,k) = \left(\frac{en}{k}\right)^k$, $M(n,\ell) = \left(\frac{en}{\ell}\right)^\ell$ (unless ℓ is 0, in which case we may take $\ell = 1$ as this does not alter the quantities we are considering by more than linear factors) and $M(n,\ell+d-k) = \left(\frac{en}{\ell+d-k}\right)^{\ell+d-k}$. Similarly, $M(n,k_0) = \left(\frac{en}{k_0}\right)^{k_0}$ and $M(n,\ell+\ell_0) = \left(\frac{en}{\ell+\ell_0}\right)^{\ell+\ell_0}$. These approximations follow as we can upper bound $\left(1+\frac{x}{y}\right)^x$, and upper and lower bound $\left(1-\frac{x}{y}\right)^x$ by a constant when $x^2 = O(y)$. Note that the real valued function defined by $f(x) := x\cdot\ln(en/x)$ over $x \in [1, \ell+d-k]$ is concave. Hence applying Jensen's inequality we obtain that $f(\ell+\ell_0) - f(\ell) \geq \frac{\ell_0}{d-k}\cdot(f(\ell+d-k) - f(\ell))$ or equivalently, $\frac{\left(\frac{en}{\ell+\ell_0}\right)^{\ell+\ell_0}}{\left(\frac{en}{\ell}\right)^\ell} \geq \left(\frac{\left(\frac{en}{\ell+d-k}\right)^{\ell+d-k}}{\left(\frac{en}{\ell}\right)^\ell}\right)^{\frac{\ell_0}{d-k}} = \text{poly}(n,d)\cdot\left(\frac{en}{k}\right)^{\frac{k}{d-k}\cdot\ell_0}$ as $M(n,k)M(n,\ell) = M(n,\ell+d-k)$ and $\ell_0 \leq d-k$. Therefore,

$$\begin{aligned}
\frac{M(n,k)\cdot M(n,\ell)}{M(n,k_0)\cdot M(n,\ell+\ell_0)} &\leq \text{poly}(n,d)\cdot\frac{\left(\frac{en}{k}\right)^k\cdot\left(\frac{en}{\ell}\right)^\ell}{\left(\frac{en}{k_0}\right)^{k_0}\cdot\left(\frac{en}{\ell+\ell_0}\right)^{\ell+\ell_0}} \\
&\leq \text{poly}(n,d)\cdot\frac{(en/k)^{k-\frac{k}{d-k}\cdot\ell_0}}{(en/k_0)^{k_0}}.
\end{aligned}$$

Thus, in both cases, ignoring some polynomial factors and $(1+o(1))^k \leq 2^{o(k)}$ because they are asymptotically smaller than the $2^{\Omega(k)}$ factor, we get that

$$\Lambda \leq \min_{\substack{k_0,\ell_0\geq 0:\\ k_0+\frac{k}{d-k}\cdot\ell_0\leq k-\gamma}} \frac{(en/k)^{k-\frac{k}{d-k}\cdot\ell_0}}{(en/k_0)^{k_0}}$$
$$= \min_{\substack{k_0,\ell_0\geq 0:\\ k_0+\frac{k}{d-k}\cdot\ell_0\leq k-\gamma}} \frac{(en/k)^{k-k_0-\frac{k}{d-k}\cdot\ell_0}}{(k/k_0)^{k_0}}, \tag{30}$$

Thus to show that $\Lambda \leq 2^{-\Omega(k)}\cdot n^{O(\gamma)}$, we only need to show that there exist values of k_0 and ℓ_0 for which the fraction in the R.H.S. of (30) is at most $2^{-\Omega(k)}\cdot n^{O(\gamma)}$. To find such k_0, ℓ_0, we let k_0 be a function of ℓ_0 defined as $k_0 = \left\lfloor k-\gamma-\frac{k}{d-k}\cdot\ell_0 \right\rfloor$. As we keep adding 1 to ℓ_0 starting from $\left\lfloor \frac{d-k}{3} \right\rfloor$ to $\left\lfloor \frac{d-k}{2} \right\rfloor$, note that k_0 decreases from nearly $\lfloor 2k/3-\gamma \rfloor$ to nearly $\lfloor k/2-\gamma \rfloor$, with the decrease being at most 1 at a time (as $k \leq d-k$). Therefore, as $\gamma = o(k)$, there must be a choice of $\ell_0 \in [0..(d-k)]$ such that $k_0 \approx_2 2k/3$. Further, for this value of ℓ_0 and k_0, we also have $k-\gamma-1 \leq k_0+\frac{k}{d-k}\cdot\ell_0 \leq k-\gamma$. Thus, (30) yields $\Lambda \leq 2^{-\Omega(k)}\cdot n^{O(\gamma)}$.

E Proofs from Section 5

Algorithm 2 Degree sequence of a right-heavy binary tree

1. **function** DEG-SEQ($\mathcal{T}$)
2. $v_0 \leftarrow$ root node of $\mathcal{T}$.
3. **if** v_0 is a leaf **then**
4. **return** (1).
/* returning a singleton tuple */
5. **end if**
6. $d \leftarrow \text{leaves}(v_0)$, $i \leftarrow 0$.
7. **while** v_i is not a leaf **do**
8. $v_{i+1} \leftarrow$ right child of v_i, $i \leftarrow i+1$.
9. **end while**
10. $v \leftarrow v_j$ corresponding to the largest index j such that $\text{leaves}(v_j) > \frac{d}{3}$.
11. $d_1 \leftarrow d - \text{leaves}(v)$.
12. **return** $(d_1, \text{DEG-SEQ}(\mathcal{T}_v))$.
/* To avoid a tuple of tuples, we may assume that $(d_1, \text{DEG-SEQ}(\mathcal{T}_v))$ is flattened before returning. */
13. **end function**

E.1 Proof of Lemma 5.1

By induction on d. If $d = 1$, then $\text{DEG-SEQ}(\mathcal{T}) = (d_1) = (1)$, so all the conditions are trivially met. For $d \geq 2$, consider the node v of $\mathcal{T}$ defined at line 10 of Algorithm 2. Suppose the left and right

children of v are v_L and v_R respectively. We have

$$\text{leaves}(v) = \text{leaves}(v_L) + \text{leaves}(v_R) \leq 2 \cdot \text{leaves}(v_R) \leq \frac{2d}{3}.$$

Then by line 11,

$$e_1 = d - d_1 = \text{leaves}(v) \in \left(\frac{d}{3}, \frac{2d}{3}\right] = \left(\frac{e_0}{3}, \frac{2e_0}{3}\right].$$

Note that if v itself was a leaf, then $\text{leaves}(v) \leq \frac{2d}{3}$ still holds as $d \geq 2$. From line 12, it is evident that $(d_2, \ldots, d_t) = \text{DEG-SEQ}(\mathcal{T}_v)$. As the number of leaves in $\mathcal{T}_v$ is $\text{leaves}(v) < d$, by induction, we have that for all $i \in [t-2]$,

$$f_i \in \left(\frac{f_{i-1}}{3}, \frac{2f_{i-1}}{3}\right]$$

where $f_i := \text{leaves}(v) - \sum_{j=2}^{i+1} d_j$. But, notice that $f_i = d - d_1 - \sum_{j=2}^{i+1} d_j = e_{i+1}$. Hence, we have, for $i \in [t-1]$,

$$e_i = f_{i-1} \in \left(\frac{f_{i-2}}{3}, \frac{2f_{i-2}}{3}\right] = \left(\frac{e_{i-1}}{3}, \frac{2e_{i-1}}{3}\right].$$

Also, by induction, we trivially get $d_t = 1$ and $e_t = d - d_1 - \sum_{j=2}^{t} d_j = \text{leaves}(v) - \sum_{j=2}^{t} d_j = 0$. Using $e_i \in \left(\frac{e_{i-1}}{3}, \frac{2e_{i-1}}{3}\right]$ and $e_0 = d$, we get $e_i \in \left(\frac{d}{3^i}, \frac{2^i \cdot d}{3^i}\right]$, so $1 = d_t = e_{t-1} - e_t = e_{t-1}$. Hence, $\frac{d}{3^{t-1}} \leq 1 \leq \frac{2^{t-1} \cdot d}{3^{t-1}}$ and $\log_3 d + 1 \leq t \leq \log_{3/2} d + 1$. □

E.2 Proof of Lemma 5.2

The proof is by induction on the size of the formula C. In the base case, C computes a variable, say x_1. Then, we have $d = s = \text{size}(C) = t = d_1 = 1$ and $f = x_1$, all consistent with the lemma statement.

Let v be the node in the canonical parse tree $\mathcal{T} := \mathcal{T}(C)$ at line 10 in Algorithm 2 – that is, it is the last node in the rightmost path of $\mathcal{T}$ which has more than $\frac{d}{3}$ leaves in its subtree. Now, let $\mathcal{P}$ be an arbitrary parse tree of C and let ϕ denote an isomorphism from $\mathcal{P}$ to $\mathcal{T}$ – we know such an isomorphism exists from Proposition B.10. Let u be the node in $\mathcal{P}$ such that $\phi(u) = v$. Let g be the multiplication gate in C that corresponds to u.

Recall that C_g and $C_{g \leftarrow y}$, respectively, denote the sub-formula at g and the formula obtained by replacing the gate g with y, where y could be a new variable or a constant from the field. Due to homogeneity, $\deg(C_g) = \text{leaves}(u) = \text{leaves}(v)$. Also, note that C_g is a UPT formula as it is a sub-formula of C. However, the formula $C_{g \leftarrow 0}$ need not be homogeneous, as we require all the leaves of homogeneous formulas to be labelled by variables. Nevertheless, one can easily eliminate the zero gates by simplifying the formula using the rules $g' \times 0 = 0$ and $g' + 0 = g'$; this ultimately results in a homogeneous formula. In fact, we argue below that it would be a UPT formula.

Let h be the first addition gate on the path from g to the root of C. If no such gate exists, then $C_{g \leftarrow 0} = 0$ and thus can be considered to be a UPT formula with its canonical parse tree being the empty tree. Otherwise, let h' be the child of h such that g is in the subformula rooted at h'. Then, $C_{g \leftarrow 0} = 0$ is equivalent to the formula obtained by removing the edge from h' to h in C and the entire sub-formula at h'. Now, a parse tree is constructed by picking only one child of

every addition gate. Since $C_{g\leftarrow 0} = 0$ is equivalent to the formula obtained by removing a child of an addition gate from C, every parse tree of $C_{g\leftarrow 0}$ is also a parse tree of C. Thus, $C_{g\leftarrow 0}$ is a UPT formula. We will make use of the fact that C_g and $C_{g\leftarrow 0}$ are UPT formulas later.

For a new variable y, the formula $C_{g\leftarrow y}$ uses exactly one copy of y, so $C_{g\leftarrow y}$ computes a polynomial in $\mathbb{F}[x_1, \ldots, x_n, y]$ of the form $A_g \cdot y + B_g$, where A_g and B_g are polynomials in $\mathbb{F}[x_1, \ldots, x_n]$. Substituting the variable y to 0 in $C_{g\leftarrow y} = A_g \cdot y + B_g$, we see that B_g is computed by the formula $C_{g\leftarrow 0}$. Plugging in y to be the polynomial computed by C_g results in the original formula C. Therefore,

$$C = C_{g\leftarrow y}|_{y=C_g} = A_g \cdot C_g + B_g = A_g \cdot C_g + C_{g\leftarrow 0}. \tag{31}$$

It is easy that A_g is a homogeneous polynomial of degree $d - \text{leaves}(v)$.

As C_g and $C_{g\leftarrow 0}$ are smaller formulas compared to C, by induction, we have the expressions:

$$C_g = \sum_{i=1}^{s_1} Q_{i,1}^{(1)} \cdots Q_{i,t_1}^{(1)},$$

and

$$C_{g\leftarrow 0} = \sum_{i=1}^{s_2} Q_{i,1}^{(2)} \cdots Q_{i,t_2}^{(2)},$$

for some homogeneous polynomials $\left\{Q_{i,j}^{(1)}\right\}_{i,j}$ and $\left\{Q_{i,j}^{(2)}\right\}_{i,j}$, $t_1 \geq 1, t_2 \geq 1$, $s_1 \in [0..\text{size}(C_g)]$, and $s_2 \in [0..\text{size}(C_{g\leftarrow 0})]$. Plugging these expressions in (31), we get

$$C = A_g \cdot C_g + C_{g\leftarrow 0} = A_g \cdot \sum_{i=1}^{s_1} Q_{i,1}^{(1)} \cdots Q_{i,t_1}^{(1)} + \sum_{i=1}^{s_2} Q_{i,1}^{(2)} \cdots Q_{i,t_2}^{(2)} \tag{32}$$

Since the canonical parse tree of C_g is

$$\begin{aligned}
\mathcal{T}(C_g) &= \text{can}(\mathcal{P}_u) && \text{(as } \mathcal{P} \text{ is a parse tree of } C, \mathcal{P}_u \text{ is a parse tree of } C_g\text{)} \\
&= \text{can}(\mathcal{P})_{\phi(u)} && \text{(from Proposition B.10)} \\
&= \text{can}(\mathcal{P})_v && \text{(as } \phi(u) = v\text{)} \\
&= \mathcal{T}(C)_v && \text{(as } \mathcal{T}(C) = \text{can}(\mathcal{P})\text{)}
\end{aligned}$$

and that of $C_{g\leftarrow 0}$ is $\mathcal{T}(C)$, we have $\text{DEG-SEQ}(\mathcal{T}(C)_v) = (d_1^{(1)}, \ldots, d_{t_1}^{(1)})$ and $\text{DEG-SEQ}(\mathcal{T}(C)) = (d_1^{(2)}, \ldots, d_{t_2}^{(2)})$, where $d_j^{(\kappa)} := \deg\left(Q_{i,j}^{(\kappa)}\right)$ for any $\kappa \in [2], i \in [s_\kappa], j \in [t_\kappa]$. From line 12 of the function DEG-SEQ for input $\mathcal{T}(C)$, we have $\text{DEG-SEQ}(\mathcal{T}(C)) = (d - \text{leaves}(v), \text{DEG-SEQ}(\mathcal{T}(C)_v))$; comparing these two sequences element-wise, we get, $d_1^{(2)} = d - \text{leaves}(v)$, and $d_{j+1}^{(2)} = d_j^{(1)}$ for j in $[t_1]$, and $t_2 = 1 + t_1$. From (32), we have

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

where $s := s_1 + s_2$, $t := t_2$, and

$$Q_{i,j} := \begin{cases} A_g & \text{for } i \in [s_1] \text{ and } j = 1 \\ Q_{i,j-1}^{(1)} & \text{for } i \in [s_1] \text{ and } j \in [2, t_2] \\ Q_{i-s_1,j}^{(2)} & \text{for } i \in [s_1 + 1, s_1 + s_2] \text{ and } j \in [t_2] \end{cases}$$

All that remains to be checked now is that the degrees of $Q_{i,j}$ for different j's matches the degree sequence $\textsc{Deg-seq}(\mathcal{T}(C)) = (d_1^{(2)}, \ldots, d_{t_2}^{(2)})$.

For $i > s_1$, clearly $\deg(Q_{i,j}) = \deg(Q^{(2)}_{i-s_1,j}) = d_j^{(2)}$. For $i \le s_1$, we have $\deg(Q_{i,1}) = \deg(A_g) = d - \text{leaves}(v) = d_1^{(2)}$, and for $j \in [2, t_2]$, $\deg(Q_{i,j}) = \deg(Q^{(1)}_{i,j-1}) = d^{(1)}_{j-1} = d_j^{(2)}$. As desired, $s = s_1 + s_2 \le \text{size}(C_g) + \text{size}(C_{g\leftarrow 0}) \le \text{size}(C)$. □

E.3 Proof of Lemma 5.3

The definition of residue implies that it is sufficient to show that $\sum_{p=1}^{t} \left| k_p - \frac{k}{d} \cdot d_p \right| \ge \frac{\log_3 d - 10}{108}$ for arbitrary integers $k_1, \ldots, k_t$. We now argue that this is indeed the case. Let us assume that $d > 3^{10}$ as otherwise, the lower bound on residue is trivial. We import the definitions of $\{e_i\}_{i\in[0..t]}$ (line 4), m (line 6), and the function $\mathcal{J}$ (line 8) from Algorithm 3. By Lemma 5.1, we have $d_t = 1$, $e_t = 0$, $t \ge \log_3 d + 1 \ge 11$, and for all $i \in [t-1]$,

$$e_i \in \left(\frac{e_{i-1}}{3}, \frac{2 \cdot e_{i-1}}{3} \right]. \tag{33}$$

We have the following property of $\mathcal{J}$.

Claim E.1. For all $i \in [3m]$, we have $3^{i-1} < e_{\mathcal{J}(i)} \le 3^i$, hence $\mathcal{J} : [3m] \to [t-2]$ is an injective mapping.

A proof of the above claim can be found in Section E.3.1. We now fix an $i \in [m]$ and $j := \mathcal{J}(3i) \in [t-2]$ at line 12 in Algorithm 3 and continue our analysis. Note that, by the definition of e_j and Equation (33),

$$d_{j+1} = e_j - e_{j+1} \in \left[\frac{e_j}{3}, \frac{2e_j}{3} \right) \subseteq \left(3^{3i-2}, 2 \cdot 3^{3i-1} \right), \tag{34}$$

where the last containment follows from the fact that $e_j \in (3^{3i-1}, 3^{3i}]$ (by applying Claim E.1 for the index $3i$). From line 21, we have

$$\alpha = \sum_{p=1}^{m} \frac{a_p}{3^{3p}}.$$

Then,

$$\alpha \cdot d_{j+1} = \left(\sum_{p=1}^{m} \frac{a_p}{3^{3p}} \right) \cdot d_{j+1} = \left(\sum_{p=1}^{i-1} \frac{a_p}{3^{3p}} + \frac{a_i}{3^{3i}} + \sum_{p=i+1}^{m} \frac{a_p}{3^{3p}} \right) \cdot d_{j+1} = s_1 + s_2 + s_3,$$

where

$$s_1 := \left(\sum_{p=1}^{i-1} \frac{a_p}{3^{3p}} \right) \cdot d_{j+1},$$

$$s_2 := \left(\frac{a_i}{3^{3i}} \right) \cdot d_{j+1},$$

Algorithm 3 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

and

$$s_3 := \left(\sum_{p=i+1}^{m} \frac{a_p}{3^{3p}}\right) \cdot d_{j+1} \leq \left(\sum_{p=i+1}^{\infty} \frac{1}{3^{3p}}\right) \cdot d_{j+1} \leq \left(\frac{27}{26 \cdot 3^{3(i+1)}}\right) \cdot d_{j+1} \leq \frac{27 \cdot 2 \cdot 3^{3i-1}}{26 \cdot 3^{3(i+1)}} = \frac{1}{39}.$$

(using Equation (34))

Note that $|b_1 - b_0| = b_1 - b_0 = \frac{d_{j+1}}{3^{3i}} \subseteq \left[\frac{1}{9}, \frac{8}{9}\right]$. We now consider two cases $\{b_0\} \in \left[\frac{1}{18}, \frac{17}{18}\right]$ and $\{b_0\} \notin \left[\frac{1}{18}, \frac{17}{18}\right]$ based on the if-else condition at line 15. The following simple claim whose proof can be found in Section E.3.2 will be helpful in analysing these cases.

Claim E.2. For real numbers b_0 and b_1, if $|b_1 - b_0| \in \left[\frac{1}{9}, \frac{8}{9}\right]$, then either $\{b_0\} \in \left[\frac{1}{18}, \frac{17}{18}\right]$ or $\{b_1\} \in \left[\frac{1}{18}, \frac{17}{18}\right]$.

Case 1: $\{b_0\} \in \left[\frac{1}{18}, \frac{17}{18}\right]$. In this case, $a_i = 0$, so $s_2 = 0$. Because $s_1 = b_0$, we have

$$\begin{aligned}\{\alpha \cdot d_{j+1}\} &= \{s_1 + s_2 + s_3\} = \{b_0 + 0 + s_3\} = \{\{b_0\} + s_3\} \\ &= \{b_0\} + s_3 \in \left[\frac{1}{18} - \frac{1}{39}, \frac{17}{18} + \frac{1}{39}\right] = \left[\frac{7}{234}, \frac{227}{234}\right]. \qquad \text{(as } s_3 \leq 1/39\text{)}\end{aligned}$$

Case 2: $\{b_0\} \notin \left[\frac{1}{18}, \frac{17}{18}\right]$. By Claim E.2, $\{b_1\} \in \left[\frac{1}{18}, \frac{17}{18}\right]$. As $a_i = 1$ in this case,

$$\begin{aligned}\{\alpha \cdot d_{j+1}\} &= \{(s_1 + s_2) + s_3\} = \{b_1 + s_3\} = \{\{b_1\} + s_3\} \\ &= \{b_1\} + s_3 \in \left[\frac{1}{18} - \frac{1}{39}, \frac{17}{18} + \frac{1}{39}\right] = \left[\frac{7}{234}, \frac{227}{234}\right]. \qquad \text{(as } s_3 \leq 1/39\text{)}\end{aligned}$$

Thus, in both the cases, we have, for all $i \in [m]$ and $j = \mathcal{J}(3i)$ that,

$$\{\alpha \cdot d_{j+1}\} \in \left[\frac{7}{234}, \frac{227}{234}\right].$$

As k_{j+1} is an integer,

$$\begin{aligned}\left|k_{j+1} - \alpha \cdot d_{j+1}\right| &\geq \left|\left\lfloor \alpha \cdot d_{j+1} \right\rceil - \alpha \cdot d_{j+1}\right| \\ &= \min\left\{\left\{\alpha \cdot d_{j+1}\right\}, 1 - \left\{\alpha \cdot d_{j+1}\right\}\right\} \\ &\geq \frac{7}{234}.\end{aligned}$$

Now let $\mathcal{X} := \{\mathcal{J}(3i) + 1 : i \in [m-2]\} \subseteq [t-1]$. Then the above condition translates to: For all $p \in \mathcal{X}$,

$$|k_p - \alpha \cdot d_p| \geq \frac{7}{234} \tag{35}$$

and Equation (34) implies that

$$d_p \leq 2 \cdot 3^{3(m-2)-1} \leq \frac{d}{3^6}. \tag{36}$$

We thus have

$$\sum_{p=1}^{t} \left|k_p - \frac{k \cdot d_p}{d}\right| \geq \sum_{p \in \mathcal{X}} \left|k_p - \frac{\lfloor \alpha \cdot d \rfloor}{d} \cdot d_p\right| \qquad \text{(using line 22)}$$

$$
\begin{aligned}
&\geq \sum_{p\in\mathcal{X}} |k_p - \alpha\cdot d_p| - \left|\{\alpha\cdot d\}\cdot\frac{d_p}{d}\right| && \text{(using } |x-y|\geq|x|-|y|\text{)}\\
&\geq \sum_{p\in\mathcal{X}} |k_p - \alpha\cdot d_p| - \frac{d_p}{d}\\
&\geq \sum_{p\in\mathcal{X}} \frac{7}{234} - \frac{1}{3^6} && \text{(using Equations (35) and (36))}\\
&\geq \frac{1}{36}\cdot|\mathcal{X}| = \frac{m-2}{36} && \text{(as } \mathcal{J} \text{ is injective)}\\
&= \frac{1}{36}\cdot\left(\left\lfloor\frac{\log_3 d-1}{3}\right\rfloor - 2\right)\\
&\geq \frac{\log_3 d - 10}{108}.
\end{aligned}
$$

□

E.3.1 Proof of Claim E.1

First, the function $\mathcal{J} : [3m] \to [t-2]$ is well-defined: Consider any $i \in [3m]$. Then the set $\{j \in [0..t] : e_j \leq 3^i\}$ is not empty; it contains $t-2$, as $e_{t-2} < 3\cdot e_{t-1} = 3 \leq 3^i$.[22] And, 0 is not contained in this set as

$$e_0 = d > 3^{3\left\lfloor\frac{\log_3 d-1}{3}\right\rfloor} = 3^{3m} \geq 3^i.$$

Let $j := \mathcal{J}(i) \in [t-2]$. We show that $3^{i-1} < e_j$ by contradiction. Suppose that $e_j \leq 3^{i-1}$. From the minimality condition in the definition of $\mathcal{J}$, we have $e_{j-1} > 3^i$. Hence, $e_j \leq 3^{i-1} < \frac{e_{j-1}}{3}$, which contradicts Equation (33). It follows that $\mathcal{J}$ is injective because an integer cannot lie between two different pairs of consecutive powers of 3. □

E.3.2 Proof of Claim E.2

As b_0 and b_1 are close to each other, so are their integer parts. Indeed, $|\lfloor b_1\rfloor - \lfloor b_0\rfloor|$ is 0 or 1. We will only analyse the former case as the latter case can be easily reduced to the former. Hence we have, $|\{b_1\} - \{b_0\}| = |(b_1 - \lfloor b_1\rfloor) - (b_0 - \lfloor b_0\rfloor)| = |b_1 - b_0| \in \left[\frac{1}{9}, \frac{8}{9}\right]$. As $\{b_0\}$ and $\{b_1\}$ are in the range $[0,1]$, if neither of them lies in $\left[\frac{1}{18}, \frac{17}{18}\right]$, the difference between them cannot be in the range $\left[\frac{1}{9}, \frac{8}{9}\right]$. Therefore, at least one of $\{b_0\}$ or $\{b_1\}$ lies in $\left[\frac{1}{18}, \frac{17}{18}\right]$. □

E.4 Proof of Theorem 5.4

Let C be a UPT formula computing $IMM_{n,d}$, $(d_1,\ldots,d_t) := \textsc{Deg-seq}(\mathcal{T}(C))$, and $k := \textsc{Upt-K}(d_1,\ldots,d_t)$. We now delve into the program $\textsc{Upt-K}(d_1,\ldots,d_k)$ in Algorithm 3 to understand the range of the k that it outputs. Notice that for $i=1$, b_0 gets the value 0 (at line 13), so the condition in line 15 fails resulting in the value of a_1 being 1. Hence, $k = \lfloor\alpha\cdot d\rfloor \geq \left(\frac{a_1}{3^3}\right)\cdot d - 1 = \frac{d}{27} - 1 \geq \frac{d}{30}$. On the other hand, $k \leq \alpha\cdot d \leq \left(\sum_{p=1}^{\infty}\frac{1}{3^{3p}}\right)\cdot d \leq \frac{d}{26} \leq \frac{d}{2}$. Hence Lemma 4.6 is applicable with

[22] $e_{t-1} = 1$ is shown in the proof of Lemma 5.1.

$k = \text{UPT-K}(\mathcal{T}(C))$ and $h = \lfloor \log n \rfloor$ – giving a polynomial $P_{\mathbf{w}}$ in $\tilde{n}$ (say) variables such that its SP measure is large. By Lemma B.13, this means that there exists a UPT formula C' of similar size computing $P_{\mathbf{w}}$ (in fact, $\mathcal{T}(C') = \mathcal{T}(C)$).

Lemma 5.2 and Lemma 5.3 when put together give

$$P_{\mathbf{w}} = \sum_{i=1}^{s} Q_{i,1} \dots Q_{i,t}$$

for some $s \leq \text{size}(C')$ such that for all $i \in s$, $\text{residue}_k(\text{DEG-SEQ}(\mathcal{T}(C'))) \geq \Omega(\log d)$. Here we are using the fact that $k = \text{UPT-K}(\text{DEG-SEQ}(\mathcal{T}(C')))$ as $\mathcal{T}(C) = \mathcal{T}(C')$. Now since $\mathbf{w}$ is obtained by Lemma 4.6, applying Lemma 4.3 with $\gamma = \Omega(\log d)$ gives that $s \geq 2^{-O(d)} \cdot n^{\Omega(\log d)}$. Hence, $\text{size}(C) \geq \text{size}(C') \geq s \geq 2^{-O(d)} \cdot n^{\Omega(\log d)}$.

If $d \leq \epsilon \cdot \log n \cdot \log\log n$ for some $\epsilon > 0$, then $\frac{d}{\log d} \leq \frac{\epsilon \cdot \log n \cdot \log\log n}{\log\log n + \log\log\log n} \leq \epsilon' \cdot \log n$ for some $0 < \epsilon' \leq \epsilon$. Hence, $d = \epsilon' \cdot \log n \cdot \log d$ and $2^{-O(d)} \cdot n^{\Omega(\log d)} = n^{\Omega(\log d)}$ if ϵ (and thus ϵ') is a small enough constant. □

E.5 Proof of Theorem 5.6

Let C be any UPT formula computing P, $(d_1, \dots, d_t) := \text{DEG-SEQ}(\mathcal{T}(C))$, and $k := \text{UPT-K}(d_1, \dots, d_t)$. Observe that the formula C' obtained by setting $y_i = 0$ for all $i \in [d] \setminus \{k\}$ in C computes $NW_{q,d,k}(\mathbf{x})$. It follows from the proof of Lemma 5.2 that not only is C' a UPT formula, but also that its canonical parse tree $\mathcal{T}(C')$ is the same $\mathcal{T}(C)$. Hence, $(d_1, \dots, d_t) := \text{DEG-SEQ}(\mathcal{T}(C'))$ and $k = \text{UPT-K}(\text{DEG-SEQ}(\mathcal{T}(C')))$. Then, arguing as in the proof of Theorem 5.4, we get that $\text{size}(C') \geq (qd)^{\Omega(d)} = n^{\Omega(d)}$. As $\text{size}(C') \leq \text{size}(C)$, this proves the theorem. □

F Large-degree set-multilinear lower bound (using [LST21])

One of the primary motivations for our alternate approach of the lower bounds of [LST21] is based on a hope that we can achieve exponential constant-depth homogeneous formula lower bounds (rather than simply superpolynomial) and resolve Open Problem 1.2 from Section 1. A necessary condition for us to be able to achieve this is to at least get such lower bounds for *set-multilinear* constant-depth formulas. Although exponential set-multilinear (or even multilinear) constant-depth lower bounds already follow from [RY09, CLS19, CELS18], these techniques do not give non-FPT lower bounds, unlike [LST21].

In this section, we show how to obtain an exponential non-FPT lower bound against constant-depth set-multilinear formulas by making a small adjustment to the lower bound proof in [LST21]. However, we note that this change does not allow us to retain the iterated matrix multiplication as the hard polynomial, instead we are only able get the lower bound for some polynomial in VNP. For the sake of simplicity, we shall present the idea only for depth-5 set-multilinear formulas; the parameters that work for larger depths are similar to the parameters we use in Section 4. We will assume that the reader is familiar with the original paper [LST21], where the authors handle degree at most $O(\log^2 n)$ (Lemma 13 in [LST21]), whereas the following claim handles the full range of degrees up to n.

Claim F.1. For any integers $d \leq n$, there is a set-multilinear polynomial (family) $P \in \mathsf{VNP}$ of degree d over at most $N = nd$ variables such that any set-multilinear formula of product-depth 2 computing it has size $n^{\Omega(\sqrt{d})}$. In particular, this gives a $N^{\Omega(N^{1/4})}$ lower bound.

Proof. The main idea to handle a larger degree is that the set sizes need not be powers of 2 if we are content with proving the lower bound for some polynomial in VNP, and not necessarily for the IMM polynomial (or its projection).

Let $\mathbf{x} = \mathbf{x}_1 \sqcup \cdots \sqcup \mathbf{x}_d$ be the variable sets; of which some are "positive" and the rest are "negative" sets; in particular, let $\mathbf{w} = (w_1, \ldots, w_d)$ be a tuple of (not necessarily integers) formed by h and $-h_0$ such that all the prefix sums are at most h in absolute value. According to the signs of the w_i's, we classify the variable sets as positive or negative, and fix the sizes of the positive sets to be $n = 2^h$, and that of negative sets to be $n_0 = 2^{h_0}$. We set $h = \log n$ and $h_0 = h - c \cdot h/\sqrt{d}$ for an appropriate constant $c \in [0.9, 1]$ that we fix later so that n_0 is an integer.

Let $\mathcal{M}_+$ be the set-multilinear monomials over the positive sets and similarly $\mathcal{M}_-$ be the "negative monomials", both sets ordered in lexicographical order, where we express a monomial in a canonical way by ordering the variables in increasing order of the corresponding variable set indices. Then we define

$$P := \sum_{\substack{m_+ \in \mathcal{M}_+,\, m_- \in \mathcal{M}_- \\ \mathrm{rank}(m_+) = \mathrm{rank}(m_-)}} m_+ \cdot m_-,$$

where $\mathrm{rank}(m)$ is the position of the positive or negative monomial m in the corresponding monomial set. It is not hard to show that $P \in \mathsf{VNP}$ and $\mathsf{relrk}(P) \geq 2^{-O(h)} = n^{-O(1)}$. By following the same proof of Claim 14 in [LST21], we conclude that any product-depth 2 set-multilinear formula C must satisfy $\mathsf{relrk}(C) \leq \frac{\mathrm{size}(C)}{n^{\Omega(\sqrt{d})}}$. Hence, $\mathrm{size}(C) \geq n^{\Omega(\sqrt{d})}$ if C computes P.

All that remains to show is that there exists a value of $c \in [0.9, 1]$ such that $n_0 = 2^{h_0} = 2^{h - c \cdot h/\sqrt{d}}$ is an integer. This is true because n_0 is a monotonous real function of c with range

$$\begin{aligned}
2^{h - 0.9h/\sqrt{d}} - 2^{h - h/\sqrt{d}} &= 2^{h\left(1 - 1/\sqrt{d}\right)} \left(2^{\frac{0.1h}{\sqrt{d}}} - 1\right) \\
&\geq 2^{h\left(1 - 1/\sqrt{d}\right)} \left(\frac{0.1h}{\log e\sqrt{d}}\right) \\
&\geq n^{1 - 1/\sqrt{d}}/\sqrt{d} && \text{(assuming } h \text{ is large enough)} \\
&\geq n^{1 - 1/\sqrt{d}}/\sqrt{n} \\
&\geq 1.
\end{aligned}$$

As the range of n_0 is at least 1, it must take an integral value for some setting of c. □

G Geometric intuition behind SP and APP measures

In this section, we give the geometric intuition which led to the development of the shifted partials measure $\mathsf{SP}_{k,\ell}$ and the affine projection of partials measure APP_{k,n_0}. We will see, however, that this intuition breaks down when k is *large* and prior lower bounds did indeed only use small values of k. By using higher values of k here, we depart from this geometric intuition.

Preliminaries: the algebra-geometry dictionary. Let $\mathbb{V} \subseteq \mathbb{F}^n$ be the zero set of polynomials $\{g_1(\mathbf{x}), g_2(\mathbf{x}), \ldots, g_s(\mathbf{x})\} \subseteq \mathbb{F}[\mathbf{x}]^{=d}$. Let $P_\ell(\mathbb{V}) \subseteq \mathbb{F}[\mathbf{x}]^{=\ell}$ be the vector space of polynomials of degree ℓ which vanish at every point in $\mathbb{V}$. Now, if $\mathbb{V}$ has a small codimension, then $\mathbb{V}$ *contains a lot of* points which in turn imposes a lot of (linear) constraints on $P_\ell(\mathbb{V})$ and so intuitively we should have that:

Proposition G.1. (Informal/Qualitative). If codimension of $\mathbb{V}$ is small then dimension of $P_\ell(\mathbb{V})$ is also *small*.

Now notice that for all $\ell \geq d$, the set of polynomials $\mathbf{x}^{(\ell-d)} \cdot \{g_1(\mathbf{x}), g_2(\mathbf{x}), \ldots, g_s(\mathbf{x})\}$ is contained in $P_\ell(\mathbb{V})$ and so intuitively we should have:

Proposition G.2. (Informal/Qualitative). If codimension of $\mathbb{V}$ is small then dimension of $\mathbf{x}^{(\ell-d)} \cdot \{g_1(\mathbf{x}), g_2(\mathbf{x}), \ldots, g_s(\mathbf{x})\}$ is also *small*[23].

We also have that

Fact G.3. If $L \subseteq \mathbb{F}^n$ is a random linear subspace, then the codimension of $\mathbb{V} \cap L$ inside L is the same as the codimension of $\mathbb{V}$ inside $\mathbb{F}^n$.

We can use the above intuition to formulate the shifted partials measure $\mathsf{SP}_{k,\ell}$ and the affine projection of partials measure APP_{k,n_0} as follows.

Formulating the measures. Arithmetic circuits and formulas admit various depth reductions and in particular, can be reduced to depth 4 without increasing the size *too much*. Consider a term T of such a depth 4 formula. Specifically, let $T(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ be a product of t polynomials, i.e. $T = Q_1(\mathbf{x}) \cdot Q_2(\mathbf{x}) \cdot \ldots \cdot Q_t(\mathbf{x})$. First observe that for any $k < t$, the zero set of k-th partials of T, denoted $\mathbb{V}(\partial^k T)$ is of codimension $(k+1)$. The above intuition then suggests that

$$\mathsf{SP}_{k,\ell} := \dim \left\langle \mathbf{x}^\ell \cdot \partial^k T \right\rangle$$

should also be small. Also notice that for a subspace $L \subseteq \mathbb{F}^n$, $\pi_L\left(\partial^k T\right)$ is the set of polynomials whose zero set is $L \cap \mathbb{V}(\partial^k T)$ and so for such a T

$$\mathsf{APP}_{k,n_0}(P) := \max_{L:\mathbf{x}\to\langle\mathbf{z}\rangle} \dim \left\langle \pi_L\left(\partial^k P\right) \right\rangle$$

should be small as well. Prior work [GKKS14, KSS14, GKS20] showed that when the degree of the factors of T are also small, then these measures are indeed significantly small for such a T (as compared to a random polynomial), thereby leading to lower bounds for certain classes of depth 4 formulas (and also for some subclasses of higher depth formulas via appropriate depth reductions). But notice that the geometric intuition behind these measures holds only when k is less than the number of factors t. In particular when $k \geq t$ then $\mathbb{V}(\partial^k T)$ could be empty[24] and so its no longer clear why either $\mathsf{SP}_{k,\ell}(T)$ or $\mathsf{APP}_{k,n_0}(T)$ should be small in this regime. Our conceptual contribution is that even when $k > t$ then $\mathsf{SP}_{k,\ell}(T)$ and $\mathsf{APP}_{k,n_0}(T)$ can be small (this depends on $\gamma(k)$) and this can be used to more directly obtain the lower bounds in [LST21].

[23] An asymptotic statement that captures this qualitative intuition is obtained through the notion of the Hilbert polynomial of the variety $\mathbb{V}$.

[24] For example when $T = (x_1^3 + x_2^3 + \ldots + x_n^3)^t$, then $\mathbb{V}(\partial^k T)$ is empty for all $k \geq t$.