SIAM J. COMPUT.
Vol. 44, No. 3, pp. 669–697

HITTING-SETS FOR ROABP AND SUM OF SET-MULTILINEAR CIRCUITS*

MANINDRA AGRAWAL†, ROHIT GURJAR†, ARPITA KORWAR†, AND NITIN SAXENA†

Abstract. We give an $n^{O(\log n)}$-time (n is the input size) blackbox polynomial identity testing algorithm for unknown-order read-once oblivious arithmetic branching programs (ROABPs). The best time complexity known for blackbox polynomial identity testing (PIT) for this class was $n^{O(\log^2 n)}$ due to Forbes, Saptharishi, and Shpilka [*Proceedings of the* 2014 *ACM Symposium on Theory of Computing*, 2014, pp. 867–875]. Moreover, their result holds only when the individual degree is small, while we do not need any such assumption. With this, we match the time complexity for the unknown-order ROABP with the known-order ROABP (due to Forbes and Shpilka [*Proceedings of the* 2013 *IEEE* 54*th Annual Symposium on Foundations of Computer Science*, 2013, pp. 243–252]) and also with the depth-3 set-multilinear circuits (due to Agrawal, Saha, and Saxena [*Proceedings of the* 2013 *ACM Symposium on Theory of Computing*, 2013, pp. 321–330]). Our proof is simpler and involves a new technique called basis isolation. The depth-3 model has recently gained much importance, as it has become a stepping stone to understanding general arithmetic circuits. Multilinear depth-3 circuits are known to have exponential lower bounds but no polynomial time blackbox identity tests. In this paper, we take a step toward designing such hitting-sets. We give the first subexponential whitebox PIT for the sum of constantly many set-multilinear depth-3 circuits. To achieve this, we define the notions of *distance* and *base sets*. Distance, for a multilinear depth-3 circuit (say, in n variables and k product gates), measures how far the variable partitions corresponding to the product gates are from being a mere *refinement* of each other. The 1-distance circuits strictly contain the set-multilinear model, while n-distance captures general multilinear depth-3. We design a hitting-set in time $(nk)^{O(\Delta \log n)}$ for Δ-distance. Further, we give an extension of our result to models where the distance is large (close to n) but is small when restricted to certain base sets (of variables). We also explore a new model of ROABPs where the factor matrices are *invertible* (called invertible-factor ROABPs). We design a hitting-set in time $\text{poly}(n^{w^2})$ for width-w invertible-factor ROABPs. Further, we could do *without* the invertibility restriction when $w = 2$. Previously, the best result for width-2 ROABPs was quasi-polynomial time [M. A. Forbes, R. Saptharishi, and A. Shpilka, *Proceedings of the* 2014 *ACM Symposium on Theory of Computing*, 2014, pp. 867–875].

Key words. PIT, ROABP, sum of set-multilinear, Δ-distance, basis isolation

AMS subject classifications. 68W30, 68Q25

DOI. 10.1137/140975103

1. Introduction.

The problem of *polynomial identity testing* (PIT) is that of deciding whether a given polynomial is zero. The complexity of the question depends crucially on the way the polynomial is input to the PIT test. For example, if the polynomial is given as a set of coefficients of the monomials, then we can easily check whether the polynomial is nonzero in polynomial time. The problem has been studied for different input models. Most prominent among them is the model of arithmetic circuits. Arithmetic circuits are the arithmetic analogue of boolean circuits and are defined over a field $\mathbb{F}$. They are directed acyclic graphs, where every node is a "+" or "×" gate and each input gate is a constant from the field $\mathbb{F}$ or a variable from $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$. Every edge has a weight from the underlying field $\mathbb{F}$. The computation is done in the natural way. Clearly, the output gate computes a

*Received by the editors June 30, 2014; accepted for publication (in revised form) April 2, 2015; published electronically June 4, 2015.
http://www.siam.org/journals/sicomp/44-3/97510.html
†Department of Computer Science and Engineering, IIT Kanpur, Kanpur, UP 208016, India (manindra@iitk.ac.in, rgurjar@iitk.ac.in, arpk@iitk.ac.in, nitin@iitk.ac.in). The second author's research was supported by TCS. The fourth author's research was supported by DST-SERB.

polynomial in $\mathbb{F}[\mathbf{x}]$. We can restate the PIT problem as follows: Given an arithmetic circuit $\mathcal{C}$, decide whether the polynomial computed by $\mathcal{C}$ is zero, in time polynomial in the circuit size. Note that, given a circuit, computing the polynomial explicitly is not possible in polynomial time, as it can have exponentially many monomials. However, given the circuit, it is easy to compute an evaluation of the polynomial by substituting the variables with constants.

Though there is no known *deterministic* algorithm for PIT, there are easy randomized algorithms [12, 49, 44]. These randomized algorithms are based on the following theorem: A nonzero polynomial, evaluated at a random point, gives a nonzero value with a good probability. Observe that such an algorithm does not need to access the structure of the circuit but just uses the evaluations; it is a *blackbox* algorithm. The other kind of algorithm, where the structure of the input is used, is called a *whitebox* algorithm. Whitebox algorithms for PIT have many known applications. For example, graph matching reduces to PIT. On the other hand, blackbox algorithms (or *hitting-sets*) have connections to circuit lower bound proofs. PIT also fits well in the geometric complexity theory framework; see [30, 29]. See the surveys by Saxena [39, 40] and Shpilka and Yehudayoff [45] for more applications.

An arithmetic branching program (ABP) is another interesting model for computing polynomials. It consists of a directed acyclic graph with a source and a sink. The edges of the graph have polynomials as their weights. The weight of a path is the product of the weights of the edges present in the path. The polynomial computed by the ABP is the sum of the weights of all the paths from the source to the sink. It is well known that for an ABP, the underlying graph can be seen as a layered graph such that all paths from the source to the sink have exactly one edge in each layer. And the polynomial computed by the ABP can be written as a *matrix product*, where each matrix corresponds to a layer. The entries in the matrices are weights of the corresponding edges. The maximum number of vertices in a layer, or, equivalently, the dimension of the corresponding matrices, is called the *width* of the ABP. It is known that projections of symbolic determinant and ABPs have the same expressive power [8, 48, 28]. Ben-Or and Cleve [7] have shown that a polynomial computed by a formula can also be computed by a width-3 ABP of size polynomial in the formula size. A formula is a circuit with every node (except the input gates) having outdegree at most 1. Thus, the ABP is a strong model for computing polynomials. The following chain of reductions shows the power of the ABP and its constant-width version relative to other arithmetic computation models (see [7] and [32, Lemma 1]):

$$\begin{aligned}&\text{Constant-depth arithmetic circuits} \leq_p \text{constant-width ABP}\\ &\quad =_p \text{formulas} \leq_p \text{ABP} \leq_p \text{arithmetic circuits.}\end{aligned}$$

Our first result is for a special class of ABPs called *read-once oblivious arithmetic branching programs (ROABPs)*. An ABP is a read-once oblivious ABP (ROABP) if the weights in its n layers are univariate polynomials in n distinct variables, i.e., the ith layer has weights coming from $\mathbb{F}[x_{\pi(i)}]$, where π is a permutation on the set $\{1, 2, \ldots, n\}$. When we know this permutation π, we call it an ROABP with *known* variable order (it is significant only in the blackbox setting).

Raz and Shpilka [33] gave a poly(n, w, δ)-time whitebox algorithm for n-variate polynomials computed by a width-w ROABP with individual degree bound δ. Recently, Forbes and Shpilka [15, 16] gave a poly$(n, w, \delta)^{\log n}$-time blackbox algorithm for the same case, when the variable order is known. Subsequently, Forbes, Saptharishi, and Shpilka [14] gave a blackbox test for the case of unknown variable order, but with time complexity being $\mathrm{poly}(n)^{\delta \log w \log n}$. Note the exponential dependence on the degree. Their time complexity becomes quasi-polynomial in the case of multilinear polynomials, i.e., $\delta = 1$ (in fact, even when $\delta = \mathrm{poly}(\log n)$).

In another work Jansen, Qiao, and Sarma [21] gave a quasi-polynomial time blackbox test for a sum of constantly many multilinear "ROABP." Their definition of "ROABP" is more stringent. They assume that every variable appears at most once in the ABP. Later, this result was generalized to "read-r OABP" [20], where a variable can occur in at most one layer, and on at most r edges. Our definition of ROABP seems much more powerful than both of these.

We improve the result of [14] and match the time complexity for the unknown-order case with the known-order case (given by [15, 16]). Unlike the result of [14], our result does not have exponential dependence on the individual degree. Formally, we have the following theorem.

THEOREM 1. *Let $C(\mathbf{x})$ be an n-variate polynomial computed by a width-w ROABP (unknown order) with the degree of each variable bounded by δ. Then there is a $\mathrm{poly}(n, w, \delta)^{\log n}$-time hitting-set for C.*

REMARK. *Our algorithm also works when the layers have their weights as general sparse polynomials (still over disjoint sets of variables) instead of univariate polynomials (see the detailed version in section 3).*

A polynomial computed by a width-w ABP can be written as $S^\top D(\mathbf{x})T$, where $S, T \in \mathbb{F}^w$ and $D(\mathbf{x}) \in \mathbb{F}^{w\times w}[\mathbf{x}]$ is a polynomial over the matrix algebra. Like the authors of [5, 14], we try to construct a basis (or extract the rank) for the coefficient vectors in $D(\mathbf{x})$. We actually construct a weight assignment on the variables, which "isolates" a basis in the coefficients in $D(\mathbf{x})$. This idea is inspired by the rank extractor techniques in [5, 14]. Our approach is to directly work with $D(\mathbf{x})$, while the authors of [5, 14] have applied a rank extractor to small subcircuits of $D(\mathbf{x})$ by shifting it carefully. In fact, the idea of "basis isolating weight assignment" evolved when we tried to find a direct proof for the rank extractor in [5], which does not involve subcircuits. However, our techniques go much further than those of both [5, 14], as is evident from our strictly better time complexity results.

The boolean analogues of ROABPs, i.e., read-once ordered branching programs (ROBPs), have been studied extensively with regard to the RL versus L question. For ROBPs, a pseudorandom generator (PRG) with seed length $O(\log^2 n)$ ($n^{O(\log n)}$ size sample space) is known in the case of known variable order [31]. This is analogous to the result [16] for known-order ROABPs. On the other hand, in the unknown-order case, the best known seed length is of size $n^{1/2+o(1)}$ ($2^{n^{1/2+o(1)}}$ size sample space) [19]. One can ask the following: Can the result for the unknown-order case be matched with the known-order case in the boolean setting as well? Recently, there has been partial progress in this direction by [47].

The PIT problem has also been studied for various restricted classes of circuits. One such class is depth-3 circuits. Our second result is about a special case of this class. A depth-3 circuit is usually defined as a $\Sigma\Pi\Sigma$ circuit: The circuit gates are in three layers: the top layer has an output gate which is $+$, the second layer has all $\times$ gates, and the last layer has all $+$ gates. In other words, the polynomial computed by a $\Sigma\Pi\Sigma$ circuit is of the form $C(\mathbf{x}) = \sum_{i=1}^{k} a_i \prod_{j=1}^{n_i} \ell_{ij}$, where n_i is the number of input lines to the ith product gate and ℓ_{ij} is a linear polynomial of the form $b_0 + \sum_{r=1}^{n} b_r x_r$. An efficient solution for depth-3 PIT is still not known. Recently, it was shown by Gupta et al. [17] that depth-3 circuits are almost as powerful as general circuits. A polynomial time hitting-set for a depth-3 circuit implies a quasi-polynomial time hitting-set for general poly-degree circuits. Until now, for depth-3 circuits, efficient PIT has been known when the top fan-in k is assumed to be constant [13, 24, 23, 22, 41, 42, 43] and for certain other restrictions [38, 37, 4].

On the other hand, there are exponential lower bounds for depth-3 *multilinear* circuits [34]. Since there is a connection between lower bounds and PIT [1], we can hope that solving PIT for depth-3 multilinear circuits should also be feasible. This should also lead to new tools for general depth-3.

A polynomial is said to be multilinear if the degree of every variable in every term is at most 1. The circuit $C(\mathbf{x})$ is a multilinear circuit if the polynomial computed at every gate is multilinear. A polynomial time algorithm is known only for a subclass of multilinear depth-3 circuits, called *depth-3 set-multilinear circuits*. This algorithm is due to Raz and Shpilka [33] and is whitebox. In a depth-3 multilinear circuit, since every product gate computes a multilinear polynomial, a variable occurs in at most one of the linear polynomials input to it. Thus, each product gate naturally induces a *partition* of the variables, where each *color* (i.e., part) of the partition contains the variables present in a linear polynomial. Further, if the partitions induced by all the product gates are the same, then the circuit is called a depth-3 set-multilinear circuit.

Agrawal, Saha, and Saxena [5] gave a quasi-polynomial time blackbox PIT for the class of depth-3 set-multilinear circuits. But before this work, no subexponential time PIT (not even whitebox) was known even for sum of two set-multilinear circuits. We give a subexponential time whitebox PIT for sum of constantly many set-multilinear circuits (see also subsequent work mentioned at the end of this section).

THEOREM 2. *Let $C(\mathbf{x})$ be an n-variate polynomial which is a sum of c set-multilinear depth-3 circuits, each having top fan-in k. Then there is an $n^{O(n^{1-\epsilon}\log k)}$-time whitebox test for C, where $\epsilon := 1/2^{c-1}$.*

To achieve this, we define a new class of circuits, as a tool, called *multilinear depth-3 circuits with Δ-distance*. A multilinear depth-3 circuit has Δ-distance if there is an ordering on the partitions induced by the product gates, say $(\mathbb{P}_1, \mathbb{P}_2, \ldots, \mathbb{P}_k)$, such that for any color in the partition $\mathbb{P}_i$, there exists a set of $\leq (\Delta - 1)$ other colors in $\mathbb{P}_i$ such that the set of variables in the union of these $\leq \Delta$ colors are *exactly* partitioned in the upper partitions, i.e., $\{\mathbb{P}_1, \mathbb{P}_2, \ldots, \mathbb{P}_{i-1}\}$. As we will see, such sets of Δ colors form equivalence classes of the colors at partition $\mathbb{P}_i$. We call them friendly neighborhoods, and they help us in identifying subcircuits. Intuitively, the distance measures how far away the partitions are from a mere *refinement* sequence of partitions, $\mathbb{P}_1 \leq \mathbb{P}_2 \leq \cdots \leq \mathbb{P}_k$.[1] A refinement sequence of partitions will have distance 1. On the other hand, general multilinear depth-3 circuits can have at most n-distance.

As it turns out, a polynomial computed by a depth-3 Δ-distance circuit (top fan-in k) can also be computed by a width-$O(kn^{\Delta})$ ROABP (see Lemma 14). Thus, from Theorem 1 we get a $\text{poly}(nk)^{\Delta \log n}$-time hitting-set for this class. Next, we use a general result about finding a hitting-set for a class *m-base-sets-*C if a hitting-set is known for class C. A polynomial is in m-base-sets-C if there exists a partition of the variables into m base sets such that restricted to each base set (treat other variables as the function field constants), the polynomial is in class C. We combine these two tools to prove Theorem 2. We show that a sum of constantly many set-multilinear circuits falls into the class m-base-sets-Δ-distance for $m\Delta = o(n)$.

Agrawal et al. [3] achieved *rank concentration*, which implies a hitting-set, for the class m-base-sets-Δ-distance, but through complicated proofs. On the other hand, this work gives only a hitting-set for the same class, but with the advantage of simplified proofs.

[1] That is, for all i, each color in $\mathbb{P}_i$ gets exactly partitioned in the upper partitions.

Our third result deals again with ABPs. The results of [7] and [36] show that the constant-width ABPs capture several natural subclasses of circuits. Here, we study constant-width ABPs with some natural restrictions.

We consider a class of ROABPs where all the matrices in the matrix product are invertible. We give a blackbox test for this class of ROABPs. In contrast to [14] and our Theorem 1, this test works in *polynomial time* if the dimension of the matrices is constant.

Note that this class of ABPs, where the factor matrices are invertible, is quite powerful, as Ben-Or and Cleve [7] actually reduce formulas to width-3 ABPs with *invertible* factors. Saha, Saptharishi, and Saxena [36] reduce depth-3 circuits to width-2 ABPs with invertible factors. But the constraints of invertibility and read-once together seem to restrict the computing power of the ABPs, making them amenable to this line of attack. Interestingly, an analogous class of read-once boolean branching programs called *permutation branching programs* has been studied recently [26, 10, 46]. These works give a PRG for this class (for constant width) with seed-length $O(\log n)$ in the known variable order case. In other words, they give polynomial size sample space which can fool these programs. For the unknown variable order case, Reingold, Steinke, and Vadhan [35] give a PRG with seed-length $O(\log^2 n)$. Our polynomial size hitting-set for the arithmetic setting works for any unknown variable order. Hence, it is better as compared to the currently known results for the boolean case.

THEOREM 3 (informal version). *Let $C(\mathbf{x}) = D_0^\top(\prod_{i=1}^d D_i)D_{d+1}$ be a polynomial such that $D_0, D_{d+1} \in \mathbb{F}^w$ and for all $i \in [d]$, $D_i \in \mathbb{F}^{w\times w}[x_{j_i}]$ is an invertible matrix (order of the variables is unknown). Let the degree bound on D_i be δ for $0 \le i \le d+1$. Then there is a $\mathrm{poly}((\delta n)^{w^2})$-time hitting-set for $C(\mathbf{x})$.*

The proof technique here is very different from the first two theorems (here we show *rank concentration;* see the proof idea in section 5). Our algorithm works even when the factor matrices have their entries as general sparse polynomials (still over disjoint sets of variables) instead of univariate polynomials (see the detailed version in section 5).

If the matrices are 2×2, then we do not need the assumption of invertibility (see Theorem 37, section 5.4). So, for width-2 ROABPs our results are strictly stronger than [14] and our Theorem 1. Here again, there is a comparable result in the boolean setting. PRGs with seed-length $O(\log n)$ (polynomial size sample space) are known for width-2 ROBPs [9].

Subsequent work. The models introduced in this paper have led to some fruitful research recently. Our result on sum of set-multilinear circuits (Theorem 2) has been greatly improved by subsequent works of Gurjar et al. [18] and de Oliveira, Shpilka, and Volk [11]. The authors of [18] gave a polynomial time whitebox PIT for a sum of constantly many ROABPs (ROABPs subsume set-multilinear circuits). They also gave a quasi-polynomial time blackbox PIT for the same model. As a by-product, they showed that the hitting-set in Theorem 1 can be used as a shift to make an ROABP $O(\log w)$-concentrated. Thus, basis isolation is as strong as rank concentration.

The authors of [11] gave a subexponential time ($n^{\tilde{O}(n^{2/3})}$) blackbox PIT for depth-3 multilinear circuits. This model is equivalent to sum of arbitrarily many set-multilinear circuits.

2. Preliminaries.

Hitting-set. A set of points $\mathcal{H}$ is called a hitting-set for a class C of polynomials if, for any nonzero polynomial P in C, there exists a point in $\mathcal{H}$ where P evaluates to a nonzero value. An $f(n)$-time hitting-set would mean that the hitting-set can be generated in time $f(n)$ for input size n.

2.1. Notation. $\mathbb{Z}_+$ denotes the set $\mathbb{N} \cup \{0\}$. $[n]$ denotes the set $\{1, 2, \ldots, n\}$. $[\![n]\!]$ denotes the set $\{0, 1, \ldots, n\}$. $\mathbf{x}$ denotes a set of variables. For a set of n variables $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ and for an exponent $\mathbf{e} = (e_1, e_2, \ldots, e_n) \in \mathbb{Z}_+^n$, $\mathbf{x}^e$ denotes the monomial $\prod_{i=1}^n x_i^{e_i}$. The *support* of a monomial is the set of variables that have degree ≥ 1 in that monomial. The *support size* of the monomial is the cardinality of its support. A polynomial is called *s-sparse* if there are at most s monomials in it with nonzero coefficients. For a polynomial P, the coefficient of the monomial m in $P(\mathbf{x})$ is denoted by $\mathrm{coef}_P(m)$.

$\mathbb{F}^{m \times n}$ represents the set of all $m \times n$ matrices over the field $\mathbb{F}$. $\mathbb{M}_{m\times m}(\mathbb{F})$ denotes the algebra of $m \times m$ matrices over the field $\mathbb{F}$. Let $\mathbb{A}_k(\mathbb{F})$ be any k-dimensional algebra over the field $\mathbb{F}$. For any two elements $A = (a_1, a_2, \ldots, a_k) \in \mathbb{A}_k(\mathbb{F})$ and $B = (b_1, b_2, \ldots, b_k) \in \mathbb{A}_k(\mathbb{F})$ (having a natural basis representation in mind), their dot product is defined as $A \cdot B = \sum_{i=1}^n a_k b_k$, and the product AB denotes the product in the algebra $\mathbb{A}_k(\mathbb{F})$.

$\mathrm{Part}(S)$ denotes the set of all possible partitions of the set S. Elements in a partition are called *colors* (or parts).

2.2. Arithmetic branching programs. An ABP is a directed graph with $d+1$ layers of vertices $\{V_0, V_1, \ldots, V_d\}$ and a start node u and an end node t such that the edges are only going from u to V_0, V_{i-1} to V_i for any $i \in [d]$, V_d to t. A width-w ABP has $|V_i| \leq w$ for all $i \in [\![d]\!]$. Let the set of nodes in V_i be $\{v_{i,j} \mid j \in [w]\}$. All the edges in the graph have weights from $\mathbb{F}[\mathbf{x}]$ for some field $\mathbb{F}$. As a convention, the edges going from u and coming to t are assumed to have weights from the field $\mathbb{F}$.

For an edge e, let us denote its weight by $W(e)$. For a path p from u to t, its weight $W(p)$ is defined to be the product of weights of all the edges in it, i.e., $\prod_{e\in p} W(e)$. Consider the polynomial $C(\mathbf{x}) = \sum_{p\in \mathrm{paths}(u,t)} W(p)$, which is the sum of the weights of all the paths from u to t. This polynomial $C(\mathbf{x})$ is said to be computed by the ABP.

It is easy to see that this polynomial is the same as $S^\top(\prod_{i=1}^d D_i)T$, where $S, T \in \mathbb{F}^w$ and D_i is a $w \times w$ matrix for $1 \leq i \leq d$ such that

$$\begin{aligned} S(\ell) &= W(u, v_{0,\ell}) \quad \text{ for } 1 \leq \ell \leq w, \\ D_i(k,\ell) &= W(v_{i-1,k}, v_{i,\ell}) \quad \text{ for } 1 \leq \ell, k \leq w \text{ and } 1 \leq i \leq d, \\ T(k) &= W(v_{d,k}, t) \quad \text{ for } 1 \leq k \leq w. \end{aligned}$$

ROABP. An ABP is called a *read-once oblivious ABP (ROABP)* if the edge weights in the different layers are univariate polynomials in distinct variables. Formally, the entries in D_i come from $\mathbb{F}[x_{\pi(i)}]$ for all $i \in [d]$, where π is a permutation on the set $[d]$.

Sparse-factor ROABP. We call the ABP a *sparse-factor ROABP* if the edge weights in different layers are sparse polynomials in disjoint sets of variables. Formally, if there exists an unknown partition of the variable set $\mathbf{x}$ into d sets $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_d\}$ such that $D_i \in \mathbb{F}^{w\times w}[\mathbf{x}_i]$ is an s-sparse polynomial for all $i \in [d]$, then the corresponding ROABP is called an *s-sparse-factor* ROABP. It is read-once in the sense that any particular variable contributes to at most one edge on any path.

2.3. Kronecker map. We will often use a weight function on the variables which separates a desired set of monomials. It is a folklore trick to solve blackbox PIT for sparse polynomials. The following lemma is used toward the end of the proofs of Theorems 1 and 3 to design the hitting-sets. Let $w\colon \mathbf{x} \to \mathbb{N}$ be a weight function on the variables. Consider its natural extension to the set of all monomials $w\colon \mathbb{Z}_+^n \to \mathbb{N}$ as follows: $w(\Pi_{i=1}^n x_i^{\gamma_i}) = \sum_{i=1}^n \gamma_i w(x_i)$, where $\gamma_i \in \mathbb{Z}_+$ for all $i \in [n]$. Note that if each variable x_i is replaced with $t^{w(x_i)}$, then any monomial m just becomes $t^{w(m)}$.

LEMMA 4 (efficient Kronecker map [27, 2]). *Let $\mathcal{M}$ be the set of all monomials in n variables $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ with maximum individual degree δ. For any value a, there exists a (constructible) set of $N := na\log(\delta+1)$ weight functions $w\colon \mathbf{x} \to \{1, \ldots, 2N\log N\}$ such that for any set A of pairs of monomials from $\mathcal{M}$ with $|A| = a$, at least one of the weight functions separates all the pairs in A; i.e., for all $(m, m') \in A$, $w(m) \neq w(m')$ (the proof is described in Appendix* A*).*

3. Hitting-set for ROABP: Theorem 1. Like the authors of [5] and [14], we work with the vector polynomial. That is, for a polynomial computed by a width-w ROABP, $C(\mathbf{x}) = S^\top(\prod_{i=1}^d D_i)T$, we see the product $D := \prod_{i=1}^d D_i$ as a polynomial over the matrix algebra $\mathbb{M}_{w\times w}(\mathbb{F})$. We can write the polynomial $C(\mathbf{x})$ as the dot product $R\cdot D$, where $R = ST^\top$. The vector space spanned by the coefficients of $D(\mathbf{x})$ is called the coefficient space of $D(\mathbf{x})$. This space will have dimension at most w^2. We essentially try to construct a small set of vectors by evaluating $D(\mathbf{x})$, which can span the coefficient space of $D(\mathbf{x})$. Clearly, if $C \neq 0$, then the dot product of R with at least one of these spanning vectors will be nonzero. And thus, we get a hitting-set.

Unlike the authors of [5] and [14], we directly work with the original polynomial $D(\mathbf{x})$, instead of shifting it and breaking it into subcircuits. For a polynomial in $\mathbb{F}[\mathbf{x}]$, a usual technique for PIT is to give a univariate monomial map for the variables such that a monomial of the given polynomial is isolated (e.g., sparse PIT [25]). Our approach can be seen as a generalization of this technique. We come up with a univariate map (or weight function) on the variables which can *isolate a basis* for the coefficients of the polynomial $D(\mathbf{x}) \in \mathbb{M}_{w\times w}[\mathbf{x}]$.

We present our results for polynomials over arbitrary algebra. Let $\mathbb{A}_k(\mathbb{F})$ be a k-dimensional algebra over the field $\mathbb{F}$. Let $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ be a set of variables, and let $D(\mathbf{x})$ be a polynomial in $\mathbb{A}_k(\mathbb{F})[\mathbf{x}]$ with highest individual degree δ. Let $\mathcal{M}$ denote the set of all monomials over the variable set $\mathbf{x}$ with highest individual degree δ.

Now we will define a basis isolating weight assignment for a polynomial $D \in \mathbb{A}_k(\mathbb{F})[\mathbf{x}]$, which will lead to a hitting-set for the polynomial $C \in \mathbb{F}[\mathbf{x}]$, where $C = R\cdot D$ for some $R \in \mathbb{A}_k(\mathbb{F})$.

DEFINITION 5 (basis isolating weight assignment). *A weight function $w\colon \mathbf{x} \to \mathbb{N}$ is called a basis isolating weight assignment for a polynomial $D(\mathbf{x}) \in \mathbb{A}_k(\mathbb{F})[\mathbf{x}]$ if there exists a set of monomials $S \subseteq \mathcal{M}$ ($k' := |S| \leq k$) whose coefficients form a basis for the coefficient space of $D(\mathbf{x})$ such that*

- *for any $m, m' \in S$, $w(m) \neq w(m')$, and*
- *for any monomial $m \in \mathcal{M} \setminus S$,*

$$\mathrm{coef}_D(m) \in \mathrm{span}\{\mathrm{coef}_D(m') \mid m' \in S,\ w(m') < w(m)\}.$$

The above definition is equivalent to saying that there exists a *unique minimum* weight basis (according to the weight function w) among the coefficients of D, and also that the basis monomials have distinct weights. We skip the easy proof for this equivalence, as we will not need it. Let us emphasize here that according to this definition there could be many monomials in $\mathcal{M} \setminus S$ which have the same weight as a monomial m in S. The only requirement is that their coefficients should be linearly dependent on basis coefficients with weight *smaller* than $w(m)$.

Note that a weight assignment which gives distinct weights to all the monomials is indeed a basis isolating weight assignment. However, it will involve exponentially large weights. To find an efficient weight assignment one must use some properties of the given circuit. First, we show how such a weight assignment would lead to a hitting-set. We will actually show that it isolates a monomial in $C(\mathbf{x})$.

LEMMA 6. *Let $w\colon \mathbf{x} \to \mathbb{N}$ be a basis isolating weight assignment for a polynomial $D(\mathbf{x}) \in \mathbb{A}_k(\mathbb{F})[\mathbf{x}]$, and let $C = R \cdot D$ be a nonzero polynomial for some $R \in \mathbb{A}_k(\mathbb{F})$. Then, after the substitution $x_i = t^{w(x_i)}$ for all $i \in [n]$, the polynomial C remains nonzero, where t is an indeterminate.*

Proof. For any monomial $m \in \mathcal{M}$, let $D_m \in \mathbb{A}_k(\mathbb{F})$ denote the coefficient $\mathrm{coef}_D(m)$. It is easy to see that after the mentioned substitution, the new polynomial $C'(t)$ is equal to $\sum_{m\in\mathcal{M}}(R \cdot D_m)t^{w(m)}$.

Let us say that $S \subset \mathcal{M}$ is the set of monomials whose coefficients form the isolated basis for D. According to the definition of the basis isolating weight assignment, for any monomial $m \in \mathcal{M} \setminus S$,

$$D_m \in \mathrm{span}\{D_{m'} \mid m' \in S,\ w(m') < w(m)\}. \tag{1}$$

First, we claim that there exists $m' \in S$ such that $R \cdot D_{m'} \neq 0$. For the sake of contradiction, let us assume that for all $m' \in S$, $R \cdot D_{m'} = 0$. Taking the dot product with R on both sides of (1), we get that for any monomial $m \in \mathcal{M} \setminus S$,

$$R \cdot D_m \in \mathrm{span}\{R \cdot D_{m'} \mid m' \in S,\ w(m') < w(m)\}.$$

Hence, $R \cdot D_m = 0$ for all $m \in \mathcal{M}$. This means that $C(\mathbf{x}) = 0$, which contradicts our assumption.

Now, let m^* be the minimum weight monomial in S whose coefficient gives a nonzero dot product with R, i.e., $m^* = \arg\min_{m\in S}\{w(m) \mid R \cdot D_m \neq 0\}$. There is a unique such monomial in S because all the monomials in S have distinct weights.

We claim that $\mathrm{coef}_{C'}(t^{w(m^*)}) \neq 0$ and hence $C'(t) \neq 0$. To see this, consider any monomial m, other than m^*, with $w(m) = w(m^*)$. The monomial m has to be in the set $\mathcal{M} \setminus S$, as the monomials in S have distinct weights. From (1),

$$D_m \in \mathrm{span}\{D_{m'} \mid m' \in S,\ w(m') < w(m^*)\}.$$

Taking the dot product with R on both sides, we get

$$R \cdot D_m \in \mathrm{span}\{R \cdot D_{m'} \mid m' \in S,\ w(m') < w(m^*)\}.$$

But, by the choice of m^*, $R \cdot D_{m'} = 0$ for any $m' \in S$ with $w(m') < w(m^*)$. Hence, $R \cdot D_m = 0$ for any $m \neq m^*$ with $w(m) = w(m^*)$.

Thus, the coefficient $\mathrm{coef}_{C'}(t^{w(m^*)})$ can be written as

$$\sum_{\substack{m\in\mathcal{M}\\ w(m)=w(m^*)}} R \cdot D_m = R \cdot D_{m^*},$$

which, as we know, is nonzero. ∎

We continue to use C' and S as in the proofs of Lemma 6. To construct a hitting-set for $C'(t)$, we can try many possible field values for t. The number of such values needed will be the degree of $C'(t)$, which is at most $(n\delta \max_i w(x_i))$. Hence, the cost of the hitting-set is dominated by the *cost of the weight function*, i.e., the maximum weight given to any variable and the time taken to construct the weight function.

In the next step, we show that such a basis isolating weight assignment can indeed be found for a sparse-factor ROABP, but with cost quasi-polynomial in the input size. First, we make the following observation that it suffices that the coefficients of the monomials not in S linearly depend on any coefficients with strictly smaller weight, not necessarily coming from S.

OBSERVATION 7. *If, for a polynomial $D \in \mathbb{A}_k(\mathbb{F})[\mathbf{x}]$, there exist a weight function $w\colon \mathbf{x} \to \mathbb{N}$ and a set of monomials $S \subseteq \mathcal{M}$ ($k' := |S| \le k$) such that for any monomial $m \in \mathcal{M} \setminus S$,*

$$\mathrm{coef}_D(m) \in \mathrm{span}\{\mathrm{coef}_D(m') \mid m' \in \mathcal{M},\ w(m') < w(m)\},$$

then we can also conclude that for any monomial $m \in \mathcal{M} \setminus S$,

$$\mathrm{coef}_D(m) \in \mathrm{span}\{\mathrm{coef}_D(m') \mid m' \in S,\ w(m') < w(m)\}.$$

Proof. We are given that for any monomial $m \in \overline{S} := \mathcal{M} \setminus S$,

$$\mathrm{coef}_D(m) \in \mathrm{span}\{\mathrm{coef}_D(m') \mid m' \in \mathcal{M},\ w(m') < w(m)\}.$$

Any coefficient $\mathrm{coef}_D(m')$ on the right-hand side of this equation which corresponds to an index in $\overline{S}$ can be replaced with some other coefficients which have even smaller weight. If we keep doing this, we will be left with only the coefficients corresponding to the set S, because in each step we are getting smaller and smaller weight coefficients. ∎

In our construction of the weight function, we will create the set $\overline{S} := \mathcal{M} \setminus S$ incrementally; i.e., in each step we will make more coefficients depend on strictly smaller weight coefficients. Finally, we will be left with only k' (the rank of the coefficient space of D) coefficients in S. We present the result for an arbitrary k-dimensional algebra $\mathbb{A}_k(\mathbb{F})$ instead of just the matrix algebra.

LEMMA 8 (weight construction). *Let $\mathbf{x}$ be given by a union of d disjoint sets of variables $\mathbf{x}_1 \sqcup \mathbf{x}_2 \sqcup \cdots \sqcup \mathbf{x}_d$, with $|\mathbf{x}| = n$. Let $D(\mathbf{x}) = P_1(\mathbf{x}_1)P_2(\mathbf{x}_2)\cdots P_d(\mathbf{x}_d)$, where $P_i \in \mathbb{A}_k(\mathbb{F})[\mathbf{x}_i]$ is an s-sparse, individual degree-δ polynomial for all $i \in [d]$. Then, we can construct a basis isolating weight assignment for $D(\mathbf{x})$ with the cost being $(\mathrm{poly}(k, s, n, \delta))^{\log d}$.*

Proof. In our construction, the final weight function w will be a combination of $(\log d + 1)$ different weight functions, say $(w_0, w_1, \ldots, w_{\log d})$. The weight function w is said to give an ordering on the monomials which comes from the lexicographic ordering given by the weight functions $(w_0, w_1, \ldots, w_{\log d})$. Let us say their precedence is decreasing from left to right; i.e., w_0 has the highest precedence, and $w_{\log d}$ has the lowest precedence. As mentioned earlier, we will build the set $\overline{S}$ (the set of monomials whose coefficients are in the span of strictly smaller weight coefficients than themselves) incrementally in $(\log d + 1)$ steps, using weight function w_i in the $(i+1)$th step.

Let $\mathcal{M}_{0,1}, \mathcal{M}_{0,2}, \ldots, \mathcal{M}_{0,d}$ be the sets of monomials and $\mathcal{C}_{0,1}, \mathcal{C}_{0,2}, \ldots, \mathcal{C}_{0,d}$ be the sets of coefficients in the polynomials $P_1, P_2, \ldots, P_d$, respectively.

NOTATION. *The product of two sets of monomials $\mathcal{M}_1$ and $\mathcal{M}_2$ is defined as $\mathcal{M}_1 \times \mathcal{M}_2 = \{m_1m_2 \mid m_1 \in \mathcal{M}_1,\ m_2 \in \mathcal{M}_2\}$. The product of any two sets of coefficients $\mathcal{C}_1$ and $\mathcal{C}_2$ is defined as $\mathcal{C}_1 \times \mathcal{C}_2 = \{c_1c_2 \mid c_1 \in \mathcal{C}_1,\ c_2 \in \mathcal{C}_2\}$.*

The crucial property of the polynomial D is that the set of coefficients in D, $\mathcal{C}_0$, is just the product $\mathcal{C}_{0,1} \times \mathcal{C}_{0,2} \times \cdots \times \mathcal{C}_{0,d}$. Similarly, the set of all the monomials in D, say $\mathcal{M}_0$, can be viewed as the product $\mathcal{M}_{0,1} \times \mathcal{M}_{0,2} \times \cdots \times \mathcal{M}_{0,d}$. Let $m := m_a m_{a+1} \cdots m_b$ be a monomial, where $1 \leq a \leq b \leq d$ and $m_j \in \mathcal{M}_{0,j}$ for $a \leq j \leq b$. Then D_m will denote the coefficient $\mathrm{coef}_{P_a}(m_a)\,\mathrm{coef}_{P_{a+1}}(m_{a+1}) \cdots \mathrm{coef}_{P_b}(m_b)$.

Iteration 0: Let us fix $w_0 \colon \mathbf{x} \to \mathbb{N}$ to be a weight function on the variables which gives distinct weights to all the s monomials in $\mathcal{M}_{0,i}$ for each $i \in [d]$. As w_0 assigns distinct weights to these monomials, so does the weight function w.

For each P_i we do the following:

- arrange the coefficients in $\mathcal{C}_{0,i}$ in increasing order of their weight according to w (or, equivalently, according to w_0),
- choose a maximal set of linearly independent coefficients, in a greedy manner, going from lower weights to higher weights.

The fact that the weight functions $w_1, w_2, \ldots, w_{\log d}$ are not defined yet does not matter because w_0 has the highest precedence. The total order given to the monomials in $\mathcal{M}_{0,i}$ by w_0 is the same as given by w, irrespective of what the functions $w_1, \ldots, w_{\log d}$ are chosen to be.

This gives us a basis for the coefficients of P_i, say $\mathcal{C}'_{0,i}$. Let $\mathcal{M}'_{0,i}$ denote the monomials in P_i corresponding to these basis coefficients. From the construction of the basis, it follows that for any monomial $m \in \mathcal{M}_{0,i} \setminus \mathcal{M}'_{0,i}$,

$$D_m \in \operatorname{span}\{D_{m'} \mid m' \in \mathcal{M}'_{0,i},\ w(m') < w(m)\}. \tag{2}$$

Now, consider any monomial $m \in \mathcal{M}_0$ which is not present in the set $\mathcal{M}'_0 := \mathcal{M}'_{0,1} \times \mathcal{M}'_{0,2} \times \cdots \times \mathcal{M}'_{0,d}$. Let $m = m_1 m_2 \cdots m_d$, where $m_i \in \mathcal{M}_{0,i}$ for all $i \in [d]$. We know that for at least one $j \in [d]$, $m_j \in \mathcal{M}_{0,j} \setminus \mathcal{M}'_{0,j}$. Then using (2) we can write the following about $D_m = D_{m_1} D_{m_2} \cdots D_{m_d}$:

$$D_m \in \operatorname{span}\{D_{m_1} \cdots D_{m_{j-1}} D_{m'_j} D_{m_{j+1}} \cdots D_{m_d} \mid m'_j \in \mathcal{M}'_{0,j},\ w(m'_j) < w(m_j)\}.$$

This holds because the algebra product is bilinear. Equivalently, for any monomial $m \in \mathcal{M}_0 \setminus \mathcal{M}'_0$,

$$D_m \in \operatorname{span}\{D_{m'} \mid m' \in \mathcal{M}_0,\ w(m') < w(m)\}.$$

This is true because

$$w(m_1) + \cdots + w(m'_j) + \cdots + w(m_d) < w(m_1) + \cdots + w(m_j) + \cdots + w(m_d) = w(m).$$

Hence, all the monomials in $\mathcal{M}_0 \setminus \mathcal{M}'_0$ can be put into $\overline{S}$; i.e., their corresponding coefficients depend on strictly smaller weight coefficients.

Iteration 1: Now let us consider monomials in the set $\mathcal{M}'_0 = \mathcal{M}'_{0,1} \times \mathcal{M}'_{0,2} \times \cdots \times \mathcal{M}'_{0,d}$. Let the corresponding set of coefficients be $\mathcal{C}'_0 := \mathcal{C}'_{0,1} \times \mathcal{C}'_{0,2} \times \cdots \times \mathcal{C}'_{0,d}$. Since the underlying algebra $\mathbb{A}_k(\mathbb{F})$ has dimension at most k and the coefficients in $\mathcal{C}'_{0,i}$ form a basis for $\mathcal{C}_{0,i}$, $|\mathcal{M}'_{0,i}| \leq k$ for all $i \in [d]$. In the above product, let us make $d/2$ disjoint pairs of consecutive terms and for each pair, multiply the two terms in it. Putting it formally, let us define $\mathcal{C}_{1,j}$ to be the product $\mathcal{C}'_{0,2j-1} \times \mathcal{C}'_{0,2j}$ and similarly $\mathcal{M}_{1,j} := \mathcal{M}'_{0,2j-1} \times \mathcal{M}'_{0,2j}$ for all $j \in [d/2]$ (if d is odd, we can make it even by multiplying the identity element of $\mathbb{A}_k(\mathbb{F})$ in the end). Now, let $\mathcal{C}_1 := \mathcal{C}'_0 = \mathcal{C}_{1,1} \times \mathcal{C}_{1,2} \times \cdots \times \mathcal{C}_{1,d_1}$ and $\mathcal{M}_1 := \mathcal{M}'_0 = \mathcal{M}_{1,1} \times \mathcal{M}_{1,2} \times \cdots \times \mathcal{M}_{1,d_1}$, where $d_1 := d/2$. For any $i \in [d_1]$, $\mathcal{M}_{1,i}$ has at most k^2 monomials.

Now we fix the weight function $w_1 \colon \mathbf{x} \to \mathbb{N}$ such that it gives distinct weights to all the monomials in $\mathcal{M}_{1,i}$ for each $i \in [d_1]$. As w_1 separates these monomials, so does the weight function w. Now we repeat the same procedure of constructing a basis in a greedy manner for $\mathcal{C}_{1,i}$ according to the weight function w for each $i \in [d_1]$. Let the basis coefficients for $\mathcal{C}_{1,i}$ be $\mathcal{C}'_{1,i}$ and the corresponding monomials be $\mathcal{M}'_{1,i}$.

As argued before, any coefficient in $\mathcal{C}_1$ which is outside the set $\mathcal{C}'_1 := \mathcal{C}'_{1,1} \times \mathcal{C}'_{1,2} \times \cdots \times \mathcal{C}'_{1,d_1}$ is in the span of strictly smaller weight (than itself) coefficients. Thus, we can also put the corresponding monomials $\mathcal{M}_1 \setminus \mathcal{M}'_1$ in $\overline{S}$, where $\mathcal{M}'_1 := \mathcal{M}'_{1,1} \times \mathcal{M}'_{1,2} \times \cdots \times \mathcal{M}'_{1,d_1}$.

Iteration r: We keep repeating the same procedure for $(\log d + 1)$ rounds. After round r, say, the set of monomials we are left with is given by the product $\mathcal{M}'_{r-1} = \mathcal{M}'_{r-1,1} \times \mathcal{M}'_{r-1,2} \times \cdots \times \mathcal{M}'_{r-1,d_{r-1}}$, where $\mathcal{M}_{r-1,i}$ has at most k monomials, for each $i \in [d_{r-1}]$ and $d_{r-1} = d/2^{r-1}$. In the above product, we make $d_{r-1}/2$ disjoint pairs of consecutive terms and multiply the two terms in each pair. Let us say we get $\mathcal{M}_r := \mathcal{M}'_{r-1} = \mathcal{M}_{r,1} \times \mathcal{M}_{r,2} \times \cdots \times \mathcal{M}_{r,d_r}$, where $d_r = d_{r-1}/2$. Then, the corresponding set of coefficients is given by $\mathcal{C}_r = \mathcal{C}_{r,1} \times \mathcal{C}_{r,2} \times \cdots \times \mathcal{C}_{r,d_r}$. Note that $|\mathcal{M}_{r,i}| \leq k^2$ for each $i \in [d_r]$.

We fix the weight function w_r such that it gives distinct weights to all the monomials in the set $\mathcal{M}_{r,i}$ for each $i \in [d_r]$. We once again mention that fixing of w_r does not affect the greedy basis constructed in earlier rounds and hence the monomials which were put in the set $\overline{S}$, because w_r has less precedence than any $w_{r'}$ for $r' < r$.

For each $\mathcal{C}_{r,i}$, we construct a basis in a greedy manner going from lower weight to higher weight (according to the weight function w). Let this set of basis coefficients be $\mathcal{C}'_{r,i}$ and the corresponding monomials be $\mathcal{M}'_{r,i}$ for each $i \in [d_r]$. Let $\mathcal{C}'_r := \mathcal{C}'_{r,1} \times \mathcal{C}'_{r,2} \times \cdots \times \mathcal{C}'_{r,d_r}$ and $\mathcal{M}'_r := \mathcal{M}'_{r,1} \times \mathcal{M}'_{r,2} \times \cdots \times \mathcal{M}'_{r,d_r}$. Arguing similarly as before we can say that each coefficient in $\mathcal{C}_{r,i} \setminus \mathcal{C}'_{r,i}$ is in the span of strictly smaller weight coefficients (from $\mathcal{C}'_{r,i}$) than itself. Hence, the same can be said about any coefficient in the set $\mathcal{C}_r \setminus \mathcal{C}'_r$. So, all the monomials in the set $\mathcal{M}_r \setminus \mathcal{M}'_r$ can be put into $\overline{S}$. Now, we are left with monomials $\mathcal{M}'_r = \mathcal{M}'_{r,1} \times \mathcal{M}'_{r,2} \times \cdots \times \mathcal{M}'_{r,d_r}$ for the next round.

Iteration $\log d$: As in each round the number of terms in the product gets halved, after $\log d$ rounds we will be left with just one term, i.e., $\mathcal{M}_{\log d} = \mathcal{M}'_{\log d-1,1} \times \mathcal{M}'_{\log d-1,2} = \mathcal{M}_{\log d,1}$. Now we will fix the function $w_{\log d}$ which separates all the monomials in $\mathcal{M}_{\log d,1}$. By arguments similar to those above, we will finally be left with at most k' monomials in S, which will all have distinct weights. It is clear that for every monomial in $\overline{S}$, its coefficient will be in the span of strictly smaller weight coefficients than itself.

Now let us look at the cost of this weight function. In the first round, w_0 needs to separate at most $O(ds^2)$ many pairs of monomials. For each $1 \leq r \leq \log d$, w_r needs to separate at most $O(dk^4)$ many pairs of monomials. From Lemma 4, to construct w_r, for any $0 \leq r \leq \log d$, one needs to try $\mathrm{poly}(k, s, n, \delta)$ many weight functions each having highest weight at most $\mathrm{poly}(k, s, n, \delta)$ (as d is bounded by n). To get the correct combination of the weight functions $(w_0, w_1, \ldots, w_{\log d})$ we need to try all possible combinations of these polynomially many choices for each w_r. Thus, we have to try $(\mathrm{poly}(k, s, n, \delta))^{\log d}$ many combinations.

To combine these weight functions we can choose a large enough number B (greater than the highest weight a monomial can get in any of the weight functions) and define $w := w_0 B^{\log d} + w_1 B^{\log d - 1} + \cdots + w_{\log d}$. The choice of B ensures that the different weight functions cannot interfere with each other, and they also get the desired precedence order.

The highest weight a monomial can get from the weight function w would be $(\text{poly}(k, s, n, \delta))^{\log d}$. Thus, the cost of w remains $(\text{poly}(k, s, n, \delta))^{\log d}$. ∎

Combining Lemma 8 with Observation 7 and Lemma 6, we can get a hitting-set for ROABP.

THEOREM 1 (restated). *Let $C(\mathbf{x})$ be an n-variate polynomial computed by a width-w, s-sparse-factor ROABP, with individual degree bound δ. Then there is a $\text{poly}(w, s, n, \delta)^{\log n}$-time hitting-set for $C(\mathbf{x})$.*

Proof. As mentioned earlier, $C(\mathbf{x})$ can be written as $R \cdot D(\mathbf{x})$ for some $R \in \mathbb{M}_{w\times w}(\mathbb{F})$, where $D(\mathbf{x}) \in \mathbb{M}_{w\times w}(\mathbb{F})[\mathbf{x}]$. The underlying matrix algebra $\mathbb{M}_{w\times w}(\mathbb{F})$ has dimension w^2. The hitting-set size will be dominated by the cost of the weight function that was constructed in Lemma 8. As the parameter d in Lemma 8, i.e., the number of layers in the ROABP, is bounded by n, the hitting-set size will be $\text{poly}(w, s, n, \delta)^{\log n}$. ∎

4. Sum of constantly many set-multilinear circuits: Theorem 2. To find a hitting-set for a sum of constantly many set-multilinear circuits, we build some tools. The first is depth-3 multilinear circuits with "small distance." As it turns out, a multilinear polynomial computed by a depth-3 Δ-distance circuit (top fan-in k) can also be computed by a width-$O(kn^{\Delta})$ ROABP (Lemma 14). Thus, we get a $\text{poly}(nk)^{\Delta \log n}$-time hitting-set for this class from Theorem 1. For our main result (Theorem 2), we use only 1-distance circuits. However, we present our results for arbitrary distance circuits, as they are of independent interest and do not follow immediately from those for 1-distance.

Next, we use a general result about finding a hitting-set of size h^m for a class m-*base-sets*-C if a hitting-set of size h is known for class C (Lemma 17). A polynomial is in m-base-sets-C if there exists a partition of the variables into m base sets such that, restricted to each base set (treat other variables as field constants), the polynomial is in class C. Finally, we show that a sum of constantly many set-multilinear circuits falls into the class m-base-sets-Δ-distance for $m\Delta = o(n)$. Thus, we get Theorem 2.

4.1. $\boldsymbol{\Delta}$-distance circuits. Recall that each product gate in a depth-3 multilinear circuit induces a partition on the variables. Let these partitions be $\mathbb{P}_1, \mathbb{P}_2, \ldots, \mathbb{P}_k$.

DEFINITION 9 (distance for a partition sequence). *Let $\mathbb{P}_1, \mathbb{P}_2, \ldots, \mathbb{P}_k \in \text{Part}([n])$ be the k partitions of the variables $\{x_1, x_2, \ldots, x_n\}$. Then $\text{d}(\mathbb{P}_1, \mathbb{P}_2, \ldots, \mathbb{P}_k) = \Delta$ if for all $i \in \{2, 3, \ldots, k\}$, for all colors $Y_1 \in \mathbb{P}_i$, there exists $Y_2, Y_3, \ldots, Y_{\Delta'} \in \mathbb{P}_i$ $(\Delta' \le \Delta)$ such that $Y_1 \cup Y_2 \cup \cdots \cup Y_{\Delta'}$ equals a union of some colors in $\mathbb{P}_j$ for all $j \in [i-1]$.*

In other words, in every partition $\mathbb{P}_i$, each color Y_1 has a set of colors called a "friendly neighborhood," $\{Y_1, Y_2, \ldots, Y_{\Delta'}\}$, consisting of at most Δ colors, which is exactly partitioned in the "upper partitions." We call $\mathbb{P}_i$ an *upper* partition relative to $\mathbb{P}_j$ (and $\mathbb{P}_j$ a *lower* partition relative to $\mathbb{P}_i$) if $i < j$. For a color X_a of a partition $\mathbb{P}_j$, let $\text{nbd}_j(X_a)$ denote its friendly neighborhood. For example, for the partitions $\mathbb{P}_1 = \{\{1, 2\}, \{3\}, \{4\}, \{5, 6\}\}$ and $\mathbb{P}_2 = \{\{1, 3\}, \{2, 4\}, \{5\}, \{6\}\}$, $\text{d}(\mathbb{P}_1, \mathbb{P}_2) = 2$. In partition $\mathbb{P}_2$, the set $\{\{1, 3\}, \{2, 4\}\}$ is a friendly neighborhood, and the set $\{\{5\}, \{6\}\}$ is another.

The friendly neighborhood $\text{nbd}_j(x_i)$ of a variable x_i in a partition $\mathbb{P}_j$ is defined as $\text{nbd}_j(\text{color}_j(x_i))$, where $\text{color}_j(x_i)$ is the color in the partition $\mathbb{P}_j$ that contains the variable x_i.

DEFINITION 10 (Δ-distance circuits). *A multilinear depth-3 circuit C has Δ-distance if its product gates can be ordered to correspond to a partition sequence $(\mathbb{P}_1, \ldots, \mathbb{P}_k)$ with $\text{d}(\mathbb{P}_1, \mathbb{P}_2, \ldots, \mathbb{P}_k) \le \Delta$.*

For example, the circuit $(1+x_1+x_2)(1+x_3)(1+x_4)(1+x_5+x_6)+(1+x_1+x_3)(1+x_2+x_4)(1+x_5)(1+x_6)$ has 2-distance. Every depth-3 multilinear circuit is thus an n-distance circuit. A circuit with a partition sequence, where the partition $\mathbb{P}_i$ is a refinement of the partition $\mathbb{P}_{i+1}$ for all $i \in [k-1]$, exactly characterizes a 1-distance circuit. All depth-3 multilinear circuits have distance between 1 and n. Also observe that the circuits with 1-distance strictly subsume set-multilinear circuits. For example, a circuit whose product gates induce two different partitions $\mathbb{P}_1 = \{\{1\},\{2\},\dots,\{n\}\}$ and $\mathbb{P}_2 = \{\{1,2\},\{3,4\},\dots,\{n-1,n\}\}$ has 1-distance but is not set-multilinear.

Friendly neighborhoods. To get a better picture, we ask the following: Given a color X_a of a partition $\mathbb{P}_j$ in a circuit $D(\mathbf{x})$, how do we find its friendly neighborhood $\mathrm{nbd}_j(X_a)$? Consider a graph G_j which has the colors of the partitions $\{\mathbb{P}_1,\mathbb{P}_2,\dots,\mathbb{P}_j\}$ as its vertices. For all $i \in [j-1]$, there is an edge between the colors $X \in \mathbb{P}_i$ and $Y \in \mathbb{P}_j$ if they share at least one variable. Observe that if any two colors X_a and X_b of partition $\mathbb{P}_j$ are reachable from each other in G_j, then they should be in the same neighborhood. As reachability is an equivalence relation, *the neighborhoods are equivalence classes of colors.*

Moreover, observe that, for any two variables x_a and x_b, if their respective colors in partition $\mathbb{P}_j$, $\mathrm{color}_j(x_a)$ and $\mathrm{color}_j(x_b)$, are reachable from each other in G_j, then their respective colors in partition $\mathbb{P}_{j+1}$, $\mathrm{color}_{j+1}(x_a)$ and $\mathrm{color}_{j+1}(x_b)$, are also reachable from each other in G_{j+1}. Hence, we get the following.

OBSERVATION 11. *If at some partition the variables x_a and x_b are in the same neighborhood, then they will be in the same neighborhood in all of the lower partitions. That is,* $\mathrm{nbd}_j(x_a) = \mathrm{nbd}_j(x_b) \implies \mathrm{nbd}_i(x_a) = \mathrm{nbd}_i(x_b)$ *for all $i \geq j$.*

In other words, if we define a new sequence of partitions such that the jth partition has x_a and x_b in the same color if $\mathrm{nbd}_j(x_a) = \mathrm{nbd}_j(x_b)$, then the upper partitions are *refinements* of the lower partitions.

4.1.1. Reduction to ROABP. Now we show that any polynomial computed by a low-distance multilinear depth-3 circuit can also be computed by a small size ROABP. First we make the following observation about sparse polynomials.

OBSERVATION 12. *Any multilinear polynomial $C(\mathbf{x})$ with sparsity s can be computed by a width-s ROABP, in any variable order.*

Proof. Let $\mathcal{M}$ denote the set of monomials in C, and let C_m denote $\mathrm{coef}_C(m)$. Consider an ABP with $n+1$ layers of vertices $V_1, V_2, \dots, V_{n+1}$, each having s vertices (one for each monomial in $\mathcal{M}$) together with a start vertex v_0 and an end vertex v_{n+2}. Let $v_{i,m}$ denote the mth vertex of the layer V_i for any $i \in [n+1]$ and any $m \in \mathcal{M}$.

The edge labels in the ABP are given as follows: For all $m \in \mathcal{M}$,

- the edge $(v_0, v_{1,m})$ is labelled by C_m,
- the edge $(v_{n+1,m}, v_{n+2})$ is labelled by 1,
- for all $i \in [n]$, the edge $(v_{i,m}, v_{i+1,m})$ is labelled by x_i if the monomial m contains x_i, otherwise by 1.

All other edges get labelled by 0. Clearly, the ABP constructed computes the polynomial $P(\mathbf{x})$, and it is an ROABP.

Also, note that this construction can be done with any desired variable order. ∎

Now consider a depth-3 Δ-distance multilinear polynomial $P = \sum_{i=1}^{k} a_i Q_i$, where each $Q_i = \prod_{j=1}^{n_i} \ell_{ij}$ is a product of linear polynomials. We will construct an ROABP for each Q_i. We can combine these ROABPs to construct a single ROABP if they all have the same variable order. To achieve this we use the *refinement* property described above (from Observation 11).

LEMMA 13 (achieving same order). *Let $P = \sum_{i=1}^{k} a_i Q_i$ be a multilinear polynomial computed by a Δ-distance circuit. Then we can make a width-$O(n^{\Delta})$ ROABP for each Q_i, in the same variable order.*

Proof. Each Q_i is a product of linear forms in a disjoint set of variables, say $Q_i = \prod_{j=1}^{n_i} \ell_{ij}$. Let the partition induced on the variable set, by the product Q_i, be $\mathbb{P}_i$ for all $i \in [k]$. Without loss of generality let the partition sequence $(\mathbb{P}_1, \mathbb{P}_2, \ldots, \mathbb{P}_k)$ have distance Δ. For each $i \in [k]$, let us define a new partition $\mathbb{P}'_i$ such that the union of colors in each neighborhood of $\mathbb{P}_i$ forms a color of $\mathbb{P}'_i$. This is a valid definition, as neighborhoods are equivalence classes of colors. From Observation 11, the partition $\mathbb{P}'_i$ is a refinement of partition $\mathbb{P}'_j$ for any $i < j$.

For a partition $\mathbb{P}$ of the variable set $\mathbf{x}$, an ordering on its colors $(c_1 < c_2 < \cdots < c_r)$ naturally induces a partial ordering on the variables; i.e., for any $x_i \in c_j$ and $x_{i'} \in c_{j'}$, $c_j < c_{j'} \implies x_i < x_{i'}$. The variables in the same color do not have any relation.

Let us say that a variable (partial) order $(<^*)$ *respects* a partition $\mathbb{P}$ with colors $\{c_1, c_2, \ldots, c_r\}$ if there exists an ordering of the colors $(c_{j_1} < c_{j_2} < \cdots < c_{j_r})$ such that its induced partial order $(<)$ on the variables can be extended to $<^*$. We claim that there exists a variable order $(<^*)$ which *respects* partition $\mathbb{P}'_i$ for all $i \in [k]$.

We build this variable order $(<^*)$ iteratively. We start with $\mathbb{P}'_k$. We give an arbitrary ordering to the colors in $\mathbb{P}'_k$, say $(c_{k,1} < c_{k,2} < \cdots < c_{k,r_k})$, which induces a partial order $(<_k)$ on the variables. For any $k > i \geq 1$, let us define a partial order $(<_i)$ inductively as follows: Let $(<_{i+1})$ be a partial order on the variables induced by an ordering on the colors of $\mathbb{P}'_{i+1}$. As mentioned earlier, the colors of $\mathbb{P}'_i$ are just further partitions of the colors of $\mathbb{P}'_{i+1}$. Hence, we can construct an ordering on the colors of $\mathbb{P}'_i$, such that the induced partial order $(<_i)$ is an extension of $(<_{i+1})$. To achieve this, we do the following: For each color c in $\mathbb{P}'_{i+1}$, fix an arbitrary ordering among those colors of $\mathbb{P}'_i$ whose union forms c.

Clearly, the partial order $(<_1)$ defined in such a way respects $\mathbb{P}'_i$ for all $i \in [k]$. We further fix an arbitrary ordering among variables belonging to the same color in $\mathbb{P}'_1$. Thus, we get a total order $(<^*)$, which is an extension of $<_1$ and hence respects $\mathbb{P}'_i$ for all $i \in [k]$.

Now we construct an ROABP for each Q_i in the variable order $<^*$. First, we multiply out the linear forms which belong to the same neighborhood in each Q_i. That is, we write Q_i as the product $\prod_{j=1}^{r_i} Q_{ij}$, where r_i is the number of neighborhoods in $\mathbb{P}_i$ (number of colors in $\mathbb{P}'_i$) and each Q_{ij} is the product of linear forms (colors) which belong to the same neighborhood in $\mathbb{P}_i$. As the partition sequence has distance Δ, the neighborhoods have at most Δ colors. So, the degree of each Q_{ij} is bounded by Δ, and hence the sparsity is bounded by $O(n^{\Delta})$. By Observation 12, we can construct a width-$O(n^{\Delta})$ ROABP for Q_{ij} in the variable order given by $<^*$.

Let c_{ij} denote the color of $\mathbb{P}'_i$ corresponding to Q_{ij}. As the order $<^*$ respects $\mathbb{P}'_i$, it gives an order on its colors, say $c_{ij_1} < c_{ij_2} < \cdots < c_{ij_{r_i}}$. Now we arrange the ROABPs for Q_{ij}'s in the order $Q_{ij_1} Q_{ij_2} \ldots Q_{ij_{r_i}}$, while identifying the end vertex of Q_{ij_a} with the start vertex of $Q_{ij_{a+1}}$ for all $a \in [r_i - 1]$. Clearly the ROABP thus constructed computes the polynomial Q_i and has variable order $<^*$. ∎

Once we have ROABPs for the polynomials Q_i's in the same variable order, let us make a new start node and connect it with the start node of the ROABP for Q_i with label a_i for all $i \in [k]$. Also, let us make a new end node and connect it with the end node of the ROABP for Q_i with label 1 for all $i \in [k]$. Clearly, the ROABP thus constructed computes the polynomial $P = \sum_{i=1}^{k} a_i Q_i$ and has width $O(kn^{\Delta})$. Thus, we can write the following.

LEMMA 14 (Δ-distance to ROABP). *An n-variate multilinear polynomial computed by a depth-3, Δ-distance circuit with top fan-in k has a width-$O(kn^{\Delta})$ ROABP.*

Hence, from Theorem 1 we get the following theorem.

THEOREM 15 (Δ-distance hitting-set). *Let $C(\mathbf{x})$ be a depth-3, Δ-distance, n-variate multilinear circuit with top fan-in k. Then there is an $(nk)^{O(\Delta \log n)}$-time hitting-set for $C(\mathbf{x})$.*

4.2. Base sets with $\boldsymbol{\Delta}$-distance. In this section we describe our second tool toward finding a hitting-set for sum of constantly many set-multilinear polynomials. We further generalize the class of polynomials, for which we can give an efficient test, beyond low-distance. Basically, it is enough to have low-distance "projections."

DEFINITION 16. *A multilinear depth-3 circuit $C(\mathbf{x})$ is said to have m-base-sets-Δ-distance if there is a partition of the variable set $\mathbf{x}$ into base sets $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\}$ such that for any $i \in [m]$, restriction of C on the ith base set (i.e., other variables are considered as function field constants) has Δ-distance.*

For example, the circuit $(1 + x_1 + y_1)(1 + x_2 + y_2) \cdots (1 + x_n + y_n) + (1 + x_1 + y_2)(1 + x_2 + y_3) \cdots (1 + x_n + y_1)$ has n-distance, but when restricted to either base set $\{x_i\}_i$ or base set $\{y_i\}_i$, it has 1-distance. Thus, it has 2-base-sets-1-distance. We will show that there is an efficient hitting-set for this class of polynomials. In fact, we can show a general result for a polynomial whose restriction on one base set falls into a class C for which a hitting-set is already known. Here, by $\overline{\mathbb{F}}$ we mean the algebraic closure of a field $\mathbb{F}$. (Actually, a sufficiently large field extension of $\mathbb{F}$ suffices for the proof.)

LEMMA 17 (hybrid argument). *Let $\mathcal{H}$ be the hitting-set for a class C of n-variate polynomials over field $\mathbb{F}$. Let $\mathbf{x}$ be a union of m disjoint sets of variables $\mathbf{x}_1 \sqcup \mathbf{x}_2 \sqcup \cdots \sqcup \mathbf{x}_m$, called base sets, each with size at most n. Let $C(\mathbf{x})$ be a polynomial such that for all $i \in [m]$, its restriction to the base set $\mathbf{x}_i$ is in class C; i.e., for all points $(\mathbf{a}_1, \dots, \mathbf{a}_{i-1}, \mathbf{a}_{i+1}, \dots, \mathbf{a}_m) \in \overline{\mathbb{F}}^{\sum_{j \neq i} |\mathbf{x}_j|}$, the polynomial $C(\mathbf{x}_1 = \mathbf{a}_1, \dots, \mathbf{x}_{i-1} = \mathbf{a}_{i-1}, \mathbf{x}_{i+1} = \mathbf{a}_{i+1}, \dots, \mathbf{x}_m = \mathbf{a}_m)$ is in class C. Then there is a hitting-set for $C(\mathbf{x})$ of size $|\mathcal{H}|^m$ (with the knowledge of the base sets).*

Proof. Let us assume that the set $\mathbf{x}_i$ has cardinality n for all $i \in [m]$. If not, then we can introduce dummy variables. Now we claim that if $C(\mathbf{x}) \neq 0$, then there exist m points $\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_m \in \mathcal{H}$ such that $C(\mathbf{x}_1 = \mathbf{h}_1, \mathbf{x}_2 = \mathbf{h}_2, \mathbf{x}_m = \mathbf{h}_m) \neq 0$.

We prove the claim inductively.

Base case: The polynomial $C(\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m) \neq 0$. This follows from the assumption.

Induction hypothesis: There exist points $\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_i \in \mathcal{H}$ such that the partially evaluated polynomial $C'(\mathbf{x_{i+1}}, \dots, \mathbf{x}_m) := C(\mathbf{x}_1 = \mathbf{h}_1, \dots, \mathbf{x}_i = \mathbf{h}_i, \mathbf{x}_{i+1}, \dots, \mathbf{x}_m) \neq 0$.

Induction step: We show that there exists $\mathbf{h_{i+1}} \in \mathcal{H}$ such that the polynomial $C'(\mathbf{x}_{i+1} = \mathbf{h}_{i+1}, \mathbf{x}_{i+2}, \dots, \mathbf{x}_m) \neq 0$.

As the polynomial $C'(\mathbf{x}_{i+1}, \dots, \mathbf{x}_m)$ is nonzero, there exist points $\mathbf{a}_{i+2}, \dots, \mathbf{a}_m \in \overline{\mathbb{F}}^n$ such that $C''(\mathbf{x}_{i+1}) := C'(\mathbf{x}_{i+1}, \mathbf{x}_{i+2} = \mathbf{a}_{i+2}, \dots, \mathbf{x}_m = \mathbf{a}_m) \neq 0$ (from the Schwartz–Zippel lemma; see [45, Fact 4.1]). We know that $C''(\mathbf{x}_{i+1})$ is in class C. So, there must exist a point $\mathbf{h}_{i+1} \in \mathcal{H}$ such that $C''(\mathbf{x}_{i+1} = \mathbf{h}_{i+1}) \neq 0$. This clearly implies that $C'(\mathbf{x}_{i+1} = \mathbf{h}_{i+1}, \mathbf{x}_{i+2}, \dots, \mathbf{x}_m) \neq 0$. Thus, the claim is true.

Now, to construct a hitting-set for C, one needs to substitute the set $\mathcal{H}$ for each base set $\mathbf{x}_i$, i.e., the Cartesian product $\mathcal{H} \times \mathcal{H} \times \cdots \times \mathcal{H}$ (m times). Hence, we get a hitting-set of size $|\mathcal{H}|^m$. ∎

Note that in the above proof the knowledge of the base sets is crucial. This lemma, together with Theorem 15, gives us the following.

THEOREM 18 (m-base-sets-Δ-distance PIT). *If $C(\mathbf{x})$ is a depth-3 multilinear circuit, with top fan-in k, having m-base-sets (known) with Δ-distance, then there is an $(nk)^{O(m\Delta \log n)}$-time hitting-set for C.*

4.3. Sum of set-multilinear circuits reduces to m-base-sets-Δ-distance. In this section, we will reduce the PIT for sum of constantly many set-multilinear depth-3 circuits to the PIT for depth-3 circuits with m-base-sets-Δ-distance, where $m\Delta = o(n)$. Thus, we get a subexponential time whitebox algorithm for this class (from Theorem 18). Note that a sum of constantly many set-multilinear depth-3 circuits is equivalent to a depth-3 multilinear circuit such that the number of distinct partitions, induced by its product gates, is constant.

We first look at the case of two partitions. For a partition $\mathbb{P}$ of $[n]$, let $\mathbb{P}|_B$ denote the restriction of $\mathbb{P}$ on a base set $B \subseteq [n]$. E.g., if $\mathbb{P} = \{\{1,2\},\{3,4\},\{5,6,\ldots,n\}\}$ and $B = \{1,3,4\}$, then $\mathbb{P}|_B = \{\{1\},\{3,4\}\}$. Recall that $d(\mathbb{P}_1,\mathbb{P}_2,\ldots,\mathbb{P}_c)$ denotes the *distance* of the partition sequence $(\mathbb{P}_1,\mathbb{P}_2,\ldots,\mathbb{P}_c)$ (Definition 9). For a partition sequence $(\mathbb{P}_1,\mathbb{P}_2,\ldots\mathbb{P}_c)$ and a base set $B \subseteq [n]$, let $d_B(\mathbb{P}_1,\mathbb{P}_2,\ldots,\mathbb{P}_c)$ denote the distance of the partition sequence when restricted to the base set B, i.e., $d(\mathbb{P}_1|_B,\mathbb{P}_2|_B,\ldots,\mathbb{P}_c|_B)$.

LEMMA 19 ($c = 2$). *For any two partitions $\{\mathbb{P}_1,\mathbb{P}_2\}$ of the set $[n]$, there exists a partition of $[n]$, into at most $2\sqrt{n}$ base sets $\{B_1,B_2,\ldots,B_m\}$ ($m < 2\sqrt{n}$), such that for any $i \in [m]$, either $d_{B_i}(\mathbb{P}_1,\mathbb{P}_2) = 1$ or $d_{B_i}(\mathbb{P}_2,\mathbb{P}_1) = 1$.*

Proof. Let us divide the set of colors in the partition $\mathbb{P}_1$ into two types of colors: one with at least $\sqrt{n}$ elements and the other with fewer than $\sqrt{n}$ elements. In other words, $\mathbb{P}_1 = \{X_1,X_2,\ldots,X_r\} \cup \{Y_1,Y_2,\ldots,Y_q\}$ such that $|X_i| \geq \sqrt{n}$ and $|Y_j| < \sqrt{n}$ for all $i \in [r],\ j \in [q]$. Let us make each X_i a base set, i.e., $B_i = X_i$ for all $i \in [r]$. As $|X_i| \geq \sqrt{n}$ for all $i \in [r]$, we get $r \leq \sqrt{n}$. Now, for any $i \in [r]$, $\mathbb{P}_1|_{B_i}$ has only one color. Hence, irrespective of what colors $\mathbb{P}_2|_{B_i}$ has, $d_{B_i}(\mathbb{P}_2,\mathbb{P}_1) = 1$ for all $i \in [r]$.

Now, for the other kind of colors, we will make base sets which have exactly one element from each color Y_j. More formally, let $Y_j = \{y_{j,1},y_{j,2},\ldots,y_{j,r_j}\}$ for all $j \in [q]$. Let $r' = \max\{r_1,r_2,\ldots,r_q\}$ ($r' < \sqrt{n}$). Now define base sets $B'_1,B'_2,\ldots,B'_{r'}$ such that for any $a \in [r']$, $B'_a = \{y_{j,a} \mid j \in [q],\ |Y_j| \geq a\}$. In other words, all those Y_j's which have at least a elements contribute their ath element to B'_a. Now, for any $a \in [r']$, $\mathbb{P}_1|_{B'_a} = \{\{y_{j,a}\} \mid j \in [q],\ |Y_j| \geq a\}$; i.e., it has exactly one element in each color. Clearly, irrespective of what colors $\mathbb{P}_2|_{B'_a}$ has, $d_{B'_a}(\mathbb{P}_1,\mathbb{P}_2) = 1$ for all $a \in [r']$. $\{B_1,B_2,\ldots,B_r\} \cup \{B'_1,B'_2,\ldots,B'_{r'}\}$ is our final set of base sets. Clearly, they form a partition of $[n]$. The total number of base sets $m = r + r' < 2\sqrt{n}$. ∎

Now, we generalize Lemma 19 to any constant number of partitions by induction.

LEMMA 20 (reduction to m-base-sets-1-distance). *For any set of c partitions $\{\mathbb{P}_1,\mathbb{P}_2,\ldots,\mathbb{P}_c\} \subseteq \mathrm{Part}([n])$, there exists a partition of the set $[n]$ into m base sets $\{B_1,B_2,\ldots,B_m\}$ with $m < (4n)^{1-(1/2^{c-1})}$ such that for any $i \in [m]$, there exists a permutation of the partitions, $(\mathbb{P}_{i_1},\mathbb{P}_{i_2},\ldots,\mathbb{P}_{i_c})$ with $d_{B_i}(\mathbb{P}_{i_1},\mathbb{P}_{i_2},\ldots,\mathbb{P}_{i_c}) = 1$.*

Proof. Let $f(c,n) := (4n)^{1-(1/2^{c-1})}$. The proof is by induction on the number of partitions.

Base case: For $c = 2$, $f(c,n)$ becomes $2\sqrt{n}$. Hence, the statement follows from Lemma 19.

Induction hypothesis: The statement is true for any $c-1$ partitions.

Induction step: As in Lemma 19, we divide the set of colors in $\mathbb{P}_1$ into two types of colors. Let $\mathbb{P}_1 = \{X_1,X_2,\ldots,X_r\} \cup \{Y_1,Y_2,\ldots,Y_q\}$ such that $|X_i| \geq \sqrt{n}$ and $|Y_j| < \sqrt{n}$ for all $i \in [r],\ j \in [q]$. Let us set $B_i = X_i$, and let $n_i := |B_i|$ for all

$i \in [r]$. Our base sets will be further subsets of these B_i's. For a fixed $i \in [r]$, let us define $\mathbb{P}'_h = \mathbb{P}_h|_{B_i}$ as a partition of the set B_i for all $h \in [c]$. Clearly, $\mathbb{P}'_1$ has only one color. Now we focus on the partition sequence $(\mathbb{P}'_2, \mathbb{P}'_3, \ldots, \mathbb{P}'_c)$. From the inductive hypothesis, there exists a partition of B_i into m_i base sets $\{B_{i,1}, B_{i,2}, \ldots, B_{i,m_i}\}$ ($m_i < f(c-1, n_i)$) such that for any $u \in [m_i]$, there exists a permutation of $(\mathbb{P}'_2, \mathbb{P}'_3, \ldots, \mathbb{P}'_c)$, given by $(\mathbb{P}'_{i_2}, \mathbb{P}'_{i_3}, \ldots, \mathbb{P}'_{i_c})$, with $d_{B_{i,u}}(\mathbb{P}'_{i_2}, \mathbb{P}'_{i_3}, \ldots, \mathbb{P}'_{i_c}) = 1$. As $\mathbb{P}'_1$ has only one color, so does $\mathbb{P}'_1|_{B_{i,u}}$. Hence, $d_{B_{i,u}}(\mathbb{P}'_{i_2}, \mathbb{P}'_{i_3}, \ldots, \mathbb{P}'_{i_c}, \mathbb{P}'_1)$ is also 1. From this, we easily get $d_{B_{i,u}}(\mathbb{P}_{i_2}, \mathbb{P}_{i_3}, \ldots, \mathbb{P}_{i_c}, \mathbb{P}_1) = 1$. The above argument can be made for all $i \in [r]$.

Now for the other colors, we proceed as in Lemma 19. Let $Y_j = \{y_{j,1}, y_{j,2}, \ldots, y_{j,r_j}\}$ for all $j \in [q]$. Let $r' = \max\{r_1, r_2, \ldots, r_q\}$ ($r' < \sqrt{n}$). Now define sets $B'_1, B'_2, \ldots, B'_{r'}$ such that for any $a \in [r']$, $B'_a = \{y_{j,a} \mid j \in [q],\ |Y_j| \geq a\}$. In other words, all those Y_j's which have at least a elements contribute their ath element to B'_a. Let $n'_a := |B'_a|$ for all $a \in [r']$. Our base sets will be further subsets of these B'_a's. For a fixed $a \in [r']$, let us define $\mathbb{P}'_h = \mathbb{P}_h|_{B'_a}$ as a partition of the set B'_a for all $h \in [c]$. Clearly, $\mathbb{P}'_1$ has exactly one element in each of its colors. Now we focus on the partition sequence $(\mathbb{P}'_2, \mathbb{P}'_3, \ldots, \mathbb{P}'_c)$. From the inductive hypothesis, there exists a partition of B'_a into m'_a base sets $\{B'_{a,1}, B'_{a,2}, \ldots, B'_{a,m'_a}\}$ ($m'_a < f(c-1, n'_a)$) such that for any $u \in [m'_a]$, there exists a permutation of $(\mathbb{P}'_2, \mathbb{P}'_3, \ldots, \mathbb{P}'_c)$, given by $(\mathbb{P}'_{i_2}, \mathbb{P}'_{i_3}, \ldots, \mathbb{P}'_{i_c})$, with $d_{B'_{a,u}}(\mathbb{P}'_{i_2}, \mathbb{P}'_{i_3}, \ldots, \mathbb{P}'_{i_c}) = 1$. As $\mathbb{P}'_1$ has exactly one element in each of its colors, so does $\mathbb{P}'_1|_{B'_{a,u}}$. Hence, $d_{B'_{a,u}}(\mathbb{P}'_1, \mathbb{P}'_{i_2}, \mathbb{P}'_{i_3}, \ldots, \mathbb{P}'_{i_c})$ is also 1. From this, we easily get $d_{B'_{a,u}}(\mathbb{P}_1, \mathbb{P}_{i_2}, \mathbb{P}_{i_3}, \ldots, \mathbb{P}_{i_c}) = 1$. The above argument can be made for all $a \in [r']$.

Our final set of base sets will be $\{B_{i,u} \mid i \in [r],\ u \in [m_i]\} \cup \{B'_{a,u} \mid a \in [r'],\ u \in [m'_a]\}$. As argued above, when restricted to any of these base sets, the given partitions have a sequence which has distance 1. Now, we need to bound the number of these base sets,

$$m = \sum_{i \in [r]} m_i + \sum_{a \in [r']} m'_a.$$

From the bounds on m_i and m'_a, we get

$$m < \sum_{i \in [r]} f(c-1, n_i) + \sum_{a \in [r']} f(c-1, n'_a).$$

Recall that $n_i \geq \sqrt{n}$ for all $i \in [r]$ and $\sum_{i \in [r]} n_i \leq n$; thus $r < \sqrt{n}$. Also, we know that $r' < \sqrt{n}$. Let us combine the two sums by defining $n_{r+a} := n'_a$ for all $a \in [r']$ and $r'' := r + r'$:

$$m < \sum_{i \in [r'']} f(c-1, n_i).$$

We know that $r'' < 2\sqrt{n}$ and $\sum_{i \in [r'']} n_i = n$. Observe that $f(c-1, z)$, as a function of z, is a concave function (its derivative is monotonically decreasing when $z > 0$). From the properties of a concave function, we know that

$$\begin{aligned} \frac{1}{r''} \sum_{i \in [r'']} f(c-1, n_i) &\leq f\left(c-1, \frac{1}{r''} \sum_{i \in [r'']} n_i\right) \\ &= f\Big(c-1, \frac{n}{r''}\Big). \end{aligned}$$

Equivalently,

$$\begin{aligned}\sum_{i\in[r'']} f(c-1,n_i) &\le r''(4n/r'')^{1-(1/2^{c-2})}\\ &= (4n)^{1-(1/2^{c-2})}\cdot (r'')^{1/2^{c-2}}\\ &< (4n)^{1-(1/2^{c-2})}\cdot (4n)^{1/2^{c-1}} \quad (\because r'' < \sqrt{4n})\\ &= (4n)^{1-(1/2^{c-1})}.\end{aligned}$$

Thus we get

$$m < (4n)^{1-(1/2^{c-1})}. \qquad \square$$

Now we combine these results with our hitting-sets for depth-3 circuits having m-base-sets with Δ-distance.

THEOREM 2 (restated). *Let $C(\mathbf{x})$ be an n-variate polynomial which can be computed by a sum of c set-multilinear depth-3 circuits, each having top fan-in k. Then there is an $(nck)^{O(n^{1-\epsilon}\log n)}$-time whitebox PIT test for C, where $\epsilon := 1/2^{c-1}$.*

Proof. As mentioned earlier, the polynomial $C(\mathbf{x})$ can be viewed as being computed by a depth-3 multilinear circuit such that its product gates induce at most c distinct partitions. From Lemma 20, we can partition the variable set into m base sets such that for each of these base sets, the partitions can be sequenced to have distance 1, where $m < (4n)^{1-\epsilon}$. Hence, the polynomial C has m-base-sets with 1-distance and top fan-in ck. Moreover, from the proof of Lemma 20, it is clear that such base sets can be computed in $n^{O(c)}$-time. From Theorem 18, we know that there is an $(nck)^{O(m\log n)}$-time whitebox PIT test for such a circuit. As $m = O(n^{1-\epsilon})$, we get the result. $\square$

Tightness of this method for $c = 2$. Lemma 19 can be stated in other words as follows: Any two partitions have m-base-sets-Δ-distance with $m\Delta = O(\sqrt{n})$. We can, in fact, show that this result is tight.

Showing the lower bound: Let $d(\mathbb{P}_1,\mathbb{P}_2) = \Delta$. Then each color of $\mathbb{P}_2$ has a friendly neighborhood (of at most Δ colors) which is exactly partitioned in $\mathbb{P}_1$. Now construct Δ base sets such that ith base set takes the variables of ith color from every neighborhood of $\mathbb{P}_2$. Clearly, when restricted to one of these bases sets, $d(\mathbb{P}_1,\mathbb{P}_2)$ is 1. In other words, $\mathbb{P}_1$ and $\mathbb{P}_2$ have Δ-base-sets-1-distance. Similarly, one can argue that if $\mathbb{P}_1$ and $\mathbb{P}_2$ have m-base-sets-Δ-distance, then they also have $m\Delta$-base-sets-1-distance. Now we will show that if we want m-base-sets-1-distance for two partitions, then $m = \Omega(\sqrt{n})$.

Consider the following example (assuming that n is a square): $\mathbb{P}_1 = \{\{1,2,\ldots\sqrt{n}\}, \{\sqrt{n}+1,\sqrt{n}+2,\ldots,2\sqrt{n}\},\ldots,\{\sqrt{n}(\sqrt{n}-1)+1,\sqrt{n}(\sqrt{n}-1)+2,\ldots,n\}\}$ and $\mathbb{P}_2 = \{\{1,\sqrt{n}+1,\ldots,n-\sqrt{n}+1\},\{2,\sqrt{n}+2,\ldots,n-\sqrt{n}+2\},\ldots,\{\sqrt{n},2\sqrt{n},\ldots,n\}\}$. Basically, $\mathbb{P}_2$ has the residue classes (mod $\sqrt{n}$).

OBSERVATION 21. *A base set B, such that $d_B(\mathbb{P}_1,\mathbb{P}_2) = 1$, has at most $\sqrt{n}$ variables.*

Proof. Suppose it has more than $\sqrt{n}$ variables. Then there is at least one color in $\mathbb{P}_1$ which contributes two variables to B. These two variables have to be in two different colors of $\mathbb{P}_2$ (because of our design of $\mathbb{P}_1$ and $\mathbb{P}_2$). So, $d_B(\mathbb{P}_1,\mathbb{P}_2)$ is at least 2. We get a contradiction. $\square$

The number of such base sets has to be at least $\sqrt{n}$. Combining this with the reduction from m-base-sets-Δ-distance to $m\Delta$-base-sets-1-distance, we get $m\Delta = \Omega(\sqrt{n})$.

It is not clear whether Lemma 20 is tight. We conjecture that for any set of partitions, $m\Delta = O(\sqrt{n})$ can be achieved.

5. Sparse-invertible width-w ROABP: Theorem 3. As mentioned in the preliminaries (section 2), a polynomial $C(\mathbf{x})$ computed by an s-sparse-factor width-w ROABP can be written as $D_0^\top(\prod_{i=1}^d D_i)D_{d+1}$, where $D_0, D_{d+1} \in \mathbb{F}^w$, $D_i \in \mathbb{F}^{w\times w}[\mathbf{x}_i]$ is an s-sparse polynomial for all $i \in [d]$, and $\mathbf{x}_1, \mathbf{x}_2 \ldots, \mathbf{x}_d$ are disjoint sets of variables.

We will show a hitting-set for a sparse-factor ROABP $D_0(\prod_{i=1}^d D_i)D_{d+1}$, where the D_i's are *invertible* matrices for all $i \in [d]$. Hence, we name this model *sparse-invertible-factor ROABP*.

For a polynomial D, let its sparsity $\mathsf{s}(D)$ be the number of monomials in D with nonzero coefficients.

THEOREM 3 (restated). *Let $\mathbf{x} = \mathbf{x}_1 \sqcup \cdots \sqcup \mathbf{x}_d$, with $|\mathbf{x}| = n$. Let $C(\mathbf{x}) = D_0^\top D D_{d+1} \in \mathbb{F}[\mathbf{x}]$ be a polynomial where $D_0, D_{d+1} \in \mathbb{F}^w$, $D(\mathbf{x}) = \prod_{i=1}^d D_i(\mathbf{x}_i)$, and, for all $i \in [d]$, $D_i \in \mathbb{F}^{w\times w}[\mathbf{x}_i]$ is an invertible matrix. For all $i \in [d]$, D_i has degree bounded by δ and sparsity $\mathsf{s}(D_i) \leq s$. Then there is a hitting-set of size $\mathrm{poly}((n\delta s)^{w^2 \log w})$ for $C(\mathbf{x})$.*

REMARK 1. *If the width w is constant, then it is clear that we get a polynomial sized hitting-set.*

REMARK 2. *If the D_i's are univariate, then we get an $(n\delta)^{O(w^2)}$ sized hitting-set. The proof is presented along with the proof of Theorem 3.*

Like the authors of [5] and [14], we find a hitting-set by showing a *low-support concentration*. Low-support concentration in the polynomial $D(\mathbf{x}) = \prod_{i=1}^d D_i$ means that the coefficients of the low-support monomials in $D(\mathbf{x})$ span the whole coefficient space of $D(\mathbf{x})$.

Let $\mathbf{x}$ be $\{x_1, x_2 \ldots, x_n\}$. For an exponent $e = (e_1, e_2, \ldots, e_m) \in \mathbb{Z}_+^m$ and for a set of variables $\mathbf{y} = \{y_1, y_2, \ldots, y_m\}$, $\mathbf{y}^e$ will denote ${y_1}^{e_1}{y_2}^{e_2}\ldots{y_m}^{e_m}$. For any $e \in \mathbb{Z}_+^n$, support of the monomial $\mathbf{x}^e$ is defined as $\mathrm{S}(e) := \{i \in [n] \mid e_i \neq 0\}$, and support size is defined as $\mathrm{s}(e) := |\mathrm{S}(e)|$. Now, we define *$\ell$-concentration* for a polynomial $D(\mathbf{x}) \in \mathbb{F}^{w\times w}[\mathbf{x}]$.

DEFINITION 22 (ℓ-concentration). *Polynomial $D(\mathbf{x}) \in \mathbb{F}^{w\times w}[\mathbf{x}]$ is ℓ-concentrated if* $\mathrm{rank}_{\mathbb{F}}\{\mathrm{coef}_D(\mathbf{x}^e) \mid e \in \mathbb{Z}_+^n,\ \mathrm{s}(e) < \ell\} = \mathrm{rank}_{\mathbb{F}}\{\mathrm{coef}_D(\mathbf{x}^e) \mid e \in \mathbb{Z}_+^n\}$.

We will see later that the low-support concentration in polynomial $D(\mathbf{x})$ implies low-support concentration in polynomial $C(\mathbf{x})$ (defined similarly). In other words, $C(\mathbf{x})$ will have a nonzero coefficient for at least one of the low-support monomials. Thus, we get a hitting-set by testing these low-support coefficients. We use the following lemma from [5].

LEMMA 23. *If $C(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ is an n-variate, ℓ-concentrated polynomial with highest individual degree δ, then there is an $(n\delta)^{O(\ell)}$-time hitting-set for $C(\mathbf{x})$.*

Proof. ℓ-concentration for $C(\mathbf{x})$ simply means that it has at least one $(< \ell)$-support monomial with nonzero coefficient. We will construct a hitting-set which essentially will test all these $(< \ell)$-support coefficients. We go over all subsets S of $\mathbf{x}$ with size $\ell - 1$ and do the following: Substitute 0 for all the variables outside the set S. There will be at least one choice of S for which the polynomial $C(\mathbf{x})$ remains nonzero after the substitution. Now it is an $(\ell-1)$-variate nonzero polynomial. We take the usual hitting-set $\mathcal{H}^{\ell-1}$ for this, where $\mathcal{H} \subseteq \mathbb{F}$ is a set of size $\delta + 1$ (see, for example, [45, Fact 4.1]). In other words, each of these $\ell - 1$ variables is assigned a value from the set $\mathcal{H}$.

The number of sets S we need to try is $\binom{n}{\ell-1}$. Hence, the overall hitting-set size is $(n\delta)^{O(\ell)}$. $\square$

It is known that low-support concentration in $D(\mathbf{x})$ gives low-support concentration in $C(\mathbf{x})$ (Lemma 33). Now we move on to show how to achieve low-support concentration in $D(\mathbf{x}) = \prod_{i=1}^{d} D_i$. To achieve this we will use some efficient shift. By shifting by a tuple $\alpha := (\alpha_1, \alpha_2, \ldots, \alpha_n)$, we mean replacement of x_i with $x_i + \alpha_i$. Note that $C(\mathbf{x} + \alpha) \neq 0$ if and only if $C(\mathbf{x}) \neq 0$. Hence, a hitting-set for $C(\mathbf{x} + \alpha)$ gives us a hitting-set for $C(\mathbf{x})$. Instead of constants, we will shift $C(\mathbf{x})$ by univariate polynomials, say, given by the map $\phi\colon \mathbf{t} \to \{t^a\}_{a\geq 0}$, where $\mathbf{t} := \{t_1, t_2, \ldots, t_n\}$. The ϕ is said to be an efficient map if $\phi(t_i)$ is efficiently computable for each $i \in [n]$.

Block-support. Let the matrix product $D(\mathbf{x}) := \prod_{i=1}^{d} D_i$ correspond to an ROABP such that $D_i \in \mathbb{F}^{w\times w}[\mathbf{x}_i]$ for all $i \in [d]$. Let n_i be the cardinality of $\mathbf{x}_i$, and let $n = \sum_{i=1}^{d} n_i$. Viewing D_i as belonging to $\mathbb{F}^{w\times w}[\mathbf{x}_i]$, one can write $D_i := \sum_{e\in\mathbb{Z}_+^{n_i}} D_{ie}\mathbf{x}_i^e$, where $D_{ie} \in \mathbb{F}^{w\times w}$ for all $e \in \mathbb{Z}_+^{n_i}$. In particular, $D_{i\mathbf{0}}$ refers to the constant part of the polynomial D_i.

Any monomial $\mathbf{x}^e$ for $e \in \mathbb{Z}_+^n$ can be seen as a product $\prod_{i=1}^{d} {\mathbf{x}_i}^{e_i}$, where $e_i \in \mathbb{Z}_+^{n_i}$ for all $i \in [d]$, such that $e = (e_1, e_2, \ldots, e_d)$. We define *block-support* of e, $\mathrm{bS}(e)$, as $\{i \in [d] \mid e_i \neq \mathbf{0}\}$ and *block-support size* of e, $\mathrm{bs}(e) = |\mathrm{bS}(e)|$. Thus, the block-support of a monomial is the set of blocks which contribute nontrivially to the monomial.

The coefficient of the monomial $\mathbf{x}^e$ is $D_e := \prod_{i=1}^{d} D_{ie_i}$. Observe that when $i \notin \mathrm{bS}(e)$, the ith block contributes its constant part, $D_{i\mathbf{0}}$, to the coefficient. The coefficient D_f is called a *substring* of the coefficient D_e if $\mathrm{bs}(f) < \mathrm{bs}(e)$ and $f_i = e_i$ whenever $f_i \neq \mathbf{0}$. We will use the operator substrings(e) to denote the set $\{f \mid D_f \text{ is a substring of } D_e\}$.

Consider the toy example $D = (A_1 + B_{11}x_1 + B_{12}{x_1}^2)(A_2 + B_2x_2)(A_3 + B_3x_3 + B_4x_3x_4)$. Here, $D_{i\mathbf{0}} = A_i$ for all i, and the block-support of the monomial $x_1x_3x_4$ is $\{1, 3\}$. $A_1B_2B_3$ is a substring of $B_{12}B_2B_3$, whereas $B_{11}B_2B_3$ is not.

A substring D_f of D_e is called its *prefix* if for all i ($f_i \neq e_i \implies (\forall j \geq i, f_j = \mathbf{0})$). In the above example, it means that only the trailing B's can be replaced with the corresponding A's. Thus, $B_{12}A_2A_3$ is a prefix of $B_{12}B_2B_4$, whereas $B_{12}A_2B_4$ is not, though both are its substrings. Observe that when the $D_{i\mathbf{0}}$'s are invertible, if D_f is a prefix of D_e, then we can write $D_e = D_fA^{-1}B$, where $A := \prod_{i=r+1}^{d} D_{i\mathbf{0}}$ and $B := \prod_{i=r+1}^{d} D_{ie_i}$ with $r = \max \mathrm{bS}(f)$. We will denote the product $A^{-1}B$ as $D_{f^{-1}e}$.

Proof idea. Let $\ell = w^2$ and $\ell' = \log w^2 + 1$. There are four steps that we take to prove low-support concentration in the sparse-invertible-factor, width-w ROABP D.

1. Given an invertible matrix $D_i \in \mathbb{F}^{w\times w}[\mathbf{x}_i]$, we use a shift, so that the constant part of the shifted matrix is invertible.
2. Assuming the constant part of the shifted matrix D is invertible, we show that any coefficient with block-support ($= \ell$) is linearly dependent on its substrings with ($< \ell$) block-support (Lemma 25).
3. We use this lemma to show that any coefficient in D is linearly dependent on its substrings with $< \ell$ block-support (Lemma 26).
4. We show that the shifted polynomial has ℓ'-concentration within each of the blocks.

The contribution/novel part of this section is in the second step. The remaining steps were already known.

We will give the proof idea for the second step through a toy example. Consider $D = (A_1 + B_1x_1)(A_2 + B_2x_2)\cdots(A_n + B_nx_n)$, where each of the A_i's is invertible and $A_i, B_i \in \mathbb{F}^{w\times w}$ for all i. Take the ℓ-block-support monomial $x_{[\ell]} := x_1x_2\cdots x_\ell$. We will show that its coefficient is linearly dependent on its substrings with ($< \ell$)

block-support. Let $M_j = \prod_{i=1}^{j} B_i \prod_{i=j+1}^{\ell} A_i$ for $0 \le j \le \ell$. Consider the set of matrices $\{M_j\}_{j=0}^{\ell}$. These $\ell+1$ matrices lie in $\mathbb{F}^{\ell}$. Hence, there exists an $r \in [\ell]$ such that $M_r = \sum_{j=0}^{r-1} \alpha_j M_j$, where $\alpha_j \in \mathbb{F}$. All the M_j's on the right-hand side have $< r$ many B's. Since the A_i's are invertible, we can postmultiply throughout by $(\prod_{i=r+1}^{\ell} A_i)^{-1} \prod_{i=r+1}^{\ell} B_i \prod_{i=\ell+1}^{n} A_i$ to obtain that the coefficient of $x_{[\ell]}$ is linearly dependent on strictly smaller block-support coefficients.

The third step is a simple extension of the above idea. For the first and fourth steps, we use an appropriate shift (section 5.1). The sparsity of D_i is used crucially here.

Note that we have to assume that $D_i(\mathbf{x}_i)$ is an invertible matrix for all $i \in [d]$. For the shifted polynomial $D_i'(\mathbf{x}_i) := D_i(\mathbf{x}_i + \phi(\mathbf{t}_i))$, its constant term $D_{i\mathbf{0}}'$ is just an evaluation of $D_i(\mathbf{x})$, i.e., $D_i|_{\mathbf{x}_i=\phi(\mathbf{t}_i)}$. Hence, if $\det(D_i(\mathbf{x}_i)) = 0$ (viewing $D_i(\mathbf{x}_i)$ as an element in $(\mathbb{F}[\mathbf{x}_i])^{w\times w}$), then $\det(D_{i\mathbf{0}}') = 0$. This means that if $\det(D_i(\mathbf{x}_i)) = 0$, then even after shifting, $D_{i\mathbf{0}}'$ cannot become invertible.

Let us now prove that a particular kind of dependency can be lifted.

LEMMA 24. *Let D_e be a prefix of D_{e^*}. If D_e is linearly dependent on its substrings, then D_{e^*} is linearly dependent on its substrings.*

Proof. Since D_e is a prefix of D_{e^*},

$$D_{e^*} = D_e D_{e^{-1}e^*}. \tag{3}$$

Let the dependence of D_e on its substrings be the following:

$$D_e = \sum_{f \in \text{substrings}(e)} \alpha_f D_f.$$

Using (3) we can write

$$D_{e^*} = \sum_{f \in \text{substrings}(e)} \alpha_f D_f D_{e^{-1}e^*}.$$

Now we just need to show that for any substring D_f of D_e, $D_f D_{e^{-1}e^*}$ is a valid coefficient of some monomial in $D(\mathbf{x})$ and also that it is a substring of D_{e^*}.

Let $r = \max\{\text{bS}(e)\}$. Recall that $D_{e^{-1}e^*} = A^{-1}B$, where $A := \prod_{i=r+1}^{d} D_{i\mathbf{0}}$ and $B := \prod_{i=r+1}^{d} D_{ie_i^*}$. Thus, $D_f D_{e^{-1}e^*}$ is the coefficient of $\mathbf{x}^{f^*} := \prod_{i=1}^{r} \mathbf{x}_i{}^{f_i} \prod_{i=r+1}^{d} \mathbf{x}_i{}^{e_i^*}$. Since $f \in \text{substrings}(e)$ and $e \in \text{substrings}(e^*)$, $f \in \text{substrings}(e^*)$. From these two facts, it is easy to see that $D_f D_{e^{-1}e^*}$ is a substring of D_{e^*}. ∎

We will now prove the existence of a dependency for any ℓ-block-support coefficient.

LEMMA 25. *Let D_e be a coefficient in D with* $\text{bs}(e) = \ell$. *Then D_e $\mathbb{F}$-linearly depends on its substrings.*

Proof. Consider the set of coefficients $\{M_0, M_1, \ldots, M_\ell\}$, where M_j is the prefix of D_e with block-support size j for $0 \le j < \ell$ and where $M_\ell = D_e$. These $\ell+1$ vectors lie in $\mathbb{F}^{\ell} \cong \mathbb{F}^{w\times w}$. Hence, there exists an $r \in [\ell]$ such that M_r is linearly dependent on $\{M_j\}_{j=0}^{r-1}$. (Note that $M_r = \mathbf{0}$ is also a dependency. Also note that $r > 0$, since M_0 is invertible.) The coefficients in the set $\{M_j\}_{j=0}^{r-1}$ are prefixes of M_r and, thus, substrings of M_r.

Now, by applying Lemma 24, we conclude that D_e is dependent on its substrings. ∎

We will now prove that any coefficient in $D(\mathbf{x})$ is dependent on coefficients in D with block-support $\leq \ell - 1$. We refer to this as *ℓ-block-concentration* in $D(\mathbf{x})$.

LEMMA 26 (ℓ-block-concentration). *Let $D(\mathbf{x}) = \prod_{i=1}^{d} D_i(\mathbf{x}_i) \in \mathbb{F}^{w \times w}[\mathbf{x}]$ be a polynomial with $D_{i\mathbf{0}}$ being invertible for each $i \in [d]$. Then $D(\mathbf{x})$ has ℓ-block-concentration.*

Proof. We will actually prove that for any coefficient D_e with $\mathrm{bs}(e) \geq \ell$ (the case when $\mathrm{bs}(e) < \ell$ is trivial),

$$D_e \in \mathrm{span}\{D_f \mid f \in \mathbb{Z}_+^n,\ f \in \mathrm{substrings}(e) \text{ and } \mathrm{bs}(f) \leq \ell - 1\}.$$

We will prove this by induction on the block-support of D_e, $\mathrm{bs}(e)$.

Base case: When $\mathrm{bs}(e) = \ell$, this has been already shown in Lemma 25.

Induction hypothesis: For any coefficient D_e with $\mathrm{bs}(e) = i - 1$ for $i - 1 \geq \ell$,

$$D_e \in \mathrm{span}\{D_f \mid f \in \mathbb{Z}_+^n,\ f \in \mathrm{substrings}(e) \text{ and } \mathrm{bs}(f) \leq \ell - 1\}.$$

Induction step: Let us take a coefficient D_e with $\mathrm{bs}(e) = i$. Consider the unique prefix $D_{e'}$ of D_e such that $\mathrm{bs}(e') = i - 1$.

As $\mathrm{bs}(e') = i - 1$, by our induction hypothesis, $D_{e'}$ is linearly dependent on its substrings. So, from Lemma 24, D_e is linearly dependent on its substrings. In other words,

$$D_e \in \mathrm{span}\{D_f \mid f \in \mathrm{substrings}(e) \text{ and } \mathrm{bs}(f) \leq i - 1\}. \tag{4}$$

Again, by our induction hypothesis, for any coefficient D_f, with $\mathrm{bs}(f) \leq i - 1$,

$$D_f \in \mathrm{span}\{D_g \mid g \in \mathrm{substrings}(f) \text{ and } \mathrm{bs}(g) \leq \ell - 1\}. \tag{5}$$

Combining (4) and (5), we get

$$D_e \in \mathrm{span}\{D_g \mid g \in \mathrm{substrings}(e) \text{ and } \mathrm{bs}(g) \leq \ell - 1\}. \qquad \square$$

In Lemma 26, we had assumed that the constant term $D_{i\mathbf{0}}$ is invertible for every block D_i. In the next subsection, we will show how to achieve this invertibility and low-support concentration within each block D_i.

5.1. Achieving invertibility and low-support concentration through shifting. Let $D' := D(\mathbf{x} + \phi(\mathbf{t}))$. Then, $D' = \prod_{i=1}^{d} D'_i$, and $D'_{i\mathbf{0}}$ is the constant part of D'_i. Shifting will serve two purposes.

- Recall that for Lemma 26, we need invertibility of the constant term $D'_{i\mathbf{0}}$ in D'_i for all $i \in [d]$.
- D'_i should have low-support concentration after shifting.

Now we want a shift for D_i which would ensure that $\det(D'_{i\mathbf{0}}) \neq 0$ and that D'_i has low-support concentration. For both goals we use the sparsity of the polynomial.

DEFINITION 27. *For a polynomial p, let its sparsity set $\mathsf{S}(p)$ be the set of monomials in p with nonzero coefficients, and let $\mathsf{s}(p)$ be its sparsity, i.e., $\mathsf{s}(p) = |\mathsf{S}(p)|$. Let $\mathsf{S}^w(p) := \{m_1 m_2 \cdots m_w \mid m_i \in \mathsf{S}(p) \text{ for all } i \in [w]\}$.*

A map ϕ over $\mathbf{t}$ *separates* all the monomials in a set S if for any two monomials $\mathbf{t}^{e_1}, \mathbf{t}^{e_2} \in S$, $\phi(\mathbf{t}^{e_1}) \neq \phi(\mathbf{t}^{e_2})$.

Let us now characterize the shift which makes the determinant of $D'_{i\mathbf{0}}$ nonzero.

LEMMA 28. *Suppose D_i is invertible. Let $\phi\colon \mathbf{t} \to \{t^i\}_{i=0}^{\infty}$ be a monomial map which separates all the monomials in $\mathsf{S}^w(D_i)$. Then the constant term $D'_{i\mathbf{0}}$ of the shifted polynomial $D'_i := D_i(\mathbf{x} + \phi(\mathbf{t}))$ is invertible.*

Proof. Observe that $\mathsf{S}(\det(D_i)) \subseteq \mathsf{S}^w(D_i)$.

Since ϕ separates all the monomials in $\det(D_i(\mathbf{t}))$, $\det(D_i) \neq 0$ implies that $\det(D_i|_{\mathbf{x}=\phi(\mathbf{t})}) \neq 0$. Hence, $\det(D'_{i\mathbf{0}}) = \det(D_i|_{\mathbf{x}=\phi(\mathbf{t})}) \neq 0$. ∎

Gurjar et al. proved in [18, Lemma 19] that shifting by a basis isolating weight assignment gives concentration. The proof of Lemma 30 uses this.

LEMMA 29 (isolation to concentration [18]). *Let $D(\mathbf{x})$ be a polynomial over a k-dimensional algebra $\mathbb{A}_k$. Let* w *be a basis isolating weight assignment for D (Definition 5). Then, $D(\mathbf{x} + t^{\mathrm{w}})$ has $\lceil \log(k+1) \rceil$-concentration.*

Recall that $\ell' := \log(w^2) + 1$.

LEMMA 30. *Let $\phi\colon \mathbf{t} \to \{t^i\}_{i=0}^{\infty}$ be a monomial map which separates all the monomials in $\mathsf{S}(D_i)$. Then, $D'_i := D_i(\mathbf{x} + \phi(\mathbf{t}))$ is ℓ'-concentrated.*

Proof. D_i is a polynomial over a w^2-dimensional algebra, $\mathbb{F}^{w\times w}$. A map ϕ which separates all the monomials in $\mathsf{S}(D_i)$ is trivially a basis isolating weight assignment for D_i. Thus, by Lemma 29, $D_i(\mathbf{x} + \phi(\mathbf{t}))$ is $(\log w^2 + 1)$-concentrated. ∎

We will now show how to find such a map ϕ.

LEMMA 31. *Let $D(\mathbf{x}) = \prod_{i=1}^{d} D_i(\mathbf{x}_i)$ be a polynomial in $\mathbb{F}^{w\times w}[\mathbf{x}]$ such that for all $i \in [d]$, $\det(D_i) \neq 0$, D_i has degree bounded by δ and the sparsity $\mathsf{s}(D_i) \leq s$. Let $M := n^2 s^{2w} \log(w\delta)$. There is a set of M monomial maps with degree bounded by $2M \log M$ such that for at least one of the maps ϕ, all D'_i's are ℓ'-concentrated and all $D'_{i\mathbf{0}}$'s are invertible, where $D' = D(\mathbf{x} + \phi(\mathbf{t}))$.*

Proof. We will provide a map ϕ that satisfies the preconditions of Lemmas 28 and 30. This will ensure that all D'_i's are ℓ'-concentrated and all $D'_{i\mathbf{0}}$'s are invertible.

Observe that a map which separates all the monomials in $\mathsf{S}^w(D_i)$ also separates all the monomials in $\mathsf{S}(D_i)$. This can be proved by considering the two monomials $M_1 := m_1 m_2 \ldots m_w$ and $M'_1 = m'_1 m_2 \ldots m_w$, where $m_j \in \mathsf{S}(D_i)$ for all $j \in [w]$ and $m'_1 \in \mathsf{S}(D_i)$. $M_1, M'_1 \in \mathsf{S}^w$. Thus, if ϕ separates M_1 and M'_1, then ϕ should also separate m_1 and m'_1.

Hence, it is enough if ϕ separates all the monomial pairs in $\mathsf{S}^w(D_i)$ for $i \in [d]$ simultaneously. There are n variables, the number of monomial pairs is $\leq d \cdot s^{2w} \leq n \cdot s^{2w}$, and the degree of the monomials (in the determinant of D_is) is bounded by $w \cdot \delta$. Hence, by Lemma 4, $M = n^2 s^{2w} \log(w\delta)$ suffices. ∎

5.2. Concentration in $D(\mathbf{x})$. Now we want to show that if $D(\mathbf{x}) = \prod_{i=1}^{d} D_i$ has low-block-concentration, and, moreover, if each D_i has low-support concentration, then $D(\mathbf{x})$ has an appropriate low-support concentration.

LEMMA 32 (composition). *If $D(\mathbf{x})$ has ℓ-block-concentration and $D_i(\mathbf{x}_i)$ has ℓ'-support concentration for all $i \in [d]$, then $D(\mathbf{x})$ has $\ell\ell'$-support concentration.*

Proof. We have to prove that for any monomial $\mathbf{x}^e$,

$$D_e := \mathrm{coef}_D(\mathbf{x}^e) \in \mathrm{span}\left\{\mathrm{coef}_D(\mathbf{x}^g) \mid \mathrm{s}(g) < \ell\ell'\right\}.$$

Since $D(\mathbf{x})$ has ℓ-block-concentration,

$$D_e \in \mathrm{span}\left\{D_f \mid \mathrm{bs}(f) < \ell\right\}. \tag{6}$$

Recall that as D_i's are polynomials over disjoint sets of variables, any coefficient D_f in $D(\mathbf{x})$ can be written as

$$D_f = \prod_{i=1}^{d} D_{if_i}, \tag{7}$$

where $f = (f_1, f_2, \ldots, f_d)$ and D_{if_i} is the coefficient corresponding to the monomial $\mathbf{x}_i^{f_i}$ in D_i for all $i \in [d]$. Also, $|\{i : f_i \neq \mathbf{0}\}| < \ell$.

From ℓ'-support concentration of $D_i(\mathbf{x}_i)$, we know that for any coefficient D_{if_i},

$$D_{if_i} \in \operatorname{span}\{D_{ig_i} \mid g_i \in \mathbb{Z}_+^{n_i},\ \mathrm{s}(g_i) \leq \ell' - 1\}. \tag{8}$$

Using (7) and (8), we can write the following for any coefficient D_f:

$$D_f \in \operatorname{span}\left\{\prod_{i=1}^{d} D_{ig_i} \mid g_i \in \mathbb{Z}_+^{n_i},\ \mathrm{s}(g_i) \leq \ell' - 1\ \forall i \in [d] \right.$$
$$\left. \text{and } g_i = \mathbf{0}\ \forall i \notin \mathrm{bS}(f)\right\}.$$

Note that the product $\prod_{i=1}^{d} D_{ig_i}$ will be the coefficient of a monomial $\mathbf{x}^g$ such that $\mathrm{bS}(g) \subseteq \mathrm{bS}(f)$ because $g_i = \mathbf{0}$ for all $i \notin \mathrm{bS}(f)$. Clearly, if $\mathrm{s}(g_i) \leq \ell' - 1$ for all $i \in \mathrm{bS}(f)$, then $\mathrm{s}(g) \leq (\ell' - 1)\,\mathrm{bs}(f)$. So, one can write

$$D_f \in \operatorname{span}\{D_g \mid g \in \mathbb{Z}_+^{n},\ \mathrm{s}(g) \leq (\ell' - 1)\,\mathrm{bs}(f)\}. \tag{9}$$

Using (6) and (9), we can write for any coefficient D_e

$$D_e \in \operatorname{span}\{D_g \mid g \in \mathbb{Z}_+^{n},\ \mathrm{s}(g) \leq (\ell' - 1)(\ell - 1)\}. \qquad \square$$

We are now ready to go to the final step of the proof—getting the actual hitting-set.

5.3. From concentration to hitting-set. Let $C(\mathbf{x}) = D_0^\top D D_{d+1} \in \mathbb{F}[\mathbf{x}]$. Since any coefficient $\mathrm{coef}_C(\mathbf{x}^f)$ in C can be written as $D_0^\top \mathrm{coef}_D(\mathbf{x}^f) D_{d+1}$, we get the following lemma.

LEMMA 33 (concentration in C). *Let $C(\mathbf{x}) = D_0^\top D D_{d+1} \in \mathbb{F}[\mathbf{x}]$ be a polynomial with $D_0, D_{d+1} \in \mathbb{F}^w$. If $D(\mathbf{x})$ has $\ell\ell'$-concentration, then $C(\mathbf{x})$ has $\ell\ell'$-concentration.*

Now we come back to the proof of Theorem 3. From Lemmas 31, 26, 32, and 33, we get $\mathrm{poly}(ns^w log(w\delta))$ maps, such that for at least one map $\phi : \mathbf{t} \to \{t_i\}_{i=0}^{\infty}$, $C'(\mathbf{x}) = C(\mathbf{x} + \phi(\mathbf{t}))$ is $(w^2(\log w^2 + 1))$-concentrated. The degree of t is bounded by $\mathrm{poly}(ns^w log(w\delta))$. Hence, by Lemma 23 we get a hitting-set for $C'(\mathbf{x}) = C(\mathbf{x} + \phi(\mathbf{t}))$ of size $(n\delta)^{O(w^2 \log w)}$. Each of these evaluations of C will be a polynomial in t with degree bounded by $\mathrm{poly}(ns^w log(w\delta))$. Hence, total time complexity becomes $\mathrm{poly}(s^w (n\delta)^{w^2 \log w})$.

For the proof of Remark 2, observe that when the D_i's are univariate, D_i''s are 1-concentrated and sparsity $s \leq \delta$. Thus, when the D_i's are univariate, we get an $(n\delta)^{O(w^2)}$ sized hitting-set.

5.4. Width-2 ROABP. In the previous section, the crucial part in finding a hitting-set for an ROABP is the assumption that the matrix product $D(\mathbf{x})$ is invertible. Now we will show that for width-2 ROABP, this assumption is not required. Via a factorization property of 2×2 matrices, we will show that PIT for width-2 sparse-factor ROABPs reduces to PIT for width-2 sparse-invertible-factor ROABPs. This factorization of width-2 ABPs has also been studied by Allender and Wang [6], but their reduction cannot maintain the sparsity of the matrix entries.

LEMMA 34 (2×2 invertibility). *Let $C(\mathbf{x}) = D_0^\top (\prod_{i=1}^{d} D_i) D_{d+1}$ be a polynomial computed by a width-2 sparse-factor ROABP. Then for some nonzero $\alpha \in \mathbb{F}[\mathbf{x}]$ and* *some $m \le d$, we can write $\alpha(\mathbf{x})C(\mathbf{x}) = C_1(\mathbf{x})C_2(\mathbf{x})\cdots C_{m+1}(\mathbf{x})$, where each of the C_i's are of the form $P^\top{}_i Q_i R_i$. $Q_i \in \mathbb{F}^{2\times 2}[\mathbf{x}]$ is a polynomial computed by a width-2 sparse-invertible-factor ROABP, $P_i, R_i \in \mathbb{F}^2[\mathbf{x}]$, and P_i, Q_i, and R_i are over disjoint sets of variables for all $i \in [m+1]$.*

Proof. Let us say, for some $i \in [d]$, that $D_i(\mathbf{x}_i)$ is not invertible. Let $D_i = \left[\begin{smallmatrix} a_i & b_i \\ c_i & d_i \end{smallmatrix}\right]$ with $a_i, b_i, c_i, d_i \in \mathbb{F}[\mathbf{x}_i]$ and $a_i d_i = b_i c_i$. Without loss of generality, at least one of $\{a_i, b_i, c_i, d_i\}$ is nonzero. Let us say that $a_i \ne 0$ (other cases are similar). Then we can write

$$\begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix} = \frac{1}{a_i}\begin{bmatrix} a_i \\ c_i \end{bmatrix}\begin{bmatrix} a_i & b_i \end{bmatrix}.$$

In other words, we can write $\alpha_i D_i = A_i B_i^\top$, where $A_i, B_i \in \mathbb{F}^2[\mathbf{x}_i]$ and $0 \ne \alpha_i \in \{a_i, b_i, c_i, d_i\}$. Note that $\mathsf{s}(\alpha_i), \mathsf{s}(A_i), \mathsf{s}(B_i) \le \mathsf{s}(D_i)$. Let us say that the set of non-invertible D_i's is $\{D_{i_1}, D_{i_2}, \ldots, D_{i_m}\}$. Writing all of them in the above form, we get

$$C(\mathbf{x})\prod_{j=1}^{m}\alpha_{i_j} = \prod_{j=1}^{m+1} C_j,$$

where

$$C_j := \begin{cases} D_0^\top\left(\prod_{i=1}^{i_1-1} D_i\right)A_{i_1} & \text{if } j = 1, \\ B_{i_{j-1}}^\top\left(\prod_{i=i_{j-1}+1}^{i_j-1} D_i\right)A_{i_j} & \text{if } 2 \le j \le m, \\ B_{i_m}^\top\left(\prod_{i=i_m+1}^{d} D_i\right)D_{d+1} & \text{if } j = m+1. \end{cases}$$

Clearly, for all $j \in [m+1]$, $(\prod_{i=i_{j-1}+1}^{i_j-1} D_i)$ can be computed by a sparse-invertible-factor ROABP. The required P_j is $B_{i_{j-1}}$, and the required Q_j is A_{i_j}. Moreover, P_j, Q_j, and R_j are over disjoint variables for all j. ∎

Now, from the above lemma it is easy to construct a hitting-set. First we write a general result about hitting-sets for a product of polynomials from some class [45, Observation 4.1].

LEMMA 35 (Lagrange interpolation). *Suppose $\mathcal{H}$ is a hitting-set for a class of polynomials $\mathcal{C}$. Let $C(\mathbf{x}) = C_1(\mathbf{x})C_2(\mathbf{x})\cdots C_m(\mathbf{x})$ be a degree-δ' polynomial over $\mathbb{F}$, where $C_i \in \mathcal{C}$ for all $i \in [m]$. There is a hitting-set of size $\delta'|\mathcal{H}| + 1$ for $C(\mathbf{x})$.*

Proof. Let $h = |\mathcal{H}|$ and $\mathcal{H} = \{\alpha_1, \alpha_2, \ldots, \alpha_h\}$. Let $B := \{\beta_i\}_{i=1}^h$ be a set of constants. The Lagrange interpolation $\alpha(u)$ of the points in $\mathcal{H}$ is defined as follows:

$$\alpha(u) := \sum_{i=1}^{h}\frac{\prod_{j\ne i}(u-\beta_j)}{\prod_{j\ne i}(\beta_i-\beta_j)}\alpha_i.$$

The key property of the interpolation is that when we put $u = \beta_i$, then $\alpha(\beta_i) = \alpha_i$ for all $i \in [h]$. For any $a \in [m]$, we know that $C_a(\alpha_i) \ne 0$ for some $i \in [h]$. Hence, $C_a(\alpha(u))$ as a polynomial in u is nonzero because $C_a(\alpha(\beta_i)) = C_a(\alpha_i) \ne 0$. So, we can say $C(\alpha(u)) \ne 0$ as a polynomial in u.

The degree of $\alpha(u)$ is $h-1$. So, the degree of $C(\alpha(u))$ in u is bounded by $\delta'(h-1)$. We can put $\delta' h$ distinct values of u to get a hitting-set for $C(\alpha(u))$. ∎

Consider the polynomial $C(\mathbf{x}) = P(\mathbf{x})^\top Q(\mathbf{x})R(\mathbf{x})$, where $Q \in \mathbb{F}^{w\times w}[\mathbf{x}]$ is a polynomial computed by a width-w s-sparse-invertible-factor ROABP, $P, R \in \mathbb{F}^w[\mathbf{x}]$, and

P, Q, and R are over disjoint sets of variables for all $i \in [m+1]$. It is similar to the polynomial described in Theorem 3, except that P and R are now polynomials over $\mathbb{F}^w$. By adapting the proof of Theorem 3, we can show a hitting set of size $\mathrm{poly}(n\delta s)^{w^2 \log w}$ in such a model. Lemma 30 can also be applied to P and R to make them $(\log w^2 + 1)$-concentrated. Since Q is w^2-block-concentrated by Lemma 26, C will be (w^2+2)-block-concentrated. The rest of the proof goes through similarly. We thus get the following lemma.

LEMMA 36. *Let $C(\mathbf{x}) = P(\mathbf{x})^\top Q(\mathbf{x}) R(\mathbf{x})$, where $Q \in \mathbb{F}^{w \times w}[\mathbf{x}]$ is a polynomial computed by a width-w s-sparse-invertible-factor ROABP, $P, R \in \mathbb{F}^w[\mathbf{x}]$, and P, Q, and R are over disjoint sets of variables. Let the degree of each layer in Q be bounded by δ and the sparsity of each layer in Q be bounded by s. Then there is a hitting-set of size* $\mathrm{poly}((n\delta s)^{w^2 \log w})$ *for $C(\mathbf{x})$.*

Note that a hitting-set for $\alpha(\mathbf{x})C(\mathbf{x})$ is also a hitting-set for $C(\mathbf{x})$ if α is a nonzero polynomial. We get a hitting-set for each of the factors by Lemma 36. Lemma 34 tells us how to factorize a width-2 ROABP into a product of width-2 invertible ROABPs. Combining these results with Lemma 35, we get the following.

THEOREM 37. *Let $C(\mathbf{x}) = D_0^\top(\mathbf{x}_0)(\prod_{i=1}^{d} D_i(\mathbf{x}_i)) D_{d+1}(\mathbf{x}_{d+1})$ be a polynomial in $\mathbb{F}[\mathbf{x}]$ computed by a width-2 ROABP such that for all $1 \le i \le d$, D_i has degree bounded by δ and sparsity $\mathsf{s}(D_i)$ bounded by s. Then there is a hitting-set of size* $\mathrm{poly}(n\delta s)$.

6. Discussion. The first open problem is to do basis isolation for ROABP with only a *polynomially* large weight assignment. Also, our technique of finding a basis isolating weight assignment seems general. Further work is needed to determine to which other classes can it be applied. In particular, can it be used to solve depth-3 multilinear circuits? An easier question, perhaps, could be to improve Theorem 18 to get a truly blackbox PIT for the 2-base-sets-1-distance model.

Another question is whether we can find a similar result in the boolean setting, i.e., get a PRG for unknown-order ROBP with the same seed-length as in the known-order case.

In the case of constant-width ROABP, we could show constant-support concentration but only after assuming that the factor matrices are invertible. It seems that the invertibility assumption restricts the computing power of ROABP significantly. It is desirable to have low-support concentration without the assumption of invertibility.

As in the case of invertible ROABP and width-2 ROABP, analogous results hold in the boolean setting, it will be interesting to see whether there is some connection, at the level of techniques, between PRGs for boolean and arithmetic models.

Appendix. Separating a set of monomials.

LEMMA 4 (restated). *Let $\mathcal{M}$ be the set of all monomials in n variables $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ with maximum individual degree δ. For any value a, there exists a (constructible) set of $N := na\log(\delta+1)$ weight functions $\mathrm{w}\colon \mathbf{x} \to \{1, \ldots, 2N\log N\}$ such that for any set A of pairs of monomials from $\mathcal{M}$ with $|A| = a$, at least one of the weight functions separates all the pairs in A; i.e., for all $(m, m') \in A$, $\mathrm{w}(m) \neq \mathrm{w}(m')$.*

Proof. Since we want to separate the n-variate monomials with maximum individual degree δ, we use the naive Kronecker map $W\colon x_i \mapsto (\delta+1)^{i-1}$ for all $i \in [n]$. It can be easily seen that W will give distinct weights to any two monomials (with maximum individual degree δ), but the weights given by W are exponentially high.

Thus, we take the weight function W modulo p for many small primes p. Each prime p leads to a different weight function, forming our set of candidate weight functions. We need to bound the number N of primes that ensures that at least one of the weight functions separates all the monomial pairs in A. We choose the smallest

N primes, say $\mathcal{P}$ is the set. By the effective version of the prime number theorem, the highest value in the set $\mathcal{P}$ is less than $2N \log N$.

To bound the number N of primes, we want a $p \in \mathcal{P}$ such that for all $(m, m') \in A$, $W(m) - W(m') \not\equiv 0 \pmod{p}$. That is,

$$\exists p \in \mathcal{P},\ p \nmid \prod_{(m,m') \in A} \left(W(m) - W(m')\right).$$

In other words,

$$\prod_{p \in \mathcal{P}} p \nmid \prod_{(m,m') \in A} \left(W(m) - W(m')\right).$$

This can be ensured by setting $\prod_{p \in \mathcal{P}} p > \prod_{(m,m') \in A} \left(W(m) - W(m')\right)$. There are $|A|$ such monomial pairs and each $W(m) < (\delta + 1)^n$. Also, $\prod_{p \in \mathcal{P}} p > 2^N$. Hence, $N = na \log(\delta + 1)$ suffices. $\square$

Acknowledgments. We thank Chandan Saha for suggestions that improved this paper. Several useful ideas about Δ-distance circuits and base sets came up during discussions with him. We thank Michael Forbes for suggesting a possible reduction from Δ-distance circuits to ROABP (Lemma 14). We thank the anonymous reviewers for various simplifications, improvements, and useful suggestions.

REFERENCES

[1] M. Agrawal, *Proving lower bounds via pseudo-random generators*, in FSTTCS 2005: Foundations of Software Technology and Theoretical Computer Science, Lecture Notes in Comput. Sci. 3821, Springer, Berlin, 2005, pp. 92–105.

[2] M. Agrawal and S. Biswas, *Primality and identity testing via Chinese remaindering*, J. ACM, 50 (2003), pp. 429–443.

[3] M. Agrawal, R. Gurjar, A. Korwar, and N. Saxena, *Hitting-sets for low-distance multilinear depth-3*, Electronic Colloquium on Computational Complexity (ECCC), 20 (2013), 174.

[4] M. Agrawal, C. Saha, R. Saptharishi, and N. Saxena, *Jacobian hits circuits: Hitting-sets, lower bounds for depth-D occur-k formulas & depth-3 transcendence degree-k circuits*, in Proceedings of the 2012 ACM Symposium on Theory of Computing (STOC), 2012, pp. 599–613.

[5] M. Agrawal, C. Saha, and N. Saxena, *Quasi-polynomial hitting-set for set-depth-Δ formulas*, in Proceedings of the 2013 ACM Symposium on Theory of Computing (STOC), 2013, pp. 321–330.

[6] E. Allender and F. Wang, *On the power of algebraic branching programs of width two*, in Automata, Languages and Programming. Part I, Lecture Notes in Comput. Sci. 6755, Springer, Heidelberg, 2011, pp. 736–747.

[7] M. Ben-Or and R. Cleve, *Computing algebraic formulas using a constant number of registers*, SIAM J. Comput., 21 (1992), pp. 54–58.

[8] S. J. Berkowitz, *On computing the determinant in small parallel time using a small number of processors*, Inform. Process. Lett., 18 (1984), pp. 147–150.

[9] A. Bogdanov, Z. Dvir, E. Verbin, and A. Yehudayoff, *Pseudorandomness for width-2 branching programs*, Theory Comput., 9 (2013), pp. 283–292.

[10] A. De, *Pseudorandomness for permutation and regular branching programs*, in Proceedings of the 26th Annual IEEE Conference on Computational Complexity, 2011, pp. 221–231.

[11] R. M. de Oliveira, A. Shpilka, and B. L. Volk, *Subexponential size hitting sets for bounded depth multilinear formulas*, Electronic Colloquium on Computational Complexity (ECCC), 21 (2014), 157 (to appear in Proceedings of the 2015 IEEE Conference on Computational Complexity).

[12] R. A. Demillo and R. J. Lipton, *A probabilistic remark on algebraic program testing*, Inform. Process. Lett., 7 (1978), pp. 193–195.

[13] Z. Dvir and A. Shpilka, *Locally decodable codes with two queries and polynomial identity testing for depth 3 circuits*, SIAM J. Comput., 36 (2007), pp. 1404–1434.
[14] M. A. Forbes, R. Saptharishi, and A. Shpilka, *Hitting sets for multilinear read-once algebraic branching programs, in any order*, in Proceedings of the 2014 ACM Symposium on Theory of Computing (STOC), 2014, pp. 867–875.
[15] M. A. Forbes and A. Shpilka, *On identity testing of tensors, low-rank recovery and compressed sensing*, in Proceedings of the 2012 ACM Symposium on Theory of Computing (STOC), 2012, pp. 163–171.
[16] M. A. Forbes and A. Shpilka, *Quasipolynomial-time identity testing of non-commutative and read-once oblivious algebraic branching programs*, in Proceedings of the 2013 IEEE 54th Annual Symposium on Foundations of Computer Science (FOCS), 2013, pp. 243–252.
[17] A. Gupta, P. Kamath, N. Kayal, and R. Saptharishi, *Arithmetic circuits: A chasm at depth three*, in Proceedings of the 2013 IEEE 54th Annual Symposium on Foundations of Computer Science (FOCS), 2013, pp. 578–587.
[18] R. Gurjar, A. Korwar, N. Saxena, and T. Thierauf, *Deterministic identity testing for sum of read once ABPs*, Electronic Colloquium on Computational Complexity (ECCC), 21 (2014), 158 (to appear in Proceedings of the 2015 IEEE Conference on Computational Complexity).
[19] R. Impagliazzo, R. Meka, and D. Zuckerman, *Pseudorandomness from shrinkage*, in Proceedings of the 2012 IEEE 53rd Annual Symposium on Foundations of Computer Science (FOCS), 2012, pp. 111–119.
[20] M. J. Jansen, Y. Qiao, and J. Sarma, *Deterministic black-box identity testing π-ordered algebraic branching programs*, in Proceedings of the 30th International Conference on Foundations of Software Technology and Theoretical Computer Science, LIPIcs. Leibniz. Int. Proc. Inform. 8, Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, Germany, 2010, pp. 296–307.
[21] M. J. Jansen, Y. Qiao, and J. Sarma, *Deterministic identity testing of read-once algebraic branching programs*, Electronic Colloquium on Computational Complexity (ECCC), 17 (2010), 84.
[22] Z. S. Karnin and A. Shpilka, *Black box polynomial identity testing of generalized depth-3 arithmetic circuits with bounded top fan-in*, Combinatorica, 31 (2011), pp. 333–364.
[23] N. Kayal and S. Saraf, *Blackbox polynomial identity testing for depth 3 circuits*, in Proceedings of the 2009 IEEE 50th Annual Symposium on Foundations of Computer Science (FOCS), 2009, pp. 198–207.
[24] N. Kayal and N. Saxena, *Polynomial identity testing for depth 3 circuits*, Comput. Complexity, 16 (2007), pp. 115–138.
[25] A. R. Klivans and D. A. Spielman, *Randomness efficient identity testing of multivariate polynomials*, in Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC), 2001, pp. 216–223.
[26] M. Koucký, P. Nimbhorkar, and P. Pudlák, *Pseudorandom generators for group products*, extended abstract, in Proceedings of the 43rd Annual ACM Symposium on Theory of Computing (STOC), 2011, pp. 263–272.
[27] L. Kronecker, *Grundzüge einer arithmetischen Theorie der algebraischen Grössen*, G. Reimer, Berlin, 1882.
[28] M. Mahajan and V. Vinay, *Determinant: Combinatorics, algorithms, and complexity*, Chicago J. Theoret. Comput. Sci., 1997 (1997), 5.
[29] K. D. Mulmuley, *The GCT program toward the P vs. NP problem*, Commun. ACM, 55 (2012), pp. 98–107.
[30] K. D. Mulmuley, *Geometric complexity theory V: Equivalence between blackbox derandomization of polynomial identity testing and derandomization of Noether's normalization lemma*, in Proceedings of the 2012 IEEE 53rd Annual Symposium on Foundations of Computer Science (FOCS), 2012, pp. 629–638.
[31] N. Nisan, *Pseudorandom generators for space-bounded computation*, Combinatorica, 12 (1992), pp. 449–461.
[32] N. Nisan, *Lower bounds for non-commutative computation*, extended abstract, in Proceedings of the 23rd Annual ACM Symposium on Theory of Computing, 1991, pp. 410–418.
[33] R. Raz and A. Shpilka, *Deterministic polynomial identity testing in non-commutative models*, Comput. Complexity, 14 (2005), pp. 1–19.
[34] R. Raz and A. Yehudayoff, *Lower bounds and separations for constant depth multilinear circuits*, Comput. Complexity, 18 (2009), pp. 171–207.
[35] O. Reingold, T. Steinke, and S. P. Vadhan, *Pseudorandomness for regular branching programs via Fourier analysis*, in Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, 2013, Springer, Berlin, Heidelberg, pp. 655–670.

[36] C. Saha, R. Saptharishi, and N. Saxena, *The power of depth 2 circuits over algebras*, in Foundations of Software Technology and Theoretical Computer Science—FSTTCS 2009, LIPIcs. Leibniz. Int. Proc. Inform. 4, Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, Germany, 2009, pp. 371–382.

[37] C. Saha, R. Saptharishi, and N. Saxena, *A case of depth-3 identity testing, sparse factorization and duality*, Comput. Complexity, 22 (2013), pp. 39–69.

[38] N. Saxena, *Diagonal circuit identity testing and lower bounds*, in Automata, Languages and Programming. Part I, Lecture Notes in Comput. Sci. 5125, Springer, Berlin, 2008, pp. 60–71.

[39] N. Saxena, *Progress on polynomial identity testing*, Bull. Eur. Assoc. Theor. Comput. Sci. EATCS, no. 99 (2009), pp. 49–79.

[40] N. Saxena, *Progress on polynomial identity testing*-II, in Perspectives in Computational Complexity, Progr. Comput. Sci. Appl. Logic 26, M. Agrawal and V. Arvind, eds., Springer International Publishing, Cham, Switzerland, 2014, pp. 131–146.

[41] N. Saxena and C. Seshadhri, *An almost optimal rank bound for depth-3 identities*, SIAM J. Comput., 40 (2011), pp. 200–224.

[42] N. Saxena and C. Seshadhri, *Blackbox identity testing for bounded top-fanin depth-3 circuits: The field doesn't matter*, SIAM J. Comput., 41 (2012), pp. 1285–1298.

[43] N. Saxena and C. Seshadhri, *From Sylvester-Gallai configurations to rank bounds: Improved blackbox identity test for depth-3 circuits*, J. ACM, 60 (2013), 33.

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

[45] A. Shpilka and A. Yehudayoff, *Arithmetic circuits: A survey of recent results and open questions*, Found. Trends Theor. Comput. Sci., 5 (2009), pp. 207–388.

[46] T. Steinke, *Pseudorandomness for permutation branching programs without the group theory*, Electronic Colloquium on Computational Complexity (ECCC), 19 (2012), 83.

[47] T. Steinke, S. Vadhan, and A. Wan, *Pseudorandomness and Fourier growth bounds for width 3 branching programs*, in Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, Schloss Dagstuhl, Leibniz-Zentrum für Informatik GmbH, Dagstuhl Publishing, Saarbrücken, Wadern, Germany, 2014, pp. 885–899.

[48] S. Toda, *Counting Problems Computationally Equivalent to Computing the Determinant*, Technical Report CSIM 91-07, Department of Computer Science and Information Mathematics, University of Electro-Communications, Tokyo, 1991.

[49] R. Zippel, *Probabilistic algorithms for sparse polynomials*, in Proceedings of the International Symposium on Symbolic and Algebraic Computation (EUROSAM '79), Springer-Verlag, London, 1979, pp. 216–226.