Learning Generalized Depth Three Arithmetic Circuits in the Non-Degenerate Case

Vishwas Bhargava
Department of Computer Science, Rutgers University, Piscataway, NJ, USA

Ankit Garg
Microsoft Research, Bangalore, India

Neeraj Kayal
Microsoft Research, Bangalore, India

Chandan Saha
Indian Institute of Science, Bangalore, India

Abstract

Consider a homogeneous degree d polynomial $f = T_1 + \cdots + T_s$, $T_i = g_i(\ell_{i,1}, \ldots, \ell_{i,m})$ where g_i's are homogeneous m-variate degree d polynomials and $\ell_{i,j}$'s are linear polynomials in n variables. We design a (randomized) learning algorithm that given black-box access to f, computes black-boxes for the T_i's. The running time of the algorithm is $\text{poly}(n, m, d, s)$ and the algorithm works under some *non-degeneracy* conditions on the linear forms and the g_i's, and some additional technical assumptions $n \geq (md)^2, s \leq n^{d/4}$. The non-degeneracy conditions on $\ell_{i,j}$'s constitute non-membership in a variety, and hence are satisfied when the coefficients of $\ell_{i,j}$'s are chosen uniformly and randomly from a large enough set. The conditions on g_i's are satisfied for random polynomials and also for natural polynomials common in the study of arithmetic complexity like determinant, permanent, elementary symmetric polynomial, iterated matrix multiplication. A particularly appealing algorithmic corollary is the following: Given black-box access to an $f = \text{Det}_r(L^{(1)}) + \ldots + \text{Det}_r(L^{(s)})$, where $L^{(k)} = (\ell^{(k)}_{i,j})_{i,j}$ with $\ell^{(k)}_{i,j}$'s being linear forms in n variables chosen randomly, there is an algorithm which in time $\text{poly}(n, r)$ outputs matrices $(M^{(k)})_k$ of linear forms s.t. there exists a permutation $\pi : [s] \to [s]$ with $\text{Det}_r(M^{(k)}) = \text{Det}_r(L^{(\pi(k))})$.

Our work follows the works [22, 7] which use lower bound methods in arithmetic complexity to design average case learning algorithms. It also vastly generalizes the result in [22] about learning depth three circuits, which is a special case where each g_i is just a monomial. At the core of our algorithm is the partial derivative method which can be used to prove lower bounds for generalized depth three circuits. To apply the general framework in [22, 7], we need to establish that the non-degeneracy conditions arising out of applying the framework with the partial derivative method are satisfied in the random case. We develop simple but general and powerful tools to establish this, which might be useful in designing average case learning algorithms for other arithmetic circuit models.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory; Computing methodologies → Algebraic algorithms

Keywords and phrases Arithemtic Circuits, Average-case Learning, Depth 3 Arithmetic Circuits, Learning Algorithms, Learning Circuits, Circuit Reconstruction

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2022.21

Category RANDOM

Related Version *Full Version*: `https://eccc.weizmann.ac.il/report/2021/155/`

Funding *Vishwas Bhargava*: Research supported in part by the Simons Collaboration on Algorithms and Geometry and NSF grant CCF-1909683.

Acknowledgements The authors would like to thank the anonymous referees for useful comments that improved the presentation of the results.

Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2022).
Editors: Amit Chakrabarti and Chaitanya Swamy; Article No. 21; pp. 21:1–21:22
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Arithmetic circuits are a natural model for computing polynomials using basic arithmetic operations like addition and multiplication. The problem of learning arithmetic circuits a.k.a. reconstruction is an important and well studied problem. It can be defined for various arithmetic circuit models. Unsurprisingly, there is enough evidence to point out that the problem is likely to be hard in the worst case for most arithmetic circuit models [14, 5, 23, 31].[1] Hence, it is imperative to explore algorithms for learning arithmetic circuits that are efficient and work in the average case. One classic example of a stark contrast between the worst case and average case complexities is the tensor decomposition problem. Let us focus on $n \times n \times n$ tensors for simplicity. In the language of arithmetic complexity, tensor decomposition corresponds to learning depth three set-multilinear circuits. We have three sets of variables $\mathbf{y} = \{y_1, \dots, y_n\}$, $\mathbf{z} = \{z_1, \dots, z_n\}$, $\mathbf{w} = \{w_1, \dots, w_n\}$. Then the problem is to decompose a set-multilinear polynomial $f(\mathbf{y}, \mathbf{z}, \mathbf{w}) = \sum_{j,k,\ell} T_{j,k,\ell} y_j z_k w_\ell$ as

$$\sum_{i=1}^{s} \ell_{i1}(\mathbf{y})\, \ell_{i2}(\mathbf{z})\, \ell_{i3}(\mathbf{w})$$

for the smallest possible s (here ℓ_{ij}'s are linear forms). This is NP-hard in the worst case [14]. However, it is possible to design efficient algorithms for tensor decomposition which work well under some mild assumptions. One such algorithm is due to Jennrich [13, 27] and states that given $f(\mathbf{y}, \mathbf{z}, \mathbf{w})$ we can find the above decomposition in polynomial time if $s \leq n$ and $(\ell_{1a}, \dots, \ell_{sa})$ are linearly independent for all $a \in [3]$. Note that the algorithm works under a bound[2] on s and also a mild assumption on the linear forms (which is satisfied when the linear forms are chosen randomly). Our algorithms will also work under such *non-degeneracy conditions*. Kayal and Saha [22] designed algorithms for learning depth three arithmetic circuits in the non-degenerate case. That is, they design an algorithm for decomposing

$$f(\mathbf{x}) = \sum_{i=1}^{s} \prod_{j=1}^{d} \ell_{ij}(\mathbf{x})$$

assuming a bound on s and certain non-degeneracy conditions on the ℓ_{ij}'s. Note that the above model is different from tensor decomposition or set-multilinear circuits since there is no partitioning of variables into disjoint sets and the linear forms can depend on all the variables. We prove a far-reaching generalization of the result of [22].

1.1 The model and our results

We study the model of generalized depth three circuits. A circuit in this class computing a degree d polynomial $f(\mathbf{x})$ is an expression of the following kind,

$$f(\mathbf{x}) = g_1(\ell_{11}, \dots, \ell_{1m}) + \dots + g_s(\ell_{s1}, \dots, \ell_{sm}),$$

where g_i's are m-variate degree d homogeneous polynomials and ℓ_{ij}'s are linear forms in the variables $\mathbf{x} = (x_1, \dots, x_n)$. Our main result is an algorithm for learning decompositions of the above kind assuming certain non-degeneracy conditions.

[1] Despite this, there has been much success in designing worst case reconstruction algorithms. This includes reconstruction algorithms for the models of sparse polynomials [25], read-once algebraic branching programs (ROABPs) [1, 24] and for models with bounded top fan-in [23, 17, 11, 32, 2, 3].

[2] This bound usually corresponds to the best known lower bounds we can prove for the corresponding model.

▶ **Theorem 1** (Learning generalized depth three circuits in the *non-degenerate* case)**.** *There is a randomized algorithm that given black-box access to an n-variate degree d polynomial $f = T_1 + \cdots + T_s$, where $T_i = g_i(\ell_{i1}, \ldots, \ell_{im})$ for a homogeneous m-variate polynomial g_i, outputs black-boxes for the individual summands T_i's. The running time of the algorithm is* $\mathrm{poly}(n, m, d, s)$*. The algorithm works under certain* non-degeneracy *conditions and also under some additional technical assumptions such as $n \geq (md)^2$, $s \leq n^{d/4}$, $|\mathbb{F}| \geq \mathrm{poly}(n^d, s)$ and $\mathrm{char}(\mathbb{F}) = 0$ or $\mathrm{char}(\mathbb{F}) > d$.*

The non-degeneracy conditions are mentioned explicitly in Section 2.1. These non-degeneracy conditions are satisfied when the coefficients of the linear forms are chosen uniformly and independently at random from a large enough set and when the g_i's are either random or one of the well-known polynomials in arithmetic complexity such as determinant, permanent, elementary symmetric polynomial etc. Let us mention one such appealing corollary which follows from Theorem 1 and the algorithms for equivalence-testing of the determinant [19, 6].

▶ **Corollary 2** (Learning sums of random projections of determinants)**.** *Suppose $n, r, \mathbb{F}, s$ be such that $n \geq r^6$, $s \leq n^{r/4}$, $|\mathbb{F}| \geq \mathrm{poly}(n^r, s)$ and $\mathrm{char}(\mathbb{F}) = 0$ or $\mathrm{char}(\mathbb{F}) > r$. There is a randomized* $\mathrm{poly}(n, r)$ *time algorithm that given black-box access to an $f = \mathrm{Det}_r(L^{(1)}) + \ldots + \mathrm{Det}_r(L^{(s)})$, where $L^{(k)} = (\ell^{(k)}_{i,j})_{i,j}$ with $\ell^{(k)}_{i,j}$'s being linear forms in n variables whose coefficients are chosen independently and uniformly at random from an arbitrary set $S \subset \mathbb{F}$ of size $|S| \geq \mathrm{poly}(n^r, s)$, it outputs matrices of linear forms $(M^{(k)})_k$ s.t. there exists a permutation $\pi : [s] \to [s]$ with $\mathrm{Det}_r(M^{(k)}) = \mathrm{Det}_r(L^{(\pi(k))})$.*

Remarks

1. Once we have the black-boxes for the T_i's as in Theorem 1, it is not hard to output black-boxes for $\widetilde{g}_i$'s and also $\widetilde{\ell}_{i1}, \ldots, \widetilde{\ell}_{im}$ s.t. $T_i = \widetilde{g}_i\left(\widetilde{\ell}_{i1}, \ldots, \widetilde{\ell}_{im}\right)$. This is done by finding a invertible linear transformation on g_i that restrict it to its "essential variables", see [18, Thm 4.1]. Note that we cannot hope to exactly recover the g_i's since there is some redundancy. One can always apply a linear transformation to the input variables of g_i's to obtain different decompositions.
2. We get a similar result (as in Corollary 2) with g_i's being the elementary symmetric polynomial, permanent, iterated matrix multiplication, monomials etc. Note that it could be a mixture of these. It might seem strange that we are able to handle permanent, but note that we are only dealing with black-boxes and hence the complexity of the permanent does not come into play. It is already known how to do equivalence-testing of the permanent efficiently [19] which is similar in spirit to the $s = 1$ case.
3. The field size and the size of the set S in Theorem 1 and Corollary 2 depends exponentially on the degree. This does not affect the runtime since one can do arithmetic in exponentially large fields in polynomial time. It is possible to get a polynomial dependence on the degree. We have not elaborated on this to preserve simplicity of analysis but we provide a sketch of an argument to reduce the field size in Appendix C.

1.2 Techniques and proof overview

We follow the meta framework of [22, 7] for designing learning algorithms for arithmetic circuits in the non-degenerate case via lower bounds. We note that while the meta framework is quite general, still a lot of technical work is needed to carry it out for a particular circuit class if one has lower bounds for that class. The same holds for this paper. We will not go into the full generality of the framework and refer the reader to the exposition in [7]. Instead, we will explain the details for our special case.

Let us first see how one would prove a lower bound (on the number of summands s) for the model of generalized depth three circuits. Consider the set of all partial differential operators of order k i.e. $\mathcal{L} = \partial_{\mathbf{x}}^{=k}$. These are linear maps from $\mathbb{F}[\mathbf{x}]_d$ to $\mathbb{F}[\mathbf{x}]_{d-k}$, where $\mathbb{F}[\mathbf{x}]_t$ denote the ring of homogeneous degree t polynomials in $\mathbb{F}[\mathbf{x}]$. Note that

$$\dim(\langle \mathcal{L} \circ T_i \rangle) \leq \binom{m+k-1}{k},$$

if T_i is of the form $g_i(\ell_{i1}, \ldots, \ell_{im})$. This is easy to verify if T_i were equal to $g_i(x_1, \ldots, x_m)$. Then one can use the fact that the dimension of the partial derivative space doesn't change upon an invertible linear transformation of the variables. Also note that

$$\langle \mathcal{L} \circ f \rangle \subseteq \langle \mathcal{L} \circ T_1 \rangle + \cdots + \langle \mathcal{L} \circ T_s \rangle \tag{1}$$

$$\dim(\langle \mathcal{L} \circ f \rangle) \leq \sum_{i=1}^{s} \dim(\langle \mathcal{L} \circ T_i \rangle) \leq s \binom{m+k-1}{k}.$$

It is not too hard to design an f for which $\dim(\langle \mathcal{L} \circ f \rangle) \approx \binom{n+k-1}{k}$ (when $k \leq \lfloor d/2 \rfloor$) and for such an f we get a lower bound $\approx (n/m)^k$. We can choose $k = \lfloor d/2 \rfloor$ to get the highest lower bound.

It is natural to wonder what is the connection to learning, if there is any at all. Consider Equation 1. One can hope that in the generic case, one would get

$$\langle \mathcal{L} \circ f \rangle = \langle \mathcal{L} \circ T_1 \rangle \oplus \cdots \oplus \langle \mathcal{L} \circ T_s \rangle. \tag{2}$$

That is the inclusion becomes an equality and the sum becomes a direct sum. Furthermore, let us assume that it holds for $\mathcal{L}' = \partial_{\mathbf{x}}^{=(k+1)}$ as well. That is,

$$\langle \mathcal{L}' \circ f \rangle = \langle \mathcal{L}' \circ T_1 \rangle \oplus \cdots \oplus \langle \mathcal{L}' \circ T_s \rangle. \tag{3}$$

So we have $U := \langle \mathcal{L} \circ f \rangle$, $V := \langle \mathcal{L}' \circ f \rangle$ and the linear maps $\partial_{\mathbf{x}}^{=1}$ from U to V. Let $U_i := \langle \mathcal{L} \circ T_i \rangle$ and $V_i := \langle \mathcal{L}' \circ T_i \rangle$. Note that the linear maps $\partial_{\mathbf{x}}^{=1}$ map U_i into V_i. So one is naturally led towards the following *vector decomposition problem.*

▶ **Problem 3** (Vector space decomposition). *Given the triple $(\mathcal{M}, U, V)$ consisting of vector spaces U and V and a set of linear maps $\mathcal{M}$ from U to V, decompose U and V as*

$$U = U_1 \oplus \cdots \oplus U_s \quad V = V_1 \oplus \cdots \oplus V_s$$

s.t. $\langle \mathcal{M} \circ U_i \rangle \subseteq V_i$ for all $i \in [s]$.

For our setting, one such decomposition is described in Equations (2) and (3). Once one has access to U_i's (black-box access to a basis), it is not hard to obtain black-boxes for the T_i's. So the only thing remains to prove is the uniqueness of vector space decomposition (in addition to (2) and (3) themselves). There are many efficient algorithms to solve the vector space decomposition problem. Please refer to Appendix A for specialized algorithms that work for our setting, and [7] for a thorough discussion on the general problem and related work. Let us now describe our approaches to prove Equations (2) and (3) and also the uniqueness of decomposition.

For proving uniqueness of decomposition, we employ the use of the notion of an adjoint algebra, following [7]. The adjoint algebra essentially captures "homomorphisms" of the triple $(\mathcal{M}, U, V)$. That is,

$$\mathrm{Adj}(\mathcal{M}, U, V) = \{(D, E) : D : U \to U,\ E : V \to V \text{ linear maps and } LD = EL\, \forall\ L \in \mathcal{M}\}$$

Suppose the triple $(\mathcal{M}, U, V)$ admits a vector space decomposition $U = U_1 \oplus \cdots \oplus U_s$, $V = V_1 \oplus \cdots \oplus V_s$. Then the projection maps (Π_{U_i}, Π_{V_i}) (which are identity on U_i, V_i respectively and map other vector spaces in the direct sum to 0) lie in the adjoint. We say that the adjoint algebra is *trivial* if it is spanned by these projectors. It is not hard to show that if the adjoint algebra is trivial, then the above vector space decomposition is unique (Lemma 34). Note that one can always combine blocks in an arbitrary way, but the decomposition is unique among all "finest" decompositions where one cannot decompose any block further. So we are left with proving the uniqueness of the decomposition in Equations (2) and (3). We prove that the adjoint algebra is trivial in this case (proof of Theorem 5) using a non-degeneracy condition on the g_i's (Item 3 in Section 2.1; also see Section 3.3).

So now let us see how to prove Equations (2) and (3). Showing the direct sum $U_1 + \cdots + U_s = U_1 \oplus \cdots \oplus U_s$ (and the same for V_i's) is done in a similar way to [22], Schwartz-Zippel lemma yields the direct sum once one can show the existence of some set of linear forms satisfying the direct sum property. This is done using a design construction based on Nisan-Wigderson designs. This construction is inspired from [22] but more general. We differ significantly from previous works [22, 7] in our technique for showing that $U = U_1 + \cdots + U_s$. The previous works relied on intricate design constructions to exhibit linear forms which satisfy this property (followed by a use of Schwartz-Zippel lemma). For our setting, one can get away with the above design based approach, but this can become more cumbersome and challenging as the circuit models become more complicated. Hence, we devise a general way of proving statements of the form

$$\langle \mathcal{L} \circ f \rangle = \langle \mathcal{L} \circ T_1 \rangle + \cdots + \langle \mathcal{L} \circ T_s \rangle$$

for $f = T_1 + \cdots + T_s$, which is conceptually more appealing. It is useful to have the linear maps $\mathcal{L}$ from a subspace of the operators (so for our case think of $\mathcal{L} = \langle \partial_{\mathbf{x}}^{=k} \rangle$). Since

$$\langle \mathcal{L} \circ f \rangle \subseteq \langle \mathcal{L} \circ T_1 \rangle + \cdots + \langle \mathcal{L} \circ T_s \rangle,$$

it suffices to prove that $\langle \mathcal{L} \circ T_i \rangle \subseteq \langle \mathcal{L} \circ f \rangle$ for all i. Let us consider the operators annihilating a particular term T_i.

$$\mathcal{L}_i^{\mathrm{null}} := \{ L \in \mathcal{L} : L \circ T_i = 0 \}$$

Now note that for any $L \in \cap_{j \neq i} \mathcal{L}_j^{\mathrm{null}}$, $L \circ f = L \circ T_i$. If the subspace of operators $\cap_{j \neq i} \mathcal{L}_j^{\mathrm{null}}$ was rich enough, at least to the extent relevant to T_i, then we would be done. We are able to show this by moving to the duals of the vector spaces $\mathcal{L}_i^{\mathrm{null}}$ (with respect to an appropriate bilinear form) and proving a direct sum property there (the proof of which turns out to be almost identical to the proof we have for the direct sum of the U_i's!). For more details, see Section 2.

Comparison with previous works. Our work closely follows the papers [22, 7] on learning arithmetic circuits in the non-degenerate case via lower bounds. However, there are substantial differences as well. Firstly, as explained above, we devise a general technique for proving statements of the kind $\langle \mathcal{L} \circ f \rangle = \langle \mathcal{L} \circ T_1 \rangle + \cdots + \langle \mathcal{L} \circ T_s \rangle$. Secondly, ours is the first paper that uses the full machinery of the learning from lower bounds framework in [22, 7]. In [22], the framework was present in a rudimentary form and that made the analysis more cumbersome. While the framework was fully laid out in [7], for their application of learning sums of powers of low degree polynomials, they eventually implement a somewhat ad hoc approach. Without this learning framework, it seems rather challenging to get such a general result as in Theorem 1.

1.3 Related work

[9] proved a lower bound for the more general model of generalized depth-four circuits (bounded bottom-fanin). [17] study the worst case learning algorithms for a model which is similar to our model in many ways, but their parameters are different (they also call their model generalized depth three circuits). There has been a lot of work on *worst case* reconstruction algorithms which includes reconstruction algorithms for the models of sparse polynomials [25], read-once algebraic branching programs (ROABPs) [1, 24] and for models with bounded top fan-in [23, 17, 11, 32, 2, 3].

In [10], a randomized polynomial-time proper learning algorithm was given for *non-degenerate*[3] multilinear formulas having fan-in two. A randomized polynomial-time proper learning algorithm for non-degenerate regular formulas having fan-in two was given in [12]. An efficient randomized reconstruction for non-degenerate homogeneous ABPs of width at most $\frac{\sqrt{n}}{2}$ is presented in [20]. [22] designed algorithms for learning non-degenerate depth three circuits which is a special case of our model with the g_i's being a monomial. [7], following [22], developed a meta framework for learning non-degenerate arithmetic circuits via lower bounds. They implemented it to learn sums of powers of low degree polynomials in the non-degenerate case.

As already mentioned, the problem of tensor decomposition is a special case for our model. Tensor decomposition is widely studied in the machine learning community as well (also known as CP decomposition), e.g. see the surveys [26, 4, 15]. Another kind of tensor decomposition, Tucker decomposition is also widely studied, see Section 4 in [26]. Tensor decomposition roughly corresponds to the $m = 1$ case in our model[4] Tucker decomposition roughly corresponds to $s = 1$ in our model.[5] Given the wide variety of applications of these two problems in machine learning, we hope that (noise-resilient versions of) our algorithms will handle much more challenging problems in machine learning.

1.4 Roadmap of the paper

In Section 2, we present our algorithm for learning non-degenerate generalized depth three circuits, the corresponding non-degeneracy conditions and the analysis of the algorithm assuming the non-degeneracy conditions. In Section 3, we prove that the non-degeneracy conditions are satisfied for random circuits. Section 4 contains the summary of the work and some of the open problems that arise from this work. Section A contains some basic facts about the vector space decomposition problem. Finally, Section B contains some facts about how to perform linear algebra given black boxes.

2 The learning algorithm and its analysis

In this section, we describe our algorithm for learning non-degenerate generalized depth three circuits and the analysis assuming the non-degeneracy conditions. Since we are aiming for poly(s) time-complexity, we can assume that we know s. For a field $\mathbb{F}$ and $d \in \mathbb{N}$, let $\mathbb{F}[\mathbf{x}]_d$ denote the ring of homogeneous degree d polynomials in $\mathbb{F}[\mathbf{x}]$. Consider a homogeneous

[3] The papers [10, 12] state the results for random formulas, but it is not difficult to state the non-degeneracy conditions by taking a closer look at the algorithms.

[4] Strictly speaking $m = 1$ would be symmetric tensor decomposition and exactly modeling general tensor decomposition would require higher m but in spirit tensor decomposition is closer to the $m = 1$ case than higher m.

[5] Again, ignoring some symmetry considerations here.

degree d polynomial $f \in \mathbb{F}[\mathbf{x}]_d$ which is computed by a homogeneous generalized depth three circuit i.e., $f = T_1 + \cdots + T_s$, where $T_i = g_i(\ell_{i1}, \ldots, \ell_{im})$ for $i \in [s]$. Here ℓ_{ij}'s are linear forms.

2.1 Non-degeneracy conditions

We impose the following non-degeneracy conditions on f (or more precisely the circuit computing it):

1. For each $i \in [s]$, the linear forms $(\ell_{i1}, \ldots, \ell_{im})$ are linearly independent. Also the vector spaces $W_1^{(d-k)} := \mathbb{F}[\ell_{11}, \ldots, \ell_{1m}]_{d-k}, \ldots, W_s^{(d-k)} := \mathbb{F}[\ell_{s1}, \ldots, \ell_{sm}]_{d-k}$ form a direct sum i.e.

$$W_1^{(d-k)} + \cdots + W_s^{(d-k)} = W_1^{(d-k)} \oplus \cdots \oplus W_s^{(d-k)}.$$

   The same assumption for the vector spaces $W_i^{(d-k-1)}$'s.
2. We will use $\partial^{=k}$ to denote the set of order-k partial differential operators in the variables $\mathbf{x}$. Consider the vector spaces $U := \langle \partial^{=k} f \rangle$, $V := \langle \partial^{=(k+1)} f \rangle$, $U_i := \langle \partial^{=k} T_i \rangle$, $V_i = \langle \partial^{=(k+1)} T_i \rangle$. We will assume that

$$U = U_1 \oplus \cdots \oplus U_s$$

   and

$$V = V_1 \oplus \cdots \oplus V_s.$$

3. For the polynomials $g_i \in \mathbb{F}[\mathbf{z}]_d$, $\mathbf{z} = (z_1, \ldots, z_m)$, the triple $\left(\partial_{\mathbf{z}}^{=1}, \langle \partial_{\mathbf{z}}^{=k} g_i \rangle, \langle \partial_{\mathbf{z}}^{=(k+1)} g_i \rangle\right)$ has a trivial adjoint algebra for all $i \in [s]$ (see Definitions 30 and 32). That is, if $D : \langle \partial_{\mathbf{z}}^{=k} g_i \rangle \to \langle \partial_{\mathbf{z}}^{=k} g_i \rangle$ and $E : \langle \partial_{\mathbf{z}}^{=(k+1)} g_i \rangle \to \langle \partial_{\mathbf{z}}^{=(k+1)} g_i \rangle$ are linear maps s.t. $\partial_{z_j} D(p) = E(\partial_{z_j} p)$ for all $j \in [m]$ and all $p \in \langle \partial_{\mathbf{z}}^{=k} g_i \rangle$, then D, E are both identity maps (up to a scalar multiplication). Note that Corollary 39 implies that this condition is preserved if we apply an invertible linear transformation to the $\mathbf{z}$ variables.

The algorithm is stated in Algorithm 1. We will need the following lemma in the proof of the main theorem.

▶ **Lemma 4.** *Let $h \in \mathbb{F}[\mathbf{x}]_d$ be a homogeneous degree d polynomial and $\ell_1, \ldots, \ell_m \in \mathbb{F}[\mathbf{x}]_1$ be linearly independent linear forms. Then $h \in \mathbb{F}[\ell_1, \ldots, \ell_m]_d$ iff $\sum_{j=1}^n \alpha_j \partial_{x_j} h(\mathbf{x}) = 0$ for all $\alpha \in \mathbb{F}^n$ s.t. $\ell_i(\alpha) = 0$ for all $i \in [m]$.*

Proof. Let $\ell_i = \sum_{j=1}^n \ell_{ij} x_j$. In one direction, suppose $h \in \mathbb{F}[\ell_1, \ldots, \ell_m]_d$ so that $h = g(\ell_1, \ldots, \ell_m)$ for $g \in \mathbb{F}[\mathbf{z}]$, $\mathbf{z} = (z_1, \ldots, z_m)$. Then

$$\begin{aligned}\sum_{j=1}^n \alpha_j \partial_{x_j} h(\mathbf{x}) &= \sum_{j=1}^n \alpha_j \sum_{i=1}^m \ell_{ij} \partial_{z_i} g(\mathbf{z})|_{\mathbf{z}=(\ell_1,\ldots,\ell_m)} \\ &= \sum_{i=1}^m \ell_i(\alpha) \partial_{z_i} g(\mathbf{z})|_{\mathbf{z}=(\ell_1,\ldots,\ell_m)} \\ &= 0\end{aligned}$$

for all $\alpha \in \mathbb{F}^n$ s.t. $\ell_i(\alpha) = 0$ for all $i \in [m]$. In the other direction, suppose $\sum_{j=1}^n \alpha_j \partial_{x_j} h(\mathbf{x}) = 0$ for all $\alpha \in \mathbb{F}^n$ s.t. $\ell_i(\alpha) = 0$ for all $i \in [m]$. Extend $\ell_1, \ldots, \ell_m$ to a full basis of $\mathbb{F}[\mathbf{x}]_1$, $\ell_1, \ldots, \ell_n$ (in an arbitrary way). We can write h as $g(\ell_1, \ldots, \ell_n)$ for some $g \in \mathbb{F}[\mathbf{w}]$, $\mathbf{w} = (w_1, \ldots, w_n)$. Our goal now is to prove that $\partial_{w_i} g(\mathbf{w}) = 0$ for all $i \in \{m+1, \ldots, n\}$. Now

$$\sum_{j=1}^{n} \alpha_j \partial_{x_j} h(\mathbf{x}) = \sum_{j=1}^{n} \alpha_j \sum_{i=1}^{n} \ell_{ij} \partial_{w_i} g(\mathbf{w})|_{\mathbf{w}=(\ell_1,\ldots,\ell_n)}$$
$$= \sum_{i=1}^{n} \ell_i(\alpha) \partial_{w_i} g(\mathbf{w})|_{\mathbf{w}=(\ell_1,\ldots,\ell_n)}.$$

For $i \in \{m+1, \ldots, n\}$, we can choose an α s.t. $\ell_j(\alpha) = 0$ for all $j \neq i$ and $\ell_i(\alpha) \neq 0$. Then from the assumption and the above calculation we get that $\partial_{w_i} g(\mathbf{w})|_{\mathbf{w}=(\ell_1,\ldots,\ell_n)} = 0$. Since $\ell_1, \ldots, \ell_n$ are linearly independent, we get that $\partial_{w_i} g(\mathbf{w}) = 0$ for all $i \in \{m+1, \ldots, n\}$. ◀

Algorithm 1 Learning generalized depth three circuits.

Input: black-box access to an $f \in \mathbb{F}[\mathbf{x}]_d$ that is computed by a non-degenerate homogeneous generalized depth three circuit i.e., $f = T_1 + \cdots + T_s$, where $T_i = g_i(\ell_{i1}, \ldots, \ell_{im})$ for $i \in [s]$.
Output: s black-boxes $\mathcal{B}_1, \ldots, \mathcal{B}_s$ such that there exists a permutation $\pi : [s] \to [s]$ s.t. $\mathcal{B}_i$ provides black-box access to $T_{\pi(i)}$.
Subroutines:

1. Computing black-boxes for partial derivatives from the black-box for a polynomial. (Fact 29)
2. Vector space decomposition (Algorithm 2 and Corollary 41).

Parameters: The order of partial derivatives: k.

```
Compute black-boxes for a basis of the vector spaces $U := \langle \partial^{=k} f \rangle$ and $V := \langle \partial^{=(k+1)} f \rangle$ using Subroutine 1.
Using Subroutine 2, obtain a vector space decomposition $U = U'_1 \oplus \cdots \oplus U'_{s'}$ and $V = V'_1 \oplus \cdots \oplus V'_{s'}$ for the triple $\left(\partial^{=1}, U, V\right)$. If $s' \neq s$, then abort. Otherwise continue.
For each $\boldsymbol{\alpha}$ s.t. $\sum_{i=1}^{n} \alpha_i = k$, write the corresponding differential operator acting on $f$, $\partial_{\boldsymbol{\alpha}} f$, as $u'_{\boldsymbol{\alpha}1} + \cdots + u'_{\boldsymbol{\alpha}s}$ with $u'_{\boldsymbol{\alpha}i} \in U'_i$ (note that there is a unique such representation). We only obtain black-boxes for the polynomials $u'_{\boldsymbol{\alpha}i}$'s. This step can be carried out using Corollary 41.
The black-box $\mathcal{B}_i$ on input $\mathbf{x}$ will output $\frac{(d-k)!}{d!} \sum_{\boldsymbol{\alpha}} \binom{k}{\alpha_1,\ldots,\alpha_n} \mathbf{x}^{\boldsymbol{\alpha}}\, u'_{\boldsymbol{\alpha}i}(\mathbf{x})$.
```

The next theorem states the correctness of Algorithm 1 assuming the non-degeneracy conditions.

▶ **Theorem 5.** *Suppose the non-degeneracy conditions stated above are satisfied. Then Algorithm 1 never aborts. Suppose $\mathcal{B}_1, \ldots, \mathcal{B}_s$ be the black-boxes output by the algorithm. Then there exists a permutation $\pi : [s] \to [s]$ s.t. $\mathcal{B}_i$ is a black-box for $T_{\pi(i)}$.*

Proof. It suffices to prove uniqueness of decomposition for the triple $\left(\partial^{=1}, U, V\right)$ (see Definition 33). Assuming uniqueness of decomposition, $s' = s$ and there exists a permutation $\pi : [s] \to [s]$ s.t. $U'_i = U_{\pi(i)}$ and $V'_i = V_{\pi(i)}$. Since the U'_i's form a direct sum, there is a unique way of writing each element $u \in U$ as $u = u'_1 + \cdots + u'_s$ with $u'_i \in U'_i$. For $u = \partial_{\boldsymbol{\alpha}} f$, $u = \partial_{\boldsymbol{\alpha}} T_{\pi(1)} + \cdots + \partial_{\boldsymbol{\alpha}} T_{\pi(s)}$ is one such decomposition and hence the only one. Thus $u'_{\boldsymbol{\alpha}i} = \partial_{\boldsymbol{\alpha}} T_{\pi(i)}$ in which case $\mathcal{B}_i$ computes the black-box for $T_{\pi(i)}$ by Lagrange's formula,

$$h(\mathbf{x}) = \frac{(d-k)!}{d!} \sum_{\boldsymbol{\alpha}} \binom{k}{\alpha_1, \ldots, \alpha_n} \mathbf{x}^{\boldsymbol{\alpha}}\, \partial_{\boldsymbol{\alpha}} h(\mathbf{x})$$

for a homogeneous degree d polynomial h.

To prove uniqueness of decomposition, it suffices to prove that the adjoint algebra for the triple $\left(\partial^{=1}, U, V\right)$ is trivial because of Lemma 34. Consider linear maps $D : U \to U$ and $E : V \to V$ s.t. $\partial_{x_j} D(u) = E(\partial_{x_j} u)$ for all $u \in U$. Then we need to prove that $D(U_i) \subseteq U_i$, $E(V_i) \subseteq V_i$ for all $i \in [s]$ and that (D, E) are scalar multiples of identity when restricted to (U_i, V_i) respectively. The latter follows from Item 3 in the non-degeneracy conditions, so we only need to prove the former. To prove the former, consider (D, E) in the adjoint algebra. Note that $U_i \subseteq \mathbb{F}[\ell_{i1}, \ldots, \ell_{im}]_{d-k}$ and $V_i \subseteq \mathbb{F}[\ell_{i1}, \ldots, \ell_{im}]_{d-k-1}$. Hence if $u \in U_i$, then

$$\sum_{j=1}^{n} \alpha_j \partial_{x_j} u(\mathbf{x}) = 0$$

for all α s.t. $\ell_{i1}(\alpha) = \cdots = \ell_{im}(\alpha) = 0$, by Lemma 4. Because of the relation $\partial_{x_j} D(u) = E(\partial_{x_j} u)$, we get that

$$\sum_{j=1}^{n} \alpha_j \partial_{x_j} D(u)(\mathbf{x}) = 0$$

for all α s.t. $\ell_{i1}(\alpha) = \cdots = \ell_{im}(\alpha) = 0$. Hence by Lemma 4, we get that $D(u) \in \mathbb{F}[\ell_{i1}, \ldots, \ell_{im}]_{d-k}$. Hence $D(u) \in U \cap \mathbb{F}[\ell_{i1}, \ldots, \ell_{im}]_{d-k} = U_i$ (because of the direct sum structure of the vector spaces $\mathbb{F}[\ell_{i1}, \ldots, \ell_{im}]_{d-k}$ in Item 1). This completes the proof that $D(U_i) \subseteq U_i$. Now the space V_i has a basis which consists of a subset of polynomials from $\partial_{\boldsymbol{\beta}} T_i$ as $\boldsymbol{\beta}$ varies over monomials of degree $k+1$. We can write $\partial_{\boldsymbol{\beta}} T_i$ as $\partial_{x_j} \partial_{\boldsymbol{\alpha}} T_i$ for some $j \in [n]$ and some $\boldsymbol{\alpha}$ of degree k. Then

$$E(\partial_{\boldsymbol{\beta}} T_i) = E(\partial_{x_j} \partial_{\boldsymbol{\alpha}} T_i) = \partial_{x_j} D(\partial_{\boldsymbol{\alpha}} T_i).$$

Since $D(\partial_{\boldsymbol{\alpha}} T_i) \in U_i$, we get that $E(\partial_{\boldsymbol{\beta}} T_i) \in V_i$. This completes the proof that $E(V_i) \subseteq V_i$. ◀

We will now proceed to proving Theorem 1.

Proof of Theorem 1. We will run Algorithm 1 on the given black-box with the parameter k being set to $\lceil \frac{2 \log s}{\log n} \rceil$. Notice that, by Fact 29, the time complexity of subroutine 1 is $\mathrm{poly}(d^k, n) = \mathrm{poly}(s, n)$. See Remark 6. Since Theorem 5 guarantees the correctness of our output, we just have to verify its running time. Note that the time complexity of remaining steps is $\mathrm{poly}(n^k, s) = \mathrm{poly}(n, s)$, which concludes the proof. ◀

3 Non-degeneracy of random circuits

In this section we will show that if $n > (md)^2$, $s \leq n^{d/4}$ and $k = \lceil \frac{2 \log s}{\log n} \rceil$, then a *random* $(n, d, s, \{g_i\}_{i \in [s]})$ homogeneous generalized depth three circuit is non-degenerate with high probability. To better understand the regime of parameters, we record a few relations among the parameters n, d, s and k in the following easy to verify remark.

▶ Remark 6. If $s \leq n^{d/4}$, $md \leq \sqrt{n}$ and $k = \lceil \frac{2 \log s}{\log n} \rceil$ then $k \leq d/2$ and $\binom{m+k-1}{k} \leq ns$.

We will proceed by showing that each of our non-degeneracy conditions is satisfied for random circuits, and then the result will hold directly by the union bound. We also show that $(n, d, s, \{g_i\}_{i \in [s]})$ homogeneous generalized depth three circuits are non-degenerate if the g_i's belong to special polynomial families like $\mathrm{Det}_d, \mathrm{Perm}_d, \mathrm{IMM}_{r,d}, \mathrm{Sym}_{r,d}$ and only $\ell_{i,j}$'s are chosen randomly. This is because non-degeneracy condition 1 and 2 just depend[6] on $\ell_{i,j}$'s

[6] Strictly speaking, non-degeneracy condition 2 does depend on g_i's, but we show that it holds if we just pick $\ell_{i,j}$'s randomly. See Lemma 16

and non-degeneracy condition 3 depends on the g_i's, and we can show that aforementioned polynomial families satisfy these mild technical conditions required to show non-degeneracy condition 3.

3.1 Non-degeneracy of random circuits: Condition 1

Let's begin by restating our first non-degeneracy condition for a generalized depth three circuit $\sum\limits_{i=1}^{s} g_i(\ell_{i,1}, \ell_{i,2}, \ldots, \ell_{i,m})$.

Non-degeneracy condition 1. The vector spaces $W_1^{(d-k)} := \mathbb{F}[\ell_{11}, \ldots, \ell_{1m}]_{d-k}, \ldots, W_s^{(d-k)} := \mathbb{F}[\ell_{s1}, \ldots, \ell_{sm}]_{d-k}$ form a direct sum i.e.

$$W_1^{(d-k)} + \cdots + W_s^{(d-k)} = W_1^{(d-k)} \oplus \cdots \oplus W_s^{(d-k)}.$$

And, the same assumption for the vector spaces $W_i^{(d-k-1)}$'s.

We will show that if $m \leq \frac{\sqrt{n}}{t}$ and $s \leq n^{t/2}$ then a random choice of $\{\ell_{i,j}\}_{(i,j)\in([s],[m])}$ satisfies the equality $\sum_{i=1}^{s} W_i^{(t)} = \oplus W_i^{(t)}$. To show this we will need the notion of *combinatorial designs.*

▶ **Definition 7** (Nisan-Wigderson designs [29]). *A family of sets $\mathcal{A} = \{A_1, \ldots, A_s\}$ is said to be an (n, m, d) design if $A_i \subseteq [n]$ with $|A_i| = m$ for all $i \in [s]$. And, for $i \neq j$, $|A_i \cap A_j| < d$.*

We will be using a standard construction of such designs based on the Reed-Solomon codes.

▶ **Lemma 8** (Explicit Design). *Let $m \leq \sqrt{n}$. There exists an (n, m, d)-design $\{A_1, \ldots, A_s\}$ for $s \leq m^d$.*

▶ **Lemma 9.** *Let $S \subseteq \mathbb{F}$ be a finite set. If $m \leq \frac{\sqrt{n}}{t}$ and $s \leq n^{t/2}$ then for a random choice of $\{\ell_{i,j}\}_{(i,j)\in([s],[m])}$ linear forms over S,*

$$\sum_{i=1}^{s} W_i^{(t)} = \oplus_{i\in[s]} W_i^{(t)}$$

with probability at least $1 - \frac{s\cdot\binom{m+t-1}{t}\cdot t}{|S|}$.

For proof of the above lemma see the full version.

As a direct consequence of Lemma 9 for $t = d - k - 1$ and $t = d - k$ we get that the non-degeneracy condition 1 holds with high probability.

▶ **Corollary 10.** *If $(md)^2 \leq n, k = \lceil 2\frac{\log s}{\log n}\rceil$, $s \leq n^{d/4}$ and $|S| \geq \mathrm{poly}(n^d)$ then for a random choice of $\{\ell_{i,j}\}_{(i,j)\in([s],[m])}$ linear forms over a set S, $\sum_{i=1}^{s} W_i^{(d-k)} = \oplus_{i\in[s]} W_i^{(d-k)}$ and $\sum_{i=1}^{s} W_i^{(d-k-1)} = \oplus_{i\in[s]} W_i^{(d-k-1)}$ with probability $1 - o(1)$.*

3.2 Non-degeneracy of random circuits: Condition 2

Our next non-degeneracy condition for $f(\mathbf{x}) = \sum_{i=1}^{s} g_i(\ell_{i,1}, \ell_{i,2}, \ldots, \ell_{i,m})$ requires that the vector spaces $U := \langle \partial^{=k} f\rangle$ and $V := \langle \partial^{=(k+1)} f\rangle$ have a direct sum structure. That is,

$$U = U_1 \oplus \cdots \oplus U_s \text{ and } V = V_1 \oplus \cdots \oplus V_s, \tag{4}$$

where $U_i := \langle \partial^{=k} T_i \rangle$ and $V_i := \langle \partial^{=(k+1)} T_i \rangle$ where $T_i := g_i(\ell_{i,1}, \ell_{i,2}, \ldots, \ell_{i,m})$. Note that as $U_i \subseteq W_i^{(d-k)}$, the direct sum structure of $W_i^{(d-k)}$ directly gives $U_1 + U_2 + \ldots + U_s = U_1 \oplus U_2 \oplus \ldots \oplus U_s$. Indeed for the regime of parameters we are interested in, $W_i^{(d-k)}$ do have a direct sum structure for random $\ell_{i,j}$'s by Lemma 9. Thus in order to show non-degeneracy condition 2 for random circuits, it suffices to show

$$U = U_1 + U_2 + \ldots + U_s. \tag{5}$$

Clearly, $U \subseteq U_1 + U_2 + \ldots + U_s$. To show the other direction, it suffices to show that $U \supseteq U_i$ for all $i \in [s]$. We show this via a novel technique of studying the space of partial derivative operators (i.e. $\langle \partial^{=k} \rangle$) themselves, as opposed to the space when they are applied to a polynomial (i.e. $\langle \partial^{=k} f \rangle$). Interestingly, our proof is very general and works for action of any general linear operators on a space! Thus, we state and prove it in full generality and later instantiate the setting needed for our work.

We start by elaborating on our abstract setting. Let $f = T_1 + T_2 + \ldots + T_s$ where $T_i \in \mathcal{C}_i$ and $\mathcal{L}$ is a vector space of linear operators from $\mathbb{F}[\mathbf{x}]$ to W. Here, $\mathcal{C}_i$ is a circuit class consisting of polynomials in $\mathbb{F}[x]$. Also, let $\mathcal{B} : \mathcal{L} \times \mathcal{L} \to \mathbb{F}$ be a non-degenerate bilinear form, that is for any non-zero $u \in \mathcal{L}$ there exists a $v \in \mathcal{L}$ s.t. $\mathcal{B}(u, v) \neq 0$. Furthermore, let $\mathcal{L}_i^{\perp} := \{L \in \mathcal{L} \mid Lh = 0, \forall\, h \in \mathcal{C}_i\}$. Using the bilinear product $\mathcal{B}$ and any subspace U of $\mathcal{L}$, we define $U^{\perp_{\mathcal{B}}}$ as the linear operators (in $\mathcal{L}$) s.t. for all $u \in U$ the bilinear product is 0. Formally, $U^{\perp_{\mathcal{B}}} := \{L \in \mathcal{L} \mid \forall\, u \in U, \mathcal{B}(L, u) = 0\}$.

Our next lemma shows that under a direct sum structure of $\sum\limits_{i \in [s]} (\mathcal{L}_i^{\perp})^{\perp_{\mathcal{B}}}$, $\mathcal{L}(f) = \sum\limits_{i \in [s]} \mathcal{L}(T_i)$.

▶ **Lemma 11.** *Let $\mathcal{L}, \mathcal{B}, f(\mathbf{x})$, and T_i's be as defined above. If $\sum\limits_{i \in [s]} (\mathcal{L}_i^{\perp})^{\perp_{\mathcal{B}}} = \oplus_{i \in [s]} (\mathcal{L}_i^{\perp})^{\perp_{\mathcal{B}}}$ then $\mathcal{L}(f) = \sum_i \mathcal{L}(T_i)$.*

For proof of the above lemma see the full version.

For our application of showing $U \supset U_i$, we set $\mathcal{L} = \langle \partial^{=k} \rangle$, $\mathcal{C}_i$ is the class of polynomials $g_i(\ell_{i,1}, \ell_{i,2}, \ldots, \ell_{i,m})$ where g_i is an m-variate degree d polynomial and $\ell_{i,1}, \ell_{i,2}, \ldots, \ell_{i,m}$ are random n-variate linear forms. Also, $\mathcal{L}_i^{\perp} = \mathcal{D}_i^{\perp} := \{D \in \langle \partial^{=k} \rangle \mid Dh = 0, \forall h \in W_i^{(d)}\}$. Note that in order to apply Lemma 11 in our setting, we have to come up with a non-degenerate bilinear map $\mathcal{B}$, s.t. $\sum_i (\mathcal{D}_i^{\perp})^{\perp_{\mathcal{B}}} = \oplus_i (\mathcal{D}_i^{\perp})^{\perp_{\mathcal{B}}}$. Let's first note that if $(\mathcal{D}_i^{\perp})^{\perp_{\mathcal{B}}}$ does satisfy the direct sum property then we are done! Indeed, on setting, $\mathcal{L} = \langle \partial^{=k} \rangle$ and $\mathcal{L}_i^{\perp} = \mathcal{D}_i^{\perp}$ to $\mathcal{L}(f) = \sum_i \mathcal{L}(T_i)$ gives $\langle \partial^{=k}(f) \rangle = \sum_i \langle \partial^{=k} \rangle (T_i)$, thus implying (5).

In the rest of the section, we will focus on coming up with a bilinear form and showing that it is indeed non-degenerate. And later, via another application of Lemma 9, show the direct sum structure of $(\mathcal{D}_i^{\perp})^{\perp_{\mathcal{B}}}$. For two polynomials f and g, define

$$\mathcal{B}(f, g) := f\left(\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \ldots, \frac{\partial}{\partial x_n}\right) \cdot g.$$

This inner product among two polynomials is known as apolar inner product, and is a fundamental notion with a lot of applications, see [30] and the references therein. It is easy to see that $\mathcal{B}(\ell_1, \ell_2) = v_{\ell_1} \cdot v_{\ell_2}$; where $\ell_1(\mathbf{x}), \ell_2(\mathbf{x})$ are two linear forms, v_{ℓ_1}, v_{ℓ_2} are canonical vectors associated with them and $v_{\ell_1} \cdot v_{\ell_2}$ is the standard dot product among vectors. The non-degenerate bilinear map needed for our purpose acts on $\mathcal{L} \times \mathcal{L}$ instead of polynomials as defined above. But in our case $\mathcal{L} = \langle \partial^{=k} \rangle$ is nothing but polynomials of degree k with $\frac{\partial}{\partial x_i}$ as variables, thus the definition of $\mathcal{B}$ extends naturally.

In order to show that our bilinear map is non-degenerate, it will be convenient to work with an orthogonal basis of $\ell_{i,j}$'s. We will therefore need the following lemma.

▶ **Lemma 12.** *When* $\mathrm{char}(\mathbb{F}) \neq 2$*, there exists an orthogonal basis of any finite dimensional vector space over* $\mathbb{F}$ *with respect to any non-degenerate bilinear (dot) product.*

For proof of the above lemma see the full version.

Let V be the space of some linear forms in $\mathbb{F}[\mathbf{x}]$, then by the above lemma one can assume that there exist an orthogonal basis of V as long as $\mathrm{char}(\mathbb{F}) \neq 2$. Now, we will state an observation on what is $\mathcal{B}(f, \cdot)$ when f and g are expressed as polynomials in an orthogonal basis.

▶ **Observation 13.** *If* $\ell_1(\mathbf{x}), \ldots, \ell_n(\mathbf{x})$ *is an orthogonal basis of* $\mathbb{F}[\mathbf{x}]_1$ *then if* $g = \sum_{\boldsymbol{\alpha}} c_{\boldsymbol{\alpha}} \boldsymbol{\ell}^{\boldsymbol{\alpha}}$ *and* $f = \sum_{\boldsymbol{\alpha}} d_{\boldsymbol{\alpha}} \boldsymbol{\ell}^{\boldsymbol{\alpha}}$ *are degree* d *polynomials. Then*

$$\mathcal{B}(f, g) = \sum_{\boldsymbol{\alpha}} c_{\boldsymbol{\alpha}} \cdot d_{\boldsymbol{\alpha}}\, \boldsymbol{\alpha}!.$$

Here $\boldsymbol{\alpha}! = \prod\limits_{i \in [n]} \alpha_i!$ *and* $\boldsymbol{\alpha}$ *as an index varies over exponent vector of* n*-variate monomials of degree exactly* d*.*

The above observation follows directly by observing it when g is a monomial and extending by linearity. Now, if $\mathrm{char}(\mathbb{F}) > d$ or 0, then using this observation we directly get that $\mathcal{B}(f, g)$ is non-degenerate.

▶ **Lemma 14.** *The bi-linear map* $\mathcal{B}(f, g)$ *over* $\mathbb{F}[\mathbf{x}]_d$ *is non-degenerate if* $\mathrm{char}(\mathbb{F}) > d$ *or* 0*. That is for all non-zero* $g \in \mathbb{F}[\mathbf{x}]_d$ *there exist* $f \in \mathbb{F}[\mathbf{x}]_d$ *s.t.* $\mathcal{B}(f, g) \neq 0$*.*

Proof. Let $\ell_1(\mathbf{x}), \ldots, \ell_n(\mathbf{x})$ be an orthogonal basis of $\mathbb{F}[\mathbf{x}]_1$. Now, for any $g = \sum_{\boldsymbol{\alpha}} c_{\boldsymbol{\alpha}} \ell^{\boldsymbol{\alpha}} \neq 0$, let $\boldsymbol{\alpha}_o$ be an exponent vector s.t. $c_{\boldsymbol{\alpha}_o} \neq 0$. On choosing $f = \ell^{\boldsymbol{\alpha}_o}$ we get $\mathcal{B}(f, g) = c_{\boldsymbol{\alpha}_o} \boldsymbol{\alpha}_o! \neq 0$ if $\mathrm{char}(\mathbb{F}) > d$ or 0. ◀

3.2.1 Direct sum structure of $(\mathcal{D}_i^{\perp})^{\perp_{\mathcal{B}}}$

The only thing left to show non-degeneracy condition 2 is a direct sum structure on derivative operators $(\mathcal{D}_i^{\perp})^{\perp_{\mathcal{B}}}$. We will first study the space $D_i^{\perp}$, as it will help us show the required direct sum structure. Let's assume (WLOG) that $\ell_{i,1}, \ldots \ell_{i,m}, q_{i,1}, \ldots q_{i,n-m}$ is an orthogonal basis of $\mathbb{F}^n$ w.r.t. $\mathcal{B}$. That is, for $i \neq j$ $\langle \ell_{1,i}, \ell_{1,j} \rangle = 0$, $\langle \ell_{1,i}, q_{1,j} \rangle = 0$ and $\langle \ell_{1,i}, \ell_{1,i} \rangle \neq 0$. Notice that,

$$\mathcal{D}_i^{\perp} \supseteq q_{i,1} \cdot \langle \partial^{=(k-1)} \rangle + q_{i,2} \cdot \langle \partial^{=(k-1)} \rangle + \ldots + q_{i,n-m} \cdot \langle \partial^{=(k-1)} \rangle. \tag{6}$$

▷ Claim 15. Let $\mathrm{char}(\mathbb{F}) > d$ or $\mathrm{char}(\mathbb{F}) = 0$, then $W_i^{(k)} := \mathbb{F}[\ell_{i,1}, \ell_{i,2}, \ldots \ell_{i,m}]_k \supseteq (\mathcal{D}_i^{\perp})^{\perp_{\mathcal{B}}}$.

Proof. For brevity we will denote the space $(q_{i,1} \cdot \langle \partial^{=(k-1)} \rangle + q_{i,2} \cdot \langle \partial^{=(k-1)} \rangle + \ldots + q_{i,n-m} \cdot \langle \partial^{=(k-1)} \rangle)$ by Q. We have that $\mathcal{D}_i^{\perp} \supseteq Q$, thus $(\mathcal{D}_i^{\perp})^{\perp_{\mathcal{B}}} \subseteq Q^{\perp_{\mathcal{B}}}$. The proof concludes by showing that $Q^{\perp_{\mathcal{B}}} = W_i^{(k)}$. Clearly, $Q^{\perp_{\mathcal{B}}} \supseteq W_i^{(k)}$. For the other direction, let $p \in Q^{\perp_{\mathcal{B}}}$ s.t. $p \notin W_i^{(k)}$. We can write, $p = w + q$ where $w \in W_i^{(k)}$ and $q \in Q$ s.t. $q \neq 0$. Now, since $p \in Q^{\perp_{\mathcal{B}}}$, $\mathcal{B}(p, q') = 0$ $\forall q' \in Q$. Notice that for any $q' \in Q$, $\mathcal{B}(w, q') = 0$. Thus, $\mathcal{B}(p, q') = \mathcal{B}(q{+}w, q') = \mathcal{B}(q, q')$. Now, just like in the proof of Lemma 14 we can choose $q' \in Q$ s.t. $\mathcal{B}(q, q') \neq 0$. That is, pick q' to be any monomial in $\mathbb{F}[\ell_{i,1}, \ell_{i,2}, \ldots \ell_{i,m}, q_{i,1}, \ldots q_{i,m}]_d$ with non-zero coefficient in q and by observation 13 we get that $\mathcal{B}(q, q') \neq 0$. This implies $p \in Q$ and $\mathcal{B}(p, q') \neq 0$ $p(\bar{\partial}) \cdot p(\bar{x}) \neq 0$, thus contradicting $p \in Q^{\perp_{\mathcal{B}}}$. ◁

▶ **Lemma 16.** *For homogeneous degree d polynomials $\{g_i\}_{i\in[s]}$, let $f = \sum_{i=1}^{s} g_i(\ell_{i,1}, \ell_{i,2}, \ldots, \ell_{i,m}) \in \mathbb{F}[\mathbf{x}]_d$ and $U := \langle \partial^{=k} f\rangle$, $V := \langle \partial^{=(k+1)} f\rangle$, $U_i := \langle \partial^{=k} T_i\rangle$, $V_i = \langle \partial^{=(k+1)} T_i\rangle$. If* $\mathrm{char}(\mathbb{F}) > d$ *or* $\mathrm{char}(\mathbb{F}) = 0, (md)^2 \le n, k = \lceil 2\frac{\log s}{\log n}\rceil$ *and $s \le n^{d/4}$ then for a random choice of $\{\ell_{i,j}\}_{(i,j)\in([s],[m])}$ linear forms over a set $S \subset \mathbb{F}$ such that $|S| \ge \mathrm{poly}(n^d)$, $U = U_1 \oplus \cdots \oplus U_s$ and $V = V_1 \oplus \cdots \oplus V_s$ with probability at least $1 - o(1)$.*

Proof. By Lemma 9 we get that for a finite set $S \subseteq \mathbb{F}$, if $m \le \frac{\sqrt{n}}{d}$ and $s \le n^{d/4}$ then for a random choice of $\{\ell_{i,j}\}_{(i,j)\in([s],[m])}$ linear forms over S, $\sum_{i=1}^{s} W_i^{(k)} = \oplus W_i^{(k)}$ with probability at least $1 - o(1)$. Now, since $W_i^{(k)}$ has a direct sum structure, the same will hold for their respective subspaces. Thus, $\sum_{i\in[s]}(\mathcal{D}_i^{\perp})^{\perp_{\mathcal{B}}} = \oplus_{i\in[s]}(\mathcal{D}_i^{\perp})^{\perp_{\mathcal{B}}}$. Now, using Lemma 11 with $\mathcal{L} = \langle \partial^{=k}\rangle$ and $\mathcal{L}_i^{\perp} = \mathcal{D}_i^{\perp} := \{D \in \langle \partial^{=k}\rangle | \, Dh = 0, \forall h \in W_i^{(d)}\}$ implies $U = U_1 + U_2 + \ldots + U_s$. And, as $U_i \subseteq W_i^{(d-k)}$, the direct sum structure of $W_i^{(d-k)}$directly gives $U_1 + U_2 + \ldots + U_s = U_1 \oplus U_2 \oplus \ldots \oplus U_s$. Notice that, $W_i^{(d-k)}$ have direct sum structure by corollary 10 as $m \le \frac{\sqrt{n}}{d}$ and $s \le n^{d/4}$. The proof for $V = V_1 \oplus \cdots \oplus V_s$ is identical. ◀

3.3 Non-degeneracy condition 3: Adjoint algebra is trivial

We will start by restating non-degeneracy condition 3.

Non-degeneracy condition 3. For a generalized depth 3 circuit $f = \sum_{i=1}^{s} g_i(\ell_{i,1}, \ell_{i,2}, \ldots, \ell_{i,m})$ where $g_i \in \mathbb{F}[\mathbf{z}]_d$, $\mathbf{z} = (z_1, \ldots, z_m)$, the triple $\left(\partial_{\mathbf{z}}^{=1}, \langle \partial_{\mathbf{z}}^{=k} g_i\rangle, \langle \partial_{\mathbf{z}}^{=(k+1)} g_i\rangle\right)$ has a trivial adjoint algebra for all $i \in [s]$. That is, if $D : \langle \partial_{\mathbf{z}}^{=k} g_i\rangle \to \langle \partial_{\mathbf{z}}^{=k} g_i\rangle$ and $E : \langle \partial_{\mathbf{z}}^{=(k+1)} g_i\rangle \to \langle \partial_{\mathbf{z}}^{=(k+1)} g_i\rangle$ are linear maps s.t. $\partial_{z_j} D(p) = E(\partial_{z_j} p)$ for all $p \in \langle \partial_{\mathbf{z}}^{=k} g_i\rangle$, then D, E are both identity maps (up to a scalar multiple).

We will show this for random g_i's as well as various interesting polynomial families like monomials, determinant, permanent, elementary symmetric polynomial and iterated matrix multiplication. This is done by observing that under mild technical conditions on g, the triple $\left(\partial_{\mathbf{z}}^{=1}, \langle \partial_{\mathbf{z}}^{=k} g\rangle, \langle \partial_{\mathbf{z}}^{=(k+1)} g\rangle\right)$ has a trivial adjoint algebra. Define $\{\partial_{\mathbf{z}}^{=k} g\} := \{\partial_{m}^{=k} g \mid \forall\, m$ degree k monomials in $\mathbb{F}[\mathbf{z}]\}$. And, let $var(f)$ denote the set of variables f depends on. We start by stating our technical condition:

▶ **Technical condition 17.** *Let $g \in \mathbb{F}[\mathbf{z}]_d$, we need $var(p) \ne var(p')$ for any non-zero (and distinct) $p, p' \in \{\partial_{\mathbf{z}}^{=k} g\}$. And all non-zero elements of $\{\partial_{\mathbf{z}}^{=k+1} g\}$ to be $\mathbb{F}$-linearly independent.*

▶ **Lemma 18.** *Let $g \in \mathbb{F}[\mathbf{z}]_d$ that satisfies condition 17 and $D : \langle \partial_{\mathbf{z}}^{=k} g\rangle \to \langle \partial_{\mathbf{z}}^{=k} g\rangle$ and $E : \langle \partial_{\mathbf{z}}^{=(k+1)} g\rangle \to \langle \partial_{\mathbf{z}}^{=(k+1)} g\rangle$ be two linear maps. If $\forall j \in [m]$ and all $p \in \langle \partial_{\mathbf{z}}^{=k} g\rangle$, $\partial_{z_j} D(p) = E(\partial_{z_j} p)$, then $D(p) = c_p \cdot p$ for all $p \in \{\partial_{\mathbf{z}}^{=k} g\}$, where $c_p \in \mathbb{F}$ which could depend on p.*

For proof of the above lemma see the full version.

Note that, the above lemma *doesn't* imply that the adjoint algebra is trivial, as c_{p_i} could possibly depend on g_i. To show that the adjoint algebra is trivial, we need to prove that c_{p_i} is the same constant for all p_i's. In order to do that we will need the following notion of graph associated with a polynomial.

▶ **Definition 19.** *For a polynomial $g \in \mathbb{F}[\mathbf{z}]_d$, let G_g^k be the graph whose vertices are degree k multilinear monomials m s.t. $\partial_m^{=k} g \ne 0$ and the edge set consist of pairs of monomials (m_1, m_2) with $\Delta(m_1, m_2) = 2$ and $\partial_{lcm(m_1,m_2)}^{=k+1} g \ne 0$. Here $\Delta(m_1, m_2)$ is the hamming distance among the exponent vectors of m_1 and m_2.*

▶ **Lemma 20.** *Let $g \in \mathbb{F}[\mathbf{z}]_d$ be a polynomial s.t. it satisfies condition 17. Additionally, if G_g^k is connected, then the triple $\left(\partial_{\mathbf{z}}^{=1}, \langle \partial_{\mathbf{z}}^{=k} g\rangle, \langle \partial_{\mathbf{z}}^{=(k+1)} g\rangle\right)$ has a trivial adjoint algebra.*

For proof of the above lemma see the full version.

It is an easy exercise to see that for a *random* multilinear g condition 17 is satisfied and G_f^k is connected. We will now show the same holds for various polynomial families which includes monomials, determinant, permanent, elementary symmetric polynomial and iterated matrix multiplication. The argument for showing connectivity of G_g^k stems from this simple observation.

▶ **Observation 21.** *If m_1 and m_2 are degree k monomials with $\Delta(m_1, m_2) = \delta$ and let $m_1 = m^{(0)}, m^{(1)}, \dots, m^{(\delta)} = m_2$ be a path made of distance two monomials (i.e. $\Delta(m^{(i-1)}, m^{(i)}) = 2$ for $i \in [\delta]$) from m_1 to m_2 s.t. $\partial_{\tilde{m}^{(i)}} g \neq 0$ ($\tilde{m}^{(i)} := lcm(m^{(i)}, m^{(i+1)})$) then m_1 and m_2 are connected.*

▶ **Lemma 22.** *If g is one of the following polynomials Det_d, $Perm_d$ (with $3k \leq d$); $Sym_{r,d}$, monomial (with $k+1 < d$) or $IMM_{r,d}$ (with $3k \leq d$) then G_g^k is connected and condition 17 is satisfied.*

For proof of the above lemma, see the full version.

3.4 Adjoint algebra for random g_i's

We will now show that the adjoint algebra is trivial for *random* g_i's. This is done by showing that the adjoint algebra is trivial if the space spanned by k-th order partial derivatives applied to g have full dimension. Followed by observing that random g_i's have this property.

▶ **Lemma 23.** *Let $g \in \mathbb{F}[\mathbf{z}]_d$ be a polynomial such that $\dim\left(\langle \partial^{=k} g\rangle\right) = \binom{k+m-1}{k}$. Also, let $U_g = \langle \partial^{=k} g\rangle$, $V_g = \langle \partial^{=k+1} g\rangle$ and $D : U_g \to U_g$ and $E : V_g \to V_g$ be any linear maps. If for all $j \in [m]$ and $p \in U_g$, $\partial_{z_j} D(p) = E(\partial_{z_j} p)$, then D and E are identity maps up to a scalar multiple. That is, the triple $\left(\partial^{=1}, \langle \partial^{=k} g\rangle, \langle \partial^{=(k+1)} g\rangle\right)$ has a trivial adjoint algebra.*

For proof of the above lemma see the full version.

We can instantiate the above lemma for *random* g_i's. Indeed, the condition $\dim\left(\partial^{=(k)} g\right) = \binom{k+m-1}{k}$ boils down to showing that the determinant of a matrix with dimension $\binom{k+m-1}{k}$ is non-zero. And, there exist standard constructions of explicit polynomials with this property (see [8]). Thus, via the Schwartz-Zippel lemma, we get the following corollary.

▶ **Corollary 24.** *For a random choice of degree d homogeneous polynomials $\{g_i\}_{i \in [s]}$ over a set S, the triple $\left(\partial_{\mathbf{z}}^{=1}, \langle \partial_{\mathbf{z}}^{=k} g_i\rangle, \langle \partial_{\mathbf{z}}^{=(k+1)} g_i\rangle\right)$ has a trivial adjoint algebra for all $i \in [s]$, with probability at least $1 - \frac{sd \cdot \binom{m+k-1}{k}}{|S|}$.*

Now, we can combine corollary 10, 24 and Lemma 22, 16 to show that a random generalized depth 3 circuit is non-degenerate with high probability.

▶ **Lemma 25** (Random generalized depth 3 circuits are non-degenerate)**.** *Let $\mathcal{C} \equiv \sum_{i=1}^{s} g_i(\ell_{i,1}, \ell_{i,2}, \dots, \ell_{i,m})$ be a homogeneous generalized depth 3 circuit, where $\{g_i\}_{i \in [s]}$ are homogeneous degree d polynomials, and $n \geq (md)^2$. Suppose the coefficients of $\ell_{i,j}$'s are chosen uniformly and independently at random from a set $S \subset \mathbb{F}$ of size $|S| \geq \mathrm{poly}(n^d, s)$. Additionally, suppose one of the following cases is true:*

1. *g_i's belong to one of the polynomial families: $Det_d, Perm_d, IMM_{r,d}, Sym_{r,d}$, monomials with $s \leq n^{d/6}$.*
2. *Coefficients of all g_i's are chosen uniformly and independently at random from S and $s \leq n^{d/4}$.*

Then, with probability $1 - o(1)$, $\mathcal{C}$ is non-degenerate.

The proof is immediate using union bound along with Remark 6 and hence omitted. As a direct consequence of Lemma 25 and Theorem 5, we get the following theorem about learning random generalized depth circuits.

▶ **Theorem 26** (Learning random generalized depth 3 circuits). *Let $\mathcal{C} \equiv \sum_{i=1}^{s} g_i(\ell_{i,1}, \ell_{i,2}, \ldots, \ell_{i,m})$ be a homogeneous generalized depth 3 circuit, where $\{g_i\}_{i \in [s]}$ are homogeneous degree d polynomials, and $n \geq (md)^2$. Suppose the coefficients of $\ell_{i,j}$'s are chosen uniformly and independently at random from a set $S \subset \mathbb{F}$ of size $|S| \geq \text{poly}(n^d, s)$ and $\text{char}(\mathbb{F}) > d$ or char$(\mathbb{F}) = 0$. Additionally, suppose one of the following cases is true:*

1. *g_i's belong to one of the polynomial families: $Det_d, Perm_d, IMM_{r,d}, Sym_{r,d}$, monomials with $s \leq n^{d/6}$.*
2. *Coefficients of all g_i's are chosen uniformly and independently at random from S and $s \leq n^{d/4}$.*

Then, given black-box access to $\mathcal{C}$ we can reconstruct it in randomized $\text{poly}(n, m, d, s)$ time.

Note that, the m subsumes the dependence on r in the above theorem. Also, the reconstruction algorithm of Theorem 26 is proper, i.e. it outputs a homogeneous generalized depth 3 circuit.

3.5 From black-box access to learning generalized depth three circuits

Theorem 5 gives a black-box for each g_i's under the technical conditions discussed. It is natural to ask if we can find $\ell_{i,j}$'s and a generalized depth 3 representation as well. This is related to the well studied equivalence-testing problem, specifically to the search version of it. The equivalence-testing question is the following: given polynomials f and g, find an invertible linear transformation A on variables such that $f = g(A\mathbf{x})$, if such A exists. Observe that if g_i belongs to a family for which we can solve the equivalence-testing problem, then we can find $\ell_{i,j}$'s as well. This follows directly by seeing each blackbox as an instance of equivalence-testing (search version). Note that in our non-degenerate setting, $\ell_{i,j}$'s are linearly independent for each $i \in [s]$ thus satisfies the requirement that the linear transformation has to be invertible. As a direct consequence of this we get the following corollary.

▶ **Corollary 27.** *Suppose we are given black-box access to f, an n-variate, homogeneous degree d polynomial computable by a generalized depth 3 circuit of size s, s.t. the non-degeneracy condition 1, 2 and 3 hold. Additionally, if each g_i belongs to a family of polynomials for which there exist a poly(n, m, d) time equivalence-testing algorithm. Then there exist a poly(s, n, d, m) time algorithm that learns a generalized depth 3 representation of f.*

Note that if g_i is just a monomial (the special case for depth 3 circuits) then equivalence-testing follows directly from black-box factoring [16]. Hence, when g_i's are monomials the previous corollary along with Lemma 22 (monomials satisfy non-degeneracy condition 3) gives an algorithm for learning non-degenerate homogenous depth three circuits! Thus, our result is truly a generalization of the result by [22].

In general, equivalence-testing is considered to be a very hard problem (see [19, 18]) but it has been solved in several interesting cases, we list some of them below. For ease of representation, let us define some notation representing the complexity of the search version of the polynomial equivalence problem over a particular field. Given $m, d, r \in \mathbb{N}$ and black-box access to an m-variate polynomial g of degree d, let $\mathrm{Eqv}_{\mathbb{F}}(r, d, m, f)$ denote the randomized time complexity of finding an invertible linear transformation A s.t. $g(\mathbf{x}) = f(A\mathbf{x})$ if it exists, otherwise output "no-solution".

▶ **Theorem 28.** *Following results are known for equivalence-testing of special families of polynomials:*

1. $Eqv_{\mathbb{F}}(r, d, m, Sym_{r,d}) = \mathrm{poly}(r, d, m)$*, if* $\mathrm{char}(\mathbb{F}) > d\, or\, 0$*. See [19].*
2. $Eqv_{\mathbb{F}}(r, d, m, Perm_r) = \mathrm{poly}(r, d, m)$*. See [19].*
3. $Eqv_{\mathbb{F}}(r, d, m, Det_r) = \mathrm{poly}(r, d, m)$ *if* $\mathrm{char}(\mathbb{F}) \nmid r(r-1)$ *or* $\mathbb{F} = \mathbb{C}$ *. See [19, 6].*
4. $Eqv_{\mathbb{F}}(r, d, m, IMM_{r,d}) = \mathrm{poly}(r, d, m)$ *if* $\mathrm{char}(\mathbb{F}) = 0$ *or greater than* d^c *(c some fixed constant). See [21].*

Thus, corollary 27 along with Theorem 28 and 26 gives a randomized $poly(n, d, m, s)$-time algorithm that outputs a generalized depth three representation, assuming $\ell_{i,j}$'s are chosen randomly, g_i's belong to one of the polynomial families: $\mathrm{Det}_d, \mathrm{Perm}_d, \mathrm{IMM}_{r,d}, \mathrm{Sym}_{r,d}$, monomials and the corresponding assumptions on $\mathbb{F}$ holds.

4 Conclusion and open problems

We design an algorithm for learning generalized depth three circuits in the non-degenerate case. We follow the learning from lower bounds framework of [22, 7] and design new tools for proving that non-degeneracy conditions hold for random circuits, which could be useful for other such problems. Our model captures widely applicable problems such as tensor decomposition and Tucker decomposition as special cases. We are hopeful that our algorithms will find powerful applications in machine learning. We list some of the most interesting open problems next.

1. **Going beyond tensor decomposition.** Can we capture more general and powerful problems in machine learning via the model of generalized depth three circuits?
2. **Making the algorithms noise-resilient.** Can we make our algorithms robust to noise? That is, if one is given (explicitly or black box access) $f(\mathbf{x}) = \sum_{i=1}^{s} g_i(\ell_{i1}, \ldots, \ell_{im}) + E(\mathbf{x})$, for some error term $E(\mathbf{x})$, can we approximately recover the summands? Such a noise-resilient version is relevant for machine learning applications. While our algorithm may seem too algebraic to be made robust, it is in fact linear algebraic and there is a good chance it can be made noise-resilient using standard tools such as SVD etc.
3. **Learning other arithmetic circuit models.** Can we learn other arithmetic circuit models in the non-degenerate case, for which we already have lower bounds? Perhaps the most appealing model to go for next is that of constant depth set-multilinear circuits. There are even new lower bounds for this model now [28].

References

1 Amos Beimel, Francesco Bergadano, Nader H. Bshouty, Eyal Kushilevitz, and Stefano Varricchio. Learning functions represented as multiplicity automata. *J. ACM*, 47(3):506–530, 2000. Conference version appeared in the proceedings of FOCS 1996.

2 Vishwas Bhargava, Shubhangi Saraf, and Ilya Volkovich. Reconstruction of depth-4 multilinear circuits. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 2144–2160. SIAM, 2020.

3 Vishwas Bhargava, Shubhangi Saraf, and Ilya Volkovich. Reconstruction algorithms for low-rank tensors and depth-3 multilinear circuits. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 809–822, 2021.

4 Lieven De Lathauwer. A survey of tensor methods. In *2009 IEEE International Symposium on Circuits and Systems*, pages 2773–2776. IEEE, 2009.

5 Lance Fortnow and Adam R. Klivans. Efficient learning algorithms yield circuit lower bounds. *J. Comput. Syst. Sci.*, 75(1):27–36, 2009. Conference version appeared in the proceedings of COLT 2006.

6 Ankit Garg, Nikhil Gupta, Neeraj Kayal, and Chandan Saha. Determinant equivalence test over finite fields and over q. In *46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

7 Ankit Garg, Neeraj Kayal, and Chandan Saha. Learning sums of powers of low-degree polynomials in the non-degenerate case. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 889–899. IEEE, 2020.

8 Fulvio Gesmundo and Joseph M. Landsberg. Explicit polynomial sequences with maximal spaces of partial derivatives and a question of k. mulmuley. *Theory of Computing*, 15(3):1–24, 2019. doi:10.4086/toc.2019.v015a003.

9 Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Approaching the Chasm at Depth Four. *J. ACM*, 61(6):33:1–33:16, 2014. Conference version appeared in the proceedings of CCC 2013.

10 Ankit Gupta, Neeraj Kayal, and Satyanarayana V. Lokam. Efficient Reconstruction of Random Multilinear Formulas. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 778–787, 2011.

11 Ankit Gupta, Neeraj Kayal, and Satyanarayana V. Lokam. Reconstruction of depth-4 multilinear circuits with top fan-in 2. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 625–642, 2012.

12 Ankit Gupta, Neeraj Kayal, and Youming Qiao. Random arithmetic formulas can be reconstructed efficiently. *Computational Complexity*, 23(2):207–303, 2014. Conference version appeared in the proceedings of CCC 2013.

13 R Harshman. Foundations of the parafac procedure: Model and conditions for an explanatory factor analysis. *Technical Report UCLA Working Papers in Phonetics 16, University of California, Los Angeles, Los Angeles, CA*, 1970.

14 Johan Håstad. Tensor Rank is NP-Complete. *J. Algorithms*, 11(4):644–654, 1990. Conference version appeared in the proceedings of ICALP 1989.

15 Majid Janzamin, Rong Ge, Jean Kossaifi, Anima Anandkumar, et al. Spectral learning on matrices and tensors. *Foundations and Trends® in Machine Learning*, 12(5-6):393–536, 2019.

16 Erich Kaltofen and Barry M. Trager. Computing with Polynomials Given By Black Boxes for Their Evaluations: Greatest Common Divisors, Factorization, Separation of Numerators and Denominators. *J. Symb. Comput.*, 9(3):301–320, 1990.

17 Zohar Shay Karnin and Amir Shpilka. Reconstruction of generalized depth-3 arithmetic circuits with bounded top fan-in. In *Proceedings of the 24th Annual IEEE Conference on Computational Complexity, CCC 2009, Paris, France, 15-18 July 2009*, pages 274–285, 2009.

18 Neeraj Kayal. Efficient algorithms for some special cases of the polynomial equivalence problem. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, pages 1409–1421, 2011.

19 Neeraj Kayal. Affine projections of polynomials: extended abstract. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 643–662, 2012.

20 Neeraj Kayal, Vineet Nair, and Chandan Saha. Average-case linear matrix factorization and reconstruction of low width algebraic branching programs. *Computational Complexity*, 28(4):749–828, 2019.

21 Neeraj Kayal, Vineet Nair, Chandan Saha, and Sébastien Tavenas. Reconstruction of Full Rank Algebraic Branching Programs. In *32nd Computational Complexity Conference, CCC 2017, July 6-9, 2017, Riga, Latvia*, pages 21:1–21:61, 2017.

22 Neeraj Kayal and Chandan Saha. Reconstruction of non-degenerate homogeneous depth three circuits. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019.*, pages 413–424, 2019.

23 Adam R. Klivans and Alexander A. Sherstov. Cryptographic hardness for learning intersections of halfspaces. *J. Comput. Syst. Sci.*, 75(1):2–12, 2009. Conference version appeared in the proceedings of FOCS 2006.

24 Adam R. Klivans and Amir Shpilka. Learning restricted models of arithmetic circuits. *Theory of Computing*, 2(10):185–206, 2006. Conference version appeared in the proceedings of COLT 2003.

25 Adam R. Klivans and Daniel A. Spielman. Randomness efficient identity testing of multivariate polynomials. In *Proceedings on 33rd Annual ACM Symposium on Theory of Computing, July 6-8, 2001, Heraklion, Crete, Greece*, pages 216–223, 2001.

26 Tamara G Kolda and Brett W Bader. Tensor decompositions and applications. *SIAM review*, 51(3):455–500, 2009.

27 Sue E Leurgans, Robert T Ross, and Rebecca B Abel. A decomposition for three-way arrays. *SIAM Journal on Matrix Analysis and Applications*, 14(4):1064–1083, 1993.

28 Nutan Limaye, Srikanth Srinivasan, and Sébastien Tavenas. Superpolynomial lower bounds against low-depth algebraic circuits. In *FOCS 2021*, 2022.

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

30 K. Pratt. Waring rank, parameterized and exact algorithms. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 806–823, 2019. `doi:10.1109/FOCS.2019.00053`.

31 Yaroslav Shitov. How hard is the tensor rank? *arXiv*, abs/1611.01559, 2016. `arXiv:1611.01559`.

32 Gaurav Sinha. Reconstruction of real depth-3 circuits with top fan-in 2. In *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, pages 31:1–31:53, 2016.

A Adjoint algebra and vector space decomposition

In this section, we prove some basic facts about the adjoint algebra and vector space decomposition, for completeness. We will start by stating that we can compute black-box access to partial derivatives of f from black-box access to f.

▶ **Fact 29.** *Given black-box access to a (n, d) polynomial f and a monomial $\mathbf{x}^{\boldsymbol{\alpha}}$, a black-box access to $\partial_{\boldsymbol{\alpha}}^{=k} f$ can be computed in deterministic* $\mathrm{poly}(n, d^k)$ *time.*

This follows from the fact that black-box access to a first-order derivative of f can be computed in deterministic polynomial time from black-box access to f.

Next, we define the adjoint algebra.

▶ **Definition 30** (Adjoint algebra). *Consider a collection of linear maps $\mathcal{L}$ from vector space U to vector space V (over a field $\mathbb{F}$). The adjoint algebra for this collection of linear maps is defined as follows:*

$$\mathrm{Adj}(\mathcal{L}, U, V) = \{(D, E) \mid D : U \to U,\ E : V \to V \ \text{ are linear maps s.t. } LD = EL \ \text{ for all } L \in \mathcal{L}\}.$$

Next we define the notion of a vector space decomposition.

▶ **Definition 31** (Vector space decomposition)**.** *Consider a collection of linear maps $\mathcal{L}$ from vector space U to vector space V. We say that $U = U_1 \oplus \cdots \oplus U_s$ and $V = V_1 \oplus \cdots \oplus V_s$ is a vector space decomposition for the triple $(\mathcal{L}, U, V)$ if $\mathcal{L}(U_i) \subseteq V_i$ for all $i \in [s]$ (and at least one of U_i, V_i is a non-trivial subspace). We say that the decomposition is further indecomposable if the triples $(\mathcal{L}, U_i, V_i)$ are indecomposable for all i.*

The next definition is about when the adjoint algebra is trivial.

▶ **Definition 32** (Trivial Adjoint algebra)**.** *Consider a collection of linear maps $\mathcal{L}$ from vector space U to vector space V. Also consider a decomposition, $U = U_1 \oplus \cdots \oplus U_s$, $V = V_1 \oplus \cdots \oplus V_s$ that is further indecomposable. We say that the adjoint algebra is trivial if*

$$\mathrm{Adj}(\mathcal{L}, U, V) = \{(D, E) : \exists\, scalars\, \lambda_1, \ldots, \lambda_s \; s.t.\; D|_{U_i} = \lambda_i \mathbb{1}_{U_i}, E|_{V_i} = \lambda_i \mathbb{1}_{V_i} \; for\; all\, i \in [s]\}.$$

Next we define what we mean by uniqueness of decomposition.

▶ **Definition 33** (Uniqueness of decomposition)**.** *Consider a collection of linear maps $\mathcal{L}$ from vector space U to vector space V. Also consider a decomposition, $U = U_1 \oplus \cdots \oplus U_s$, $V = V_1 \oplus \cdots \oplus V_s$ that is further indecomposable. We say that the decomposition is unique if for any other further indecomposable decomposition, $U = U'_1 \oplus \cdots \oplus U'_{s'}$, $V = V'_1 \oplus \cdots \oplus V'_{s'}$, it turns out that $s = s'$ and there exists a permutation $\pi : [s] \to [s]$ s.t. $U'_i = U_{\pi(i)}$ and $V'_i = V_{\pi(i)}$ for all $i \in [s]$.*

The next lemma states the uniqueness of decomposition in the case when the adjoint algebra is trivial. The uniqueness of decomposition holds in a much more general setting by a reduction to the Krull-Schmidt theorem (see [7]) but here we only focus on a special case that is relevant to us.

▶ **Lemma 34.** *Consider a collection of linear maps $\mathcal{L}$ from vector space U to vector space V. Also consider a decomposition, $U = U_1 \oplus \cdots \oplus U_s$, $V = V_1 \oplus \cdots \oplus V_s$ that is further indecomposable. Suppose the adjoint algebra is trivial. Then the above is the unique decomposition that is further indecomposable.*

For proof of the above lemma see the full version.

Next we state an algorithm for vector space decomposition. While an algorithm in a much more general setting follows from known algorithms for module decomposition (see [7]), the algorithm we state here has the advantage that it is simpler and works for large enough fields (as opposed to algebraically closed fields). This algorithm is also present in [7] but not in a very explicit form, so we restate it here for completeness as well.

▶ **Lemma 35.** *Algorithm 2 with parameter ℓ computes the correct decomposition when the adjoint algebra is trivial, with probability at least $1 - \binom{s}{2}/\ell$.*

Proof. Since the adjoint algebra is trivial,

$$\mathrm{Adj}(\mathcal{L}, U, V) = \{(D, E) : \exists\; \lambda_1, \ldots, \lambda_s \;\text{ s.t. }\; D|_{U_i} = \lambda_i \mathbb{1}_{U_i}, E|_{V_i} = \lambda_i \mathbb{1}_{V_i} \;\text{ for all }\; i \in [s]\}$$

Let $(\lambda_1^{(j)}, \ldots, \lambda_s^{(j)})$ be the tuple corresponding to (D_j, E_j). Then

$$D'|_{U_i} = \left(\sum_{j=1}^{s} \mu_j \lambda_i^{(j)} \right) \mathbb{1}_{U_i}.$$

Algorithm 2 Vector space decomposition when adjoint algebra is trivial.

```
Input: Set of linear maps $\mathcal{L}$ between vector spaces $U$ and $V$ s.t. the triple $(\mathcal{L}, U, V)$ admits a further indecomposable decomposition $U = U_1 \oplus \cdots \oplus U_s$, $V = V_1 \oplus \cdots \oplus V_s$. Also the adjoint algebra is trivial.
Output: $s$ vector spaces $U'_1, \ldots, U'_s$ s.t. there exists a permutation $\pi : [s] \to [s]$ s.t. $U'_i = U_{\pi(i)}$.
Subroutine: Diagonalizing a diagonalizable linear map $D : U \to U$.
Parameters: Randomness parameter $\ell$.

Compute a basis $(D_1, E_1), \ldots, (D_s, E_s)$ of the adjoint algebra $\mathrm{Adj}(\mathcal{L}, U, V)$ (this is a system of linear equations). (If dimension is not $s$, then abort).
Pick $\mu_1, \ldots, \mu_s$ uniformly at random from a set of size $\ell$. Set $D' = \mu_1 D_1 + \cdots \mu_s D_s$.
Compute the eigenvalues of $D'$. If it has $s$ distinct eigenvalues, call them $\lambda_1, \ldots, \lambda_s$. If not (or it is not diagonalizable), abort.
Set $U'_i$ to be the eigenspace of $D'$ corresponding to $\lambda_i$.
```

We know that the vectors $(\lambda_1^{(j)}, \ldots, \lambda_s^{(j)})$, for $j \in [s]$, are linearly independent. Hence the vectors $(\lambda_i^{(1)}, \ldots, \lambda_i^{(s)})$, for $i \in [s]$, are also linearly independent. Hence for $i \neq i'$, the linear polynomial (in the μ_j's) $\sum_{j=1}^{s} \mu_j (\lambda_i^{(j)} - \lambda_{i'}^{(j)})$ is non-zero and hence if the μ_j's are chosen at random from a set of size ℓ, then with probability at least $1 - 1/\ell$,

$$\sum_{j=1}^{s} \mu_j (\lambda_i^{(j)} - \lambda_{i'}^{(j)}) \neq 0.$$

By a union bound, with probability at least $1 - \binom{s}{2}/\ell$, for any $i \neq i'$,

$$\sum_{j=1}^{s} \mu_j (\lambda_i^{(j)} - \lambda_{i'}^{(j)}) \neq 0.$$

Thus there are s distinct eigenvalues of D', one each corresponding to the eigenspace U_i. This completes the proof. ◀

We next define the concept of isomorphism between tuples $(\mathcal{L}, U, V)$ and $(\mathcal{L}', U', V')$, and relate the adjoint algebras for isomorphic tuples.

▶ **Definition 36** (Isomorphic tuples). *We say that $(\mathcal{L}, U, V)$ and $(\mathcal{L}', U', V')$ are isomorphic if there is an invertible linear transformation $\phi : \langle \mathcal{L} \rangle \to \langle \mathcal{L}' \rangle$ and invertible linear maps $T : U \to U'$, $S : V \to V'$ s.t. $\phi(L)T = SL$ for all $L \in \mathcal{L}$.*

▶ **Proposition 37** (Adjoint algebras under isomorphism). *Let $(\mathcal{L}, U, V)$ and $(\mathcal{L}', U', V')$ be isomorphic tuples. Then $(D, E) \in \mathrm{Adj}(\mathcal{L}, U, V)$ iff $(TDT^{-1}, SES^{-1}) \in \mathrm{Adj}(\mathcal{L}', U', V')$.*

Proof. It suffices to prove one direction because of symmetry. Suppose $(D, E) \in \mathrm{Adj}(\mathcal{L}, U, V)$ i.e. $LD = EL$ for all $L \in \mathcal{L}$.Then

$$\phi(L)TDT^{-1} = SLDT^{-1} = SELT^{-1} = SES^{-1}\phi(L)$$

for all $L \in \mathcal{L}$. Since $\{\phi(L)\}_{L \in \mathcal{L}}$ span $\langle \mathcal{L}' \rangle$, we get that $L'TDT^{-1} = SES^{-1}L'$ for all $L' \in \mathcal{L}'$. That is $(TDT^{-1}, SES^{-1}) \in \mathrm{Adj}(\mathcal{L}', U', V')$. ◀

This yields the following corollary:

▶ **Corollary 38.** *Let* $(\mathcal{L}, U, V)$ *and* $(\mathcal{L}', U', V')$ *be isomorphic tuples. Then* $\mathrm{Adj}(\mathcal{L}, U, V)$ *is trivial iff* $\mathrm{Adj}(\mathcal{L}', U', V')$ *is trivial.*

Next we state an instantiation of the above corollary which we need for our analysis.

▶ **Corollary 39.** *Let* $g \in \mathbb{F}[\mathbf{z}]_d$*,* $\mathbf{z} = (z_1, \ldots, z_m)$*. Also* $h = g(\ell_1, \ldots, \ell_m)$*, where* ℓ_i'*s linearly independent linear forms in the* $\mathbf{z}$ *variables. Then* $\mathrm{Adj}\left(\partial_{\mathbf{z}}^{=1}, \langle \partial_{\mathbf{z}}^{=k} g\rangle, \langle \partial_{\mathbf{z}}^{=(k+1)} g\rangle\right)$ *is trivial iff* $\mathrm{Adj}\left(\partial_{\mathbf{z}}^{=1}, \langle \partial_{\mathbf{z}}^{=k} h\rangle, \langle \partial_{\mathbf{z}}^{=(k+1)} h\rangle\right)$ *is trivial.*

For proof of the above lemma see the full version.

B Linear algebra with black boxes

In Algorithm 1, we need to perform linear algebra given black boxes for polynomials. We give references for how to do this here. We will need the following lemma from [18].

▶ **Lemma 40** (Section A.1 in [18]). *Given black boxes for the polynomials* $f_1, \ldots, f_\ell \in \mathbb{F}[\mathbf{x}]_d$*, there is a randomized* $\mathrm{poly}(n, \ell, d)$ *time algorithm that computes a basis for the following vector space*

$$(f_1, \ldots, f_\ell)^\perp := \{(\alpha_1, \ldots, \alpha_\ell) : \sum_{i=1}^{\ell} \alpha_i f_i = 0\}.$$

In particular, we get the following corollary.

▶ **Corollary 41.** *Given black boxes for the polynomials* $f_1, \ldots, f_\ell \in \mathbb{F}[\mathbf{x}]_d$ *which are linearly independent and for a* $p \in \mathbb{F}[\mathbf{x}]_d$ *which linearly depends on* $f_1, \ldots, f_\ell$*, there is a randomized* $\mathrm{poly}(n, \ell, d)$ *time algorithm that computes* $\beta_1, \ldots, \beta_\ell$ *s.t.*

$$p = \sum_{i=1}^{\ell} \beta_i f_i.$$

Using Corollary 41, one can compute the matrices corresponding to the linear maps $\mathcal{L}$ in Algorithm 2 if one is given only black boxes for bases of U and V. One can also carry out the Step 3 in Algorithm 1 using Corollary 41.

C Reducing the field size

In this section, we provide a sketch of how to reduce the field size in Theorem 1. For this, we will have to change the non-degeneracy conditions slightly. We state the new non-degeneracy conditions next for the circuit $f = \sum_{i=1}^{s} g_i(\ell_{i1}, \ldots, \ell_{im})$.

1. For each $i \in [s]$, the linear forms $(\ell_{i1}, \ldots, \ell_{im})$ are linearly independent. Let us denote by $d_{i,k} := \dim\left(\partial_{\mathbf{z}}^{=k} g_i(\mathbf{z})\right)$. Consider the vector spaces $U := \langle \partial^{=k} f\rangle$, $V := \langle \partial^{=(k+1)} f\rangle$ (here the partials are w.r.t. the $\mathbf{x}$ variables). We impose $\dim(U) = \sum_{i=1}^{s} d_{i,k}$ and $\dim(V) = \sum_{i=1}^{s} d_{i,k+1}$.
2. We impose that $\mathrm{Adj}(\partial^{=1}, U, V)$ is trivial i.e. $\dim\left(\mathrm{Adj}(\partial^{=1}, U, V)\right) = s$.
3. This is the same as the Item 3 in Section 2.1.

Let us first compare these conditions with the conditions in Section 2.1. It can be verified that Item 1 is the same as $U = U_1 \oplus \cdots \oplus U_s$ and $V = V_1 \oplus \cdots \oplus V_s$ i.e. Item 2 in Section 2.1. Item 2 here is new and assuming this implies uniqueness of decomposition and this can be used directly in the proof of Theorem 5 (instead of Item 1 in Section 2.1).

We now sketch the argument on why random $\ell_{i,j}$'s would satisfy these conditions. In Sections 3.1 and 3.2, we provide a particular setting of $\ell_{i,j}$'s s.t. Items 1 and 2 in Section 2.1) are satisfied. These imply that Items 1 and 2 stated here are satisfied (for Item 2, one would need to combine the proof of Theorem 5 and Item 3). So we just need the Schwartz-Zippel argument. First consider Item 1. The condition about U, for example, is about the rank of a matrix whose dimensions are $\binom{n+k-1}{k} \times \binom{n+d-k-1}{d-k}$ and entries are homogeneous polynomials of degree k in the coefficients of $\ell_{i,j}$'s. We know that the rank is always atmost $D := \sum_{i=1}^{s} d_{i,k}$ and also that the rank is equal to D for a particular setting of $\ell_{i,j}$'s. This implies the existence of a $D \times D$ minor which has full rank for a particular setting of $\ell_{i,j}$'s. Hence by Schwartz-Zippel lemma, we get that this minor is full rank for a random choice of $\ell_{i,j}$'s if the field size is atleast $\text{poly}(D, k) = \text{poly}(s\binom{m+k-1}{k}, k)$ which is $\text{poly}(n, d, s)$ since we choose $\Theta(\log(s)/\log(n))$.

Regarding the condition on the adjoint, note that adjoint is the solution to a linear system of equations. Hence $\dim\left(\text{Adj}(\partial^{=1}, U, V)\right) = s$ is equivalent to the corank of a matrix being atmost s (it is atleast s by definition). The dimensions of the matrix are $(\dim(U)^2 + \dim(V)^2) \times (n \cdot \dim(U) \cdot \dim(V))$ and the entries are homogeneous polynomials of degree $O(k)$ in the coefficients of $\ell_{i,j}$'s. Again here the Schwartz-Zippel argument can be carried out over a field of size $\text{poly}(\dim(U), \dim(V), n, k)$ which is $\text{poly}(n, d, s)$ because of the choice of k.