Separation Between Read-once Oblivious Algebraic Branching Programs (ROABPs) and Multilinear Depth-three Circuits

NEERAJ KAYAL, Microsoft Research, India
VINEET NAIR and CHANDAN SAHA, Indian Institute of Science, India

We show an exponential separation between two well-studied models of algebraic computation, namely, read-once oblivious algebraic branching programs (ROABPs) and multilinear depth-three circuits. In particular, we show the following:

(1) There exists an explicit n-variate polynomial computable by linear sized multilinear depth-three circuits (with only two product gates) such that every ROABP computing it requires $2^{\Omega(n)}$ size.
(2) Any multilinear depth-three circuit computing $\mathrm{IMM}_{n,d}$ (the iterated matrix multiplication polynomial formed by multiplying d, $n \times n$ symbolic matrices) has $n^{\Omega(d)}$ size. $\mathrm{IMM}_{n,d}$ can be easily computed by a poly(n, d) sized ROABP.
(3) Further, the proof of (2) yields an exponential separation between multilinear depth-four and multilinear depth-three circuits: There is an explicit n-variate, degree d polynomial computable by a poly(n) sized multilinear depth-four circuit such that any multilinear depth-three circuit computing it has size $n^{\Omega(d)}$. This improves upon the quasi-polynomial separation of Reference [36] between these two models.

The hard polynomial in (1) is constructed using a novel application of expander graphs in conjunction with the evaluation dimension measure [15, 33, 34, 36], while (2) is proved via a new adaptation of the dimension of the partial derivatives measure of Reference [32]. Our lower bounds hold over any field.

CCS Concepts: • **Theory of computation → Algebraic complexity theory**;

Additional Key Words and Phrases: Multilinear depth-three circuits, read-once oblivious algebraic branching programs, evaluation dimension, skewed partial derivatives, expander graphs, iterated matrix multiplication

ACM Reference format:
Neeraj Kayal, Vineet Nair, and Chandan Saha. 2020. Separation Between Read-once Oblivious Algebraic Branching Programs (ROABPs) and Multilinear Depth-three Circuits. *ACM Trans. Comput. Theory* 12, 1, Article 2 (February 2020), 27 pages.
https://doi.org/10.1145/3369928

Authors' addresses: N. Kayal, Microsoft Research, Vigyan 9, Lavelle Road, Shanthala Nagar, Ashok Nagar, Bengaluru, Karnataka, 560001, India; email: neeraka@microsoft.com; V. Nair and C. Saha, Indian Institute of Science, Bengaluru, Karnataka, 560012, India; emails: {vineet, chandan}@iisc.ac.in.

1942-3454/2020/02-ART2 $15.00
https://doi.org/10.1145/3369928

1 INTRODUCTION

Proving lower bounds and separating complexity classes lie at the heart of complexity theory. In algebraic complexity, separating classes VP and VNP (the algebraic analogues of P and NP) equates to proving super-polynomial lower bounds for arithmetic circuits. Another prominent and pertinent problem is polynomial identity testing (PIT). To solve PIT, we need to determine whether a multivariate polynomial computed by an arithmetic circuit over some field is identically zero. A polynomial time randomized algorithm for PIT follows easily from References [10, 40, 45]. PIT is one of the very few natural problems in BPP (in fact, in co-RP) not known to be in P. Showing arithmetic circuit lower bounds and derandomizing PIT are closely related: Reference [23] showed that a polynomial time PIT over integers implies a super-polynomial arithmetic circuit lower bound for the family of permanent polynomials or NEXP $\nsubseteq$ P/poly. References [1, 20] showed that a polynomial time blackbox[1] PIT implies exponential lower bounds for circuits computing polynomials whose coefficients can be computed in PSPACE. Conversely, Reference [23] also showed that a super-polynomial (exponential) circuit lower bound for any family of exponential-time computable multilinear[2] polynomials implies a sub-exponential (quasi-polynomial) time algorithm for blackbox PIT, using Nisan-Wigderson generators [31] and Kaltofen's [24] polynomial factorization algorithm. Reference [13] showed a similar connection between lower bounds and PIT for low-depth circuits, that is lower bounds for bounded depth circuits imply efficient PIT for bounded depth circuits computing polynomials with low individual degree. So, in this certain sense the complexity of proving strong lower bounds and devising efficient PIT algorithms are quite similar. Derandomizing PIT is also interesting in its own right. It is well-known that such a derandomization would imply the problem of checking existence of a perfect matching in a given graph is in NC [28, 43].

Research over the the past several years has made notable progress on both lower bounds and PIT for interesting special cases of arithmetic circuits and helped identify the frontiers of our current knowledge. In particular, we understand better the reason why super-polynomial lower bounds and poly-time PIT have remained elusive even for depth-three circuits: An exponential lower bound (similarly, a poly-time blackbox PIT) for depth-three circuits over fields of characteristic zero implies an exponential lower bound (similarly, quasi-polynomial-time PIT) for general circuits [17]. For more on lower bounds and PIT refer to the surveys in References [8, 26, 37–39, 41].

A potentially useful and interesting restriction to consider at depth three is multilinearity. Most of the hard polynomials used in the literature are multilinear, e.g., determinant, permanent, iterated matrix multiplication, Nisan-Wigderson polynomials, and so on. So, it is worthwhile to develop a fuller understanding of multilinear models [12, 33–36] (meaning, every product gate computes a multilinear polynomial). We do know of strong lower bounds for multilinear depth-three circuits due to Reference [36] and also this article (Theorem 1.7), but as yet no efficient (meaning, quasi-polynomial) PIT is known for this model. One reason for this is the absence of hardness versus randomness tradeoff results for bounded depth multilinear circuits. Recently, Reference [9] has given a sub-exponential time blackbox PIT algorithm for multilinear depth-three circuits using recently found quasi-polynomial time blackbox PIT for another model, namely, read-once oblivious algebraic branching programs (ROABPs) [2, 15] (Definition 1.2), thereby connecting these two interesting models of computation. Could there be a more efficient reduction from multilinear depth-three circuits to ROABPs? If so, then that would readily imply an efficient PIT algorithm for multilinear depth-three circuits. This question has lead us to this work.

[1] Here, blackbox means we are only allowed to evaluate the circuit at points from $\mathbb{F}^n$, where n is the number of inputs and $\mathbb{F}$ the underlying field.

[2] A polynomial is multilinear if the degree of every variable in it is at most one.

Related work and motivation. The model ROABP (see Definition 1.2) has been studied intensely in the recent years in the context of black-box PIT, equivalently *hitting-set generators* (Definition 2.10). This has resulted in deterministic, quasi-polynomial time hitting-set generators for ROABPs [2, 15] and other associated models like set-multilinear algebraic branching programs [14, 15] (a special case of which is set-multilinear depth-three circuits [3, 15]), non-commutative algebraic branching programs [15] and diagonal depth-three circuits [3, 15]. Quite recently, Reference [9] has given a $2^{\tilde{O}(n^{\frac{2}{3}(1+\delta)})}$ time hitting-set generator for multilinear depth-three circuits of size at most $2^{n^{\delta}}$ by "reducing" a multilinear depth three circuit to a collection of ROABPs and "putting together" the hitting-sets of the ROABPs. This "putting together" process raises the hitting-set complexity from quasi-polynomial (for a single ROABP) to sub-exponential (for a composition of several ROABPs). Had it been the case that a multilinear depth-three circuit can be directly reduced to a single small size ROABP, an efficient hitting set for the former would have ensued immediately from References [2, 15]. One of the results in the article (Theorem 1.6), rules out this possibility. In fact, Theorem 1.6 shows something stronger as described below.

A closer look at References [2, 9] reveals an interesting, and potentially useful, intermediate model that we call *superposition of (two or more) set-multilinear depth-three circuits* (see Definition 1.5). An example of superposition of two set-multilinear depth-three circuits is

$$C(X, Y) = (1 + 3x_1 + 5y_2)(4 + x_2 + y_1) + (6 + 9x_1 + 4y_1)(2 + 5x_2 + 3y_2).$$

The variable sets $X = \{x_1, x_2\}$ and $Y = \{y_1, y_2\}$ are completely disjoint and are called the *base sets* of $C(X, Y)$. When projected on X variables (i.e., after putting the Y variables to zero), $C(X, Y)$ is a set-multilinear depth-three circuit in the X variables. A similar thing is true for the Y variables. Thus, every base set is associated with a set-multilinear depth-three circuit and vice versa. Any multilinear depth-three circuit can be trivially viewed as a superposition of n set-multilinear depth-three circuits with single variable in every base set, where n is the number of variables. A crucial observation in Reference [9] is that every multilinear depth-three circuit is *"almost"* a superposition of n^{ϵ} set-multilinear depth-three circuits for some $\epsilon < 1$, and further the associated n^{ϵ} base sets can be found in sub-exponential time using k-wise independent hash functions. Once we know the $r = n^{\epsilon}$ base sets corresponding to r set-multilinear depth-three circuits whose superposition forms a circuit of size s, finding a hitting set for the circuit in time $s^{r \cdot \log s}$ follows easily by taking a direct product of hitting sets for r many set-multilinear depth-three circuits (in fact r many ROABPs, as polynomial sized set-multilinear depth-three circuits reduces to polynomial sized ROABPs). We think a useful model to consider at this juncture is superposition of constantly many set-multilinear depth-three circuits with <u>*unknown*</u> base sets. In this case knowing the $r = O(1)$ base sets readily gives us a quasi-polynomial time hitting set generator, but finding these base sets from a given circuit is NP-hard for $r \geq 3$ (as we show in Observation 1.1), which rules out the possibility of knowing the base-sets even if we are allowed to see the circuit (as in the *white-box* case). Indeed, even in this special case where the given multilinear depth-three circuit is promised to be a superposition of constantly many (say, 2) set-multilinear depth-three circuits, the algorithm in Reference [9] finds and works with many base sets and the resulting hitting set complexity grows to roughly $\exp(\sqrt{n})$. Could it be that the superposition of constantly many set-multilinear depth-three circuits efficiently reduce to ROABPs? Unfortunately, the answer to this also turns out to be negative as Theorem 1.6 gives an explicit example of a superposition of *two* set-multilinear depth-three circuit computing an n-variate polynomial f such that any ROABP computing f has width $2^{\Omega(n)}$.

While comparing two models (here multilinear depth-three circuits and ROABPs), it is desirable to show a separation in both directions whenever an efficient reduction from one to the other seems infeasible. In this sense, we show a complete seperation between the models under consideration by giving an explicit polynomial computable by a polynomial sized ROABP such that every multilinear depth-three circuit computing it requires exponential size. In fact, this explicit polynomial is simply the Iterated Matrix Multiplication $\mathrm{IMM}_{n,d}$—the $(1, 1)$th entry of a product of d $n \times n$ symbolic matrices (Theorem 1.7). $\mathrm{IMM}_{n,d}$ can be easily computed by a polynomial-sized ROABP (see Observation 1.3). Although, a $2^{\Omega(d)}$ lower bound for multilinear depth-three circuit computing Det_d is known [36], this does not imply a lower bound for $\mathrm{IMM}_{n,d}$ (despite the fact that Det and IMM are both complete for algebraic branching programs (ABPs) [29]) as the projection from IMM to Det can make the circuit non-multilinear. Another related work by Reference [12] showed a separation between multilinear ABPs and multilinear formulas by exhibiting an explicit polynomial (namely, an *arc-full-rank* polynomial) that is computable by a linear size multilinear ABP but requires super-polynomial size multilinear formulas. But again multilinearity of a circuit can be lost when IMM is projected to the arc-full-rank polynomial used in Reference [12], and hence this result too does not imply a lower bound for IMM. An extension of Theorem 1.7 to a super-polynomial lower bound for multilinear formulas computing IMM will have interesting consequences in separating noncommutative formulas and noncommutative ABPs. In a contemporary work [25], some of the authors of this work and Sébastien Tavenas have been able to show an $n^{\Omega(\sqrt{d})}$ lower bound for multilinear depth-four circuits computing $\mathrm{IMM}_{n,d}$ by significantly extending a few of the ideas present in this work and building upon (thereby improving) the work of Reference [16]. In summary, the models poly-sized ROABPs and poly-sized multilinear depth-three ciruits have provably different computational powers, although they share a non-trivial intersection as poly-sized set-multilinear depth-three circuits is harbored in both.

An interesting outcome of the proof of the lower bound for multilinear depth-three circuits computing IMM is an exponential separation between multilinear depth-three and multilinear depth-four circuits. Previously, Reference [36] showed a super-polynomial separation between multilinear constant depth h and depth $h + 1$ circuits, which when applied to the depth-three versus depth-four setting gives a quasi-polynomial separation between the two models. In comparison, Theorem 1.8 gives an exponential separation.

The models and our results. We define the relevant models and state our results now.

Definition 1.1 (Algebraic Branching Program). An Algebraic Branching Program (ABP) in the variables $X = \{x_1, x_2, \ldots, x_n\}$ is a directed acyclic graph with a source vertex s and a sink vertex t. It has $(d + 1)$ sets or layers of vertices $V_1, V_2, \ldots, V_{d+1}$, where V_1 and V_{d+1} contain only s and t, respectively. The width of an ABP is the maximum number of vertices in any of the $(d + 1)$ layers. All the edges in an ABP are such that an edge starts from a vertex in V_i and is directed to a vertex in V_{i+1}, where V_i belongs to the set $\{V_1, V_2, \ldots, V_d\}$. The edges in an ABP are labelled by polynomials[3] over a base field $\mathbb{F}$. The weight of the path between any two vertices u and v in an ABP is computed by taking the product of the edge labels on the path from u to v. An ABP computes the sum of the weights of all the paths from s to t.

A special kind of ABP, namely, ROABP, is defined in Reference [15].

Definition 1.2 (Read-Once Oblivious Algebraic Branching Program). A Read-Once Oblivious Algebraic Branching Program(ROABP) over a field $\mathbb{F}$ has an associated permutation $\pi : [n] \to [n]$ of the variables in X. The number of variables is equal to the number of layers of vertices minus one; i.e., $n = (d + 1) - 1 = d$. The label associated with an edge from a vertex in V_i to a vertex in V_{i+1} is an univariate polynomial over $\mathbb{F}$ in the variable $x_{\pi(i)}$.

[3] The edges are labeled by linear polynomials in the standard definition of an ABP.

Definition 1.3 (Multilinear Depth-four and Depth-three Circuits). A circuit $C = \sum_{i=1}^{s} \prod_{j=1}^{d_i} Q_{ij}(X_j^i)$ is a multilinear depth-four ($\Sigma\Pi\Sigma\Pi$) circuit in X variables over a field $\mathbb{F}$, if $X = \uplus_{j=1}^{d_i} X_j^i$ [4] and $Q_{ij} \in \mathbb{F}[X_j^i]$ is a multilinear polynomial for every $i \in [s]$ and $j \in [d_i]$. If Q_{ij}'s are linear polynomials, then C is a multilinear depth-three ($\Sigma\Pi\Sigma$) circuit. The parameter s is the *top fan-in* of C.

Definition 1.4 (Set-multilinear Depth-three Circuit). A circuit $C = \sum_{i=1}^{s} \prod_{j=1}^{d} l_{ij}(X_j)$ is a set-multilinear depth-three ($\Sigma\Pi\Sigma$) circuit in X variables over a field $\mathbb{F}$, if $X = \uplus_{j=1}^{d} X_j$ and $l_{ij} \in \mathbb{F}[X_j]$ is a linear polynomial for every $i \in [s]$ and $j \in [d]$. The sets $X_1, X_2, \ldots, X_d$ are called the *colors* of X. If $|X_j| = 1$ for every $j \in [d]$, then we say X has singleton colors and C is a set-multilinear depth-three circuit with singleton colors.

As a bridge between multilinear and set-multilinear depth-three circuits, we define a model called superposition of set-multilinear depth-three circuits.

Definition 1.5 (Superposition of Set-multilinear Depth-three Circuits). A multilinear depth-three ($\Sigma\Pi\Sigma$) circuit C over a field $\mathbb{F}$ is a superposition of t set-multilinear depth-three circuits over variables $X = \uplus_{i=1}^{t} Y_i$, if for every $i \in [t]$, C is a set-multilinear depth-three circuit in Y_i variables over the field $\mathbb{F}(X \setminus Y_i)$. The sets $Y_1, \ldots, Y_t$ are called the *base sets* of C. Further, we restrict the Y_i to have singleton colors for every $i \in [t]$.

Note that although the notion of superposition makes sense even if Y_i's do not have singleton colors, we restrict to singletons as this model itself captures multilinear depth-three circuits. We make the following initial observation for superposition of set-multilinear depth-three circuits.

Observation 1.1. *Given a circuit C, if C is a superposition of t set-multilinear circuits on unknown base sets $Y_1, Y_2, \ldots, Y_t$, finding t base sets $Y_1', Y_2', \ldots, Y_t'$ such that C is a superposition of t set-multilinear circuits on base sets $Y_1', Y_2', \ldots, Y_t'$ is NP-hard when $t > 2$.*

The proof of the observation appears in Section 6.1. We now state the main results of this article. In Theorem 1.6, we use $\mathbb{P}$ to denote the set of prime numbers.

Theorem 1.6 (Main Theorem 1).

(1) *There is an explicit family of $2n$-variate polynomials $\{f_n\}_{n \in \mathbb{P},\, n \geq 11}$ over any field $\mathbb{F}$ such that the following hold: f_n is computable by a multilinear depth-three circuit C over $\mathbb{F}$ with top fan-in three and C is also a superposition of two set-multilinear depth-three circuits. Any ROABP over $\mathbb{F}$ computing f_n has width $2^{\Omega(n)}$.*

(2) *There is an explicit family of $3n$-variate polynomials $\{g_n\}_{n \in \mathbb{P}}$ over any field $\mathbb{F}$ such that the following hold: g_n is computable by a multilinear depth-three circuit C over $\mathbb{F}$ with top fan-in two and C is also a superposition of three set-multilinear depth-three circuits. Any ROABP over $\mathbb{F}$ computing g_n has width $2^{\Omega(n)}$.*

We prove Theorem 1.6 in Section 3. The tightness of the theorem is exhibited by this observation.

Observation 1.2. *A polynomial computed by a multilinear $\Sigma\Pi\Sigma$ circuit with top fan-in two and at most two variables per linear polynomial can also be computed by an ROABP with constant width.*

The proof of Observation 1.2 is in Section 6.1. Thus, it follows from Theorem 1.6 that if we increase either the top fan-in or the number of variables per linear polynomial from two to three in multilinear depth-three circuits then there exist polynomials computed by such circuits such that ROABPs computing these polynomials have exponential width. We now state the "converse" of Theorem 1.6.

[4] Here $\uplus$ is used to denote the disjoint union of sets.

Theorem 1.7 (Main Theorem 2). *Any multilinear depth-three circuit (over any field) computing* $\mathrm{IMM}_{n,d}$*, the* $(1,1)$*th entry of a product of* d $n \times n$ *symbolic matrices, has top fan-in* $n^{\Omega(d)}$ *for* $n \geq 6$.

Theorem 1.7 also implies a lower bound for determinant, see Corollary 4.2. We prove Theorem 1.7 in Section 4. It is not hard to observe the following.

Observation 1.3. $\mathrm{IMM}_{n,d}$ *can be computed by an* n^2 *width ROABP.*

The proof of Observation 1.3 given in Section 6.1 presents a brute force way to compute $\mathrm{IMM}_{n,d}$ by an ROABP, whereas a more careful analysis yields a width $2n$ ROABP computing $\mathrm{IMM}_{n,d}$. Thus, Theorem 1.6, Theorem 1.7 and Observation 1.3 together imply a complete separation between multilinear depth-three circuits and ROABPs. As a consequence of the proof of Theorem 1.7, we also get an exponential separation between multilinear depth-three and multilinear depth-four circuits. We prove Theorem 1.8 in Section 4.

Theorem 1.8. *There is an explicit family of* $O(n^2 d)$*-variate polynomials of degree* d*,* $\{f_d\}_{d\geq 1}$*, such that* f_d *is computable by a* $O(n^2 d)$*-sized multilinear depth-four circuit with top fan-in* one *(i.e., a* $\Pi\Sigma\Pi$ *circuit) and every multilinear depth-three circuit computing* f_d *has top fan-in* $n^{\Omega(d)}$ *for* $n \geq 11$.

The hard polynomials used in Theorem 1.6 belong to a special class of multilinear depth-three circuits—they are both superpositions of constantly many set-multilinear depth-three circuits and simultaneously a sum of constantly many set-multilinear depth-three circuits. Here is an example of a circuit from this class:

$$C(X,Y) = (1 + 3x_1 + 5y_2)(4 + x_2 + y_1) + (9 + 6x_1 + 4y_2)(3 + 2x_2 + 5y_1)$$
$$+(6 + 9x_1 + 4y_1)(2 + 5x_2 + 3y_2) + (3 + 6x_1 + 9y_1)(5 + 8x_2 + 2y_2).$$

$C(X,Y)$ is a superposition of two set-multilinear depth-three circuits with base sets $X = \{x_1\} \cup \{x_2\}$ and $Y = \{y_1\} \cup \{y_2\}$. But $C(X,Y)$ is also a sum of two set-multilinear depth-three circuits with $\{x_1, y_2\}$, $\{x_2, y_1\}$ being the colors in the first set-multilinear depth-three circuit (corresponding to the first two products) and $\{x_1, y_1\}$, $\{x_2, y_2\}$ the colors in the second set-multilinear depth-three circuit (corresponding to the last two products). For such a subclass of multilinear depth-three circuits, we give a quasi-polynomial time hitting set by extending the proof technique of Reference [3].

Theorem 1.9. *Let* $C_{n,m,l,s}$ *be a subclass of multilinear depth-three circuits computing* n*-variate polynomials such that every circuit in* $C_{n,m,l,s}$ *is a superposition of at most* m *set-multilinear depth-three circuits and simultaneously a sum of at most* l *set-multilinear depth-three circuits, and has top fan-in bounded by* s*. There is a hitting-set generator for* $C_{n,m,l,s}$ *running in* $(ns)^{O(lm\log s)}$ *time.*

When m and l are bounded by $\mathrm{poly}(\log ns)$, we get quasi-polynomial time hitting sets. The proof of Theorem 1.9, which extends the shift and rank concentration technique of Reference [3], is given in Section 5. To our understanding, even if m and l are constants, Reference [9]'s algorithm yields an $exp(\sqrt{n})$ hitting set complexity. Also, Reference [18] has recently given a $(ndw)^{O(l2^l \log(ndw))}$ time hitting set generator for n-variate, individual (variable) degree d polynomials computed by sum of l ROABPs each of width less than w. Sum of l set-multilinear depth-three circuits reduces to sum of l ROABPs as set-multilinear depth-three circuits readily reduce to poly-sized ROABPs. But, observe the doubly exponential dependence on l in their result. On the contrary, in Theorem 1.9 the dependence is singly exponential in l. So, the hitting-set complexity remains quasi-polynomial for $l = (\log n)^{O(1)}$, whereas Reference [18] gives an exponential time hitting-set generator when applied to the model in Theorem 1.9. However, it is also important to note that the model considered in Theorem 1.9 is somewhat weaker than the sum of ROABPs model in Reference [18] because of the additional restriction that our model is also a superposition of m set-multilinear depth-three circuits.

Proof ideas for Theorems 1.6 and 1.7. Theorem 1.6 is proved by connecting the notion of edge expansion (Definition 2.4) with the evaluation dimension measure (Definition 2.1). Starting with an explicit 3-regular bipartite expander G, we associate distinct variables with distinct vertices. Every edge now corresponds to a linear polynomial—it is the sum of the variables associated with the vertices on which the edge is incident upon. A multilinear depth-three circuit C is derived from the expander G as follows: C has three product terms, each term formed by taking product of the linear polynomials associated with the edges of a matching in G. Now, edge expansion of G can be used to argue that for every subset S of variables of a certain size there exists of a product term in C that has high evaluation dimension with respect to S. Further, one can show that high evaluation dimension of a product term implies high evaluation dimension of C with respect to S by restricting the circuit modulo two linear polynomials to nullify the other two product terms. However, for every ROABP there is a set S (of any size) such that the evaluation dimesion of the ROABP with respect to S is bounded by its width. This gives a lower bound on the width of the ROABP computing the same polynomial as C thereby proving part 1 of Theorem 1.6. Part 2 is proved similarly, but now we associate edges and vertices of a bipartite expander G with variables and linear polynomials, respectively. Circuit C is formed by adding two product terms, each term formed by multiplying the linear polynomials associated with the left or the right vertex set of G. As before, edge expansion of C implies for every set S of variables of a certain size there is a product term of C with high evaluation dimension and this in turn implies high evaluation dimension of C.

While writing this article, we came to know about a recent work by Jukna [22] that uses Ramanujan graphs to give an alternate proof of a known exponential lower bound for monotone arithmetic circuits. To our understanding, it does seem that Jukna's proof also implictly relates expansion with evaluation dimension, but the argument in Reference [22] is directed towards monotone circuits and it does not seem to imply any of the lower bounds shown in this work. In particular, the hard polynomial in Reference [22] could have any complexity, whereas in our case we need the hard polynomial to be computable by a small multilinear depth-three circuit.

Theorem 1.7 is proved by introducing a new variant of the dimension of the space of partial derivatives measure that is inspired by References [32, 34]. At a high level, the idea is to consider a polynomial f in two sets of variables X and Y such that $|Y| \gg |X|$. If we take derivatives of f with respect to all degree k monomials in Y-variables and set all the Y-variables to zero after taking derivatives, then we do expect to get a "large" space of derivatives (especially, when f is a "hard" polynomial), simply because $|Y|$ is large. However, in any depth-three multilinear circuit C computing f, the dimension of the space of derivatives of a product term is influenced only by the number of linear polynomials containing the X-variables as all the Y-variables are set to zero subsequently. Thus, the measure is somewhat small for a product term of C as $|X| \ll |Y|$. By subadditivity of the measure (Lemma 2.3), this implies high top fan-in of C computing f. A notable difference with References [34, 36] is that the variable sets X and Y are fixed deterministically, *a priori*, and not by random partitioning of the entire set of variables.

2 PRELIMINARIES

Measures. We have used two complexity measures, namely, evaluation dimension and a novel variant of the dimension of the space of partial derivatives, to prove Theorem 1.6 and 1.7, respectively. Evaluation dimension was first defined in Reference [15].[5] Let X be a set of variables.

[5] They attributed the notion to Ramprasad Saptharishi.

Definition 2.1 (Evaluation Dimension). The evaluation dimension of a polynomial $g \in \mathbb{F}[X]$ with respect to a set $S \subseteq X$, denoted as $\text{Evaldim}_S(g)$, is defined as

$$\dim(\text{span}_{\mathbb{F}}\{g(X)|_{\forall x_j \in S\, x_j = \alpha_j} : \forall x_j \in S\ \alpha_j \in \mathbb{F}\}).$$

Evaluation dimension is a nearly equivalent variant of another measure, the *rank of the partial derivatives matrix*, first defined in Reference [30] to prove lower bounds for non-commutative models. Rank of the partial derivatives matrix measure was also used in References [12, 33–36] to prove lower bounds and separations for several multilinear models. These two measures are identical over fields of characteristic zero (or sufficiently large size).

The partial derivatives measure was introduced in Reference [32]. The following is a simple variant of this measure that is also inspired by the measure used in Reference [34].

Definition 2.2 ("Skewed" Partial Derivatives). Let $f \in \mathbb{F}[X, Y]$, where X and Y are disjoint sets of variables, and $\mathcal{Y}_k$ be the set of all monomials in Y variables of degree $k \in \mathbb{N}$. Define the measure $\text{PD}_{\mathcal{Y}_k}(f)$ as

$$\dim\left(\text{span}_{\mathbb{F}}\left\{\left[\frac{\partial f(X,Y)}{\partial m}\right]_{\forall y \in Y\ y=0} : m \in \mathcal{Y}_k\right\}\right).$$

In proving Theorem 1.7, we apply the above measure with a significant difference (or *skew*) between the number of X and Y variables—it is this imbalance that plays a crucial role in the proof. Both the above measures obey the property of subadditivity. The proof of Lemma 2.3 is in Section 6.2.

Lemma 2.3 (Subadditivity).

(1) *Let* $g_1, g_2 \in \mathbb{F}[X]$ *and* $S \subseteq X$, *then* $\text{Evaldim}_S(g_1 + g_2) \leq \text{Evaldim}_S(g_1) + \text{Evaldim}_S(g_2)$.
(2) *Let* $f_1, f_2 \in \mathbb{F}[X, Y]$, *then* $\text{PD}_{\mathcal{Y}_k}(f_1 + f_2) \leq \text{PD}_{\mathcal{Y}_k}(f_1) + \text{PD}_{\mathcal{Y}_k}(f_2)$.

Expander Graphs. A vital ingredient that helps us construct the hard polynomials in Theorem 1.6 is a family of explicit 3-regular expanders. We recall a few basic definitions from Reference [21].

Definition 2.4 (Edge Expansion and Family of Expanders). Let $G = (V, E)$ be an undirected d-regular graph. For $S \subseteq V$, let $E(S, \overline{S})$ be the set of edges with one end incident on a vertex in S and the other incident on a vertex in $\overline{S} = V \backslash S$. The *edge expansion* of G denoted $h(G)$ is defined as

$$h(G) = \min_{S:\,|S| \leq \frac{|V|}{2}} \frac{|E(S, \overline{S})|}{|S|}.$$

A sequence of d-regular graphs $\{G_i\}_{i \in \mathbb{N}}$ of size increasing with i is a *family of d-regular expanders* if there exists an $\epsilon > 0$ such that $h(G_i) > \epsilon$ for every i.

Definition 2.5 (Spectral Expansion of a Graph). Let $G = (V, E)$ be a d-regular graph with $|V| = n$. Let A_G be the adjacency matrix of G and $d = \lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$ the n eigenvalues of A_G. Then $\lambda(G) \overset{def}{=} \max\{|\lambda_2|, |\lambda_n|\}$. The ordered set of eigenvalues $(\lambda_1, \lambda_2, \ldots, \lambda_n)$ is called the spectrum of G.

We use Theorem 2.6 in the proof of Lemma 2.9. This theorem is due to References [4, 5, 6, 7, 11].

Theorem 2.6 (Cheeger's Inequality). *Let G be a d-regular graph with spectrum $(\lambda_1, \lambda_2, \ldots, \lambda_n)$. Then,*

$$\frac{d - \lambda_2}{2} \leq h(G) \leq \sqrt{2d(d - \lambda_2)}.$$

We use a three regular expander graph family to construct the hard polynomial families in Theorem 1.6. Before we state an explicitly constructible three regular expander graph family, we make precise the notion of explicit expander graphs.

Definition 2.7 (Mildly Explicit Expanders). Let $\mathcal{G} = \{G_i\}_{i \in \mathbb{N}}$ be a family of d-regular expanders such that the number of vertices in G_i is bounded by a polynomial in i. $\mathcal{G}$ is *mildly explicit* if there exists an algorithm that takes input i and constructs G_i in time polynomial in the size of G_i.

A family of mildly explicit expanders. We will use $\mathbb{P}$ to denote the set of prime numbers. Reference [21] mentions a family of mildly explicit 3-regular p-vertex expanders $\{G_p\}_{p \in \mathbb{P}}$ such that for every graph G_p in the family: $h(G_p) > \frac{3}{2}10^{-4}$. The vertices of G_p correspond to elements in $\mathbb{Z}_p$. A vertex x in G_p is connected to $x+1$, $x-1$ and to its inverse x^{-1} (operations are modulo p and inverse of 0 is defined as 0, and a self-loop increases the degree of the vertex by one). We refer the reader to Reference [21], Section 11.1.2, for more details. Denote this family of 3-regular p-vertex expanders by $\mathcal{S}$.

Double Cover. The proof of Theorem 1.6 works with bipartite expanders. It is standard to transform a d-regular expander graph to a d-regular bipartite expander graph by taking its double cover.

Definition 2.8 (Double Cover). The double cover of a graph $G = (V, E)$ is the bipartite graph $\tilde{G} = (L \uplus R, \tilde{E})$ where $|L| = |R| = |V|$. Corresponding to a vertex $u \in V$, we have two vertices $u_L \in L$ and $u_R \in R$. Edges (u_L, v_R) and $(u_R, v_L) \in \tilde{E}$ if and only if there is an edge $(u, v) \in E$.

Lemma 2.9. *Let $\mathcal{S} = \{G_p\}_{p \in \mathbb{P}}$ be the family of expanders as described above, and $\tilde{\mathcal{S}} = \{\tilde{G_p}\}_{p \in \mathbb{P}}$ the family of double covers of graphs in $\mathcal{S}$. Then $h(\tilde{G_p}) > \frac{3}{2} \cdot 10^{-4}$ for every $p \in \mathbb{P}$.*

Proof. The family $\mathcal{S} = \{G_p\}_{p \in \mathbb{P}}$ is such that $\lambda(G_p) < 3(1 - 10^{-4})$ for every p. This has been argued in Section 11.1.2 of [21], where they show that the normalized value of $\lambda(G_p)$ is at most $(1 - 10^{-4})$, i.e., $\frac{\lambda(G_p)}{3} < 1 - 10^{-4}$. Observe that if $(\lambda_1, \ldots, \lambda_p)$ is the spectrum of G_p then $\{\pm\lambda_1, \ldots, \pm\lambda_p\}$ are exactly the eigenvalues of the adjacency matrix of the bipartite graph $\tilde{G_p}$. Hence, $\lambda(G_p)$ is the second largest eigenvalue of $A_{\tilde{G_p}}$. By applying Cheeger's inequality (Theorem 2.6), $h(\tilde{G_p}) > \frac{3}{2} \cdot 10^{-4}$ for every p as $\tilde{G_p}$ is 3-regular. □

Hitting-set generators. In Theorem 1.9, we give a quasi-polynomial time hitting-set generator for a subclass of multilinear depth-three circuits.

Definition 2.10 (Hitting-set Generators). A hitting-set generator for a class of circuits $\mathcal{C}$ is a Turing machine $\mathcal{H}$ that takes $(1^n, 1^s)$ as input and outputs a set $\{a_1, \ldots, a_m\} \subseteq \mathbb{Z}^n$ such that for every circuit $C \in \mathcal{C}$ of size bounded by s and computing a nonzero n-variate polynomial over a field $\mathbb{F} \supset \mathbb{Z}$, there is an $i \in [m]$ for which $C(a_i) \neq 0$. Complexity of $\mathcal{H}$ is its running time.[6]

Hitting set generators are also defined as a polynomial map $\mathcal{H} = (h_1, h_2, \ldots, h_n)$, where each h_i is a t-variate polynomial ($t \ll n$), such that for every circuit $C \in \mathcal{C}$ computing a nonzero n-variate polynomial, $C(h_1, h_2, \ldots, h_n)$ is a nonzero t-variate polynomial. If $|\mathbb{F}| > n$, then it is not hard to argue that the two definitions are equivalent (see Section 4.1 [42]).

Technical Lemmas. The following lemmas are used in the proof of Theorem 1.6. Lemma 2.11 follows from Hall's marriage Theorem [19].

Lemma 2.11. *A d-regular bipartite graph can be split into d edge-disjoint perfect matchings.*

[6] Hitting-set generators can be defined similarly over finite fields by considering field extensions.

Lemma 2.12. *Suppose $g_1(X), g_2(X), \ldots, g_m(X) \in \mathbb{F}[X]$ are $\mathbb{F}$-linearly independent polynomials in the variables $X = \{x_1, x_2, \ldots, x_n\}$ where $m = 2^n$. If $Y = \{y_1, y_2, \ldots, y_n\}$ are n variables different from X, then (by identifying an $i \in [m]$ with an $S \subseteq [n]$),*

$$\text{Evaldim}_Y\left(\sum_{S\subseteq[n]} y_S \cdot g_S(X)\right) = m, \quad \textit{where for } S \subseteq [n],\ y_S := \prod_{j\in S} y_j.$$

Proof. Consider the following $\mathbb{F}$-evaluations of $\{y_1, y_2, \ldots, y_n\}$: for every $S \subseteq [n]$, if $j \in S$ set $y_j = 1$ else set $y_j = 0$. There are $m = 2^n$ such evaluations. By taking appropriate $\mathbb{F}$-linear combinations of these evaluations of the polynomial $\sum_{S\subseteq[n]} y_S \cdot g_S$, one can get the m polynomials $\{g_S\}_{S\subseteq[n]}$. Since these m polynomials are given to be $\mathbb{F}$-linearly independent, $\text{Evaldim}_Y(\sum_{S\subseteq[n]} y_S g_S(X)) \geq m$. However, any $\mathbb{F}$-evaluation of the Y-variables of the polynomial $\sum_{S\subseteq[n]} y_S \cdot g_S(X)$ is a $\mathbb{F}$-linear combination of the m polynomials $\{g_S\}_{S\subseteq[n]}$ and hence $\text{Evaldim}_Y(\sum_{S\subseteq[n]} y_S \cdot g_S(X)) \leq m$. □

Lemma 2.13. *If R is a width-k ROABP that computes $g(X)$, then for every $i \in [0, |X|]$ there exists a set $S \subseteq X$ of size i such that $\text{Evaldim}_S(g) \leq k$.*

Proof. Without loss of generality, assume the permutation π associated with the ROABP R is the identity permutation. Hence, R can be equivalently viewed as a product of n matrices $M_1, \ldots, M_n$ computing $g(X) = M_1 \cdot M_2 \cdots M_n$, where M_1 is a $1 \times k$ matrix with entries from $\mathbb{F}[x_1]$, M_n is a $k \times 1$ matrix with entries from $\mathbb{F}[x_n]$, and M_j is a $k \times k$ matrix with entries from $\mathbb{F}[x_j]$ for every $j \in [2, n-1]$. Let $S = \{x_1, x_2, \ldots, x_i\}$. Consider any $\mathbb{F}$-evaluation of the S variables in $g(X)$. Denote the resulting polynomial by $g_1(X \setminus S) \in \mathbb{F}[x_{i+1}, \ldots, x_n]$. Observe that $g_1(X \setminus S) = M_{eval} \cdot M_{i+1} \cdots M_n$ where $M_{eval} \in \mathbb{F}^{1\times k}$. Let $M = M_{i+1} \cdots M_n$ be the $k \times 1$ column vector with entries from $\mathbb{F}[x_{i+1}, \ldots, x_n]$. Thus, $g_1(X \setminus S) = M_{eval} \cdot M$ is an $\mathbb{F}$-linear combination of k polynomials in $\mathbb{F}[x_{i+1}, \ldots, x_n]$ that do not depend on which evaluation of the $\{x_1, \ldots, x_i\}$-variables we began with. Hence, evaluation dimension of $g(X)$ with respect to S is upper bounded by k. □

3 LOWER BOUNDS FOR ROABP: PROOF OF THEOREM 1.6

Proof of Part 1

Construction of the polynomial family. We construct a family of $2n$-variate multilinear polynomials $\{f_n\}_{n\in\mathbb{P}, n\geq 11}$ from the explicit family of 3-regular expander graphs $\mathcal{S}$ (described in section 2). From an n-vertex graph $G = (V, E)$ in $\mathcal{S}$, construct a polynomial $f(X, Y)$ in variables $X = \{x_1, \ldots, x_n\}$ and $Y = \{y_1, \ldots, y_n\}$ as follows: Let $\tilde{G} = (L \uplus R, \tilde{E})$ be the double cover of G. By Lemma 2.9, $h(\tilde{G}) > \frac{3}{2}10^{-4}$. With every vertex in L (similarly, R) associate a unique variable in X (respectively, Y), thus vertices in L and R are identified with the X and Y variables, respectively. An edge between x_i and y_j is associated with the linear polynomial $(x_i + y_j)$. By Lemma 2.11, $\tilde{G}$ can be split into three edge-disjoint perfect matchings. Corresponding to every perfect matching, we have a product term formed by taking product of the linear polynomials associated with the edges of the matching. Polynomial $f(X, Y)$ is the sum of the three product terms corresponding to the three edge-disjoint perfect matchings of $\tilde{G}$. It is easy to show the following claim, proof given in Section 6.3.

Claim 3.1. *Polynomial f (constructed above) is computed by a multilinear depth-three circuit C of size $\Theta(n)$ and top fan-in three, and C is a superposition of two set-multilinear depth-three circuits.*

High evaluation dimension of $f(X, Y)$. It turns out that the evaluation dimension of $f(X, Y)$ with respect to any subset of variables of size $n/10$ is large.

Lemma 3.1. *For any set $S \subseteq X \uplus Y$ of size $n/10$,* $\text{Evaldim}_S(f) \geq 2^{\epsilon n}$ *where $\epsilon > 0$ is a constant.*

Proof. Consider any subset S of $n/10$ variables from $X \uplus Y$. With respect to set S, we can classify the linear polynomials in the product terms of $f(X, Y)$ into three types: *untouched*—if none of the two variables in the linear polynomial belong to S, *partially touched*—if exactly one of the variables in the linear polynomial belongs to S, and *completely touched*—if both variables belong to S. Call the three product terms of f: P_1, P_2, and P_3. □

Claim 3.2. *There exists a set $X_0 \subseteq X$ of $(\frac{7n}{10} - 4)$ X-variables such that every $x \in X_0$ appears in an untouched linear polynomial in every P_i (for $i \in [3]$), and further if $(x + y_{j_1}), (x + y_{j_2})$ and $(x + y_{j_3})$ are the linear polynomials occurring in P_1, P_2, and P_3, respectively, then $y_{j_1} \neq y_{j_2} \neq y_{j_3}$.*

Proof. For every $i \in [3]$, let D_i represent the set of touched linear polynomials in product gate i. Hence, $|D_1| + |D_2| + |D_3| \leq \frac{3n}{10}$. Thus, the number of X-variables that are part of these touched linear polynomials is at most $\frac{3n}{10}$ as every linear polynomial has exactly one X-variable. This implies at least $\frac{7n}{10}$ X-variables are part of untouched linear polynomials in every product gate. As $f(X, Y)$ is constructed from $\tilde{G}$, two product gates contain the same linear polynomial l if and only if there is a double edge between the endpoints of the edge corresponding to the linear polynomial l in $\tilde{G}$. Graph $\tilde{G}$ is the double cover of the n-vertex graph $G \in \mathcal{S}$ where $n \geq 11$ is a prime. A double edge between vertices u_L and v_R in $\tilde{G}$ implies existence of a double edge between vertices u and v in G. Vertices of G are identified with elements of $\mathbb{Z}_n$. A vertex a in G_n is connected to $a + 1, a - 1$ and a^{-1} (operations are modulo n and inverse of 0 is 0). Thus, there is a double edge incident on a vertex a in G if and only if any two of the vertices $a + 1, a - 1$ and a^{-1} are the same. If $a + 1 = a - 1 \mod n$, then $2 = 0 \mod n$, which cannot be true as $n \geq 11$. Hence, if there is a double edge incident on a then either $a + 1 = a^{-1} \mod n$, or $a - 1 = a^{-1} \mod n$. This means G has exactly two sets of double edges – between $\frac{-1-\sqrt{5}}{2}$ and $\frac{1-\sqrt{5}}{2}$, and between $\frac{-1+\sqrt{5}}{2}$ and $\frac{1+\sqrt{5}}{2}$ – if 5 is a square in $\mathbb{Z}_n$; otherwise, G has no double edge. As a double edge in G gives rise to two double edges in $\tilde{G}$, the latter has at most four double edges. Thus, at most four out of the $\frac{7n}{10}$ X-variables are part of untouched linear polynomials that appear in more than one product gate. We remove these four variables. X_0 is the set of the remaining X-variables of size at least $(\frac{7n}{10} - 4)$. Naturally, every variable in X_0 has the desired property as stated in the claim. □

For $i \in [3]$, let B_i be the set of partially touched linear polynomials in term P_i.

Claim 3.3. *There is an $i \in [3]$ such that $|B_i| \geq \epsilon n$ where $\epsilon = 10^{-6}$.*

Proof. Let $T = \max_{i \in [3]} \{|B_i|\}$. Recall that f has been constructed from the bipartite expander $\tilde{G}$, and vertices in $\tilde{G}$ identified with the variable set $X \uplus Y$. We denote the vertices in $\tilde{G}$ corresponding to the variables in S also by S, and denote the set of edges going out from S to $\overline{S} = L \uplus R \backslash S$ in $\tilde{G}$ by $\tilde{E}(S, \overline{S})$. Using the expansion property of $\tilde{G}$,

$$|\tilde{E}(S, \overline{S})| \geq h(\tilde{G}) \cdot |S| \geq \frac{3}{2} 10^{-4} \cdot \left(\frac{n}{10}\right).$$

Every edge in $\tilde{E}(S, \overline{S})$ corresponds to a partially touched linear polynomial. Since $\tilde{G}$ is 3-regular, at least $\frac{|\tilde{E}(S, \overline{S})|}{3}$ of the edges correspond to distinct partially touched linear polynomials. By assumption, the number of such partially touched linear polynomials is at most $3T$; and so $T \geq 10^{-6} \cdot n$. □

The next claim completes the proof of Lemma 3.1.

Claim 3.4. *If there exists an $i \in [3]$ such that $|B_i| \geq \epsilon n$ for $\epsilon > 0$, then* $\text{Evaldim}_S(f) \geq 2^{\epsilon n}$.

Proof. Without loss of generality, assume $|B_1| \geq \epsilon n$. Pick two variables, say x and x', from the set X_0 (as described in Claim 3.2). Here, $|X_0| \geq \frac{7n}{10} - 4 \geq 2$, for $n \geq 11$. Let $(x + y_{j_2})$ and $(x' + y'_{j_3})$ be the linear polynomials appearing in P_2 and P_3, respectively. By substituting $x = -y_{j_2}$ and $x' = -y'_{j_3}$ in g, the terms P_2 and P_3 vanish but P_1 does not (by Claim 3.2). Let $\hat{f}$ be the polynomial f after the substitution. Polynomial $\hat{f}$ has only one product term $\hat{P_1}$ (i.e., P_1 under the substitution), and $\hat{P_1}$ has as many partially touched linear polynomials as P_1. At this point, we use the following observation. □

Observation 3.1. $\text{Evaldim}_S(f) \geq \text{Evaldim}_S(\hat{f}) = \text{Evaldim}_S(\hat{P_1}) \geq 2^{\epsilon n}$.

Proof. It is easy to see $\text{Evaldim}_S(f) \geq \text{Evaldim}_S(\hat{f}) = \text{Evaldim}_S(\hat{P_1})$ as follows. Let

$$V = \text{span}_{\mathbb{F}}\{f(X,Y)|_{\forall u_j \in S\, u_j = \alpha_j} : \forall u_j \in S\ \alpha_j \in \mathbb{F}\},$$

$$\hat{V} = \text{span}_{\mathbb{F}}\{\hat{P_1}(X,Y)|_{\forall u_j \in S\, u_j = \alpha_j} : \forall u_j \in S\ \alpha_j \in \mathbb{F}\},$$

and $t = \text{Evaldim}_S(f)$. Let $\{h_1, \ldots, h_t\}$ be a basis of V. Since the linear polynomials $(x + y_{j_2})$ and $(x' + y'_{j_3})$ are untouched, the variables x, x', y_{j_2}, y_{j_3} do not belong to S and hence the polynomials $\{\hat{h}_1, \ldots, \hat{h}_t\}$ span the space $\hat{V}$, where $\hat{h}_i$ is polynomial h_i under the substitution $x = -y_{j_2}$ and $x' = y_{j_3}$. Below, we show $\text{Evaldim}_S(\hat{P_1}) \geq 2^{\epsilon n}$.

Suppose $\hat{P_1}$ has $T \geq \epsilon n$ partially touched linear polynomials $\{l_1, l_2, \ldots, l_T\}$. For every $r \in [T]$, let $l_r = z_r + u_r$ where $z_r \in S$ and $u_r \in (X \cup Y)\backslash S$. Let $Z = \{z_1, z_2, \ldots, z_T\}$. Then substitute all variables in $S\backslash Z$ to 1. Suppose $\tilde{P_1}$ is equal to $\hat{P_1}$ after this substitution. Then it follows easily that $\text{Evaldim}_Z(\tilde{P_1}) \leq \text{Evaldim}_S(\hat{P_1})$ as $Z \subseteq S$. Let q be the polynomial formed by multiplying all linear polynomials in $\tilde{P_1}$ that are free of variables in Z. Then,

$$\tilde{P_1} = \left(\sum_{\nu \subseteq [T]} z_\nu u_{[T]\backslash \nu}\right) \cdot q,$$

where $z_\nu = \prod_{j \in \nu} z_j$ and $u_{[T]\backslash\nu} = \prod_{j \in [T]\backslash \nu} u_j$. Since q is Z-free, by Lemma 2.12, we have $\text{Evaldim}_Z(\tilde{P_1}) = \text{Evaldim}_Z(\sum_{\nu \subseteq [T]} z_\nu u_{[T]\backslash\nu}) = 2^T$. □

This completes the proof of Claim 3.4.

From Lemmas 2.13 and 3.1, we conclude that any ROABP computing $f(X,Y)$ has width at least $2^{\epsilon n}$.

Proof of Part 2

Construction of the polynomial family. Similar to part 1, we construct a family of $3n$-variate multilinear polynomials $\{g_n\}_{n \in \mathbb{P}}$ from the explicit family of 3-regular expanders $\mathcal{S}$ – but this time edges will be associated with variables and vertices with linear polynomials. From an n-vertex graph $G = (V, E)$ in $\mathcal{S}$, construct a polynomial $g(X, Y, Z)$ in variables $X = \{x_1, \ldots, x_n\}$, $Y = \{y_1, \ldots, y_n\}$ and $Z = \{z_1, \ldots, z_n\}$ as follows: Let $\tilde{G} = (L \uplus R, \tilde{E})$ be the double cover of G, and as before $h(\tilde{G}) > \frac{3}{2}10^{-4}$. Edges of $\tilde{G}$ can be split into three edge-disjoint perfect matchings (by Lemma 2.11). Label the edges of the first perfect matching by distinct X-variables, the edges of the second matching by distinct Y-variables, and the edges of the third by distinct Z-variables. Vertices of $\tilde{G}$ now correspond to linear polynomials naturally—if the three edges incident on a vertex are labelled x_i, y_j, and z_k, then associate the linear polynomial $(x_i + y_j + z_k)$ with the vertex. Let P_1 be the product of the linear polynomials associated with the vertices of L, and P_2 the product of linear polynomials associated with the vertices of R. Polynomial $g(X, Y, Z)$ is the sum of P_1 and P_2. The following claim is easy to show (just like Claim 3.1). For completeness, we prove Claim 3.5 in Section 6.3.

Claim 3.5. *Polynomial g (constructed above) is computed by a multilinear depth-three circuit C of size $\Theta(n)$ and top fan-in two, and C is a superposition of three set-multilinear depth-three circuits.*

High evaluation dimension of $g(X, Y)$. The proof of the following lemma is similar to that of Lemma 3.1, differences arise only due to the "dual" nature of g.

Lemma 3.2. *For any $S \subseteq X \uplus Y \uplus Z$ of size $n/10$, $\mathrm{Evaldim}_S(g) \geq 2^{\epsilon n}$ where $\epsilon > 0$ is a constant.*

Proof. Let S be any set of $\frac{n}{10}$ variables from $X \uplus Y \uplus Z$. The definitions of untouched, partially touched and completely touched linear polynomials are almost the same as in the proof of Lemma 3.1. The difference is we have three variables instead of two in a linear polynomial in g. So, a linear polynomial is partially touched if at most two of the three variables belong to S. For $i \in [2]$, let B_i be the set of partially touched linear polynomials and C_i the set of completely touched linear polynomials in product term P_i of g. □

Claim 3.6. *There is an $i \in [2]$ such that $|B_i| \geq \epsilon n$ where $\epsilon = 10^{-7}$.*

Proof. Let $T = \max_{i \in [2]} \{|B_i|\}$. It is easy to observe the following. □

Observation 3.2. *$|C_1| + |C_2|$ is at least $\frac{n}{15} - \frac{8T}{3}$.*

Proof. The number of variables in S that are part of partially touched linear polynomials in either of the product gates is at most $4T$; $2T$ from each product gate. Hence, at least $\frac{n}{10} - 4T$ variables in S are part of completely touched linear polynomials in each of the product gates. Since the number of variables per linear polynomial is 3, the number of completely touched linear polynomials in each of the product gates is at least $(\frac{n}{30} - \frac{4T}{3})$. Hence, $|C_1| + |C_2| \geq (\frac{n}{15} - \frac{8T}{3})$. □

Let C be the set of vertices in $\tilde{G}$ corresponding to the completely touched linear polynomials in either of the product gates, thus $|C| = |C_1| + |C_2|$ and $\frac{n}{15} - \frac{8T}{3} \leq |C| \leq \frac{n}{15}$. Each edge in $\tilde{E}(C, \overline{C})$ connects a vertex that corresponds to a completely touched linear polynomial to a vertex that corresponds to a partially touched linear polynomial. Using expansion of $\tilde{G}$,

$$|\tilde{E}(C, \overline{C})| \geq h(\tilde{G}) \cdot |C| \geq \frac{3}{2} 10^{-4} \cdot \left(\frac{n}{15} - \frac{8T}{3}\right).$$

Since edges in $\tilde{E}(C, \overline{C})$ are associated with variables in S, a vertex corresponding to a partially touched linear polynomial has at most two edges from $\tilde{E}(C, \overline{C})$ incident on it. Hence, the number of vertices corresponding to partially touched linear polynomials is at least $\frac{|\tilde{E}(C, \overline{C})|}{2}$. But, by assumption, the number of such vertices is at most $2T$. Thus,

$$2T \geq \frac{|\tilde{E}(C, \overline{C})|}{2} \geq \frac{3}{2} 10^{-4} \cdot \left(\frac{n}{15} - \frac{8T}{3}\right) \quad \Rightarrow \quad T \geq 10^{-7} n.$$

The proof of the next claim is much like that of Claim 3.4.

Claim 3.7. *If there exists an $i \in [2]$ such that $|B_i| \geq \epsilon n$ for $\epsilon > 0$, then $\mathrm{Evaldim}_S(g) \geq 2^{\epsilon n}$.*

Proof. Without loss of generality assume $|B_1| \geq \epsilon n$. Since $\tilde{G}$ is the double cover of a graph $G \in \mathcal{S}$, it is easy to argue that no two vertices in $\tilde{G}$ have all the three edges in common. Hence, the linear polynomial l is unique to a product gate, i.e., if l is a linear factor of P_2 then l is not a linear factor of P_1. Pick an untouched linear polynomial: $(x + y + z)$ in P_2 such that x is part of an untouched linear polynomial in P_1—we know there are at least $n - \frac{2n}{10} = \frac{4n}{5}$ such X-variables. By substituting $x = -(y + z)$, P_2 vanishes but P_1 remains nonzero. Let $\hat{g}$ be the polynomial we get after this substitution. $\hat{g}$ has just one product term $\hat{P_1}$ (corresponding to P_1 after substitution). $\hat{P_1}$ has as many partially touched linear polynomials as P_1. From here on a similar argument used to

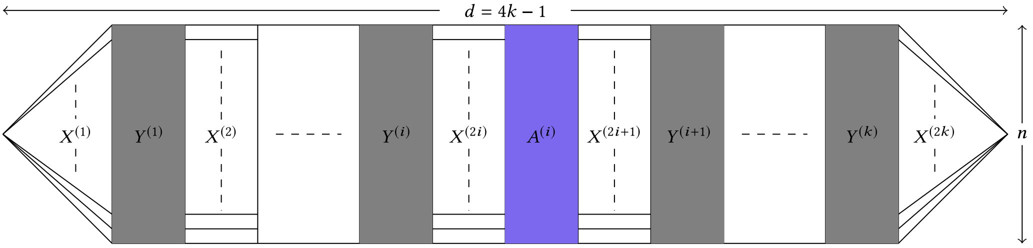

Fig. 1. ABP $\mathcal{M}$.

prove Observation 3.1 above can be used to show $\text{Evaldim}_S(g) \geq \text{Evaldim}_S(\hat{g}) = \text{Evaldim}_S(\hat{P}_1) \geq 2^{\epsilon n}$. □

This completes the proof of Lemma 3.2. From Lemmas 2.13 and 3.2, we conclude that any ROABP computing g has width at least $2^{\epsilon n}$.

4 LOWER BOUNDS FOR MULTILINEAR DEPTH THREE CIRCUITS

The proofs of Theorems 1.7 and 1.8 are inspired by a particular kind of *projection* of $\text{IMM}_{n,d}$ considered in Reference [16]. We say a polynomial f is a *simple projection* of another polynomial g if f is obtained by simply setting some variables to field constants in g.

Proof of Theorem 1.7. The proof proceeds by constructing an ABP $\mathcal{M}$ of width n and with $d + 1$ layers of vertices such that (a) the polynomial computed by $\mathcal{M}$, say f, is a simple projection of $\text{IMM}_{n,d}$, and (b) any multilinear depth-three circuit computing f has top fan-in $n^{\Omega(d)}$. Since an ABP can be viewed equivalently as a product of matrices, we will describe $\mathcal{M}$ using matrices. Figure 1 depicts the ABP $\mathcal{M}$.

Description of $\mathcal{M}$. The polynomial f, computed by $\mathcal{M}$, is defined over two disjoint sets of variables, X and Y. The Y variables are contained in k matrices, $\{Y^{(1)}, \ldots, Y^{(k)}\}$; the (u, v)th entry in $Y^{(i)}$ is a formal variable $y^{(i)}_{u,v}$. There are $(k - 1)$ matrices $\{A^{(1)}, \ldots, A^{(k-1)}\}$, such that all the entries in these matrices are ones. The X variables are contained in $2k$ matrices, $\{X^{(1)}, \ldots, X^{(2k)}\}$. Matrices $X^{(1)}$ and $X^{(2k)}$ are row and column vectors of size n, respectively. The uth entry in $X^{(1)}$ (similarly, $X^{(2k)}$) is $x^{(1)}_u$ (respectively, $x^{(2k)}_u$). All the remaining matrices $\{X^{(2)}, \ldots, X^{(2k-1)}\}$ are diagonal matrices in the X variables, i.e., the (u, u)th entry in $X^{(i)}$ is $x^{(i)}_u$ and all other entries are zero for $i \in [2, 2k - 1]$. The matrices are placed as follows: Between two adjacent Y matrices, $Y^{(i)}$ and $Y^{(i+1)}$, we have three matrices ordered from left to right as $X^{(2i)}, A^{(i)}$, and $X^{(2i+1)}$ for every $i \in [1, k - 1]$. Ordered from left to right, $X^{(1)}$ is on the left of $Y^{(1)}$ and $X^{(2k)}$ is on the right of $Y^{(k)}$. Naturally, we have the following relation among k and d: $d = 4k - 1$, i.e., $k = \frac{d+1}{4}$. Thus, $|X| = 2nk$ and $|Y| = n^2k$. This imbalance between the number of X and Y variables plays a vital role in the proof. Denote the polynomial computed by this ABP $\mathcal{M}$ as $f(X, Y)$.

The following claim is easy to verify as f is a simple projection of $\text{IMM}_{n,d}$. The proof of Claim 4.1 is given in Section 6.4.

Claim 4.1. *If $\text{IMM}_{n,d}$ is computed by a multilinear depth-three circuit having top fan-in s, then f is also computed by a multilinear depth-three circuit having top fan-in s.*

We show every multilinear depth-three circuit computing f has top fan-in $n^{\Omega(d)}$ for $n \geq 6$.

Lower bounding $\text{PD}_{\mathcal{Y}_k}(f)$. Let $\tilde{\mathcal{Y}_k} \subseteq \mathcal{Y}_k$ be the set of monomials formed by picking exactly one Y-variable from each of the matrices $Y^{(1)}, \ldots, Y^{(k)}$ and taking their product. Then, $|\tilde{\mathcal{Y}_k}| = n^{2k}$. Recall $\text{PD}_{\mathcal{Y}_k}(f)$ denotes the skewed partial derivative of f as defined in Definition 2.2.

Claim 4.2. $\text{PD}_{\mathcal{Y}_k}(f(X,Y)) = |\tilde{\mathcal{Y}_k}| = n^{2k}$.

Proof. The derivative of f with respect to a monomial $m \in \mathcal{Y}_k$ is nonzero if and only if $m \in \tilde{\mathcal{Y}_k}$. Also, such a derivative $\frac{\partial f}{\partial m}$ is a multilinear degree-r monomial in X-variables. The derivatives of f with respect to two distinct monomials m and m' in $\tilde{\mathcal{Y}_k}$ give two distinct multilinear degree-r monomials in X-variables. Hence, $\text{PD}_{\mathcal{Y}_k}(f) = |\tilde{\mathcal{Y}_k}|$. □

Upper bounding $\text{PD}_{\mathcal{Y}_k}$ of a multilinear depth-three circuit.

Lemma 4.1. *Let C be a multilinear depth-three circuit having top fan-in s computing a polynomial in X and Y variables. Then* $\text{PD}_{\mathcal{Y}_k}(C) \leq s \cdot (k+1) \cdot \binom{|X|}{k}$ *if* $k \leq \frac{|X|}{2}$.

Proof. Let $C = \sum_{i=1}^{s} T_i$, where each T_i is a product of linear polynomials on disjoint sets of variables. From Lemma 2.3, $\text{PD}_{\mathcal{Y}_k}(C) \leq s \cdot \max_{i \in [s]} \text{PD}_{\mathcal{Y}_k}(T_i)$. We need to upper bound the dimension of the "skewed" partial derivatives of a term $T_i = T$ (say). Let $T = \prod_{j=1}^{q} l_j$, where l_j is a linear polynomial. Among the q linear polynomials at most $|X|$ of them contain the X variables. Without loss of generality, assume the linear polynomials $l_1, \ldots, l_p$ contain X-variables and the remaining $l_{p+1}, \ldots, l_q$ are X-free (here $p \leq |X|$). Let $Q = \prod_{j=p+1}^{q} l_j$. Then, $T = Q \cdot \prod_{j=1}^{p} l_j$. We take the derivative of T with respect to a monomial $m \in \mathcal{Y}_k$ and then substitute the Y variables to zero. Applying the product rule of differentiation and observing that the derivative of a linear polynomial with respect to a variable makes it a constant, we have the following:

$$\left[\frac{\partial T}{\partial m}\right]_{Y=\bar{0}} = \sum_{\substack{S \subseteq [p] \\ |S| \leq k}} \alpha_S \prod_{j \in [p] \setminus S} [l_j]_{Y=\bar{0}},$$

where α_S's are constants from the field. Here, m is a representative element of the set $\mathcal{Y}_k$. Hence, every such derivative can be expressed as a linear combination of $\sum_{t=0}^{k} \binom{p}{t} \leq (k+1) \cdot \binom{|X|}{k}$ polynomials, where the last inequality is due to $k \leq \frac{|X|}{2}$ (if $t > p$, then $\binom{p}{t} \overset{\text{def}}{=} 0$). Therefore, $\text{PD}_{\mathcal{Y}_k}(T) \leq (k+1) \cdot \binom{|X|}{k}$ and $\text{PD}_{\mathcal{Y}_k}(C) \leq s \cdot (k+1) \cdot \binom{|X|}{k}$. □

It follows from Claim 4.2 and Lemma 4.1 that the top fan-in s of any multilinear depth-three circuit computing $f(X,Y)$ is such that

$$s \geq \frac{n^{2k}}{(k+1) \cdot \binom{2nk}{k}} \geq \frac{n^{2k}}{(k+1) \cdot (2ne)^k} = n^{\Omega(d)},$$

as $n \geq 6$ and $k \leq |X|/2$ (required in Lemma 4.1). Claim 4.1 now completes the proof of Theorem 1.7.

Theorem 1.7 implies the following corollary (already known due to Reference [36]) as $\text{IMM}_{n,d}$ is a simple projection of $\text{Det}_{nd \times nd}$, the determinant of an $nd \times nd$ symbolic matrix [44].

Corollary 4.2 ([36]). *Any multilinear depth-three circuit (over any field) computing* Det_d, *the determinant of a $d \times d$ symbolic matrix, has top fan-in* $2^{\Omega(d)}$.

Proof of Theorem 1.8. We now show that the polynomial $f(X,Y)$, computed by the ABP $\mathcal{M}$, can also be computed a multilinear depth-four circuit of size $O(n^2 d)$ and having top fan-in just one. ABP $\mathcal{M}$ has k matrices, $Y^{(1)}, \ldots, Y^{(k)}$, containing the Y-variables. Associate with each matrix

$Y^{(i)}$ two matrices containing the X-variables, one on the immediate left $X^{(2i-1)}$, and one on the immediate right $X^{(2i)}$. Every monomial in f is formed by picking exactly one variable from every matrix and taking their product. Once we pick $y_{u,v}^{(i)}$ from $Y^{(i)}$, this automatically fixes the variables picked from $X^{(2i-1)}$, and $X^{(2i)}$, as these are diagonal matrices. Moreover, any variable can be picked from $Y^{(i)}$ irrespective of which other Y-variables are picked from $Y^{(1)}, \ldots, Y^{(i-1)}, Y^{(i+1)}, \ldots, Y^{(k)}$. This observation can be easily formalized to show that

$$f = \prod_{i=1}^{k} \sum_{u,v \in [n]} x_u^{(2i-1)} \cdot y_{u,v}^{(i)} \cdot x_v^{(2i)} \ .$$

The size of this multilinear $\Pi\Sigma\Pi$ circuit is $O(n^2 k) = O(n^2 d)$.

5 PROOF OF THEOREM 1.9

We prove Theorem 1.9 in this section. In particular, we use the shift and rank concentration technique used in Reference [3] to give a quasi-polynomial time hitting set for a restricted class of multilinear depth-three circuits. The model we consider is a multilinear depth-three circuit that is both a superposition of m set-multilinear depth-three circuits and simultaneously a sum of l set-multilinear depth-three circuits, where m and l are constants. Before we prove Theorem 1.9, we briefly review the shift and rank concentration technique from [3].

Shift and rank concentration. Suppose we wish to check whether a polynomial computed by a set-multilinear depth-three circuit is identically zero. Let the given circuit be $C(X) = \sum_{i=1}^{s} \prod_{j=1}^{d} l_{i,j}(X_j)$, where $X = \uplus_{j=1}^{d} X_j$, $X_j = \{x_{j,1}, x_{j,2}, \ldots, x_{j,n}\}$ and $l_{i,j}$'s are linear polynomials in variables X_j. We view the polynomial C as a s coordinate vector where the ith coordintae is the polynomial computed by the ith product gate. A dot product with the all ones vector $\overline{1}$, would give us the polynomial C. In shift and rank concentration, we shift each variable $x_{j,r}$ to $x_{j,r} = x_{j,r} + t_{j,r}$, where $t_{j,r}$'s are formal variables. Let $T_j = \{t_{j,1}, t_{j,1}, \ldots, t_{j,n}\}$, $T = \uplus_{j=1}^{d} T_j$, $S \subseteq X$, $\nu_S = \prod_{x_{j,r} \in S} x_{j,r}$ and Z_{ν_S} be the coefficient vector over $\mathbb{F}(T)$ corresponding to the monomial ν_S in $C(X)$. The idea is to use a map $\tau : t_{j,r} \to t^{\omega_{j,r}}$, where t is a fresh variable different from X and T, such that

$$\mathrm{span}_{\mathbb{F}(t)}\{Z_{\nu_S} : |S| \leq \lceil \log s \rceil\} \ = \ \mathrm{span}_{\mathbb{F}(t)}\{Z_{\nu_S}\},$$

where $\mathrm{span}_{\mathbb{F}(t)}\{Z_{\nu_S}\}$ denotes the span of the coefficient vectors over $\mathbb{F}(t)$ corresponding to the different monomials in the shifted polynomial and $|S|$ equals the support[7] of the monomial ν_S. We say that such a map τ achieves $\lceil \log s \rceil$ concentration. [3] showed that it is sufficient to try $nd^{O(\log s)}$ many maps to find a map that achieves $\lceil \log s \rceil$ concentration such that the ω_{jr}'s of such a map are bounded by $(nd)^{O(\log s)}$. After such a shift using the desired map, the polynomial C is nonzero if and only if there a exists a monomial in the shifted polynomial with support less than or equal to $\lceil \log s \rceil$ and a nonzero coefficient in $\mathbb{F}(t)$. Thus, we check whether the shifted polynomial has a nonzero monomial with support less than or equal to $\lceil \log s \rceil$, by projecting over all possible choices of $\lceil \log s \rceil$ variables and test if the shifted polynomial is nonzero using Reference [27] in $(nd)^{O(\log s)}$ time. Now we prove Theorem 1.9.

Theorem 1.9 (restated). *Let $C_{n,m,l,s}$ be a subclass of multilinear depth-three circuits computing n-variate polynomials such that every circuit in $C_{n,m,l,s}$ is a superposition of at most m set-multilinear depth-three circuits and simultaneously a sum of at most l set-multilinear depth-three circuits, and has top fan-in s. There is a hitting-set generator for $C_{n,m,l,s}$ running in $(ns)^{O(lm \log s)}$ time.*

[7] Support of a monomial is the number of variables in the monomial with degree at least 1.

Proof. Circuit C is a superposition of m set-multilinear depth-three circuits in base sets $X_1, X_2, \dots, X_m$. Circuit C is also a sum of l set-multilinear depth-three circuits $C_1, C_2, \dots, C_l$ with top fan-in $s_1, s_2, \dots, s_l$, respectively, and $s_1 + s_2 + \cdots + s_l = s$. We make the following assumptions on C:

(1) For all $u \in [m]$, $|X_u| = a$ and $X_u = \{x_{u,1}, x_{u,2}, \dots, x_{u,a}\}$.
(2) Every product node in C computes a degree a polynomial in X variables.

The second assumption allows us to associate m permutations $\sigma_{k,1}, \dots, \sigma_{k,m}$ corresponding to base sets $X_1, \dots, X_m$, respectively, such that circuit C_k computes the polynomial

$$\sum_{i=1}^{s_k} \prod_{j=1}^{a} \left(\alpha_{i,j} + z^{(i)}_{1,\sigma_{k,1}(j)} x_{1,\sigma_{k,1}(j)} + z^{(i)}_{2,\sigma_{k,2}(j)} x_{2,\sigma_{k,2}(j)} + \cdots + z^{(i)}_{m,\sigma_{k,m}(j)} x_{m,\sigma_{k,m}(j)} \right),$$

where $\alpha_{i,j}, z^{(i)}_{u,\sigma_{k,u}(j)} \in \mathbb{F}$ for all $i \in [s_k]$, $u \in [m]$, and $j \in [a]$. These assumptions are without loss of generality and the arguments continue to hold in their absence. In particular, these assumptions enable us to present the main ideas of the proof clearly. We outline these ideas in brief below after we setup a few more notations. We have m sets of shift variables denoted $T_u = \{t_{u,1}, t_{u,2}, \dots, t_{u,a}\}$, for all $u \in [m]$, and $T = \uplus_{u \in [m]} T_u$. For convenience, we denote the union of the first r base sets of variables as U_r, i.e., $U_r = \uplus_{u \in [r]} X_u$, and $W_r = X \setminus U_r$.

Proof outline. The variable $x_{u,j}$ is shifted to $x_{u,j} + t_{u,j}$, and at first we argue that after this shift there is a monomial μ in X variables of support at most $m \log s$ with a nonzero coefficient in $\mathbb{F}[T]$ if and only if C computes a nonzero polynomial. Naturally, this is true for any polynomial, but the way we prove it for $C(X)$, enables us to importantly show in the second part that we can construct a map that sets $t_{u,j}$ to $t^{\omega_{u,j}}$, where t is a fresh variable and $\omega_{u,j}$ has an appropriate small value, such that after applying the map μ has a nonzero coefficient polynomial in $\mathbb{F}[t]$. We argue the first part iteratively: in the first iteration we show there is a monomial μ_1 in X_1 variables of support at most $\log s$ whose coefficient polynomial in $\mathbb{F}[W_1 \uplus T]$ is nonzero. We induct on this nonzero coefficient polynomial, which is computed by a depth-three circuit that is a superposition of $m - 1$ set-multilinear depth-three circuits and is a sum of l set-multilinear depth-three circuits. In particular, at step r, we have a polynomial in $\mathbb{F}[W_{r-1} \uplus T]$, that is computed by a depth-three circuit that is a superposition of $m - (r - 1)$ set-multilinear depth-three circuits and a sum of l set-multilinear depth-three circuits. Such a polynomial we show has a monomial μ_r in X_r variables of support at most $\log s$ whose coefficient polynomial in $\mathbb{F}[W_r \uplus T]$ is nonzero. Thus, at the end of step m, the product of the monomials $\mu = \prod_{r=1}^{m} \mu_r$ has support at most $m \log s$ and a nonzero coefficient polynomial in $\mathbb{F}[T]$. The next part of the proof is the most important, here we argue that we can efficiently construct a map that sets $t_{u,j}$ to $t^{\omega_{u,j}}$, where t is a fresh variable and $\omega_{u,j}$ is bounded by $(ns)^{O(lm \log s)}$, such that after applying the map, at every step r, μ_r has a nonzero coefficient over $\mathbb{F}[t, W_r]$, and hence μ has a nonzero coefficient over $\mathbb{F}[t]$. Once we show this, finding a hitting set is easy: project over all possible choices of $(m \log s)$ variables and test if the shifted polynomial is nonzero over $\mathbb{F}[t]$ using sparse PIT [27].

Part 1: The polynomial computed by C after shifting the variables $x_{u,j}$ to $x_{u,j} + t_{u,j}$, for all $u \in [m]$ and $j \in [a]$ is denoted as $C(X \uplus T)$. We argue inductively that $C(X \uplus T)$ when viewed as a polynomial over $\mathbb{F}[T]$ has a monomial μ of support at most $m \log s$ with a nonzero coefficient over $\mathbb{F}[T]$. We present the inductive step here, the base case can be argued similarly. At step r, we are ensured that there are monomials $\mu_1, \mu_2, \dots, \mu_{r-1}$ in variables $X_1, X_2, \dots, X_{r-1}$, respectively, each having support at most $\log s$ such that $\prod_{i=1}^{r-1} \mu_i$ has a nonzero coefficient over $\mathbb{F}[W_{r-1} \uplus T]$. An easy to see observation is that such a coefficient, when viewed as a polynomial over $\mathbb{F}[T]$

is computed by a circuit $C^{(r)}$ that is a superposition of $m-(r-1)$ set-multilinear depth-three circuits in base sets $X_r, \ldots, X_m$, and is a sum of l set-multilinear depth-three circuits. For $k \in [l]$, the k-th set-multilinear depth-three circuit, denoted $C_k^{(r)}$, is the circuit computing the coefficient of the monomial $\prod_{i=1}^{r-1} \mu_i$ in C_k. Without loss of generality and reusing symbols, the polynomial $C_k^{(r)}(W_{r-1} \uplus T)$ computed by circuit $C_k^{(r)}$ can be represented as

$$\sum_{i=1}^{s_k} \prod_{\substack{j=1+\\(r-1)\log s}}^{a} \Big(\alpha_{i,j} + z^{(i)}_{1,\sigma_{k,1}(j)}(t_{1,\sigma_{k,1}(j)}) + \cdots + z^{(i)}_{r-1,\sigma_{k,r-1}(j)}(t_{r-1,\sigma_{k,r-1}(j)})$$

$$+z^{(i)}_{r,\sigma_{k,r}(j)}(x_{r,\sigma_{k,r}(j)} + t_{r,\sigma_{k,r}(j)}) + \cdots + z^{(i)}_{m,\sigma_{k,m}(j)}(x_{m,\sigma_{k,m}(j)} + t_{m,\sigma_{k,m}(j)})\Big).$$

Without loss of generality, we may assume for all $k \in [l]$ $\sigma_{k,r}$ is the identity permutation. Further for convenience of notation, in each product gate we can include the first $(r-1)\log s$ linear polynomials by assuming for all $k \in [l]$ $i \in [s_k]$, $u \in [m]$ and $j \in [(r-1)\log s]$ $z^{(i)}_{u,\sigma_{k,u}(j)} = 0$ and $\alpha_{i,j} = 1$. Also using the same argument as in [3], we may also assume for all $k \in [l]$ $i \in [s_k]$, $u \in [m]$ and $j \in [a] \setminus [(r-1)\log s]$ $\alpha_{i,j} = 1$. Thus, reusing symbols, we may represent $C_k^{(r)}(W_{r-1} \uplus T)$ as

$$\sum_{i=1}^{s_k} \prod_{j=1}^{a} \Big(1 + z^{(i)}_{1,\sigma_{k,1}(j)} t_{1,\sigma_{k,1}(j)} + \cdots + z^{(i)}_{r-1,\sigma_{k,r-1}(j)} t_{r-1,\sigma_{k,r-1}(j)}$$

$$+ z^{(i)}_{r,\sigma_{k,r}(j)}(x_{r,\sigma_{k,r}(j)} + t_{r,\sigma_{k,r}(j)}) + \cdots + z^{(i)}_{m,\sigma_{k,m}(j)}(x_{m,\sigma_{k,m}(j)} + t_{m,\sigma_{k,m}(j)})\Big).$$

For convenience, we define $\rho_{i,j}$ as

$$z^{(i)}_{1,\sigma_{k,1}(j)} t_{1,\sigma_{k,1}(j)} + \cdots + z^{(i)}_{r-1,\sigma_{k,r-1}(j)} t_{r-1,\sigma_{k,r-1}(j)} + z^{(i)}_{r+1,,\sigma_{k,r+1}(j)}(x_{r+1,\sigma_{k,r+1}(j)}$$

$$+ t_{r+1,\sigma_{k,r+1}(j)}) + \cdots + z^{(i)}_{m,\sigma_{k,m}(j)}(x_{m,\sigma_{k,m}(j)} + t_{m,\sigma_{k,m}(j)}),$$

and hence we express $C_k^{(r)}(W_{r-1} \uplus T)$ as

$$\sum_{i=1}^{s_k} \prod_{j=1}^{a} \Big(1 + \rho_{i,j} + z^{(i)}_{r,j}(x_{r,j} + t_{r,j})\Big). \tag{1}$$

We view the polynomial $C^{(r)}(W_{r-1} \uplus T) = \sum_{k=1}^{l} C_k^{(r)}(W_{r-1} \uplus T)$ as a polynomial in X_r variables over the function field $\mathbb{F}(W_r \uplus T)$, and we will prove that there is monomial μ_r in X_r variables of support at most $\log s$ with a nonzero coefficient polynomial in $\mathbb{F}[W_r \uplus T]$. To this end, we rewrite Equation (1) as follows:

$$C_k^{(r)}(W_{r-1} \uplus T) = \sum_{i=1}^{s_k} \left(\prod_{j=1}^{a} (1 + \rho_{i,j} + z^{(i)}_{r,j} t_{r,j}) \cdot \prod_{j=1}^{a} \left(1 + \frac{z^{(i)}_{r,j} x_{r,j}}{1 + \rho_{i,j} + z^{(i)}_{r,j} t_{r,j}}\right)\right).$$

Let

$$\tilde{z}^{(i)}_{r,j} = \frac{z^{(i)}_{r,j}}{1 + \rho_{i,j} + z^{(i)}_{r,j} t_{r,j}} \quad \Rightarrow \quad z^{(i)}_{r,j} = \frac{\tilde{z}^{(i)}_{r,j}(1 + \rho_{i,j})}{1 - \tilde{z}^{(i)}_{r,j} t_{r,j}}. \tag{2}$$

Then,

$$C_k^{(r)}(W_{r-1} \uplus T) = \sum_{i=1}^{s_k} \left(\prod_{j=1}^{a} \Big(1 + \rho_{i,j} + z^{(i)}_{r,j} t_{r,j}\Big) \cdot \prod_{j=1}^{a} \Big(1 + \tilde{z}^{(i)}_{r,j} x_{r,j}\Big)\right).$$

Recall, circuit $C^{(r)}$ is a depth-three circuit with top fan-in s, and is the sum of l set-multilinear depth-three circuits $C_1^{(r)}, \ldots, C_l^{(r)}$ with top fan-in $s_1, \ldots, s_l$, respectively, where $s_1 + s_2 + \cdots + s_l = s$. In the remaining part of the proof, we use the index i to either refer to a product gate in circuit $C^{(r)}$, in which case $i \in [s]$, or to a product gate in circuit $C_k^{(r)}$, in which case $i \in [s_k]$, for all $k \in [l]$. This will be clear from the context.

For a set $J \subset [a]$, we associate the monomial $\nu_J = \prod_{j \in J} x_{r,j}$ with J, and with every monomial ν_J, we associate two vectors $Z_{\nu_J}, \tilde{Z}_{\nu_J} \in \mathbb{F}(W_r \uplus T)^s$. For all $i \in [s]$, the i-th entry of Z_{ν_J} is equal to $(\prod_{j \in J} z_{r,j}^{(i)})(\prod_{j \in [a] \setminus J}(1 + \rho_{i,j}))$, and the ith entry of $\tilde{Z}_{\nu_J}$ is $\prod_{j \in J} \tilde{z}_{r,j}^{(i)}$. Since the coefficient of a monomial ν_J in $C^{(r)}(W_{r-1} \uplus T)$, for all $J \subseteq [a]$, is the dot product of $\tilde{Z}_{\nu_J}$ with another vector whose ith component is $\prod_{j=1}^{a}(1 + \rho_{i,j} + z_{r,j}^{(i)} t_{r,j})$, for all $i \in [s]$, the following claim implies that if $C^{(r)}(W_{r-1} \uplus T) \neq 0$ then $C^{(r)}(W_{r-1} \uplus T)$ has a monomial μ_r in X_r variables of support at most $\log s$ with a nonzero coefficient polynomial in $\mathbb{F}[W_r \uplus T]$. □

Claim 5.1. *$span_{\mathbb{F}(W_r \uplus T)}\{\tilde{Z}_{\nu_J} | J \subseteq [a]\} = span_{\mathbb{F}(W_r \uplus T)}\{\tilde{Z}_{\nu_J} | J \subseteq [a], |J| \leq \log s\}$.*

Proof. Pick a monomial ν_J, such that $|J| = \log s + 1$[8], and consider all monomial ν_I such that $I \subseteq J$. Since there are $2^{\log s + 1} > s$ many monomials, the coefficient vectors of these monomials are $\mathbb{F}(W_r \uplus T)$-linearly dependent, i.e.,

$$\sum_{I \subseteq J} b_I Z_{\nu_I} = 0, \tag{3}$$

where for all $I \subseteq J$, $b_I \in \mathbb{F}(W_r \uplus T)$ and there is a $I \subseteq J$ such that $b_I \neq 0$. We write Equation (3) corresponding to the i-th entry of the vectors Z_{ν_I}, where $i \in [s]$, and $I \subseteq J$

$$\sum_{I \subseteq J} b_I \Big(\prod_{j \in I} z_{r,j}^{(i)} \prod_{j \in [a] \setminus I} (1 + \rho_{i,j}) \Big) = 0. \tag{4}$$

Since $\prod_{j \in [a] \setminus J}(1 + \rho_{i,j}) \neq 0$, we have

$$\sum_{I \subseteq J} b_I \Big(\prod_{j \in I} z_{r,j}^{(i)} \prod_{j \in J \setminus I} (1 + \rho_{i,j}) \Big) = 0. \tag{5}$$

In the above equation, we substitute the value of $z_{r,j}^{(i)}$ from Equation (2),

$$\sum_{I \subseteq J} b_I \Bigg(\prod_{j \in I} \Big(\frac{\tilde{z}_{r,j}^{(i)}(1 + \rho_{i,j})}{1 - \tilde{z}_{r,j}^{(i)} t_{r,j}} \Big) \prod_{j \in J \setminus I} (1 + \rho_{i,j}) \Bigg) = 0\,,$$

$$\Rightarrow \quad \sum_{I \subseteq J} b_I \Bigg(\prod_{j \in I} \Big(\frac{\tilde{z}_{r,j}^{(i)}}{1 - \tilde{z}_{r,j}^{(i)} t_{r,j}} \Big) \prod_{j \in J} (1 + \rho_{i,j}) \Bigg) = 0\,.$$

Again, as $\prod_{j \in J}(1 + \rho_{i,j}) \neq 0$, we have

$$\sum_{I \subseteq J} b_I \Bigg(\prod_{j \in I} \frac{\tilde{z}_{r,j}^{(i)}}{1 - \tilde{z}_{r,j}^{(i)} t_{r,j}} \Bigg) = 0\,.$$

Multiplying both sides of the above equation by $\prod_{j \in J}(1 - \tilde{z}_{r,j}^{(i)} t_{r,j})$, we have

$$\sum_{I \subseteq J} b_I \prod_{j \in I} \tilde{z}_{r,j}^{(i)} \prod_{j \in J \setminus I} \Big(1 - \tilde{z}_{r,j}^{(i)} t_{r,j}\Big) = 0. \tag{6}$$

[8] We avoid ceil, floor notation for ease of exposition.

Since this is true for all $i \in [s]$, from Equation (6), we get the following relation among the vectors $\tilde{Z}_{v_I}$, for all $I \subseteq J$:

$$\underbrace{\left(\sum_{I \subseteq J} b_I \cdot (-1)^{\log s + 1 - |I|} \prod_{j \in J \setminus I} t_{r,j}\right)}_{g_J(W_r \uplus T)} \cdot \tilde{Z}_{v_J} + \sum_{I \subset J} g_I(W_r \uplus T) \cdot \tilde{Z}_{v_I} = 0. \tag{7}$$

Since among all $b_I \in \mathbb{F}(W_r \uplus T)$, where $I \subseteq J$, there is a nonzero b_I, $g_J(W_r \uplus T)$ is nonzero in the above equation. Thus, we conclude $\tilde{Z}_{v_J}$ is $\mathbb{F}(W_r \uplus T)$-linearly dependent on vectors $\tilde{Z}_{v_I}$ for $I \subset J$. Similarly, for every $I \subseteq [a]$, $\tilde{Z}_{v_I}$ can be inductively expressed as an $\mathbb{F}(W_r \uplus T)$-linear combination of the vectors $\tilde{Z}_{v_K}$, where $K \subseteq [a]$ and $|K| \leq \log s$. □

Part 2: In part 1, we iteratively showed that there is a monomial $\mu = \prod_{i \in [m]} \mu_i$, with a nonzero coefficient over $\mathbb{F}[T]$. Corresponding to every iteration $r \in [m]$ in part 1, we have an equation similar to Equation (7), and the correctness of the proof hinges on the coefficient of the $\log s + 1$ support monomial (i.e., $g_J(W_r \uplus T)$) being nonzero in these equations. In this part, we analyze the structure of $g_J(W_r \uplus T)$ for all $r \in [m]$, and $J \subseteq [a]$ such that $|J| = \log s + 1$, to argue that it is possible to construct a map ψ in time $(ns)^{O(lm \log s)}$ that sets $t_{u,j}$ to $t^{\omega_{u,j}}$, where $\omega_{u,j}$'s are also bounded by $(ns)^{O(lm \log s)}$, such that $g_J(t, W_r) \in \mathbb{F}(t, W_r)$ remains nonzero after applying this map. It follows immediately that after applying ψ the monomial μ has a nonzero coefficient over $\mathbb{F}[t]$. We begin by proving the following claim.

Claim 5.2. *The expression denoted $g_J(W_r \uplus T)$ in Equation (7) is a rational function in $\mathbb{F}(W_r \uplus T)$ with at most $O(lm \log s)$ distinct T variables appearing in it. Further, the degree of the polynomials in the numerator and denominator of $g_J(W_r \uplus T)$ is bounded by $O(s^2 \log s)$.*

Proof. Recall Equations (3), (4), and (5). We rewrite Equation (5) below

$$\sum_{I \subseteq J} b_I \left(\prod_{j \in I} z_{r,j}^{(i)} \prod_{j \in J \setminus I} (1 + \rho_{i,j}) \right) = 0. \tag{8}$$

Define vectors $\overline{Z}_{v_I} \in \mathbb{F}[W_r \uplus T]^s$, for all $I \subseteq J$, such that the ith entry of $\overline{Z}_{v_I}$ is $\prod_{j \in I} z_{r,j}^{(i)} \prod_{j \in J \setminus I}(1 + \rho_{i,j})$. Then,

$$\sum_{I \subseteq J} b_I \overline{Z}_{v_I} = 0. \tag{9}$$

Since for all $i \in [s]$ and $j \in [J]$, $\rho_{i,j}$ is a linear polynomial in $2(m-1)$ variables from $W_r \uplus T$, the expression $\prod_{j \in J \setminus I}(1 + \rho_{i,j})$ is a polynomial in $O(m \log s)$ variables from $W_r \uplus T$. We call the variables appearing in the expression $\prod_{j \in J \setminus I}(1 + \rho_{i,j})$ as the variable set of the expression. Recall that circuit $C^{(r)}$ is a sum of l set-multilinear depth-three circuits, and if two expressions correspond to product gates from the same set-multilinear depth-three circuit then they have the same variable set, i.e., for all $k \in [l]$, and $i, i' \in [s_k]$, the expressions $\prod_{j \in J \setminus I}(1 + \rho_{i,j})$ and $\prod_{j \in J \setminus I}(1 + \rho_{i',j})$ have the same variable set of size $O(m \log s)$. Hence, there is a set of variables $S \subset W_r \uplus T$ of size $O(lm \log s)$ such that the variable sets of the entries in $\overline{Z}_{v_I}$ for all $I \subseteq J$ are subsets of S. From Cramer's rule, we infer that b_I, for all $I \subseteq J$ is a rational function in the variables from S, and the degree of the numerator/denominator in b_I is $O(s \log s)$. Thus, $g_J(W_r \uplus T)$ is also a rational function in $O(lm \log s)$ variables from $W_r \uplus T$ and the degree of the polynomials in the numerator/denominator in $g_J(W_r \uplus T)$ is $O(s^2 \log s)$. □

From Claim 5.2 it follows that the number of monomials in T variables in the numerator and denominator of $g_J(W_r \uplus T)$ is equal to $s^{O(lm \log s)}$. Thus, the number of monomials in T variables in the numerator/denominator of $g_J(W_r \uplus T)$, at every iteration $r \in [m]$, and for all $J \subset [a]$ of size $\log s + 1$ is $(ns)^{O(lm \log s)}$. The map ψ maps these $(ns)^{O(lm \log s)}$ monomials in T variables to distinct monomials in $\mathbb{F}[t]$, and it is standard to compute such a map $\psi(t_{u,j}) = t^{\omega_{u,j}}$ in $(ns)^{O(lm \log s)}$ time such that the values of $\omega_{u,j}$ are also bounded by $(ns)^{O(lm \log s)}$ [27].

6 PROOF OF TECHNICAL CLAIMS

6.1 Proofs of Observations in Section 1

Observation 1.1 (restated). *Given a circuit C, if C is a superposition of t set-multilinear circuits on* <u>*unknown*</u> *base sets $Y_1, Y_2, \ldots, Y_t$, finding t base sets $Y_1', Y_2', \ldots, Y_t'$ such that C is a superposition of t set-multilinear circuits on base sets $Y_1', Y_2', \ldots, Y_t'$ is NP-hard when $t > 2$.*

Proof. We will reduce the t-coloring problem to this problem. Given a graph G, the t-coloring problem asks to color the vertices of G with t colors such that no two adjacent vertices of G have the same color. Suppose we are given a graph $G = (V, E)$. From G construct a circuit C as follows. Let $V = \{u_1, \ldots, u_n\}$, identify these vertices with n-variables. The circuit C contains a product gate P multiplying the n variables $(u_1) \cdots (u_n)$. If there exists an edge between two vertices u_1 and u_2 in G, then add a product gate P_{u_1,u_2} in C having a single linear polynomial $(u_1 + u_2)$. We argue below that the circuit C is a superposition of t set-multilinear depth-three circuits if and only if the graph G is t-colorable.

Suppose G is t-colorable. Then the set of vertices in G with the same color form a valid base set. Thus, the t sets of vertices of G with t different colors correspond to t valid base sets in C and thus C is a superposition of t set-multilinear depth-three circuits. In the reverse direction, say C is a superposition of t set-multilinear depth-three circuits, which implies C has t base sets. A t-coloring of G can be obtained by giving every base set a unique color. If two variables u_i and u_j belong to the same base set, then as u_i and u_j appear in different linear polynomials in product gate P there is no product gate in C in which u_i and u_j appear in the same linear polynomial. But this implies there is no edge between u_i and u_j in G else there would have been a product gate P_{u_i,u_j} having a single linear polynomial $(u_i + u_j)$ in C. □

Observation 1.2 (restated). *A polynomial computed by a multilinear $\Sigma\Pi\Sigma$ circuit with top fan-in two and at most two variables per linear polynomial can also be computed by an ROABP with constant width.*

Proof. Let C be a multilinear depth-three circuit with top fan-in two and at most two variables per linear polynomial computing the polynomial $f(X)$ in n variables $\{x_1, \ldots x_n\}$. Let $\sigma : [n] \to [n]$ be a permutation function. Then without loss of generality $f(X)$ can be expressed as

$$f(X) = \prod_{i \in [n], i \text{ odd}} (1 + x_i + x_{i+1}) + \prod_{i \in [n], i \text{ odd}} (1 + x_{\sigma(i)} + x_{\sigma(i+1)}).$$

We have assumed that the coefficients of x_i's and the constant term in every linear polynomial is 1, and n is even for simplicity. The arguments for this case can be adapted appropriately to prove it for the general case. Let $P_1 = \prod_{i \in [n], i \text{ odd}} (1 + x_i + x_{i+1})$ and $P_2 = \prod_{i \in [n], i \text{ odd}} (1 + x_{\sigma(i)} + x_{\sigma(i+1)})$. Product gates P_1 and P_2 can be easily computed individually by ROABPs of width two but with different variable orderings. We express the two ROABPs in the same variable ordering and add the polynomials computed by them (P_1 and P_2) to get an ROABP computing f.

We partition the linear polynomials in P_1 and P_2 into sets $\{L_{11}, L_{12}, \ldots, L_{1k}\}$ and $\{L_{21}, L_{22}, \ldots, L_{2k}\}$, respectively, such that the sets of variables appearing in the linear polynomials in L_{1t} and L_{2t}, where $t \in [k]$, are equal and this set is completely disjoint from the set of variables appearing in the linear polynomials in L_{mr}, where $m \in [2]$ and $r \in [k] \setminus \{t\}$. We give a "greedy"

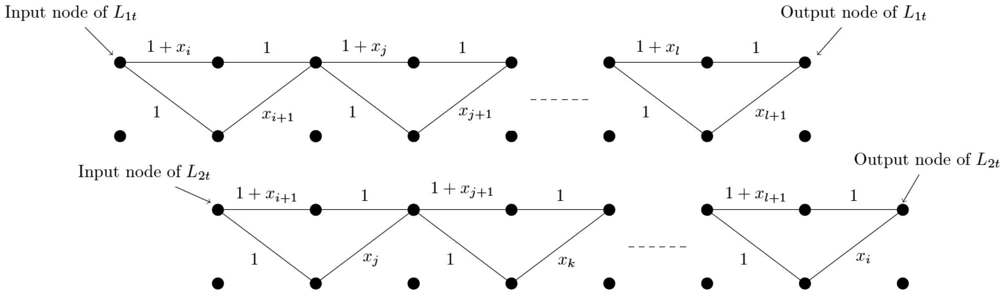

Fig. 2. ROABPs corresponding to L_{1t} and L_{2t}.

partition procedure below. Mark all the linear polynomials in P_1 and P_2 as unpicked. Initialize $t = 1$ and $i = 1$:

(1) Pick an unpicked linear polynomial $l_p = (1 + x_i + x_{i+1})$ in P_1 and put it in L_{1t}. Mark l_p as picked. Store the value i in temp: temp $= i$.
(2) Let the linear polynomial in which the variable x_{i+1} appears in P_2 be $l_q = (1 + x_{i+1} + x_j)$. Put l_q in L_{2t} and mark l_q as picked.
(3) If j is equal to temp, then increment t and start from step 1.
(4) Else set $i = j$ and let the linear polynomial in which the variable x_i appears in P_1 be $l_r = (1 + x_i + x_{i+1})$. Put l_r in L_{1t} and mark l_r as picked.
(5) Repeat from step 2.

Clearly, the sets of variables appearing in the linear polynomials in L_{1t} and L_{2t}, where $t \in [k]$, are equal and this set is disjoint from the set of variables appearing in the linear polynomials in L_{mr}, for $m \in [2]$ and $r \in [k] \setminus \{t\}$. Notice that if some of the coefficients of the variables in the linear polynomial were zero (instead of 1 as in the assumption made by us) or some of the linear polynomials involved just a single variable then the sets of variables appearing in the linear polynomials in L_{1t} and L_{2t} may not be the same but their union will still be completely disjoint from the set of variables appearing in the linear polynomials in L_{mr}, where $m \in [2]$ and $r \in [k] \setminus \{t\}$. Also, the above partition procedure can be changed appropriately to handle these cases.

We express the two ROABPs computing P_1 and P_2 in the same variable ordering as a sequence of k parts. In part t, we compute the product of linear polynomials in L_{1t} and L_{2t} separately using two ROABPs such that the variable orderings in these two ROABPs are the same. Finally, we connect the ROABPs from these k parts to give a single ROABP of width six. We argue how to construct an ROABP corresponding to the linear polynomials in L_{1t} and L_{2t}. Arrange the linear polynomials in L_{1t} and L_{2t} in the order they are picked during the partition process. Suppose after this arrangement we have $L_{1t} = \{(1 + x_i + x_{i+1}), (1 + x_j + x_{j+1}), \ldots, (1 + x_l + x_{l+1})\}$ and $L_{2t} = \{(1 + x_{i+1} + x_j), (1 + x_{j+1} + x_k), \ldots, (1 + x_{l+1} + x_i)\}$. Figure 2 shows the two ROABPs computing the product of linear polynomials in L_{1t} and L_{2t}, respectively. Consider the input and output nodes of L_{1t} and L_{2t} marked in Figure 2 as the sources and sinks of these two ROABPs, respectively. The variables are arranged such that except x_i all variables are in the same order in the two ROABPs. We order x_i by breaking the second ROABP in two parts as shown in Figure 3. The first part computes the polynomial in which x_i does not appear and the second part brings x_i to the beginning and computes the polynomial in which x_i appears. Finally, we add these two parts

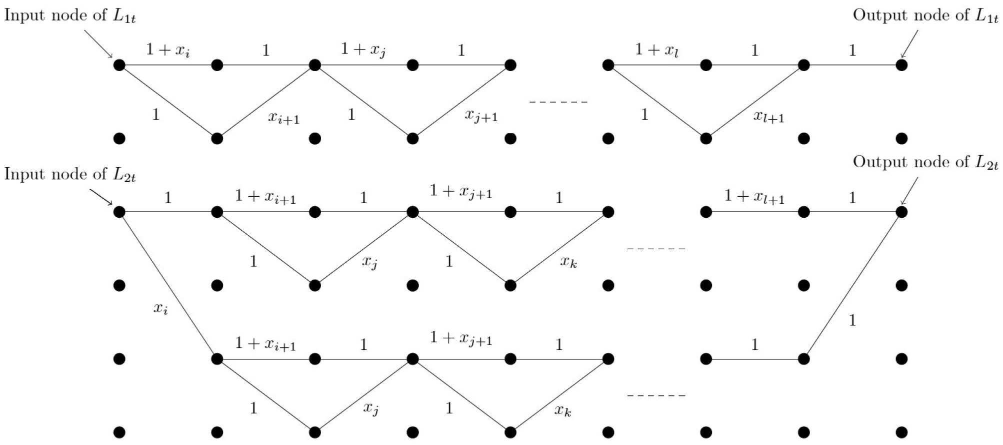

Fig. 3. ROABPs (with same variable ordering) corresponding to L_{1t} and L_{2t}.

by adding an extra layer.[9] In a general case where some of the coefficients of the variables in the linear polynomial are zero or some of the linear polynomials involved just a single variable, the variables in the linear polynomials in L_{1t} and L_{2t} may define a path instead of a cycle. But notice that handling this case is easier, as both the ROABPs can be expressed in the same variable ordering to begin with. Finally, we have directed acyclic graphs each consisting of two ROABPs with consistent variable ordering from all the k pairs of sets of linear polynomials. We connect these k graphs by adding weight 1 edges between the input nodes of L_{1r}, L_{2r} and the output nodes of $L_{1(r+1)}, L_{2(r+1)}$, respectively, where $r \in [k-1]$. The resulting graph is an ROABP of width six computing f. □

Observation 1.3 (restated). *$\text{IMM}_{n,d}$ can be computed by an n^2 width ROABP.*

Proof. We transform the width n ABP computing $\text{IMM}_{n,d}$ to a width n^2 ROABP computing the same. Let $\{X^{(1)}, X^{(2)}, \ldots, X^{(d)}\}$ be the d matrices in $\text{IMM}_{n,d}$. The (j,k)-th entry in $X^{(i)}$ is $x^{(i)}_{j,k}$. We replace matrix $X^{(i)}$ by n^2+2 matrices: $A^{(i,1)}$, $A^{(i,2)}$ and $A^{(i,j,k)}$ where $j,k \in [n]$. $A^{(i,1)}$ and $A^{(i,2)}$ are rectangular matrices of dimension $n \times n^2$ and $n^2 \times n$, respectively. For $j,k \in [n]$, $A^{(i,j,k)}$ are diagonal matrices of dimension n^2. Ordered from left to right $A^{(i,1)}$ and $A^{(i,2)}$ are first and last, respectively, and $A^{(i,j_1,k_1)}$ comes before $A^{(i,j_2,k_2)}$ if $j_1 < j_2$ or if $j_1 = j_2$ and $k_1 < k_2$. The (a,a)-th entry of $A^{(i,j,k)}$ is $x^{(i)}_{j,k}$ if $a = n \cdot (j-1) + k$ and 1 otherwise. The (a,b)th entry of $A^{(i,1)}$ is 1 if $(a-1) \cdot n + 1 \le b \le a \cdot n$ and 0 otherwise. Similarly, the (a,b)th entry of $A^{(i,2)}$ is 1 if $b \equiv a \mod n$ and 0 otherwise. Figure 4 shows the part of ROABP corresponding to the split of a matrix X into n^2+2 matrices, when $n=4$, as explained above. When we split $X^{(i)}$ into n^2+2 matrices as above the corresponding part of ROABP computing the product of these n^2+2 matrices has n vertices in both the leftmost and rightmost layers of vertices. There is a unique path from the jth vertex in leftmost layer to the kth vertex in rightmost layer with weight $x^{(i)}_{j,k}$. Hence, the product of the n^2+2 matrices arranged as above is $X^{(i)}$.

To transform the ABP computing $\text{IMM}_{n,d}$ to an ROABP, we have introduced between every pair of adjacent layers of vertices in the ABP, n^2 layers with n^2 vertices in each layer, hence the width of the ROABP is n^2. □

[9] If every edge between two layers are labelled by constants, then these edges can be absorbed by the edges in the adjacent layer. For example, in Figure 3 the edges between the last two layers can be absorbed by edges labelled by x_{l+1}.

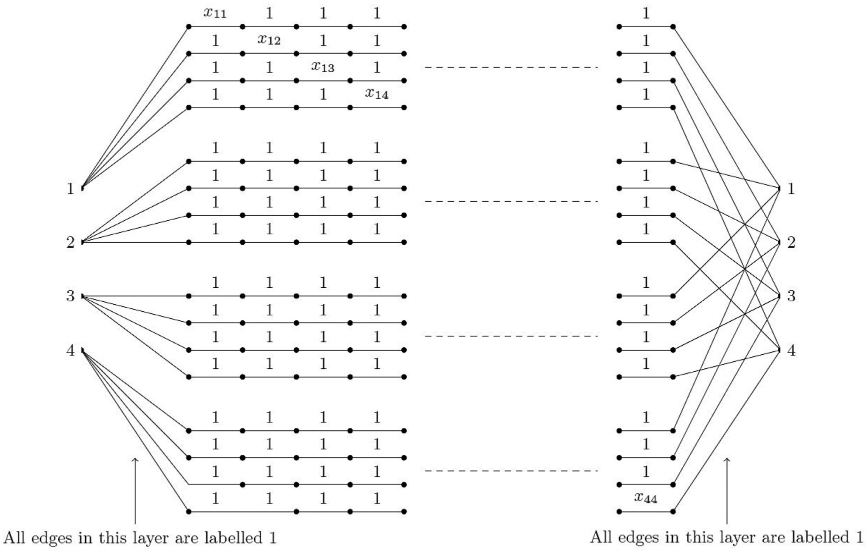
x_{11}
x_{12}
x_{13}
x_{14}
x_{44}
1
2
3
4
All edges in this layer are labelled 1
All edges in this layer are labelled 1

Fig. 4. ROABP corresponding to the split of a matrix X.

6.2 Proofs of Lemmas in Section 2

Lemma 2.3 (restated).

(1) *Let* $g_1, g_2 \in \mathbb{F}[X]$ *and* $S \subseteq X$, *then* $\text{Evaldim}_S(g_1 + g_2) \leq \text{Evaldim}_S(g_1) + \text{Evaldim}_S(g_2)$.
(2) *Let* $f_1, f_2 \in \mathbb{F}[X, Y]$, *then* $\text{PD}_{\mathcal{Y}_k}(f_1 + f_2) \leq \text{PD}_{\mathcal{Y}_k}(f_1) + \text{PD}_{\mathcal{Y}_k}(f_2)$.

Proof. For $i \in \{1, 2\}$, let

$$\begin{aligned} V_i &= \text{span}_{\mathbb{F}}\{g_i(X)|_{\forall x_j \in S\, x_j = \alpha_j} : \forall x_j \in S\, \alpha_j \in \mathbb{F}\} \text{ and} \\ W &= \text{span}_{\mathbb{F}}\{(g_1 + g_2)(X)|_{\forall x_j \in S\, x_j = \alpha_j} : \forall x_j \in S\, \alpha_j \in \mathbb{F}\}. \end{aligned}$$

Every polynomial in W belongs to $V_1 + V_2$, where $V_1 + V_2 = \{f_1 + f_2 | f_1 \in V_1, f_2 \in V_2\}$. Hence, $\text{Evaldim}_S(g_1 + g_2) = \dim(W) \leq \dim(V_1 + V_2) \leq \dim(V_1) + \dim(V_2) = \text{Evaldim}_S(g_1) + \text{Evaldim}_S(g_2)$. Proving part two is similar to part one. For $i \in \{1, 2\}$, let

$$\begin{aligned} A_i &= \text{span}_{\mathbb{F}}\left\{\left[\frac{\partial f_i(X, Y)}{\partial m}\right]_{\forall y \in Y\, y=0} : m \in \mathcal{Y}_k\right\} \text{ and} \\ B &= \text{span}_{\mathbb{F}}\left\{\left[\frac{\partial (f_1 + f_2)(X, Y)}{\partial m}\right]_{\forall y \in Y\, y=0} : m \in \mathcal{Y}_k\right\}. \end{aligned}$$

Again observing B is a subspace of $A_1 + A_2$, where $A_1 + A_2 = \{g_1 + g_2 | g_1 \in A_1, g_2 \in A_2\}$, part two follows. □

6.3 Proofs of Observations and Claims in Section 3

Claim 3.1 (restated). *Polynomial* f *(as constructed in Section 3, proof of part 1) is computed by a multilinear depth-three circuit* C *of size* $\Theta(n)$ *and top fan-in three, and* C *is a superposition of two set-multilinear depth-three circuits.*

Proof. Since f is a sum of three product terms, where each product term is a product of linear polynomials on disjoint sets of variables, it can be computed by a multilinear depth-three circuit C

with top fan-in three. The bottom fan-in (fan-in of the sum gates at layer 3) is three, since there are two variables and the field constant 1 per linear polynomial. The fan-in of every product gate is n. As there are three product gates, the total number of edges in C is $3 + 3(n(1 + 3)) = 3 + 12n = \Theta(n)$. Every linear polynomial of a product gate has two variables, an X and a Y variable. Hence, the circuit is a superposition of two set-multilinear depth-three circuits on base sets X and Y. □

Claim 3.5 (restated). *Polynomial g (as constructed in Section 3, proof of part 2) is computed by a multilinear depth-three circuit C of size $\Theta(n)$ and top fan-in two, and C is a superposition of three set-multilinear depth-three circuits.*

Proof. Since g is a sum of two product terms, where each product term is a product of linear polynomials on disjoint sets of variables, it can be computed by a multilinear depth-three circuit C with top fan-in two. From 2.11, we know a 3-regular bipartite graph can be split into three edge disjoint perfect matchings. Each linear polynomial in g contains three variables corresponding to edges from three edge disjoint perfect matchings. We group the variables corresponding to edges in a single matching into a base set. Thus, the variables are split into three distinct base sets, X, Y, and Z. Hence, g can be computed by a circuit C, which is a superposition of three set-multilinear depth-three circuits. C has two product gates and each product gate has n linear polynomials, where each linear polynomial has three variables and a constant. Hence, $|C| = 1 + 2 + 2(n(1 + 4)) = 3 + 10n = \Theta(n)$. □

6.4 Proof of Claim in Section 4

Claim 4.1 (restated). *If $\text{IMM}_{n,d}$ is computed by a multilinear depth-three circuit having top fan-in s, then f is also computed by a multilinear depth-three circuit having top fan-in s.*

Proof. f is computed by the ABP $\mathcal{M}$ of width n and length d as described in Section 4. Each edge in $\mathcal{M}$ is labelled by a distinct variable or 1. Let $\text{IMM}_{n,d}$ be the $(1, 1)$th entry of a product of d $n \times n$ symbolic matrices $\{Z^{(1)}, Z^{(2)}, \dots, Z^{(d)}\}$ ordered from left to right. The (j, k)th entry in $Z^{(i)}$ is the formal variable $z^{(i)}_{j,k}$. We project $\text{IMM}_{n,d}$ to f as follows. Recall $\mathcal{M}$ has three kinds of matrices: X, Y and A. The matrices $\{Z^{(1)}, Z^{(2)}, \dots, Z^{(d)}\}$ would correspond to the X, Y and A matrices in the same order as they appear in the ABP $\mathcal{M}$. $Z^{(1)}$ corresponds to the row vector $X^{(1)}$, so $z^{(1)}_{1,l}$ maps to $x^{(1)}_l$ and $z^{(1)}_{m,l}$ to 0 for $m \in [2, n]$. Similarly, $Z^{(d)}$ corresponds to the column vector $X^{(r)}$, so $z^{(d)}_{m,1}$ maps to $x^{(r)}_m$ and $z^{(d)}_{m,l}$ to 0 for $l \in [2, n]$. If $Z^{(i)}$ corresponds to $X^{(j)}$ for $j \in [2, r - 2]$, then we map $z^{(i)}_{m,m}$ to $x^{(j)}_m$ and $z^{(i)}_{m,l}$ to 0 if $m \neq l$. If $Z^{(i)}$ corresponds to $Y^{(j)}$, then we map $z^{(i)}_{m,l}$ to $y^{(j)}_{m,l}$. If $Z^{(i)}$ corresponds to $A^{(j)}$, then we map all the variables in $Z^{(i)}$ to 1. Such a projection of $\text{IMM}_{n,d}$ equates to f. Suppose $\text{IMM}_{n,d}$ is computed by a multilinear depth-three circuit C. Then by applying the map on the variables of $\text{IMM}_{n,d}$ and C, we get that the image of C computes f. Two distinct variables are not mapped to the same variable under this map. Hence, image of C is still a multilinear depth-three circuit having top fan-in same as that of C. □

ACKNOWLEDGMENTS

V. N. and C. S. thank Rohit Gurjar for helpful discussions on observations 1.1 and 1.2. Thanks also to Nitin Saxena and Arpita Korwar for some early discussions at the onset of this work. N. K. and C. S. thank Sébastien Tavenas for attending a presentation of this work and giving us some useful feedback.

REFERENCES

[1] Manindra Agrawal. 2005. Proving lower bounds via pseudo-random generators. In *Proceedings of the 25th International Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS'05) (Lecture Notes in Computer Science)*, Vol. 3821. Springer, 92–105.

[2] Manindra Agrawal, Rohit Gurjar, Arpita Korwar, and Nitin Saxena. 2015. Hitting-sets for ROABP and sum of set-multilinear circuits. *SIAM J. Comput.* 44, 3 (2015), 669–697.

[3] Manindra Agrawal, Chandan Saha, and Nitin Saxena. 2013. Quasi-polynomial hitting-set for set-depth-Δ formulas. In *Proceedings of the Symposium on Theory of Computing Conference (STOC'13).* ACM, 321–330.

[4] N. Alon. 1986. Eigenvalues and expanders. *Combinatorica* (1986).

[5] N. Alon and V. D. Milman. 1985. Isoperimetric inequalities for graphs and superconcentrators. *J. Combin. Theory Ser.* (1985).

[6] P. Buser. 1982. A note on the isopoermitric constant. *Annu. Sci. Ecole Norm.* 4, 15 (1982), 213–230.

[7] J. Cheeger. 1970. Lower bound for the smallest eigenvalue of the laplacian. *Prob. Anal.* (1970), 195–199.

[8] Xi Chen, Neeraj Kayal, and Avi Wigderson. 2011. Partial derivatives in arithmetic complexity and beyond. *Found. Trends Theor. Comput. Sci.* 6, 1–2 (2011), 1–138.

[9] Rafael Mendes de Oliveira, Amir Shpilka, and Ben lee Volk. 2016. Subexponential size hitting sets for bounded depth multilinear formulas. *Comput. Complex.* 25, 2 (2016), 455–505.

[10] Richard A. DeMillo and Richard J. Lipton. 1978. A probabilistic remark on algebraic program testing. *Info. Process. Lett.* 7, 4 (1978), 193–195.

[11] J. Dodziuk. 1984. Difference equations, isoperimetric inequality and transience of certain random walks. *SIAM J. Comput.* 284, 2 (1984), 787–794.

[12] Zeev Dvir, Guillaume Malod, Sylvain Perifel, and Amir Yehudayoff. 2012. Separating multilinear branching programs and formulas. In *Proceedings of the 44th Symposium on Theory of Computing Conference (STOC'12).* 615–624.

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

[14] Michael A. Forbes, Ramprasad Saptharishi, and Amir Shpilka. 2014. Hitting sets for multilinear read-once algebraic branching programs, in any order. In *Proceedings of the Symposium on Theory of Computing (STOC'14).* ACM, 867–875.

[15] Michael A. Forbes and Amir Shpilka. 2013. Quasipolynomial-time identity testing of non-commutative and read-once oblivious algebraic branching programs. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS'13).* 243–252.

[16] Hervé Fournier, Nutan Limaye, Guillaume Malod, and Srikanth Srinivasan. 2014. Lower bounds for depth-4 formulas computing iterated matrix multiplication. In *Proceedings of the Symposium on Theory of Computing (STOC'14).* 128–135.

[17] Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. 2013. Arithmetic circuits: A chasm at depth three. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS'13).* 578–587.

[18] Rohit Gurjar, Arpita Korwar, Nitin Saxena, and Thomas Thierauf. 2015. Deterministic identity testing for sum of read-once oblivious arithmetic branching programs. In *Proceedings of the 30th Conference on Computational Complexity (CCC'15).* 323–346.

[19] Philip Hall. 1935. On representatives of subsets. *J. London Math. Soc.* 10, 1 (1935), 26–30.

[20] Joos Heintz and Claus-Peter Schnorr. 1980. Testing polynomials which are easy to compute (extended abstract). In *Proceedings of the 12th Annual ACM Symposium on Theory of Computing.* ACM, 262–272.

[21] Shlomo Hoory, Nathan Linial, and Avi Wigderson. 2006. Expander graphs and their applications. *Bull. Amer. Math. Soc.* 43, 4 (2006), 439–561.

[22] Stasys Jukna. 2015. Lower bounds for tropical circuits and dynamic programs. *Theory Comput. Syst.* 57, 1 (2015), 160–194.

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

[24] Erich Kaltofen. 1989. Factorization of polynomials given by straight-line programs. In *Randomness and Computation.* JAI Press, 375–412.

[25] Neeraj Kayal, Chandan Saha, and Sébastien Tavenas. 2015. Formulas having low individual degree.

[26] Neeraj Kayal and Ramprasad Saptharishi. 2014. A selection of lower bounds for arithmetic circuits. *Perspect. Comput. Complex.* (2014).

[27] Adam Klivans and Daniel A. Spielman. 2001. Randomness efficient identity testing of multivariate polynomials. In *Proceedings on 33rd Annual ACM Symposium on Theory of Computing.* 216–223.

[28] László Lovász. 1979. On determinants, matchings, and random algorithms. In *Proceedings of the Symposium on Fundamentals of Computation Theory (FCT'79).* 565–574.

[29] Meena Mahajan and V. Vinay. 1997. Determinant: Combinatorics, algorithms, and complexity. *Chicago J. Theor. Comput. Sci.* 1997 (1997).

[30] Noam Nisan. 1991. Lower bounds for non-commutative computation (extended abstract). In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing*. 410–418.

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

[32] Noam Nisan and Avi Wigderson. 1997. Lower bounds on arithmetic circuits via partial derivatives. *Comput. Complex.* 6, 3 (1997), 217–234.

[33] Ran Raz. 2006. Separation of multilinear circuit and formula size. *Theory Comput.* 2, 1 (2006), 121–135.

[34] Ran Raz. 2009. Multi-linear formulas for permanent and determinant are of super-polynomial size. *J. ACM* 56, 2 (2009).

[35] Ran Raz and Amir Yehudayoff. 2008. Balancing syntactically multilinear arithmetic circuits. *Comput. Complex.* 17, 4 (2008), 515–535.

[36] Ran Raz and Amir Yehudayoff. 2009. Lower bounds and separations for constant depth multilinear circuits. *Comput. Complex.* 18, 2 (2009), 171–207.

[37] Ramprasad Saptharishi. 2014. Recent progress on arithmetic circuit lower bounds. *Bull. EATCS* 114 (2014).

[38] Nitin Saxena. 2009. Progress on polynomial identity testing. *Bull. EATCS* 99 (2009), 49–79.

[39] Nitin Saxena. 2013. Progress on polynomial identity testing II. *Electr. Colloq. Comput. Complex.* 20 (2013), 186.

[40] Jacob T. Schwartz. 1980. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM* 27, 4 (1980), 701–717.

[41] Amir Shpilka and Amir Yehudayoff. 2010. Arithmetic circuits: A survey of recent results and open questions. *Found. Trends Theor. Comput. Sci.* 5, 3–4 (Mar. 2010), 207–388. DOI: https://doi.org/10.1561/0400000039

[42] Amir Shpilka and Amir Yehudayoff. 2010. Arithmetic circuits: A survey of recent results and open questions. *Found. Trends Theor. Comput. Sci.* 5, 3–4 (2010), 207–388.

[43] W. T. Tutte. 1947. The factorization of linear graphs. *J. London Math. Soc.* 22 (1947), 107–111.

[44] L. G. Valiant. 1979. Completeness Classes in Algebra. In *Proceedings of the 11th Annual ACM Symposium on Theory of Computing (STOC 79)*. ACM Press, New York, NY, 249–261.

[45] Richard Zippel. 1979. Probabilistic algorithms for sparse polynomials. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation (EUROSAM'79)*. 216–226.

Received October 2018; revised August 2019; accepted September 2019