Electronic Colloquium on Computational Complexity, Report No. 158 (2014)

Deterministic Identity Testing for Sum of Read Once ABPs

Rohit Gurjar[1], Arpita Korwar[1], Nitin Saxena[1] and Thomas Thierauf[2]

[1]Department of Computer Science and Engineering, IIT Kanpur
[2]Fakultät Elektronik und Informatik, Hochschule Aalen

Abstract

A read once ABP is an arithmetic branching program with each variable occurring in at most one layer. We give the first polynomial time whitebox identity test for a polynomial computed by a sum of constantly many ROABPs. We also give a corresponding blackbox algorithm with quasi-polynomial time complexity, i.e. $n^{O(\log n)}$. The motivating special case of this model is sum of constantly many set-multilinear depth-3 circuits. The prior results for that model were only slightly better than brute-force (i.e. exponential-time).

Our techniques are a new interplay of three concepts for ROABP: low evaluation dimension, basis isolating weight assignment and low-support rank concentration.

1 Introduction

Polynomial Identity Testing (PIT) is the problem of testing whether a given n-variate polynomial is identically zero or not. The input to the PIT problem may be in the form of arithmetic circuits or arithmetic branching programs (ABP). They are the arithmetic analogues of boolean circuits and boolean branching programs, respectively. The input size of the problem is the size of the circuit (or ABP). It is usually assumed that the size of the input is polynomial in the number of variables. In this paper we follow the same convention. It is well known that PIT$\in$ RP e.g. [Sch80]. The algorithm is to just evaluate the polynomial at a random point. They show that a nonzero polynomial evaluates to a nonzero value, on a random point, with a good probability. Since all problems with randomized polynomial-time solutions are conjectured to have deterministic polynomial-time algorithms, we expect that such an algorithm exists for PIT. It is also known that any sub-exponential time algorithm for PIT implies a lower bound [KI03, Agr05]. Other general connections about PIT are available in the surveys [Sax09, Sax14, SY10].

An efficient deterministic solution for PIT is known only for very restricted input models, for example, sparse polynomials [BOT88, KS01], constant fan-in depth-3 ($\Sigma\Pi\Sigma$) circuits [DS07, KS07, KS09, KS11, SS11, SS12], set-multilinear circuits [RS05, FS12a, ASS13], read once ABP [RS05, FS13, FSS14, AGKS14]. This lack of progress is not surprising, as a recent result by Gupta et. al. [GKKS13] has shown, that a polynomial time test for depth-3 circuits would imply a sub-exponential time test for general circuits. For now, even a sub-exponential solution for depth-3 circuits remains elusive. However, an efficient test for depth-3 multilinear circuits looks within reach as a lower bound against this class of circuits is already known [RY09]. A circuit is called *multilinear* if all its gates compute a multilinear polynomial, i.e. polynomials such that maximum degree of any variable is 1.

A first step towards this goal would be to find an efficient test for the sum of two set-multilinear polynomials. A depth-3 multilinear circuit is called *set-multilinear* if all the

rgurjar@iitk.ac.in, arpk@iitk.ac.in, nitin@iitk.ac.in, thomas.thierauf@htw-aalen.de

ISSN 1433-8092

product gates in it induce the same partition on the set of variables. It is easy to see that a depth-3 multilinear circuit is a sum of polynomially many set-multilinear circuits. The only non-trivial test known for sum of two set-multilinear circuits was a sub-exponential whitebox[1] algorithm by Agrawal et. al. [AGKS14]. Our results imply the first polynomial-time whitebox algorithm, and the first quasi-polynomial-time blackbox algorithm, for the sum of two set-multilinear circuits.

We actually deal with a more general problem: the sum of ROABPs. An ABP can be seen as an iterated matrix product. The entries of each matrix are 'simple' polynomials, e.g. the entries could be univariate polynomials or linear polynomials. The maximum dimension of these matrices is called the *width* of the ABP. A read-once ABP (ROABP) is an ABP with a variable occurring in at most one matrix. Without loss of generality, we can assume that there is at most one variable in one matrix. (Using [AGKS14, Observation 12], one can write a multivariate matrix as a product of univariate matrices.) It is well-known that ROABPs subsume set-multilinear circuits (see, for example [AGKS14, Lemma 14]).

There has been a long chain of work on identity testing for ROABP. In 2005, Raz and Shpilka [RS05] gave a polynomial-time whitebox test for ROABP. Then, Forbes and Shpilka [FS13] gave a $n^{O(\log n)}$ blackbox algorithm for ROABP with known variable order. This was followed by a complete blackbox test [FSS14] that took $n^{O(d \log^2 n)}$ steps, where d is the syntactic degree bound of any variable. Their paper used the idea of *rank-concentration*, introduced by Agrawal et.al. [ASS13], who gave a $n^{O(\log n)}$ hitting set for set-multilinear circuits. Later, [AGKS14] gave an improved test for ROABP that took $n^{O(\log n)}$ time. They removed the exponential dependence on the degree d. Their test is based on the idea of *basis isolating weight assignment*. Given a polynomial over an algebra, it assigns weights to the variables, and naturally extends it to monomials, such that there is a unique minimum weight basis among the coefficients of the polynomial.

In this work, we give a polynomial time test for sum of constantly many ROABPs. Since each matrix contains at most one variable, we get a natural variable order associated with the ROABP. It is easy to see that a sum of two ROABPs with same variable order, can be expressed as one single ROABP with the same variable order. Thus, the question about a sum of c ROABPs makes sense only when they have different variable order. The older ideas for a single ROABP would not work here in a straightforward manner, as it can be shown that there is a polynomial $P(\mathbf{x})$ computed by sum of two ROABPs (in fact, sum of two set-multilinear circuits) such that any ROABP computing $P(\mathbf{x})$ must be of width $2^{\Omega(\sqrt{n})}$ [NS]. Hence, at best, one can get a $2^{O(\sqrt{n})}$-time test for sum of two ROABPs using the already known techniques.

Theorem 1. *Let $A(\mathbf{x})$ be a n-variate polynomial computed by a sum of c width-w ROABPs, with degree of each variable bounded by d. Then, there is $(w^{2^c} nd)^{O(c)}$-time algorithm to test if $A = 0$.*

Our algorithm (Section 3) uses the well-known characterizing property of an ROABP, i.e. evaluation-dimension or dimension of partial derivative polynomials is low (see subsection 2.3 for the definition of the dimension of a set of polynomials, see [FS12b, Section 6] for the equivalence between these two notions of dimension). The dimension is actually equal to the width of the ROABP. This notion of dimension was introduced by Nisan [Nis91] to prove lower bounds against ROABPs. It has also been used by Klivans and Shpilka [KS06]

[1] A *whitebox* algorithm is the one which can look inside the given circuit; a *blackbox* algorithm is the one which cannot see inside the given circuit, it can only evaluate the circuit.

for *learning* ROABPs. Like [KS06], we consider partial derivatives/evaluations only with respect to those subsets, which correspond to a prefix of the variable sequence associated with the ROABP, instead of arbitrary subsets.

We view identity testing for sum of two ROABPs as testing equivalence of two ROABPs. Our idea is inspired from a similar result in the boolean world. Testing equivalence of two read once ordered boolean branching programs (ROBP) is known [SW97]. ROBP too have a similar property of small evaluation-dimension, except that the notion of *linear dependence* becomes equality in the boolean setting.

For any ROABP A, there exists a set of key partial derivative polynomials for every prefix set, whose linear dependence essentially specifies the ROABP A (Lemma 3). We call these a *characterizing set of dependencies*. These sets of polynomials are small as the partial derivative space has low dimension. The equivalence test of A and B, is essentially taking the characterizing set of dependencies for A, and verifying them for the partial derivative polynomials of B (Algorithm 1). As B is an ROABP, the verification of these dependencies for B reduces to identity testing for a single ROABP (Lemma 5).

To generalize this test to sum of c ROABPs we take A as one ROABP and B as sum of $c-1$ ROABPs. In this case, the verification of the dependencies for B becomes the question of identity testing of sum of $c-1$ ROABPs, which we solve recursively (Section 3.2).

The same idea can be applied to decide the equivalence of an ROBP with the XOR of $c-1$ ROBPs. We skip these details here as we are interested only in the arithmetic case.

We also give an identity test for a sum of ROABPs in the blackbox setting (Section 4). To be clear, we are given blackbox access to a sum of ROABPs and *not* to the individual ROABPs.

Theorem 2. *Let $A(\mathbf{x})$ be a n-variate polynomial computed by a sum of c width-w ROABPs, with degree of each variable bounded by d. Then there is a $(wnd)^{O(c \cdot 2^c \log(wnd))}$-time hitting set for $A(\mathbf{x})$.*

Here, along with the low evaluation-dimension property, we also use the *basis isolating weight assignment* given by [AGKS14]. Essentially, we show that the hitting set for a width-w^{2^c} ROABP, given by [AGKS14], would also 'work' for a sum of c width-w ROABPs (Lemma 13). This is surprising because, as mentioned above sum of c ROABPs is not captured by an ROABP with polynomially bounded width [NS].

A novel part of our proof is that for a polynomial over an $\mathbb{F}$-algebra $\mathbb{A}_k$ (k-dimensional), a shift by a basis isolating weight assignment achieves low-support concentration (Section 5, Lemma 19). To elaborate, let $\mathrm{w}\colon \mathbf{x} \to \mathbb{N}$ be a basis isolating weight assignment for a polynomial $P(\mathbf{x}) \in \mathbb{A}_k[\mathbf{x}]$ then $P(\mathbf{x} + t^{\mathrm{w}})$ has $O(\log k)$-concentration over $\mathbb{F}(t)$. ℓ-Concentration in a polynomial $P(\mathbf{x})$ means that all its coefficients are in the linear span of its $(< \ell)$-support coefficients. Our proof significantly differs from the older rank concentration proofs [ASS13, FSS14], which always assume *distinct* weights for all the monomials (or coefficients). Here, we only require that weight of a coefficient is greater than the basis coefficients, it linearly depends on. As Agrawal et. al. [AGKS14] gave a basis isolating weight assignment for ROABP, we can use it to get low-support concentration in an ROABP.

Coming back to the equivalence test of two ROABPs A and B, let the variable order of A be, wlg, $(x_1, x_2, \ldots, x_n)$. We verify the characterizing set of dependencies of A, for the corresponding partial derivatives of B, starting from the variable set $\{x_1\}$ and going over all the prefix sets of $(x_1, x_2, \ldots, x_n)$. There will be a point where the dependency verification will fail (otherwise B has an ROABP in the same order). We claim that a common ROABP R can be constructed for A and B up to this point (Lemma 7). This can be done, as an ROABP is essentially defined by the characterizing set of dependencies. We know that there is a dependency at the next level which holds for A but not for B. We use a shift of variables to push this difference of A and B in low-support monomials. This can be done as the dependency itself is computed by an ROABP and hence, it can be concentrated using an appropriate shift. Further, we show that R has a full rank coefficient space (Lemma 7) and shift this full rank to its low-support coefficients. Finally, we argue that the coefficients involved in the dependency can be obtained by a linear combination of low-support coefficients in R. All this implies that B has a low-support coefficient which is different from A, after the appropriate shift (Lemma 11). The final test is to check all the low-support coefficients (Lemma 14).

The generalization to sum of c ROABPs is analogous to the whitebox case where we recursively use the test for $c-1$ ROABPs (Lemma 13).

As a by-product, we design a shift that makes a sum of c ROABPs low-support rank-concentrated over the matrix algebra (Corollary 15).

2 Preliminaries

2.1 Notation

Let $\mathbf{x}$ denote the tuple of variables $(x_1, x_2 \ldots, x_n)$. For any $\mathbf{a} = (a_1, a_2, \ldots, a_n) \in \mathbb{Z}_+^n$, $\mathbf{x}^\mathbf{a}$ would denote the monomial $\prod_{i=1}^n x_i^{a_i}$. For a polynomial $P(\mathbf{x})$ and monomial $\mathbf{x}^\mathbf{a}$, $\text{coeff}_P(\mathbf{x}^\mathbf{a})$ denotes the coefficient of the monomial $\mathbf{x}^\mathbf{a}$ in P. *Support size* of a monomial $\mathbf{x}^\mathbf{a}$ is given by $\text{supp}(\mathbf{a}) = |\{a_i \neq 0 \mid i \in [n]\}|$.

For a matrix R, $R(i, \cdot)$ and $R(\cdot, i)$ would denote the i-th row and the i-th column of the matrix R, respectively.

For any $a, b \in \mathbb{F}^k$, $\langle a, b \rangle$ would denote the usual dot product. By abusing this notation, for any $a \in \mathbb{F}^{w^2}$ and a matrix $R \in \mathbb{F}^{w \times w}$, $\langle a, R \rangle$ would mean the same dot product, by identifying $\mathbb{F}^{w \times w}$ with $\mathbb{F}^{w^2}$.

For any polynomial $P(\mathbf{x})$ over a k-dimensional $\mathbb{F}$-algebra $\mathbb{A}_k$, the *coefficient space* of a set of monomials $\mathcal{M}$ in P is $\text{span}_\mathbb{F}\{\text{coeff}_P(\mathbf{x}^\mathbf{a}) \mid \mathbf{a} \in \mathcal{M}\}$.

By $A \otimes B$, we denote the tensor product of A and B.

2.2 Arithmetic Branching Programs (ABPs)

An ABP is a directed graph with $n+1$ layers of vertices $(V_0, V_1, \ldots, V_n)$ (Wlg, the length of the ABP is n). The layers V_0 and V_n, each have only one vertex, say v_0 and v_n respectively. The edges are only going from the vertices in the layer V_{i-1} to the vertices in the layer V_i, for any $i \in [d]$. A width-w ABP has $|V_i| \leq w$ for all $1 \leq i \leq n-1$. 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 $\mathbb{F}$.

For an edge e, let us denote its *weight* by $W(e)$. For a path p from v_0 to v_n, 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 $C(\mathbf{x}) = \sum_{p \in \text{paths}(v_0, v_n)} W(p)$ which is the sum of the weights of all the paths from v_0 to v_n. This polynomial $C(\mathbf{x})$ is said to be computed by the ABP.

The branching program can also be represented by a matrix product $\prod_{i=1}^n D_i$, where $D_1 \in \mathbb{F}[\mathbf{x}]^{1 \times w}$, $D_n \in \mathbb{F}[\mathbf{x}]^{w \times 1}$ and $D_i \in \mathbb{F}[\mathbf{x}]^{w \times w}$ for $2 \leq i \leq n-1$ such that

$$\begin{aligned}
D_1(\ell) &= W(v_0, v_{1,\ell}), \quad \text{for } 1 \leq \ell \leq w, \\
D_i(k, \ell) &= W(v_{i-1,k}, v_{i,\ell}), \quad \text{for } 1 \leq \ell, k \leq w \text{ and } 2 \leq i \leq n-1, \\
D_n(k) &= W(v_{n-1,k}, v_n), \quad \text{for } 1 \leq k \leq w.
\end{aligned}$$

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 $i \in [n]$, where π is a permutation on the set $[n]$. The order $(x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(n)})$ is said to be the *variable order* of the ROABP. We will often view D_i as a polynomial in the variable $x_{\pi(i)}$, whose coefficients come from $\mathbb{F}^{w \times w}$. The read once property of the ABP gives us the following useful observation.

Observation 1 (Easy coefficients)**.** *For a polynomial $C(\mathbf{x})$ computed by an ROABP, $D_1(x_{\pi(1)})D_2(x_{\pi(2)}) \cdots D_n(x_{\pi(n)})$, $\text{coeff}_C(\mathbf{x}^{\mathbf{a}}) = \prod_{i=1}^{n} \text{coeff}_{D_i}(x_{\pi(i)}^{a_i})$.*

2.3 Characterization of ROABP

The $\mathbb{F}$-span of a set of polynomials $\{p_i\}_i$ is defined as $\text{span}_{\mathbb{F}}\{p_i\}_i := \left\{\sum_i \alpha_i p_i \mid \alpha_i \in \mathbb{F}\right\}$. A set of polynomials is said to be $\mathbb{F}$-linearly dependent if there exists a set of field constants $\{\alpha_i\}_i, \alpha_i \in \mathbb{F}$, not all α_i are zero, such that, $\sum_i \alpha_i p_i = 0$. The *dimension* $\dim_{\mathbb{F}} \mathcal{P}$ of a set of polynomials $\mathcal{P} = \{p_i\}_i$ is the cardinality of the largest $\mathbb{F}$-linearly independent subset of $\mathcal{P}$.

Consider a partition of the variables $\mathbf{x}$ into two parts, $\mathbf{x} = \mathbf{y} \sqcup \mathbf{z}$ with $|\mathbf{y}| = k$. Any polynomial $C(\mathbf{x})$ can be viewed as a polynomial in variables $\mathbf{y}$, where the coefficients come from $\mathbb{F}[\mathbf{z}]$. I.e. $C(\mathbf{x})$ can be written as $\sum_{\mathbf{a} \in \mathbb{N}^k} C_{(\mathbf{y},\mathbf{a})} \cdot \mathbf{y}^{\mathbf{a}}$, where $C_{(\mathbf{y},\mathbf{a})} \in \mathbb{F}[\mathbf{z}]$ is the coefficient of the monomial $\mathbf{y}^{\mathbf{a}}$. The coefficient $C_{(\mathbf{y},\mathbf{a})}$ is also sometimes expressed in the literature as a partial derivative, $\frac{\partial C}{\partial \mathbf{y}^{\mathbf{a}}}$ evaluated at $\mathbf{y} = \mathbf{0}$ (and multiplied by an appropriate constant).

The well-known characterizing property of an ROABP is that the dimension of the set of all "partial derivative polynomials" for a fixed $\mathbf{y}$, is small, when $\mathbf{y}$ is chosen appropriately, in spite of the fact that the number of these polynomials is large. The next lemma describes it formally.

Lemma 2 ([Nis91])**.** *Let $C(\mathbf{x})$ be an individual degree d polynomial computed by a width-w ROABP with variable order $(x_1, x_2, \ldots, x_n)$. Consider a subset of variables $\mathbf{y}$ which is a prefix of this variable sequence, say $\mathbf{y} = \{x_1, x_2, \ldots, x_k\}$ for any $k \leq n$. Then, $\dim_{\mathbb{F}}\{C_{(\mathbf{y},\mathbf{a})} \mid \mathbf{a} \in \{0, 1, \ldots, d\}^k\} \leq w$.*

Proof. Let $C = D_1(x_1)D_2(x_2) \cdots D_n(x_n)$, where $D_1 \in \mathbb{F}^{1 \times w}[x_1]$, $D_n \in \mathbb{F}^{w \times 1}[x_n]$ and $D_i \in \mathbb{F}^{w \times w}[x_i]$, for all $2 \leq i \leq n-1$.

Let $\mathbf{z} = \mathbf{x} \setminus \mathbf{y}$. Let $P(\mathbf{y}) \in \mathbb{F}[\mathbf{y}]^{1 \times w}$ and $Q(\mathbf{z}) \in \mathbb{F}[\mathbf{z}]^{w \times 1}$ be defined as: $P(\mathbf{y}) = D_1 D_2 \cdots D_k$ and $Q(\mathbf{z}) = D_{k+1} D_{k+2} \cdots D_n$. Let $P(\mathbf{y}) = [P_1(\mathbf{y}) \; P_2(\mathbf{y}) \; \ldots \; P_w(\mathbf{y})]$ and $Q = [Q_1(\mathbf{z}) \; Q_2(\mathbf{z}) \; \ldots \; Q_w(\mathbf{z})]^\top$, where $P_i(\mathbf{y}) \in \mathbb{F}[\mathbf{y}]$ and $Q_i(\mathbf{z}) \in \mathbb{F}[\mathbf{z}]$ for all i. Clearly, $C = \sum_{i=1}^{w} P_i(\mathbf{y}) Q_i(\mathbf{z})$. For any $\mathbf{a} \in \{0, 1, \ldots, d\}^k$, we can write $C_{(\mathbf{y},\mathbf{a})} = \sum_{i=1}^{w} \text{coeff}_{P_i}(\mathbf{y}^{\mathbf{a}}) \cdot Q_i(\mathbf{z})$. As each $C_{(\mathbf{y},\mathbf{a})}$ is in the $\mathbb{F}$-span of the polynomials in $\{Q_i\}_{i=1}^{w}$, the dimension of $\{C_{(\mathbf{y},\mathbf{a})}\}_{\mathbf{a}}$ is bounded by w. □ (Lemma 2)

For a general polynomial, this dimension can be exponential in n. Now, we will show that if this dimension is small for a polynomial then there exists a small width ROABP for that polynomial. This construction differs from [FS12b, Section 6] in the sense that we only take partial derivative with respect to a prefix set. And thus, the ROABP we construct is in a specific order, not in any order.

Lemma 3 ([Nis91])**.** *Let $A(\mathbf{x})$ be a individual degree-d polynomial such that for any $1 \leq k \leq n$,*

$$\dim_{\mathbb{F}}\{A_{(\{x_1, x_2, \ldots, x_k\},\mathbf{a})} \mid \mathbf{a} \in \{0, 1, \ldots, d\}^k\} \leq w.$$

Then there exists a width-w ROABP for $A(\mathbf{x})$ in the variable order $(x_1, x_2, \dots, x_n)$.

Before proving this lemma, let us define a *characterizing set of dependencies* of a polynomial $A(\mathbf{x})$ with respect to a variable order $\{x_1, x_2, \dots, x_n\}$. This set of dependencies will essentially give us an ROABP for A in the variable order $\{x_1, x_2, \dots, x_n\}$.

Characterizing set of dependencies: Let us define $\mathbf{y}_k = \{x_1, x_2, \dots, x_k\}$ and $\mathbf{z}_k = \mathbf{x} \setminus \mathbf{y}_k$. Let $A_{(\mathbf{y}_k,*)}$ denote the set $\{A_{(\mathbf{y}_k,\mathbf{a})} \mid \mathbf{a} \in \{0,1,\dots,d\}^k\}$. Now, we define the following two sets for all $0 \leq k \leq n$: $\text{span}_k(A)$, $\text{depend}_k(A) \subseteq \{0,1,\dots,d\}^k$, with their sizes bounded by w and $w(d+1)$, respectively. They will be defined such that for any $\mathbf{b} \in \text{depend}_k(A)$, $A_{(\mathbf{y}_k,\mathbf{b})}$ will be in the span of $\{A_{(\mathbf{y}_k,\mathbf{a})} \mid \mathbf{a} \in \text{span}_k(A)\}$. Let us first define $\text{span}_0(A) := \{\epsilon\}$ and $\text{depend}_0(A) = \emptyset$. As a convention, $\mathbf{y}_0 = \emptyset$ and $A_{(\emptyset,\epsilon)}$ would mean the polynomial A itself. Now, we define $\text{span}_k(A)$ and $\text{depend}_k(A)$ recursively, for all $1 \leq k \leq n$:

- Elements in $\text{depend}_k(A)$ are just all possible extensions of elements in $\text{span}_{k-1}(A)$, i.e. $\text{depend}_k(A) := \{(\mathbf{a}, j) \mid \mathbf{a} \in \text{span}_{k-1}(A),\ 0 \leq j \leq d\}$.
- $\text{span}_k(A)$ is defined to be a subset of $\text{depend}_k(A)$, of size $\leq w$, such that the set $\{A_{(\mathbf{y}_k,\mathbf{a})} \mid \mathbf{a} \in \text{span}_k(A)\}$ spans $\{A_{(\mathbf{y}_k,\mathbf{a})} \mid \mathbf{a} \in \text{depend}_k(A)\}$.

Note that there can be more than one candidates for $\text{span}_k(A)$. We *fix* one of them. If $A_{(\mathbf{y}_k,*)}$ has dimension bounded by w then the size of $\text{span}_k(A)$ is bounded by w. And thus, the size of $\text{depend}_{k+1}(A)$ is bounded by $w(d+1)$. The dependencies of the polynomials in $\{A_{(\mathbf{y}_k,\mathbf{a})} \mid \mathbf{a} \in \text{depend}_k(A)\}$ over $\{A_{(\mathbf{y}_k,\mathbf{a})} \mid \mathbf{a} \in \text{span}_k(A)\}$ are the *characterizing set of dependencies*.

Now, we move on to construct an ROABP for A.

Proof of Lemma 3. We can assume $|\text{span}_n(A)| = 1$, as $\{A_{(\mathbf{y}_n,\mathbf{a})} \mid \mathbf{a} \in \{0,1,\dots,d\}^n\}$ is just a set of constants. Let the set $\text{span}_k(A)$ be $\{\mathbf{a}_{k,1}, \mathbf{a}_{k,2}, \dots, \mathbf{a}_{k,w_k}\}$, with $w_k \leq w$, for each $1 \leq k \leq n-1$. Here, $w_n = 1$.

We will construct matrices $D_i \in \mathbb{F}[x_i]^{w_{i-1} \times w_i}$ ($w_0 = w_n = 1$), for all $1 \leq i \leq n$ such that for each $1 \leq k \leq n$, $A(\mathbf{x}) = D_1 D_2 \cdots D_k\ [A_{(\mathbf{y}_k,\mathbf{a}_{k,1})}\ A_{(\mathbf{y}_k,\mathbf{a}_{k,2})}\ \cdots\ A_{(\mathbf{y}_k,\mathbf{a}_{k,w_k})}]^\top$. We know that

$$A(\mathbf{x}) = \sum_{j=0}^{d} A_{(\mathbf{y}_1,j)} x_1^j. \tag{1}$$

Recall that $\text{depend}_1(A) = \{0,1,\dots,d\}$. As the set $\{A_{(\mathbf{y}_1,\mathbf{a})} \mid \mathbf{a} \in \text{span}_1(A)\}$ spans $\{A_{(\mathbf{y}_1,\mathbf{a})} \mid \mathbf{a} \in \text{depend}_1(A)\}$, we know there exists constants $\{\gamma_{ji}\}_{i,j}$ such that for all $0 \leq j \leq d$,

$$A_{(\mathbf{y}_1,j)} = \sum_{i=1}^{w_1} \gamma_{ji} A_{(\mathbf{y}_1,\mathbf{a}_{1,i})}. \tag{2}$$

From Equations (1) and (2) we get, $A(\mathbf{x}) = \sum_{i=1}^{w_1} \left(\sum_{j=0}^{d} \gamma_{ji} x_1^j \right) A_{(\mathbf{y}_1,\mathbf{a}_{1,i})}$. Now, we can define $D_1 = [D_{1,1}\ D_{1,2}\ \dots\ D_{1,w_1}]$, where $D_{1,i} = \sum_{j=0}^{d} \gamma_{ji} x_1^j$ for all $i \in [w_1]$. Clearly,

$$A = D_1 [A_{(\mathbf{y}_1,\mathbf{a}_{1,1})}\ A_{(\mathbf{y}_1,\mathbf{a}_{1,2})}\ \cdots\ A_{(\mathbf{y}_1,\mathbf{a}_{1,w_1})}]^\top. \tag{3}$$

Now, for any $1 \leq k \leq n-1$ we will construct $D_{k+1} \in \mathbb{F}[x_{k+1}]^{w_k \times w_{k+1}}$ such that

$$[A_{(\mathbf{y}_k,\mathbf{a}_{k,1})}\ A_{(\mathbf{y}_k,\mathbf{a}_{k,2})} \cdots A_{(\mathbf{y}_k,\mathbf{a}_{k,w_k})}]^\top = D_{k+1} [A_{(\mathbf{y}_{k+1},\mathbf{a}_{k+1,1})} \cdots A_{(\mathbf{y}_{k+1},\mathbf{a}_{k+1,w_{k+1}})}]^\top. \tag{4}$$

We know that for each $1 \leq i \leq w_k$,

$$A_{(\mathbf{y}_k, \mathbf{a}_{k,i})} = \sum_{j=0}^{d} A_{(\mathbf{y}_{k+1}, (\mathbf{a}_{k,i}, j))} x_{k+1}^{j}. \tag{5}$$

Observe that $(\mathbf{a}_{k,i}, j)$ is just an extension of $\mathbf{a}_{k,i}$ and thus belongs to $\mathrm{depend}_{k+1}(A)$, for each $0 \leq j \leq d$. Hence, we can say that there exists a set of constants $\{\gamma_{ijh}\}_{i,j,h}$ such that for all $0 \leq j \leq d$,

$$A_{(\mathbf{y}_{k+1}, (\mathbf{a}_{k,i}, j))} = \sum_{h=1}^{w_{k+1}} \gamma_{ijh} A_{(\mathbf{y}_{k+1}, \mathbf{a}_{k+1,h})}. \tag{6}$$

From Equations (5) and (6), $A_{(\mathbf{y}_k, \mathbf{a}_{k,i})} = \sum_{h=1}^{w_{k+1}} \left(\sum_{j=0}^{d} \gamma_{ijh} x_{k+1}^{j} \right) A_{(\mathbf{y}_{k+1}, \mathbf{a}_{k+1,h})}$, for each $1 \leq i \leq w_k$.

Now, we can define $D_{k+1}(i,h) = \sum_{j=0}^{d} \gamma_{ijh} x_{k+1}^{j}$, for all $i \in [w_k]$ and $h \in [w_{k+1}]$. Clearly, D_{k+1} is the desired matrix in Equation 4.

Combining Equations (3) and (4),we get $A = D_1 D_2 \dots D_n A_{(\mathbf{y}_n, \mathbf{a}_{n,1})}$. One can absorb the constant $A_{(\mathbf{y}_n, \mathbf{a}_{n,1})}$ in the last matrix to get an ROABP for A. □ (Lemma 3)

3 Whitebox identity testing

We will now use this characterization of ROABPs to design an algorithm to check if two given ROABPs compute the same polynomial. If the two ROABPs have same variable order then one can combine them to make a single ROABP which computes their difference. And then one can apply the test for one ROABP (whitebox [RS05], blackbox [AGKS14]). So, the problem is non-trivial only when the two ROABPs are in different variable order. Wlg, A has order $(x_1, x_2, \dots, x_n)$.

3.1 Testing the equivalence of two ROABPs

Main Idea: Let us say A and B are two polynomials, computed by two explicitly given ROABPs. The idea is to find out the characterizing set of dependencies among the partial derivative polynomials of A, and verify that the same dependencies hold for the corresponding partial derivative polynomials of B. We will see that if they indeed hold then B is just a multiple of A. As, the dimension of the set of these partial derivative polynomials is bounded by the width, the characterizing set of dependencies is polynomially bounded.

We will present the test in the following order. (i) We describe a procedure to find these dependencies, i.e. how to find the sets $\{A_{(\mathbf{y}_k, \mathbf{a})} \mid \mathbf{a} \in \mathrm{span}_k(A)\}$ and $\{A_{(\mathbf{y}_k, \mathbf{a})} \mid \mathbf{a} \in \mathrm{depend}_k(A)\}$ and moreover, their dependencies. (ii) We show that it is easy to verify these dependencies for B. (iii) We show that if B satisfies all these dependencies and moreover they are equal on a particular evaluation point then A and B are the same polynomials. We give an iterative construction, starting from $\mathrm{span}_0(A) = \{\epsilon\}$.

Constructing $\mathrm{depend}_k(A)$**:** By definition, $\mathrm{depend}_k(A)$ is just all possible one-step extensions of $\mathrm{span}_{k-1}(A)$.

Constructing $\mathrm{span}_k(A)$ **from** $\mathrm{depend}_k(A)$**:** Let $A(\mathbf{x}) = D_1(x_1) D_2(x_2) \cdots D_n(x_n)$. From Observation 1, for any $\mathbf{b} = (b_1, b_2, \dots, b_k) \in \{0, 1, \dots, d\}^k$,

$$A_{(\mathbf{y}_k, \mathbf{b})} = \left(\prod_{i=1}^{k} \mathrm{coeff}_{D_i}(x_i^{b_i}) \right) D_{k+1} \cdots D_n.$$

Here, $\mathrm{coeff}_{D_1}(x_1^{b_1}) \in \mathbb{F}^{1\times w}$ and $\mathrm{coeff}_{D_i}(x_i^{b_i}) \in \mathbb{F}^{w\times w}$ for all $2 \leq i \leq k$. For any $\mathbf{b}$, let us define $R_{\mathbf{b}} \in \mathbb{F}^{1\times w}$ to be the product $\prod_{i=1}^{k} \mathrm{coeff}_{D_i}(x_i^{b_i})$. We would get $A_{(\mathbf{y}_k,\mathbf{b})} = R_{\mathbf{b}} D_{k+1} \cdots D_n$.

Now, consider the set $\{R_{\mathbf{b}} \mid \mathbf{b} \in \mathrm{depend}_k(A)\}$. Take $\mathrm{span}_k(A)$ to be a subset (of size $\leq w$) of $\mathrm{depend}_k(A)$ such that the set $\{R_{\mathbf{b}} \mid \mathbf{b} \in \mathrm{span}_k(A)\}$ spans $\{R_{\mathbf{b}} \mid \mathbf{b} \in \mathrm{depend}_k(A)\}$. Clearly, the set $\{A_{(\mathbf{y}_k,\mathbf{b})} \mid \mathbf{b} \in \mathrm{span}_k(A)\}$ will span $\{A_{(\mathbf{y}_k,\mathbf{b})} \mid \mathbf{b} \in \mathrm{depend}_k(A)\}$. Moreover, for any $\mathbf{b} \in \mathrm{depend}_k(A)$, if $R_{\mathbf{b}} = \sum_{\mathbf{a}\in \mathrm{span}_k(A)} \gamma_{\mathbf{a}} R_{\mathbf{a}}$ then $A_{(\mathbf{y}_k,\mathbf{b})} = \sum_{\mathbf{a}\in \mathrm{span}_k(A)} \gamma_{\mathbf{a}} A_{(\mathbf{y}_k,\mathbf{a})}$. Thus, it is easy to find these dependencies.

Observe that for $k = n$, $R_{\mathbf{b}}$ is a 1×1 matrix. Thus, $\mathrm{span}_n(A)$ will have only one element, let it be $\mathbf{a}_{n,1}$. We now describe the equivalence test in Algorithm 1.

Algorithm 1: Testing equivalence of an ROABP with another polynomial

```
   input: A polynomial $A$ computed by an ROABP with variable sequence
          $(x_1, x_2, \ldots, x_n)$ and a polynomial $B$.
 1 foreach $k \in [n]$ do
 2     foreach $\mathbf{b} \in \mathrm{depend}_k(A)$ do
 3         Find a set of constants $\{\gamma_{\mathbf{a}}\}_{\mathbf{a}\in \mathrm{span}_k(A)}$ such that
           $A_{(\mathbf{y}_k,\mathbf{b})} = \sum_{\mathbf{a}\in \mathrm{span}_k(A)} \gamma_{\mathbf{a}} A_{(\mathbf{y}_k,\mathbf{a})}$;
 4         if $B_{(\mathbf{y}_k,\mathbf{b})} \neq \sum_{\mathbf{a}\in \mathrm{span}_k(A)} \gamma_{\mathbf{a}} B_{(\mathbf{y}_k,\mathbf{a})}$ then
 5             Output '$A \neq B$'
 6         end
 7     end
 8 end
 9 if $B_{(\mathbf{y}_n,\mathbf{a}_{n,1})} \neq A_{(\mathbf{y}_n,\mathbf{a}_{n,1})}$ then
10     Output '$A \neq B$'.
11 end
12 Output '$A = B$'.
```

Note that the test actually works for any polynomial B for which we can verify a given dependency of its partial derivatives (Line 4 in Algorithm 1). Here, we describe how to verify these dependencies for polynomial B, when an ROABP computing B is given.

Verifying the dependencies for B: We show that it is easy to verify the dependencies for B.

Lemma 4. *If B has a width-w ROABP in some variable order then $B_{(\mathbf{y}_k,\mathbf{a})}$ also has a width-w ROABP in the same variable order (ignoring the variables in $\mathbf{y}_k$), for any variable set $\mathbf{y}_k$ and any monomial $\mathbf{y}_k^{\mathbf{a}}$.*

Proof. Let $E_1(x_{\pi(1)})E_2(x_{\pi(2)})\cdots E_n(x_{\pi(n)})$ be the ROABP computing the polynomial B, where π is a permutation on $[n]$. It is easy to see that $B_{(\mathbf{y}_k,\mathbf{a})}$ is computed by $E'_1 E'_2 \cdots E'_n$ where,

$$E'_i = \begin{cases} E_i(x_{\pi(i)}), & \text{if } x_{\pi(i)} \notin \mathbf{y}_k, \\ \mathrm{coeff}_{E_i}(x_{\pi(i)}^{a_{\pi(i)}}), & \text{otherwise.} \end{cases}$$

Hence, $B_{(\mathbf{y}_k,\mathbf{a})}$ has a width-w ROABP in the same variable order. □ (Lemma 4)

Now, we will show that the polynomial $B_{(\mathbf{y}_k,\mathbf{b})} - \sum_{\mathbf{a}\in \mathrm{span}_k(A)} \gamma_{\mathbf{a}} B_{(\mathbf{y}_k,\mathbf{a})}$ (Algorithm 1, Line 4) also has a small width ROABP.

Lemma 5. *The polynomial $B_{(\mathbf{y}_k,\mathbf{b})} - \sum_{\mathbf{a}\in\mathrm{span}_k(A)} \gamma_{\mathbf{a}} B_{(\mathbf{y}_k,\mathbf{a})}$ has an ROABP with width at most $w(w+1)$.*

Proof. The mentioned polynomial is a sum of (at most) $w+1$ polynomials (as $|\mathrm{span}_k(A)| \leq w$), each computed by an ROABP with the variable order given by π. We can combine these ROABPs to make it one ROABP: in the layered graph representation, put the graphs in parallel and identify all the start nodes and all the end nodes. The width of the new ROABP is $(w+1)$ times the width of B, and hence, it has width $w(w+1)$. □ (Lemma 5)

So, the question of verifying the dependency reduces to that of testing the zeroness of a width-$w(w+1)$ ROABP. This can be done in $\mathsf{poly}(n, w, d)$ time [RS05].

$B_{(\mathbf{y}_n,\mathbf{a}_{n,1})} = A_{(\mathbf{y}_n,\mathbf{a}_{n,1})}$**:** The last part of Algorithm 1 is to check this equality (Line 9). $B_{(\mathbf{y}_n,\mathbf{a}_{n,1})}$ and $A_{(\mathbf{y}_n,\mathbf{a}_{n,1})}$ actually belong to the field $\mathbb{F}$, hence it is easy to verify the equality.

Correctness of Algorithm 1: If $A = B$ then clearly, all the characterizing dependencies of A will hold for B. $B_{(\mathbf{y}_n,\mathbf{a})} = A_{(\mathbf{y}_n,\mathbf{a})}$ will also hold for any $\mathbf{a}$. Thus, the algorithm will output '$A = B$'.

Now, we show the other direction.

Lemma 6. *If all the characterizing dependencies of A also hold for B (Algorithm 1, Line 4) then B is just a constant multiple of A.*

Proof. By definition, $\mathrm{span}_0(A) = \mathrm{span}_0(B)$. As, the characterizing set of dependencies of A also hold for B, for each $1 \leq k \leq n$, $\{B_{(\mathbf{y}_k,\mathbf{a})} \mid \mathbf{a} \in \mathrm{span}_k(A)\}$ spans $\{B_{(\mathbf{y}_k,\mathbf{a})} \mid \mathbf{a} \in \mathrm{depend}_k(A)\}$. Recall, the construction of an ROABP from the given set of characterizing dependencies (Lemma 3). The matrices $D_1, D_2, \ldots, D_n$ are constructed just using $\mathrm{span}_k(A)$, $\mathrm{depend}_k(A)$ and the constants involved in the dependencies. Hence, using that construction one can write, $A = D_1 D_2 \ldots D_n A_{(\mathbf{y}_n,\mathbf{a}_{n,1})}$ and $B = D_1 D_2 \ldots D_n B_{(\mathbf{y}_n,\mathbf{a}_{n,1})}$.

Thus, we have the lemma. □ (Lemma 6)

Moreover, if $A_{(\mathbf{y}_n,\mathbf{a}_{n,1})} = B_{(\mathbf{y}_n,\mathbf{a}_{n,1})}$ then clearly $A = B$, implying the correctness of Algorithm 1.

3.2 Sum of Constantly Many ROABPs

In this section, we describe a deterministic poly-time identity test for a sum of constantly many ROABPs. Here again, the question is interesting only when the ROABPs have different variable orders. Because, if some of them have the same variable order then they can be combined to make a single ROABP, thus reducing the question to a smaller number of ROABPs. Suppose $A_1(\mathbf{x}), A_2(\mathbf{x}), \ldots, A_c(\mathbf{x})$ are polynomials, each computed by an ROABP of width w and individual variable degree d. The goal is to test whether $A_1 + A_2 + \cdots + A_c = 0$.

Let us rephrase the question as testing equivalence of $-A_1$ and $A_2 + A_3 + \cdots + A_c$. Recall that Algorithm 1 can test the equivalence of an ROABP A with any polynomial B, as long as we can verify a set of given dependencies for partial derivative polynomials of B. Here, we take $A = -A_1$ and $B = A_2 + A_3 + \cdots + A_c$. The only question that remains is whether we can verify a given dependency for this particular B. Recall the form of the dependency we need to verify in Algorithm 1 (Line 4). For a $\mathbf{b} \in \mathrm{depend}_k(A)$, whether $B_{(\mathbf{y}_k,\mathbf{b})} = \sum_{\mathbf{a}\in\mathrm{span}_k(A)} \gamma_{\mathbf{a}} B_{(\mathbf{y}_k,\mathbf{a})}$. Consider the polynomial $Q = B_{(\mathbf{y}_k,\mathbf{b})} - \sum_{\mathbf{a}\in\mathrm{span}_k(A)} \gamma_{\mathbf{a}} B_{(\mathbf{y}_k,\mathbf{a})}$. Substituting the value of B, we get

$$Q = \sum_{i=2}^{c} \left(A_{i\,(\mathbf{y}_k,\mathbf{b})} - \sum_{\mathbf{a}\in\mathrm{span}_k(A)} \gamma_{\mathbf{a}} A_{i\,(\mathbf{y}_k,\mathbf{a})} \right).$$

From Lemma 5, one can make a combined ROABP of width $w(w+1)$, which computes $A_{i(\mathbf{y}_k,\mathbf{b})} - \sum_{\mathbf{a}\in \mathrm{span}_k(A)} \gamma_{\mathbf{a}} A_{i(\mathbf{y}_k,\mathbf{a})}$, for each i. Thus, Q can be written as a sum of $c-1$ ROABPs each having width $w(w+1)$. To test the zeroness of Q, we recursively use the same algorithm for sum of $c-1$ ROABPs.

Time Complexity: Now, let us see how many such dependencies we need to verify. Algorithm 1 goes over all $k \in [n]$ (Line 1) and all $\mathbf{b} \in \mathrm{depend}_k(A)$ (Line 2). As $|\mathrm{depend}_k(A)| \leq w(d+1)$, total number of dependencies verified is $nw(d+1)$. Thus, we get the following recursive formula for $T(c,w)$, time complexity for testing zeroness of sum of c ROABPs, each having width w: $T(c,w) = nw(d+1) \cdot T(c-1, w(w+1)) + \mathsf{poly}(n,w,d)$. Solving this, we get $T(c,w) = w^{O(2^c)}\mathsf{poly}(n^c, d^c)$.

4 Blackbox Identity Testing

We move on to give a blackbox test for sum of constantly many ROABPs. To be precise, we will construct a set of points in $\mathbb{F}'^n$ (where $\mathbb{F}'$ is an appropriate field extension) such that any nonzero polynomial, which can be written as a sum of constantly many ROABPs, will evaluate to a nonzero value at one of the points. As in the whitebox test, we make it a question of testing the equivalence of an ROABP A with another polynomial B. Here again, we use the characterization of an ROABP given in Lemmas 2 and 3.

Here, abusing the term ROABP, we say a polynomial $R(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]^{r\times r'}$ can be computed by a width-w ROABP ($r, r' \leq w$), if there exists matrices $D_1 \in \mathbb{F}[x_1]^{r\times w}$, $D_n \in \mathbb{F}[x_n]^{w\times r'}$ and $D_i \in \mathbb{F}[x_i]^{w\times w}$ for all $2 \leq i \leq n-1$ such that $R = D_1 D_2 \cdots D_n$.

Main Idea: Let $A \in \mathbb{F}[\mathbf{x}]$ have a width-w ROABP in variable order $(x_1, x_2, \ldots, x_n)$. We try to build an ROABP for $B \in \mathbb{F}[\mathbf{x}]$ in the same variable order as A. This is done by checking whether the characterizing set of dependencies (Section 3.1) of A also hold for B. Without loss of generality, we can assume that B does not have a width-w ROABP in the variable order $(x_1, x_2, \ldots, x_n)$, otherwise the question would reduce to the identity testing of one ROABP, which is already solved [AGKS14]. Hence, by Lemma 6, not all the characterizing dependencies of A will hold for B. Let $1 \leq k \leq n$ be the first index such that a characterizing dependency amongst $\{A_{(\mathbf{y}_k,\mathbf{a})} \mid \mathbf{a} \in \mathrm{depend}_k(A)\}$ does not hold for the corresponding partial derivative polynomials of B. We show that we can make a common ROABP for A and B up to this point.

To elaborate, we can write $A = RP$ and $B = RQ$, where $R \in \mathbb{F}[\mathbf{y}_k]^{1\times w'}$ ($w' \leq w(d+1)$) is a polynomial which can be computed by a width-w' ROABP and $P, Q \in \mathbb{F}[\mathbf{z}_k]^{w'\times 1}$ consists of partial derivative polynomials of A and B respectively. The construction also implies that coefficient space of R has full rank w'. Moreover, there exists a constant vector $\Gamma \in \mathbb{F}^{1\times w'}$ such that $\Gamma P = 0$ but $\Gamma Q \neq 0$. We want to extract out this difference as a certificate for $A \neq B$.

For this we use the concept of *low support rank concentration* defined in [ASS13]. For a polynomial over an algebra, we shift each variable to concentrate the rank of its coefficients (or non-zeroness of the coefficients, in case of a polynomial over a field) to low support coefficients. We construct this kind of a shift for an ROABP using the hitting set given by [AGKS14]. We show that if B has an ROABP then so does ΓQ (of a higher width). Thus, when appropriately shifted, ΓQ will have a low support nonzero coefficient, while ΓP has all zero coefficients. But, how do we get Γ? As mentioned earlier, the coefficient space of R is full rank, which can be concentrated in low support coefficients by the same shift. Thus, Γ can be generated by a linear combination of low support coefficients of shifted R. All this together implies that after the shift, A will have a low support coefficient different from the corresponding coefficient in B. Checking all the low support coefficients will give us the final test.

First, we see the construction of a partial ROABP common for both A and B.

Lemma 7 (Partial ROABP). *There exists $1 \leq k \leq n$ such that we can construct a polynomial $R = [R_1 \ R_2 \ \cdots \ R_{w'}] \in \mathbb{F}[\mathbf{y}_k]^{1 \times w'}$ ($w' \leq w(d+1)$) computed by a width-w' ROABP with the following properties:*

1. *$A = \sum_{i=1}^{w'} R_i P_i$ and $B = \sum_{i=1}^{w'} R_i Q_i$, where $\{P_i\}_i = \{A_{(\mathbf{y}_k, \mathbf{a})} \mid \mathbf{a} \in \mathrm{depend}_k(A)\}$ and $\{Q_i\}_i = \{B_{(\mathbf{y}_k, \mathbf{a})} \mid \mathbf{a} \in \mathrm{depend}_k(A)\}$.*

2. *There exists a set of constants $\{\gamma_i\}_{i=1}^{w+1}$ such that $\sum_{i=1}^{w+1} \gamma_i P_i = 0$ and $\sum_{i=1}^{w+1} \gamma_i Q_i \neq 0$.*

3. *The coefficient space of the polynomial R has full rank w'.*

Proof. Recall the ROABP construction from Lemma 3. Here, we assume that the set $\mathrm{span}_k(A)$ was chosen with minimum possible number of elements, i.e. the polynomials in the set $\{A_{(\mathbf{y}_k, \mathbf{a})} \mid \mathbf{a} \in \mathrm{span}_k(A)\}$ are all linearly independent. This would mean that any polynomial in $\{A_{(\mathbf{y}_k, \mathbf{a})} \mid \mathbf{a} \in \mathrm{depend}_k(A)\}$ has a unique dependency over $\{A_{(\mathbf{y}_k, \mathbf{a})} \mid \mathbf{a} \in \mathrm{span}_k(A)\}$. The matrix D_k is constructed using just these dependencies.

Let $1 \leq k \leq n$ be the first index where a dependency for A is not followed by B. Formally, for any $1 \leq k' < k$ and $\mathbf{b} \in \mathrm{depend}_{k'}(A)$, $A_{(\mathbf{y}_{k'}, \mathbf{b})} = \sum_{\mathbf{a} \in \mathrm{span}_{k'}(A)} \gamma_{\mathbf{a}} A_{(\mathbf{y}_{k'}, \mathbf{a})} \implies B_{(\mathbf{y}_{k'}, \mathbf{b})} = \sum_{\mathbf{a} \in \mathrm{span}_{k'}(A)} \gamma_{\mathbf{a}} B_{(\mathbf{y}_{k'}, \mathbf{a})}$, and there exists $\mathbf{b} \in \mathrm{depend}_k(A)$ such that

$$A_{(\mathbf{y}_k, \mathbf{b})} = \sum_{\mathbf{a} \in \mathrm{span}_k(A)} \gamma_{\mathbf{a}} A_{(\mathbf{y}_k, \mathbf{a})}, \quad \text{but } B_{(\mathbf{y}_k, \mathbf{b})} \neq \sum_{\mathbf{a} \in \mathrm{span}_k(A)} \gamma_{\mathbf{a}} B_{(\mathbf{y}_k, \mathbf{a})}. \tag{7}$$

From Lemma 3, we can construct a width-w ROABP $D_1 D_2 \ldots D_{k-1}$ such that

$$A(\mathbf{x}) = D_1 D_2 \cdots D_{k-1} [A_{(\mathbf{y}_{k-1}, \mathbf{a}_{k-1,1})} \ A_{(\mathbf{y}_{k-1}, \mathbf{a}_{k-1,2})} \ \cdots \ A_{(\mathbf{y}_{k-1}, \mathbf{a}_{k-1, w_{k-1}})}]^\top \tag{8}$$

$$B(\mathbf{x}) = D_1 D_2 \cdots D_{k-1} [B_{(\mathbf{y}_{k-1}, \mathbf{a}_{k-1,1})} \ B_{(\mathbf{y}_{k-1}, \mathbf{a}_{k-1,2})} \ \cdots \ B_{(\mathbf{y}_{k-1}, \mathbf{a}_{k-1, w_{k-1}})}]^\top \tag{9}$$

Here, $\{\mathbf{a}_{k-1,i}\}_{i=1}^{w_{k-1}}$ is $\mathrm{span}_{k-1}(A)$. Recall Equation (5) from Lemma 3. $\forall\, 1 \leq i \leq w_{k-1}$,

$$A_{(\mathbf{y}_{k-1}, \mathbf{a}_{k-1,i})} = \sum_{j=0}^{d} A_{(\mathbf{y}_k, (\mathbf{a}_{k-1,i}, j))} x_k^j \quad \text{and} \quad B_{(\mathbf{y}_{k-1}, \mathbf{a}_{k-1,i})} = \sum_{j=0}^{d} B_{(\mathbf{y}_k, (\mathbf{a}_{k-1,i}, j))} x_k^j. \tag{10}$$

Let $w' = w_{k-1}(d+1)$. Define a new matrix $\mathbb{F}[x_k]^{w_{k-1} \times w'} \ni E_k := I_{w_{k-1}} \otimes \left[x_k^0 \ x_k^1 \ \cdots \ x_k^d\right]$. Let $P = [A_{(\mathbf{y}_k, (\mathbf{a}_{k-1,1}, 0))} \cdots A_{(\mathbf{y}_k, (\mathbf{a}_{k-1,1}, d))} \cdots A_{(\mathbf{y}_k, (\mathbf{a}_{k-1, w_{k-1}}, 0))} \cdots A_{(\mathbf{y}_k, (\mathbf{a}_{k-1, w_{k-1}}, d))}]^\top$ be the vector of coefficient polynomials of all the one-step extensions of $\mathrm{span}_{k-1}(A)$. From Equation (10) it is easy to see that $[A_{(\mathbf{y}_{k-1}, \mathbf{a}_{k-1,1})} \cdots A_{(\mathbf{y}_{k-1}, \mathbf{a}_{k-1, w_{k-1}})}]^\top = E_k P$. Thus, from Equation (8), we get $A(\mathbf{x}) = D_1 D_2 \cdots D_{k-1} E_k P$. By the same arguments, $B(\mathbf{x}) = D_1 D_2 \cdots D_{k-1} E_k Q$, where $Q = [B_{(\mathbf{y}_k, (\mathbf{a}_{k-1,1}, 0))} \cdots B_{(\mathbf{y}_k, (\mathbf{a}_{k-1,1}, d))} \cdots B_{(\mathbf{y}_k, (\mathbf{a}_{k-1, w_{k-1}}, 0))} \cdots B_{(\mathbf{y}_k, (\mathbf{a}_{k-1, w_{k-1}}, d))}]^\top$. Setting $R := [R_1 \ R_2 \ \cdots \ R_{w'}] = D_1 D_2 \cdots D_{k-1} E_k$, we get $A = RP$ and $B = RQ$. Clearly, R has a width-w' ROABP. Moreover, the set of polynomials in P and Q are exactly $\{A_{(\mathbf{y}_k, \mathbf{a})} \mid \mathbf{a} \in \mathrm{depend}_k(A)\}$ and $\{B_{(\mathbf{y}_k, \mathbf{a})} \mid \mathbf{a} \in \mathrm{depend}_k(A)\}$ respectively. Hence, we get Statement 1 of the Lemma.

Let $\{P_i\}_{i=1}^{w'} := \{A_{(\mathbf{y}_k, \mathbf{a})} \mid \mathbf{a} \in \mathrm{depend}_k(A)\}$ and $\{Q_i\}_{i=1}^{w'} := \{B_{(\mathbf{y}_k, \mathbf{a})} \mid \mathbf{a} \in \mathrm{depend}_k(A)\}$. From Equation (7) we get Statement 2 of the Lemma (after renumbering the indices).

Now, we want to show that the coefficient space of R has full rank. Recall that for each $1 \leq k \leq n$, $\mathrm{span}_k(A) = \{\mathbf{a}_{k,1}, \mathbf{a}_{k,1}, \ldots, \mathbf{a}_{k, w_k}\}$. Let $A_k = D_1 D_2 \cdots D_k \in \mathbb{F}[\mathbf{y}_k]^{1 \times w_k}$. We first prove the following.

Claim 8 (Basis monomials)**.** *For any $1 \leq k \leq n$ and $1 \leq h \leq w_k$, $\operatorname{coeff}_{A_k}(\mathbf{y}_k^{\mathbf{a}_{k,h}}) = e_h$, where $e_h \in \mathbb{F}^{1\times w_k}$ is such that it has 1 as the h-th entry and 0 at other places.*

Proof. We prove it by induction on k.

Base Case: For $k = 0$, it is vacuously true (assuming $D_0 = 1$).

Induction Hypothesis: The claim is true for $k - 1$.

Induction Step: Let $\mathbf{a}_{k,h} = (\mathbf{a}, j)$ where $\mathbf{a} \in \{0, 1, \ldots, d\}^{k-1}$ and $0 \leq j \leq d$. By construction of $\operatorname{span}_k(A)$, it is clear that $\mathbf{a} \in \operatorname{span}_{k-1}(A)$. Let $\mathbf{a} = \mathbf{a}_{k-1,i}$ for some $1 \leq i \leq w_{k-1}$. It is easy to see that $\operatorname{coeff}_{A_k}(\mathbf{y}_k^{\mathbf{a}_{k,h}}) = \operatorname{coeff}_{A_{k-1}}(\mathbf{y}_{k-1}^{\mathbf{a}_{k-1,i}}) \operatorname{coeff}_{D_k}(x_k^j)$. By inductive hypothesis, $\operatorname{coeff}_{A_{k-1}}(\mathbf{y}_k^{\mathbf{a}_{k-1,i}}) = e_i$. Now, we need to find $e_i \operatorname{coeff}_{D_k}(x_k^j)$, which is nothing but $\operatorname{coeff}_{D_k(i,\cdot)}(x_k^j)$.

Recall from Lemma 3, Equation (6), $D_k(i, h') = \sum_{j'=0}^{d} \gamma_{ij'h'} x_k^{j'}$ for all $1 \leq h' \leq w_k$, where $\{\gamma_{ij'h'}\}_{ij'h'}$ are such that $A_{(\mathbf{y}_k,(\mathbf{a}_{k-1,i},j'))} = \sum_{h'=1}^{w} \gamma_{ij'h'} A_{(\mathbf{y}_k,\mathbf{a}_{k,h'})}$. Let us put $j' = j$ in this equation. We know that $(\mathbf{a}_{k-1,i}, j) = \mathbf{a}_{k,h}$. If we look at the dependency of $A_{(\mathbf{y}_k,\mathbf{a}_{k,h})}$ over the set $\{A_{(\mathbf{y}_k,\mathbf{a}_{k,h'})} \mid 1 \leq h' \leq w_k\}$, it would be simply given by e_h. Hence,

$$\gamma_{ijh'} = \begin{cases} 1 & \text{if } h' = h \\ 0 & \text{otherwise.} \end{cases}$$

Now, observe that $\operatorname{coeff}_{D_k(i,\cdot)}(x_k^j) = [\gamma_{ij1} \;\; \gamma_{ij2} \;\; \cdots \;\; \gamma_{ijw_k}] = e_h$. Hence, the claim is proved. □ (Claim 8)

Recall that $R = A_{k-1}E_k$, and $\operatorname{depend}_k(A) = \{(\mathbf{a}_{k-1,i}, j) \mid 1 \leq i \leq w_{k-1},\ 0 \leq j \leq d\}$ with $|\operatorname{depend}_k(A)| = w' = w_{k-1}(d + 1)$.

Claim 9. *For any $1 \leq i \leq w_{k-1}$ and $0 \leq j \leq d$, $\operatorname{coeff}_R(\mathbf{y}_{k-1}^{\mathbf{a}_{k-1,i}} x_k^j) = e_{(i-1)(d+1)+j+1}$, where $e_h \in \mathbb{F}^{1\times w'}$ is such that it has 1 as the h-th entry and 0 at other places.*

Proof. Observe that $\operatorname{coeff}_R(\mathbf{y}_{k-1}^{\mathbf{a}_{k-1,i}} x_k^j) = \operatorname{coeff}_{A_{k-1}}(\mathbf{y}_{k-1}^{\mathbf{a}_{k-1,i}}) \operatorname{coeff}_{E_k}(x_k^j)$. From Claim 8, $\operatorname{coeff}_{A_{k-1}}(\mathbf{a}_{k-1,i}) = e_i$. Now, we just need to find the i-th row of $\operatorname{coeff}_{E_k}(x_k^j)$. From the definition of E_k, it comes out to be $e_{(i-1)(d+1)+j+1}$. □ (Claim 9)

Clearly, $\{e_{(i-1)(d+1)+j+1} \mid 1 \leq i \leq w_{k-1},\ 0 \leq j \leq d\}$ has rank w'. Thus, coefficient space of R has full rank. □ (Lemma 7)

Before going to the actual blackbox test let us describe low support concentration. Support size of a monomial $\mathbf{x}^{\mathbf{a}}$ is given by $\operatorname{supp}(\mathbf{a}) = |\{a_i \neq 0 \mid i \in [n]\}|$. $\mathcal{M}$ will denote the set of all monomials in $\mathbf{x}$ with individual degree bound d i.e. $\mathcal{M} = \{\mathbf{a} \in \mathbb{Z}^n \mid 0 \leq a_i \leq d\ \forall i \in [n]\}$ (a monomial can be represented by an n-tuple, consisting of powers of x_i's in the monomial).

A polynomial $D(\mathbf{x})$ over an $\mathbb{F}$-algebra $\mathbb{A}$ is called low-support concentrated if its low-support coefficients span all its coefficients. Formally,

Definition 10 (ℓ-support concentration [ASS13])**.** *A polynomial $D(\mathbf{x}) \in \mathbb{A}[\mathbf{x}]$ is said to have an ℓ-support concentration if $\forall\, \mathbf{a} \in \mathcal{M}$*

$$\operatorname{coeff}_D(\mathbf{x}^{\mathbf{a}}) \in \operatorname{span}_{\mathbb{F}}\{\operatorname{coeff}_D(\mathbf{x}^{\mathbf{b}}) \mid \mathbf{b} \in \mathcal{M},\ \operatorname{supp}(\mathbf{b}) < \ell\}.$$

Thus, for a non-zero polynomial over a field, low-support concentration just means that one of the low-support coefficients is nonzero. Although, a polynomial might not already have a low-support concentration, for example $D(\mathbf{x}) = x_1 x_2 \dots x_n$, Agrawal et al. [ASS13] showed that it can be achieved through an appropriate shift. In the mentioned example, if we shift every variable by 1, i.e. $D(\mathbf{x} + \mathbf{1}) = (x_1 + 1)(x_2 + 1) \dots (x_n + 1)$, then it will have 1-support concentration. In fact, it can be shown that a random shift can achieve low-support concentration in an arbitrary polynomial over an algebra [ASS13, FSS14, AGKS13]. In Section 5, we show an efficient shift which achieves concentration in polynomials computed by ROABP. Note that as shift is an invertible process, it always preserves the coefficient space of a polynomial. By shifting a polynomial $A(\mathbf{x})$ by an n-tuple $\mathbf{f} = (f_1, f_2, \dots, f_n)$, we would mean $A(\mathbf{x} + \mathbf{f}) := A(x_1 + f_1, x_2 + f_2, \dots, x_n + f_n)$.

Let $w^{(c)} := (d+1)(2w)^{2^{c-1}}$ and $\ell_{w,c} := \log((w^{(c)})^2 + 1)$. Let $\mathbf{f}_{w,c}(t) \in \mathbb{F}[t]^n$ be an n-tuple such that for any individual degree d polynomial $D(\mathbf{x})$, computed by an ROABP of width $\leq w^{(c)}$, $D(\mathbf{x} + \mathbf{f}_{w,c})$ is $\ell_{w,c}$-concentrated. Corollary 27 and Lemma 28 (Section 5) show that such an n-tuple can be constructed in time $(nw^{2^{c-1}}d)^{O(\log n)}$ and has degree $(nw^{2^{c-1}}d)^{O(\log n)}$.

We claim that the shift by $\mathbf{f}_{w,c}$ will also work for the sum of c width-w ROABPs. First, let us see the case of sum of two ROABPs.

Lemma 11. *Let A and B be two n-variate, width-w ROABPs. Then, the polynomial $(A+B)' := (A+B)(\mathbf{x} + \mathbf{f}_{w,2})$, is $2\ell_{w,2}$-concentrated.*

Proof. We start from the construction of Lemma 7 with the added information that, now, B is also computed by an ROABP of width w.

Consider the polynomial $\Gamma Q := \sum_{i=1}^{w+1} \gamma_i Q_i$ from Lemma 7. As each Q_i is a partial derivative of B, ΓQ has a width-$(w+1)w$ ROABP (from Lemma 5). It is also known that the vector polynomial R (from Lemma 7) is computed by a width $(d+1)w$ ROABP.

As $(w+1)w \leq w^{(2)}$, the vector polynomial $\Gamma Q' = \Gamma Q(\mathbf{x} + \mathbf{f}_{w,2})$ is $\ell_{w,2}$-concentrated. Similarly, as $(d+1)w < w^{(2)}$, the vector polynomial $R' = R(\mathbf{x} + \mathbf{f}_{w,2})$ is $\ell_{w,2}$-concentrated.

Since $\Gamma Q \neq 0$ (Lemma 7), and $\Gamma Q'$ is $\ell_{w,2}$-concentrated, there exists at least one monomial $\mathbf{b} \in \{0, 1, \dots, d\}^{n-k}$ with $\mathrm{supp}(\mathbf{b}) < \ell_{w,2}$ in the variables $\mathbf{z}_k$ such that $\sum_{i=1}^{w+1} \gamma_i \cdot \mathrm{coeff}_{Q_i'}(\mathbf{z}_k{}^{\mathbf{b}}) \neq 0$. And $\sum_{i=1}^{w+1} \gamma_i \cdot \mathrm{coeff}_{P_i'}(\mathbf{z}_k{}^{\mathbf{b}}) = 0$, because $\sum_{i=1}^{w+1} \gamma_i \cdot P_i = 0$. Hence,

$$\sum_{i=1}^{w+1} \gamma_i \cdot \mathrm{coeff}_{(P_i' + Q_i')}(\mathbf{z}_k{}^{\mathbf{b}}) \neq 0. \tag{11}$$

Let $S \subset \{0, 1, \dots, d\}^k$ be the set of all $< \ell_{w,2}$-support monomials in $\mathbf{y}_k$.

Claim 12. *Any arbitrary vector $[\gamma_1, \gamma_2, \dots, \gamma_{w'}] \in \mathbb{F}^{w'}$ can be written as a linear combination of the coefficients of the S monomials in R' .*

Proof. Since $\mathrm{rank}_{\mathbb{F}(t)}\{\mathrm{coeff}_R(\mathbf{y}_k{}^{\mathbf{a}}) \mid \mathbf{a} \in \{0, 1, \dots, d\}^k\} = w'$ and moreover, R' is $\ell_{w,2}$-concentrated, $\mathrm{rank}_{\mathbb{F}(t)}\{\mathrm{coeff}_{R'}(\mathbf{y}_k{}^{\mathbf{a}}) \mid \mathbf{a} \in S\} = w'$. □ (Claim 12)

By Claim 12, let $[\gamma_1\ \gamma_2\ \dots\ \gamma_{w+1}\ 0\ 0\ \dots\ 0] = \sum_{\mathbf{a} \in S} \alpha_{\mathbf{a}} \cdot \mathrm{coeff}_{R'}(\mathbf{y}_k{}^{\mathbf{a}})$. I.e. $\gamma_i =$

$\sum_{\mathbf{a}\in S} \alpha_{\mathbf{a}} \cdot \mathrm{coeff}_{R_i'}({\mathbf{y}_k}^{\mathbf{a}})$. Thus, Equation (11) becomes

$$\begin{aligned}
\sum_{i=1}^{w'} \left(\sum_{\mathbf{a}\in S} \alpha_{\mathbf{a}}\, \mathrm{coeff}_{R_i'}({\mathbf{y}_k}^{\mathbf{a}}) \right) \cdot \mathrm{coeff}_{(P_i'+Q_i')}({\mathbf{z}_k}^{\mathbf{b}}) &\neq 0 \\
\implies \sum_{i=1}^{w'} \sum_{\mathbf{a}\in S} \alpha_{\mathbf{a}}\, \mathrm{coeff}_{R_i'(P_i'+Q_i')}({\mathbf{y}_k}^{\mathbf{a}}{\mathbf{z}_k}^{\mathbf{b}}) &\neq 0 \\
\implies \sum_{\mathbf{a}\in S} \alpha_{\mathbf{a}}\, \mathrm{coeff}_{(A+B)'} \left(\mathbf{x}^{(\mathbf{a},\mathbf{b})} \right) &\neq 0
\end{aligned}$$

$\mathrm{supp}((\mathbf{a},\mathbf{b})) = \mathrm{supp}(\mathbf{a}) + \mathrm{supp}(\mathbf{b}) < \ell_{w,2} + \ell_{w,2}$. Thus, there is a $(< 2\ell_{w,2})$-support monomial that has a non-zero coefficient in the polynomial $(A+B)'$. □ (Lemma 11)

4.1 Sum of Constantly Many ROABPs

Like the white-box test, now the goal is to test whether an oracle computing $A_1 + A_2 + \cdots + A_c = 0$ is identically zero.

Lemma 13. *Let $\{A_j\}_{j=1}^{c}$ be c-many n-variate width-w ROABPs. Then, the polynomial $(A_1 + A_2 + \cdots + A_c)' := (A_1 + A_2 + \cdots + A_c)(\mathbf{x} + \mathbf{f}_{w,c})$, is $(c\ell_{w,c})$-concentrated.*

Proof. The proof runs on similar lines as that of Lemma 11. We will actually prove the following statement.

Claim: If $\mathbf{f}$ is an n-tuple such that for any individual degree-d polynomial $D(\mathbf{x})$ computed by a width-$w^{(c')}$ ROABP, $D(\mathbf{x}+\mathbf{f})$ has $\ell_{w,c'}$-concentration then any sum of c' ROABPs will have $c'\ell_{w,c'}$-concentration after a shift by $\mathbf{f}$.

The proof is by induction on the number of ROABPs, c.

Base Case: The case of $c' = 2$ is given by Lemma 11.

Induction Hypothesis: Now, let us assume the claim is true for $c' = c - 1$.

Induction Step: We prove the claim for $c' = c$. By Lemma 7, we write $A_1 = \sum_{i=1}^{w'} R_i P_i$, and $\sum_{j=2}^{c} A_j = \sum_{i=1}^{w'} R_i Q_i$, with $w' \leq (d+1)w$. Recall that $\Gamma Q := \sum_{i=1}^{w+1} \gamma_i Q_i = \sum_{i=1}^{w+1} \gamma_i B_{(y_k,\mathbf{a}_i)}$ for some $\{\mathbf{a}_i\}_i \subseteq \mathrm{depend}_k(A)$, and $B := \sum_{j=2}^{c} A_j$.

Hence, $\Gamma Q = \sum_{i=1}^{w+1} \gamma_i \sum_{j=2}^{c} A_{j\,(y_k,\mathbf{a}_i)} = \sum_{j=2}^{c} \sum_{i=1}^{w+1} \gamma_i A_{j\,(y_k,\mathbf{a}_i)}$.

Thus, by Lemma 5, ΓQ can be computed by a sum of $(c-1)$ ROABPs, each of width $(w+1)w \leq 2w^2 =: w_1$. I.e. there are $c-1$-many, $(\leq n)$-variate width-w_1 ROABPs.

As $w^{(c)} = w_1^{(c-1)}$, a shift by $\mathbf{f}$ will achieve $(\ell_{w_1,c-1})$-concentration in any width-$w_1^{(c-1)}$ ROABP. Thus, by the Induction Hypothesis, $B' := \Gamma Q(\mathbf{x}+\mathbf{f})$ has $(c-1)\ell_{w_1,c-1}$-concentration, which is same as $(c-1)\ell_{w,c}$-concentration.

The rest of the argument is similar to that in Lemma 11: it can be shown that $\sum_{j=1}^{c} A_j' = A_1' + B'$ is $(\ell_{w,c} + (c-1)\ell_{w,c}) \leq c\ell_{w,c}$-concentrated. □ (Lemma 13)

An easy identity test for a low-support concentrated polynomial was given in [ASS13].

Lemma 14 ([ASS13])**.** *If $A(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ is an n-variate, ℓ-concentrated polynomial with highest individual degree d, then there is a $(nd)^{O(\ell)}$-time hitting-set for $A(\mathbf{x})$.*

Proof of Theorem 2. By combining Lemma 13, and Lemma 14 we get a hitting set of size $(nd)^{O(c\ell_{w,c})}$. Each evaluation of the shifted polynomial $A(\mathbf{x} + \mathbf{f}_{w,c})$ is a polynomial over $\mathbb{F}(t)$. Its degree would be $(nw^{2^{c-1}}d)^{O(\log n)}$. Thus, the identity test takes $(wnd)^{O(c \cdot 2^c \log(wnd))}$ steps. □ (Theorem 2)

4.2 Concentration in matrix polynomials

As a by-product, we show that low-support concentration can be achieved even when we have a sum of matrix polynomials, each computed by an ROABP.

Corollary 15. *Let $\mathcal{D} \subseteq F^{w\times w}[\mathbf{x}]$ be the family of matrix polynomials computed by sum of c-many width w ROABPs. I.e. $\mathcal{D} \ni D(\mathbf{x}) = A_1(\mathbf{x}) + A_2(\mathbf{x}) + \cdots + A_c(\mathbf{x})$, where each $A_j \in \mathbb{F}^{w\times w}[\mathbf{x}]$ is a matrix polynomial computed by a width w ROABP. Then, $\forall D \in \mathcal{D}, D(\mathbf{x} + \mathbf{f}_{w^2,c})$ is $(c\ell_{w^2,c})$-concentrated.*

Proof. Let $\mathbb{F}[\mathbf{x}] \supseteq \mathcal{C} := \{\langle \alpha, D\rangle \mid \alpha \in \mathbb{F}^{w^2}, D \in \mathcal{D}\}$.

Hence, $\mathcal{C} = \Big\{\langle \alpha, A_1\rangle + \langle \alpha, A_2\rangle + \cdots + \langle \alpha, A_c\rangle \mid \alpha \in \mathbb{F}^{w^2}, A_j \in \mathcal{A}\Big\}$, where, $\mathcal{A}$ is the family of all matrix polynomials computed by width w ROABPs.

By Lemma 17, each $\langle \alpha, A_j\rangle$ is computed by a width w^2 ROABP. Hence, $\forall C \in \mathcal{C}$, C is computed by a sum of c ROABPs, each of width w^2.

Hence, by Lemma 13, $\forall C \in \mathcal{C}, C(\mathbf{x} + \mathbf{f}_{w^2,c})$ is $(c\ell_{w^2,c})$-concentrated.

Hence, by Lemma 16, $\forall D \in \mathcal{D}, D(\mathbf{x}+\mathbf{f}_{w^2,c})$ is $(c\ell_{w^2,c})$-concentrated. □ (Corollary 15)

The following lemma is of independent interest.

Lemma 16. *Let $\mathcal{D}$ be a family of n-variate polynomials over a k-dimensional $\mathbb{F}$-algebra $\mathbb{A}_k$. Let $\mathcal{C}$ be a family of n-variate polynomials over $\mathbb{F}$, defined by $\mathcal{C} := \big\{\langle \alpha, D\rangle \mid \alpha \in \mathbb{F}^k, D \in \mathcal{D}\big\}$. Let $\mathbf{f}(t)$ be an n-tuple. Then, $\forall C \in \mathcal{C}, C(\mathbf{x} + \mathbf{f})$ is ℓ-concentrated, iff $\forall D \in \mathcal{D}, D(\mathbf{x} + \mathbf{f})$ is ℓ-concentrated.*

Proof. ($\Leftarrow$) is a special case of Lemma 28.

($\Rightarrow$) Let $D \in \mathcal{D}$ be any polynomial such that $\text{span}_{\mathbb{F}(t)}\{\text{coeff}_{D'}(\mathbf{x}^{\mathbf{a}}) \mid \text{supp}(\mathbf{a}) < \ell\} \subsetneq \text{span}_{\mathbb{F}(t)}\{\text{coeff}_D(\mathbf{x}^{\mathbf{a}}) \mid \mathbf{a} \in \mathcal{M}\}$, where $D' = D(\mathbf{x} + \mathbf{f})$. Hence, there exists a monomial $\mathbf{b} \in \mathcal{M}$ such that $\text{coeff}_D(\mathbf{x}^{\mathbf{b}}) \notin \text{span}_{\mathbb{F}(t)}\{\text{coeff}_{D'}(\mathbf{x}^{\mathbf{a}}) \mid \text{supp}(\mathbf{a}) < \ell\}$. Hence, $\exists \alpha \in \mathbb{F}^k$: $C := \langle \alpha, D\rangle \neq 0$, but, $\forall \mathbf{a}$ with $\text{supp}(\mathbf{a}) < \ell, \langle \alpha, \text{coeff}_{D'}(\mathbf{x}^{\mathbf{a}})\rangle = 0$. We thus found a $C \in \mathcal{C}$ such that $C \neq 0$, but $C(\mathbf{x} + \mathbf{f})$ is not ℓ-concentrated. □ (Lemma 16)

We now show how to compute the dot product of any vector in $\mathbb{F}^{w\times w}$ with a matrix polynomial computed by an ROABP.

Lemma 17. *Let $D \in \mathbb{F}^{w\times w}[\mathbf{x}]$ be a matrix polynomial computed by a width w ROABP. Then, $\forall \alpha \in \mathbb{F}^{w^2}$, $\langle \alpha, D\rangle$, the dot product of α and D can be computed by an ROABP of width w^2.*

Proof. Let $C = \langle \alpha, D\rangle$. Take $R_D = (I_w \otimes D)\gamma$, with $\gamma = [e_1^\top \; e_2^\top \; \ldots \; e_w^\top]^\top$, where e_is are the elementary vectors of dimension w. Let $D = D_1(x_1)D_2(x_2)\cdots D_n(x_n)$. Thus, $R_D = (I_w \otimes D)\gamma = (I_w \otimes D_1)(I_w \otimes D_2)\cdots(I_w \otimes D_n)\gamma$ is computed by a width w^2 ROABP. Thus, $C = \langle \alpha, R_D\rangle$ has a width-w^2 ROABP. □ (Lemma 17)

5 Fast Low-Support Concentration in ROABP

Recall that a polynomial $D(\mathbf{x})$ over an $\mathbb{F}$-algebra $\mathbb{A}$ is called low-support concentrated if its low-support coefficients span all its coefficients. Here, we show an efficient shift which achieves concentration in polynomials computed by ROABP. $\mathcal{M}$ will denote the set of all monomials in $\mathbf{x}$ with individual degree bound d, i.e. $\mathcal{M} = \{\mathbf{a} \in \mathbb{N}^n \mid 0 \leq a_i \leq d, \forall i \in [n]\}$ (a monomial can be represented by an n-tuple, consisting of powers of x_i's in the monomial).

Recently, a quasi-polynomial $(\mathsf{poly}(n, w, d)^{\log n})$ size hitting set was given by [AGKS14] for ROABP. Their hitting set involves replacing x_i with $t^{\mathrm{w}(i)}$, where w is a weight function on the variables, which does a *basis isolation*. We recall the definition of a basis isolating weight assignment from [AGKS14]. A weight function $\mathrm{w}\colon [n] \to \mathbb{N}$ on the set of variables $\mathbf{x}$ can be naturally extended to the set of monomials in variables $\mathbf{x}$. Weight of a monomial $\mathbf{x}^{\mathbf{a}}$ is defined to be $\mathrm{w}(\mathbf{a}) = \sum_{i=1}^{n} \mathrm{w}(i) a_i$, for any $\mathbf{a} = (a_1, a_2, \ldots, a_n) \in \mathbb{N}^n$.

Definition 18 (Basis Isolating Weight Assignment)**.** *A weight function* $\mathrm{w}\colon [n] \to \mathbb{N}$ *is called a basis isolating weight assignment for a polynomial* $D(\mathbf{x}) \in \mathbb{A}_k[\mathbf{x}]$ *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* $\mathbf{a}, \mathbf{b} \in S$, $\mathrm{w}(\mathbf{a}) \neq \mathrm{w}(\mathbf{b})$ *and*
- $\forall$ *monomial* $\mathbf{a} \in \mathcal{M} \setminus S$, $\mathrm{coeff}_D(\mathbf{x}^{\mathbf{a}}) \in \mathrm{span}_{\mathbb{F}}\{\mathrm{coeff}_D(\mathbf{x}^{\mathbf{b}}) \mid \mathbf{b} \in S,\ \mathrm{w}(\mathbf{b}) < \mathrm{w}(\mathbf{a})\}$.

Agrawal et al. [AGKS14, Lemma 8] gave a quasi-polynomial time construction of such a weight assignment for ROABP. Now, we prove that if instead we shift the polynomial by $\{t^{\mathrm{w}(i)}\}_{i=1}^{n}$, i.e. x_i is replaced with $x_i + t^{\mathrm{w}(i)}$ then the polynomial will have a low support concentration. Note that here the dependence of the higher support coefficients on the lower support coefficients will be over the function field $\mathbb{F}(t)$. Let $D'(\mathbf{x})$ denote the shifted polynomial, i.e. $D(\mathbf{x} + t^{\mathrm{w}}) := D(x_1 + t^{\mathrm{w}(1)}, x_2 + t^{\mathrm{w}(2)}, \ldots, x_n + t^{\mathrm{w}(n)})$. It is easy to see that coefficients of D' are linear combinations of coefficients of D, and are given by the following equation:

$$\mathrm{coeff}_{D'}(\mathbf{x}^{\mathbf{a}}) = \sum_{\mathbf{b} \in \mathcal{M}} \binom{\mathbf{b}}{\mathbf{a}} t^{\mathrm{w}(\mathbf{b}-\mathbf{a})} \cdot \mathrm{coeff}_D(\mathbf{x}^{\mathbf{b}}). \tag{12}$$

Here, $\binom{\mathbf{b}}{\mathbf{a}} := \prod_{i=1}^{n} \binom{b_i}{a_i}$ for any $\mathbf{a}, \mathbf{b} \in \mathbb{N}^n$. And $\binom{b}{a} = 0$, if $b < a$ for any $a, b \in \mathbb{N}$.

Lemma 19 (Isolation to concentration)**.** *Let* $D(\mathbf{x})$ *be a polynomial over a* k*-dimensional algebra* $\mathbb{A}_k$*. Let* w *be basis isolating weight assignment for* D*. Then,* $D(\mathbf{x} + t^{\mathrm{w}})$ *has* ℓ*-concentration, where* $\ell := \lceil \log(k+1) \rceil$.

Proof. Let $D'(\mathbf{x}) = D(\mathbf{x} + t^{\mathrm{w}})$. The relation between the coefficients of D and D' can be seen as a linear transformation. Let B and B' be $\mathcal{M} \times [k]$ matrices containing the coefficients of the polynomials D and D' respectively ($\mathbf{a}$-th row contains the coefficient of $\mathbf{x}^{\mathbf{a}}$, for any $\mathbf{a} \in \mathcal{M}$). Then we can write, $B' := \mathcal{D}^{-1}\mathcal{T}\mathcal{D}B$, where $\mathcal{T}$ is a $\mathcal{M} \times \mathcal{M}$ *transfer matrix* given by $\mathcal{T}(\mathbf{a}, \mathbf{b}) = \binom{\mathbf{b}}{\mathbf{a}}$ and $\mathcal{D}$ is a diagonal matrix given by $\mathcal{D}(\mathbf{a}, \mathbf{a}) = t^{\mathrm{w}(\mathbf{a})}$. As shifting is an invertible operation, the matrices $\mathcal{T}$ and $\mathcal{D}$ are invertible and $\mathrm{rank}(B') = \mathrm{rank}(B)$.

Let $\mathcal{M}_\ell$ be the set of monomials with support $(< \ell)$, i.e. $\mathcal{M}_\ell = \{\mathbf{a} \in \mathcal{M} \mid \mathrm{supp}(\mathbf{a}) < \ell\}$. Let B'_ℓ be a $\mathcal{M}_\ell \times [k]$ matrix containing the $(< \ell)$-support coefficients of the polynomial D'. To show ℓ-concentration in D' we need to prove that $\mathrm{rank}(B'_\ell) = \mathrm{rank}(B)$.

B'_ℓ can be written as follows: $B'_\ell = \mathcal{D}_\ell^{-1}\mathcal{T}_\ell\mathcal{D}B$, where $\mathcal{T}_\ell$ is a $\mathcal{M}_\ell \times \mathcal{M}$ submatrix of the matrix $\mathcal{T}$, containing the rows indexed by the monomials in $\mathcal{M}_\ell$ and $\mathcal{D}_\ell$ is a $\mathcal{M}_\ell \times \mathcal{M}_\ell$ submatrix of $\mathcal{D}$ containing the rows and columns indexed by the monomials in $\mathcal{M}_\ell$.

As $\mathcal{D}_\ell^{-1}$ is an invertible matrix, it suffices to show that $\mathrm{rank}(\mathcal{T}_\ell\mathcal{D}B) = \mathrm{rank}(B)$. A row of matrix B indexed by a monomial $\mathbf{a} \in \mathcal{M}$, denoted by $B(\mathbf{a}, \cdot)$, is said to have weight $\mathrm{w}(\mathbf{a})$. First, let us arrange the rows in B in a monotonically increasing order according to the weight function w. The rows with the same weight can be arranged in an arbitrary order. Accordingly, the columns of $\mathcal{T}_\ell$ and the rows and columns of $\mathcal{D}$ are also permuted. The matrix $\mathcal{D}$ remains a diagonal matrix.

Now, recall that as w is a basis isolating weight assignment, there exists a set $S \subseteq \mathcal{M}$ of monomials such that for any $\mathbf{a} \in \overline{S} := \mathcal{M} \setminus S$,

$$B(\mathbf{a}, \cdot) \in \operatorname{span}\{B(\mathbf{b}, \cdot) \mid \mathbf{b} \in S,\ \mathrm{w}(\mathbf{b}) < \mathrm{w}(\mathbf{a})\}. \tag{13}$$

Let the set S be $\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{k'}\}$ ($k' \leq k$). Consider a matrix $B_0 \in \mathbb{F}^{k' \times k}$ such that its i-th row is $B(\mathbf{s}_i, \cdot)$. Then, the matrix B can be written as the product CB_0, where C is an $\mathcal{M} \times [k']$ matrix with its $\mathbf{a}$-th row being $(\gamma_1, \gamma_2, \ldots, \gamma_{k'})$, if $B(\mathbf{a}, \cdot) = \sum_{j=1}^{k'} \gamma_j B(\mathbf{s}_j, \cdot)$, for all $\mathbf{a} \in \mathcal{M}$.

Observe that the $\mathbf{s}_i$-th row of C is simply e_i, i.e. 1 in the i-th column and 0 in others. Further, from Equation 13,

Observation 20. *For any $\mathbf{a} \in \overline{S}$, $C(\mathbf{a}, j) \neq 0$ only when $\mathrm{w}(\mathbf{s}_j) < \mathrm{w}(\mathbf{a})$.*

We will actually show that the matrix $\mathcal{T}_\ell \mathcal{D} C$ is a full rank matrix. This would immediately imply that $\operatorname{rank}(\mathcal{T}_\ell \mathcal{D} C B_0) = \operatorname{rank}(B_0) = \operatorname{rank}(B)$.

Observation 21. *The first index, where the i-th column of the matrix C has a nonzero entry, is $\mathbf{s}_i$.*

Proof. As the rows of C are arranged in an increasing order according to the weight function w, from Observation 20, it is clear that this index has to be $\mathbf{s}_i$. □ (Observation 21)

Let $R \in \mathbb{F}^{|\mathcal{M}_\ell| \times k'}$ denote the matrix product $\mathcal{T}_\ell \mathcal{D} C$. Let us view its j-th column $R(\cdot, j)$, as a polynomial over vectors, i.e. as an element in $\mathbb{F}^{|\mathcal{M}_\ell|}[t]$. Let $\operatorname{lc}(R(\cdot, j)) \in \mathbb{F}^{|\mathcal{M}_\ell|}$ denote the coefficient of the lowest degree term in the polynomial $R(\cdot, j)$. Let us define a new $\mathcal{M}_\ell \times [k']$ matrix R_0 whose j-th column is given by $\operatorname{lc}(R(\cdot, j))$.

Claim 22. *If the matrix R_0 is a full rank matrix then so is R.*

Proof. If R_0 is full rank then there exists a set of k' rows, such that its restriction to these rows, say R_0', has a nonzero determinant. Let R' denote the restriction of R to the same set of rows. It is easy to see that $\operatorname{lc}(\det(R')) = \det(R_0')$. Hence, $\det(R') \neq 0$ and R is a full rank matrix. □ (Claim 22)

Now, we show that the matrix R_0 is full rank. The j-th column of R can be written as $R(\cdot, j) = \sum_{\mathbf{a} \in \mathcal{M}} \mathcal{T}_\ell(\cdot, \mathbf{a}) C(\mathbf{a}, j) t^{\mathrm{w}(\mathbf{a})}$. By Observation 21, the first nonzero entry in the column $C(\cdot, j)$ is $C(\mathbf{s}_j, j) = 1$. Moreover, by Observation 20, if $C(\mathbf{a}, j) \neq 0$, for any $\mathbf{a} \neq \mathbf{s}_j$ then $\mathrm{w}(\mathbf{a}) > \mathrm{w}(\mathbf{s}_j)$. Hence, we can see that $\operatorname{lc}(R(\cdot, j)) = \mathcal{T}_\ell(\cdot, \mathbf{s}_j)$. Thus, the j-th column of R_0 is given by $\mathcal{T}_\ell(\cdot, \mathbf{s}_j)$. By Lemma 23, the columns of matrix $\mathcal{T}_\ell$, indexed by the set S, are linearly independent. So, we get that all columns of R_0 are linearly independent. By Claim 22, $R = \mathcal{T}_\ell \mathcal{D} C$ is full rank. This proves the Lemma. □ (Lemma 19)

Now, the only remaining thing is to show that the columns of matrix $\mathcal{T}_\ell$, indexed by the set S, are linearly independent. In fact, we will show that any $2^\ell - 1$ columns of $\mathcal{T}_\ell$ are independent.

Lemma 23 (Transfer Matrix Property)**.** *Consider the transfer matrix $\mathcal{T}_\ell$ described in Lemma 19. Any $2^\ell - 1$ columns of $\mathcal{T}_\ell$ are linearly independent.*

Proof. Recall that $\mathcal{T}_\ell$ is $\mathcal{M}_\ell \times \mathcal{M}$ matrix with $\mathcal{T}_\ell(\mathbf{a}, \mathbf{b}) = \binom{\mathbf{b}}{\mathbf{a}}$. Now, consider a set of monomials $S \subseteq \mathcal{M}$ of size $k := 2^\ell - 1$. Let $\mathcal{T}_{\ell s}$ be a $\mathcal{M}_\ell \times S$ submatrix of $\mathcal{T}_\ell$ consisting of columns indexed by the monomials in the set S. The following claim will suffice to prove the lemma.

Claim 24. *For any $v \in \mathbb{F}^{|S|}$, $\mathcal{T}_{\ell s} v \neq 0$.*

The matrix $\mathcal{T}_{\ell s}$ represents a shift by $\mathbf{1}$, i.e. every variable is shifted by 1. To see this, consider a polynomial $V(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ given by $V(\mathbf{x}) = \sum_{\mathbf{a} \in S} v_{\mathbf{a}} \mathbf{x}^{\mathbf{a}}$. Now, observe that $\mathcal{T}_{\ell s} v$ actually gives the $(< \ell)$-support coefficients of the polynomial $V'(\mathbf{x}) := V(\mathbf{x} + \mathbf{1})$. Formally, for any $\mathbf{a} \in \mathcal{M}_\ell$, $\text{coeff}_{V'}(\mathbf{a}) = \mathcal{T}_{\ell s}(\mathbf{a}, \cdot) v$. So, essentially we need to show that there exists a $(< \ell)$-support coefficient in $V'(\mathbf{x})$ which is nonzero. Our next claim proves this.

Claim 25 (Concentration in sparse polynomials)**.** *Let $V(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ be an n-variate polynomial with sparsity bounded by $2^\ell - 1$, where $\ell > 0$. Then, $V'(\mathbf{x}) = V(\mathbf{x} + \mathbf{1})$ has a nonzero $(< \ell)$-support coefficient.*

Proof. We prove it by induction on the number of variables, n.

Base Case: $(n = 1)$ The only nontrivial case is $\ell = 1$. Then $V(\mathbf{x})$ is a univariate polynomial with sparsity 1. Clearly, $V(\mathbf{x} + \mathbf{1})$ has a nonzero constant part.

Induction Hypothesis: The claim is true for $n = m - 1$ and for all $\ell \in \mathbb{Z}_{>0}$.

Induction Step: $n = m$. Let $\mathbf{x}_{m-1}$ denote the set of first $m - 1$ variables. Let us write polynomial $V(\mathbf{x})$ as $\sum_{i=0}^{d} U_i x_m^i$, where $U_i \in \mathbb{F}[\mathbf{x}_{m-1}]$, for every $0 \leq i \leq d$. Let $U_i'(\mathbf{x}_{m-1})$ denote the shifted polynomial $U_i(\mathbf{x}_{m-1} + \mathbf{1})$, for every $0 \leq i \leq d$. Now there are two cases:

Case 1: There is exactly one index in $[0, d]$, let us say i, for which $U_i \neq 0$. Then U_i has sparsity $\leq 2^\ell - 1$. As U_i is an $(m-1)$-variate polynomial, by inductive hypothesis, U_i' has a nonzero $(< \ell)$-support coefficient.

Thus, $V'(\mathbf{x}) = (x_m + 1)^i U_i'$ also has a nonzero $(< \ell)$-support coefficient.

Case 2: There are at least two U_i's which are nonzero. Then there is at least one index in $[0, d]$, let us say i, such that U_i has sparsity $2^{\ell-1} - 1$. And hence, by the inductive hypothesis, U_i' has a nonzero $(< \ell - 1)$-support coefficient. Consider the highest index j such that U_j' has a nonzero $(< \ell - 1)$-support coefficient. Let the corresponding monomial be $\mathbf{x}_{m-1}^{\mathbf{a}}$. Now, as $V'(\mathbf{x}) = \sum_{i=0}^{d} (x_m + 1)^i U_i'$, we can see that

$$\text{coeff}_{V'}(\mathbf{x}_{m-1}^{\mathbf{a}} x_m^j) = \sum_{r=j}^{d} \binom{r}{j} \text{coeff}_{U_r'}(\mathbf{x}_{m-1}^{\mathbf{a}}).$$

We know that $\text{coeff}_{U_j'}(\mathbf{x}_{m-1}^{\mathbf{a}}) \neq 0$ and for any $r > j$, $\text{coeff}_{U_r'}(\mathbf{x}_{m-1}^{\mathbf{a}}) = 0$. Hence, $\text{coeff}_{V'}(\mathbf{x}_{m-1}^{\mathbf{a}} x_m^j) \neq 0$. The monomial $\mathbf{x}_{m-1}^{\mathbf{a}} x_m^j$ has support $< \ell$, which proves our claim. □ (Claim 25)

As mentioned before this proves the Lemma. □ (Lemma 23)

Now, we use Lemma 19 to achieve concentration in a polynomial computed by an ROABP. [AGKS14, Lemma 8] gives a family of n-tuples $\{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_N\}$ such that for any given ROABP of width w, at least one of them is basis isolating weight assignment and hence, provides $\log(w^2 + 1)$-concentration (the underlying matrix algebra has dimension

w^2). Their construction has $N := (nwd)^{O(\log n)}$ and $W := \max_{i\leq N,\, j\leq n}\{\deg(f_{i,j})\} = (nwd)^{O(\log n)}$. This family can be generated in time $(nwd)^{O(\log n)}$.

We now show how to combine this family of n-tuples to construct one single shift, which works for every ROABP. Let $\mathcal{F} := \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_N\}$ be a family of n-tuples. Let $\mathbf{L}(y,t) \in \mathbb{F}[y,t]^n$ be the Lagrange interpolation of $\mathcal{F}$. I.e. $L_j = \sum_{i\in[N]} f_{i,j} \prod_{i'\in[N], i'\neq i} \frac{y-\alpha_{i'}}{\alpha_i - \alpha_{i'}}$ for all $j \in [n]$, where, α_i is an arbitrary unique field element associated with i, $\forall i$ [2]. Note that $L_j \mid_{(y=\alpha_i)} = f_{i,j}$. Thus, $\mathbf{L} \mid_{(y=\alpha_i)} = \mathbf{f}_i$. Also, $\deg_y(L_j) = N-1$ and $\deg_t(L_j) = W$.

Lemma 26 (Single shift)**.** *Let $\mathfrak{D}(\mathbf{x})$ be a family of polynomials over a k-dimensional $\mathbb{F}$-algebra $\mathbb{A}_k$ and $\mathcal{F}$ be a family of n-tuples. Suppose for every polynomial $D(\mathbf{x}) \in \mathfrak{D}(\mathbf{x})$, there exists an n-tuple $\mathbf{f}_i$ in $\mathcal{F}$, such that $D'(\mathbf{x},t) := D(\mathbf{x}+\mathbf{f}_i) \in \mathbb{A}_k[\mathbf{x},t]$ is ℓ-concentrated. Then, $D''(\mathbf{x},y,t) := D(\mathbf{x}+\mathbf{L}) \in \mathbb{A}_k[\mathbf{x},y,t]$ is ℓ-concentrated, for every polynomial $D(\mathbf{x}) \in \mathfrak{D}(\mathbf{x})$.*

Proof. Take one polynomial $D(\mathbf{x}) \in \mathfrak{D}(\mathbf{x})$. Let $\mathrm{rank}_{\mathbb{F}}\{\mathrm{coeff}_D(\mathbf{x}^{\mathbf{a}}) \mid \mathbf{a} \in \mathcal{M}\} = k', (k' \leq k)$.

We need to show that there exists a set S' of $(<\ell)$-support monomials in D'', such that $\mathrm{rank}_{\mathbb{F}(y,t)}\{\mathrm{coeff}_{D''}(\mathbf{x}^{\mathbf{a}}) \mid \mathbf{a} \in S'\} = k'$.

Since $D'(\mathbf{x}) := D(\mathbf{x}+\mathbf{f}_i)$ is ℓ-concentrated, there exists a set S of $(<\ell)$-support monomials in D', such that $\mathrm{rank}_{\mathbb{F}(t)}\{\mathrm{coeff}_{D'}(\mathbf{x}^{\mathbf{a}}) \mid \mathbf{a} \in S\} = k'$. Also, $D'(\mathbf{x})$ is an evaluation of D'' at $y = \alpha_i$; $D'(\mathbf{x},t) = D''(\mathbf{x},i,t)$. Thus, for all monomials $\mathbf{a} \in \mathcal{M}$, $\mathrm{coeff}_{D'}(\mathbf{x}^{\mathbf{a}}) = \mathrm{coeff}_{D''}(\mathbf{x}^{\mathbf{a}}) \mid_{(y=\alpha_i)}$.

Let $M \in \mathbb{F}[t]^{k\times|S|}$ be the $([k]\times S)$ matrix obtained by taking the coefficients of the S monomials in D'. And let $M' \in \mathbb{F}[y,t]^{k\times|S|}$ be the $([k]\times S)$ matrix obtained by taking the D'' coefficients of the monomials in S. Then, $M = M' \mid_{(y=\alpha_i)}$.

Since, M has rank k', there are k' rows, indexed by R in M such that $M(R,\cdot)$, such that $\det(M(R,\cdot)) \neq 0$. $\det(M(R,\cdot)) = \det(M'(R,\cdot)) \mid_{(y=\alpha_i)}$. Thus, $\det(M'(R,\cdot)) \neq 0$. Hence, for the monomials in S in D'', $\mathrm{rank}_{\mathbb{F}(y,t)}\{\mathrm{coeff}_{D''}(\mathbf{x}^{\mathbf{a}}) \mid \mathbf{a} \in S\} = k'$. □ (Lemma 26)

Using this interpolation, we can construct a single shift, which works for all width-$(\leq w)$ ROABPs.

Corollary 27. *A univariate n-tuple $\mathbf{f}(t)$ of degree $(nwd)^{O(\log n)}$ can be found in time $(nwd)^{O(\log n)}$, such that when shifted by $\mathbf{f}(t)$, any n-variate, individual degree d ROABP of width $w_0 \leq w$ becomes $\log(w_0^2+1)$-concentrated.*

Proof. By Lemma 26, the Lagrange interpolation $\mathbf{L}(y,t)$, of $\{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_N\}$ obtained from [AGKS14][Lemma 8] has y- and t-degrees $= (nwd)^{O(\log n)}$. After shifting an n-variate, degree-d polynomial with $\mathbf{L}(y,t)$, its coefficients will be polynomials in y and t, with degree $dn \cdot (nwd)^{O(\log n)} =: d'$. Thus, replacing y with $t^{d'+1}$ will not affect the nonzeroness of any coefficient. We take $\mathbf{f} = \mathbf{L}(t^{d'+1}, t)$, an n-tuple of univariate polynomials in t. □ (Corollary 27)

In fact, the same shift works for any polynomial $P(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]^{r\times r'}$ computed by a width-w ROABP $(r, r' \leq w)$.

Lemma 28. *Let $\mathbf{f}$ be an n-tuple such that for any n-variate, individual degree d polynomial D over $w\times w$ matrices, $D' := D(\mathbf{x}+\mathbf{f})$ is ℓ-concentrated. Let the polynomial $P \in \mathbb{F}[\mathbf{x}]^{r\times r'}$ be computed by a width w ROABP of the form $P := MDN$, where $M \in \mathbb{F}^{r\times w}$, $D \in \mathbb{F}[\mathbf{x}]^{w\times w}$ and $N \in \mathbb{F}^{w\times r'}$.*

Then, $P' := P(\mathbf{x}+\mathbf{f})$ is ℓ-concentrated.

[2] $\{\alpha_i\}_{i\in[N]}$ are N distinct field elements. They would exist only in a field extension of $\mathbb{F}$ which has size $\geq N$. In this paper, wlg, we assume that $\mathbb{F}$ is already large enough.

Proof. Since $D' = D(\mathbf{x}+\mathbf{f})$ is ℓ-concentrated,

$\mathrm{span}_{\mathbb{F}(t)}\{\mathrm{coeff}_D(\mathbf{x}^{\mathbf{a}}) \mid \mathbf{a} \in \mathcal{M}\} = \mathrm{span}_{\mathbb{F}(t)}\{\mathrm{coeff}_{D'}(\mathbf{x}^{\mathbf{a}}) \mid \mathrm{supp}(\mathbf{a}) < \ell\}$.

Hence, $\mathfrak{T}\left(\mathrm{span}_{\mathbb{F}(t)}\{\mathrm{coeff}_D(\mathbf{x}^{\mathbf{a}}) \mid \mathbf{a} \in \mathcal{M}\}\right) = \mathfrak{T}\left(\mathrm{span}_{\mathbb{F}(t)}\{\mathrm{coeff}_{D'}(\mathbf{x}^{\mathbf{a}}) \mid \mathrm{supp}(\mathbf{a}) < \ell\}\right)$, where $\mathfrak{T}$ is the linear transformation given by pre-multiplying with M and post-multiplying with N. □ (Lemma 28)

6 Discussion

The first question is whether one can make the time complexity proportional to $w^{O(c)}$ instead of $w^{O(2^c)}$. This blow up happens because, when we want to combine (w+1)-many partial derivative polynomials of a width-w ROABP, it becomes a width-$O(w^2)$ ROABP. There are examples where this bound seems tight. So, a new property of sum of ROABPs needs to be discovered.

It also needs to be investigated if these ideas can be generalized to work for sum of more than constantly many ROABPs, or depth-3 multilinear circuits?

As mentioned in the introduction, the idea for equivalence of two ROABPs was inspired from the equivalence of two read once boolean branching programs (ROBP). It would be interesting to know if there are concrete connections between arithmetic and boolean branching programs. In particular, can ideas from identity testing of an ROABP be applied to construct pseudorandomness for ROBP.

E.g. the less investigated model, XOR of constantly many ROBPs can be checked for unsatisfiability by modifying our techniques.

7 Acknowledgements

We thank Manindra Agrawal, Chandan Saha and Vineet Nair for very useful discussions and encouragement. RG thanks TCS PhD research fellowship for support. NS thanks DST-SERB for the funding support. TT thanks DFG-DST for the generous travel support.

References

[AGKS13] Manindra Agrawal, Rohit Gurjar, Arpita Korwar, and Nitin Saxena. Hitting-sets for low-distance multilinear depth-3. *Electronic Colloquium on Computational Complexity (ECCC)*, 20:174, 2013.

[AGKS14] Manindra Agrawal, Rohit Gurjar, Arpita Korwar, and Nitin Saxena. Hitting-sets for ROABP and sum of set-multilinear circuits. *Electronic Colloquium on Computational Complexity (ECCC)*, 21:85, 2014.

[Agr05] Manindra Agrawal. Proving lower bounds via pseudo-random generators. In *FSTTCS*, volume 3821 of *Lecture Notes in Computer Science*, pages 92–105, 2005.

[ASS13] Manindra Agrawal, Chandan Saha, and Nitin Saxena. Quasi-polynomial hitting-set for set-depth- formulas. In *STOC*, pages 321–330, 2013.

[BOT88] Michael Ben-Or and Prasoon Tiwari. A deterministic algorithm for sparse multivariate polynomial interpolation. In *Proceedings of the Twentieth Annual*

ACM Symposium on Theory of Computing, STOC '88, pages 301–309, New York, NY, USA, 1988. ACM.

[DS07] Zeev Dvir and Amir Shpilka. Locally decodable codes with two queries and polynomial identity testing for depth 3 circuits. *SIAM J. Comput.*, 36(5):1404–1434, 2007.

[FS12a] Michael A. Forbes and Amir Shpilka. On identity testing of tensors, low-rank recovery and compressed sensing. In *STOC*, pages 163–172, 2012.

[FS12b] Michael A. Forbes and Amir Shpilka. Quasipolynomial-time identity testing of non-commutative and read-once oblivious algebraic branching programs. *CoRR*, abs/1209.2408, 2012.

[FS13] Michael A. Forbes and Amir Shpilka. Quasipolynomial-time identity testing of non-commutative and read-once oblivious algebraic branching programs. In *FOCS*, pages 243–252, 2013.

[FSS14] Michael A. Forbes, Ramprasad Saptharishi, and Amir Shpilka. Hitting sets for multilinear read-once algebraic branching programs, in any order. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 867–875, 2014.

[GKKS13] Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Arithmetic circuits: A chasm at depth three. *FOCS*, pages 578–587, 2013.

[KI03] Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *STOC*, pages 355–364, 2003.

[KS01] Adam Klivans and Daniel A. Spielman. Randomness efficient identity testing of multivariate polynomials. In *STOC*, pages 216–223, 2001.

[KS06] Adam Klivans and Amir Shpilka. Learning restricted models of arithmetic circuits. *Theory of computing*, 2(10):185–206, 2006.

[KS07] Neeraj Kayal and Nitin Saxena. Polynomial identity testing for depth 3 circuits. *Computational Complexity*, 16(2):115–138, 2007.

[KS09] Neeraj Kayal and Shubhangi Saraf. Blackbox polynomial identity testing for depth 3 circuits. In *FOCS*, pages 198–207, 2009.

[KS11] Zohar Shay Karnin and Amir Shpilka. Black box polynomial identity testing of generalized depth-3 arithmetic circuits with bounded top fan-in. *Combinatorica*, 31(3):333–364, 2011.

[Nis91] Noam Nisan. Lower bounds for non-commutative computation (extended abstract). In *Proceedings of the 23rd ACM Symposium on Theory of Computing, ACM Press*, pages 410–418, 1991.

[NS] Vineet Nair and Chandan Saha. Personal communication, 2014.

[RS05] Ran Raz and Amir Shpilka. Deterministic polynomial identity testing in non-commutative models. *Computational Complexity*, 14(1):1–19, 2005.

[RY09] Ran Raz and Amir Yehudayoff. Lower bounds and separations for constant depth multilinear circuits. *Computational Complexity*, 18(2):171–207, 2009.

[Sax09] Nitin Saxena. Progress on polynomial identity testing. *Bulletin of the EATCS*, 99:49–79, 2009.

[Sax14] Nitin Saxena. Progress on polynomial identity testing - 2. *CoRR*, abs/1401.0976, 2014.

[Sch80] Jacob T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, October 1980.

[SS11] Nitin Saxena and Comandur Seshadhri. An almost optimal rank bound for depth-3 identities. *SIAM J. Comput.*, 40(1):200–224, 2011.

[SS12] Nitin Saxena and Comandur Seshadhri. Blackbox identity testing for bounded top-fanin depth-3 circuits: The field doesn't matter. *SIAM J. Comput.*, 41(5):1285–1298, 2012.

[SW97] Petr Savický and Ingo Wegener. Efficient algorithms for the transformation between different types of binary decision diagrams. *Acta Informatica*, 34(4):245–256, 1997.

[SY10] Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science*, 5(3-4):207–388, 2010.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de