Electronic Colloquium on Computational Complexity, Report No. 90 (2022)

On the Partial Derivative Method Applied to Lopsided Set-Multilinear Polynomials

Sébastien Tavenas* Srikanth Srinivasan† Nutan Limaye‡

June 24, 2022

Abstract

We make progress on understanding a lower bound technique that was recently used by the authors to prove the first superpolynomial constant-depth circuit lower bounds against algebraic circuits.

More specifically, our previous work applied the well-known partial derivative method in a new setting, that of *lopsided* set-multilinear polynomials. A set-multilinear polynomial $P \in \mathbb{F}[X_1, \ldots, X_d]$ (for disjoint sets of variables $X_1, \ldots, X_d$) is a linear combination of monomials, each of which contains one variable from $X_1, \ldots, X_d$. A lopsided space of set-multilinear polynomials is one where the sets $X_1, \ldots, X_d$ are allowed to have different sizes (we use the adjective 'lopsided' to stress this feature). By choosing a suitable lopsided space of polynomials, and using a suitable version of the partial-derivative method for proving lower bounds, we were able to prove constant-depth superpolynomial set-multilinear formula lower bounds even for very low-degree polynomials (as long as d is a growing function of the number of variables N). This in turn implied lower bounds against general formulas of constant-depth.

A priori, there is nothing stopping these techniques from giving us lower bounds against algebraic formulas of *any* depth. We investigate the extent to which this lower bound can extend to greater depths. We prove the following results.

1. We observe that our choice of the lopsided space and the kind of partial-derivative method used can be modeled as the choice of a multiset $W \subseteq [-1, 1]$ of size d. Our first result completely characterizes, for any product-depth Δ, the best lower bound we can prove for set-multilinear formulas of product-depth Δ in terms of some combinatorial properties of W, that we call the *depth-Δ tree bias* of W.
2. We show that the maximum depth-3 tree bias, over multisets W of size d, is $\Theta(d^{1/4})$. This shows a stronger formula lower bound of $N^{\Omega(d^{1/4})}$ for set-multilinear formulas of product-depth 3, and also puts a non-trivial constraint on the best lower bounds we can hope to prove at this depth in this framework (a priori, we could have hoped to prove a lower bound of $N^{\Omega(\Delta d^{1/\Delta})}$ at product-depth Δ).
3. Finally, we show that for small Δ, our proof technique cannot hope to prove lower bounds of the form $N^{\Omega(d^{1/\operatorname{poly}(\Delta)})}$. This seems to strongly hint that new ideas will be required to prove lower bounds for formulas of unbounded depth.

*Univ. Grenoble Alpes, Univ. Savoie Mont Blanc, CNRS, LAMA. Email: `sebastien.tavenas@univ-smb.fr`.

†Department of Computer Science, Aarhus University. Email: `srikanth@cs.au.dk`. On leave from Department of Mathematics, IIT Bombay, India.

‡ITU-Copenhagen. Email: `nuli@itu.dk`.

ISSN 1433-8092

1 Introduction and Motivation

Basic background. This paper is motivated by questions arising in the area of *Algebraic Circuit complexity,* which studies the computational complexity of problems defined by families of multivariate polynomials. Given an infinite family of polynomials $(P_N(x_1,\ldots,x_N))_{N\geqslant 1}$ over a field $\mathbb{F}$, we consider the computational problem of evaluating P_N at input point $a\in\mathbb{F}^N$. Many natural and important computational problems can be stated in this language, including the problems of computing the determinant and the permanent, and that of multiplying matrices.

Algebraic circuits are succinct representations of multivariate polynomials that allow us to solve computational problems of the above form. More precisely, an algebraic circuit is a directed acyclic graph, where the sources are labelled by variables $x_1,\ldots,x_N$ or field elements and internal nodes (or *gates*) by algebraic operations $+$ and $\times$. Each internal node thus represents a polynomial in the variables $x_1,\ldots,x_N$ and a designated output gate represents the polynomial computed by the algebraic circuit. The *size* of the algebraic circuit is given by the number of gates. The *depth* and *product-depth* of an algebraic circuit denote the maximum number of gates and $\times$-gates respectively, on a directed path in the circuit.[1] Finally, we call an algebraic circuit an *Algebraic formula* if the underlying directed graph is a tree. (Equivalently, an Algebraic formula is just a nested algebraic expression made up of additions and multiplications, as one might write down on paper, represented in the form of a tree.)

An algebraic circuit for a polynomial P allows us to evaluate the polynomial P on a given input in time polynomially related to the size of the circuit. Thus, algebraic circuits are a restricted, but natural, model of computation for computational problems of this form. The study of this model of computation is one of the principal topics of study in Algebraic circuit complexity, and has received much attention over the past four decades (see e.g. [BCS97, SY10, Sap15] for nice introductions). Many central questions in Boolean circuit complexity have analogous and closely-related versions in the algebraic setting. For instance, the VP vs. VNP question [Val79], which is the problem of proving explicit lower bounds against algebraic circuits, is formally easier than the (non-uniform) P vs. NP question [Bü00]. The problem of proving lower bounds against algebraic formulas is similarly closely related to the problem of proving lower bounds against the Boolean complexity class NC^1.

A recent result [LST22]. While circuit lower bounds in the algebraic setting are formally easier than the Boolean setting, they still have been hard to come by. For example, a famous line of research in the 1980s [Ajt83, FSS84, Hå86, Raz86, Smo87] showed exponential lower bounds against Boolean circuits of constant-depth, but did not yield such results for algebraic circuits.[2] This situation was somewhat rectified recently by the authors [LST22], building on some important earlier results in the area [NW97, Raz09]. In particular, we were able to prove superpolynomial lower bounds against constant-depth algebraic circuits over fields of characteristic zero.

This paper is motivated by the problem of extending this lower bound to stronger models of computation. At a high level, our results are as follows.

- We show that our previous result [LST22] can be formulated purely in terms of a combinatorial property of the space of polynomials under consideration.

[1] W.l.o.g., we may assume that the product-depth and depth of a circuit are related to each other by a multiplicative factor of 2. However, some results are easier to state in terms of product-depth.

[2] Note that algebraic circuit lower bounds are not necessarily easier than Boolean circuit lower bounds in the constant-depth setting. However, some of these ideas did translate in the setting of constant-sized fields. [GK98, GR00]

- We characterize the best lower bound that can be achieved in this framework at product-depth 3. It is better than the analogous lower bound from [LST22], but not as good as one might hope at first sight (as explained below).
- We place limitations on how well the bound extends to higher depths.

To describe these results in more detail, we first need to recall the outline of the proof of [LST22].

The proof of [LST22]. The proof of [LST22] proceeds in two steps. In the first step, we reduce the problem of proving lower bounds for general circuits of depth Δ to proving lower bounds for *product-depth*-$(\Delta - 1)$ circuits that have a special structure. In the second step, we prove lower bounds for the structured circuits. We describe these steps in some more detail next.

Step 1: Set Multilinearization. We work throughout with a partition of the variable set $X = \{x_1, \ldots, x_N\}$ into $X_1 \cup X_2 \cup \cdots \cup X_d$. Given such a partition, a *set-multilinear* monomial w.r.t. this variable partition is a monomial of degree d that contains exactly one variable from each of $X_1, X_2, \ldots, X_d$. A set-multilinear polynomial P is a linear combination of set-multilinear monomials. We denote the space of set-multilinear polynomials w.r.t. $X_1, \ldots, X_d$ by $\mathbb{F}_{\mathrm{sm}}[X_1, \ldots, X_d]$. A set-multilinear circuit or formula is one where each gate computes a set-multilinear polynomial w.r.t. a subset of $\{X_1, \ldots, X_d\}$. An important example of a set-multilinear polynomial is the *Iterated Matrix Multiplication* polynomial $\mathrm{IMM}_{n,d}$, where $X_1, \ldots, X_d$ are square matrices of dimension $n \times n$ with distinct indeterminates, and the polynomial represents, say, the $(1, 1)$th entry of the product of these matrices.

In the first step of the proof, we show that if a polynomial $P \in \mathbb{F}_{\mathrm{sm}}[X_1, \ldots, X_d]$ has a circuit C of depth Δ and size s, then it also has a set-multilinear circuit C' of product-depth $\Delta - 1$ and size $s' = \mathrm{poly}(s) \cdot d^{O(d)}$. Note that while the blow-up in size in going from C to C' is large as a function of d, it can be made small (say $\mathrm{poly}(N)$) assuming that d is a slow-growing function of N (say, $d = O(\log N / \log\log N)$). So, to prove superpolynomial constant-depth circuit lower bounds, it suffices to prove superpolynomial lower bounds for constant-depth set-multilinear circuits in this *low-degree setting.*

Step 2: Set-multilinear lower bounds for low-degree polynomials. Lower bounds for constant-depth set-multilinear circuits have been known since the work of Nisan and Wigderson [NW97] from the 1990s. However, such lower bounds were typically of the form $\exp(d^{\Omega(1)}) \cdot \mathrm{poly}(N)$, which are not good enough for our purposes in the low-degree setting. The main contribution of [LST22] was to prove a lower bound of the form $N^{\omega_d(1)}$, which yields a superpolynomial lower bound for any degree $d = d(N)$ which is a growing function of N.

Somewhat surprisingly, the proof of this latter lower bound used just the lower bound technique of Nisan and Wigderson [NW97], which goes by the name of the *partial derivative method.* The key observation was to apply this technique to a suitable space of set-multilinear polynomials. Specifically, it is crucial in the proof to allow for the sets $X_1, \ldots, X_d$ to have fairly different sizes. To stress this feature, we refer to such a space of set-multilinear polynomials as *lopsided.*

For such polynomials that have efficient small-depth set-multilinear formulas, we argue that certain matrices associated to these polynomials have low rank. This is the basic recipe suggested by the partial derivative method, and is described in more detail later.

To complete the argument, we need to find explicit polynomials for which the associated matrices have high (ideally maximal) rank. We do this by considering suitable restrictions of $\mathrm{IMM}_{n,d}$ where $n = \max_{i \in [d]} |X_i|$. Using this idea, we showed [LST22] a lower bound of $N^{d^{\exp(-O(\Delta))}}$ for set-multilinear circuits of product-depth Δ. In conjunction with Step 1, this implies a superpolynomial lower bound for constant-depth algebraic circuits, and in fact for circuits of depth $o(\log \log d)$.

The potential of this lower bound technique. Can the above proof strategy be used to prove lower bounds for stronger models of computation, such as algebraic formulas of unbounded depth or, optimistically, even algebraic circuits? It turns out that Step 1 of the strategy still works, as shown in previous work of Nisan and Wigderson [NW97] and Raz [Raz13]. Consequently, proving superpolynomial set-multilinear lower bounds against these models in the low-degree setting imply general circuit or formula lower bounds.

However, a problem arises because of the technique used in Step 2. As $\mathrm{IMM}_{n,d}$ (or more precisely, its restrictions) is a polynomial of 'maximal complexity' for the partial derivative method, we cannot use it to prove lower bounds for computational models that can compute this polynomial efficiently. In particular, this suggests a new idea is required to prove lower bounds for, say, set-multilinear circuits of depth $O(\log d)$, which can compute $\mathrm{IMM}_{n,d}$ efficiently.

Nevertheless, this does not seem to rule out lower bounds for circuits of depth $o(\log d)$, or for formulas (of any depth). A simple, folklore divide-and-conquer strategy shows that $\mathrm{IMM}_{n,d}$ has set-multilinear circuits of product-depth Δ and size $n^{O(d^{1/\Delta})}$, and also set-multilinear formulas of product-depth Δ and size $n^{O(\Delta d^{1/\Delta})}$. Given the fact that this basic bound has not been improved upon significantly[3] for a long time, it is tempting to conjecture that it is tight, at least in the set-multilinear setting. If so, it seems that we could hope to prove lower bounds for set-multilinear circuits of depth $o(\log d)$ and formulas of any depth. Doing this would yield at least lower bounds for general algebraic formulas, which would be a very interesting result. This brings us to our main motivating question.

Question 1. *Can we hope to use the partial derivative method (as applied to lopsided spaces of set-multilinear polynomials) to prove set-multilinear lower bounds that match the standard divide and conquer algorithms for* $\mathrm{IMM}_{n,d}$*?*

Our results in this paper indicate that the answer to this question is probably 'No', and that, alone, the proof technique from [LST22] is not powerfull enough to handle formulas of depth $(\log d)^{o(1)}$. In the process of proving these results, we also introduce what we believe is a clean framework for studying the power of this technique.

We start with a more formal description of the partial derivative method and then state our results.

The partial derivative method for lopsided set-multilinear polynomials. We prove lower bounds for set-multilinear polynomials $P(X_1, \ldots, X_d)$ where each $|X_i| = n^{\alpha_i}$ for some $\alpha_i \in (0, 1]$. Given such a polynomial P, we associate with it a matrix as follows. We partition $[d]$ into sets $\mathcal{P}$ and $\mathcal{N}$. The rows of the matrix are associated with set-multilinear monomials over the variable sets $\{X_i \ : \ i \in \mathcal{P}\}$, and the columns symmetrically with the set-multilinear monomials over $\{X_j \ : \ j \in \mathcal{N}\}$. Given a row labelled by monomial m_1 and a column labelled by monomial m_2, the corresponding entry in the matrix is the coefficient of the set-multilinear monomial $m_1 m_2$ in the polynomial P. We use the rank of this matrix (or, more precisely, how close it is to full-rank) to prove lower bounds on the algebraic circuit complexity of P.

[3] A famous result of Gupta, Kamath, Kayal and Saptharishi [GKKS16] does improve this bound, but gives up on set-multilinearity. Moreover, the basic form of the bound is still preserved. More precisely, their work implies circuits of product-depth Δ and size $n^{O(d^{1/2\Delta})}$.

We define this more precisely now. Note that the matrix is completely specified by the choice of the numbers $\alpha_1, \ldots, \alpha_d$ and the partition $[d] = \mathcal{P} \cup \mathcal{N}$. We can describe these together by the multiset $W \subseteq [-1, 1]$, defined by $W = \{\alpha_i : i \in \mathcal{P}\} \cup \{-\alpha_j : j \in \mathcal{N}\}$. Finally, we use $M_W(P)$ to denote the above matrix.

Note that $M_W(P)$ is a matrix with $R = n^{\sum_{\alpha \in W \cap (0,1]} \alpha}$ rows and $C = n^{\sum_{\alpha \in W \cap [-1,0)} |\alpha|}$ columns. In particular, the rank of the matrix $M_W(P)$ is bounded by the minimum of these quantities. We consider the *relative rank* of P, defined as follows.

$$\mathrm{relrk}_W(P) = \frac{\mathrm{rank}(M_W(P))}{\sqrt{RC}} = \frac{\mathrm{rank}(M_W(P))}{n^{\frac{1}{2}\sum_{\alpha \in W} |\alpha|}}. \tag{1}$$

Observe that the quantity in the denominator is the geometric mean of the number of rows and the number of columns of $M_W(P)$ and hence $\mathrm{relrk}_W(P) \in [0, 1]$. In fact, more generally, it is not hard to see that as $\mathrm{rank}(M_W(P)) \leqslant \min\{R, C\}$, we have $\mathrm{relrk}_W(P) \leqslant n^{-|\sum_{\alpha \in W} \alpha|/2}$.

Further, it was shown by the authors [LST22] that for any W, there is a polynomial P_0 such that $\mathrm{relrk}_W(P_0) = n^{-|\sum_{\alpha \in W} \alpha|/2}$ and P_0 can be obtained by starting with an instance of $\mathrm{IMM}_{\mathrm{poly}(n),d}$ and setting some variables to 0 and identifying variables within certain submatrices, i.e. by a *set-multilinear projection.*

High-level description of the results. Our results give a better understanding of what lower bounds the partial derivative method can hope to show in this setting.

- Our first main result is a transformation of our problem to a combinatorial problem about labelled trees. More precisely, we show that understanding the best lower bound our techniques can hope to prove in the low-degree setting is perfectly captured by the best-possible "tree-like decomposition" of the set W.[4]

  While this transformation is simple, it is conceptually clean, and simplifies the problem in multiple ways. Firstly, it eliminates the parameter n (which is roughly the number of variables in the underlying polynomial) and makes completely clear the dependence of the lower bound on properties of the multiset W. Secondly, this reformulation of the problem completely eliminates any mention of polynomials or algebra from the problem. It is now purely a problem about the 'additive structure' of W.

- Our second result uses the above characterization of the problem to give a near-perfect understanding of the best lower bounds we can prove for set-multilinear formulas of product-depth 3 (i.e. $\Sigma\Pi\Sigma\Pi\Sigma\Pi\Sigma$ formulas). More precisely, we show that the best product-depth-3 lower bound we can prove via our proof technique is $n^{\Theta(d^{1/4})}$. This is interesting for the following two different reasons.

  For one, this is a stronger lower bound than known previously for set-multilinear formulas of product-depth 3 in the low-degree regime: Nisan and Wigderson [NW97] showed a lower bound of $\exp(\Omega(d^{1/3})) \cdot \mathrm{poly}(N)$ (which does not yield anything for $d = O(\log N)$), while in our earlier work [LST22], we showed lower bounds of $n^{\Omega(d^{1/7})}$.

  On the other hand, the result also implies that this technique does not go as far as we would like. Recall from above that the (suspected) optimal lower bound for $\mathrm{IMM}_{n,d}$ at product-depth 3 is $n^{\Omega(d^{1/3})}$. So, our result implies that this technique cannot be used to obtain this bound at product-depth 3.

[4] This is not to be confused with standard tree decompositions of graphs, which have no connection with objects studied here.

- The above results already indicate that we are not able to prove the best possible lower bound we could hope for product-depth-3 set-multilinear formulas. However, it is still conceivable that we can hope to prove a lower bound which stays 'close' to the right expected bound for $\mathrm{IMM}_{n,d}$ (say a bound of the form $n^{\Delta d^{\Omega(1/\Delta)}}$), which could as yet lead to superpolynomial formula lower bounds.

  In our third result, we give strong indication that this is not the case, by showing that this technique cannot prove lower bounds of the form $n^{d^{1/\Gamma(\Delta)}}$ for a quasipolynomial function $\Gamma(\cdot)$, and small enough Δ.

1.1 Formal description of the results.

To describe the results formally, we introduce a combinatorial measure of the complexity of the multiset $W \subseteq [-1,1]$. In the low-degree setting, this will characterize the best lower bound we can prove via our lower bound technique.

Notation. Let $W \subseteq \mathbb{R}$ be a multiset. Throughout $|W|$ denotes the size of the multiset (i.e. counted with multiplicity) and $\mathrm{Sum}(W)$ denote the sum of its elements. Finally, $\|W\|_1$ denotes the L_1-norm of W (i.e. the sum of the absolute values of the elements of W).

Definition 2 (W-trees, path bias, tree bias)**.** *Let $W = \{\alpha_1, \ldots, \alpha_d\}$ be a multiset contained in $[-1,1]$. A* W-tree T*, or equivalently* a tree T for W*, is a rooted, directed tree*[5] *with $d = |W|$ leaves which are labelled by distinct elements of the form (i, α_i) $(i \in [d])$.*[6] *Any vertex v of T thus corresponds to a subset W_v of W (corresponding to the leaves of the subtree induced by v) and we define* $\mathrm{Sum}(v)$ *to be* $\mathrm{Sum}(W_v)$*.*

An internal path π *in T is a path from the root to an internal (i.e. non-leaf) node. Given such an internal path π, we define the set of* Off-path nodes *of π, denoted* $\mathrm{Offpath}(\pi)$ *to be the set of nodes v of the tree T that are not on the path π, but have a parent on the path π. We define* the bias of the path π*, denoted* $\mathrm{bias}(\pi) = \left(\sum_{v \in \mathrm{Offpath}(\pi)} |\mathrm{Sum}(v)|\right) - |\mathrm{Sum}(r)|$ *where r is the root of T.*

(It is easy to check that if π is any internal path, then $W = W_r$ is the disjoint union of W_v $(v \in \mathrm{Offpath}(\pi))$. Hence, by the triangle inequality, we have $|\mathrm{Sum}(r)| \leqslant \sum_{v \in \mathrm{Offpath}(\pi)} |\mathrm{Sum}(v)|$. Thus, $\mathrm{bias}(\pi) \geqslant 0$ *for any internal path π.)*

Finally, we define the path bias of T w.r.t. W*, denoted* $\mathrm{Pathbias}_W(T)$*, to be the* maximum *bias of any internal path of T. If the tree T has depth 0 (i.e. it consists of just the root node), then we define the path bias of T w.r.t. W to be 0.*

With the above notation in place, we can define the combinatorial measure mentioned above. We define the depth-Δ tree bias of W *to be the* minimum *path bias of any depth-Δ W-tree T. We denote this quantity by* $\mathrm{Treebias}_\Delta(W)$*.*

Our first theorem relates the depth-Δ tree bias of $W = \{\alpha_1, \ldots, \alpha_d\} \subseteq [-1,1]$ with the best lower bound we can prove using the complexity measure $\mathrm{relrk}_W(\cdot)$.

Theorem 3 (Connecting tree bias with relative rank)**.** *Let n, d be positive integer parameters.*[7] *Let $\Delta \geqslant 1$ be any integer. Assume $W \subseteq [-1,1]$ is a multiset of size d such that* $\mathrm{Treebias}_\Delta(W) = t$*. Then, for any set-multilinear formula F of product-depth at most Δ and size at most s, we have*

$$\mathrm{relrk}_W(F) \leqslant (d^{3d} \cdot s \cdot n^{-t/2}) \cdot n^{-|\mathrm{Sum}(W)|/2}.$$

[5] The edges are directed away from the root.

[6] We require the label to be a pair here as W is a multiset where elements may repeat. If the elements of W are all distinct, then we can think of the labels as simply elements of W.

[7] We think of d as a slow-growing function of n.

Conversely, for any n and d, there is a set-multilinear formula F with at most $3^d n^{t/2}$ leaves and of product-depth Δ such that $\mathrm{relrk}_W(F) \geqslant 2^{-d} \cdot n^{-|\mathrm{Sum}(W)|/2}$.

This theorem is the consequence of Lemmas 13 and 14 and will be proved in Section 3. As already noted above, for any polynomial $P \in \mathbb{F}_{sm}[X_1, \ldots, X_d]$ (with $|X_i| = n^{|\alpha_i|}$ for each $i \in [d]$), we have $\mathrm{relrk}_W(P) \leqslant n^{-|\mathrm{Sum}(W)|/2}$. Theorem 3 shows that this maximum possible relative rank can be achieved by product-depth-Δ formulas of size $n^{O(t)}$, but not those of size $n^{o(t)}$, where $t = \mathrm{Treebias}_\Delta(T)$. This means that the best lower bound we can hope to prove via this technique is $n^{\Theta(t)}$.

The next couple of theorems give an understanding of the maximum possible tree bias for various depths Δ. The first result gives tight bounds on the maximum possible tree bias of a given multiset W for depth 3 (Section 4 will be dedicated to this result).

Theorem 4 (Tight bounds on tree bias for depth 3)**.** *Let d be a growing integer parameter. Then,*

$$\max_W \mathrm{Treebias}_3(W) = \Theta(d^{1/4})$$

where W ranges over multisets from $[-1, 1]$ of size d in the expression above.

The second result (proved in Section 5) gives an asymptotic bound for larger depths (as long as Δ is bounded by a small function of d).

Theorem 5 (Bounds on tree bias for larger depths)**.** *Let d, Δ be growing integer parameters with $\Delta = 2^{o(\sqrt{\log\log d})}$. Then, we have*

$$\max_W \mathrm{Treebias}_\Delta(W) \leqslant d^{1/\Delta^{\Omega(\log \Delta)}},$$

where W ranges over multisets from $[-1, 1]$ of size d.

1.2 Proof Outline

Throughout this section, we work with a multiset $W = \{\alpha_1, \ldots, \alpha_d\} \subseteq [-1, 1]$ and a space of lopsided set-multilinear polynomials $\mathbb{F}_{\mathrm{sm}}[X_1, \ldots, X_d]$ where $|X_i| = n^{|\alpha_i|}$. Recall also that we are working in the low-degree setting, i.e. d is a slow-growing function of n. All formulas in this section should be assumed to be set-multilinear.

Motivation for tree bias. We start by motivating the notion of tree bias which, at first sight, might appear mysterious to the reader. In fact, this notion comes up quite naturally in the course of constructing small set-multilinear formulas that have large relative rank. These constructions, in turn, are motivated by the following basic properties of relative rank which are all slight modifications of standard facts used in the literature. In this form they can be found in our earlier work [LST].[8]

Lemma 6 (Properties of Relative Rank)**.**

1. *(Imbalance) Say $P \in \mathbb{F}_{\mathrm{sm}}[X_1, \ldots, X_d]$. Then,* $\mathrm{relrk}_W(P) \leqslant n^{-|\mathrm{Sum}(W)|/2}$.
2. *(Sub-additivity) Say $P, Q \in \mathbb{F}_{\mathrm{sm}}[X_1, \ldots, X_d]$. Then* $\mathrm{relrk}_W(P+Q) \leqslant \mathrm{relrk}_W(P) + \mathrm{relrk}_W(Q)$.

[8] The paper deals with a related notion of relative rank w.r.t. *ordered* W (or equivalently, W is replaced by a tuple $(\alpha_1, \ldots, \alpha_d)$). However, the proof works in the same way for multisets.

3. *(Multiplicativity) Say $P = P_1 \cdot P_2 \cdot \ldots \cdot P_t$ and assume that for each $i \in [t]$, $P_i \in \mathbb{F}_{\mathrm{sm}}[X_j : j \in S_i]$, where $\{S_1, \ldots, S_t\}$ is a partition of $[d]$. Then*

$$\mathrm{relrk}_W(P) = \mathrm{relrk}_W(P_1 \cdot P_2 \cdot \ldots \cdot P_t) = \prod_{i \in [t]} \mathrm{relrk}_{W_i}(P_i)$$

where $W_i = \{\alpha_j \mid j \in S_i\}$.

With these properties in mind, we try to construct small set-multilinear formulas with optimally large relative rank. We do not lose much generality in assuming that $\mathrm{Sum}(W) \approx 0$, which we will do in the rest of this proof outline. So, the optimal relative rank is 1.

It is instructive to consider the example of W such that $\alpha_1 = \cdots = \alpha_{d/2} = 1$ and $\alpha_{d/2+1} = \cdots = \alpha_d = -1$. We start with a trivial formula F that consists of a single variable $x_1 \in X_1$, which has relative rank $n^{-1/2}$. Does it make sense to take linear combinations of such formulas? From the perspective of relative rank, the answer is No, because that increases the size without increasing the relative rank at all, by the Imbalance criterion in Lemma 6. So we can only multiply variables (from different sets, as we are dealing with set-multilinear formulas). Moreover, it makes sense to multiply variables such that the corresponding α_is have different signs, as multiplying variables from X_1 and X_2 (say) would only make the imbalance worse. So we multiply $x_1 \in X_1$ and $x_{d/2+1} \in X_{d/2+1}$. This creates a formula of relative rank $1/n$, by the property of Multiplicativity. By Sub-additivity, we need to sum at least n such formulas to get a formula of relative rank 1 (which is optimal). And indeed, this can be done, say, with an inner product between the variables of X_1 and $X_{d/2+1}$. Multiplying $d/2$ such formulas together (for a partition of $\alpha_1, \ldots, \alpha_d$ into positive and negative pairs) gives us a formula of product-depth 2, size $O_d(n)$, and relative rank 1.[9] One can see that the underlying multiplicative structure of the formula thus constructed naturally suggests a W-tree T of the form shown in Figure 1. This is a W-tree of depth-2 and bias 2 (which is the best possible for this W).

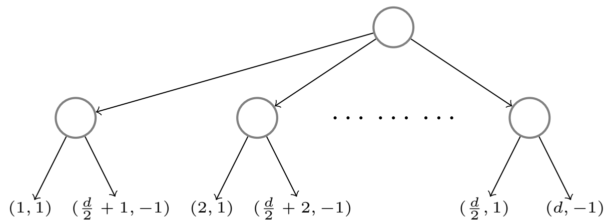

Figure 1: The W-tree of depth 2 and bias 2 arising from the formula construction above.

The above indicates a general technique for constructing formulas of large relative rank. Start by finding a $W' \subseteq W$ such that $|\mathrm{Sum}(W')|$ is small. Construct a formula of plausibly optimal relative rank (i.e. $n^{-|\mathrm{Sum}(W')|/2}$) over the variable sets corresponding to W' by adding enough set-multilinear monomials so that sub-additivity no longer indicates that the rank of the formula is small. In doing this, we end up taking a sum of size n^b where

$$b := \frac{1}{2} \sum_{i \in W'} |\alpha_i| - \frac{|\mathrm{Sum}(W')|}{2}. \tag{2}$$

This indicates that it helps to take W' to be a small set, since otherwise this formula would be too large (if there were no such constraint, we could simply have taken $W' = W$). We partition W into small sets $W'_1, \ldots, W'_r$ this way, and construct formulas for each. Then, applying again the same principle to the multiset $\{\mathrm{Sum}(W'_1), \ldots, \mathrm{Sum}(W'_r)\}$, we get a high-rank set-multilinear

[9] This is an example of Nisan and Wigderson [NW97], aptly called the *Product of Inner Products* polynomial.

formula over all of $X_1, \ldots, X_d$. As in the simple example above, this gives rise to a multiplicative structure that can be described by means of a W-tree T. The set W' constructed above corresponds to one of the nodes at height 1 in T and the quantity b in (2) is (almost) something we will define later to be the bias of the corresponding node, and $n^{b/2}$ lower bounds the size of the constructed formula. However, a careful analysis of the construction shows that the size of the formula is actually larger: at each node of the tree T, the formula uses a sum governed by the bias of the corresponding node. This naturally ends up yielding a formula whose size is governed by the path bias of T. Minimizing this over the choice of all trees yields the tree bias of W, as defined above.

Proof of Theorem 3. The above outline already indicates how to construct a set-multilinear formula of product-depth Δ and size $n^{O(\mathrm{Treebias}_\Delta(W))}$ that computes a polynomial of optimal relative rank. The only part that is unclear is how to ensure that the bounds on relative rank imposed by sub-additivity are actually tight. We do this by a careful inductive definition of the formulas. In a revision of our earlier paper [LST22], we showed how to do this for a specific W which contains only the two distinct elements -1 and $1/\sqrt{2}$. In this paper, we extend this construction to all W. This gives the second part of Theorem 3.

In the process, we note that the formulas we construct all have a special property: they have a *unique* multiplicative structure, i.e. they build up all their set-multilinear monomials in the same way, given by a single W-tree T. In principle, a general set-multilinear formula could contain many different kinds of trees (e.g. by summing formulas corresponding to different trees). These special formulas that we construct have been studied before: they are called *Pure* formulas [NW97] or *Unique Parse Tree* (UPT) formulas [LMP19, LLS19]. We use the latter terminology.

For the first part of Theorem 3, we proceed as follows. We first show that UPT formulas of product-depth Δ have indeed the claimed upper bound on the relative rank, by using the basic properties of relative rank from Lemma 6 and a simple inductive argument. To argue about a general set-multilinear formula F, we show that any set-multilinear formula can be written as a sum of $O_d(1)$ many UPT formulas of the same size and product-depth. Using the sub-additivity of relative rank and the bound for UPT formulas, we see that F also has small relative rank.

We illustrate the power of the latter theorem with a short proof of one of the main results of [LST22]: an $n^{\Omega(\sqrt{d})}$ lower bound for set-multilinear formulas of product-depth 2.[10] By Theorem 3, it suffices to construct a multiset $W \subseteq [-1, 1]$ with $|\mathrm{Sum}(W)| = 0$ and tree bias $\Omega(\sqrt{d})$. Consider a W with $\Theta(d)$ copies each of (-1) and $\alpha := (1 - 1/\sqrt{d})$ so that $\mathrm{Sum}(W) = 0$. Given any depth-2 W-tree T, it can be checked that one of the following hold.

- There is a depth-1 vertex u with $t_u \geqslant \sqrt{d}/2$ children. In this case, any path through u has bias $\Omega(\sqrt{d})$.
- Every u at depth-1 has $t_u < \sqrt{d}/2$ children, in which case $|\mathrm{Sum}(u)| \geqslant t_u/(2\sqrt{d})$. This implies that any path in T has bias $\sum_u t_u/(2\sqrt{d}) = \sqrt{d}/2$.

Proof of Theorem 4 In a similar way, we can also extend the results of [LST22] to show improved lower bounds for product-depth 3 (i.e. $\Sigma\Pi\Sigma\Pi\Sigma\Pi\Sigma$ formulas). More precisely, taking W as above but redefining $\alpha = 1 - (1/d^{1/4}) - (1/d^{3/4})$, we are able to prove a tree-bias lower bound of $\Omega(d^{1/4})$. This implies a formula lower bound of $n^{\Omega(d^{1/4})}$, which improves upon a lower bound of $n^{\Omega(d^{1/7})}$ from our previous work.

[10] This is essentially the heart of the argument of [LST22], abstracting away the details about algebraic formulas, and keeping only the combinatorial core.

In the second part of the proof, we show that this is the best bound that this technique can prove, for *any* choice of W. Equivalently, we can show that every W has depth-3 tree bias $O(d^{1/4})$. We illustrate the idea with a sketch of the special case when W has two distinct elements (as in the two lower bounds above). In this case, it is not hard to argue that without loss of generality, the two distinct elements of W are (-1) and $\alpha \in (0,1]$.

First of all, we observe that any W has a tree of depth Δ and path bias $O(\Delta\|W\|_1^{1/\Delta})$, where $\|W\|_1$ denotes the sum of the absolute values of the elements of W. This is analogous to the fact that $\mathrm{IMM}_{n,d}$ has set-multilinear formulas of depth-Δ and size $n^{O(\Delta d^{1/\Delta})}$. Call this the "basic construction".

Now, given W as above, we proceed as follows. By a classical result of Dirichlet (see, e.g. [Juk11, Theorem 4.9]), for any t, there exist integers $q \in [t]$ and $p \in \{0,\ldots,t\}$ such that $|q\alpha - p| \leqslant 1/t$. Note that this gives a multiset $W' \subseteq W$ of size $p+q$ such that $|\mathrm{Sum}(W')| \leqslant 1/t$. We apply this result with $t = \sqrt{d}$ and proceed in one of two ways depending on the value of $p+q$.

- If $p+q \geqslant d^{1/4}$, then we can partition W into at most $r \leqslant d^{3/4}$ sets $W_1,\ldots,W_r$ of size $p+q$, each of which has sum at most $1/\sqrt{d}$. As $p+q \leqslant 2\sqrt{d}$, using the basic construction of depth 2, we get a tree T_i of bias $O(d^{1/4})$ for each W_i. Attaching all these to a common root gives a tree of path-bias $O(d^{1/4})$ (the root adds at most $d^{3/4}\cdot(1/\sqrt{d}) = d^{1/4}$ to the bias of any path).

- If $p+q \leqslant d^{1/4}$, then by using $d^{1/4}/(p+q)$ many disjoint sets of sum $1/\sqrt{d}$ each, we get a set W' of size $d^{1/4}$ and sum at most $d^{1/4}/((p+q)\cdot\sqrt{d}) \leqslant 1/d^{1/4}$. We partition W into $r \leqslant d^{3/4}$ sets $W'_1,\ldots,W'_r$ of this form. We use a tree T_i of depth-1 for each W'_i (which has path bias at most $d^{1/4}$ trivially) and attach these to the leaves of a depth-2 tree for the set $\tilde{W} = \{\mathrm{Sum}(W_1),\ldots,\mathrm{Sum}(W_r)\}$. The latter tree is constructed using the basic construction of depth 2, and has bias $O(d^{1/4})$ as $\|\tilde{W}\|_1 \leqslant r/d^{1/4} \leqslant \sqrt{d}$.

This gives the argument in the case of W with only two distinct elements. For general W, we use a similar high-level argument. However, we need a suitable replacement for Dirichlet's theorem, which only works for the special W dealt with above. We prove a generalization of this theorem (see Lemma 9 below) to the setting of arbitrary multisets W. We think the statement is natural and interesting in its own right, but could not find mention of it in the literature.

In the special case that W contains d copies of $\alpha \in (0,1)$ and d copies of -1, the above implies the standard Dirichlet theorem used above. With the above generalized theorem in place, we can follow the structure of the argument for the special case, with technical modifications. This yields the depth-3 relative rank upper bound for any W.

Proof of Theorem 5 for depth Δ. While the proof of this theorem employs the same high-level argument as Theorem 23 described above, it is considerably more technical. We illustrate the idea again with the case when W contains only two distinct elements, which we can assume to be -1 and some $\alpha \in [0,1]$. Let $\mathrm{Bias}(\Delta,d)$ denote the largest possible bias of a depth-Δ W-tree. We give a constructive bound on this quantity by an inductive construction (based on Δ).

For $\Delta = 1$, we have the trivial bound $\mathrm{Bias}(1,d) \leqslant d$. For $\Delta > 1$, we use Dirichlet's theorem to find integers $p,q \leqslant d^{1-\varepsilon}$ such that $|q - p\alpha| \leqslant d^{-(1-\varepsilon)}$. This gives us a set $W' \subseteq W$ of size $p+q$ such that $|\mathrm{Sum}W'| \leqslant d^{-(1-\varepsilon)}$. There are again two cases to consider based on the magnitude of q.

- If $q \geqslant d^{\varepsilon}$, then this yields that $|W'| \geqslant d^{\varepsilon}$. Partitioning W into $t = d^{1-\varepsilon}$ subsets $W'_1,\ldots,W'_t$ of this form and using a recursive construction for each of $W'_1,\ldots,W'_t$, we get a W-tree of bias $\mathrm{Bias}(\Delta - 1, d^{\varepsilon}) + O(1)$. (Here, the last $O(1)$ term accounts for the bias accrued at the root, which is only a constant.)

- Conversely, if $q \leqslant d^{\varepsilon}$, we pick as many sets $W'_1, \ldots, W'_r$ as we can to form a set W'' of size (roughly) $d^{1-\varepsilon}$. Note that $|\mathrm{Sum}(W'')| \leqslant d^{1-\varepsilon}/d^{1-\varepsilon} \leqslant 1$. We partition W into $s \leqslant d^{\varepsilon}$ sets $W''_1, \ldots, W''_s$ of this form. We can construct a W-tree T of depth $\Delta = \Delta_1 + \Delta_2$ by
  - Constructing a W''_i-tree T_i of depth Δ_1 by constructing W'_j-tree $T_{i,j}$ of depth $\Delta_1 - 1$ for each $W'_j \subseteq W''_i$ and connecting these trees to a common root.
  - Constructing a depth-Δ_2 $\tilde{W}$-tree $\tilde{T}$, where $\tilde{W} = \{\mathrm{Sum}(W''_1), \ldots, \mathrm{Sum}(W''_s)\}$[11] and replacing the leaf labelled i with the tree T_i.

  As the sets $\tilde{W}$ and W'_i have size d^{ε} each, it makes sense to take $\Delta_1 = \Delta_2 = \Delta/2$. This leads to a bound on the bias of the tree T of $2 \cdot \mathrm{Bias}(\Delta/2, d^{\varepsilon}) + O(1)$.

We choose ε to balance the bias obtained from each of the above two strategies. It is clear that if $\varepsilon < 1/(2\Delta)$ (say), then the first strategy yields a bad bound of $d^{1/2}$ (or worse). This implies that we must take $\varepsilon \geqslant 1/2\Delta$, which can yield a best possible upper bound of $d^{1/\Delta^{O(\log \Delta)}}$ from the second strategy. We show that this upper bound is indeed achievable, by taking $\varepsilon = \Theta(\log^2 \Delta/\Delta)$.

1.3 Related Work

Barriers for lower bound techniques. The partial derivative method and its variants have been used to prove several lower bounds in algebraic complexity theory including the recent work of the authors. While these techniques have been quite useful, it is unclear whether they can be used to separate VP from VNP. In the last decade, there were many attempts at understanding the limitations of these lower bound techniques. This has led to a body of work about *barrier* results [SA08, Gro15, FSV18, GKSS17, ELSW18, EGdOW18] in algebraic complexity theory. These results typically consider a large family of lower bound techniques and argue that such techniques cannot be used to prove strong lower bounds. However, all such results are either conditional, or hold for relatively weak models of computation (such as set-multilinear formulas of product-depth 1). In contrast to these results, here we focus on a specific technique, namely the technique that gave the first super-polynomial lower bound for low-depth circuits. We show an unconditional limitation on this technique with respect to a reasonably strong model of computation. Hence, our work is incomparable to this literature.

Our other recent work [LST21]. In a different recent paper, we prove algebraic formula lower bounds for formulas of larger depths. Specifically, we are able to prove superpolynomial set-multilinear formula lower bounds for $\mathrm{IMM}_{n,n}$ and *non-commutative* formula[12] lower bounds for formulas of depths up to $o(\sqrt{\log d})$. Note that the first of these results is a lower bound in the *high-degree* setting. This does not immediately imply a lower bound for general formulas, as we do not know of an efficient transformation to set-multilinear formulas when the degree is large. The second result does not imply any lower bounds in the commutative setting, as far as we know. The results of this paper are thus somewhat orthogonal, as they apply to set-multilinear (commutative) formulas in the low-degree setting.

[11] There is a small technical point here, which is that we will be left with a few more elements not covered by any of the W''_is. We ignore this here.

[12] This means that the operations of the formula are those of the non-commutative polynomial ring $\mathbb{F}\langle x_1, \ldots, x_N \rangle$ where variables do not commute.

Organization. We start with some preliminaries in Section 2. We then prove Thorems 3, 4 and 5 in Sections 3, 4 and 5 respectively.

2 Basic Preliminaries and Results from Previous Work

Fix any multiset $W = \{\alpha_1, \ldots, \alpha_d\} \subseteq [-1, 1]$ and let $\mathbb{F}_{\mathrm{sm}}[X_1, \ldots, X_d]$ be a lopsided set-multilinear space of polynomials with $|X_i| = n^{\alpha_i}$.

The following is a consequence of earlier work of the authors.

Lemma 7 (Lower bounds from relative rank, Implicit in [LST22])**.** *Let d and n be integer parameters. Assume that $W = \{\alpha_1, \ldots, \alpha_d\} \subseteq [-1, 1]$ is an arbitrary multiset and consider the space $\mathbb{F}_{sm}[X_1, \ldots, X_d]$ where $|X_i| = n^{|\alpha_i|}$. Assume that we have shown the following: for any set-multilinear formula F (over variable sets $X_1, \ldots, X_d$) of size at most $s(n, d)$ and product-depth at most Δ, we have*

$$\mathrm{relrk}_W(F) \leqslant C_d \cdot \varepsilon_n \cdot n^{-|\mathrm{Sum}(W)|/2},$$

where C_d depends only on d and $\varepsilon_n \to 0$ as $n \to \infty$.

Then, for n large enough in comparison to d, any set-multilinear formula F of product-depth Δ computing $\mathrm{IMM}_{\mathrm{poly}(n),d}$ must have size at least $s(n, d)$. Further, any (possibly non-set-multilinear) formula of depth at most $\Delta + 1$ computing $\mathrm{IMM}_{\mathrm{poly}(n),d}$ must have size at least $s(n, d)/d^{O(\Delta d)}$.

The following simple proposition regarding path bias will be useful.

Proposition 8. *Let $W \subseteq [-1, 1]$ be any finite multiset and let T be a W-tree with internal vertex u. If u has children $u_1, \ldots, u_r$, then*

$$\mathrm{Pathbias}_{W_u}(T_u) = \left(\max_{i\in[r]} \mathrm{Pathbias}_{W_{u_i}}(T_{u_i})\right) + \left(\sum_{j=1}^{r} |\mathrm{Sum}(u_j)|\right) - |\mathrm{Sum}(u)|$$

where T_v denotes the subtree rooted at vertex v (which is, by definition, a W_v-tree in the natural way).

Proof. Let p_v denote $\mathrm{Pathbias}_{W_v}(T_v)$ for any vertex v of T.

For any $i \in [r]$, let π_{u_i} denote the path of bias p_{u_i} in T_{u_i}. Let π_u denote the path in T_u obtained by adding the vertex u to π_{u_i}. Note that the off-path nodes of π_u are precisely the off-path nodes of π_{u_i} along with u_j ($j \neq i$). Thus, the bias of π_u can be written as

$$\begin{aligned}
\mathrm{bias}(\pi_u) &= \left(\sum_{v\in\mathrm{Offpath}(\pi_u)} |\mathrm{Sum}(v)|\right) - |\mathrm{Sum}(u)| \\
&= \mathrm{bias}(\pi_{u_i}) + |\mathrm{Sum}(u_i)| + \sum_{j\in[r]\setminus\{i\}} |\mathrm{Sum}(u_j)| - |\mathrm{Sum}(u)| \\
&= p_{u_i} + \left(\sum_{j=1}^{r} |\mathrm{Sum}(u_j)|\right) - |\mathrm{Sum}(u)|.
\end{aligned}$$

As this holds for each $i \in [r]$, we have shown that

$$p_u \geqslant \left(\max_{i\in[r]} p_{u_i}\right) + \left(\sum_{j=1}^{r} |\mathrm{Sum}(u_j)|\right) - |\mathrm{Sum}(u)|. \tag{3}$$

For the reverse inequality, we proceed in the same way. Let π_u be a path in T_u of bias p_u. If π_u has length 0, then we have

$$p_u = \mathrm{bias}(\pi_u) = \left(\sum_{j=1}^{r} |\mathrm{Sum}(u_j)|\right) - |\mathrm{Sum}(u)| \leqslant \left(\max_{i\in[r]} p_{u_i}\right) + \left(\sum_{j=1}^{r} |\mathrm{Sum}(u_j)|\right) - |\mathrm{Sum}(u)|$$

and hence we are trivially done. Otherwise, the path π_u passes through some child u_i of u. Let π_{u_i} be the path in T_{u_i} obtained by removing u from π_u. Then, through the same sequence of equalities proved above, we get

$$p_u = p_{u_i} + \left(\sum_{j=1}^{r} |\mathrm{Sum}(u_j)|\right) - |\mathrm{Sum}(u)| \leqslant \left(\max_{i\in[r]} p_{u_i}\right) + \left(\sum_{j=1}^{r} |\mathrm{Sum}(u_j)|\right) - |\mathrm{Sum}(u)|.$$

Hence, we have proved the reverse inequality to (3) and we are done. $\square$

2.1 A Generalized form of Dirichlet's theorem

Here we prove a generalized form of the standard Dirichlet Principle (see, e.g. [Juk11, Theorem 4.9]), which we will use in Sections 4 and 5.

Lemma 9 (A Generalized Form of the Dirichlet Principle)**.** *Assume $d \geqslant 2$. Let $W \subseteq [-1,1]$ be any multiset with at least d non-negative and d non-positive elements. Then, for each positive integer $t \leqslant 2d$, there is a multiset $T \subseteq W$ of size at most t such that $|\mathrm{Sum}(T)| \leqslant 4/(t-1)$.*

Proof. The proof is via the Pigeonhole principle. Fix a t as above and let $\ell = \lfloor t/2 \rfloor$. If W contains an element x such that $|x| \leqslant 2/\ell$, then we are done trivially, so we assume that this is not the case.

Let $\{x_1, \ldots, x_\ell\}$ and $\{-y_1, \ldots, -y_\ell\}$ be any ℓ positive and negative elements of W respectively (here, $x_i, y_i \in (2/\ell, 1]$ for each i).

For $i \in \{0, \ldots, \ell\}$, define $u_i = \sum_{j=1}^{i} x_j$ and $v_i = \sum_{j=1}^{i} y_j$. For $i, j \in \{0, \ldots, \ell\}$, let $w_{i,j} = u_i + v_j$. Note that as $x_i, y_i \in [0,1]$ for each $i \in [\ell]$, we have $u_i, v_j \in [0, \ell]$ and $w_{i,j} \in [0, 2\ell]$ for each $i, j \in \{0, \ldots, \ell\}$. Also note that $u_0, \ldots, u_\ell$ and $v_0, \ldots, v_\ell$ are increasing sequences in which the difference between any pair of elements is strictly more than $2/\ell$.

Divide the interval $[0, 2\ell]$ into ℓ^2 sub-intervals of length $2/\ell$ each. By the pigeonhole principle, there exist distinct (i,j) and (i',j') from $\{0,\ldots,\ell\} \times \{0,\ldots,\ell\}$ such that $w_{i,j}$ and $w_{i',j'}$ lie in the same interval. In particular, we have

$$|w_{i,j} - w_{i',j'}| = |(u_i - u_{i'}) - (v_{j'} - v_j)| \leqslant \frac{2}{\ell}. \tag{4}$$

Fix such (i,j) and (i',j'). Since these pairs are distinct, they must differ in some coordinate. We assume that they differ in the first coordinate (the other case is similar).

Without loss of generality, assume that $i > i'$. We note that it cannot be the case that $j \geqslant j'$. This is because we would then have

$$|w_{i,j} - w_{i',j'}| = (u_i + v_j) - (u_{i'} + v_{j'}) \geqslant u_i - u_{i'} > \frac{2}{\ell}$$

where for the inequalities we use the fact that $u_0, \ldots, u_\ell$ and $v_0, \ldots, v_\ell$ are increasing sequences in which the difference between any pair of elements is strictly more than $2/\ell$. This contradicts (4) above. In particular, this implies that $j < j'$. By (4), this yields

$$|(u_i - u_{i'}) - (v_{j'} - v_j)| = \left|\sum_{p=i'+1}^{i} x_k - \sum_{q=j+1}^{j'} y_j\right| \leqslant \frac{2}{\ell}.$$

This implies that to get a set T satisfying the requirements of the lemma, it is sufficient to take $T = \{x_{i'+1}, \ldots, x_i, -y_{j+1}, \ldots, -y_{j'}\}$. Note that $|T| \leqslant 2\ell \leqslant t$, and by the above computation $|\mathrm{Sum}(T)| \leqslant 2/\ell \leqslant 4/(t-1)$. $\square$

3 The Lower Bound technique and Tree bias

In this section we will show that tight bounds on the tree bias yield the best possible bound on the relative-rank of set-multilinear low-depth formulas. Specifically, we prove Theorem 3.

3.1 Set-multilinear formulas and Unique Parse Trees

First, it will be helpful to make some structural changes to the formula. We will write a set-multilinear formula as a *small* sum of set-multilinear formulas such that each formula has a unique *parse tree*. In order to describe this we introduce some definitions.

Definition 10 (Parse Formula)**.** *Let F be a set-multilinear formula. A parse formula F' is obtained from F as follows.*

- *The root + gate is added to F'.*
- *For every + gate added to F', one of its children is added to F'.*
- *For every $\times$ gate added to F', all its children are added to F'.*

Note that, such a parse formula computes a set-multilinear monomial. The polynomial computed by F is the sum of monomials computed by its parse formulas.

Parse trees and W-trees. Let F' be a parse formula from a set-multilinear formula F. We define the parse trees of F as follows. Let g be a + gate with the parent u and child v. We draw a direct edge between u and v and remove the + gate from F'. We do this *short-circuiting* step for each + gate of the parse formula. Similarly, we remove the + root of F'. Let $\mathfrak{T}$ be the tree thus obtained. We call this the *shape* of F'.

Let ℓ be a leaf of $\mathfrak{T}$. It corresponds to a gate g in F which is either a + gate in F' or a leaf in F'. The polynomial computed by g is a linear polynomial on variable set X_i for some $i \in [d]$. We label ℓ with (i, α_i). This way, we label each leaf of $\mathfrak{T}$ with elements of W. We call the W-tree T thus obtained a *parse tree* of F. Note that the depth of T is the same as the product-depth of F.

Definition 11 (UPT formula)**.** *We say that a set-multilinear formula F is a Unique Parse Tree formula (or UPT) if all the parse trees of F are identical.*

Lemma 12. *Let F be a set-multilinear formula of size s and depth Δ. Then F can be written as a sum of at most d^{3d} many set-multilinear UPT formulas such that each formula has size at most s and depth Δ.*

Proof. For the set-multilinear formula F as above, we will show that there are at most d^{3d} many different parse trees. This will prove the lemma.

As F is a set-multilinear formula computing a polynomial of degree d, any parse tree of F has d leaves. We can also assume without loss of generality that each internal node of the tree has at least two children. If this is not the case, then in the formula there is a $\times$ gate with only 1 child. Any such gate can be merged with its child before we create a parse tree.

Any tree with d leaves in which each internal node has at least two children has at most d internal nodes. For a tree T, let L_T denote a sequence $(\langle v, \text{parent}(v)\rangle : v \in T)$. Here, we assume that parent(root) is defined as $\perp$. This sequence L_T completely specifies the shape of the tree. There are at most d^{2d} distinct sequences of this form. Moreover to define a parse tree one has to label also the leaves. Each one of the d leaves is associated to a distinct element from $\{(i, \alpha_i) \mid i \in [1, d]\}$. So there are at most $d! \leqslant d^d$ such labelings. Thus, we conclude that there are at most d^{3d} many parse tree for any set-multilinear formula computing a polynomial of degree d. $\square$

3.2 Tree bias lower bounds imply formula lower bounds

In this section, we show how lower bounds on $\text{Treebias}_\Delta(W)$ imply set-multilinear formula lower bounds in the low-degree setting. By Lemma 7, this implies lower bounds for general formulas as well.

We first show this connection for a UPT formula and then use the lemma from the previous section to conclude the same for general set-multilinear formulas. Specifically, we prove the following statement.

Lemma 13. *Let n, d be positive integers. Let $\Delta \geqslant 1$. Let W be a multiset of $[-1, 1]$ of size d. Let F be a set-multilinear UPT formula of size s, product-depth Δ, and parse tree T. Assume, moreover, that* $\text{Pathbias}_W(T) = p$. *Then,*

$$\text{relrk}_W(F) \leqslant (s \cdot n^{-p/2}) \cdot n^{-|\text{Sum}(W)|/2}.$$

We first use this lemma to prove part (1) of Theorem 3.

Proof of Part (1) of Theorem 3. Let W and t be as in the statement of Theorem 3. Let F be a set-multilinear formula of product depth Δ and size at most s. From Lemma 12 we know that F can be written as a sum of UPT formulas, say $\Psi_1, \Psi_2, \ldots, \Psi_r$, where $r \leqslant d^{3d}$. We also know that the size of each Ψ_i is at most s and their depth is Δ. Let $\Gamma_1, \ldots, \Gamma_r$ be the parse trees of these formulas and let $p_i = \text{Pathbias}_W(\Gamma_i)$ for $i \in [r]$.

By Lemma 13, for each $i \in [r]$, $\text{relrk}_W(\Psi_i) \leqslant (s \cdot n^{-p_i/2}) \cdot n^{-|\text{Sum}(W)|/2}$. As $\text{Treebias}_\Delta(W) = t$, we have $p_i \geqslant t$ for each $i \in [r]$. Therefore, we get

$$i \in [r], \text{relrk}_W(\Psi_i) \leqslant (s \cdot n^{-t/2}) \cdot n^{-|\text{Sum}(W)|/2}.$$

As $F = \sum_{i=1}^{r} \Psi_i$, $r \leqslant d^{3d}$ and by sub-additivity of relrk, we get the claimed bound on the relrk of F, i.e.

$$\text{relrk}_W(F) \leqslant (d^{3d} \cdot s \cdot n^{-t/2}) \cdot n^{-|\text{Sum}(W)|/2}.$$

$\square$

We now prove Lemma 13.

Proof of Lemma 13. We prove the statement by induction on the depth of T (which is also the product depth of F).

Base case. Let $F = \sum_i \prod_j F_{i,j}$ be a set-multilinear UPT formula of product-depth $\Delta = 1$. Let T be the W-tree corresponding to F. Let u_0 be the root of T and let $u_1, \ldots, u_d$ be the children of u_0 with labels $(1, \alpha_1), \ldots, (d, \alpha_d)$, respectively.

By sub-additivity and sub-multiplicativity (Lemma 6, Items 2 and 3) of relrk, we can say that $\mathrm{relrk}_W(F) \leqslant \sum_i \prod_j \mathrm{relrk}_{\{\alpha_j\}}(F_{i,j})$. By using the Imbalance bound (Lemma 6 Item 1) on the relative rank of each $F_{i,j}$ we get that

$$\mathrm{relrk}_W(F) \leqslant \sum_i n^{-\sum_j |\alpha_j|/2} = s \cdot n^{-\sum_j |\alpha_j|/2} = sn^{-p/2} n^{-|\mathrm{Sum}(W)|/2}$$

where the last equality follows from Proposition 8. We get the desired bound.

Induction step. Let $F = \sum_i \prod_j F_{i,j}$ be a set-multilinear UPT formula of depth $\Delta > 1$. Let T be the W-tree corresponding to F. Let u_0 be the root of T and let $u_1, \ldots, u_k$ be the children of u_0. Let $T_1, \ldots, T_k$ be the trees rooted at $u_1, \ldots, u_k$ respectively.

As F is a UPT formula, we have that for each $i \neq i'$ and for any $j \in [k]$, the parse tree of $F_{i,j}$ is the same as the parse tree of $F_{i',j}$. Without loss of generality let us say the parse tree of $F_{i,j}$ is T_j for every i.

Also, for T, let us assume without loss of generality that the path bias of T is realised by a path π, where $\pi = u_0 \cdot u_1 \cdot \pi'$, i.e. specifically it passes through u_1. Let p_1 denote $\mathrm{Pathbias}_{W_{u_1}}(T_1)$.

Finally, let $s_{i,j}$ denote the size of the subformula $F_{i,j}$. Note that $\sum_{i,j} s_{i,j} \leqslant s$.

$$\begin{aligned}
\mathrm{relrk}_W(F) &\leqslant \sum_i \mathrm{relrk}_{W_{u_1}}(F_{i,1}) \cdot \prod_{j \geqslant 2} \mathrm{relrk}_{W_{u_j}}(F_{i,j}) && \text{Properties of relrk} \\
&\leqslant \sum_i \mathrm{relrk}_{W_{u_1}}(F_{i,1}) \cdot \prod_{j \geqslant 2} n^{-|\mathrm{Sum}(u_j)|/2} && \text{Trivial bound on relrk} \\
&\leqslant \sum_i \left((s_{i,1} \cdot n^{-p_1/2}) \cdot n^{-|\mathrm{Sum}(u_1)|/2} \right) \cdot \prod_{j \geqslant 2} n^{-|\mathrm{Sum}(u_j)|/2} && \text{Induction Hypothesis} \\
&\leqslant \sum_i s_{i,1} \cdot n^{-\left(p_1 + \sum_{j=1}^k \mathrm{Sum}(u_j)\right)/2} \\
&= \sum_i s_{i,1} \cdot n^{-(p + |\mathrm{Sum}(W)|)/2} && \text{Proposition 8} \\
&\leqslant s \cdot n^{-p/2} \cdot n^{-|\mathrm{Sum}(W)|/2}.
\end{aligned}$$

□

3.3 Tree bias upper bounds imply formula upper bounds

We now prove the second part of Theorem 3. The main idea is an abstraction of a proof from our earlier result [LST22][13] where we constructed polynomials to show that our lower bound technique was 'tight' for certain concrete spaces of lopsided set-multilinear polynomials. In this section, we essentially show that the lower bound proved via tree-bias is tight for *all* lopsided spaces.

The main technical result (which generalizes [LST22, Lemma 26]) is the following, which handles the case where each $|X_i|$ is a power of 2.

[13] More specifically, this result appeared in a later version of the paper that can be found on ECCC.

Lemma 14. *Let n, d be growing parameters and Δ any positive integer. Let $W = \{\alpha_1, \ldots, \alpha_d\} \subseteq [-1, 1]$ be a multiset. Assume that $\mathbb{F}_{\mathrm{sm}}[X_1, \ldots, X_d]$ be a lopsided space of set-multilinear polynomials with $|X_i| = n^{|\alpha_i|} = 2^{k_i}$ for non-negative integers $k_1, \ldots, k_d$.*

Let T be any W-tree of depth Δ with $\mathrm{Pathbias}_W(T) = p$. *Then, there is a UPT formula F of parse tree T (and hence product-depth Δ) with at most $d \cdot n^{p/2}$ leaves such that* $\mathrm{rank}(M_W(F))$ *is as large as possible (i.e. equal to either the number of its rows or columns).*

Proof. The high-level idea of the proof is quite simple. As each $|X_i| = 2^{k_i}$, we identify the variables of X_i with elements of the set $\{0,1\}^{k_i}$. Similarly, each monomial labeling a row or column of the matrix $M_W(P)$ (for some polynomial $P \in \mathbb{F}_{\mathrm{sm}}[X_1, \ldots, X_d]$) can be identified with a Boolean string of the appropriate length $k_+ = \sum_{i \in \mathcal{P}} k_i$ or $k_- = \sum_{i \in \mathcal{N}} k_i$ respectively. Any set-multilinear monomial $m \in \mathbb{F}_{\mathrm{sm}}[X_1, \ldots, X_d]$ can thus be identified with a pair of strings $\sigma_+ \in \{0,1\}^{k_+}$ and $\sigma_- \in \{0,1\}^{k_-}$. Assume for now that $k_+ \geqslant k_-$. We will construct a formula F such that the polynomial computed by F has 0-1 coefficients, and only contains monomials m such that the corresponding string σ_- is an initial segment of σ_+, after some (known) permutation of the coordinates. It is easy to see that for any such formula F, the underlying matrix $M_W(F)$ has the maximum possible rank, and this will complete the proof.

To make this idea precise, we will need to be able to make precise the inductive structure of the polynomials computed by the various sub-formulas of F. This requires quite a bit of notation, which makes the proof cumbersome. The notation and proof below follow the proof of [LST22, Lemma 26] closely.

Notation.

- We identify each X_i with elements of $\{0,1\}^{k_i}$. Given a variable $x \in X_i$, we use $\sigma(x)$ to denote the corresponding string in $\{0,1\}^{k_i}$.

- For a set $S \subseteq [d]$, we use $k(S)$ to denote $\sum_{i \in S} k_i$. We also use $W(S)$ to denote the multiset $\{\alpha_i \mid i \in S\}$ and $\mathcal{M}_S$ to denote the set of all set-multilinear monomials w.r.t. the variable sets $\{X_i \mid i \in S\}$.

- Given $S \subseteq [d]$, we define $S_+ = \{i \in S \mid \alpha_i \geqslant 0\}$ and $S_- = \{i \in S \mid \alpha_i < 0\}$. If S is clear from context, we use k_+ and k_- instead of $k(S_+)$ and $k(S_-)$. We say S is $\mathcal{P}$-heavy if $k_+ \geqslant k_-$ and $\mathcal{N}$-heavy otherwise.

  Fix an $S \subseteq [d]$. Given a polynomial $P \in \mathbb{F}_{\mathrm{sm}}[X_i : i \in S]$, the corresponding partial derivative matrix is $M_{W(S)}(P)$. The number of rows in this matrix is $\prod_{i \in S_+} n^{\alpha_i}$, which we denote by $R(S)$. Similarly, we denote the number of columns, which is $\prod_{i \in S_-} n^{-\alpha_i}$, by $C(S)$. Note that the maximum possible rank of $M_{W(S)}(P)$ is $\min\{R(S), C(S)\}$. This quantity is $R(S)$ if S is $\mathcal{N}$-heavy and $C(S)$ if S is $\mathcal{P}$-heavy.

- Let $I = [K]$ where $K = \sum_{i=1}^{d} k_i$. We partition $I = I_1 \cup \cdots \cup I_d$ where each I_j is the interval of length k_j starting at $\sum_{i<j} k_i + 1$. Given any $S \subseteq [d]$, we let $I(S) = \bigcup_{j \in S} I_j$.

- Say $S = \{i_1 < i_2 < \cdots < i_r\} \subseteq [d]$. Consider a monomial $m \in \mathcal{M}_S$, say $m = x_1 \cdots x_r$ where $x_j \in X_{i_j}$. We define the string $\sigma(m)$ to be the string obtained by concatenating $\sigma(x_1) \cdots \sigma(x_r)$. We will think of $\sigma(m)$ as a function mapping $I(S)$ to $\{0,1\}$, such that its restriction to any I_{i_j} (for $j \in [r]$) is exactly $\sigma(x_j)$.

- Any monomial $m \in \mathcal{M}_S$ can be written uniquely as a product of a 'positive monomial' $m_+ \in \mathcal{M}_{S_+}$ and a 'negative monomial' $m_- \in \mathcal{M}_{S_-}$.

We now define the kinds of polynomials that will be computed by sub-formulas of the UPT formula F that we construct.

Fix any $S \subseteq [d]$.

- Let $J_+ \subseteq I(S_+)$ and $J_- \subseteq I(S_-)$ be such that $|J_+| = |J_-| = \min\{k(S_+), k(S_-)\}$. Equivalently, $J_+ = I(S_+)$ if S is $\mathcal{N}$-heavy, and $J_- = I(S_-)$ if S is $\mathcal{P}$-heavy, and both J_+ and J_- have the same size.
- Let π denote a bijection from J_+ to J_-.

We call such a tuple (S, J_+, J_-, π) *valid.*

Fix a valid (S, J_+, J_-, π). Now, given a $\tau \in \{0,1\}^{|k(S_+)-k(S_-)|}$, we interpret τ as a function mapping $I(S_+)\backslash J_+$ to $\{0,1\}$ if S is $\mathcal{P}$-heavy and as a function mapping $I(S_-)\backslash J_-$ to $\{0,1\}$ if S is $\mathcal{N}$-heavy. We define the polynomial $P_{(S,J_+,J_-,\pi,\tau)}$ to be the sum of all monomials m that have the following two properties.

1. π-*Consistency*: $\sigma(m_+)(j) = \sigma(m_-)(\pi(j))$ for each $j \in J_+$, and
2. τ-*Agreement*: $\sigma(m_+)(j) = \tau(j)$ for all $j \in I(S_+)\backslash J_+$ if S is $\mathcal{P}$-heavy or $\sigma(m_-)(j) = \tau(j)$ for all $j \in I(S_-)\backslash J_-$ if S is $\mathcal{N}$-heavy.

It is an easy observation that for any valid (S, J_+, J_-, π) and any $\tau \in \{0,1\}^{|k(S_+)-k(S_-)|}$, the matrix $M_{W(S)}(P_{(S,J_+,J_-,\pi,\tau)})$ has a square sub-matrix of order $\min\{R(S), C(S)\}$ that is a permutation matrix. Hence, this matrix has rank $\min\{R(S), C(S)\}$ for any (S, J_+, J_-, π, τ) as above.

The main technical claim is the following.

Claim 15. *For any vertex u of T, let S_u denote the set of $i \in [d]$ that appear in the labels of leaves of the sub-tree T_u rooted at u. There exist some $J_{u,+}, J_{u,-} \subseteq [d]$ and π_u such that*

1. *$(S_u, J_{u,+}, J_{u,-}, \pi_u)$ is valid.*
2. *For every $\tau \in \{0,1\}^{|k(S_{u,+})-k(S_{u,-})|}$, the polynomial $P_{(S_u,J_{u,+},J_{u,-},\pi_u,\tau)}$ has a set-multilinear formula $F_{(S_u,J_{u,+},J_{u,-},\pi_u,\tau)}$ over the variable sets $\{X_i : i \in S_u\}$ that satisfies the following properties.*
    - *$F_{(S_u,J_{u,+},J_{u,-},\pi_u,\tau)}$ is UPT of parse tree T_u,*
    - *$F_{(S_u,J_{u,+},J_{u,-},\pi_u,\tau)}$ has at most $|S_u| \cdot n^{p_u/2}$ leaves, where $p_u = \mathrm{Pathbias}_{W_u}(T_u)$.*

The statement of Lemma 14 immediately follows from the above claim in the case that u is the root of T. It suffices therefore to prove the claim.

Proof of Claim 15. The proof is by induction on the height of u in T.

Base case: u has height 0. In this case, S_u is a singleton and hence, $S_u = \{i\}$ and one of $k(S_{u,+})$ or $k(S_{u,-})$ is 0. Hence, both $J_{u,+}$ and $J_{u,-}$ must be empty and π_u is a trivial (empty) bijection. Given any $\tau \in \{0,1\}^{|k(S_{u,+})-k(S_{u,-})|} = \{0,1\}^{k_i}$, it can be checked the polynomial $P_{(S_u,J_{u,+},J_{u,-},\pi_u,\tau)}$ is just a single variable x from X_i and hence has a trivial formula consisting of just a single leaf labelled by x. This formula $F_{(S_u,J_{u,+},J_{u,-},\pi_u,\tau)}$ has all the required properties.

Induction. Now, we consider a u of height $h > 1$. Say u has children $u_1, \ldots, u_r$. W.l.o.g., we assume that $u_1, \ldots, u_t$ are $\mathcal{P}$-heavy and $u_{t+1}, \ldots, u_r$ are $\mathcal{N}$-heavy for some $t \in \{0, \ldots, r\}$. By induction, we know that for each $i \in [r]$, we have a valid $(S_{u_i}, J_{u_i,+}, J_{u_i,-}, \pi_{u_i})$ so that the conclusion of the claim holds. We define $J_{u,+}, J_{u,-}$ and π_u as follows.

- W.l.o.g. assume that S_u is $\mathcal{P}$-heavy (the other case is similar). So we have $J_{u,-} = I(S_{u,-})$.
- Let $J'_{u,+} = \bigcup_{i\in[r]} J_{u_i,+}$ and similarly $J'_{u,-} = \bigcup_{i\in[r]} J_{u_i,-}$. Note that we have $|J'_{u,+}| = |J'_{u,-}|$. Let $\pi'_u : J'_{u,+} \to J'_{u,-}$ be the bijection obtained by taking the union of the bijections π_{u_i} ($i \in [r]$). That is, if $j \in J_{u_i,+}$, then $\pi'_u(j) = \pi_{u_i}(j)$.
- Let $J''_{u,+}$ be any subset of $I(S_{u,+})\backslash J'_{u,+}$ that has the same size as $I(S_{u,-})\backslash J'_{u,-}$. Note that there must exist such a $J''_{u,+}$ as we have assumed that S_u is $\mathcal{P}$-heavy. Fix an arbitrary bijection π''_u between $J''_{u,+}$ and $I(S_{u,-})\backslash J'_{u,-}$.
- Set $J_{u,+} = J'_{u,+} \cup J''_{u,+}$.
- Finally, set π_u be the union of the bijections π'_u and π''_u.

We now see how to construct the claimed formulas for the polynomials $P_{(S_u,J_{u,+},J_{u,-},\pi_u,\tau)}$ for any given $\tau : I(S_{u,+})\backslash J_{u,+} \to \{0,1\}$. Fix such a τ. The polynomial $P_{(S_u,J_{u,+},J_{u,-},\pi_u,\tau)}$ is the sum of all monomials that satisfy π_u-consistency and τ-agreement. We further partition this set of monomials based on their behaviour on $J''_{u,+}$.

More precisely, given any $\tau'' : J''_{u,+} \to \{0,1\}$, we say that a monomial m that has π_u-consistency and τ-agreement has *type* τ'' if $\sigma(m_+)(j) = \tau''(j)$ for each $j \in J''_{u,+} = J_{u,+}\backslash J'_{u,+}$. If we factor this monomial as $m = m_1 m_2 \cdots m_r$ where $m_i \in \mathcal{M}_{W_{u_i}}$, then we have the following properties.

- For each $i \in [r]$, m_i is π_{u_i}-consistent. This is because m is π_u-consistent and $\pi_{u_i} \subseteq \pi_u$ for each $i \in [r]$.
- For each $i \in [t]$, m_i is τ_i-consistent, where τ_i is the restriction of $\tau \cup \tau''$ to $I(S_{u_i,+})\backslash J_{u_i,+}$.
- For each $i \in \{t+1, \ldots, s\}$, m_i is τ_i-consistent, where τ_i is the restriction of $\tau'' \circ (\pi''_u)^{-1}$ to $I(S_{u_i,-})\backslash J_{u_i,-}$.

Conversely, if $m_1, \ldots, m_r$ have the above properties, then $m = m_1 \cdots m_r$ has π_u-consistency, τ-agreement and type τ''. Algebraically, this implies the sum of all such monomials m is precisely the product $\prod_{i=1}^r P_{(S_{u_i},J_{u_i,+},J_{u_i,-},\pi_{u_i},\tau_i)}$. Summing over all τ'' gives us the polynomial $P_{(S_u,J_{u,+},J_{u,-},\pi_u,\tau)}$.

Using the inductive construction of the formulas for $u_1, \ldots, u_r$, we can define

$$F_{(S_u,J_{u,+},J_{u,-},\pi_u,\tau)} = \sum_{\tau'':J''_{u,+}\to\{0,1\}} \prod_{i=1}^r F_{(S_{u_i},J_{u_i,+},J_{u_i,-},\pi_{u_i},\tau_i)}. \tag{5}$$

(Note that the dependence of each summand on τ'' is implicit: the string τ_i is defined using τ''.) By construction, the formula $F_{(S_u,J_{u,+},J_{u,-},\pi_u,\tau)}$ computes $P_{(S_u,J_{u,+},J_{u,-},\pi_u,\tau)}$.

Any parse tree of the constructed formula is a root attached to r subtrees, each of which is a parse-tree of $F_{(S_{u_i},J_{u_i,+},J_{u_i,-},\pi_{u_i},\tau_i)}$ ($i \in [r]$). Inductively, therefore, each parse tree of $F_{(S_u,J_{u,+},J_{u,-},\pi_u,\tau)}$ is equal to T_u as required.

Finally, we need to analyze the size of $F_{(S_u,J_{u,+},J_{u,-},\pi_u,\tau)}$. Using the induction hypothesis, each summand of (5) defines a formula with at most

$$\sum_{i=1}^r |S_{u_i}| \cdot n^{p_{u_i}/2} \leqslant n^{\frac{1}{2}\cdot\max_{i\in[r]} p_{u_i}} \cdot |S_u|$$

many leaves. Since the sum in (5) has size $2^{|J''_{u,+}|}$, the number of leaves in the formula constructed is at most

$$2^{|J''_{u,+}|} \cdot |S_u| n^{\frac{1}{2}\cdot \max_{i\in[r]} p_{u_i}}.$$

We claim that

$$2^{|J''_{u,+}|} = n^{\frac{1}{2}\sum_{i=1}^{r} |\mathrm{Sum}(u_i)| - \frac{1}{2}|\mathrm{Sum}(u)|}. \tag{6}$$

This implies that the number of leaves in the formula $F_{(S_u, J_{u,+}, J_{u,-}, \pi_u, \tau)}$ is at most

$$|S_u| \cdot n^{\frac{1}{2}\cdot \max_{i\in[r]} p_{u_i} + \frac{1}{2}\sum_{j=1}^{r} |\mathrm{Sum}(u_j)| - \frac{1}{2}|\mathrm{Sum}(u)|} = |S_u| \cdot n^{p_u/2}$$

where the latter inequality follows from Proposition 8. This finishes the induction assuming (6).

It remains to prove (6). To show this, define $A_i = S_{u_i} \backslash (J_{u_i,+} \cup J_{u_i,-})$ for $i \in [r]$, and let $A = \bigcup_{i\in[r]} A_i$. We count the size of A in two different ways. On the one hand, by definition, we have

$$\begin{aligned}
|A| = \sum_{i=1}^{r} |A_i| &= \sum_{i=1}^{t} (|I(S_{u_i,+})| - |J_{u_i,+}|) + \sum_{i=t+1}^{r} (|I(S_{u_i,-}| - |J_{u_i,-}|) \\
&= \sum_{i=1}^{t} (k(S_{u_i,+}) - k(S_{u_i,-})) + \sum_{i=t+1}^{r} (k(S_{u_i,-}) - k(S_{u_i,+})) \\
&= \sum_{i=1}^{r} |k(S_{u_i,+}) - k(S_{u_i,-})| \qquad (7)
\end{aligned}$$

where the second and last equalities use the fact that $S_{u_1}, \ldots, S_{u_t}$ are $\mathcal{P}$-heavy while $S_{u_{t+1}}, \ldots, S_{u_r}$ are $\mathcal{N}$-heavy. On the other hand, we also have $A = B_1 \cup B_2 \cup B_3$ where $B_1 = (S_u \backslash (J_{u,+} \cup J_{u,-}))$, $B_2 = (J_{u,+} \backslash J'_{u,+})$, and $B_3 = (J_{u,-} \backslash J'_{u,-})$. Note that B_2 is exactly $J''_{u,+}$ and B_3 is in bijective correspondence with B_2 via the bijection π''_u. Hence, $|B_2| = |B_3| = |J''_{u,+}|$. Secondly, we have

$$|B_1| = |I(S_{u,+})| - |J_{u,+}| = k(S_{u,+}) - k(S_{u,-}) = |k(S_{u,+}) - k(S_{u,-})|$$

where for the last equality we used our assumption that S_u is $\mathcal{P}$-heavy. We have thus shown that

$$|A| = 2|J''_{u,+}| + |k(S_{u,+}) - k(S_{u,-})|.$$

Putting this together with (7), we get

$$\begin{aligned}
|J''_{u,+}| &= \frac{1}{2}\sum_{i=1}^{r} |k(S_{u_i,+}) - k(S_{u_i,-})| - \frac{1}{2}|k(S_{u,+}) - k(S_{u,-})| \\
&= \frac{1}{2}\sum_{i=1}^{r} \left| \sum_{j\in S_{u_i,+}} k_j - \sum_{j\in S_{u_i,-}} k_j \right| - \frac{1}{2}\left| \sum_{j\in S_{u,+}} k_j - \sum_{j\in S_{u,-}} k_j \right| \\
&= \frac{\log n}{2}\sum_{i=1}^{r} \left| \sum_{j\in S_{u_i}} \alpha_j \right| - \frac{\log n}{2}\left| \sum_{j\in S_u} \alpha_j \right| \\
&= \frac{\log n}{2}\sum_{i=1}^{r} |\mathrm{Sum}(u_i)| - \frac{\log n}{2}|\mathrm{Sum}(u)|
\end{aligned}$$

where for the second-to-last equality, we used the fact that $k_j = \alpha_j \log n$ if $i \in S_{u_j,+}$ and $k_j = -\alpha_j \log n$ otherwise. Exponentiating both sides of the above equality yields (6). This completes the proof. □

□

Finishing the proof of Theorem 3. We now use Lemma 14 to prove the second part of Theorem 3.

Assume $W = \{\alpha_1, \ldots, \alpha_d\}$ and say $X_1, \ldots, X_d$ are such that $|X_i| = n^{|\alpha_i|}$ are as given in the statement of the theorem. For each $i \in [d]$, fix a set $X_i' \subseteq X_i$ of size 2^{k_i} where 2^{k_i} is the largest power of 2 upper bounded by $n^{|\alpha_i|}$. Fix α_i' of the same sign as α_i with $2^{k_i} = n^{|\alpha_i'|}$. Let $W' = \{\alpha_1', \ldots, \alpha_d'\}$.

Note that we have $|\alpha_i - \alpha_i'| \in [0, 1/\log n)$. In particular, if T is a depth-Δ W-tree such that $\mathrm{Pathbias}_W(T) = \mathrm{Treebias}_\Delta(W) = t$, then the corresponding W'-tree T' (just replace each leaf labelled α_i by one labelled α_i') has roughly the same path bias. More precisely, it follows easily from the definition of path bias that

$$p := \mathrm{Pathbias}_{W'}(T') \leqslant \mathrm{Pathbias}_W(T) + \sum_{i=1}^{d} |\alpha_i - \alpha_i'| \leqslant t + \frac{d}{\log n}.$$

Applying Lemma 14 to the lopsided space $\mathbb{F}_{\mathrm{sm}}[X_1', \ldots, X_d']$ and the W'-tree T', we see that there is a set-multilinear formula F' with at most $d \cdot n^{p/2}$ leaves and $\mathrm{relrk}_{W'}(F') = n^{-|\mathrm{Sum}(W')|/2}$. Plugging in the bound on p from the above display, we see that the number of leaves of F' is at most

$$d \cdot n^{p/2} \leqslant d \cdot n^{t/2 + \frac{d}{2\log n}} = d2^{d/2} \cdot n^{t/2} \leqslant 3^d \cdot n^{t/2}.$$

We claim that the set-multilinear formula F' satisfies the requirements of the theorem. Firstly, we note that as $X_i' \subseteq X_i$ for each i, the formula F' is also set-multilinear w.r.t. the partition $\{X_1, \ldots, X_d\}$. Secondly, it satisfies the required size and depth constraints. Finally, we claim that its relative rank is as large as claimed. To see this, note that the matrix $M_{W'}(F')$ is a submatrix of $M_W(F')$. Say that $M_{W'}(F')$ has R' rows and C' columns, and $M_W(F')$ has R rows and C columns. Then we have

$$\frac{R'}{R} = \prod_{i:\alpha_i \geqslant 0} \frac{n^{|\alpha_i'|}}{n^{|\alpha_i|}} \geqslant \frac{1}{2^d}.$$

A similar bound holds for $\frac{C'}{C}$ as well. By construction $M_{W'}(F')$ has the maximum possible rank, i.e. $\min\{R', C'\}$. In particular, this quantity is at least $\frac{1}{2^d} \cdot \min\{R, C\}$. As $M_{W'}(F')$ is a submatrix of $M_W(F')$, this also lower bounds the rank of the latter matrix. Thus, we have

$$\mathrm{relrk}_W(F') \geqslant \frac{1}{2^d} \cdot \frac{\min\{R, C\}}{\sqrt{RC}} = \frac{1}{2^d} \cdot n^{-|\mathrm{Sum}(W)|/2}.$$

This concludes the proof of the theorem.

4 Optimal bounds for depth 3 via our technique

This section is devoted to the proof of Theorem 4, which characterizes (up to constant factors) the maximum possible tree bias of a tree of depth 3.

In proving this theorem, it will be useful to consider a variant on the notion of tree bias defined above, that we will call *node bias*. The node bias of W (at any given depth Δ) is equal to the tree bias of W up to a factor of $O(\Delta)$. By Theorem 3, for constant-depths Δ, the node bias also captures the best lower bound that we can hope to prove via our technique.

Definition 16 (Node bias). *Fix a W-tree T. For an internal node v of T, we define the bias of v, denoted* $\mathrm{bias}(v)$*, to be $\sum_u |\mathrm{Sum}(u)|$ where the sum runs over the children u of v. The node bias of T, denoted* $\mathrm{Nodebias}_W(T)$*, is the largest bias of any internal node v of T. Further, the depth-Δ node bias of W, is the minimum node bias of any depth-Δ, W-tree T. This quantity is denoted* $\mathrm{Nodebias}_\Delta(W)$.

The following basic proposition relates the node bias of W and the tree bias of W.

Proposition 17. *For any depth-Δ W-tree T, we have*

$$\mathrm{Nodebias}_W(T) \leqslant \mathrm{Pathbias}_W(T) + |\mathrm{Sum}(W)| \leqslant \Delta \cdot \mathrm{Nodebias}_W(T).$$

In particular, for any multiset $W \subseteq [-1,1]$ and any depth Δ, we have $\mathrm{Nodebias}_\Delta(W) \leqslant \mathrm{Treebias}_\Delta(W) + |\mathrm{Sum}(W)| \leqslant \Delta \cdot \mathrm{Nodebias}_\Delta(W)$.

Proof. Let us start with the first inequality. Let u be an internal node of bias equal to $\mathrm{Nodebias}_W(T)$. Let us consider the internal path π from the root to u. Then, all the children of u belong to $\mathrm{Offpath}(\pi)$. In particular the bias of this path is at least $\mathrm{Nodebias}_W(T) - |\mathrm{Sum}(W)|$. For the second inequality, let us consider a path π of bias $\mathrm{Pathbias}_W(T)$. By definition, any node from $\mathrm{Offpath}(\pi)$ lies in $\bigcup_{u \text{ node of } \pi} \mathrm{children}(u)$. So $\mathrm{bias}(\pi) + |\mathrm{Sum}(W)| \leqslant \sum_{u \text{ node of } \pi} \mathrm{bias}(u) \leqslant \Delta \mathrm{Nodebias}_W(T)$ since any internal path has at most Δ nodes. $\square$

4.1 Some simple claims

Groupings of W. Given a partition[14] P of the elements of W, we define the *grouping* W' of W to be the multiset obtained by taking the sums of elements of P. Formally,

$$W' = \{\mathrm{Sum}(A) \mid A \in P\}.$$

The following basic lemma shows how to construct a W-tree from trees of its groupings and subsets.

Lemma 18. *Assume that $P = \{W_1, \ldots, W_t\}$ is a partition of W and let W' be the corresponding grouping. Say we have a W'-tree T' of node bias b' and depth Δ' and for each $i \in [t]$, a W_i-tree T_i of depth Δ_i and node bias at most b_i. Then, there is a W-tree T of node bias at most $\max\{b', b_i\}$ and depth at most $\Delta' + \max_{i\in[t]} \Delta_i$.*

Moreover, if each W_i is sign-monochromatic *(i.e., all elements of W_i have the same sign)*[15]*, then there is a W-tree T of depth Δ' and bias b'.*

Proof. In the first case, let us construct the W-tree in the most straightforward way. Start with T' and replace the leaf labelled $\gamma_i = \mathrm{Sum}(W_i) \in W'$ by the tree T_i. The depth is bounded by $\Delta' + \max_{i\in[t]} \Delta_i$. Let u be an internal node of T. If u comes from an internal node in T_i (with $1 \leqslant i \leqslant t$), then we still have that $\mathrm{bias}(u) \leqslant b_i$. So, let us consider a node u coming from an internal node of T'. Let v be a child of u, and $\gamma_{i_1}, \ldots, \gamma_{i_p}$ be the leaves of the subtree of T' rooted in v. So, in T, $\mathrm{Sum}(v) = \sum_{j=1}^{p} \mathrm{Sum}(W_{i_j})$ which equals $\mathrm{Sum}(v)$ in T'. In particular u has the same node bias in T as in T', and so it is bounded by b'.

In the second case, let us replace the leaf of T' labelled $\mathrm{Sum}(W_i)$ by $|W_i|$ many singleton leaves labelled by the distinct elements of W_i. Clearly the new tree has depth Δ'. Since, we only replace each leaf by some number of leaves of the same sum, the quantity $\mathrm{Sum}(v)$ has the same value in T and in T' for any internal node v of T; in particular, any node u that has only internal nodes as children has the same node bias in T and T'. Now consider a node u that has some leaf children. If v is a child of u in T which is also a leaf, then v is labelled by $\mathrm{Sum}(W_i)$ (for some $i \in [t]$); in T', v is replaced by nodes $v_1, \ldots, v_r$ where $\sum_{j=1}^{r} |\mathrm{Sum}(v_j)| = |\mathrm{Sum}(v)|$. Consequently, u has the same node bias in T as in T'. $\square$

[14] We follow the usual convention that $\varnothing \notin P$.

[15] Here, we think of 0 as having the same sign as any other number.

Lemma 19 (Preprocessing Lemma). *Let $W \subseteq [-1,1]$ be any multiset. Then, there is a partition $P = \{W_1, \ldots, W_t\}$ of W such that each W_i is sign-monochromatic (as in Lemma 18),* $\mathrm{Sum}(W_i) \in [-1,1]$ *for all $i \in [t]$ and* $\mathrm{Sum}(W_i) \in [-1,-1/2] \cup [1/2,1]$ *for each $i \in \{3,\ldots,t\}$.*

In particular, there is a grouping $W' \subseteq [-1,1]$ of W such that $|W' \setminus ([-1,-1/2] \cup [1/2,1])| \leqslant 2$ and for each W'-tree T', there is a W-tree T of same depth and node bias.

Proof. The 'in particular' part of the lemma follows directly from the first statement and Lemma 18. So it suffices to prove the first statement.

To see this, we construct the partition iteratively. We start with the trivial partition P where each element of W is a singleton. Now, as long as there are two elements A and B in P such that the elements of A and B have the same sign (recall that 0 has the same sign as any other number) and $|\mathrm{Sum}(A) + \mathrm{Sum}(B)| \leqslant 1$, we replace A and B by the set $A \cup B$ in P and continue.

When the above process terminates, we are left with at most one $A \in P$ with $\mathrm{Sum}(A) \in [-1/2, 0]$ and at most one $B \in P$ with $\mathrm{Sum}(B) \in [0, 1/2]$. Note also that at each step, we preserve the fact that each $A \in P$ is sign-monochromatic and satisfies $\mathrm{Sum}(A) \in [-1,1]$.

This partition thus has the required properties. □

The next lemma shows, in particular, how to construct W-trees of depth Δ and node bias $O(d^{1/\Delta})$ for any multiset $W \subseteq [-1,1]$ of size d and of sum at most 1.

Lemma 20. *Let $W \subseteq [-1,1]$ such that $\|W\|_1 \leqslant L$ and $|\mathrm{Sum}(W)| \leqslant 1$. Then, for any $\Delta \geqslant 1$, there is a W-tree of depth at most Δ and node bias at most $5L^{1/\Delta}$.*

Proof. We claim the result by induction on Δ.

The base case $\Delta = 1$ is trivial. Consider $\Delta \geqslant 2$. If $L \leqslant 5L^{1/\Delta}$, we can use a trivial tree of depth 1 and we are done. So we assume $L^{1-1/\Delta} \geqslant 5$.

Assume wlog that $\mathrm{Sum}(W) \geqslant 0$. Order the elements W as $(w_1, \ldots, w_d)$ (where $|W| = d$) in the following way. Having fixed $(w_1, \ldots, w_{i-1})$ for $i \in [d]$, we choose w_i by the following strategy.

- If $\sum_{j<i} w_j < 0$, then we set w_i to be some non-negative element of $W \setminus \{w_1, \ldots, w_{i-1}\}$ (such an element exists because $\mathrm{Sum} W \geqslant 0$).
- If $\sum_{j<i} w_j \geqslant 0$, then we set w_i to be some non-positive element of $W \setminus \{w_1, \ldots, w_{i-1}\}$, if such an element exists (otherwise, w_i is set arbitrarily).

It is easy to check that with this ordering of the elements of W, we have $|\sum_{j \leqslant i} w_i| \leqslant 1$ for each $i \leqslant d$.

Let $c = 2L^{1/\Delta}$.

Define a sequence of indices $i_0, \ldots, i_r$ as follows. Set $i_0 = 0$ and given $i_0, \ldots, i_t$, define i_{t+1} to be the least $i > i_t$ with $i < d$ such that

$$\sum_{j \leqslant i} w_j \geqslant 0 \text{ and } \sum_{i_t < j \leqslant i} |w_j| \geqslant c.$$

The sequence ends when we cannot continue the process any longer and we define $i_{r+1} = d$.

Now define a partition $P = \{W_1, \ldots, W_{r+1}\}$ of W by $W_t = \{w_i \mid i_{t-1} < i \leqslant i_t\}$. Note that for each $t \leqslant r+1$, $|\mathrm{Sum}(W_t)| \leqslant 1$ as we have

$$\mathrm{Sum}(W_t) = \sum_{i \leqslant i_{t+1}} w_i - \sum_{i \leqslant i_t} w_i$$

which is a difference between two non-negative numbers of absolute value at most 1 and hence at most 1 in absolute value. Also note that each $\|W_t\|_1 \geqslant c$ for $t \leqslant r$. So in particular, we have $r \leqslant L/c$. Finally, we also have $\|W_t\|_1 \leqslant c+3$ for each $t \in [r+1]$. Hence, for each $t \in [r+1]$, there is a W_t-tree T_t of depth 1 and node bias at most $\|W_t\|_1 \leqslant c+3 \leqslant 5L^{1/\Delta}$.

Let W' be the grouping of W given by P. Then, we have $|\mathrm{Sum}(W')| = |\mathrm{Sum}(W)| \leqslant 1$. As each $|\mathrm{Sum}(W_t)| \leqslant 1$, we have $W' \subseteq [-1,1]$ and also

$$\|W'\|_1 \leqslant r+1 \leqslant \frac{L}{2L^{1/\Delta}} + 1 \leqslant L^{1-1/\Delta} =: L_1$$

where the final inequality follows from $L^{1-1/\Delta} > 2$. Thus, by induction, there is a W'-tree T' of depth at most $\Delta - 1$ and node bias at most $5L_1^{1/(\Delta-1)} = 5L^{1/\Delta}$. Now, using Lemma 18, we are done. □

We also have the following simple 'pasting' lemma.

Lemma 21. *Let $P = \{W_1, \ldots, W_r\}$ be a partition of W and assume that for all i there is a W_i-tree T_i of depth at most Δ, node bias at most b_i and such that the root node of each T_i has bias at most b_i'. Then, there is a W-tree T of depth at most Δ and of node bias at most $\max\{b_1, \ldots, b_r, \sum_i b_i'\}$.*

Proof. We construct T by simply identifying the root nodes of all the T_i. □

Finally, the following claim will allow us to balance a given subset of W so that removing this subset results in two sets of absolute sum at most 1.

Lemma 22 (Balancing lemma)**.** *Say $W \subseteq [-1,1]$ is such that $|\mathrm{Sum}(W)| \leqslant 1$. Let $W' \subseteq W$ be arbitrary. Then there exists $W'' \subseteq W$ such that $W'' \supseteq W'$ and $\|W'' \backslash W'\|_1 \leqslant |\mathrm{Sum}(W')| + 1$ and $|\mathrm{Sum}(W'')|, |\mathrm{Sum}(W \backslash W'')| \leqslant 1$.*

Proof. Assume wlog that $\mathrm{Sum}(W) \geqslant 0$.

If $0 \leqslant \mathrm{Sum}(W') \leqslant 1$, then we can take $W'' = W'$.

If $\mathrm{Sum}(W') > 1$, we construct W'' by adding to W' the minimum number of negative elements of $W \backslash W'$ such that $\mathrm{Sum}(W'') \leqslant 1$ (this is possible as the overall sum is at most 1). Note that the total L_1-weight of all the elements added is at most $|\mathrm{Sum}(W')|$. We also have $\mathrm{Sum}(W'') \in [0,1]$. Hence, $\mathrm{Sum}(W \backslash W'') = \mathrm{Sum}(W) - \mathrm{Sum}(W'')$, which is at most 1 in absolute value.

If $\mathrm{Sum}(W') < 0$, then we construct W'' by adding to W' the minimum number of positive elements of $W \backslash W'$ such that $\mathrm{Sum}(W'') \geqslant 0$ (possible as the overall sum is positive). The total L_1-weight of all the elements added is at most $|\mathrm{Sum}(W')| + 1$. We also have $\mathrm{Sum}(W'') \in [0,1]$ as in the previous case. So we are done similarly. □

4.2 Depth-3 trees of small bias

The main theorem of this section is the following.

Theorem 23. *Let $W \subseteq [-1,1]$ be any multiset such that $|W| \leqslant d$ and $|\mathrm{Sum}(W)| \leqslant 1$. Then, there is a W-tree T of depth 3 and node bias $O(d^{1/4})$.*

The rest of the section is devoted to the proof of the above theorem. To construct the required W-tree T, we use the following procedure.

1. Preprocessing: By the Preprocessing lemma (Lemma 19), it suffices to consider multisets W such that $|W \backslash ([-1,-1/2] \cup [1/2,1])| \leqslant 2$.

2. We apply the following procedure to our multiset W.

   Algorithm $\mathcal{A}(W)$:

   Assignment $d := |W|$.

   Initialization If $d \leqslant 25$ then return the trivial depth-1 W-tree of node bias at most 25.

   Phase 1: As long as it is possible, pick pairwise disjoint sets A such that $|A| \leqslant d^{1/4}$ and
   $$\frac{|\mathrm{Sum}(A)|}{|A|} \leqslant \frac{12}{d^{1/2}}.$$
   When this is no longer possible, let $A_1, \ldots, A_{e_1}$ be the sequence of sets picked and let $W_1' = A_1 \cup A_2 \cdots \cup A_{e_1}$. Using the Balancing lemma, let $W_1 = W_1' \cup \{a_1, \ldots, a_{f_1}\} \subseteq W$ be such that $\sum_{i \leqslant f_1} |a_i| \leqslant |\mathrm{Sum}(W_1')| + 1$ and $|\mathrm{Sum}(W_1)|, |\mathrm{Sum}(W \backslash W_1)| \leqslant 1$.

   Construct a W_1-tree T_1 in the following way. Fix the grouping $\tilde{W}_1$ corresponding to the partition $P_1 = \{A_1, \ldots, A_{e_1}, \{a_1\}, \ldots, \{a_{f_1}\}\}$ of W_1. For each element of P_1, construct a trivial tree of depth-1 and for the grouping $\tilde{W}_1$, construct a depth-2 tree $\tilde{T}_1$ of node bias at most $5\sqrt{\|\tilde{W}_1\|_1}$ (using Lemma 20). Combine these using Lemma 18 to get a tree T_1 of depth 3 for W_1.

   Set $W' = W \backslash W_1$ and continue.

   Phase 2: As long as it is possible, pick pairwise disjoint sets $B \subseteq W'$ such that $|B| \leqslant d^{1/2}$ and
   $$\frac{|\mathrm{Sum}(B)|}{|B|} \leqslant \frac{12}{d^{3/4}}.$$
   When this is no longer possible, let $B_1, \ldots, B_{e_2}$ be the sequence of sets picked and let $W_2' = B_1 \cup B_2 \cdots \cup B_{e_2}$. Using the balancing lemma, let $W_2 = W_2' \cup \{b_1, \ldots, b_{f_2}\} \subseteq W'$ be such that $\sum_{i \leqslant f_1} |b_i| \leqslant |\mathrm{Sum}(W_2')| + 1$ and $|\mathrm{Sum}(W_2)|, |\mathrm{Sum}(W' \backslash W_2)| \leqslant 1$.

   Construct a W_2-tree T_2 in the following way. Fix the grouping $\tilde{W}_2$ corresponding to the partition $P_2 = \{B_1, \ldots, B_{e_2}, \{b_1\}, \ldots, \{b_{f_2}\}\}$ of W_2. Construct a trivial depth-1 $\tilde{W}_2$-tree $\tilde{T}_2$ of node bias $\|\tilde{W}_2\|_1$. For each element B of P_2, construct a depth-2 tree of node bias at most $5\sqrt{\|B\|_1}$ (using Lemma 20). Combine these using Lemma 18 to get a tree T_2 of depth 3 for W_2.

   Set $W'' = W' \backslash W_2$ and continue.

   Recursive call Compute $T_3 = \mathcal{A}(W'')$.

   Return The W-tree T of node bias at most $b_1 + b_2 + b_3$, where $b_i = \mathrm{Nodebias}(T_i)$ (for $i \in [3]$) given by T_1, T_2, T_3 and Lemma 21.

We now analyze the above construction. We first state a technical lemma.

Lemma 24. *If $d > 25$, then after Phases 1 and 2, we have $|W''| \leqslant d/2$.*

Let us assume the above lemma for now and prove the theorem.

Let $b_i(d)$ denote the node bias of the tree T_i ($i \in [3]$) assuming that the word W has size at most d. Then, the node bias of the tree is $b_1(d) + b_2(d) + b_3(d)$. By Lemma 24, we can bound $b_3(d)$ by $b_1(d/2) + b_2(d/2) + b_3(d/2)$. Continuing recursively in this way (until d becomes smaller than 25) we have
$$\mathrm{Nodebias}(T) \leqslant \left(\sum_{i \geqslant 0} b_1(d/2^i) + b_2(d/2^i) \right) + 25.$$

So to prove Theorem 23, it suffices to show that $b_1(d), b_2(d) \leqslant O(d^{1/4})$. From now on, we fix d and let $b_i = b_i(d)$ for $i \in [2]$.

We first bound b_1. By construction each element of the partition P_1 is a set A of size at most $d^{1/4}$ and hence has a depth-1 tree of node bias at most $d^{1/4}$. Moreover, we have

$$\begin{aligned}\|\tilde{W}_1\|_1 &= \sum_{i\leqslant e_1} |\mathrm{Sum}(A_i)| + \sum_{j\leqslant f_1} |a_j| \\ &\leqslant \sum_{i\leqslant e_1} |\mathrm{Sum}(A_i)| + |\mathrm{Sum}(W_1')| + 1 \leqslant 2\sum_{i\leqslant e_1} |\mathrm{Sum}(A_i)| + 1 \\ &\leqslant 2\sum_{i\leqslant e_1} \frac{12|A_i|}{d^{1/2}} + 1 \leqslant O(d^{1/2}).\end{aligned}$$

Hence, the tree $\tilde{T}_1$ has node bias $\tilde{b}_1 = O(d^{1/4})$. Hence, by Lemma 18, we see that $b_1 \leqslant O(d^{1/4})$.

We can bound b_2 similarly. By construction, each element of P_2 is a set B of size at most $d^{1/2}$ and hence by Lemma 20 has a depth-2 tree of node bias at most $5d^{1/4}$. Moreover, we have

$$\begin{aligned}\|\tilde{W}_2\|_1 &= \sum_{i\leqslant e_2} |\mathrm{Sum}(B_i)| + \sum_{j\leqslant f_1} |b_j| \\ &\leqslant \sum_{i\leqslant e_2} |\mathrm{Sum}(B_i)| + |\mathrm{Sum}(W_2')| + 1 \leqslant 2\sum_{i\leqslant e_2} |\mathrm{Sum}(B_i)| + 1 \\ &\leqslant 2\sum_{i\leqslant e_1} \frac{12|B_i|}{d^{3/4}} + 1 \leqslant O(d^{1/4}).\end{aligned}$$

In particular, this implies that the tree $\tilde{T}_2$ has node bias $\tilde{b}_2 = O(d^{1/4})$. In particular, by Lemma 18, we see that $b_2 \leqslant O(d^{1/4})$.

Thus, we have shown that $b_1, b_2 = O(d^{1/4})$ and we are done.

It remains only to prove Lemma 24, which we do now.

Proof of Lemma 24. Let $d'' = |W''|$. Assuming that $d > 25$ and $d'' > d/2$, we will show that Phases 1 and 2 of the algorithm could not have concluded, and hence derive a contradiction.

Let W''_+ and W''_- denote the positive and negative elements of W'' respectively. Recall that $W'' \subseteq W$ and the latter set contains at most two elements of absolute value less than 1/2 (by the preprocessing in Step 1). Further using the fact that $|\mathrm{Sum}(W'')| \leqslant 1$, it is easy to see that $|W''_+|, |W''_-| \geqslant d'''$ where $d''' = (d'' - 4)/3 > 2$.

By Lemma 9, it follows that there is a non-empty set $T \subseteq W''$ of size $t \leqslant \sqrt{d'''} + 1$ such that $|\mathrm{Sum}T| \leqslant 4/\sqrt{d'''}$. Since $d \geqslant 24$, this set T has size at most $\sqrt{d}$ and satisfies $|\mathrm{Sum}T| \leqslant 12/\sqrt{d}$.

Now, we do a short case analysis. Assume $|T| \leqslant d^{1/4}$. Then, T is the kind of set that the algorithm tries to find in Phase 1. Hence, the existence of such a T tells us that Phase 1 could not have concluded.

Otherwise, we have $|T| > d^{1/4}$. In this case, we have $|\mathrm{Sum}T|/|T| \leqslant 12d^{-3/4}$ and is hence the kind of set that the algorithm tries to find in Phase 2. Hence, the existence of such a T tells us that Phase 2 could not have concluded.

In either case, we are done. □

4.3 Optimality of the quartic bound

We will show here that the bound of Theorem 23 is optimal.

Proposition 25. *Let d be a growing integer parameter. There exists a multiset $W \subseteq [-1, 1]$ such that $|W| \leqslant d$, $|\mathrm{Sum}W| \leqslant 1$, and for all W-tree T of depth 3, T has node bias at least $\Omega(d^{1/4})$.*

Proof. If $d < 16$ the result follows immediately (just adapt the constant in the $\Omega()$ to deal with these cases). So let us assume that $d \geqslant 16$. Let d' be the largest integer such that $d' \leqslant d$ and d' is a fourth power of an integer. So $d'^{1/4} \geqslant 2$ and $d' \geqslant d/16$.

Let q be the closest integer to $\frac{d'}{2-1/d'^{1/4}+1/(2d'^{3/4})}$. So $\left|q - \frac{d'}{2-1/d'^{1/4}+1/(2d'^{3/4})}\right| \leqslant \frac{1}{2}$. Let us construct W with q copies of $1-1/d'^{1/4}+1/(2d'^{3/4})$ and $p = d'-q$ copies of -1. So $|W| \leqslant d' \leqslant d$ and

$$\begin{aligned}|\mathrm{Sum}(W)| &= \left|-p + q(1 - 1/d'^{1/4} + 1/(2d'^{3/4}))\right| \\ &= \left|-d' + q(2 - 1/d'^{1/4} + 1/(2d'^{3/4}))\right| \\ &\leqslant \left|(-d' + d')\right| + \frac{1}{2}\left|(2 - 1/d'^{1/4} + 1/(2d'^{3/4}))\right| \leqslant 1.\end{aligned}$$

It is sufficient to prove that any W-tree has large enough node bias.

Let T be any W-tree. Let us assume that $\mathrm{Nodebias}_W(T) < d'^{1/4}/4$.

Since every internal node α at distance two of the roots with k children (in particular the children of α are leaves of T) has bias at least $\mathrm{bias}(\alpha) \geqslant k \min_{v\in W} |v| \geqslant k/2$, it implies that $k < d'^{1/4}/2$.

Assume then that there is an internal node α at distance one of the root such that the subtree rooted in α has at least $d'^{1/2}$ leaves. Notice that for any children β of α with p_β negative children and q_β positive ones we have

$$\begin{aligned}|\mathrm{Sum}\beta| &= \left|-p_\beta + q_\beta(1 - 1/d'^{1/4} + 1/(2d'^{3/4}))\right| \\ &= \left|(-p_\beta + q_\beta) - q_\beta(1 - 1/2d'^{1/2})/d'^{1/4}\right|.\end{aligned}$$

Since $q_\beta \leqslant p_\beta + q_\beta < d'^{1/4}/2$ and p_β, q_β are integers, it implies that the fractional part of $|\mathrm{Sum}\beta|$ is at least $q_\beta(1 - 1/2d'^{1/2})/d'^{1/4}$. Moreover, if $|\mathrm{Sum}(\beta)| < 1$, it means that $q_\beta \geqslant p_\beta$, i.e., $q_\beta \geqslant (p_\beta + q_\beta)/2$. Hence in all cases,

$$|\mathrm{Sum}(\beta)| \geqslant \frac{q_\beta + p_\beta}{2} \cdot \frac{1}{2} \cdot \frac{1}{d'^{1/4}}.$$

Consequently,

$$\mathrm{bias}(\alpha) = \sum_{\beta \text{ child of } \alpha} |\mathrm{Sum}(\beta)| \geqslant \frac{d'^{1/2}}{4d'^{1/4}} = \frac{d'^{1/4}}{4}$$

which contradicts the hypothesis. So any node at depth 1 of the tree has less than $d'^{1/2}$ leaves in its subtree.

Let us show that finally the root ρ of T has large bias. Let β one of its children. Say that in the tree rooted in β, there are p_β negative leaves and q_β positive ones. So,

$$\begin{aligned}|\mathrm{Sum}\beta| &= \left|-p_\beta + q_\beta(1 - 1/d'^{1/4} + 1/(2d'^{3/4}))\right| \\ &= \left|(-p_\beta d'^{1/4} + q_\beta d'^{1/4} - q_\beta)\frac{1}{d'^{1/4}} + q_\beta \frac{1}{2d'^{3/4}}\right|.\end{aligned}$$

Since $q_\beta/(2d'^{3/4}) < 1/(2d'^{1/4})$, it implies that the distance of $|\mathrm{Sum}(\beta)|$ to the set $\mathbb{N}/d'^{1/4}$ is at lest $q_\beta/(2d'^{3/4})$. Again, if $|\mathrm{Sum}(\beta)| < 1$, it ensures that $q_\beta \geqslant (p_\beta + q_\beta)/2$. So in all cases,

$$|\mathrm{Sum}(\beta)| \geqslant \frac{p_\beta + q_\beta}{2} . \frac{1}{2d'^{3/4}}.$$

Consequently,

$$\text{bias}(\rho) = \sum_{\beta \text{ child of } \rho} |\text{Sum}(\beta)| \geqslant \frac{1}{4d'^{3/4}} \sum_{\beta \text{ child of } \rho} p_\beta + q_\beta = \frac{d'^{1/4}}{4}$$

which again contradicts the hypothesis.

In conclusion, we have that for any W-tree T

$$\text{Nodebias}_W(T) \geqslant \frac{d'^{1/4}}{4} \geqslant \frac{d^{1/4}}{8}.$$

□

Remark 26. *We can generalize the previous proof to larger depths by defining q to be the closest integer to $d'/(2+\sum_{i=1}^{\Delta-1}(-1)^i/d'^{(2^i-1)/2^{\Delta-1}})$. It implies that for all Δ, there exists a multiset W such that any W-tree of depth Δ has node bias at least $\Omega(d^{1/2^{\Delta-1}})$. It improves the constant in the exponent slightly in the lower bound from [LST22].*

5 Limitations on the technique for higher depths

We also claim a limitation for higher depths.

Theorem 27. *For any constant $\Delta \geqslant 2$, let $\Gamma = \Gamma(\Delta) = \Delta^{(\log \Delta)/10}$. Let $W \subseteq [-1,1]$ be any multiset such that $|W| \leqslant d$ and $|\text{Sum}(W)| \leqslant 1$. Then, there is a W-tree T of depth Δ and node bias $O(d^{1/\Gamma})$. (Here the constant implicit in the $O(\cdot)$ notation depends on Δ.)*

We follow the high-level outline of the depth-3 case, but with a slightly different recursion. For any $\Delta \geqslant 2$, let $\varepsilon_\Delta = \log^2 \Delta/\Delta$. When Δ is clear from context, we will simply say ε instead of ε_Δ.

Let Nodebias(Δ, d) be defined as follows.

$$\text{Nodebias}(\Delta, d) = \max_{W\subseteq[-1,1]:|W|\leqslant d, |\text{Sum}(W)|\leqslant 1} \ \min_{W\text{-trees } T} \text{Nodebias}_W(T).$$

We will show, by induction on Δ that for all $d, \Delta \geqslant 2$, Nodebias$(\Delta, d) \leqslant C_\Delta d^{1/\Gamma(\Delta)}$ where $C_\Delta = 2 \cdot 2^{\Delta\Gamma(\Delta)} \geqslant 8$ is a constant that only depends on Δ. By Lemma 20, this is clear for Δ such that $\Gamma(\Delta) \leqslant \Delta$ (and hence $\log \Delta \leqslant 10$).

Now consider a Δ such that $\log \Delta > 10$ (i.e. $\Delta \geqslant 1025$). Consider the following procedure.

1. Preprocessing: By the Preprocessing lemma (Lemma 19), it suffices to consider multisets W such that $|W\backslash([-1,-1/2] \cup [1/2,1])| \leqslant 2$.

2. Apply the following recursive algorithm on the input W.

   Algorithm $\mathcal{A}_\Delta(W)$:

   Assignment $d := |W|$.

   Initialization: If $d \leqslant C_\Delta$ then return the depth-1 W-tree of node bias at most C_Δ.

   Phase 1: For as long as possible, find pairwise disjoint subsets $A \subseteq W$ such that $|A| \leqslant 2d^{1-\varepsilon}$ and
   $$\frac{|\text{Sum}(A)|}{|A|} \leqslant \frac{1}{d}.$$

When this is no longer possible, let $A_1, \ldots, A_{e_1}$ be the sequence of sets picked and let $W_1' = A_1 \cup A_2 \cdots \cup A_{e_1}$. Using the balancing lemma (Lemma 22), let $W_1 = W_1' \cup \{a_1, \ldots, a_{f_1}\} \subseteq W$ be such that $\sum_{i \leqslant f_1} |a_i| \leqslant |\mathrm{Sum}(W_1')| + 1$ and $|\mathrm{Sum}(W_1)|, |\mathrm{Sum}(W \backslash W_1)| \leqslant 1$.

Let T_1 be a W_1-tree constructed as follows.

For each $i \in [e_1]$, let T_i' be an A_i-tree of depth at most $\Delta - 1$, and node bias at most $\mathrm{Nodebias}(\Delta - 1, 2d^{1-\varepsilon})$. The tree T_1 has depth Δ, where the root has as children the roots of $T_1', \ldots, T_{e_1}'$, and f_1 leaves labelled $a_1, \ldots, a_{f_1}$.

Set $W' = W \backslash W_1$ and continue.

Phase 2: As long as it is possible, find pairwise disjoint sets $B \subseteq W'$ such that $1 \leqslant |B| \leqslant 4d^{\varepsilon}$ and

$$|\mathrm{Sum}(B)| \leqslant \frac{4}{d^{1-\varepsilon}}.$$

When this is no longer possible, let $B_1, \ldots, B_{e_2}$ be the sequence of sets picked and let $W_2' = B_1 \cup B_2 \cdots \cup B_{e_2}$. Using the balancing lemma (Lemma 22), let $W_2 = W_2' \cup \{b_1, \ldots, b_{f_2}\} \subseteq W'$ be such that $\sum_{i \leqslant f_2} |b_i| \leqslant |\mathrm{Sum}(W_2')| + 1$ and $|\mathrm{Sum}(W_2)|, |\mathrm{Sum}(W' \backslash W_2)| \leqslant 1$.

Construct a W_2-tree T_2 in the following way.

Repeatedly pick $t = \lfloor d^{1-\varepsilon}/4 \rfloor$ many B_is. Their union B' satisfies $|\mathrm{Sum}(B')| \leqslant 1$. Do this as many times as possible (the last time we pick at most t many B_i to form such a B'). Let $\{B_1', \ldots, B_{e_2'}'\}$ be the set of B's obtained this way. For each B_i' ($i \in [e_2']$), we obtain a B_i'-tree T_i'' as follows. Say $B_i' = \bigcup_{j \in S_i} B_j$ where $|S_i| \leqslant t$. We have $e_2' \leqslant \left\lfloor \frac{d}{\lfloor d^{1-\varepsilon}/4 \rfloor} \right\rfloor + 1 \leqslant 6d^{\varepsilon}$. For each $j \in S_i$, we construct a B_j-tree $T_{i,j}'$ of depth at most $\lfloor (\Delta - 1)/2 \rfloor$ and node bias at most $\mathrm{Nodebias}(\lfloor (\Delta-1)/2 \rfloor, 4d^{\varepsilon})$. The tree T_i'' is a tree of depth at most $1 + \lfloor (\Delta - 1)/2 \rfloor$ where the root is connected to various subtrees $T_{i,j}'$.

Now construct the grouping $\tilde{W}'$ according to the partition $P_2 = \{B_1', \ldots, B_{e_2'}', b_1, \ldots, b_{f_2}\}$ of W_2. We know that $\sum |b_i| \leqslant 1 + \sum_{i=1}^{e_2} |\mathrm{Sum}(B_i)| \leqslant 1 + 4d^{\varepsilon}$. So, since at most two b_is have norm lower than $1/2$, it implies $f_2 \leqslant 4 + 8d^{\varepsilon}$ and $e_2' + f_2 \leqslant 15d^{\varepsilon}$ (since $d^{\varepsilon} \geqslant 4$). Let $\tilde{T}'$ denote a $\tilde{W}'$-tree of node bias at most $\mathrm{Nodebias}(\lfloor (\Delta-1)/2 \rfloor, 15d^{\varepsilon})$. Using the tree $\tilde{T}'$ for $\tilde{W}'$ and the tree T_i'' for each B_i' from the last paragraph, we construct a W_2-tree T_2 of depth at most Δ as given by Lemma 18.

Set $W'' = W' \backslash W_2$ and continue.

Recursive call: Compute $T_3 = \mathcal{A}_{\Delta}(W'')$.

Return The W-tree T of node bias at most $\beta_1 + \beta_2 + \beta_3$ where $\beta_i = \mathrm{Nodebias}(T_i)$ (for $i \in [3]$) given by T_1, T_2, T_3 and Lemma 21.

We now analyze the bias of the tree T output by the above procedure. The main observation is the following.

Lemma 28. *Fix any positive integer $\Delta \geqslant 2$. For $d > C_{\Delta}$, after Phases 1 and 2 of the algorithm $\mathcal{A}_{\Delta}(W)$, we have $|W''| \leqslant d/2$.*

We assume the above lemma and finish our analysis of $\mathrm{Nodebias}(T)$. We need to show that

$$\mathrm{Nodebias}(T) \leqslant C_{\Delta} d^{1/\Gamma(\Delta)} \tag{8}$$

for each input W to $\mathcal{A}_\Delta$ of size at most d. We prove this by induction on $d = |W|$. The claim is trivial for $d \leqslant C_\Delta$ (as $\mathrm{Nodebias}(T) \leqslant d$ for any W-tree T). So we assume that $d > C_\Delta$. In particular, as $\Gamma(\Delta) > \Delta \geqslant 1025$, we have $d > \left(2^{1025}\right)^\Delta$.

For $i \in \{1, 2, 3\}$, let β_i denote the node bias of tree T_i constructed by the algorithm. By Lemma 21 (the pasting lemma), we know that $\mathrm{Nodebias}(T) \leqslant \beta_1 + \beta_2 + \beta_3$. By Lemma 28 and induction, we have

$$\beta_3 \leqslant C_\Delta \left(\frac{d}{2}\right)^{1/\Gamma(\Delta)}. \tag{9}$$

We now bound β_1 and β_2. To bound β_1, we note that

- each internal non-root node of T_1 has node bias at most $\mathrm{Nodebias}(\Delta - 1, 2d^{1-\varepsilon})$, and
- the root node of T_1 has node bias at most 3, as we can bound the node bias of the root node by

$$\sum_{i=1}^{e_1} |\mathrm{Sum}(A_i)| + \sum_{j=1}^{f_1} |a_j| \leqslant 2\sum_{i=1}^{e_1} |\mathrm{Sum}(A_i)| + 1 \leqslant 2\sum_{i=1}^{e_1} \frac{|A_i|}{d} + 1 \leqslant 3,$$

  where for the first inequality we have used the fact that $\sum_{i \leqslant f_1} |a_i| \leqslant |\mathrm{Sum}(W_1')| + 1 \leqslant \sum_{i=1}^{e_1} |\mathrm{Sum}(A_i)| + 1$; for the second inequality we have used the fact that $|\mathrm{Sum}(A_i)| \leqslant |A_i|/d$; and for the third inequality we have used the fact that $\sum_{i=1}^{e_1} |A_i| \leqslant d$ as these sets are pairwise disjoint.

We have thus shown that $\beta_1 \leqslant \max\{\mathrm{Nodebias}(\Delta - 1, 2d^{1-\varepsilon}), 3\}$. By induction on Δ, we know therefore that

$$\beta_1 \leqslant C_{\Delta-1} 2^{1/\Gamma(\Delta-1)} d^{(1-\varepsilon)/\Gamma(\Delta-1)} \leqslant C_\Delta d^{(1-\varepsilon)/\Gamma(\Delta-1)}. \tag{10}$$

To bound β_2, observe that by Lemma 18, we have $\beta_2 \leqslant \max\{\mathrm{Nodebias}(\tilde{T}')\} \cup \{\mathrm{Nodebias}(T_i'') \mid i \in [e_2']\}$. By definition, we have $\mathrm{Nodebias}(\tilde{T}') \leqslant \beta' := \mathrm{Nodebias}(\lfloor \Delta - 1/2 \rfloor, 15d^\varepsilon)$. To bound $\mathrm{Nodebias}(T_i'')$, we note that

- each internal non-root vertex of T_i'' is a vertex of some $T_{i,j}'$ (for some $j \in [t]$) and hence has node bias at most β',
- the root node has at most $d^{1-\varepsilon}/4$ children, and the sum of each child is at most $4/d^{1-\varepsilon}$, hence the root node has node bias at most $1 \leqslant \beta'$.

Hence, we see that $\mathrm{Nodebias}(T_i') \leqslant \beta'$ as well. Finally, $\beta_2 \leqslant \beta'$ and so by the induction hypothesis, we see that $\beta_2 \leqslant C_{\Delta/2} 15^{1/\Gamma(\lfloor \Delta - 1/2 \rfloor)} d^{\varepsilon/\Gamma(\lfloor (\Delta-1)/2 \rfloor)} \leqslant C_\Delta d^{\varepsilon/\Gamma(\lfloor (\Delta-1)/2 \rfloor)}$. Putting this together with (9) and (10), we get

$$\begin{aligned}\mathrm{Nodebias}(T) &\leqslant \beta_1 + \beta_2 + \beta_3 \\ &\leqslant C_\Delta d^{(1-\varepsilon)/\Gamma(\Delta-1)} + C_\Delta d^{\varepsilon/\Gamma(\lfloor (\Delta-1)/2 \rfloor)} + C_\Delta \left(\frac{d}{2}\right)^{1/\Gamma(\Delta)}.\end{aligned}$$

Thus, to prove the inductive statement (8), it suffices to show

$$d^{(1-\varepsilon)/\Gamma(\Delta-1)} + d^{\varepsilon/\Gamma(\lfloor (\Delta-1)/2 \rfloor)} + \left(\frac{d}{2}\right)^{1/\Gamma(\Delta)} \leqslant d^{1/\Gamma(\Delta)}. \tag{11}$$

The above in turn is implied by the following sequence of inequalities.

$$d^{1/\Gamma(\Delta)} - \left(\frac{d}{2}\right)^{1/\Gamma(\Delta)} \geqslant \frac{d^{1/\Gamma(\Delta)}}{2\Gamma(\Delta)} \tag{12}$$

$$d^{(1-\varepsilon)/\Gamma(\Delta-1)} \leqslant \frac{d^{1/\Gamma(\Delta)}}{4\Gamma(\Delta)} \tag{13}$$

$$d^{\varepsilon/\Gamma(\lfloor(\Delta-1)/2\rfloor)} \leqslant \frac{d^{1/\Gamma(\Delta)}}{4\Gamma(\Delta)} \tag{14}$$

We now prove each of (12), (13) and (14) in turn.

Proof of (12).

$$d^{1/\Gamma(\Delta)} - \left(\frac{d}{2}\right)^{1/\Gamma(\Delta)} = d^{1/\Gamma(\Delta)} \cdot \left(1 - \frac{1}{2^{1/\Gamma(\Delta)}}\right) \geqslant d^{1/\Gamma(\Delta)} \cdot \frac{1}{2\Gamma(\Delta)}$$

where for the final inequality we have used the standard fact that $1 - 2^{-x} \geqslant x/2$ for $x \in [0, 1/2]$.

Proof of (13). We note that

$$\frac{\Gamma(\Delta-1)}{\Gamma(\Delta)} = \exp(\log e \cdot (\ln^2(\Delta-1) - \ln^2\Delta)/10).$$

The derivative of the function $f(x) = \ln^2 x$ is at most $2\ln(\Delta-1)/(\Delta-1)$ as x ranges over the interval $[\Delta-1, \Delta]$. Thus by the Mean Value theorem, we can lower bound the final term in the above computation by $\exp(-2\log(e)\ln(\Delta-1)/10(\Delta-1)) = \exp(-\log(\Delta-1)/5(\Delta-1))$. Using the standard inequality $e^x \geqslant 1 + x$ for all $x \in \mathbb{R}$, we get

$$\frac{\Gamma(\Delta-1)}{\Gamma(\Delta)} \geqslant 1 - \frac{\log(\Delta-1)}{5(\Delta-1)} \geqslant 1 - \frac{\varepsilon}{2}.$$

where the final inequality uses the fact that $\varepsilon = \frac{\log^2\Delta}{\Delta} \geqslant \frac{(4/5)\log(\Delta-1)}{2(\Delta-1)} = \frac{2\log(\Delta-1)}{5(\Delta-1)}$.

In particular, we get

$$\frac{1-\varepsilon}{\Gamma(\Delta-1)} = \frac{1-\varepsilon/2}{\Gamma(\Delta-1)} - \frac{\varepsilon}{2\Gamma(\Delta-1)} \leqslant \frac{1}{\Gamma(\Delta)} - \frac{\log(4\Gamma(\Delta))}{\log d}, \tag{15}$$

where the final inequality follows from the following computation

$$\log d \geqslant \Gamma(\Delta)\Delta \geqslant \Gamma(\Delta-1)\Delta \cdot \frac{4 + 2\log^2(\Delta)/10}{\log^2\Delta} = \Gamma(\Delta-1) \cdot \log(4\Gamma(\Delta)) \cdot \frac{2}{\varepsilon}$$

implying the claimed inequality.

Exponentiating both sides of (15) with base d yields (13).

Proof of (14). Similarly, some elementary calculus as above give

$$\begin{aligned}\frac{\Gamma(\lfloor(\Delta-1)/2\rfloor)}{\Gamma(\Delta)} &\geqslant \frac{(7\Delta/15)^{\log(7\Delta/15)/10}}{\Delta^{(\log\Delta)/10}} = \exp\left(-\frac{\ln 2}{10}(\log^2(\Delta) - \log^2(7\Delta/15))\right) \\ &\geqslant \exp\left(-\frac{\ln 2}{10} \cdot \frac{2}{\ln 2}\frac{\log(7\Delta/15)}{7\Delta/15} \cdot \frac{8\Delta}{15}\right) \geqslant \left(\frac{15}{7\Delta}\right)^{8/35\ln 2} \geqslant \frac{5}{4\Delta^{1/3}} \geqslant (5/4)\varepsilon.\end{aligned}$$

We therefore have

$$\frac{\varepsilon}{\Gamma(\lfloor(\Delta-1)/2\rfloor)} = \frac{(5/4)\varepsilon}{\Gamma(\lfloor(\Delta-1)/2\rfloor)} - \frac{(1/4)\varepsilon}{\Gamma(\lfloor(\Delta-1)/2\rfloor)} \leqslant \frac{1}{\Gamma(\Delta)} - \frac{\varepsilon}{4\Gamma(\lfloor(\Delta-1)/2\rfloor)} \leqslant \frac{1}{\Gamma(\Delta)} - \frac{\log(4\Gamma(\Delta))}{\log d},$$

where the last inequality follows from

$$\log d \geqslant \Gamma(\Delta)\Delta \geqslant \Gamma(\lfloor\Delta-1/2\rfloor)\Delta \cdot \frac{8+4\log^2(\Delta)/10}{\log^2\Delta} = \Gamma(\lfloor\Delta-1/2\rfloor)\cdot\log(4\Gamma(\Delta))\cdot\frac{4}{\varepsilon}.$$

Exponentiating both sides of the above inequality with base d yields (14).

We have thus proved (12), (13) and (14), which implies (11). This in turn implies the claimed bound on $\mathrm{Nodebias}(T)$ in (8), which proves the inductive statement and hence the theorem.

We now prove Lemma 28.

Proof of Lemma 28. Again, we will show that if $d'' = |W''| \geqslant d/2$ (and the other hypotheses of the lemma), Phases 1 and 2 of the algorithm $\mathcal{A}_\Delta$ could not have concluded, hence deriving a contradiction.

Let W''_+ and W''_- denote the positive and negative elements of W'' respectively. As $W'' \subseteq W$ where the latter set contains at most 2 elements of absolute value less than 1/2 (by the preprocessing step), and $|\mathrm{Sum}(W'')| \leqslant 1$, it is easy to see that $|W''_+|, |W''_-| \geqslant d'''$ for some $d''' \geqslant (d''-4)/3$. Note that $d''' \geqslant d/10 \geqslant d^{1-\varepsilon}$, where we have used the facts that $d \geqslant C_\Delta \geqslant \max(10^\Delta, 20)$ and our assumption that $d'' \geqslant d/2$.

By Lemma 9, we know that there is a set $T \subseteq W''$ such that $|T| \leqslant 2d^{1-\varepsilon}$ and $|\mathrm{Sum}(T)| \leqslant 4/(2d^{1-\varepsilon}-1) \leqslant 4/d^{1-\varepsilon}$.

We now apply a simple case analysis. If $|T| > 4d^\varepsilon$, we have a set of the form that we look for in Phase 1. Finding such a T implies that Phase 1 could not have concluded. On the other hand, if $|T| \leqslant 4d^\varepsilon$, we see that Phase 2 could not have concluded. In either case, we have the required contradiction, and hence we are done. □

References

[Ajt83] Miklós Ajtai. Σ_1^1-formulae on finite structures. *Annals of Pure and Applied Logic*, 24(1):1 – 48, 1983.

[BÖ0] Peter Bürgisser. Cook's versus valiant's hypothesis. *Theoretical Computer Science*, 235(1):71–88, 2000.

[BCS97] Peter Bürgisser, Michael Clausen, and Mohammad Amin Shokrollahi. *Algebraic complexity theory*, volume 315 of *Grundlehren der mathematischen Wissenschaften*. Springer, 1997.

[EGdOW18] Klim Efremenko, Ankit Garg, Rafael Mendes de Oliveira, and Avi Wigderson. Barriers for rank methods in arithmetic complexity. In *Proceedings of the 9th Innovations in Theoretical Computer Science Conference (ITCS)*, volume 94 of *LIPIcs*, pages 1–19, 2018.

[ELSW18] Klim Efremenko, J. M. Landsberg, Hal Schenck, and Jerzy Weyman. The method of shifted partial derivatives cannot separate the permanent from the determinant. *Mathematics of Computation*, 87(312):2037–2045, 2018.

[FSS84] Merrick Furst, James B. Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Mathematical systems theory*, 17(1):13–27, 1984.

[FSV18] Michael A. Forbes, Amir Shpilka, and Ben Lee Volk. Succinct hitting sets and barriers to proving lower bounds for algebraic circuits. *Theory of Computing*, 14(1):1–45, 2018.

[GK98] Dima Grigoriev and Marek Karpinski. An exponential lower bound for depth 3 arithmetic circuits. In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing (STOC)*, pages 577–582, 1998.

[GKKS16] Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Arithmetic circuits: A chasm at depth 3. *SIAM Journal of Computing*, 45(3):1064–1079, 2016.

[GKSS17] Joshua A. Grochow, Mrinal Kumar, Michael E. Saks, and Shubhangi Saraf. Towards an algebraic natural proofs barrier via polynomial identity testing. *CoRR*, abs/1701.01717, 2017.

[GR00] Dima Grigoriev and Alexander A. Razborov. Exponential lower bounds for depth 3 arithmetic circuits in algebras of functions over finite fields. *Applicable Algebra in Engineering, Communication and Computing*, 10(6):465–487, 2000.

[Gro15] Joshua A. Grochow. Unifying known lower bounds via geometric complexity theory. *Computational Complexity*, 24(2):393–475, 2015.

[H$\mathring{8}$6] Johan Håstad. Almost optimal lower bounds for small depth circuits. In *Proceedings of the 18th Annual ACM Symposium on Theory of Computing (STOC)*, page 6–20, 1986.

[Juk11] Stasys Jukna. *Extremal Combinatorics - With Applications in Computer Science*. Texts in Theoretical Computer Science. An EATCS Series. Springer, 2011.

[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.

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

[LST]

[LST21] Nutan Limaye, Srikanth Srinivasan, and Sébastien Tavenas. Set-multilinear and non-commutative formula lower bounds for iterated matrix multiplication. *To appear in STOC 2022. Electron. Colloquium Comput. Complex.*, page 94, 2021.

[LST22] Nutan Limaye, Srikanth Srinivasan, and Sébastien Tavenas. Superpolynomial lower bounds against low-depth algebraic circuits, 2022.

[NW97] Noam Nisan and Avi Wigderson. Lower bounds on arithmetic circuits via partial derivatives. *Computational Complexity*, 6(3):217–234, 1997.

[Raz86] Alexander A. Razborov. Lower bounds on the size of constant-depth networks over a complete basis with logical addition. *Mathematicheskie Zametki*, 41(2):598–607, 1986.

[Raz09] Ran Raz. Multi-linear formulas for permanent and determinant are of super-polynomial size. *Journal of the ACM*, 56(2):8:1–8:17, 2009.

[Raz13] Ran Raz. Tensor-rank and lower bounds for arithmetic formulas. *Journal of the ACM*, 60(6):40:1–40:15, 2013.

[SA08] Andrew Drucker Scott Aaronson. Arithmetic natural proofs theory is sought. *Blog post*, 24(1):1 – 48, 2008.

[Sap15] Ramprasad Saptharishi. A survey of lower bounds in arithmetic circuit complexity. Github survey, 2015.

[Smo87] Roman Smolensky. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing (STOC)*, page 77–82, 1987.

[SY10] Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science*, 5:207–388, 2010.

[Val79] Leslie G. Valiant. Completeness classes in algebra. In *Proceedings of the 11h Annual ACM Symposium on Theory of Computing (STOC)*, pages 249–261. ACM, 1979.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il