Lower Bounds for Set-Multilinear Branching Programs

Prerona Chatterjee
School of Computer Sciences, NISER Bhubaneswar, India

Deepanshu Kush
Department of Computer Science, University of Toronto, Canada

Shubhangi Saraf
Department of Computer Science, University of Toronto, Canada
Department of Mathematics, University of Toronto, Canada

Amir Shpilka
Blavatnik School of Computer Science, Tel-Aviv University, Israel

Abstract

In this paper, we prove super-polynomial lower bounds for the model of *sum of ordered set-multilinear algebraic branching programs*, each with a possibly different ordering ($\sum$smABP). Specifically, we give an explicit nd-variate polynomial of degree d such that any $\sum$smABP computing it must have size $n^{\omega(1)}$ for d as low as $\omega(\log n)$. Notably, this constitutes the first such lower bound in the low degree regime. Moreover, for $d = \text{poly}(n)$, we demonstrate an exponential lower bound. This result generalizes the seminal work of Nisan (STOC, 1991), which proved an exponential lower bound for a single ordered set-multilinear ABP.

The significance of our lower bounds is underscored by the recent work of Bhargav, Dwivedi, and Saxena (TAMC, 2024), which showed that super-polynomial lower bounds against a sum of ordered set-multilinear branching programs – for a polynomial of sufficiently low degree – would imply super-polynomial lower bounds against general ABPs, thereby resolving Valiant's longstanding conjecture that the permanent polynomial can not be computed efficiently by ABPs. More precisely, their work shows that if one could obtain such lower bounds when the degree is bounded by $O(\log n/\log\log n)$, then it would imply super-polynomial lower bounds against general ABPs.

Our results strengthen the works of Arvind & Raja (Chic. J. Theor. Comput. Sci., 2016) and Bhargav, Dwivedi & Saxena (TAMC, 2024), as well as the works of Ramya & Rao (Theor. Comput. Sci., 2020) and Ghoshal & Rao (International Computer Science Symposium in Russia, 2021), each of which established lower bounds for related or restricted versions of this model. They also strongly answer a question from the former two, which asked to prove super-polynomial lower bounds for general $\sum$smABP.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory

Keywords and phrases Lower Bounds, Algebraic Branching Programs, Set-multilinear polynomials

Digital Object Identifier 10.4230/LIPIcs.CCC.2024.20

Related Version *Full Version*: https://eccc.weizmann.ac.il/report/2023/212/
Full Version: https://arxiv.org/abs/2312.15874

Funding *Prerona Chatterjee*: This work was done as a postdoctoral student at Tel Aviv University, where the research was funded by the Azrieli International Postdoctoral Fellowship, the Israel Science Foundation (grant number 514/20) and the Len Blavatnik and the Blavatnik Family foundation.
Shubhangi Saraf: Research partially supported by a Sloan research fellowship and an NSERC Discovery Grant.
Amir Shpilka: Research leading to these results has received funding from the Israel Science Foundation (grant number 514/20) and from the Len Blavatnik and the Blavatnik Family foundation.

39th Computational Complexity Conference (CCC 2024).
Editor: Rahul Santhanam; Article No. 20; pp. 20:1–20:20
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Acknowledgements Parts of this work were done while the first author was visiting TIFR Mumbai and ICTS-TIFR Bengaluru, and she would like to thank Venkata Susmita Biswas, Ramprasad Saptharishi, Prahladh Harsha and Jaikumar Radhakrishnan for the hospitality. The first author would also like to thank Anamay Tengse for useful discussions.

1 Introduction

1.1 Background on Algberaic Complexity

In his seminal work ([40]) in 1979, Valiant proposed an algebraic framework to study the computational complexity of computing *polynomials*. *Algebraic Complexity Theory* is this study of the complexity of computational problems which can be described as computing a multivariate polynomial $P(x_1, \ldots, x_N)$ over some elements $x_1, \ldots, x_N$ lying in a fixed field . Several fundamental computational tasks such as computing the determinant, permanent, matrix product, etc., can be represented using this framework. The natural computational models that we investigate in this setting are models such as *algebraic circuits*, *algebraic branching programs*, and *algebraic formulas*.

An *algebraic circuit* over a field for a multivariate polynomial $P(x_1, \ldots, x_N)$ is a directed acyclic graph (DAG) whose internal vertices (called gates) are labeled as either $+$ (sum) or $\times$ (product), and leaves (vertices of in-degree zero) are labeled by the variables x_i or constants from . A special output gate (the root of the DAG) represents the polynomial P. If the DAG happens to be a tree, such a resulting circuit is called an *algebraic formula*. The size of a circuit or formula is the number of nodes in the DAG. We also consider the product-depth of the circuit, which is the maximum number of product gates on a root-to-leaf path. The class VP (respectively, VF) is then defined to be the collection of all polynomials having at most polynomially large degree which can be computed by polynomial-sized circuits (respectively, formulas).

The class VP is synonymous to what we understand as efficiently computable polynomials. The class VNP, whose definition is similar to the boolean class NP, is in some sense a notion of what we deem as *explicit*. Much like the problem of proving circuit size lower bounds for explicit boolean functions, the problem of proving them for explicit *polynomials* (i.e., showing VP $\neq$ VNP) has also remained elusive for many decades. However, because the latter only deals with formal *symbolic* computation as opposed to modelling *semantic* truth-table constraints, it is widely believed to be easier to resolve than its boolean counterpart. In fact, it is even known to be a *pre-requisite* to the P $\neq$ NP conjecture in the non-uniform setting ([8]).

An *algebraic branching program* (ABP) is a layered DAG with two special nodes in it: a start-node and an end-node. All edges of the ABP go from layer $\ell - 1$ to layer ℓ for some ℓ (say start-node is the unique node in layer 0 and end-node is the unique node in the last layer) and are labeled by a linear polynomial. Every directed path γ from start-node to end-node computes the monomial P_γ, which is the product of all labels on the path γ. The ABP computes the polynomial $P = \sum_\gamma P_\gamma$, where the sum is over all paths γ from start-node to end-node. Its size is simply the number of nodes in the DAG, its *depth* is the length of the longest path from the start-node to the end-node, and *width* is the maximum number of nodes in any layer. The class VBP is then defined to be the collection of all polynomials (with polynomially-bounded degree) which can be computed by polynomial-sized branching programs. ABPs are known to be of intermediate complexity between formulas and circuits; in other words, we know the inclusions VF $\subseteq$ VBP $\subseteq$ VP $\subseteq$ VNP.

It is conjectured that all of these inclusions are strict, and resolving any of these conjectures would represent a dramatic advancement in algebraic complexity theory, and even more broadly, in circuit complexity overall. Valiant's original hypothesis in [40] pertains to showing a super-polynomial separation between the complexity of computing the determinant and the permanent polynomials. This is known to be equivalent to the $\mathsf{VBP} \neq \mathsf{VNP}$ conjecture, i.e., showing super-polynomial size lower bounds against ABPs computing *explicit* polynomials. At present, the best known lower bound against ABPs is only quadratic ([9]), and it appears as though we are quite distant from addressing this conjecture. On the other hand, as we now elaborate, while not directly improving upon this quadratic bound, this paper makes significant progress towards a different line of attack aimed at resolving Valiant's conjecture.

1.2 Set-Multilinearity: A Key Syntactic Restriction

One key advantage that algebraic models offer over their boolean counterparts is that of *syntactic* restrictions. A recurring theme in algebraic complexity theory is to first efficiently convert general models of computation (such as circuits or formulas) to special kinds of syntactically-restricted models, show strong lower bounds against these restricted models, and then recover non-trivial lower bounds against the original general models owing to the efficiency of this conversion. This phenomenon is termed *hardness escalation*. In this subsection, we describe one crucial example of a syntactic restriction in detail, that of *set-multilinearity*.

A polynomial is said to be *homogeneous* if each monomial has the same total degree and *multilinear* if every variable occurs at most once in any monomial. Now, suppose that the underlying variable set is partitioned into d sets $X_1, \dots, X_d$. Then the polynomial is said to be *set-multilinear* with respect to this variable partition if each monomial in P has *exactly* one variable from each set. Note that a set-multilinear polynomial is both multilinear and homogeneous, and has degree precisely d if it is set-multilinear over d sets. Next, we define different models of computation corresponding to these variants of polynomials classes. An algebraic formula/branching program/circuit is set-multilinear with respect to a variable partition $(X_1, \dots, X_d)$ if each internal node in the formula/branching program/circuit computes a set-multilinear polynomial.[1] Multilinear and homogeneous formulas/branching programs/circuits are defined analogously.

We now describe several important hardness escalation results, each reducing general models to corresponding *set-multilinear* models.

Constant depth circuits

The recent celebrated breakthrough work of Limaye, Srinivasan, and Tavenas ([27]) establishes super-polynomial lower bounds for general algebraic circuits for *all* constant-depths, a problem that was open for many decades. In order to show this, it is first shown that general low-depth algebraic formulas can be converted to set-multilinear algebraic formulas of low depth as well, and without much of a blow-up in size (as long as the degree is small). Subsequently, strong lower bounds are established for low-depth set-multilinear circuits (of small enough degree), which when combined with the first step yields the desired lower bound for general constant-depth circuits.

[1] Of course, a non-root node need not be set-multilinear with respect to the *entire* variable partition. Nevertheless, here we demand that it must be set-multilinear with respect to some *subset* of the collection $\{X_1, \dots, X_d\}$.

Formulas

Raz [33] showed that if an N-variate set-multilinear polynomial of degree d has an algebraic formula of size s, then it also has a set-multilinear formula of size $\mathrm{poly}(s) \cdot (\log s)^d$. In particular, for a set-multilinear polynomial P of degree $d = O(\log N/\log\log N)$, it follows that P has a formula of size $\mathrm{poly}(N)$ if and only if P has a set-multilinear formula of size $\mathrm{poly}(N)$. Thus, having $N^{\omega_d(1)}$ set-multilinear formula size lower bounds for such a low degree would imply super-polynomial lower bounds for general formulas. A recent line of work by Kush and Saraf ([25, 26]) can be viewed as an attempt to prove general formula lower bounds via this route.

Algebraic Branching Programs

In fact, in the context of ABPs as well, the very recent work of Bhargav, Dwivedi, and Saxena ([5]) reduces the problem of showing lower bounds against general ABPs to proving lower bounds against sums of ordered set-multlinear ABPs (again, as long as the degree is small enough). Ordered set-multilinear ABPs are, in fact, historically well-studied models and extremely well-understood. However, despite their apparent simplicity, the work [5] implies that understanding their *sums* – a model that is far less studied – is at the forefront of understanding Valiant's conjecture. We state their result formally as Theorem 1.5 in Section 1.3.

First however, as this is also the main model considered in this paper, we begin by formally defining *ordered* set-multilinear ABPs and outlining their importance.

▶ **Definition 1.1** (Ordered smABP). *Given a variable partition $(X_1, \ldots, X_d)$, we say that a set-multilinear branching program of depth d is said to be* ordered *with respect to an ordering (or permutation) $\sigma \in S_d$ if for each $\ell \in [d]$, all edges of the ABP from layer $\ell - 1$ to layer ℓ are labeled using a linear form over the variables in $X_{\sigma(\ell)}$. It is simply said to be* ordered *if there* exists *an ordering σ such that it is ordered with respect to σ.*

At this point, it is essential to take note of the terminology in this context: in this paper, a general (or "unordered") set-multilinear branching program refers to an ABP for which each internal node computes a polynomial that is set-multilinear with respect to *some* subset of the global partition, whereas an *ordered* set-multilinear branching program is more specialized and has the property that any two nodes in the same layer compute polynomials that are set-multilinear with respect to the *same* partition.

▶ Remark 1.2. This notion of ordered set-multilinear branching programs turns out to be equivalent to the more commonly used notions of (i) "read-once oblivious algebraic branching programs (ROABPs)", as well as (ii) "non-commutative algebraic branching programs" (see, for example, [12]). This relationship, especially with the former model, is described in more detail later in Section 1.4.

▶ **Definition 1.3** ($\sum$smABP). *Given a polynomial $P(X)$ that is set-multilinear with respect to the variable partition $X = (X_1, \ldots, X_d)$, we say that $\sum_{i=1}^{t} A_i$ is a $\sum$smABP computing P if indeed $\sum_{i=1}^{t} A_i(X) = P(X)$, and each A_i is an ordered set-multilinear branching program i.e., each A_i is ordered with respect to some $\sigma_i \in S_d$. We call t (i.e., the number of summands in a $\sum$smABP) its* support size *and define its* max-width *and* total-width *to be the maximum over the width of each A_i and the sum of the width of each A_i, respectively.*

We have known *exponential* width lower bounds against a *single* ordered set-multilinear ABP since the foundational work of Nisan. In [29], he showed that there are explicit polynomials (in fact, in VP) which require any ordered set-multilinear ABP computing them to be of exponentially large width. Viewed differently, this work even shows that in the non-commutative setting, $\mathsf{VBP} \neq \mathsf{VP}$[2]. More crucially however, this work introduced a powerful technique – a notion known as the *partial derivative method* – that has been instrumental in the bulk of the major advancements in algebraic complexity theory over the past three decades (such as [30, 32, 36, 19, 23, 20, 24, 27, 39], see also [38, 37]).

Despite the considerable development of the partial derivative technique over the course of these works (and many more) for proving strong lower bounds against various algebraic models, relatively little is known about a *general sum* of ordered set-multilinear ABPs – a simple and direct generalization of the original model considered by Nisan. There is some progress in the literature towards this goal but which still requires additional structural restrictions on either the max-width or the support size or the size of each part in the variable partition. The work [3] of Arvind and Raja shows that any $\sum$smABP of support size t computing the $n \times n$ permanent polynomial requires max-width (and therefore, total-width) at least $2^{\Omega(n/t)}$. Note that for this bound to be super-polynomial, the support size needs to be heavily restricted i.e., t must be sub-linear. On the other hand, the work [5] also shows a super-polynomial lower bound in this context: it implies that no $\sum$smABP of polynomially-bounded total-width can compute the iterated matrix multiplication (IMM) polynomial. However, their work requires the additional assumption that the max-width of such an $\sum$smABP is $n^{o(1)}$, that is *sub-polynomial* in the number of variables.

Apart from these, Ramya and Rao ([31]) use the partial derivative method to show an exponential lower bound against the related model of sum of ROABPs in the multilinear setting, as well as some other structured multilinear ABPs. Their lower bounds are for a multilinear polynomial that is computable by a small multilinear circuit. Ghoshal and Rao ([13]) partially extend their work by proving an exponential lower bound, for a polynomial that is computable even by a small multilinear ABP, against sums of ROABPs that have polynomially bounded width. Notably, these results can be viewed as lower bounds against the $\sum$smABP model where each variable set in the variable partition has size 2 (that is, the total number of variables is $2d$). This is because a multilinear polynomial and any multilinear model computing it (such as circuit, formula, or branching program) can be converted, in a generic manner, to a set-multilinear polynomial and the corresponding set-multilinear model respectively, with each variable set having size 2 (also see Section 1.5 for a discussion). However, from the perspective of hardness escalation of [5] that is described above – and which is indeed the focus of our work – the setting of d that is far more interesting is when it is allowed to be considerably smaller than n. More precisely, the framework of [5] requires $d = O(\log n/\log\log n)$ (stated formally as Theorem 1.5 below). A detailed discussion about the results in [31], [13] and how they compare with our work can be found in Section 1.5.

1.3 Our Results

Our main result is in this paper is a super-polynomial lower bound against an unrestricted sum of ordered set-multilinear branching programs, for a hard polynomial with "small" degree. We first state this result formally below, and then subsequently explain the connection with the hardness escalation result of [5] that is alluded to in the previous subsections.

▶ **Theorem 1.4** ("Low"-Degree $\sum$smABP Lower Bounds). *Let $d \leq n$ be growing parameters satisfying $d = \omega(\log n)$. There is a $\Theta(dn)$-variate degree d set-multilinear polynomial $F_{n,d}$ in* VP *such that $F_{n,d}$ cannot be computed by a $\sum$smABP of total-width* $\mathrm{poly}(n)$.

[2] We briefly explain the connection between ordered set-multilinear ABPs and non-commutative computation in Section 1.4.

Next, we formally state the aforementioned hardness escalation result of [5]. In words, in order to show $\mathsf{VBP} \neq \mathsf{VP}$, it suffices to show lower bounds for any $\sum$smABP computing a polynomial P whose degree is at most *about logarithmic* in the number of variables. Towards this goal, our main result above (Theorem 1.4) shows a super-polynomial lower bound for any $\sum$smABP computing an explicit set-multilinear polynomial, whose degree is *barely super-logarithmic* in the number of variables. In this sense, it approaches the resolution of Valiant's conjecture.

▶ **Theorem 1.5** (Hardness Escalation of [5]). *Let n, d be growing parameters with $d = O(\log n/ \log\log n)$. Let $P_{n,d}$ be a $\Theta(dn)$-variate degree d set-multilinear polynomial in VP (respectively, VNP). If $P_{n,d}$ cannot be computed by a $\sum$smABP of total-width $\mathrm{poly}(n)$, then $\mathsf{VBP} \neq \mathsf{VP}$ (respectively, $\mathsf{VBP} \neq \mathsf{VNP}$).*

Next, we also give an explicit set-multilinear polynomial (with polynomially-large degree) such that any $\sum$smABP computing it must require *exponential* total-width. This strongly answers a question left open in both [3] and [5].

▶ **Theorem 1.6** (Exponential Lower Bounds for $\sum$smABP). *There is a set-multilinear polynomial $F_{n,n}$ in VP, in $\Theta(n^2)$ variables and of degree $\Theta(n)$, such that any $\sum$smABP computing $F_{n,n}$ requires total-width $\exp(\Omega(n^{1/3}))$.*

Theorem 1.6 and Theorem 1.4 are also true when $F_{n,d}$ (as defined in Section 3.3) is replaced by the appropriate Nisan-Wigderson polynomial $NW_{n,d}$ (as defined in Section 3.2), which is known to be in VNP. In fact, we first indeed established them for the Nisan-Wigderson polynomial, and then used some of the ideas presented in a recent work by Kush and Saraf ([26]) to make the hard polynomial lie in VP.[3]

With additional effort, and building upon the machinery[4] of [26] (which, in turn, uses the techniques developed in [10]), we can almost recover the same lower bounds as in Theorem 1.6 and Theorem 1.4 for a set-multilinear polynomial even in VBP. We preferred to first state Theorem 1.6 and Theorem 1.4 in the manner above because (i) the proof is less intricate and in fact, even serves as a prelude to the proof of the latter, and (ii) to draw a direct comparison and contrast with the hardness escalation statement (Theorem 1.5). We now state these results for when the hard polynomial is the VBP polynomial and then describe two intriguing consequences.

▶ **Theorem 1.6'.** *There is a fixed constant $\delta \geq 1/100$ and a set-multilinear polynomial $G_{n,n}$ in VBP, in $\Theta(n^2)$ variables and of degree $\Theta(n)$, such that any $\sum$smABP computing $G_{n,n}$ requires total-width $\exp(\Omega(n^{\delta}))$.*

▶ **Theorem 1.4'.** *Let $d \leq n$ be growing parameters satisfying $d = \omega(\log n)$. There is a $\Theta(dn)$-variate, degree $\Theta(d)$ set-multilinear polynomial $G_{n,d}$ in VBP such that $G_{n,d}$ cannot be computed by a $\sum$smABP of total-width $\mathrm{poly}(n)$.*

The first intriguing consequence of proving the statements above is that we are able to show that the ABP set-multilinearization process given in [5] is nearly tight, as $G_{n,d}$ is known to have a small set-multilinear branching program and yet, any $\sum$smABP computing it must have large total-width. To make this point effectively, we first state the following key ingredient in the proof of Theorem 1.5, and subsequently state our tightness result.

---

[3] We also acknowledge that an exponential lower bound – with weaker quantitative parameters – for the related model of multilinear ROABPs was obtained in [31]. For a comparison of this model with the $\sum$smABP model, see Sections 1.4 and 1.5.

[4] This is explained in more detail in Section 1.6.

▶ **Lemma 1.7** (ABP Set-Multilinearization in [5]). *Let $P_{n,d}$ be a polynomial of degree d that is set-multilinear with respect to the partition $X = (X_1, \ldots, X_d)$ where $|X_i| \leq n$ for all $i \in [d]$. If $P_{n,d}$ can be computed by an ABP of size s, then it can also be computed by a $\sum$smABP of max-width s and total-width $2^{O(d \log d)} s$.*

▶ **Theorem 1.8** (Near-Tightness of ABP Set-Multilinearization). *For large enough integers $\omega(\log n) = d \leq n$, there is a polynomial $G_{n,d}(X)$ which is set-multilinear over the variable partition $X = (X_1, \ldots, X_d)$ with each $|X_i| \leq n$, and such that:*
- *it has a branching program of size $\mathrm{poly}(n)$,*
- *but any $\sum$smABP of max-width $\mathrm{poly}(n)$ computing $G_{n,d}$ requires total-width $2^{\Omega(d)}$.*

The second intriguing consequence is the fact that Theorem 1.8 can also be viewed as an exponential separation between the model of (general) small-width set-multilinear branching programs and the model of sums of small-width ordered set-multilinear branching programs. Moreover, we can improve this bound much further in the case of a single ordered set-multilinear branching program. More precisely, in Theorem 1.9 below, we answer a question posed in [26] about the relative strength of an unordered and (a single) ordered set-multilinear branching program, by obtaining a *near-optimal* separation. A priori, as is shown in [26] and as mentioned earlier in the introduction, if these two models coincided (i.e., if a general set-multilinear ABP could be simulated by a small and ordered one), then it would have led to super-polynomial lower bounds for general algebraic formulas.

▶ **Theorem 1.9** (Near-Optimal Separation between Ordered and Unordered smABPs). *There is a polynomial $G_{n,d}(X)$ which is set-multilinear over the variable partition $X = (X_1, \ldots, X_d)$ with each $|X_i| \leq n$, and such that:*
- *it has a set-multilinear branching program of size $\mathrm{poly}(n, d)$,*
- *but any ordered set-multilinear branching program computing $G_{n,d}$ requires width $n^{\Omega(d)}$.*

Note that $G_{n,d}$ has at most n^d monomials and so, it trivially has an ordered set-multilinear ABP of width n^d. Therefore, the lower bound above is essentially optimal.

1.4 The ROABP Perspective

One can also view all of our results described in Section 1.3 through the lens of another well-studied model in algebraic complexity theory, namely *read-once oblivious algebraic branching programs* (ROABPs).

▶ **Definition 1.10** (ROABP). *For integers n, d and a permutation $\sigma \in S_n$, an ABP over the variables $x_1, \ldots, x_n$ is said be a read-once oblivious algebraic branching program (ROABP) in the order σ of individual degree d if for each $\ell \in [n]$, all edges from layer $\ell - 1$ to ℓ are labelled by univariate polynomials in $x_{\sigma(i)}$ of degree at most d.*

ROABPs were first introduced in this form by Forbes and Shpilka in [12], where it is also noted that proving lower bounds against ordered set-multilinear ABPs (as in Definition 1.1) is equivalent to proving lower bounds against ROABPs as well as non-commutative ABPs.

Suppose $f \in [X_1, \ldots, X_d]$ is a set-multilinear polynomial with respect to $X_1 \sqcup \cdots \sqcup X_d$ with $X_i = \{x_{i,1}, \ldots, x_{i,n}\}$. Then we can define an associated polynomial $g_f \in [x_1, \ldots, x_d]$ as follows.

$$g_f(x_1, \ldots x_d) = \sum_{\mathbf{e} \in [n]^d} \prod_{i=1}^{n} x_i^{e_i} \cdot \text{ coefficient of } x_{i,e_i}.$$

Now let us assume that g_f can be computed by an ROABP of size s that is ordered with respect to $\sigma \in S_n$. Then a set-multilinear ABP ordered with respect to σ can be constructed using it, by simply replacing $x_i^{e_i}$ by x_{i,e_i} and erasing any degree zero components on each edge. It is easy to check that this computes f and we can use the lower bound against ordered set-multilinear ABPs for f to prove a lower bound against ROABPs for g_f. Conversely, given $g \in [x_1, \ldots, x_n]$, we can define $f_g \in [X_1, \ldots, X_n]$ with $X_i = \{x_{i,0}, x_{i,1}, \ldots, x_{i,d}\}$ by by replacing $x_i^{e_i}$ with x_{i,e_i}. We could then use an ordered set-multilinear ABP computing f_g to construct an ROABP (in the same order) computing g by using the inverse transformation, thereby proving that lower bounds against ROABPs imply lower bounds against ordered set-multilinear ABPs. Furthermore, the computation that an ROABP (or an ordered set-multilinear ABP) performs can be seen to be *non-commutative*. This is because the variables (or linear forms) along a path get multiplied in the *same* order σ as that of the ROABP (or ordered set-multilinear ABP).

As a consequence, exponential lower bounds follow for a single ROABP from the work of Nisan ([29]), and also from later works ([18, 21]). Using the transformation described above, our lower bounds (Theorem 1.6 and Theorem 1.6') can also be viewed as exponential lower bounds for the model of sum of ROABPs. The work of Ramya & Rao [31] also prove (weaker) exponential lower bounds against this model for a multilinear polynomial computable by multilinear circuits. In a follow-up work, Ghoshal & Rao [13] prove an exponential lower bound against sums of ROABPs with the additional restriction that the summand ROABPs have pollynomially-bounded width for a mulilinear polynomial computable by multlinear ABPs. On the other hand, the works of Arvind & Raja ([3]) and Bhargav, Dwivedi & Saxena ([5] provide lower bounds in certain restricted versions of this model. Along with these, the work of Anderson, Forbes, Saptharishi, Shpilka, and Volk ([2]) also implies an exponential lower bound for a restricted version (for the sum of k ROABPs when $k = o(\log n)$).

Finally, we note that ROABPs have been studied extensively in the context of another central problem in algebraic complexity theory: that of polynomial identity testing (PIT). The PIT question for a general algebraic model $\mathcal{M}$ is the following: Given access to an n-variate polynomial f of degree at most d that can be computed in the model $\mathcal{M}$ of (an appropriate measure of) complexity at most s, determine whether $f \equiv 0$ in $\mathrm{poly}(n, d, s)$ time. When one is given access to the model computing f explicitly, this flavour of PIT is called white-box PIT, and when one is merely provided query access to f, it is called black-box PIT.

The solution to the PIT problem for ROABPs in the white-box setting follows from a result by Raz and Shpilka ([34] – where it is stated in the equivalent language of non-commutative computation). However, the corresponding problem in the black-box setting remains open to this date, with the best-known time bound in the black-box setting still being only $s^{O(\log s)}$ due to the work by Forbes and Shpilka ([12]), who additionally assumed that the ordering of the ROABP is known. This was matched later by Agrawal Gurjar, Korwar, and Saxena ([1]) in the unknown order setting, improving upon the work of Forbes, Saptharishi and Shpilka ([11]). Guo and Gurjar improved the result further by improving the dependence on the width [14]. Additionally, there have been various improvements to this result in restricted settings ([15, 17, 6]) and some other works that study PIT for a small sum of ROABPs ([16, 7, 14]). When the number of summands is super-constant, the question of even white-box PIT remains wide open.

1.5 Related Work

In this subsection, we discuss two closely related papers, namely those of Ramya & Rao [31] and Ghoshal & Rao [13][5], which study the model of sum of ROABPs in the multilinear setting. In [31, Theorem 1], the authors show that there exists an explicit multilinear polynomial (computable by a small multilinear circuit) such that any sum of ROABPs computing it has exponential size. In [13, Theorem 2], the authors show a similar lower bound for an explicit multilinear polynomial (computable by a small multilinear ABP) – albeit, in the restricted setting where the summand ROABPs have polynomially-bounded width.

Using the transformation described in Section 1.4, one can then view these lower bounds as ones against the $\sum$smABP model in the special case that each bucket in the variable partition has size 2. (To see how a multilinear polynomial say over the variable set $x_1, \ldots, x_d$ can be set-multilinearized trivially, here is a sketch: for each variable x_i, have a variable set X_i comprising of two fresh variables $x_{i,0}$ and $x_{i,1}$ in the new set-multilinear polynomial; here, the latter is to signify the "presence" of x_i in any monomial of the original multilinear polynomial, whereas the former is to signify its "absence".) Additionally, it is not hard to see that the set-multilinearized version of the hard polynomials (in the manner just described) used in [31, Theorem 1] and [13, Theorem 2] are efficiently computable by small set-multilinear circuits and set-multilinear ABPs respectively. We note, however, that even so, our result in the high-degree setting where the hard polynomial is in VP (Theorem 1.6) is quantitatively better than [31, Theorem 1]. Additionally, our result in the high-degree setting where the hard polynomial is in VBP (Theorem 1.6') is *both* quantitatively as well as qualitatively better than [13, Theorem 2] – the latter since we do not assume any bound on the width of the individual summand ordered set-multilinear ABPs. More crucially, our techniques enable us to prove super-polynomial bounds even when the degree is vastly smaller than the number of variables – in particular, when d is as low as $\omega(\log n)$ (Theorem 1.4 and Theorem 1.4') – which is the more interesting regime of parameters due to the work of [5].

Ramya and Rao [31] also study another model, which they call *sum of α-set-multilinear ABPs*. They define α-set-multilinear ABPs to be ABPs with N^α layers, where N is the number of variables in the polynomial being computed. Any edge between layer $\ell - 1$ and ℓ in an α-set-multilinear ABP is labelled by an arbitrary multilinear polynomial over X_ℓ, where $X = X_1 \sqcup \cdots \sqcup X_{N^\alpha}$ is a partition of the variable set. Then, for $\alpha \geq 1/10$, they establish exponential lower bounds against sum of α-set-multilinear ABPs for a polynomial that is multilinear, but which is *not* set-multilinear under the variable partition that the model respects. Hence, even though this model is more general than ordered set-multilinear ABPs, this result [31, Theorem 3] is also not comparable with ours as our hard polynomial is set-multilinear. Again, more crucially, the result [31, Theorem 3] does not handle the "low-degree" regime – a setting in which our techniques allow us to prove lower bounds.

1.6 Proof Overview

The organization of this subsection is as follows: we first describe the basics of the partial derivative method and summarize its typical application in proving lower bounds against a generic set-multilinear model of computation. Next, we briefly describe Nisan's original partial derivative method from [29] to prove lower bounds specifically against a single ordered set-multilinear branching program. We then describe an alternative approach that yields a

[5] We thank Ben Lee Volk and Utsab Ghosal for pointing out these papers to us after the release of an initial pre-print of this article, which erroneously claimed that it was the first to show super-polynomial lower bounds in the sum of ROABPs model.

slightly weaker bound for the same model, but nevertheless is versatile enough that we can generalize it considerably more in order to prove Theorem 1.6 and Theorem 1.4. Finally, we describe the additional ideas needed in order to situate the hard polynomial in these theorems in VBP and in the process, establish the tightness result for ABP set-multilinearization (Theorem 1.8).

Partial Derivative Measure Basics

The high-level idea is to work with a measure that we show to be "small" for all polynomials computed by a specified model of computation – the model against which we wish to prove lower bounds. If we can also show that there is a "hard" polynomial for which the measure is in fact "large", then it follows that this polynomial cannot be computed by the specified model. These *partial derivative measures*, after the initial work ([29]) by Nisan, were further developed by Nisan and Wigderson in [30], who used them to prove some constant-depth set-multilinear formula lower bounds. Since then, variations of these measures have also been used to prove various other stronger set-multilinear formula lower bounds (e.g., [27, 39, 28, 4, 25, 26]).

Given a variable partition $(X_1, \ldots, X_d)$, the idea is to label each set of variables X_i as "$+1$" or "-1" according to *some* rule (called a "word") $w \in \{-1, 1\}^d$. Let $\mathcal{P}_w$ and $\mathcal{N}_w$ denote the set of positive and negative indices (or coordinates) respectively, and let $\mathcal{M}_w^{\mathcal{P}}$ and $\mathcal{M}_w^{\mathcal{N}}$ denote the sets of all set-multilinear monomials over $\mathcal{P}_w$ and $\mathcal{N}_w$ respectively. For a polynomial f that is set-multilinear over the given variable partition $(X_1, \ldots, X_d)$, the measure then is simply the rank of the "partial derivative matrix" $\mathcal{M}_w(f)$, whose rows are indexed by the elements of $\mathcal{M}_w^{\mathcal{P}}$ and columns indexed by $\mathcal{M}_w^{\mathcal{N}}$, and the entry of this matrix corresponding to a row m_1 and a column m_2 is the coefficient of the monomial $m_1 \cdot m_2$ in f.

For a subset $S \subseteq [d]$, let w_S denote the sum of those coordinates of w that lie in S. In other words, $|w_S|$ measures the amount of "bias" that the rule w exhibits when restricted to the S coordinates. Note that the rank of $M_w(f)$ can never exceed $n^{(d-|w_{[d]}|)/2}$. Furthermore, we have that the rank measure is *multiplicative*: if f and g are polynomials that are set-multilinear over *disjoint* subsets of the global partition $(X_1, \ldots, X_d)$, then the rank of $\mathcal{M}_w(f \cdot g)$ is the product of the ranks of $\mathcal{M}_w(f)$ and $\mathcal{M}_w(g)$. These two observations, combined with the sub-additivity of rank, provide a recipe for showing lower bounds against any given set-multilinear model of computation: the overall idea is to carefully split up the original model into smaller, multiplicatively disjoint parts and then argue the existence of a rule for which enough of these parts exhibit high bias. This process allows us to prove that the measure is small for the model of computation. Therefore, one can conclude that any explicit polynomial for which the measure is provably high – which needs to established separately – can not be computed by this model. It is known ([25, 26]) that there is a set-multilinear polynomial $NW_{n,d}$ in VNP (see Section 3.2) as well as a set-multilinear polynomial $F_{n,d}$ in VP (see Section 3.3) for which the matrices $\mathcal{M}_w(NW_{n,d}), \mathcal{M}_w(F_{n,d})$, have full-rank, whenever $|\mathcal{P}_w| = |\mathcal{N}_w|$.

Nisan's original lower bound

Let us first summarize how Nisan's original partial derivative method from [29], as alluded to in Section 1.2, can be applied in this context to obtain lower bounds against the size of a single ordered set-multilinear ABP (ordered smABP) computing the aforementioned "full-rank" polynomials. Given any set-multilinear branching program A ordered with respect to some permutation $\sigma \in S_d$ computing $F_{n,d}$, the idea is to pick a word w such that the $+1$ labels in w precisely correspond to the "left half" of the ordering σ, and the -1 labels correspond to the "right half". One can then observe that the rank of $\mathcal{M}_w(F_{n,d}) = \mathcal{M}_w(A)$ serves as a lower bound on the number of nodes s in the middle layer of the ABP, yielding a near-optimal $n^{\Omega(d)}$ lower bound: this is because the matrix $\mathcal{M}_w(A)$ is easily seen to be the product of an $n^{d/2} \times s$ and an $s \times n^{d/2}$ matrix.

We now sketch an alternate proof: rather than constructing a word dependent on the ordering of variable sets X_i in the ordered smABP A as above, choose a uniformly random[6] word w from $\{-1,1\}^d$. We demonstrate that, with positive probability, the rank of $\mathcal{M}_w(A)$ is bounded by $s \cdot n^{d/2-\Omega(\sqrt{d})}$, where s is the width of the middle layer in A: Standard anti-concentration bounds imply that, with at least constant probability, the bias in the left and right halves of A is $\Omega(\sqrt{d})$. Since A can be expressed as a sum of s polynomials $f_i \cdot g_i$ for $i \in [s]$, where each f_i and g_i are ordered smABPs with respect to disjoint subsets of the global partition, we encounter a loss of a factor of $n^{\Omega(\sqrt{d})}$ in the rank of the product polynomial $\mathcal{M}_w(f_i \cdot g_i)$ due to the bias of w. This, combined with the sub-additivity of rank, shows the desired bound of $s \cdot n^{d/2-\Omega(\sqrt{d})}$ on the rank of $\mathcal{M}_w(A)$. Finally, we exploit the full-rank property of $F_{n,d}$ with respect to such words to establish a lower bound of $n^{\Omega(\sqrt{d})}$ on the width s of a single ordered smABP computing $F_{n,d}$. Notably, this bound is indeed slightly worse than what one can obtain by manually defining a rule w deterministically, which ensures a *maximal* bias of $d/2$ in each half of A as described in the paragraph above.

Generalization of the alternative argument

The alternative argument described above yields an exponential lower bound even for a *sum* of ordered smABPs, assuming the number of summands is small. Consider a $\sum$smABP of the form $\sum_{i=1}^{t} A_i$, of max-width s, computing $F_{n,d}$. For each summand A_i, the analysis above provides an upper bound of $s \cdot n^{d/2-\Omega(\sqrt{d})}$ on the rank of $\mathcal{M}_w(A_i)$ with constant probability. If the number of summands t is a small enough constant, the union bound ensures the existence of a word w such that the rank of $\mathcal{M}_w(\sum A_i)$ is at most $t \cdot s \cdot n^{d/2-\Omega(\sqrt{d})}$. Thus[7], we obtain an exponential lower bound on $t \cdot s$ since this $\sum$smABP computes a full-rank polynomial. However, because of the use of the union bound in this manner, this method faces an inherent limitation – it is unable to handle more than a very small number of summands, even if we lower the bias demand from each half (e.g., from $\Omega(\sqrt{d})$ to $\Omega(\sqrt[4]{d})$ or a smaller polynomial in d). In fact, one can construct a sum of d ordered smABPs (by starting with a single smABP ordered arbitrarily and considering the d cyclic shifts of this ordering) such that any unbiased word w (i.e., $w_{[d]} = 0$) has the property that for at least one of the summands, the left and right halves will have no bias! Evidently then, in order to prove lower bounds against an *unrestricted* number of summands, we need another method to analyze the rank of the summands. Nonetheless, a conceptual takeaway from the exercise above is that selecting a rule w that is oblivious to the orderings of individual summands (and in particular, a *random* rule) still lets us derive strong lower bounds for the sum of *multiple* ordered smABPs.

Suppose instead of slicing an ordered smABP A down the middle, we slice it into three roughly equal pieces. Then, it is possible to write the polynomial computed by A as a sum over s^2 terms, each of the form $f_i \cdot g_i \cdot h_i$ where for each i, each of f_i, g_i, h_i depends on $d/3$

6 We also need to suitably condition on the event that the word w is symmetric (i.e., $|\mathcal{P}_w| = |\mathcal{N}_w|$) in order to use the full-rank property of the hard polynomial – the probability of this event is $\Theta(\frac{1}{\sqrt{d}})$. For ease of exposition, we omit the technical details in this sketch.

7 See footnote 6.

disjoint variable sets of the global partition. We can then perform a similar analysis as above to show enough bias across these 3 pieces, thereby obtaining a rank deficit. More precisely, we can conclude that for a single ordered smABP A, again with a constant probability, the rank of $\mathcal{M}_w(A)$ is at most $s^2 \cdot n^{d/2-\Omega(\sqrt{d})}$. When we slice the ABP into 3 pieces in this way, it is not immediately clear where the gain is. In fact, for a single ordered smABP, this method actually gives a worse lower bound on s due to the presence of the factor of s^2. Where we gain is in the magnitude of the *probability* with which we can guarantee that a single ordered smABP has a rank deficit – we will now describe how this observation allows us to take a union bound over many more summands.

In order to illustrate this trade-off more clearly, we will partition the ordered smABP A into many more pieces. Suppose we slice it into $q \approx \sqrt{d}$ pieces, each of size roughly $r = d/q \approx \sqrt{d}$ (this is just one setting of parameters; q and r are suitably optimized in the final proof). Thus, the polynomial that A computes can be written as a sum of at most s^{q-1} terms, where each term is a product of q polynomials – each set-multilinear over a disjoint subset of the global partition, where each piece has size r. When a word w is chosen randomly, each such piece again exhibits a bias of about $\Omega(\sqrt{r})$ with constant probability. The crucial observation then is that by known concentration bounds, it can be shown that with probability *exponentially* close to 1, the sum of the biases across all the q pieces is $\Omega(q\sqrt{r}) = \mathrm{poly}(d)$. For a single ordered smABP A, this shows that the rank of $\mathcal{M}_w(A)$ is at most $s^q \cdot n^{-\Omega(q\sqrt{r})}$, which is still enough to show an exponential lower bound on s, even though it is worse than what we obtained by slicing into fewer pieces.

The key advantage in implementing this analysis is that it provides a way to argue that for a random word w, $\mathcal{M}_w(A)$ has low rank for a single ordered smABP A – with probability *exponentially* close to 1. In particular, this allows us to union bound over exponentially many ordered smABPs and show that even if we have an $\sum$smABP computing $F_{n,d}$ of exponential support size, with high probability, each summand will have a rank deficit. Then, again using the sub-additivity of rank, we can conclude that the sum has a rank deficit as well.

This method of analyzing the rank of an ordered smABP by partitioning it into *numerous* pieces and tactfully using concentration bounds is novel, and conceptually the most essential aspect of the proof. As we demonstrated above, this method of analysis indeed gives a worse bound for a single smABP. However, while mildly sacrificing what we can prove about the rank of a single ordered smABP, we are able to leverage it to still prove something meaningful about the rank of a *sum* with a much larger number of summands.

Our partial derivative measure draws inspiration from previously known lower bounds in the context of multilinear and set-multilinear *formulas* ([32, 25]). One noteworthy distinction lies in the analysis of the measure: whereas the partitioning is present intrinsically in those formula settings, in our setting of ABPs, we deliberately introduce the partitioning at the expense of a notable increase in the number of summands or the total-width (and therefore, in the number of events we union bound over). The substantial advantage gained in utilizing this partitioning for rank analysis justifies the tolerable increase in the total-width.

Tightness of ABP set-multilinearization

In order to make the hard polynomial in Theorems 1.6 and 1.4 lie in VBP, one might wonder if we can get away with using the same rank measure (i.e., rank of the matrix $M_w(\cdot)$ for a uniformly random word $w \in \{-1,1\}^d$) that was used in the analysis above for the VP polynomial $F_{n,d}$. However, as far as we know, full-rank polynomials (in the sense described above) may also require super-polynomial sized set-multilinear ABPs. Thus, in order to prove a separation between (general) set-multilinear ABPs and (sums of) ordered set-multilinear ABPs, we seek a property that is weaker than being full-rank and yet is still useful enough for proving lower bounds against our model. For this, we rely upon the *arc-partition* framework that is developed in [26] in order to prove near-optimal set-multilinear formula lower bounds (building upon the initial ingenious construction given in [10] for the multilinear context), tailor the framework to our $\sum$smABP model, and use a more delicate concentration bound analysis in order to prove our results.

An *arc-partition* is a special kind of symmetric word w from $\{-1,1\}^d$: we will now describe a distribution over $\{-1,1\}^d$; the words that will have positive probability of being obtained in this distribution will be called arc-partitions. The distribution is defined according to the following (iterative) sampling algorithm. Position the d variable sets on a cycle with d nodes so that there is an edge between i and $i+1$ modulo d. Start with the arc $[L_1, R_1] = \{1,2\}$ (an arc is a connected path on the cycle). At step $t > 1$ of the process, maintain a partition of the arc $[L_t, R_t]$. "Grow" this partition by first picking a pair uniformly at random out of the three possible pairs $\{L_t-2, L_t-1\}, \{L_t-1, R_t+1\}, \{R_t+1, R_t+2\}$, and then choosing a labelling (or partition) Π on this pair i.e., assigning one of them "$+1$" and the other "-1" uniformly at random. After $d/2$ steps, we have chosen a partition (i.e., a word w from $\{-1,1\}^d$) of the d variable sets into two disjoint, equal-size sets of variables $\mathcal{P}$ and $\mathcal{N}$. It is known from [26] that there exist set-multilinear polynomials $G_{n,d}$ (as defined in Section 3.4) that are *arc-full-rank* i.e., $\mathcal{M}_w(G_{n,d})$ is full-rank for every arc-partition w. Analogous to the proofs of Theorems 1.6 and 1.4, we establish our $\sum$smABP lower bounds by showing that with high probability, every $\sum$smABP has an appropriately large rank deficit with respect to the arc-partition distribution. However, as we now briefly explain, this analysis turns out to be significantly more intricate.

Similar to the analysis as in the VP case, we partition an ordered smABP A into q pieces of size r each, and write the polynomial that it computes as a sum of at most s^q terms. Again, the task is to show that an arc-partition w exhibits a large total bias across the q pieces: more precisely, we show that if the pieces are labelled as $S_1, \ldots, S_q$, then with probability *exponentially* close to 1, the sum $\sum_{i=1}^{q} |w_{S_i}|$ (i.e., the total bias of w across these pieces) is $\Omega(qr^{\varepsilon})$, which is polynomially large in d for an appropriate setting of q, r. This then yields the desired rank deficit similar to the VP analysis (albeit with mildly worse parameters).

The bias lower bound is established in the following sequence of steps:

- View the partition $(S_1, \ldots, S_q)$ of $[d]$ as a fixed "coloring" of the latter. We say that a *pair* – as sampled in the construction of an arc-partition described above – "violates" a color S if exactly one of the elements of the pair is colored by the set S. Then, we show that with probability exponentially close to 1, "many" colors must have "many" violations: more precisely, that at least a constant fraction of the colors (i.e., $\Omega(q)$ many) have at least $r^{2\varepsilon}$ many violations each (for some small constant $\varepsilon > 0$). Such a "many violations" lemma is also established in [26] in the context of proving set-multilinear formula lower bounds. We show that this lemma, in fact, holds for a much wider range of parameters than was previously known; this extension is indeed necessary for our use. The proof of this strengthened many violations lemma is deferred to the appendix.
- We then use the strengthened many violations lemma to argue that even though w is not chosen uniformly at random and as such, its coordinates are not truly independent, it possesses "enough" inherent independence that a similar concentration bound as in the VP analysis is applicable. More precisely, we show that with high probability, there is an ordering of a set of $\Omega(q)$ colors such that each such color has at least $r^{2\varepsilon}$ violations and a more nuanced application of standard concentration bounds shows that w exhibits a total bias of at least $\Omega(qr^{\varepsilon})$.

2 Relative Rank and its Properties

We first describe the notation that we need to define the measures that we use to prove our results described in Section 1.3. Instead of directly working with the rank of the partial derivative matrix, we work with the following normalized form.

▶ **Definition 2.1.** *Let $w = (w_1, w_2, \ldots, w_d)$ be a tuple (or word) of non-zero real numbers. For a subset $S \subseteq [t]$, we shall refer to the sum $\sum_{i\in S} w_i$ by w_S, and by $w|_S$, we will refer to the tuple obtained by considering only the elements of w that are indexed by S. Given a word $w = (w_1, \ldots, w_d)$, we denote by $\overline{X}(w)$ a tuple of d sets of variables $(X(w_1), \ldots, X(w_d))$ where $|X(w_i)| = 2^{|w_i|}$.*[8] *We denote by $\mathbb{F}_{\mathrm{sm}}[\mathcal{T}]$ the set of set-multilinear polynomials over the tuple of sets of variables $\mathcal{T}$.*

▶ **Definition 2.2** (Relative Rank Measure of [27])**.** *Let $\overline{X} = (X_1, \ldots, X_d)$ be a tuple of sets of variables such that $|X_i| = n_i$ and let $f \in \mathbb{F}_{\mathrm{sm}}[\overline{X}]$. Let $w = (w_1, w_2, \ldots, w_d)$ be a tuple (or word) of non-zero real numbers such that $2^{|w_i|} = n_i$ for all $i \in [d]$. Corresponding to a word w, define $\mathcal{P}_w := \{i \mid w_i > 0\}$ and $\mathcal{N}_w := \{i \mid w_i < 0\}$. Let $\mathcal{M}_w^{\mathcal{P}}$ be the set of all set-multilinear monomials over the subset of the variable sets $X_1, X_2, \ldots, X_d$ precisely indexed by $\mathcal{P}_w$, and similarly let $\mathcal{M}_w^{\mathcal{N}}$ be the set of all set-multilinear monomials over these variable sets indexed by $\mathcal{N}_w$.*

Define the 'partial derivative matrix' matrix $\mathcal{M}_w(f)$ whose rows are indexed by the elements of $\mathcal{M}_w^{\mathcal{P}}$ and columns indexed by the elements of $\mathcal{M}_w^{\mathcal{N}}$ as follows: the entry of this matrix corresponding to a row m_1 and a column m_2 is the coefficient of the monomial $m_1 \cdot m_2$ in f. We define

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

The following is a simple result that establishes various useful properties of the relative rank measure.

▷ Claim 2.3 ([27]).

1. (Imbalance) Say $f \in \mathbb{F}_{\mathrm{sm}}[\overline{X}(w)]$. Then, $\mathrm{relrk}_w(f) \le 2^{-|w_{[d]}|/2}$.
2. (Sub-additivity) If $f, g \in \mathbb{F}_{\mathrm{sm}}[\overline{X}(w)]$, then $\mathrm{relrk}_w(f+g) \le \mathrm{relrk}_w(f) + \mathrm{relrk}_w(g)$.
3. (Multiplicativity) Say $f = f_1 f_2 \cdots f_t$ and assume that for each $i \in [t]$, $f_i \in \mathbb{F}_{\mathrm{sm}}[\overline{X}(w|_{S_i})]$, where $(S_1, \ldots, S_t)$ is a partition of $[d]$. Then

$$\mathrm{relrk}_w(f) = \prod_{i\in[t]} \mathrm{relrk}_{w|_{S_i}}(f_i).$$

3 The Hard Polynomial

We now describe the different hard polynomials we use for our results.

3.1 Inner Product Gadget

The following observation is used crucially to construct the hard polynomials in VP as well as VBP.

---

[8] In particular, $2^{|w_i|} \in \mathbb{N}$.

▶ **Observation 3.1** ([26]). *Let $n = 2^k$ and $X_1 = \{x_{1,1}, \dots, x_{1,n}\}$ and $X_2 = \{x_{2,1}, \dots, x_{2,n}\}$ be two disjoint sets of variables. Then, for any* symmetric *word $w \in \{k, -k\}^2$ (i.e., where $w_1 + w_2 = 0$) and for the inner product "gadget" $f = X_1 \cdot X_2 = \sum_{i=1}^{n} x_{1,i}x_{2,i}$,* $\mathrm{relrk}_w(f) = 1$ *i.e., $\mathcal{M}_w(f)$ is full-rank.*

3.2 A Hard Set-multilinear Polynomial in VNP

As is done in previous lower bounds using the NW polynomials (for example, see [22]), we will identify the set of the first n integers as elements of $_n$ via an arbitrary correspondence $\phi : [n] \to_n$. If $f(z) \in_n [z]$ is a univariate polynomial, then we abuse notation to let $f(i)$ denote the evaluation of f at the i-th field element via the above correspondence i.e., $f(i) \coloneqq \phi^{-1}(f(\phi(i)))$. To simplify the exposition, in the following definition, we will omit the correspondence ϕ and identify a variable $x_{i,j}$ by the point $(\phi(i), \phi(j)) \in_n \times_n$.

▶ **Definition 3.2** (Nisan-Wigderson Polynomials). *For a prime power n, let $_n$ be a field of size n. For an integer $d \le n$ and the set X of nd variables $\{x_{i,j} : i \in [n], j \in [d]\}$, we define the degree d homogeneous polynomial $NW_{n,d}$ over any field as*

$$NW_{n,d}(X) = \sum_{\substack{f(z) \in_n [z] \\ \deg(f) < d/2}} \prod_{j \in [d]} x_{f(j),j}.$$

▷ Claim 3.3 ([25]). For an integer $n = 2^k$ and $d \le n$, let $w \in \{k, -k\}^d$ with $w_{[d]} = 0$. Then $\mathrm{relrk}_w(NW_{n,d}) = 1$ i.e., $\mathcal{M}_w(NW_{n,d})$ has full rank.

Proof. Fix $n = 2^k$ and d, so that we can also write NW for $NW_{n,d}$, and let $n' = d/2$. The condition on w implies that $|\mathcal{P}_w| = |\mathcal{N}_w| = n'$. Observe that $\mathcal{M}_w(NW)$ is a square matrix of dimension $|\mathcal{M}_w^{\mathcal{P}}| = |\mathcal{M}_w^{\mathcal{N}}| = n^{n'}$. Consider a row of $\mathcal{M}_w(NW)$ indexed by a monomial $m_1 = x_{i_1,j_1} \cdots x_{i_{n'},j_{n'}} \in \mathcal{M}_w^{\mathcal{P}}$. m_1 can be thought of as a map from $S = \{j_1, \dots, j_{n'}\}$ to $_n$ which sends j_ℓ to i_ℓ for each $\ell \in [n']$. Next, by interpolating the pairs $(j_1, i_1), \dots, (j_{n'}, i_{n'})$, we know that there exists a unique polynomial $f(z) \in_n (z)$ of degree $< n'$ for which $f(j_\ell) = i_\ell$ for each $\ell \in [n']$. As a consequence, there is a unique "extension" of the monomial $x_{i_1,j_1} \cdots x_{i_{n'},j_{n'}}$ that appears as a term in NW, which is precisely $m_1 \cdot \prod_{j \in \mathcal{N}_w} x_{f(j),j}$. Therefore, all but one of the entries in the row corresponding to m_1 must be zero, and the remaining entry must be 1. Applying the same argument to the columns of $\mathcal{M}_w(NW)$, we deduce that $\mathcal{M}_w(NW)$ is a permutation matrix, and so has full rank. ◁

3.3 A Hard Set-multilinear Polynomial in VP

Let d be an even integer and let $X = (X_1, \dots, X_d)$ be a collection of sets of variables where each $|X_i| = n$, and similarly, let $Y = (Y_1, \dots, Y_d)$ be a distinct collection of sets of variables where each $|Y_i| = n$. We shall refer to the Y-variables as the *auxiliary* variables. For i and $j \in \{1, \dots, d\}$, let $X_i \cdot X_j$ denote the inner-product quadratic form $\sum_{k=1}^{n} x_{ik}x_{jk}$. Here, we shall assume that $X_i = \{x_{i,1}, \dots, x_{i,n}\}$ and $Y_i = \{y_{i,1}, \dots, y_{i,n}\}$.

For two integers $i \in \mathbb{N}$ and $j \in \mathbb{N}$, we denote $[i, j] = \{k \in \mathbb{N} : i \le k \text{ and } k \le j\}$ and call such a set an *interval*. For every interval $[i, j] \subseteq [d]$, we define a polynomial $f_{i,j}(X, Y) \in \mathbb{F}_{\mathrm{sm}}[X_i, \dots, X_j, Y_i, \dots, Y_j]$ as follows:

$$f_{i,j} = \begin{cases} y_{i,j}y_{j,i}(X_i \cdot X_j) & \text{if } j = i + 1 \\ 0 & \text{if } j - i \text{ is even} \\ y_{i,j}y_{j,i}(X_i \cdot X_j) \cdot f_{i+1,j-1} + \sum_{r=i+1}^{j-1} f_{i,r} f_{r+1,j} & \text{otherwise} \end{cases}$$

These $f_{i,j}$ in present form were defined in [26], but were in turn inspired from an earlier work of Raz and Yehudayoff ([35]) in the multilinear context. [26] shows that they have the following full-rank property that will be instrumental for us.

▶ **Lemma 3.4** ([26]). *Let $n = 2^k$ and $d \leq n$ be an even integer. Over any field of characteristic zero, the polynomial $F_{n,d} = f_{1,d} \in \mathbb{F}_{\mathrm{sm}}[X, Y]$ as defined above satisfies the following: For any $w \in \{-k, k\}^d$ with $w_{[d]} = 0$, $\mathcal{M}_w(F_{n,d})$ is full-rank when viewed as a matrix over the field (Y), the field of rational functions over the Y variables.*

3.4 A Hard Set-Multilinear Polynomial in VBP

3.4.1 Arc-partition Measure Description

This subsection is adapted from Section 2 of [10]. Let $n = 2^k$, $d \leq n$ be an even integer, and let $X = (X_1, X_2, \ldots, X_d)$ be a collection of disjoint sets of n variables each. An *arc-partition* will be a special kind of *symmetric* word $w \in \{-k, k\}^d$ (i.e., a one-to-one map Π from X to $\{-k, k\}^d$). For the purpose of this subsection, the reader can even choose to think of the alphabet of w as $\{-1, 1\}$ (i.e., one "positive" and one "negative" value) – we use $k, -k$ only to remain consistent with Definition 2.2.

Identify X with the set $\{1, 2, \ldots, d\}$ in the natural way. Consider the d-cycle graph, i.e., the graph with nodes $\{1, 2, \ldots, d\}$ and edges between i and $i + 1$ modulo d. For two nodes $i \neq j$ in the d-cycle, denote by $[i, j]$ the arc between i, j, that is, the set of nodes on the path $\{i, i + 1, \ldots, j - 1, j\}$ from i to j in d-cycle. First, define a distribution $\mathcal{D}_P$ on a family of pairings (a list of disjoint pairs of nodes in the cycle) as follows. A random pairing is constructed in $d/2$ steps. At the end of step $t \in [d/2]$, we shall have a pairing $(P_1, \ldots, P_t)$ of the arc $[L_t, R_t]$. The size of $[L_t, R_t]$ is always $2t$. The first pairing contains only $P_1 = \{L_1, R_1\}$ with $L_1 = 1$ and $R_1 = 2$. Given $(P_1, \ldots, P_t)$ and $[L_t, R_t]$, define the random pair $P_t + 1$ (independently of previous choices) by

$$P_{t+1} = \begin{cases} \{L_t - 2, L_t - 1\} & \text{with probability } 1/3 \\ \{L_t - 1, R_t + 1\} & \text{with probability } 1/3 \\ \{R_t + 1, R_t + 2\} & \text{with probability } 1/3 \end{cases}$$

Define

$$[L_{t+1}, R_{t+1}] = [L_t, R_t] \cup P_{t+1}.$$

So, L_{t+1} is either $L_t - 2$, $L_t - 1$ or L_t, each value is obtained with probability $1/3$, and similarly (but not independently) for R_{t+1}.

The final pairing is $P = (P_1, P_2, \ldots, P_{d/2})$. Denote by $P \sim \mathcal{D}_P$ a pairing distributed according to $\mathcal{D}_P$.

Once a pairing P has been obtained, a word $w \in \{-k, k\}^d$ is obtained by simply randomly assigning $+k$ and $-k$ to the indices of any pair P_i. More formally, for every $t \in [d/2]$, if $P_t = \{i_t, j_t\}$, let with probability $1/2$, independently of all other choices,

$$w_{i_t} = +k \text{ and } w_{j_t} = -k,$$

and with probability $1/2$,

$$w_{i_t} = -k \text{ and } w_{j_t} = +k.$$

Denote by $w \sim \mathcal{D}$ a word in $\{-1, 1\}^n$ that is sampled using this procedure. We call such a word an *arc-partition*. For a pair $P_t = \{i_t, j_t\}$, we refer to i_t and j_t as *partners*.

▶ **Definition 3.5** (Arc-full-rank). *We say that a polynomial f that is set-multilinear over $X = (X_1, \dots, X_d)$ is **arc-full-rank** if for every arc-partition $w \in \{-k, k\}^d$,* $\mathrm{relrk}_w(f) = 1$.

3.4.2 Construction of an Arc-full-rank Polynomial

Below, we describe a simple construction of a polynomial sized ABP that computes an arc-full-rank set-multilinear polynomial. The high-level idea is to construct an ABP in which every path between start-node and end-node corresponds to a specific execution of the random process which samples arc-partitions. Each node in the ABP corresponds to an arc $[L, R]$, which sends an edge to each of the nodes $[L-2, R], [L-1, R+1]$ and $[L, R+2]$. The edges have specially chosen labels that help guarantee full rank with respect to every arc-partition. For simplicity of presentation, we allow the edges of the program to be labeled by degree four set-multilinear polynomial polynomials over the corresponding subset of the variable partition. This assumption can be easily removed by replacing each edge with a polynomial-sized ABP computing the corresponding degree four polynomial.

Formally, the nodes of the program are even-size arcs in the d-cycle, d an even integer. The start-node of the program is the empty arc $\emptyset$ and the end-node is the whole cycle $[d]$ (both are "special" arcs). Let $X = (X_1, \dots, X_d)$ be a collection of sets of variables where each $|X_i| = n$, and similarly, let $Y = (Y_1, \dots, Y_d)$ be a distinct collection of sets of variables where each $|Y_i| = n$ (we shall refer to the Y-variables as *auxiliary* variables). For i and j in $\{1, \dots, d\}$, let $X_i \cdot X_j$ denote the inner-product quadratic form $\sum_{k=1}^{n} x_{ik}x_{jk}$. Here, we shall assume that $X_i = \{x_{i,1}, \dots, x_{i,n}\}$ and $Y_i = \{y_{i,1}, \dots, y_{i,n}\}$.

Construct the branching program by connecting a node/arc of size $2t$ to three nodes/arcs of size $2t+2$. For $t = 1$, there is just one node $[1, 2]$, and the edge from start-node to it is labeled $y_{1,2}y_{2,1}(X_0 \cdot X_1)$. For $t > 1$, the node $[L, R] \supset [1, 2]$ of size $2t < d$ is connected to the three nodes: $[L-2, R], [L-1, R+1]$, and $[L, R+2]$. (It may be the case that the three nodes are the end-node.) The edge labeling is:

- The edge between $[L, R]$ and $[L-2, R]$ is labeled $y_{L-2,L-1}y_{L-1,L-2}(X_{L-2} \cdot X_{L-1})$.
- The edge between $[L, R]$ and $[L-1, R+1]$ is labeled $y_{L-1,R+1}y_{R+1,L-1}(X_{L-1} \cdot X_{R+1})$.
- The edge between $[L, R]$ and $[L, R+2]$ is labeled $y_{R+1,R+2}y_{R+2,R+1}(X_{R+1} \cdot X_{R+2})$.

Consider the ABP thus described, and the polynomial $G_{n,d}$ it computes. For every path γ from start-node to end-node in the ABP, the list of edges along γ yields a pairing P; every edge e in γ corresponds to a pair $P_e = \{i_e, j_e\}$ of nodes in d-cycle. Thus,

$$G_{n,d} = \sum_{\gamma} \prod_{e=\{i_e,j_e\} \in \gamma} y_{i_e,j_e}y_{j_e,i_e} \cdot (X_{i_e} \cdot X_{j_e}). \tag{1}$$

where the sum is over all paths γ from start-node to end-node.

▶ Remark 3.6. There is in fact a one-to-one correspondence between pairings P and such paths γ (this follows by induction on t). Note that this is true only because pairings are tuples i.e., they are *ordered* by definition. Otherwise, it is of course still possible to obtain the same *set* of pairs in a given pairing using multiple different orderings. The sum defining $G_{n,d}$ can be thought of, therefore, as over pairings P.

The following statement summarizes the main useful property of $G_{n,d}$.

▶ **Lemma 3.7** ([26]). *Over any field of characteristic zero, the polynomial $G_{n,d}$ defined above is arc-full-rank as a set-multilinear polynomial in the variables X over the field (Y) of rational functions in Y.*

Proof. Let $w \sim \mathcal{D}$ be an arc-partition. We want to show that $\mathcal{M}_w(G_{n,d})$ has full rank. The arc-partition w is defined from a pairing $P = (P_1, \ldots, P_{d/2})$ (though as discussed in Remark 3.6, there could be multiple such P). The pairing P corresponds to a path γ from start-node to end-node. Consider the polynomial f that is obtained by setting every $y_{i,j} = y_{j,i} = 0$ in F such that $\{i,j\}$ is not a pair in P, and setting every $y_{i,j} = y_{j,i} = 1$ for every pair $\{i,j\}$ in P. Then, it is easy to see that the only terms that survive in Equation 1 correspond to paths (and in turn, pairings) which have the same underlying *set* of pairs as P. As a consequence, f is simply some non-zero constant times a polynomial which is full-rank (recall Observation 3.1). $M_w(f)$ being full rank then implies that $M_w(G_{n,d})$ is also full-rank. ◀

References

1 Manindra Agrawal, Rohit Gurjar, Arpita Korwar, and Nitin Saxena. Hitting-sets for ROABP and sum of set-multilinear circuits. *SIAM J. Comput.*, 44(3):669–697, 2015. doi:10.1137/140975103.

2 Matthew Anderson, Michael A. Forbes, Ramprasad Saptharishi, Amir Shpilka, and Ben Lee Volk. Identity testing and lower bounds for read-k oblivious algebraic branching programs. *ACM Trans. Comput. Theory*, 10(1):3:1–3:30, 2018. doi:10.1145/3170709.

3 Vikraman Arvind and S. Raja. Some lower bound results for set-multilinear arithmetic computations. *Chic. J. Theor. Comput. Sci.*, 2016, 2016. URL: http://cjtcs.cs.uchicago.edu/articles/2016/6/contents.html.

4 C. S. Bhargav, Sagnik Dutta, and Nitin Saxena. Improved lower bound, and proof barrier, for constant depth algebraic circuits. In Stefan Szeider, Robert Ganian, and Alexandra Silva, editors, *47th International Symposium on Mathematical Foundations of Computer Science, MFCS 2022, August 22-26, 2022, Vienna, Austria*, volume 241 of *LIPIcs*, pages 18:1–18:16. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPICS.MFCS.2022.18.

5 C.S. Bhargav, Prateek Dwivedi, and Nitin Saxena. Lower bounds for the sum of small-size algebraic branching programs. *To appear in the proceedings of the Annual Conference on Theory and Applications of Models of Computation*, 2024. URL: https://www.cse.iitk.ac.in/users/nitin/papers/sumRO.pdf.

6 Vishwas Bhargava and Sumanta Ghosh. Improved hitting set for orbit of roabps. *Comput. Complex.*, 31(2):15, 2022. doi:10.1007/S00037-022-00230-9.

7 Pranav Bisht and Nitin Saxena. Blackbox identity testing for sum of special roabps and its border class. *Comput. Complex.*, 30(1):8, 2021. doi:10.1007/S00037-021-00209-Y.

8 Peter Bürgisser. Cook's versus valiant's hypothesis. *Theor. Comput. Sci.*, 235(1):71–88, 2000. doi:10.1016/S0304-3975(99)00183-8.

9 Prerona Chatterjee, Mrinal Kumar, Adrian She, and Ben Lee Volk. Quadratic lower bounds for algebraic branching programs and formulas. *Comput. Complex.*, 31(2):8, 2022. doi:10.1007/S00037-022-00223-8.

10 Zeev Dvir, Guillaume Malod, Sylvain Perifel, and Amir Yehudayoff. Separating multilinear branching programs and formulas. In Howard J. Karloff and Toniann Pitassi, editors, *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 615–624. ACM, 2012. doi:10.1145/2213977.2214034.

11 Michael A. Forbes, Ramprasad Saptharishi, and Amir Shpilka. Hitting sets for multilinear read-once algebraic branching programs, in any order. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 867–875. ACM, 2014. doi:10.1145/2591796.2591816.

12 Michael A. Forbes and Amir Shpilka. Quasipolynomial-time identity testing of non-commutative and read-once oblivious algebraic branching programs. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 243–252. IEEE Computer Society, 2013. doi:10.1109/FOCS.2013.34.

13 Purnata Ghosal and B. V. Raghavendra Rao. Limitations of sums of bounded read formulas and abps. In Rahul Santhanam and Daniil Musatov, editors, *Computer Science - Theory and Applications - 16th International Computer Science Symposium in Russia, CSR 2021, Sochi, Russia, June 28 - July 2, 2021, Proceedings*, volume 12730 of *Lecture Notes in Computer Science*, pages 147–169. Springer, 2021. doi:10.1007/978-3-030-79416-3_9.

14 Zeyu Guo and Rohit Gurjar. Improved explicit hitting-sets for roabps. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2020)*. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020.

15 Rohit Gurjar, Arpita Korwar, and Nitin Saxena. Identity testing for constant-width, and any-order, read-once oblivious arithmetic branching programs. *Theory Comput.*, 13(1):1–21, 2017. doi:10.4086/TOC.2017.V013A002.

16 Rohit Gurjar, Arpita Korwar, Nitin Saxena, and Thomas Thierauf. Deterministic identity testing for sum of read-once oblivious arithmetic branching programs. *Comput. Complex.*, 26(4):835–880, 2017. doi:10.1007/S00037-016-0141-Z.

17 Rohit Gurjar and Ben Lee Volk. Pseudorandom bits for oblivious branching programs. *ACM Trans. Comput. Theory*, 12(2):8:1–8:12, 2020. doi:10.1145/3378663.

18 Maurice J. Jansen. Lower bounds for syntactically multilinear algebraic branching programs. In Edward Ochmanski and Jerzy Tyszkiewicz, editors, *Mathematical Foundations of Computer Science 2008, 33rd International Symposium, MFCS 2008, Torun, Poland, August 25-29, 2008, Proceedings*, volume 5162 of *Lecture Notes in Computer Science*, pages 407–418. Springer, 2008. doi:10.1007/978-3-540-85238-4_33.

19 Neeraj Kayal. An exponential lower bound for the sum of powers of bounded degree polynomials. *Electron. Colloquium Comput. Complex.*, TR12-081, 2012. arXiv:TR12-081.

20 Neeraj Kayal, Nutan Limaye, Chandan Saha, and Srikanth Srinivasan. An exponential lower bound for homogeneous depth four arithmetic formulas. *SIAM J. Comput.*, 46(1):307–335, 2017. doi:10.1137/151002423.

21 Neeraj Kayal, Vineet Nair, and Chandan Saha. Separation between read-once oblivious algebraic branching programs (roabps) and multilinear depth-three circuits. *ACM Trans. Comput. Theory*, 12(1):2:1–2:27, 2020. doi:10.1145/3369928.

22 Neeraj Kayal, Chandan Saha, and Ramprasad Saptharishi. A super-polynomial lower bound for regular arithmetic formulas. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 146–153. ACM, 2014. doi:10.1145/2591796.2591847.

23 Neeraj Kayal, Chandan Saha, and Sébastien Tavenas. An almost cubic lower bound for depth three arithmetic circuits. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, volume 55 of *LIPIcs*, pages 33:1–33:15. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.ICALP.2016.33.

24 Mrinal Kumar and Shubhangi Saraf. On the power of homogeneous depth 4 arithmetic circuits. *SIAM J. Comput.*, 46(1):336–387, 2017. doi:10.1137/140999335.

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

26 Deepanshu Kush and Shubhangi Saraf. Near-optimal set-multilinear formula lower bounds. In Amnon Ta-Shma, editor, *38th Computational Complexity Conference, CCC 2023, July 17-20, 2023, Warwick, UK*, volume 264 of *LIPIcs*, pages 15:1–15:33. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPIcs.CCC.2023.15.

27 Nutan Limaye, Srikanth Srinivasan, and Sébastien Tavenas. Superpolynomial lower bounds against low-depth algebraic circuits. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 804–814. IEEE, 2021. doi:10.1109/FOCS52979.2021.00083.

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

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

30 Noam Nisan and Avi Wigderson. Lower bounds on arithmetic circuits via partial derivatives. *Comput. Complex.*, 6(3):217–234, 1997. doi:10.1007/BF01294256.

31 C. Ramya and B. V. Raghavendra Rao. Lower bounds for special cases of syntactic multilinear abps. *Theor. Comput. Sci.*, 809:1–20, 2020. doi:10.1016/J.TCS.2019.10.047.

32 Ran Raz. Separation of multilinear circuit and formula size. *Theory Comput.*, 2(6):121–135, 2006. doi:10.4086/toc.2006.v002a006.

33 Ran Raz. Tensor-rank and lower bounds for arithmetic formulas. *J. ACM*, 60(6):40:1–40:15, 2013. doi:10.1145/2535928.

34 Ran Raz and Amir Shpilka. Deterministic polynomial identity testing in non-commutative models. *Comput. Complex.*, 14(1):1–19, 2005. doi:10.1007/S00037-005-0188-8.

35 Ran Raz and Amir Yehudayoff. Balancing syntactically multilinear arithmetic circuits. *Comput. Complex.*, 17(4):515–535, 2008. doi:10.1007/S00037-008-0254-0.

36 Ran Raz and Amir Yehudayoff. Lower bounds and separations for constant depth multilinear circuits. *Comput. Complex.*, 18(2):171–207, 2009. doi:10.1007/s00037-009-0270-8.

37 Ramprasad Saptharishi. A survey of lower bounds in arithmetic circuit complexity. *Github Survey*, 2015. URL: https://github.com/dasarpmar/lowerbounds-survey.

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

39 Sébastien Tavenas, Nutan Limaye, and Srikanth Srinivasan. Set-multilinear and non-commutative formula lower bounds for iterated matrix multiplication. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 416–425. ACM, 2022. doi:10.1145/3519935.3520044.

40 L. G. Valiant. Completeness classes in algebra. In *Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing*, STOC '79, pages 249–261, New York, NY, USA, 1979. Association for Computing Machinery. doi:10.1145/800135.804419.