Electronic Colloquium on Computational Complexity, Report No. 29 (2026)

Polynomial Identity Testing and Reconstruction for Depth-4 Powering Circuits of High Degree

Amir Shpilka[*] Yann Tal[*]

Abstract

We study deterministic polynomial identity testing (PIT) and reconstruction algorithms for depth-4 arithmetic circuits of the form

$$\Sigma^{[r]}\wedge^{[d]}\Sigma^{[s]}\Pi^{[\delta]}.$$

This model generalizes Waring decompositions and diagonal circuits, and captures sums of powers of low-degree sparse polynomials. Specifically, each circuit computes a sum of r terms, where each term is a d-th power of an s-sparse polynomial of degree δ. This model also includes algebraic representations that arise in tensor decomposition and moment-based learning tasks such as mixture models and subspace learning.

We give deterministic worst-case algorithms for PIT and reconstruction in this model. Our PIT construction applies when $d > r^2$ and yields explicit hitting sets of size $O(r^4 s^4 n^2 d\delta^3)$. The reconstruction algorithm runs in time $\mathrm{poly}(n, s, d)$ under the condition $d = \Omega(r^4\delta)$, and in particular it tolerates polynomially large top fan-in r and bottom degree δ.

Both results hold over fields of characteristic zero and over fields of sufficiently large characteristic. These algorithms provide the first polynomial-time deterministic solutions for depth-4 powering circuits with unbounded top fan-in. In particular, the reconstruction result improves upon previous work which required non-degeneracy or average-case assumptions [GKS20, BHKX22, CGK+24].

The PIT construction relies on the *ABC theorem for function fields* (Mason-Stothers theorem), which ensures linear independence of high-degree powers of sparse polynomials after a suitable projection. The reconstruction algorithm combines this with *Wronskian-based differential operators*, structural properties of their kernels, and a robust version of the Klivans-Spielman hitting set.

[*] School of Computer Science and AI, Tel Aviv University. This research was funded by the European Union (ERC, EACTP, 101142020). Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible for them.

ISSN 1433-8092

Contents

1 Introduction

Arithmetic circuits are directed acyclic graphs that compute multivariate polynomials using addition and multiplication gates. They form the standard model for studying algebraic computation, serving as the analogue of Boolean circuits in the arithmetic world.

The *Polynomial Identity Testing* (PIT) problem asks, given an arithmetic circuit that computes a multivariate polynomial $f(\mathbf{x})$, to decide whether $f \equiv 0$. Randomized algorithms for PIT have been known for decades, via the polynomial identity lemma,[1] yet obtaining a deterministic polynomial-time algorithm remains a fundamental open problem in derandomization. Hardness-randomness tradeoff results have shown a tight connection between PIT and lower bounds for arithmetic circuits [HS80, KI04, DSY09, CKS19, GKSS22, KST23, KS19]. PIT has received a great deal of attention in recent years. Deterministic algorithms are now known for several restricted models, such as bounded-depth circuits, bounded-read formulas, and read-once algebraic branching programs. Variants have also been studied for orbits of these classes under affine transformations. For surveys on PIT, see [SY10, Sax09, Sax14, DG24].

Closely related to PIT is the *reconstruction problem*. Instead of merely determining whether f is identically zero, reconstruction seeks to recover, using only black-box access to f, a circuit from a prescribed class that computes it. A deterministic reconstruction algorithm immediately implies deterministic PIT for that class, while a black-box PIT algorithm guarantees an *information-theoretic* form of reconstruction. If H is a hitting set for circuits of size $2s$, then the evaluation map

$$\mathrm{Eval}_H : C \longmapsto (C(a))_{a \in H}$$

is injective for all circuits C of size at most s. The remaining challenge is to perform this inversion efficiently. This step can be significantly more difficult: there are natural circuit classes for which deterministic PIT is known, yet no efficient reconstruction algorithm is available. There have been many works studying the reconstruction problem in different models [BBB$^+$00, KS06, SV14, FS12, GKL12, GKQ14], with a recent flurry of algorithms for small-depth circuits [Shp09, KS09a, Sin16, BSV21, Sin22, PSV24, BGKS22, BS25, BSV25, SSV25].

A series of depth-reduction results [AV08, Koi12, Tav15, GKKS16] proved that any polynomial computed by a small circuit can also be represented by a depth-4 circuit of subexponential size (and, over characteristic zero, by a depth-3 circuit). Thus, providing PIT or reconstruction algorithms for shallow arithmetic circuits is an outstanding task that can have implications for general models of arithmetic circuits.

The search for efficient reconstruction is bounded by fundamental hardness results. Starting with the work of Håstad [Hås90], it was shown that even computing tensor rank, i.e., determining the smallest top fan-in of a set-multilinear depth-3 circuit is NP-hard over $\mathbb{Q}$ and NP-complete over finite fields, even for degree-3 tensors. Extensions of this result to approximation algorithms and to other models of computation were given in subsequent papers [SŠ18, FK09, KS09c, Swe18].

[1] Often referred to as the Schwartz–Zippel–DeMillo–Lipton–Ore lemma [Ore22, DL78, Zip79, Sch80].

Because the tensor-rank problem is equivalent to minimizing the top fan-in of a set-multilinear $\Sigma\Pi\Sigma$ circuit, these hardness results extend directly to reconstruction: proper learning for such circuits is NP-hard and, in some instances, even undecidable [CGS23]. This motivated the search for efficient reconstruction algorithms for certain parameter regimes or under additional restrictions on the circuit.

Reconstruction also connects naturally to *learning theory*. An arithmetic circuit can be viewed as an algebraic hypothesis, and black-box evaluations of f correspond to examples (or membership queries). Thus, reconstruction corresponds to *exact learning* in the noiseless setting, i.e., when we obtain the exact value of f on each and every query. This stands in contrast to the *approximate* or *noisy* learning problems that are typical in machine learning. This analogy has recently become concrete through work showing that some important learning tasks, such as learning mixtures of Gaussians or subspace clustering, reduce to learning polynomials of the form

$$f(\mathbf{x}) = \alpha_1 f_1(\mathbf{x})^d + \cdots + \alpha_r f_r(\mathbf{x})^d, \tag{1}$$

where the f_i are polynomials of degree δ [GHK15, GKS20]. A representation as in (1) is also called a $\Sigma^{[r]}\wedge^{[d]}\Sigma\Pi^{[\delta]}$ circuit. When $\delta = 1$, the reconstruction problem is equivalent to symmetric tensor decomposition (Waring rank decomposition), which is fundamental in many applications in machine learning, e.g., in moment-based methods for latent variable models, cf. [GVX14, AGH$^+$14, MR14, HSS15, MSS16, HSSS16, KX25]. When $\delta = 2$, the problem corresponds to learning mixtures of Gaussians and has been extensively studied, cf. [SK01, DS07a, ABG$^+$14, RV17, LM21, dD23].

In recent years, an increasing number of works consider the case $\delta > 2$ [GKS20, CGK$^+$24, BHKX22, BESV24]. Garg, Kayal, and Saha [GKS20], and subsequent work by Bafna, Hsieh, Kothari, and Xu [BHKX22], gave randomized reconstruction algorithms for such polynomials in the noiseless case, running in time $\mathrm{poly}(n, d, r)^\delta$. Their algorithm works under a non-degeneracy assumption that holds with high probability for random polynomials but does not apply in the worst case. Chandra, Garg, Kayal, Mittal, and Sinha [CGK$^+$24] extended this connection to noisy and smoothed settings, showing, quite surprisingly, that lower-bound techniques for arithmetic circuits can be transformed into robust learning algorithms for this model. The running time of their algorithm is $\mathrm{poly}(n, d, r)^\delta$ as well.

In the next section, we formally state our results, followed by a detailed comparison with prior work.

1.1 Our Results

We consider the model of sums of powers of low degree sparse polynomials, denoted by

$$\Sigma^{[r]}\wedge^{[d]}\Sigma^{[s]}\Pi^{[\delta]}.$$

Here, each bottom product gate has degree at most δ. Above it, each addition gate computes an s-sparse polynomial. Then we take the d-th power of each sparse polynomial and sum these terms. This model naturally includes the classical $\Sigma\wedge\Sigma$ (Waring) model as a special case. We give efficient deterministic hitting sets as well as a reconstruction algorithm for this class in the *worst case*, without any genericity or average-case assumptions. Our results hold both for characteristic zero fields and for fields with polynomially large characteristic.

Our first result is a polynomial-size hitting set for $\Sigma^{[r]}\wedge^{[d]}\Sigma^{[s]}\Pi^{[\delta]}$ circuits, where the powering degree is quadratic in the *rank*, namely, the top fan-in r. The following is a simplified version of Theorem 3.2.

Theorem 1.1. *Let $n, d, r, s, \delta \in \mathbb{N}$ such that $d = \Omega(r^2)$. Let $\mathbb{F}$ be a field of characteristic $p = 0$ or $p \geqslant rd\delta(s^2 n + \delta)$. There is an explicit hitting set of size $\mathrm{poly}(s, n, d)$ for the class of $\Sigma^{[r]}\wedge^{[d]}\Sigma^{[s]}\Pi^{[\delta]}$ circuits defined over $\mathbb{F}$.*[2]

To our knowledge, this is the first polynomial-size hitting set for any model of depth-4 circuits with unbounded top fan-in.

Theorem 1.1 is instrumental in obtaining a deterministic reconstruction algorithm.

Theorem 1.2. *Let $n, d, r, s, \delta \in \mathbb{N}$ such that $d \geqslant (r+1)^4\delta$. Let $\mathbb{F}$ be a field of characteristic $p = 0$ or $p \geqslant rd\delta(s^2 n + \delta)$. There exists a deterministic algorithm that, given black box access to a polynomial f computed by a $\Sigma^{[r]}\wedge^{[d]}\Sigma^{[s]}\Pi^{[\delta]}$ circuit, defined over $\mathbb{F}$, with bit complexity B, reconstructs f.*

1. *When $p = 0$, the running time is $\mathrm{poly}(n, s, d, B)$.*
2. *When $p > rd\delta(s^2 n + \delta)$ and $|\mathbb{F}| = q$, the running time is $\mathrm{poly}(n, s, d, p, \log q)$.*

This is the first reconstruction algorithm for any model of depth-4 powering circuits with unrestricted top and bottom fan-in.

1.2 Prior Work and Related Models

In what follows, $\Sigma^{[r]}$ represents a layer of addition gates of fan-in at most r, $\wedge^{[d]}$ represents raising to power d, and $\Pi^{[\delta]}$ is a product gate of fan-in at most δ. Thus, $\Sigma^{[r]}\Pi^{[\delta]}\Sigma$ is the class of depth-3 circuits that consist of the sum of r terms, each a product of at most δ linear functions. A circuit is *multilinear* if each monomial in every gate is multilinear. A circuit is *set-multilinear* if the input is composed of disjoint sets $x = x_1 \sqcup x_2 \sqcup \cdots \sqcup x_\delta$, and each gate computes a polynomial in which each monomial contains at most one variable from each set.

1.2.1 Polynomial Identity Testing for Small-Depth Circuits

The breakthrough of Limaye, Srinivasan, and Tavenas [LST25], followed by Andrews and Forbes [AF22], established the first subexponential-size hitting sets for general constant-depth circuits. Sharper bounds are known for several restricted subclasses.

For *sparse polynomials*, that is, polynomials computable by small $\Sigma\Pi$ circuits, deterministic hitting sets were obtained in the classical work of Ben-Or and Tiwari [BT88], Grigoriev, Karpinski and Singer [GKS90], and Klivans and Spielman [KS01].

[2] Here and later, if $|\mathbb{F}|$ is not large enough then the evaluation points come from an extension field.

Building on the white-box PIT algorithm of Raz and Shpilka [RS05], Saxena [Sax08] gave a deterministic polynomial-time PIT algorithm for $\Sigma^{[r]}\wedge^{[d]}\Sigma$ circuits. In the black-box model, [FSS14, GKS16, GG20] constructed hitting sets of size $r^{\log\log r}\,\mathrm{poly}(d, n)$, which are superpolynomial in the width, but do not assume any relation between d and r.

Rank-based methods yield polynomial-size hitting sets for $\Sigma^{[r]}\Pi^{[d]}\Sigma$ circuits when the top fan-in r is constant [DS07b, KS11, KS07, KS09b, SS13, SS12]. Agrawal et al. [ASSS16] introduced the Jacobian hitting technique, obtaining polynomial-size hitting sets when the transcendence degree of the product terms is bounded. Their result also extends to certain classes of higher depths.

Using rank-based ideas, Peleg and Shpilka, and Garg, Oliveira, and Sengupta [PS21, GdOS25] gave polynomial-size hitting sets for $\Sigma^{[3]}\Pi\Sigma\Pi^{[O(1)]}$ circuits. Independently of [GdOS25], Guo and Wang [GW25] gave a hitting set for $\Sigma^{[3]}\Pi\Sigma\Pi^{[O(1)]}$ circuits, under the additional restriction that in one of the terms the multiplicity of each irreducible factor is 1. Their approach is novel and based on algebro-geometric techniques. However, the additional assumption simplifies the technical challenges encountered in earlier works. Dutta, Dwivedi, and Saxena [DDS21] obtained quasipolynomial-size hitting sets for circuits whose top and bottom fan-ins are bounded by $\mathrm{poly}(\log n)$, using the newly developed DiDIL technique. Quasipolynomial and later polynomial-size hitting sets were also constructed for multilinear circuits with bounded top fan-in [KMSV13, SV18]. As in the depth-3 setting, polynomial-size constructions are known only when the top fan-in is bounded.

For sums of powers, i.e., $\Sigma^{[r]}\wedge^{[d]}\Sigma\Pi^{[\delta]}$ circuits, Forbes [For15] constructed hitting sets of size $(nd)^{O(\delta\log r)}$. Dutta et al. [DDS21] observed that existing techniques give hitting sets of size $s^{O(\log\log s)}$ for size-s $\Sigma\wedge\Sigma\wedge$ circuits. These results do not restrict the top fan-in r, but the constructions remain superpolynomial in size.

To summarize, polynomial-size hitting sets are known for depth-2 circuits and for depth-3 and depth-4 models with bounded top fan-in. For depth-4 powering circuits, all known constructions are quasipolynomial (or slightly better but still superpolynomial) and hold without any restriction on the top fan-in.

Theorem 1.1 gives the first polynomial-size construction, requiring only $d = \Omega(r^2)$.

1.2.2 Reconstruction

Shpilka [Shp09] and Karnin and Shpilka [KS09a] gave the first reconstruction algorithms for $\Sigma^{[r]}\Pi\Sigma$ circuits, over small finite fields for $r = O(1)$. Sinha [Sin16, Sin22] obtained randomized polynomial-time reconstruction algorithms for $\Sigma^{[2]}\Pi\Sigma$ circuits over $\mathbb{R}$ and $\mathbb{C}$, relying on Sylvester–Gallai type theorems. Saraf and Shringi [SS25] extended this to $r = 3$. More recently, Saraf, Shringi, and Varadarajan [SSV25] solved the general $r = O(1)$ case, obtaining a quasipolynomial-time algorithm.

For $\Sigma\wedge\Sigma$ circuits, the first reconstruction algorithm dates back to Sylvester, who gave an algorithm for binary forms provided $r \leqslant \lfloor (d+2)/2 \rfloor$ in the generic[3] case [Syl51]. Kayal [Kay12] gave a randomized reconstruction algorithm for the generic case when r can be as large as $\binom{n+d/2-1}{d/2}$. When $d > 2r$, Kayal's algorithm can be derandomized.

[3] Being generic means belonging to a Zariski open subset.

Bhargava, Saraf, and Volkovich [BSV21] gave a deterministic polynomial-time algorithm for multilinear and set-multilinear $\Sigma^{[r]}\Pi\Sigma$ circuits with top fan-in $r = O(1)$. In particular, this yields exact tensor decomposition for constant rank (with r appearing in the exponent of the running time). Peleg, Shpilka, and Volk [PSV24] extended this to superconstant r, providing a randomized fixed-parameter algorithm with respect to r (with poor dependence on the top fan-in). For set-multilinear depth-3 circuits, Bhargava and Shringi [BS25] further obtained a deterministic fixed-parameter algorithm running in time $2^{r^{O(1)}}\operatorname{poly}(n, d)$. While these works resolve the question for fixed-parameter settings, their dependence on r remains superexponential or tower-type.

Garg, Kayal, and Saha [GKS20] studied circuits of the form $\Sigma^{[r]}\wedge^{[d]}\Sigma^{[s]}\Pi^{[\delta]}$. They obtained randomized algorithms under non-degeneracy assumptions in the noiseless case, with running time $\operatorname{poly}(n, r, d)^{\delta}$. In addition to the requirement that the input is generic, they assume $n > d^2$, $\delta \leqslant O(\sqrt{(\log d)/(\log\log d)})$, and that the field has size at least $(nr)^{\Omega(\delta)}$. On the other hand, they allow r to be as large as $n^{O(d/\delta^2)}$. An important aspect of their work is the method they introduced for translating certain algebraic lower-bound techniques to learning algorithms. Bafna et al. [BHKX22] extended the result of [GKS20] to handle polynomially large noise and to allow, roughly, $r = O(n^{2d/15})$. The subsequent work of Chandra et al. [CGK+24] generalized the method of [GKS20] and enabled transferring lower-bound techniques also to the noisy case, assuming natural conjectures on the largeness of singular values of certain random matrices. All these algorithms are randomized.

Theorem 1.2 provides a deterministic polynomial-time reconstruction algorithm in the worst-case noiseless setting, without assuming any non-degeneracy conditions.

Our requirement $d = \Omega(r^4\delta)$ is similar in spirit to the requirement that $r \leqslant n^{O(\delta)}$ from [GKS20, BHKX22, CGK+24], since both conditions guarantee uniqueness of representation: ours for the worst case, and theirs for the generic case.

Additionally, Theorem 1.2 is the first polynomial-time reconstruction algorithm for any model of depth-4 circuits with unbounded top fan-in.

1.3 Proof Overview

Our proof of Theorem 1.1 is based on the ABC theorem for function fields, proved independently by Mason [Mas84] and Stothers [Sto81]. We shall use an extension proved by Vaserstein and Wheland [VW03] (Theorem 2.1). Roughly speaking, the theorem shows that high-degree, pairwise independent univariate polynomials with relatively few distinct zeros are linearly independent. Therefore, if we start with n-variate polynomials that are high powers of sparse polynomials and project them to univariate polynomials in a way that preserves pairwise independence, the theorem guarantees that their sum cannot vanish identically. A simple interpolation then suffices to verify that the polynomial is nonzero.

The reconstruction algorithm (Theorem 1.2) is more involved and requires several additional ideas. At a high level, we first reduce the problem to reconstructing in the univariate case and then lift the solution back to the n-variate setting. We next explain the ideas used in each of the settings.

Reconstruction in the univariate case. Consider a representation $f = \sum_{i=1}^{r} \alpha_i f_i(x)^d$ with monic f_i. Our approach is to find a differential operator of order r,

$$L = \sum_{i=0}^{r} Q_i(x)\nabla^i, \qquad \text{where} \quad \nabla = \frac{d}{dx},$$

such that $L(f) = 0$. The existence of L can be proved by considering the Wronskian $W(f, f_1^d, \ldots, f_r^d)$, where $W(g_1, \ldots, g_k)$ denotes the determinant of the $k \times k$ matrix whose (i, j) entry is $\nabla^{i-1} g_j$. It is well known that the Wronskian of polynomials vanishes if and only if they are linearly dependent, and this also holds for sufficiently large characteristic (see Theorem 2.11). Expanding $W(f, f_1^d, \ldots, f_r^d)$ along the first column yields (after clearing the gcd) the desired operator L and also provides an upper bound on the degrees of the Q_i.

The next crucial observation is that $\ker(L)$ is exactly the space spanned by $\{f_1^d, \ldots, f_r^d\}$. This again follows from the ABC theorem, in a slightly more general form (Corollary 2.2), which we also use to prove the uniqueness of L (Claim 4.3).

Given this, we can compute the unique L by solving a system of linear equations.

Another important observation is that each zero of each f_i is also a zero of the leading coefficient Q_r of L. In the theory of ODEs this is usually stated for poles of analytic functions, but for polynomials one can work with roots instead of poles, and the same argument applies in large characteristic (Claim 4.7).

We have thus reduced the problem to the following: we are given an r-dimensional space of polynomials (the kernel of L) that contains r d-th powers of polynomials, whose zeros are known (but not their multiplicities), and we need to recover the underlying polynomials. A brute-force search over exponent vectors would require testing all $\binom{m+\delta}{\delta}$ possibilities, where $m = \deg(Q_r)$, but we will show a much more efficient algorithm that is polynomial in all parameters.

The starting point is to prove, by analyzing how the Wronskian factorizes, that any high-order root of the Wronskian must also be a root of one of the f_i (in practice we work with irreducible factors rather than with individual roots). This enables us to prune the space $\operatorname{span}\{f_1^d, \ldots, f_r^d\}$ and retain only the subspace of polynomials that vanish, with high multiplicity, at some root of Q_r. We can think of this as building a depth-δ tree in which each node has m children labeled by the roots of Q_r. Every path in the tree corresponds to a sequence of roots with associated multiplicities. For each node we construct the space of polynomials that vanish at these roots with given the multiplicities. At the leaves of the tree sit the f_i^d. The tree is still large, so we cannot check all its leaves. The idea is to perform a depth-first search and prune any branch that can be certified not to yield a new polynomial.

After the DFS algorithm finds the set $\{f_i^d\}$, we compute the coefficients α_i by solving a corresponding linear system.

Multivariate reconstruction. At a high level, we would like to project the blackbox multivariate polynomial to a univariate polynomial, apply the univariate reconstruction algorithm, and then, given the resulting univariate polynomials, recover the original polynomials.

To achieve this, we first observe that the projection must preserve non-associateness.[4] This can be ensured relatively easily using known tools, such as the Klivans-Spielman generator [KS01]. However, we cannot allow the degree of the projected polynomial to increase. This forces us to use projections of the form $\mathbf{x} = \mathbf{u} + t \cdot (\mathbf{u} - \mathbf{v})$, where $\mathbf{u}, \mathbf{v}$ will come from appropriate sets.

We are then left with the task of recovering the original polynomials from their projected versions. Each such projection corresponds to restricting a multivariate polynomial to a line. Consequently, reconstructing the polynomial requires its values on sufficiently many different lines, and hence we must consider many distinct projections.

At this point, another difficulty arises. Each univariate reconstruction produces a list of r polynomials, but these lists are not ordered. In particular, they are not aligned across different projections. Therefore, we must first align the lists in order to obtain black-box access to the projections of a single polynomial. This is accomplished by showing the existence of a suitable 'anchor' point $\mathbf{u}$ that enables the alignment of all projections of the form $\mathbf{u} + t \cdot (\mathbf{u} - \mathbf{v}) \to \mathbf{x}$, where $\mathbf{v}$ ranges over a sufficiently large (and structured) set. We show that such a good 'anchor' point exists in any hitting set for $\Sigma^{[2]}\wedge^{[d]}\Sigma^{[2s]}\Pi^{[\delta]}$ circuits.

Finally, we show that once alignment is achieved, the available information, obtained from the different $\mathbf{v}$, suffices to reconstruct the original polynomials. This process is reminiscent of erasure correction in error-correcting codes, except that our target objects (i.e., sparse polynomials) do not form a vector space. Nevertheless, the reconstruction can be carried out using a robust version of the Klivans–Spielman hitting set [KS01].

1.4 Organization

In Section 2 we introduce the basic mathematical tools. These include the ABC theorem (Section 2.1), the robust Klivans-Spielman generator (Section 2.2), the Wronskian and its properties (Section 2.3), and results on deterministic factorization of univariate polynomials (Section 2.4).

We present our hitting sets in Section 3. In Section 4 we give our univariate reconstruction algorithm and in Section 5 we give the algorithm for the multivariate case.

2 Preliminaries

For an integer $r \in \mathbb{N}$ we denote $[r] = \{1, \dots, r\}$ and $[[r]] = \{0, 1, \dots, r\}$.

We say that two nonzero polynomials $f, g \in \mathbb{F}[\mathbf{x}]$ are associate if there exists a nonzero scalar $\alpha \in \mathbb{F}$ such that $g(\mathbf{x}) = \alpha f(\mathbf{x})$. We say that a set of polynomials is non-associate if no two polynomials in the set are associate. This is equivalent to saying that the polynomials in the set are pairwise linearly independent.

2.1 The ABC theorem for function fields

An important tool in all proofs is the ABC theorem for function fields, known as the Mason-Stothers theorem [Mas84, Sto81].

[4] Polynomials are associate if and only if they are linearly dependent.

We state the following strengthening due to Vaserstein and Wheland [VW03]. For a polynomial $h \in \mathbb{F}[x]$, let $\nu(h)$ denote the number of *distinct* roots of h over the algebraic closure $\overline{\mathbb{F}}$.

While most of the results in [VW03] are stated for fields of characteristic zero, an inspection of the proofs in [VW03, Section 3] reveals that they also hold for fields of large enough characteristic.[5]

Theorem 2.1 ([VW03], Theorem 2.2(a)). *Let $D, r \in \mathbb{N}$ such that $r \geqslant 2$. Let $\mathbb{F}$ be a field of characteristic p such that $p = 0$ or $p > rD$. Let $h_1, \dots, h_r \in \mathbb{F}[x]$ satisfy*

$$h_1 + \cdots + h_r = h_0, \qquad \gcd(h_1, \dots, h_r) = 1,$$

where $\max_i \deg h_i \leqslant D$, not all h_i are constant and no nonempty subsum of $h_1, \dots, h_r$ vanishes. Then

$$\deg(h_0) < (r-1)\left(\sum_{i=0}^{r} \nu(h_i)\right).$$

In other words, Theorem 2.1 asserts that if h_0 has "high" degree while each h_i has "few" distinct roots, then h_0 cannot be expressed as a linear combination of the other h_i's. We will use the following corollary.

We note that we require an upper bound on $\deg(h_i)$ only in the case of positive characteristic.

Corollary 2.2. *Let $r, \eta, \delta, e \in \mathbb{N}$ such that $r \geqslant 2$. Let $\mathbb{F}$ be a field of characteristic p such that $p = 0$ or $p > r(\eta + e\delta)$. Let $P_0(x)^e, P_1(x)^e, \dots, P_r(x)^e \in \mathbb{F}[x]$ be pairwise linearly independent polynomials, such that for all i, $\deg(P_i) \leqslant \delta$. Let $g_0(x), \dots, g_r(x) \in \mathbb{F}[x]$ be additional polynomials, such that for all i, $\deg(g_i) \leqslant \eta$. Assume that*

$$e \;\geqslant\; (r^2 + r)\,(\eta + 1).$$

Then the identity

$$\sum_{i=0}^{r} g_i(x)\, P_i(x)^e = 0$$

holds if and only if each $g_i(x)$ is identically zero.

Proof. The "if" direction is immediate. For the converse, suppose

$$\sum_{i=0}^{r} g_i P_i^e = 0$$

is a nontrivial identity. Passing, if necessary, to a nontrivial relation of minimal support, we may assume that every $h_i := g_i P_i^e$ in the relation is nonzero, and hence each corresponding g_i is nonzero. Set

$$H(x) = \gcd(h_0, \dots, h_r), \qquad P(x) = \gcd(P_0, \dots, P_r).$$

[5] Vaserstein and Wheland were primarily interested in characteristic 0 or fixed characteristic independent of the degrees and number of polynomials (see [VW03, Section 5]).

Claim 2.3. *There exists* $G(x) \in \mathbb{F}[x]$ *such that* $H(x) = P(x)^e G(x)$ *and* $\deg G \leqslant (r+1)\eta$.

Proof. Clearly P^e divides each h_i, so P^e divides H. As $\gcd((P_0/P)^e, \dots, (P_r/P)^e) = 1$, we get that

$$\gcd\left(g_0(P_0/P)^e, \dots, g_r(P_r/P)^e\right) \quad \text{divides} \quad \Pi_{i=0}^r g_i.$$

Thus $G := H/P^e$ divides $\Pi_i g_i$, so

$$\deg(G) = \deg(H/P^e) \;\leqslant\; \sum_i \deg g_i \;\leqslant\; (r+1)\eta. \qquad \square$$

Let $S := \{i : \; g_i \not\equiv 0\}$ be the support, and choose $t \in S$ such that

$$\nu_{\sup} := \max_{i \in S} \nu(P_i/P) = \nu(P_t/P).$$

Since the polynomials are non-associate, $\nu_{\sup} \geqslant 1$. Define $\tilde{h}_i := h_i/H$. Then the equality $\sum_{i=0}^r h_i = 0$ holds if and only if $\sum_{i=0}^r \tilde{h}_i = 0$, and moreover

$$\gcd(\tilde{h}_0, \dots, \tilde{h}_r) = 1.$$

By Theorem 2.1, if $\sum_{i=0}^r \tilde{h}_i = 0$ and not all $\tilde{h}_i$ vanish, then

$$\begin{aligned} e\,\nu_{\sup} - \deg(G) &= e\nu(P_t/P) - \deg(G) \\ &\leqslant \deg((P_t/P)^e) - \deg(G) \\ &\leqslant \deg(\tilde{h}_t) \\ &< (r-1)\sum_{j=0}^r \nu(\tilde{h}_j) \\ &\leqslant (r-1)\sum_{j=0}^r \nu(g_j(P_j/P)^e) \\ &\leqslant (r-1)(r+1)\big(\eta + \nu_{\sup}\big). \end{aligned}$$

Hence

$$e < (r^2 + r)(\eta + 1),$$

contradicting the assumption on e. $\square$

We did not attempt to optimize the parameters in Corollary 2.2. In particular, for the special case where all g_i are scalars a tighter result is known.

Theorem 2.4 ([VW03, Corollary 3.8])**.** *Let* $r, d, \delta \in \mathbb{N}$ *such that* $r \geqslant 2$. *Let* $\mathbb{F}$ *be of characteristic* $p = 0$ *or* $p > rd\delta$. *Let* $f_0(x), \dots, f_r(x) \in \mathbb{F}[x]$ *be non-associate polynomials of degree at most* δ, *not all the* f_i *are constant. If* $(r-1)^2 \leqslant d+1$ *then* $\sum_{i=0}^r f_i^d(x) \neq 0$.

2.2 The Klivans–Spielman generator

We recall the generator construction of Klivans and Spielman [KS01], which converts a sparse multivariate polynomial into a univariate polynomial while preserving its structure.

For a prime number q, an integer $k < q$ let $\Psi_{k,q} : \mathbb{F}[\mathbf{x}] \mapsto \mathbb{F}[y]$ be defined as

$$\Psi_{k,q}(x_i) = y^{(k^{i-1} \bmod q)}, \tag{2}$$

with the natural extension to monomials and polynomials. Namely, for an exponent vector $\boldsymbol{e} = (e_1, \ldots, e_n)$

$$\Psi_{k,q}(\boldsymbol{x}^{\boldsymbol{e}}) = y^{\sum_i e_i\left(k^{(i-1)} \bmod q\right)}.$$

Clearly,

$$\deg(\Psi_{k,q}(\boldsymbol{x}^{\boldsymbol{e}})) \leqslant (q-1)\sum_i e_i = (q-1)\deg(\boldsymbol{x}^{\boldsymbol{e}})\ . \tag{3}$$

Theorem 2.5 (Corollary of Klivans–Spielman [KS01, Lemma 1])**.** *Let $f \in \mathbb{F}[\mathbf{x}]$ be a nonzero n-variate polynomial, such that f has at most s monomials and $\deg(f) \leqslant \delta$. Let q be a prime satisfying $q \geqslant \max(\delta, \binom{s}{2}n) + 1$. Then, for all but at most $\binom{s}{2}(n-1)$ values of $k \in [q-1]$, the univariate polynomial $\Psi_{k,q}(f)$ has the same number of monomials as f. Furthermore, $\deg(\Psi_{k,q}(f)) \leqslant \delta q$. Moreover, for all but at most $(s-1)(n-1)$ values of $k \in [q-1]$, the univariate polynomial $\Psi_{k,q}(f)$ is nonzero.*

Corollary 2.6. *Let $f, g \in \mathbb{F}[\mathbf{x}]$ be non-associate polynomials of individual degrees at most δ, each with at most s monomials. Let q be a prime larger than $\max(\delta, s^2 n) + 1$. Then, for all but at most $s(s-1)(n-1)$ values of $k \in [q-1]$, the univariate specializations $\Psi_{k,q}(f), \Psi_{k,q}(g)$ are linearly independent over $\mathbb{F}$.*

Proof. By Theorem 2.5, for all but $s(s-1)(n-1)$ values of $k \in [q-1]$, the specializations preserve the supports of f and g. If f and g have different supports, the property is immediate. Otherwise, they share the same set of monomials but have linearly independent coefficient vectors, and this linear independence is preserved under the substitution. □

We shall use the Klivans-Spielman generator in order to create a robust interpolation set for sparse polynomials. That is, a set of points such that knowing the value of an s-sparse polynomial on $1-\varepsilon$ fraction of the points in the set, allows efficient recovery of the polynomial.

The idea is quite simple given Theorem 2.5. Assume first the case of a characteristic zero field. Consider the modified generator

$$\Psi_{j,k,q,\lambda}(x_i) = \begin{cases} \lambda \cdot y^{(k^{j-1} \bmod q)} & i = j, \\ y^{(k^{i-1} \bmod q)} & i \neq j. \end{cases},$$

Observe that each monomial $\boldsymbol{x}^{\boldsymbol{e}}$ is mapped to

$$\Psi_{j,k,q,\lambda}(\boldsymbol{x}^{\boldsymbol{e}}) = \lambda^{e_j} \cdot y^{\sum_i e_i\left(k^{(i-1)} \bmod q\right)}.$$

Thus, if we know both $\Psi_{k,q}(\mathbf{x}^{\mathbf{e}})$ and $\Psi_{j,k,q}(\mathbf{x}^{\mathbf{e}})$ then we can recover the exponent of x_j from each of the monomials. To adapt this to finite fields, we note that all we need λ to satisfy is that its order in the multiplicative group $\mathbb{F}^*$, is larger than δ. Since the order of the group is $|\mathbb{F}| - 1$, we may have to pick λ from an extension field. Similarly, to interpolate a polynomial of degree δq we would need a field of size at least δq. This leads to the following construction of a robust interpolating set.

For the construction we shall need the following specialization of $\Psi_{j,k,q,\lambda}$ which we denote with $\Psi_{j,k,q,\lambda}[\alpha] : \mathbb{F}[\mathbf{x}] \to \mathbb{F}^n$, for any $\alpha \in \mathbb{F}$:

$$\Psi_{j,k,q,\lambda}[\alpha]_i = \begin{cases} \lambda \cdot \alpha^{(k^{j-1} \bmod q)} & i = j, \\ \alpha^{(k^{i-1} \bmod q)} & i \neq j. \end{cases}$$

In other words, $\Psi_{j,k,q,\lambda}[\alpha]$ is obtained by substituting α into y in the vector $(\Psi_{j,k,q,\lambda}(x_1), \ldots, \Psi_{j,k,q,\lambda}(x_n))$. We also denote with $\Psi_{k,q}[\alpha] : \mathbb{F}[\mathbf{x}] \to \mathbb{F}^n$ the map

$$\Psi_{k,q}[\alpha]_i = \alpha^{(k^{i-1} \bmod q)} ,$$

which is obtained by substituting α into y in the vector $(\Psi_{k,q}(x_1), \ldots, \Psi_{k,q}(x_n))$.

The following construction is far from being optimal in terms of its size, but it is relatively simple to describe.

Construction 2.7 (Robust interpolating set for sparse polynomials)**.** *Let $n, s, \delta \in \mathbb{N}$ and $\varepsilon > 0$ (ε may depend on n, s, δ). Let*

$$2\delta s^2 n^2/\varepsilon < q < 4\delta s^2 n^2/\varepsilon$$

be a prime number. Let $\mathcal{A}_{n,\delta,q,\varepsilon} \subset \mathbb{F}$ be a set of size

$$|\mathcal{A}_{n,\delta,q,\varepsilon}| = \lceil 2n\delta q/\varepsilon \rceil .$$

If $|\mathbb{F}|$ is too small then we pick $\mathcal{A}_{n,\delta,q,\varepsilon}$ from an extension field $\mathbb{E}$ of size $|\mathbb{F}|^2 \leqslant |\mathbb{E}| \leqslant 2n\delta q/\varepsilon \cdot |\mathbb{F}|$.

For $\lambda \in \mathbb{F}$ (or $\lambda \in \mathbb{E}$ if needed) denote

$$\mathcal{S}_{n,\delta,q,\varepsilon,\lambda} = \\ \{\Psi_{j,k,q,\lambda}[\alpha] \mid j \in [n], k \in [q-1], \alpha \in \mathcal{A}_{n,\delta,q,\varepsilon}\} \cup \{\Psi_{k,q}[\alpha] \mid k \in [q-1], \alpha \in \mathcal{A}_{n,\delta,q,\varepsilon}\} .$$

Observe that

$$|\mathcal{S}_{n,\delta,q,\varepsilon,\lambda}| = (n+1)(q-1)|\mathcal{A}_{n,\delta,q,\varepsilon}| = O\left(\delta n^2 q^2/\varepsilon\right) = O\left(\delta^3 n^6 s^4/\varepsilon^3\right) .$$

We first prove that $\mathcal{S}_{n,\delta,q,\varepsilon,\lambda}$ is a good hitting set for sparsity $2s$.

Lemma 2.8. *Assume the notation of Construction 2.7. Let $f \in \mathbb{F}[x]$ be an n-variate, $2s$-sparse polynomial of degree $\deg(f) = \delta$. Then, f vanishes on at most a fraction of $\frac{\varepsilon}{n}$ of the points in $\mathcal{S}_{n,\delta,q,\varepsilon,\lambda}$.*

Proof. By the "moreover" part of Theorem 2.5, the number of $k \in [q-1]$ for which $\Psi_{k,q}(f) = 0$ is at most $(2s-1)(n-1)$. For the rest of the k's, since $\deg(\Psi_{k,q}(f)) < \delta q$, it has at most δq zeroes. Thus, the total number of points in $\mathcal{S}_{n,\delta,q,\varepsilon,\lambda}$ on which f vanishes is at most

$$(2s-1)(n-1) \cdot (n+1)|\mathcal{A}_{n,\delta,q,\varepsilon}| + q \cdot q\delta < \left(\frac{2sn}{q} + \frac{\varepsilon}{n^2}\right) |\mathcal{S}_{n,\delta,q,\varepsilon,\lambda}| < \frac{\varepsilon}{n}|\mathcal{S}_{n,\delta,q,\varepsilon,\lambda}| . \qquad \square$$

We next prove that we can efficiently reconstruct from evaluations on a $(1-\varepsilon)$ fraction of the points in $\mathcal{S}_{n,\delta,q,\varepsilon,\lambda}$.

Theorem 2.9. *Let $n, s, \delta \in \mathbb{N}$ and $0 < \varepsilon < \frac{1}{50n}$. Let $\delta s^2 n < q < 2\delta s^2 n$ be a prime number. Assume that $\mathbb{F}$ is large enough to allow Construction 2.7. Let $\lambda \in \mathbb{F}^*$ be an element whose multiplicative order is larger than δq (if needed, pick $\lambda \in \mathbb{E}$ for an appropriate extension field $\mathbb{E}$).*

There is an algorithm that with the following guarantee: Let $f \in \mathbb{F}[\mathbf{x}]$ be an n-variate, s-sparse polynomial of degree $\deg(f) = \delta$. Denote with B the bit-complexity of f's coefficients.[6] *Let $\mathcal{S}' \subseteq \mathcal{S}_{n,\delta,q,\varepsilon,\lambda}$ be any subset of size $|\mathcal{S}'| \geqslant (1-\varepsilon)|\mathcal{S}_{n,\delta,q,\varepsilon,\lambda}|$. Given the set of evaluations*

$$\{(\boldsymbol{\beta}, f(\boldsymbol{\beta})) \mid \boldsymbol{\beta} \in \mathcal{S}'\} ,$$

the algorithm runs in time $\mathrm{poly}(n, s, \delta, 1/\varepsilon, B)$ *and returns* f.

Proof. For each $j \in [n]$, $k \in [q-1]$ denote

$$\mathcal{E}_{j,k,\lambda} := \{(\boldsymbol{\beta}, f(\boldsymbol{\beta})) \mid \boldsymbol{\beta} = \Psi_{j,k,q,\lambda}[\alpha], \text{ for some } \alpha \in \mathcal{A}_{n,\delta,q,\varepsilon}\} \cap \mathcal{S}' .$$

Similarly, for each $k \in [q-1]$ denote

$$\mathcal{E}_{k} := \{(\boldsymbol{\beta}, f(\boldsymbol{\beta})) \mid \boldsymbol{\beta} = \Psi_{k,q}[\alpha], \text{ for some } \alpha \in \mathcal{A}_{n,\delta,q,\varepsilon}\} \cap \mathcal{S}' .$$

By our assumption, and definition of $\mathcal{S}_{n,\delta,q,\varepsilon,\lambda}$, we have

$$(1-\varepsilon)|\mathcal{S}_{n,\delta,q,\varepsilon,\lambda}| \leqslant |\mathcal{S}'| = \sum_{j\in[n]} \sum_{k\in[q-1]} |\mathcal{E}_{j,k,\lambda}| + \sum_{k\in[q-1]} |\mathcal{E}_{k}| .$$

Call k good if for every $j \in [n]$, it holds that $|\mathcal{E}_{j,k,\lambda}| > \delta q$ and in addition, $|\mathcal{E}_k| > \delta q$.

Claim 2.10. *At most $(q-1)/4$ of the points $k \in [q-1]$ are not good.*

Proof. If k is not good then

$$|\mathcal{E}_k| + \sum_{j=1}^{n} |\mathcal{E}_{j,k,\eta}| \leqslant \delta q + n|\mathcal{A}_{n,\delta,q,\varepsilon}|$$
$$< (n+1)|\mathcal{A}_{n,\delta,q,\varepsilon}| \left(1 - \frac{1}{n+1} + \frac{\varepsilon}{2n^2}\right) < \frac{|\mathcal{S}_{n,\delta,q,\varepsilon,\lambda}|}{q-1}\left(1 - \frac{1}{4n}\right) ,$$

[6] If $\mathbb{F}$ is a finite field then $B = s \log |\mathbb{F}|$.

where we have used $|\mathcal{A}_{n,\delta,q,\varepsilon}| = \lceil 2n\delta q/\varepsilon \rceil$. Thus, if there are b bad k's, then we would have

$$(1-\varepsilon)|\mathcal{S}_{n,\delta,q,\varepsilon,\lambda}| \leqslant |\mathcal{S}'| = \sum_{k \in [q-1]} \left(|\mathcal{E}_k| + \sum_{j=1}^{n} |\mathcal{E}_{j,k,\eta}| \right) < |\mathcal{S}_{n,\delta,q,\varepsilon,\lambda}| \left(1 - \frac{b}{4nq}\right).$$

Hence $b < 4\varepsilon nq < (q-1)/4$. □

By our choice of q, Theorem 2.5 guarantees that at least half of all $k \in [q-1]$ satisfy that the univariate polynomial $\Psi_{k,q}(f)$ has the same number of monomials as f. Denote with K the set of all such $k \in [q-1]$. Then $|K| \geqslant (q-1)/2$. Claim 2.10 implies that there is some $k \in K$ which is good. We call such a k a good decoding point.

Let k be a good decoding point. Denote $f = \sum c_{\boldsymbol{e}} \boldsymbol{x}^{\boldsymbol{e}}$. Then, for every $j \in [n]$ we have

$$\Psi_{k,q}(f) = \sum c_{\boldsymbol{e}} y^{\sum_i e_i \left(k^{(i-1)} \bmod q\right)}$$

and

$$\Psi_{j,k,q,\lambda}(f) = \sum c_{\boldsymbol{e}} \lambda^{e_j} y^{\sum_i e_i \left(k^{(i-1)} \bmod q\right)} .$$

By choice of k, the number of monomials in each of these polynomials equals that of f. Furthermore, both polynomials have the same set of monomials and they differ only by their coefficients. Moreover, by (3), they have degree smaller than $\deg(f) \cdot q = \delta q$.

Thus, the evaluation points in $\mathcal{E}_k$ give us enough information to interpolate $\Psi_{k,q}(f)$. Similarly, the evaluation points in $\mathcal{E}_{j,k,\lambda}$ give us enough information to interpolate $\Psi_{j,k,q,\lambda}(f)$.

Since they have the same set of monomials, we can infer for each monomial $y^{\sum_i e_i \left(k^{(i-1)} \bmod q\right)}$ the term λ^{e_j}. As the order of λ is larger than δ, and in particular larger than e_j we can easily infer e_j from λ^{e_j}. Since we can do that for every coordinate j, we can infer the coefficient vector $\boldsymbol{e}$ corresponding to the monomial $y^{\sum_i e_i \left(k^{(i-1)} \bmod q\right)}$. Given the coefficients $c_{\boldsymbol{e}}$ that we found, we have the complete representation of f. □

2.3 Wronskian and Differential Operators over Finite Fields

We recall the definition of the *Wronskian*.[7] Let $g_1, \ldots, g_n \in \mathbb{F}[x]$. Their Wronskian is the determinant of the $n \times n$ matrix whose (i,j)-entry, for $i \in [[n-1]]$ and $j \in [n]$, is $\nabla^{(i)} g_j$, where

$$\nabla^{(i)} := \frac{d^i}{dx^i}.$$

$$W(g_1, \ldots, g_n) := \det \begin{pmatrix} g_1(x) & g_2(x) & \cdots & g_n(x) \\ \nabla g_1(x) & \nabla g_2(x) & \cdots & \nabla g_n(x) \\ \vdots & \vdots & \ddots & \vdots \\ \nabla^{(n-1)} g_1(x) & \nabla^{(n-1)} g_2(x) & \cdots & \nabla^{(n-1)} g_n(x) \end{pmatrix} \in \mathbb{F}[x]. \qquad (4)$$

[7] The Wronskian can be defined for any collection of functions that are sufficiently differentiable on an interval; here we consider only polynomials.

We treat ∇ as a linear operator on the space of polynomials, so it is well defined over any field.

The Wronskian is a classical object in the theory of differential equations (see, e.g., [Tes12, §3.4]), but it also plays an important role for polynomials over finite fields. The key fact is that if the characteristic of the field is sufficiently large, then the Wronskian vanishes if and only if the polynomials are linearly dependent. For completeness, we provide the proof of the following theorem in Appendix A.

Theorem 2.11. *Let $g_1, \dots, g_n$ be polynomials over a field $\mathbb{F}$ of characteristic p. If the maximum degree of any g_i is d and $p = 0$ or $p > d$, then $g_1, \dots, g_n$ are linearly dependent over $\mathbb{F}$ if and only if their Wronskian $W(g_1, \dots, g_n)$ is identically zero.*

The next theorem, an analogue of the fundamental theorem of linear homogeneous ODEs for polynomials over arbitrary fields, states that the solution space of an operator of order r has dimension at most r.

Theorem 2.12. *Let $\mathbb{F}$ be a field of characteristic p, and let*[8]

$$A \;=\; \sum_{i=0}^{r} Q_i(x)\,\nabla^i \;\in\; \mathbb{F}[x]\langle\nabla\rangle$$

be a nonzero operator of order r, with $Q_r \not\equiv 0$. Fix $D \in \mathbb{N}$ and assume $p = 0$ or $p > D$. Then the space

$$\mathcal{S}_{\leqslant D} \;=\; \{\, g \in \mathbb{F}[x] : \; \deg g \leqslant D, \; A(g) = 0 \,\}$$

has $\dim_{\mathbb{F}} \mathcal{S}_{\leqslant D} \leqslant r$.

Proof. Since $|\mathbb{F}| > D$, there exists $a \in \mathbb{F}$ with $Q_r(a) \neq 0$. Define the translated operator

$$\tilde{A} \;=\; \sum_{i=0}^{r} \tilde{Q}_i(x)\,\nabla^i, \qquad \tilde{Q}_i(x) := Q_i(x+a),$$

and the translated polynomial $\tilde{g}(x) := g(x+a)$. Since ordinary derivatives commute with translation, we have $A(g) = 0$ if and only if $\tilde{A}(\tilde{g}) = 0$. Write $\tilde{g}(x) = \sum_{i=0}^{D} c_i x^i$, where $c_0, \dots, c_D \in \mathbb{F}$ are indeterminates. Then

$$\nabla^k \tilde{g} = \frac{d^k \tilde{g}}{dx^k}(0) \;=\; k!\,c_k \qquad (0 \leqslant k \leqslant D).$$

Consider the identities $\nabla^t\left(\tilde{A}(\tilde{g})\right)(0) = 0$ for $t = 0, 1, \dots, D - r$. By Leibniz's rule

$$0 \;=\; \nabla^t\left(\tilde{A}(\tilde{g})\right)(0) \;=\; \sum_{i=0}^{r}\sum_{j=0}^{t} \binom{t}{j} \nabla^{t-j}\tilde{Q}_i(0)\,\nabla^{i+j}\tilde{g}_i(0),$$

[8] Since x and ∇ do not commute, we denote the noncommutative polynomial ring of differential operators with coefficients in $\mathbb{F}[x]$ by $\mathbb{F}[x]\langle\nabla\rangle$.

hence, using $\nabla^{i+j}\tilde{g}_i(0) = (i+j)!\, c_{i+j}$,

$$\sum_{i=0}^{r}\sum_{j=0}^{t}\binom{t}{j}\,\nabla^{t-j}\tilde{Q}_i(0)\,(i+j)!\,c_{i+j} \;=\; 0, \qquad t = 0,1,\ldots,D-r. \tag{5}$$

In (5), the term with $(i,j) = (r,t)$ equals

$$\binom{t}{t}\,\tilde{Q}_r(0)\,(r+t)!\,c_{r+t} \;=\; Q_r(a)\,(r+t)!\,c_{r+t},$$

while all other terms involve only c_k with $k \leqslant r+t-1$. Because $p = 0$ or $p > D$, each factorial $(r+t)!$ is invertible in $\mathbb{F}$ for $0 \leqslant t \leqslant D-r$, and $Q_r(a) \neq 0$ by construction. Thus, the coefficient of c_{r+t} is nonzero. Therefore, for each t, we can solve uniquely for c_{r+t} in terms of $c_0,\ldots,c_{r+t-1}$. Inductively, all $c_r,\ldots,c_D$ are determined by the initial block $(c_0,\ldots,c_{r-1}) \in \mathbb{F}^r$, so $\dim_{\mathbb{F}} \mathcal{S}_{\leqslant D} \leqslant r$. □

2.4 Polynomial Factorization

We shall need the following result on factorization of polynomials. The first result is the famous LLL algorithm for factorization over $\mathbb{Q}$.

Theorem 2.13 (Factorization over $\mathbb{Q}$, [LLL82]). *Let $f \in \mathbb{Q}[x]$ have degree d and bit complexity B. Then, there is a deterministic algorithm that outputs all irreducible factors of f whose running time is* $\mathrm{poly}(d, B)$.

For the following theorem, see, e.g. [vzGS92, Section 9].

Theorem 2.14 (Factorization over Finite Fields). *Let $\mathbb{F}_q$ be a field of size $q = p^k$ and characteristic $p > 0$. Let $f(x) \in \mathbb{F}_q[x]$ be a polynomial of degree d. Then, there is a deterministic algorithm that outputs all irreducible factors of f in time* $\mathrm{poly}(p, \log q, d)$.

3 Hitting set for $\Sigma^{[r]}\wedge^{[d]}\Sigma^{[s]}\Pi^{[\delta]}$ circuits with $r^2 \leqslant d$

In this section we give a simple polynomial-size hitting set for $\Sigma^{[r]}\wedge^{[d]}\Sigma^{[s]}\Pi^{[\delta]}$ circuits when $d = \Omega(r^2)$.

Theorem 3.1. *Let $r, d, s, n, \delta \in \mathbb{N}$ such that $(r-1)^2 \leqslant d+1$. Let q be a prime such that $q \geqslant \max(\delta, r^2s^2n) + 1$ and let $t \geqslant q/\varepsilon$. Let $\mathbb{F}$ be a field of characteristic p such that $p = 0$ or $p > rd\delta q$. Let $f(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ be a nonzero polynomial that is computable by a $\Sigma^{[r]}\wedge^{[d]}\Sigma^{[s]}\Pi^{[\delta]}$ circuit. Then, except for at most r^2s^2n values of $k \in [t]$, the specialization $\Psi_{k,q}(f)$ defined as in Theorem 2.5 is nonzero.*

Proof. Let $f = \sum_{i=1}^{r} f_i^d$, such that each f_i has degree at most δ and at most s monomials, and any two are non-associate. Let $\Psi_{k,q}(f_i)$ be as in Corollary 2.6. Then, by a simple application of the union bound, except for at most r^2s^2n values of k, the polynomials $\Psi_{k,q}(f_1),\ldots,\Psi_{k,q}(f_r)$ are non-associate of degree smaller than δq. Fix a k that guarantees such pairwise linear independence. Theorem 2.4 guarantees that $\Psi_{k,q}(f) = \sum_{i=1}^{r}\Psi_{k,q}(f_i)^d \neq 0$. □

Theorem 3.2 (Hitting set for $\Sigma^{[r]}\wedge^{[d]}\Sigma^{[s]}\Pi^{[\delta]}$ circuits, when $d = \Omega(r^2)$)**.** *Let* n, r, s, d, δ *be as in Theorem 3.1. Let* $\mathbb{F}$ *be a field of characteristic* p *such that* $p = 0$ *or* $p \geqslant rd\delta(s^2n + \delta)$*. Then, for every* $\varepsilon > 0$ *there is an explicit hitting set* $\mathcal{H}$ *of size* $|\mathcal{H}| = O(r^4s^4n^2d\delta^3/\varepsilon)$ *such that every nonzero polynomial computed by a* $\Sigma^{[r]}\wedge^{[d]}\Sigma^{[s]}\Pi^{[\delta]}$ *circuit is nonzero on at least a* $(1-\varepsilon)$ *fraction of the points in* $\mathcal{H}$*.*

Proof. By Bertrand's postulate there exists a prime $q \in [r^2s^2n + \delta + 1, 2r^2s^2n + 2\delta]$. Set $t = 2r^2s^2n + 2\delta + 1$. For each $k \leqslant t$, evaluate $\Psi_{k,q}(f)$ on $\deg(\Psi_{k,q}(f))/\varepsilon$ distinct points. By Theorem 2.5, $\deg(\Psi_{k,q}(f)) \leqslant \delta q d = O(\delta d(r^2s^2n + \delta))$, so $|\mathcal{H}| = O(r^4s^4n^2d\delta^3/\varepsilon)$.

For each good k, $\Psi_{k,q}(f)$ is nonzero on at least a $(1-\varepsilon)$ fraction of its evaluation points, and at least $(1-\varepsilon)t$ values of k are good. Hence f is nonzero on at least a $(1-2\varepsilon)$ fraction of the points in $\mathcal{H}$. □

Remark 3.3. Our construction of the hitting sets is not optimal and it can be improved with more care.

4 Reconstruction of Univariate Sums of Powers

In this section, we consider black-box reconstruction of univariate polynomials of the form

$$f(x) = \sum_{i=1}^{r} \alpha_i (f_i(x))^d \in \mathbb{F}[x], \quad \deg(f_i) \coloneqq \delta_i \leqslant \delta, \ \|f_i\|_0 \leqslant s, \tag{6}$$

where $\mathbb{F}$ has characteristic 0 or characteristic $p > 2rd\delta$. As we allow coefficients $\alpha_i \in \mathbb{F}$, we can assume w.l.o.g. that each f_i is a monic polynomial.

We note that we only have an upper bound on the number of summands, r, and in reality, f can be represented as a sum of $r' < r$ terms. This distinction does not significantly affect our algorithm, as we will simply try each $r' \in [1, r]$. Thus, for simplicity, we assume that r is the exact number of summands, and we continue to denote it by r. From now on, until we present the algorithm itself, we assume that f cannot be represented as in (6) with fewer terms. In particular, this implies that the polynomials f_i^d are linearly independent.

As $f(x)$ is a univariate polynomial with $\deg(f) \leqslant d\cdot\delta$, we can interpolate its coefficients to obtain a representation

$$f(x) = \sum_{i=0}^{d\delta} \beta_i x^i.$$

Hence, we may assume that this representation of $f(x)$ is given as input to the algorithm.

As described in Section 1.3, our approach is to learn a degree r differential operator L such that $L(f) = 0$. We next describe some properties of the sought-after operator.

4.1 The Differential Operator L

We shall consider the Wronskian of $f, f_1^d, \ldots, f_r^d$ (recall Section 2.3):

$$W(x) = W(f, f_1^d, \ldots, f_r^d) \coloneqq \det \begin{pmatrix} f(x) & f_1^d(x) & \cdots & f_r^d(x) \\ \nabla f(x) & \nabla f_1^d(x) & \cdots & \nabla f_r^d(x) \\ \vdots & \vdots & \ddots & \vdots \\ \nabla^r f(x) & \nabla^r f_1^d(x) & \cdots & \nabla^r f_r^d(x) \end{pmatrix}. \tag{7}$$

Clearly, $W(x) \equiv 0$ because the first column is a linear combination of the others. Expanding the determinant along the first column yields

$$\sum_{i=0}^{r} (-1)^i \, W_i(x) \, \nabla^i f(x) \equiv 0, \tag{8}$$

where $W_i(x)$ denotes the determinant of the $r \times r$ minor obtained by deleting the i-th row and the first column of the matrix in (7).

Claim 4.1. *Each polynomial $W_k(x)$, obtained from the Laplace expansion of the Wronskian, can be written as*

$$W_k(x) = \left(\Pi_{i \in [r]} f_i^{d-r}\right) \tilde{P}_k(x),$$

where $\tilde{P}_k(x)$ is a polynomial of degree at most $r^2\delta$.

Proof. Recall that $\deg(f_i) = \delta_i \leqslant \delta$. Fix $k \in [[r]]$. For each $i \in [[r]] \setminus \{k\}$ and $j \in [r]$, we can write

$$\nabla^{(i)} f_j^d \;=\; f_j^{d-r} \, g_{i,j}, \qquad \text{with} \qquad \deg(g_{i,j}) = r\delta_j - i \;\leqslant\; r\delta\,. \tag{9}$$

Hence

$$W_k(x) = (-1)^{k+1} \left(\Pi_{j \in [r]} f_j^{d-r}\right) \det\bigl(g_{i,j}(x)\bigr)_{i \in [[r]] \setminus \{k\}, j \in [r]}.$$

Define

$$\tilde{P}_k(x) \;\coloneqq\; (-1)^{k+1} \det\bigl(g_{i,j}(x)\bigr)_{i \in [[r]] \setminus \{k\}, j \in [r]}.$$

Then

$$W_k(x) = \left(\Pi_{j \in [r]} f_j^{d-r}\right) \tilde{P}_k(x).$$

Since each $g_{i,j}$ has degree at most $r\delta$, $\tilde{P}_k(x)$ has degree at most $r^2\delta$, as claimed. □

Let $P = \gcd(\tilde{P}_0, \ldots, \tilde{P}_r)$ and denote $P_i = \tilde{P}_i / P$.

Corollary 4.2. *Assume that p, the characteristic of $\mathbb{F}$, satisfies $p = 0$ or $p > d\delta$. Then the differential operator*

$$\tilde{L} \;\coloneqq\; \sum_{i=0}^{r} P_i(x) \, \nabla^{(i)} \in \mathbb{F}[x]\langle \nabla \rangle$$

is not identically zero, and its solution space is spanned by $\{f_1^d, \ldots, f_r^d\}$. In particular, $\tilde{L}$ annihilates both f and each f_i^d.

Proof. The fact that $\tilde{L}$ is not identically zero follows from Theorem 2.11, since $\tilde{L}$ would remain unchanged if f was replaced by any polynomial which is not a linear combination of $\{f_1^d, \dots, f_r^d\}$.

From Claim 4.1 and (8), we have

$$\begin{aligned} 0 &= \sum_{i=0}^{r} (-1)^i \, W_i(x) \, \nabla^i f(x) \\ &= \left(\Pi_{j\in[r]} f_j^{d-r}\right) \cdot \sum_{i=0}^{r} \tilde{P}_i(x) \, \nabla^i f(x) \\ &= \left(\Pi_{j\in[r]} f_j^{d-r}\right) \cdot P \cdot \sum_{i=0}^{r} P_i(x) \, \nabla^i f(x) \\ &= \left(\Pi_{j\in[r]} f_j^{d-r}\right) \cdot P \cdot \tilde{L}(f) \,. \end{aligned}$$

Since $\left(\Pi_{j\in[r]} f_j^{d-r}\right) \cdot P$ is not identically zero, it follows that $\tilde{L}(f) \equiv 0$. The same argument applies to any linear combination of $f_1^d, \dots, f_r^d$, hence

$$\mathrm{span}\{f_1^d, \dots, f_r^d\} \subseteq \ker\left(\tilde{L}\right) .$$

By Theorem 2.12, the solution space of an operator of order r has dimension at most r. As the f_i^d are linearly independent, they form a basis of $\ker\left(\tilde{L}\right)$. □

We now set up a linear system whose solution yields the desired operator L from Corollary 4.2. Consider the following system of linear equations with unknown coefficients $\gamma_{i,j} \in \mathbb{F}$:

$$\sum_{i=0}^{r} \left(\sum_{j=0}^{r\delta} \gamma_{i,j} x^j \right) \nabla^i f(x) \; \equiv \; 0 \,. \tag{10}$$

The system (10) consists of $\deg(f) + 1$ linear equations in the unknowns $\gamma_{i,j}$, with coefficients that are linear combinations of the coefficients of f.

Clearly, one solution is obtained by setting

$$\sum_{j=0}^{r\delta} \gamma_{i,j} x^j \; = \; P_i(x) \,,$$

which corresponds exactly to the operator L in Corollary 4.2. We next show that any nontrivial solution of the form $\sum_{j=0}^{r} Q_j \nabla^j$ satisfying $\gcd(Q_0, \dots, Q_r) = 1$ is a scalar multiple of $\tilde{L}$. In particular, any nontrivial solution that minimizes $\deg(Q_r)$ equals a scalar multiple of $\tilde{L}$. For this, we rely on the ABC theorem for function fields discussed in Section 2.1.

4.1.1 Uniqueness of L

Claim 4.3. *Assume that the characteristic of the field $\mathbb{F}$ satisfies $p = 0$ or $p > 2rd\delta$. Let $\tilde{L} \in \mathbb{F}[x]\langle\nabla\rangle$ be the operator from Corollary 4.2. Assume $d > (r+1)^4\delta$. Then any nontrivial operator $L' = \sum_{i=0}^{r} Q_i(x)\nabla^i$ satisfying $\deg(Q_i) \leqslant r^2\delta$ for all i, $\gcd(Q_0, \dots, Q_r) = 1$, and $L'(f) = 0$ must equal a scalar multiple of $\tilde{L}$.*

Proof. Let L' be as in the statement of the claim. We have that

$$\begin{aligned} 0 \equiv L'(f) &= \sum_{j=1}^{r} \beta_j L'(f_j^d) \\ &= \sum_{j=1}^{r}\sum_{i=0}^{r} \beta_j Q_i(x)\nabla^i(f_j^d) \\ &= \sum_{j=1}^{r}\sum_{i=0}^{r} \tilde{Q}_{i,j} f_j^{d-i} \\ &= \sum_{j=1}^{r} \tilde{Q}_j f_j^{d-r}, \end{aligned}$$

where $\tilde{Q}_{i,j}$ is defined similarly to (9), and $\beta_j L'(f_j^d) = \tilde{Q}_j f_j^{d-r}$. Observe that, as in (9),

$$\deg(\tilde{Q}_{i,j} f_j^{r-i}) \leqslant r^2\delta + (i\delta - i) + (r-i)\delta .$$

Hence, $\deg(\tilde{Q}_j) \leqslant (r^2+r)\delta$. Since the f_i are non-associate,

$$d - r > (r+1)^4\delta - r = (r^2+2r+1)^2\delta - r \geqslant (r^2+r)((r^2+r)\delta+1),$$

and $p = 0$ or $p \geqslant 2rd\delta > r((r^2+r)\delta + d\delta)$, Corollary 2.2 implies that $\tilde{Q}_j \equiv 0$ for all j. Consequently, $L'(f_j^d) = 0$ for all j. Theorem 2.12 implies that the kernel of L' is spanned by f_j^d. Observe now that the operator $Q_r \cdot \tilde{L} = P_r \cdot L'$ has degree at most $r-1$ and its kernel contains all f_j^d. Hence, by Theorem 2.12 it must be identically zero. As $\gcd(P_0, \ldots, P_r) = \gcd(Q_0, \ldots, Q_r) = 1$ the claim follows. □

Corollary 4.4. *Let p, d satisfy the requirements of Claim 4.3. Then there exists a unique operator*

$$L = \sum_{i=0}^{r} Q_i(x)\,\nabla^i$$

satisfying $\deg Q_i \leqslant r^2\delta$ for all i, $\gcd(Q_0, \ldots, Q_r) = 1$, Q_r monic, and $L(f) = 0$. Furthermore, L can be computed in time $\mathrm{poly}(n)$.

Proof. The claim regarding the uniqueness follows immediately from Claim 4.3 and the fact that we chose Q_r to be monic and of minimal degree.

Since we can find, in polynomial time, a basis for the solution space of (10), by simple diagonalization we can find a solution that minimizes $\deg(Q_r)$ such that Q_r is monic. □

We next show how to extract the polynomials f_i given L.

4.1.2 Properties of the Solution Space of L

Claim 4.5. *Given the solution space of L, if $r^2 + r \leqslant d$ then there are at most r polynomials in the solution space that are powers of d.*

Proof. Assume there is a polynomial $q(x)$ such that $L(q(x)^d) = 0$ and q is not a scalar multiple of any of the f_j. Since $\{f_i^d\}_{i\in[r]}$ span the solution space of L, there are scalars α_i such that $\sum_i \alpha_i f_i^d = q^d$. However, by choice of d and q, and since the polynomials $\{f_i\} \cup \{q\}$ are non-associate, we get a contradiction to Corollary 2.2. □

Thus, our task is finding all polynomials that are d-th powers in $\ker(L)$. We next show that each root of each f_j is also a root of Q_r.

Definition 4.6 ($\mathrm{ord}_z(g)$)**.** Let $g(x)$ be a polynomial. We define $\mathit{ord}_z(g)$ to be the multiplicity of z as a root of $g(x)$. Similarly, for an irreducible polynomial $\phi(x)$ we define $\mathrm{ord}_\phi(g)$ to be the largest power e such that $\phi^e | g$.

Claim 4.7. *If $d \geqslant r$ then the leading coefficient $Q_r(x)$ of $L = \sum_{i=0}^{r} Q_i(x)\nabla^{(i)}$ vanishes at every root z of every f_i.*

Proof. Fix a zero $z \in \overline{\mathbb{F}}$ of some f_j, of multiplicity $\geqslant 1$. Then, $\mathrm{ord}_z(f_j^d) = m \geqslant d \geqslant r$. It follows that for $i \leqslant r-1$, $\mathrm{ord}_z(\nabla^i(f_j^d)) = m - i > m - r \geqslant 0$. Assume $Q_r(z) \neq 0$. We now have

$$0 \equiv L(f_j^d) = Q_r\nabla^r(f_j^d) + \sum_{i=0}^{r-1} Q_i\nabla^i(f_j^d)\,.$$

Thus,

$$m - r = \mathrm{ord}_z(Q_r\nabla^r(f_j^d)) = \mathrm{ord}_z(-\sum_{i=0}^{r-1} Q_i\nabla^i(f_j^d)) \geqslant m - r + 1\,,$$

in contradiction. □

Corollary 4.8. *Each irreducible factor, over $\mathbb{F}$, of each f_i is an irreducible factor of Q_r. Furthermore, the factorization of Q_r can be computed efficiently.*

Proof. This follows immediately from Claim 4.7 by considering minimal polynomials of roots. The claim about efficiency follows from Theorem 2.13 and Theorem 2.14, depending on the underlying field. □

Remark 4.9. Corollary 4.8 tells us that, upon factorizing the leading coefficient $Q_r(x)$, we obtain a set of m irreducible factors $\phi_1, \ldots, \phi_m$ of Q_r.

Hence, we can iterate over all possible multiplicity vectors $\mathbf{e}$, of which there are at most $\binom{m+\delta}{\delta} \approx (r^2\delta)^\delta$, to construct candidate polynomials

$$g_{\mathbf{e}}(x) = \Pi_{j\in[m]}\phi_j^{e_j}.$$

For each such $g_{\mathbf{e}}$, we can apply L to $g_{\mathbf{e}}^d$ to test whether $g_{\mathbf{e}}^d \in \ker(L)$. This yields an algorithm with running time $\mathrm{poly}(n, d, (r\delta)^\delta)$.

However, our goal is to design an algorithm whose running time is polynomial in all the parameters.

4.2 Efficiently Recovering the f_i

We first note that a basis for $\ker(L)$ can be computed in polynomial time. Indeed, consider $d\delta + 1$ unknowns a_i and consider the linear system of equations

$$\sum_{j=0}^{r} Q_j \left(\sum_{i=0}^{\delta d} a_i x^i \right) \equiv 0 \,. \tag{11}$$

Clearly, this system is solvable in time $\mathrm{poly}(d, \delta)$. Furthermore, since $\deg(f_j^d) \leqslant d\delta$, each f_j^d lies in the solution space. As these polynomials span $\ker(L)$, we are guaranteed to find a basis to $\ker(L)$ in this process.

Let $h_1, \ldots, h_r$ be the basis found for $\ker(L)$. Thus,

$$\ker(L) = \mathrm{span}\{f_1^d, f_2^d, \ldots, f_r^d\} = \mathrm{span}\{h_1(x), h_2(x), ..., h_r(x)\}, \quad \forall i \in [r],\ \deg(h_i(x)) \leqslant \delta d \,.$$

Therefore, our goal is to find all f_j^d given the h_i. We achieve this by the following observation: if some $g \in \ker(L)$ has a root z such that $\mathrm{ord}_z(g) \geqslant d$, where d is large enough, then z must be a root of one of the f_j.

Claim 4.10. *Assume $d > 2r^2\delta$. If z is a root of order $t > r^2\delta + (r-1)$ of some polynomial $h \in \mathrm{span}\{f_j^d : j \in [r]\}$, then it is a root of one of the f_j.*

Proof. Denote $h = \sum_{i \in [r]} c_i f_i^d$. Assume, w.l.o.g., that $c_1 = 1$. Consider the order $(r-1)$ Wronskian

$$W(f_1^d, ..., f_r^d) \in \mathbb{F}[x], \quad W := \det(\nabla^i f_j^d)_{i \in [[r-1]], j \in [r]} \,.$$

As $c_1 = 1$, we can replace the first column of the matrix in the Wronskian with the vector whose i-th entry is $(\nabla^i h)$, without affecting W. Since $\mathrm{ord}_z(h) = t > r$, each entry satisfies $\mathrm{ord}_z(\nabla^i h) = t - i \geqslant t - (r-1)$. Consequently, z is a root of W of order at least $t - r + 1$. As in the proof of Claim 4.1, the Wronskian factorizes as $\prod_{i \in [r]} f_i^{d-r+1} W'$, where W' is a polynomial of degree at most $r^2\delta$. Since

$$\mathrm{ord}_z(W) \geqslant t - r > r^2\delta \geqslant \deg(W') \,,$$

we conclude that $\mathrm{ord}_z(\prod_{i \in [r]} f_i^{d-r+1}) > 0$. Hence, z is a root of one of the f_i. □

The following simple corollary is at the heart of our algorithm.

Corollary 4.11. *Let d be as in Claim 4.10. The subspace of all polynomials that have a root at z of order at least d is spanned by all the f_i^d that have z as a root.*

Proof. Assume w.l.o.g., that among $\{f_i^d\}$ only $f_1^d, ..., f_k^d$ have z as a root, and hence it is a root of order at least d at each of them. Assume for contradiction that some nonzero polynomial $h \in \mathrm{span}\{f_{k+1}^d, ..., f_r^d\}$ satisfies $\mathrm{ord}_z(h) \geqslant d$. Applying Claim 4.10 to $h \in \mathrm{span}\{f_{k+1}^d, ..., f_r^d\}$ shows that z must be a root of some f_j for $k < j \leqslant r$, a contradiction. □

Corollary 4.12. *Let $\phi \in \mathbb{F}[x]$ be an irreducible polynomial. Let d be as in Claim 4.10. The subspace of all polynomials that have ϕ^d as a factor is spanned by all the f_i^d such that ϕ is a factor of f_i.*

Proof. Apply Corollary 4.11 to any root of ϕ. □

Armed with Corollary 4.12, our next step is to sieve the polynomials in $\ker(L)$ that have a factor of high multiplicity.

Claim 4.13. *Given an irreducible* $\phi \in \mathbb{F}[x]$ *such that* $\deg(\phi) \leqslant \delta$, *and* $e \leqslant d$, *we can find in time* $\mathrm{poly}(d, r, \delta)$ *a basis for the space of polynomials*

$$\{\, g \in \mathrm{span}\{h_1, \ldots, h_r\} \;\big|\; \mathit{ord}_{\phi}(g) \geqslant e \,\}.$$

Proof. Let $b_1, \ldots, b_r$ and $a_0, \ldots, a_{\delta d - e\deg(\phi)}$ be indeterminates. Solve the homogeneous linear system

$$\phi^e \cdot \left(\sum_{i=0}^{\delta d - e\deg(\phi)} a_i x^i \right) = \sum_{i=1}^{r} b_i h_i \ .$$

Any basis for the solution space immediately gives a basis for the requested subspace. □

As explained in Section 1.3, we recover the monic polynomials f_i^d by performing a DFS (with pruning) on a suitable tree. To ease the reading, we repeat the main idea. Let $\{\phi_j\}_{j\in[m]}$ denote the irreducible factors of Q_r. As proved in Corollary 4.8, this set contains all factors of each f_i. Hence, each f_i corresponds to an exponent vector $\boldsymbol{e}_i$ defined by $\boldsymbol{e}_{i,j} = \mathrm{ord}_{\phi_j}(f_i)$. Equivalently,

$$f_i(x) = \Pi_{j\in[m]} \phi_j^{e_{i,j}}.$$

Clearly, for each i, we have $\sum_j e_{i,j} \deg(\phi_j) = \deg(f_i) \leqslant \delta$.

Consider now a rooted tree of depth δ, where each node has m children, and the edge to the j-th child is labeled by ϕ_j. Every root-to-leaf path in the tree corresponds to an exponent vector, counting the multiplicities of the ϕ_j encountered along the path. Label each node with the exponent vector defined by the path from the root to that node, and associate to each such node the subspace of polynomials in $\ker(L)$ that have ϕ_j^d as a factor with multiplicity equal to the number of times ϕ_j appears along the path.

Note, however, that this tree is highly redundant, since many distinct nodes correspond to the same subspace. Moreover, the tree has m^δ leaves.

Our algorithm performs a "smart" depth-first search (DFS) on this tree to recover the f_j in the lexicographic order of their exponent vectors. Importantly, the algorithm visits only those vertices that are guaranteed to lead to previously undiscovered polynomials, and thus visits in total only $\mathrm{poly}(r, \delta)$ vertices overall.

Theorem 4.14. *Let* $\{\phi_1, \ldots, \phi_m\}$ *be the set of irreducible factors of* Q_r. *Assume* $d \geqslant (r+1)^4\delta$ *and the characteristic is* $p = 0$ *or* $p > 2dr\delta$. *Then there exists a deterministic algorithm that, given a basis* $\{h_1, \ldots, h_r\}$ *for* $\ker(L)$, *recovers all* f_i^d *in time* $\mathrm{poly}(m, r, \delta, d)$, *or* $\mathrm{poly}(m, r, \delta, d, \log p)$ *when* $p > 0$.

We first give the pseudocode of the algorithm (Algorithm 1) and then prove that it satisfies the claim in the theorem.

Algorithm 1 DFS-Recover-f_i^d

```
Require: Basis $\mathcal{B}_0$ of $V(\mathbf{0}) = \ker(L)$; irreducible polynomials $\phi_1, \ldots, \phi_m$ of degree at most $\delta$
Ensure: A list $U$ containing all $f_i^d$
 1: $U \leftarrow \varnothing$                                   ▷ set of recovered polynomials $f_i^d$
 2: procedure DFS($\boldsymbol{e}, \mathcal{B}_{\boldsymbol{e}}, U$)       ▷ $\boldsymbol{e}$ is a multiplicity vector and $\mathcal{B}_{\boldsymbol{e}}$ spans the current ambient space
 3:     if $\dim \operatorname{span} \mathcal{B}_{\boldsymbol{e}} = 1$ then
 4:         let $g$ be the unique monic polynomial in $\operatorname{span} \mathcal{B}_{\boldsymbol{e}}$
 5:         $F(x) \leftarrow \left(\prod_{j=1}^{m} \phi_j^{e_j}\right)^d g(x)$
 6:         $U \leftarrow U \cup \{F\}$; return
 7:     end if
 8:     $U(\boldsymbol{e}) \leftarrow \{\, q/(\prod_{j=1}^{m} \phi_j^{e_j})^d : q \in U,\ (\prod_{j=1}^{m} \phi_j^{e_j})^d \mid q \,\}$ ▷ reduce $U$ to the current ambient space $V(\boldsymbol{e})$
 9:     for $j = 1$ to $m$ do
10:         $A_j(\boldsymbol{e}) \leftarrow \{\, g \in \operatorname{span} \mathcal{B}_{\boldsymbol{e}} : \operatorname{ord}_{\phi_j}(g) \geqslant d \,\}$
11:         if $A_j(\boldsymbol{e}) \neq \{0\}$ and $A_j(\boldsymbol{e}) \nsubseteq \operatorname{span} U(\boldsymbol{e})$ then
12:             $\mathcal{B}_{\boldsymbol{e}+\boldsymbol{e}_j} \leftarrow \{\, g/\phi_j^d : g \in \operatorname{basis}(A_j(\boldsymbol{e})) \,\}$
13:             DFS($\boldsymbol{e} + \boldsymbol{e}_j, \mathcal{B}_{\boldsymbol{e}+\boldsymbol{e}_j}, U$)
14:         end if
15:     end for
16: end procedure
17: DFS($\mathbf{0}, \mathcal{B}_0, U$)
18: return $U$
```

Proof. We naturally identify nodes in the tree by their multiplicity vectors $\mathbf{e}$, where the root corresponds to the all-zero vector.

Denote by T_{DFS} the tree defined by the DFS algorithm. That is, there is an edge between a node that is labeled by the multiplicity vector $\mathbf{e}$ to its child $\mathbf{e}_j$ only if the condition in Step 11 holds.

Define the vector space

$$V(\mathbf{0}) = \ker(L) \ .$$

We associate to each node $u \in T_{\text{DFS}}$ with multiplicity vector $\mathbf{e}$ two linear spaces $V(\boldsymbol{e})$ and $A_j'(\boldsymbol{e})$ that are defined recursively (starting from $V(\mathbf{0})$):

$$A_j'(\boldsymbol{e}) = \{\, g \in V(\boldsymbol{e}) : \operatorname{ord}_{\phi_j}(g) \geqslant d \,\} \quad \text{and} \quad V(\boldsymbol{e} + \boldsymbol{e}_j) = \{\, g/\phi_j^d : g \in A_j'(\boldsymbol{e}) \,\}. \tag{12}$$

We next show the relation of these spaces to the algorithm.

Claim 4.15. *For every node $u \in T_{DFS}$ with multiplicity vector $\mathbf{e}$, and $j \in [m]$ such that $\mathbf{e} + \mathbf{e}_j$ satisfy the condition in Step 11, the following hold:*

(I) $A_j'(\boldsymbol{e}) = A_j(\boldsymbol{e})$.

(II) $V(\boldsymbol{e})$ *is spanned by*

$$\mathcal{B}(\boldsymbol{e}) = \left\{ \left(\frac{f_i}{\prod_{j=1}^m \phi_j^{e_j}} \right)^d : \mathit{ord}_{\phi_j}(f_i) \geqslant e_j \ \forall j \right\}.$$

(III) It holds that $\mathcal{B}_{\boldsymbol{e}} = \mathcal{B}(\boldsymbol{e})$*.*

Proof. The proof is induction on the depth of the DFS tree, starting at the root.

For the root, $\boldsymbol{e} = \boldsymbol{0}$, and $\mathcal{B}(\boldsymbol{0}) = \mathcal{B}_0 = \ker(K)$. Thus, $V(\boldsymbol{0}) = \ker(L) = \operatorname{span}\{f_1^d, \ldots, f_r^d\}$, so (II) and (III) hold. Now, inspecting Step 10 we see that (I) holds as well.

Induction step. Assume (I),(II),(III) hold at a node u with multiplicity vector $\boldsymbol{e}$, and fix $j \in [m]$ such that the node with multiplicity vector $\boldsymbol{e} + \boldsymbol{e}_j$ is in T_{DFS}. By the induction hypothesis

$$A_j(\boldsymbol{e}) = A'_j(\boldsymbol{e}) = \{ g \in V(\boldsymbol{e}) : \operatorname{ord}_{\phi_j}(g) \geqslant d \}.$$

Applying Corollary 4.12 to $V(\boldsymbol{e})$ with spanning set (in fact, basis) $\mathcal{B}(\boldsymbol{e}) = \mathcal{B}_{\boldsymbol{e}}$ gives

$$\begin{aligned} A'_j(\boldsymbol{e}) &= \operatorname{span}\{ h \in \mathcal{B}(\boldsymbol{e}) : \operatorname{ord}_{\phi_j}(h) \geqslant d \} \\ &= \operatorname{span}\left\{ \left(\tfrac{f_i}{\prod_{t=1}^m \phi_t^{e_t}} \right)^d : \operatorname{ord}_{\phi_j}(f_i) \geqslant e_j + 1, \operatorname{ord}_{\phi_t}(f_i) \geqslant e_t \ \forall t \right\}. \end{aligned}$$

Dividing by ϕ_j^d yields

$$V(\boldsymbol{e} + \boldsymbol{e}_j) = \operatorname{span}\left\{ \left(\tfrac{f_i}{\prod_{t=1}^m \phi_t^{e'_t}} \right)^d : \operatorname{ord}_{\phi_t}(f_i) \geqslant e'_t \ \forall t \right\},$$

where $e'_t = e_t$ for $t \neq j$ and $e'_j = e_j + 1$, establishing (II). Moreover, since $\mathcal{B}(\boldsymbol{e}) = \mathcal{B}_{\boldsymbol{e}}$ we see that $\mathcal{B}_{\boldsymbol{e}+\boldsymbol{e}_j}$ defined in Step 12 satisfies $\mathcal{B}_{\boldsymbol{e}+\boldsymbol{e}_j} = \mathcal{B}(\boldsymbol{e} + \boldsymbol{e}_j)$ so (III) holds. Similarly, we see that for every $i \in [m]$ for which there is an edge from $\boldsymbol{e} + \boldsymbol{e}_j$ to $\boldsymbol{e} + \boldsymbol{e}_j + \boldsymbol{e}_i$, the definitions of $A_i(\boldsymbol{e} + \boldsymbol{e}_j)$ and $A'_i(\boldsymbol{e} + \boldsymbol{e}_j)$ coincide. □

For simplicity, we slightly abuse notation and use $A_j(u)$ and $V(u)$ whenever we are at a node $u \in T_{\mathrm{DFS}}$.

As a consequence of the claim we get that for every leaf u (where $\dim V(u) = 1$), there exists a unique k such that

$$V(u) = \operatorname{span}\left\{ \left(\tfrac{f_k}{\prod_{t=1}^m \phi_t^{e_t}} \right)^d \right\},$$

and for the unique monic $g \in V(u)$ we have

$$\left(\textstyle\prod_{t=1}^m \phi_t^{e_t}\right)^d g(x) = f_k^d(x),$$

thus identifying the recovered polynomial, which justifies Step 5 of the algorithm.

At node u with vector $\boldsymbol{e}$, define

$$U(\boldsymbol{e}) = \left\{ q / \left(\textstyle\prod_{t=1}^m \phi_t^{e_t}\right)^d : q \in U, \left(\textstyle\prod_{t=1}^m \phi_t^{e_t}\right)^d \mid q \right\} \subseteq V(u).$$

We recurse into child j only if

$$A_j(u) \not\subseteq \operatorname{span} U(\boldsymbol{e}).$$

If $A_j(u) \subseteq \operatorname{span} U(\boldsymbol{e})$, then $V(u_j) \subseteq \operatorname{span} U(\boldsymbol{e} + \boldsymbol{e}_j)$ and cannot yield a new leaf. Otherwise, the branch contains a new f_i^d. Hence each f_i^d is discovered exactly once, and the recursion terminates after all are found. In particular, there are r leaves.

Running time. Each recursive descent increases $\sum_j e_j$ by one, so at most $\sum_i \deg f_i \leqslant r\delta$ nodes are visited. Each node requires $\operatorname{poly}(m, r, \delta, d)$ linear-algebra operations, hence the total running time is $\operatorname{poly}(m, r, \delta, d)$, or $\operatorname{poly}(m, r, \delta, d, \log p)$ when $p > 0$. $\square$

4.3 The Univariate Reconstruction Algorithm

We now give the full reconstruction algorithm in the univariate case. In what follows we assume that the characteristic of $\mathbb{F}$ is either $p = 0$ or $p > 2rd\delta$. We further assume that $d > (r+1)^4\delta$.

Algorithm 2 RECOVER UNIVARIATE SUMS OF POWERS

```
Require: Black-box access to $f(x) \in \mathbb{F}[x]$; parameters $r, d, \delta$.
Ensure: Representation $f(x) = \sum_{i=1}^{r} \alpha_i f_i(x)^d$ with monic $f_i$.
  Interpolation. Since $\deg f \leqslant d\delta$, query $f$ at $d\delta + 1$ points and interpolate $f(x) = \sum_{i=0}^{\delta d} \beta_i x^i$.
  Perfect-power test. Decide if $f = \alpha g^d$ for some scalar $\alpha$ and polynomial $g$ with $\deg(g) \leqslant \delta$ (e.g., via repeated $\gcd(f, \nabla f)$ / squarefree decomposition). If yes, rescale $g$ to be monic and return $\{(\alpha_1, f_1)\} = \{(\alpha, g)\}$.
  Annihilating operator (normalized). Find the minimal $r' \leqslant r$ and polynomials $Q_0, \dots, Q_{r'}$ with $\deg Q_j \leqslant {r'}^2\delta$ such that $L = \sum_{j=0}^{r'} Q_j(x)\,\nabla^j$ satisfies $L(f) = 0$. Among all such solutions, choose one with minimal $\deg Q_{r'}$ and rescale so that $Q_{r'}$ is monic (as in Corollary 4.4).
  Factor. Using Theorem 2.13 or Theorem 2.14, compute the set $\{\phi_j\}$ of all irreducible factors of degree at most $\delta$ of $Q_{r'}(x)$. By Corollary 4.8, $\{\phi_j\}$ contains all irreducible factors of the $f_i$'s.
  Solution space. Compute a basis $\{h_1, \dots, h_{r'}\}$ of $V = \ker(L)$ by solving (11).
  Recover the $f_i^d$. Run Algorithm 1 on input $(\{h_i\}, \{\phi_j\})$ to obtain $U = \{f_1^d, \dots, f_{r'}^d\}$.
  Recover the coefficients $\alpha_i$. Solve the linear system (with unknown $\alpha_i$) $f(x) = \sum_{i=1}^{r} \alpha_i f_i^d$.
  return $\{(\alpha_i, f_i)\}_{i=1}^{r'}$.
```

Theorem 4.16 (Correctness and bit complexity of the univariate reconstruction). *Let $r, d, \delta \in \mathbb{N}$ with $d > (r+1)^4\delta$. Let $\mathbb{F}$ be a field of characteristic $p = 0$ or $p > 2rd\delta$.*

Suppose a nonzero univariate polynomial $f \in \mathbb{F}[x]$ admits a minimal representation

$$f(x) = \sum_{i=1}^{r'} \alpha_i\, f_i(x)^d, \qquad \deg(f_i) \leqslant \delta, \qquad r' \leqslant r,$$

where each f_i is monic, and the polynomials $\{f_i\}_{i=1}^{r'}$ are non-associate. Denote the total bit complexity of the representation with B.[9] *Then Algorithm 2 outputs, in time* $\mathrm{poly}(r, \delta, d, B)$ *(or* $\mathrm{poly}(r, \delta, d, p, \log|\mathbb{F}|)$ *if $\mathbb{F}$ is a finite field of characteristic p), a set $\{(\alpha'_i, f'_i)\}_{i=1}^{r'}$ such that, for some permutation π,*

$$f(x) = \sum_{i=1}^{r'} \alpha'_i f'_i(x)^d \qquad \text{and} \qquad f'_i = f_{\pi(i)}, \ \ \alpha'_i = \alpha_{\pi(i)}.$$

Proof. By assumption $\deg(f) \leqslant d\delta$, so interpolation is efficient. Similarly, since $p = 0$ or $p > 2rd\delta > \deg(f)$, testing whether f is a dth power can be done via repeated computation of $\gcd(f, \nabla f)$ (or by Theorem 2.13/Theorem 2.14).

Corollary 4.4 guarantees that we can compute the required operator L efficiently. Depending on the field, Theorem 2.13 and Theorem 2.14 allow us to compute all irreducible factors of $Q_{r'}$, yielding the set $\{\phi_j\}$; by Corollary 4.8 this set contains all irreducible factors of each f_i. A basis for $\ker(L)$ is obtained by solving (11). By Theorem 4.14, we can recover all f_i^d efficiently.

Since $r^2 < d$, Theorem 2.4 guarantees that the polynomials $\{f_i^d\}$ are non-associate. Hence, there is a unique solution to the linear system $f(x) = \sum_{i=1}^{r} \alpha_i f_i^d$. As we know $f, f_1, \ldots, f_r$ we can solve the system to recover $\alpha_1, \ldots, \alpha_r$.

Uniqueness (up to permutation) follows from Theorem 2.4 and the fact that the f_i are monic. □

5 Reconstruction of $\Sigma^{[r]} \wedge^{[d]} \Sigma^{[s]}\Pi^{[\delta]}$ Circuits

In this section we solve the reconstruction problem for $\Sigma^{[r]} \wedge^{[d]} \Sigma^{[s]}\Pi^{[\delta]}$ circuits: We are given black-box access to a polynomial

$$f(\mathbf{x}) = \sum_{i=1}^{r} \alpha_i f_i(\mathbf{x})^d \in \mathbb{F}[\mathbf{x}], \tag{13}$$

where each $f_i \in \mathbb{F}[\mathbf{x}]$ is an unknown s-sparse polynomial of degree $\leqslant \delta$. Furthermore, the f_i are non-associate. We also assume that $\mathbb{F}$ satisfies the requirements of Theorem 1.2.

Overview. The idea behind the algorithm is as follows. At a high level, we wish to reduce the problem to the univariate reconstruction problem, for which we have Algorithm 2. For that, we shall restrict our input to well chosen lines, run Algorithm 2 for the restricted polynomials, and reconstruct the original polynomials from that information.

In more detail, let $\mathcal{H}_{3.2}$ be the hitting set given in Theorem 3.2 for $\Sigma^{[2]}\wedge^{[d]}\Sigma^{[2s]}\Pi^{[\delta]}$ circuits, with parameter $\varepsilon_{3.2} = \frac{1}{2r^2}$. Let $\mathcal{H}_{2.7}$ be the robust interpolating set for n-variate, s-sparse polynomials of degree δ, given in Construction 2.7, for parameter $\varepsilon_{2.7} = \frac{1}{100nr^2}$.

For every $(\mathbf{u}, \mathbf{v}) \in \mathcal{H}_{3.2} \times \mathcal{H}_{2.7}$ we define the univariate polynomial

$$F_{\mathbf{u},\mathbf{v}}(t) = f(\mathbf{u} + t(\mathbf{v} - \mathbf{u})) = \sum_{i=1}^{r} \alpha_i f_i(\mathbf{u} + t(\mathbf{v} - \mathbf{u}))^d .$$

[9] As before, when $\mathbb{F}$ is a finite field, the bit complexity is $r'\delta \log|\mathbb{F}|$.

We will show that for most points $(\mathbf{u},\mathbf{v}) \in \mathcal{H}_{3.2} \times \mathcal{H}_{2.7}$, $F_{\mathbf{u},\mathbf{v}}$ cannot be represented as a sum of fewer than r powers. Therefore, Algorithm 2 will return for each "good" $(\mathbf{u},\mathbf{v})$ a list of r pairs $(\lambda_{i,\mathbf{u},\mathbf{v}}, h_{i,\mathbf{u},\mathbf{v}}(t))$ such that, up to reordering, $\lambda_{i,\mathbf{u},\mathbf{v}} h_{i,\mathbf{u},\mathbf{v}}(t)^d = \alpha_i f_i(\mathbf{u} + t(\mathbf{v} - \mathbf{u}))^d$. In particular, $\lambda_{i,\mathbf{u},\mathbf{v}} h_{i,\mathbf{u},\mathbf{v}}(1)^d = \alpha_i f_i(\mathbf{v})^d$. Thus, we have access to evaluations of f_i on enough points of $\mathcal{H}_{2.7}$. This will enable us to interpolate f_i and conclude the algorithm. An important point is that we will have to "align" the outputs of the algorithm when run on different $\mathbf{v} \in \mathcal{H}_{2.7}$ so that we know to map each output to a unique f_i. This will be done by noting that with high probability over the choice of u, the *labels* $\alpha_i f_i(\mathbf{u})^d = \lambda_{i,\mathbf{u},\mathbf{v}} h_{i,\mathbf{u},\mathbf{v}}(0)^d$ are distinct.

We next give the description of the algorithm and then its formal analysis.

Algorithm 3 Multivariate $\Sigma^{[r]} \wedge^{[d]} \Sigma^{[s]} \Pi^{[\delta]}$ Reconstruction

```
Require: Sets $\mathcal{H}_{3.2}, \mathcal{H}_{2.7}$; blackbox access to $f = \sum_{i=1}^{r} \alpha_i f_i^d$; parameters $r, \varepsilon, d, \delta$.
Ensure: Output $\{(\lambda_i, h_i)\}$, such that $\sum_{i=1}^{r} \lambda_i h_i(\mathbf{x})^d = f(\mathbf{x})$.
for all $(\mathbf{u},\mathbf{v}) \in \mathcal{H}_{3.2} \times \mathcal{H}_{2.7}$ do        ▷ Run univariate recovery for each $(\mathbf{u},\mathbf{v})$
    Let
        $F_{\mathbf{u},\mathbf{v}}(t) := f(\mathbf{u} + t(\mathbf{v} - \mathbf{u}))$ ,   and   $f_{i,\mathbf{u},\mathbf{v}}(t) := f_i(\mathbf{u} + t(\mathbf{v} - \mathbf{u}))$
    Run Algorithm 2 on $F_{\mathbf{u},\mathbf{v}}$
    if the algorithm did not abort then
        Let $\{(\lambda_{i,\mathbf{u},\mathbf{v}}, h_{i,\mathbf{u},\mathbf{v}}(t))\}_{i=1}^{r_{\mathbf{u},\mathbf{v}}}$ be its output
    end if
end for
$r \leftarrow \max_{\mathbf{u},\mathbf{v}} r_{\mathbf{u},\mathbf{v}}$
for all $\mathbf{u} \in \mathcal{H}_{3.2}$ do        ▷ Define good $\mathbf{v}$'s per $\mathbf{u}$
    $\mathcal{V}_\mathbf{u} \leftarrow \{\mathbf{v} \in \mathcal{H}_{2.7} \ : \ r = r_{\mathbf{u},\mathbf{v}}$ and the $r$ labels $\lambda_{i,\mathbf{u},\mathbf{v}} h_{i,\mathbf{u},\mathbf{v}}(0)^d$ are nonzero and distinct $\}$
end for
$\mathcal{U} \leftarrow \{\mathbf{u} : |\mathcal{V}_\mathbf{u}| \geqslant (1 - \varepsilon\binom{r}{2})|\mathcal{H}_{KS}|\}$        ▷ Define good $\mathbf{u}$'s
Fix some $\mathbf{u} \in \mathcal{U}$
for all $\mathbf{v} \in \mathcal{V}_\mathbf{u}$ do        ▷ Align indices across different $\mathbf{v}$'s
    Reorder $\{(\lambda_{i,\mathbf{u},\mathbf{v}}, h_{i,\mathbf{u},\mathbf{v}}(t))\}_{i=1}^{r}$ so that for every $\mathbf{v} \neq \mathbf{v}' \in \mathcal{V}_\mathbf{u}$,
        $\lambda_{i,\mathbf{u},\mathbf{v}} h_{i,\mathbf{u},\mathbf{v}}(0)^d = \lambda_{i,\mathbf{u},\mathbf{v}'} h_{i,\mathbf{u},\mathbf{v}'}(0)^d$ ,
    and denote $\lambda_{i,\mathbf{u}} := \lambda_{i,\mathbf{u},\mathbf{v}} h_{i,\mathbf{u},\mathbf{v}}(0)^d$
end for
for all $\mathbf{v} \in \mathcal{V}_\mathbf{u}$ do        ▷ Reconstruct the $g_i$
    for $i = 1$ to $r$ do
        $h_{i,\mathbf{u}}(\mathbf{v}) \leftarrow \frac{h_{i,\mathbf{u},\mathbf{v}}(1)}{h_{i,\mathbf{u},\mathbf{v}}(0)}$
    end for
end for
Apply Theorem 2.9 to reconstruct each $h_{i,\mathbf{u}}$ as a degree $\delta$ $s$-sparse polynomial
return the set of pairs $\{(\lambda_{i,\mathbf{u}}, h_{i,\mathbf{u}})\}$
```

Theorem 5.1. *Let* $r, d, \delta \in \mathbb{N}$ *with* $d > (r+1)^4\delta$. *Let* $\mathbb{F}$ *be a field of characteristic* $p = 0$

or $p \geqslant rd\delta(s^2n + \delta)$. *Then, Algorithm 3 runs in time* $\mathrm{poly}(n, s, d)$ *and its output satisfies* $\sum_{i=1}^{r} \lambda_{i,\mathbf{u}} h_{i,\mathbf{u}}(\mathbf{x})^d = f(\mathbf{x})$.

Proof. As the sets involved have polynomial size and each step requires polynomial time, the claim regarding the running time is clear. We next prove correctness.

The proof is composed of several claims. We shall first prove that $\mathcal{U}$ is non-empty and that for every $\mathbf{u} \in \mathcal{U}$, $|\mathcal{V}_\mathbf{u}| > (1 - \frac{1}{100n})|\mathcal{H}_{2.7}|$. We shall then show that the alignment process succeeds and that it gives access to evaluations of the different f_i. We then claim (using Theorem 2.9) that since $\mathcal{V}_\mathbf{u}$ is large and the alignment process succeeded, the g_i that we reconstructed are scalar multiples of the f_i.

Claim 5.2 (Many good $\mathbf{u}$'s)**.** *Let* $\mathcal{H}_{3.2}$ *be the hitting set as above. Then, all but at most* $\varepsilon_{3.2}\left(r + \binom{r}{2}\right)|\mathcal{H}_{3.2}| < \frac{1}{2}|\mathcal{H}_{3.2}|$ *points* $\mathbf{u} \in \mathcal{H}_{3.2}$ *satisfy the following simultaneously:*

1. $f_i(\mathbf{u}) \neq 0$ *for all* $i \in [r]$, *and*

2. *the values* $\Lambda_i := \alpha_i f_i(\mathbf{u})^d$ *are pairwise distinct.*

We call the points $\mathbf{u} \in \mathcal{H}_{3.2}$ *that satisfy both requirements* good.

Proof. Since each f_i is s-sparse, we get from Theorem 3.2 and the union bound that there are at most $\varepsilon_{3.2} r|\mathcal{H}_{3.2}|$, on which any of the f_i vanishes. Next, consider the $\binom{r}{2}$ polynomials

$$P_{i,j}(\mathbf{x}) := \alpha_i f_i(\mathbf{x})^d - \alpha_j f_j(\mathbf{x})^d.$$

Clearly, $P_{i,j}$ is a $\Sigma^{[2]}\wedge^{[d]}\Sigma^{[2s]}\Pi^{[\delta]}$ circuit. Theorem 3.2 and the union bound imply that there are at most $\varepsilon_{3.2}\binom{r}{2}|\mathcal{H}_{3.2}|$, on which any of the $P_{i,j}$ vanishes. Combining the two estimates, with our choice of $\varepsilon_{3.2} = 1/2r^2$, we conclude the proof. □

We next prove that for any such good $\mathbf{u}$, the set $\mathcal{V}_\mathbf{u}$ is large.

Claim 5.3 ($\mathcal{V}_\mathbf{u}$ is large for a good $\mathbf{u}$)**.** *Let* $\mathbf{u} \in \mathcal{H}_{3.2}$ *be a good point. It holds that* $|\mathcal{V}_\mathbf{u}| \geqslant (1 - \varepsilon_{2.7}\binom{r}{2})|\mathcal{H}_{2.7}|$ *and for every* $\mathbf{v} \in \mathcal{V}_\mathbf{u}$ *the following sets of labels are equal*

$$\left\{\lambda_{i,\mathbf{u},\mathbf{v}} h_{i,\mathbf{u},\mathbf{v}}(0)^d \mid i \in [r]\right\} = \left\{\alpha_i f_i(\mathbf{u})^d \mid i \in [r]\right\} .$$

Proof. For $1 \leqslant i < j \leqslant r$ denote

$$g_{i,j}(\mathbf{x}) \;:=\; f_j(\mathbf{u})\, f_i(\mathbf{x}) \;-\; f_i(\mathbf{u})\, f_j(\mathbf{x}) \;\in\; \mathbb{F}[\mathbf{x}].$$

Since $\mathbf{u}$ is good we have that $f_i(\mathbf{u}), f_j(\mathbf{u}) \neq 0$. As the f_i are non-associate, we conclude that $g_{i,j} \neq 0$. From Corollary 2.6 and the union bound we conclude that all but at most $\varepsilon_{2.7}\binom{r}{2}|\mathcal{H}_{2.7}|$ points $\mathbf{v} \in \mathcal{H}_{2.7}$ satisfy $g_{i,j}(\mathbf{v}) \neq 0$ for every $i < j$. Let $\mathbf{v}$ be such a point (on which no $g_{i,j}$ vanishes). We next show that $\mathbf{v} \in \mathcal{V}_\mathbf{u}$.

We first note that the univariate polynomials

$$f_{i,\mathbf{u},\mathbf{v}} := f_i(\mathbf{u} + t(\mathbf{v} - \mathbf{u}))$$

are non-associate. Indeed, if $f_{i,\mathbf{u},\mathbf{v}} \sim f_{j,\mathbf{u},\mathbf{v}}$ then this would give

$$\frac{f_i(\mathbf{v})}{f_i(\mathbf{u})} = \frac{f_{i,\mathbf{u},\mathbf{v}}(1)}{f_{i,\mathbf{u},\mathbf{v}}(0)} = \frac{f_{j,\mathbf{u},\mathbf{v}}(1)}{f_{j,\mathbf{u},\mathbf{v}}(0)} = \frac{f_j(\mathbf{v})}{f_j(\mathbf{u})} ,$$

implying $g_{i,j}(\mathbf{v}) = 0$, in contradiction. Consequently the representation

$$F_{\mathbf{u},\mathbf{v}}(t) = \sum_{i=1}^{r} \alpha_i \, f_{i,\mathbf{u},\mathbf{v}}(t)^d$$

has minimal top fan-in exactly r, so Algorithm 2 returns exactly r polynomials. Since the representation is unique we have that up to reordering, $\lambda_{i,\mathbf{u},\mathbf{v}} h_{i,\mathbf{u},\mathbf{v}}(t)^d = \alpha_i f_{i,\mathbf{u},\mathbf{v}}(t)^d$. Consequently,

$$\lambda_{i,\mathbf{u},\mathbf{v}} h_{i,\mathbf{u},\mathbf{v}}(0)^d = \alpha_i f_{i,\mathbf{u},\mathbf{v}}(0)^d = \alpha_i f_i(\mathbf{u})^d \, .$$

This concludes the proof regarding the size of $\mathcal{V}_{\mathbf{u}}$.

Now, observe that for any $\mathbf{v} \in \mathcal{V}_{\mathbf{u}}$, since Algorithm 2 returned exactly r polynomials, it must be the case that the $f_{i,\mathbf{u},\mathbf{v}}$ are linearly independent, and the same argument implies that $\left\{\lambda_{i,\mathbf{u},\mathbf{v}} h_{i,\mathbf{u},\mathbf{v}}(0)^d \mid i \in [r]\right\} = \left\{\alpha_i f_i(\mathbf{u})^d \mid i \in [r]\right\}$. □

Combining Claim 5.2 and Claim 5.3, we conclude that the set $\mathcal{U}$ is not empty (indeed, it is quite large) and that for every $\mathbf{u} \in \mathcal{U}$, for each $\mathbf{v} \in \mathcal{V}_{\mathbf{u}}$ the set of labels equals $\left\{\alpha_i f_i(\mathbf{u})^d \mid i \in [r]\right\}$. In particular, we can align the labels $\lambda_{i,\mathbf{u},\mathbf{v}} h_{i,\mathbf{u},\mathbf{v}}(0)^d$ across different $v \in \mathcal{V}_{\mathbf{u}}$. Thus, we can assume without loss of generality that for every $v \in \mathcal{V}_{\mathbf{u}}$ it holds that

$$\lambda_{i,\mathbf{u},\mathbf{v}} h_{i,\mathbf{u},\mathbf{v}}(0)^d = \alpha_i f_{i,\mathbf{u},\mathbf{v}}(0)^d = \alpha_i f_i(\mathbf{u})^d \, .$$

Hence, by uniqueness of representation as small sum of d-th powers (Theorem 2.4), we have that for all $\mathbf{v} \in \mathcal{V}_{\mathbf{u}}$ it holds that

$$\lambda_{i,\mathbf{u},\mathbf{v}} h_{i,\mathbf{u},\mathbf{v}}(t)^d = \alpha_i f_{i,\mathbf{u},\mathbf{v}}(t)^d \, .$$

Recall that Algorithm 2 returns monic polynomials, so each $h_{i,\mathbf{u},\mathbf{v}}$ is monic. As the leading coefficient of t in $f_{i,\mathbf{u},\mathbf{v}}(t)$ equals $f_i(\mathbf{v} - \mathbf{u})$ we have that $\lambda_{i,\mathbf{u},\mathbf{v}} = \alpha_i f_i(\mathbf{v} - \mathbf{u})^d$. From this we obtain that $\lambda_{i,\mathbf{u},\mathbf{v}} = \lambda_{i,\mathbf{u},\mathbf{v}'}$ for all $\mathbf{v}, \mathbf{v}' \in \mathcal{V}_{\mathbf{u}}$. We can thus rename $\lambda_{i,\mathbf{u},\mathbf{v}}$ to $\lambda_{i,\mathbf{u}}$. Note that the polynomial $\tilde{h}_{i,\mathbf{u},\mathbf{v}}(t) := h_{i,\mathbf{u},\mathbf{v}}(t)/h_{i,\mathbf{u},\mathbf{v}}(0)$ satisfies $\tilde{h}_{i,\mathbf{u},\mathbf{v}}(0) = 1$. Moreover,

$$\tilde{h}_{i,\mathbf{u},\mathbf{v}}(t)^d = \left(\frac{h_{i,\mathbf{u},\mathbf{v}}(t)}{h_{i,\mathbf{u},\mathbf{v}}(0)}\right)^d = \left(\frac{f_{i,\mathbf{u},\mathbf{v}}(t)}{f_i(\mathbf{u})}\right)^d$$

Since $f_{i,\mathbf{u},\mathbf{v}}(0)/f_i(\mathbf{u}) = 1$, we obtain that $\tilde{h}_{i,\mathbf{u},\mathbf{v}}(t) = f_{i,\mathbf{u},\mathbf{v}}(t)/f_i(\mathbf{u})$. I.e., it is the unique d-th root of the polynomial $\left(\frac{f_{i,\mathbf{u},\mathbf{v}}(t)}{f_i(\mathbf{u})}\right)^d$ whose value at 0 equals 1 (other roots evaluate to different roots of unity). In particular, $\tilde{h}_{i,\mathbf{u},\mathbf{v}}(1) = \frac{f_{i,\mathbf{u},\mathbf{v}}(1)}{f_i(\mathbf{u})} = \frac{f_i(\mathbf{v})}{f_i(\mathbf{u})}$. Thus, the function $h_{i,\mathbf{u}}$ defined in Step 19 satisfies that for all $\mathbf{v} \in \mathcal{V}_{\mathbf{u}}$,

$$h_{i,\mathbf{u}}(\mathbf{v}) = \frac{h_{i,\mathbf{u},\mathbf{v}}(1)}{h_{i,\mathbf{u},\mathbf{v}}(0)} = \tilde{h}_{i,\mathbf{u},\mathbf{v}}(1) = f_i(\mathbf{v})/f_i(\mathbf{u}) \, .$$

Since f_i is an s-sparse polynomial of degree δ, and we have its evaluations on all $\mathbf{v} \in \mathcal{V}_{\mathbf{u}}$ which consists of at least $|\mathcal{V}_{\mathbf{u}}| \geqslant (1 - \varepsilon_{2.7}\binom{r}{2})|\mathcal{H}_{2.7}| \geqslant (1 - \frac{1}{100n})|\mathcal{H}_{2.7}|$ points of $\mathcal{H}_{2.7}$ we conclude by Theorem 2.9 that Step 22 returns the polynomial $f_i(\mathbf{v})/f_i(\mathbf{u})$.

Observe that if β_i satisfies

$$\beta_i \left(\frac{f_i(\mathbf{v})}{f_i(\mathbf{u})} \right)^d = \alpha_i f_i(\mathbf{v})^d \, ,$$

then

$$\beta_i = \alpha_i f_i(\mathbf{u})^d = \lambda_{i,\mathbf{u},\mathbf{v}} h_{i,\mathbf{u},\mathbf{v}}(0)^d = \lambda_{i,\mathbf{u}} \, .$$

Thus, the algorithm indeed returns $(\lambda_{i,\mathbf{u}}, h_{i,\mathbf{u}})$ satisfying

$$\lambda_{i,\mathbf{u}} h_{i,\mathbf{u}}(\mathbf{x})^d = \alpha_i f_i(\mathbf{x})^d \, .$$

This concludes the analysis of Algorithm 3. □

Bibliography

[ABG+14] Joseph Anderson, Mikhail Belkin, Navin Goyal, Luis Rademacher, and James Voss. The more, the merrier: the blessing of dimensionality for learning large gaussian mixtures. In *Conference on Learning Theory*, pages 1135–1164. PMLR, 2014. 2

[AF22] Robert Andrews and Michael A Forbes. Ideals, determinants, and straightening: proving and using lower bounds for polynomial ideals. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing*, pages 389–402, 2022. 3

[AGH+14] Animashree Anandkumar, Rong Ge, Daniel J Hsu, Sham M Kakade, Matus Telgarsky, et al. Tensor decompositions for learning latent variable models. *J. Mach. Learn. Res.*, 15(1):2773–2832, 2014. 2

[ASSS16] Manindra Agrawal, Chandan Saha, Ramprasad Saptharishi, and Nitin Saxena. Jacobian hits circuits: Hitting sets, lower bounds for depth-d occur-k formulas and depth-3 transcendence degree-k circuits. *SIAM J. Comput.*, 45(4):1533–1562, 2016. 4

[AV08] Manindra Agrawal and V. Vinay. Arithmetic circuits: A chasm at depth four. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2008, Philadelphia, PA, USA, October 25-28, 2008*, pages 67–75. IEEE Computer Society, 2008. 1

[BBB+00] Amos Beimel, Francesco Bergadano, Nader H Bshouty, Eyal Kushilevitz, and Stefano Varricchio. Learning functions represented as multiplicity automata. *Journal of the ACM (JACM)*, 47(3):506–530, 2000. 1

[BESV24] Aditya Bhaskara, Eric Evert, Vaidehi Srinivas, and Aravindan Vijayaraghavan. New tools for smoothed analysis: Least singular value bounds for random matrices with dependent entries. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 375–386, 2024. 2

[BGKS22] Vishwas Bhargava, Ankit Garg, Neeraj Kayal, and Chandan Saha. Learning generalized depth three arithmetic circuits in the non-degenerate case. In Amit Chakrabarti and Chaitanya Swamy, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2022, September 19-21, 2022, University of Illinois, Urbana-Champaign, USA (Virtual Conference)*, volume 245 of *LIPIcs*, pages 21:1–21:22. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. 1

[BHKX22] Mitali Bafna, Jun-Ting Hsieh, Pravesh K Kothari, and Jeff Xu. Polynomial-time power-sum decomposition of polynomials. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 956–967. IEEE, 2022. 1, 2, 5

[BS25] Vishwas Bhargava and Devansh Shringi. Faster & deterministic FPT algorithm for worst-case tensor decomposition. In Keren Censor-Hillel, Fabrizio Grandoni, Joël Ouaknine, and Gabriele Puppis, editors, *52nd International Colloquium on Automata, Languages, and Programming, ICALP 2025, July 8-11, 2025, Aarhus, Denmark*, volume 334 of *LIPIcs*, pages 28:1–28:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2025. 1, 5

[BSV21] Vishwas Bhargava, Shubhangi Saraf, and Ilya Volkovich. Reconstruction algorithms for low-rank tensors and depth-3 multilinear circuits. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 809–822. ACM, 2021. 1, 5

[BSV25] Vishwas Bhargava, Shubhangi Saraf, and Ilya Volkovich. Reconstruction of depth-4 multilinear circuits. *ACM Trans. Comput. Theory*, 17(3):17:1–17:23, 2025. 1

[BT88] Michael Ben-Or and Prasoon Tiwari. A deterministic algorithm for sparse multivariate polynomial interpolation (extended abstract). In Janos Simon, editor, *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 301–309. ACM, 1988. 3

[CGK$^+$24] Pritam Chandra, Ankit Garg, Neeraj Kayal, Kunal Mittal, and Tanmay Sinha. Learning arithmetic formulas in the presence of noise: A general framework and applications to unsupervised learning. In Venkatesan Guruswami, editor, *15th Innovations in Theoretical Computer Science Conference, ITCS 2024, January 30 to February 2, 2024, Berkeley, CA, USA*, volume 287 of *LIPIcs*, pages 25:1–25:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024. 1, 2, 5

[CGS23] Suryajith Chillara, Coral Grichener, and Amir Shpilka. On hardness of testing equivalence to sparse polynomials under shifts. In Petra Berenbrink, Patricia Bouyer, Anuj Dawar, and Mamadou Moustapha Kanté, editors, *40th International Symposium on Theoretical Aspects of Computer Science, STACS 2023, Hamburg, Germany, March 7-9, 2023*, volume 254 of *LIPIcs*, pages 22:1–22:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. 2

[CKS19] Chi-Ning Chou, Mrinal Kumar, and Noam Solomon. Closure results for polynomial factorization. *Theory Comput.*, 15:1–34, 2019. 1

[dD23] Philipp J di Dio. The multidimensional truncated moment problem: Gaussian mixture reconstruction from derivatives of moments. *Journal of Mathematical Analysis and Applications*, 517(1):126592, 2023. 2

[DDS21] Pranjal Dutta, Prateek Dwivedi, and Nitin Saxena. Deterministic identity testing paradigms for bounded top-fanin depth-4 circuits. In Valentine Kabanets, editor, *36th Computational Complexity Conference, CCC 2021, July 20-23, 2021, Toronto, Ontario, Canada (Virtual Conference)*, volume 200 of *LIPIcs*, pages 11:1–11:27. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. 4

[DG24] Pranjal Dutta and Sumanta Ghosh. Sigact news complexity theory column 121. *SIGACT News*, 55(2):53–88, June 2024. 1

[DL78] Richard A. DeMillo and Richard J. Lipton. A probabilistic remark on algebraic program testing. *Inf. Process. Lett.*, 7(4):193–195, 1978. 1

[DS07a] Sanjoy Dasgupta and Leonard Schulman. A probabilistic analysis of em for mixtures of separated, spherical gaussians. *Journal of Machine Learning Research*, 8(2), 2007. 2

[DS07b] 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. 4

[DSY09] Zeev Dvir, Amir Shpilka, and Amir Yehudayoff. Hardness-randomness tradeoffs for bounded depth arithmetic circuits. *SIAM J. Comput.*, 39(4):1279–1293, 2009. 1

[FK09] Lance Fortnow and Adam R. Klivans. Efficient learning algorithms yield circuit lower bounds. *Journal of Computer and System Sciences*, 75(1):27–36, 2009. 1

[For15] Michael A. Forbes. Deterministic divisibility testing via shifted partial derivatives. In Venkatesan Guruswami, editor, *IEEE 56th Annual Symposium on Foundations of Computer Science, FOCS 2015, Berkeley, CA, USA, 17-20 October, 2015*, pages 451–465. IEEE Computer Society, 2015. 4

[FS12] Michael A. Forbes and Amir Shpilka. On identity testing of tensors, low-rank recovery and compressed sensing. In Howard J. Karloff and Toniann Pitassi, editors, *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 163–172. ACM, 2012. 1

[FSS14] Michael A Forbes, Ramprasad Saptharishi, and Amir Shpilka. Hitting sets for multilinear read-once algebraic branching programs, in any order. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 867–875, 2014. 4

[GdOS25] Abhibhav Garg, Rafael Mendes de Oliveira, and Akash Kumar Sengupta. Rank bounds and PIT for $\sigma^3 \pi \sigma \pi^d$ circuits via a nonlinear edelstein-kelly theorem. *Electron. Colloquium Comput. Complex.*, TR25-051, 2025. 4

[GG20] Zeyu Guo and Rohit Gurjar. Improved explicit hitting-sets for roabps. In Jaroslaw Byrka and Raghu Meka, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2020, August 17-19, 2020, Virtual Conference*, volume 176 of *LIPIcs*, pages 4:1–4:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020. 4

[GHK15] Rong Ge, Qingqing Huang, and Sham M. Kakade. Learning mixtures of gaussians in high dimensions. In Rocco A. Servedio and Ronitt Rubinfeld, editors, *Proceedings of the Forty-Seventh Annual ACM on Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14-17, 2015*, pages 761–770. ACM, 2015. 2

[GKKS16] Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Arithmetic circuits: A chasm at depth 3. *SIAM J. Comput.*, 45(3):1064–1079, 2016. 1

[GKL12] Ankit Gupta, Neeraj Kayal, and Satyanarayana V. Lokam. Reconstruction of depth-4 multilinear circuits with top fan-in 2. In Howard J. Karloff and Toniann Pitassi, editors, *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 625–642. ACM, 2012. 1

[GKQ14] Ankit Gupta, Neeraj Kayal, and Youming Qiao. Random arithmetic formulas can be reconstructed efficiently. *Comput. Complex.*, 23(2):207–303, 2014. 1

[GKS90] Dima Grigoriev, Marek Karpinski, and Michael F. Singer. Fast parallel algorithms for sparse multivariate polynomial interpolation over finite fields. *SIAM J. Comput.*, 19(6):1059–1063, 1990. 3

[GKS16] Rohit Gurjar, Arpita Korwar, and Nitin Saxena. Identity testing for constant-width, and any-order, read-once oblivious arithmetic branching programs. *arXiv preprint arXiv:1601.08031*, 2016. 4

[GKS20] 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, 2020. 1, 2, 5

[GKSS22] Zeyu Guo, Mrinal Kumar, Ramprasad Saptharishi, and Noam Solomon. Derandomization from algebraic hardness. *SIAM J. Comput.*, 51(2):315–335, 2022. 1

[GVX14] Navin Goyal, Santosh Vempala, and Ying Xiao. Fourier pca and robust tensor decomposition. In *Proceedings of the forty-sixth annual ACM symposium on Theory of computing*, pages 584–593, 2014. 2

[GW25] Zeyu Guo and Siki Wang. Deterministic depth-4 PIT and normalization. *CoRR*, abs/2504.15143, 2025. 4

[Hås90] Johan Håstad. Tensor rank is np-complete. *J. Algorithms*, 11(4):644–654, 1990. 1

[HS80] Joos Heintz and Claus-Peter Schnorr. Testing polynomials which are easy to compute. In *Proceedings of the twelfth annual ACM Symposium on Theory of Computing*, pages 262–272, 1980. 1

[HSS15] Samuel B Hopkins, Jonathan Shi, and David Steurer. Tensor principal component analysis via sum-of-square proofs. In *Conference on Learning Theory*, pages 956–1006. PMLR, 2015. 2

[HSSS16] Samuel B Hopkins, Tselil Schramm, Jonathan Shi, and David Steurer. Fast spectral algorithms from sum-of-squares proofs: tensor decomposition and planted sparse vectors. In *Proceedings of the forty-eighth annual ACM symposium on Theory of Computing*, pages 178–191, 2016. 2

[Kay12] Neeraj Kayal. Affine projections of polynomials: extended abstract. In *Proceedings of the Forty-Fourth Annual ACM Symposium on Theory of Computing*, STOC '12, page 643–662, New York, NY, USA, 2012. Association for Computing Machinery. 4

[KI04] Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Computational Complexity*, 13(1-2):1–46, 2004. 1

[KMSV13] Zohar S. Karnin, Partha Mukhopadhyay, Amir Shpilka, and Ilya Volkovich. Deterministic identity testing of depth-4 multilinear circuits with bounded top fan-in. *SIAM J. Comput.*, 42(6):2114–2131, 2013. 4

[Koi12] Pascal Koiran. Arithmetic circuits: The chasm at depth four gets wider. *Theor. Comput. Sci.*, 448:56–65, 2012. 1

[KS01] Adam R. Klivans and Daniel Spielman. Randomness efficient identity testing of multivariate polynomials. In *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing*, STOC '01, page 216–223, New York, NY, USA, 2001. Association for Computing Machinery. 3, 7, 10

[KS06] Adam Klivans and Amir Shpilka. Learning restricted models of arithmetic circuits. *Theory of computing*, 2(1):185–206, 2006. 1

[KS07] Neeraj Kayal and Nitin Saxena. Polynomial identity testing for depth 3 circuits. *Computational Complexity*, 16(2):115–138, 2007. 4

[KS09a] Zohar S. 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. IEEE Computer Society, 2009. 1, 4

[KS09b] Neeraj Kayal and Shubhangi Saraf. Blackbox polynomial identity testing for depth 3 circuits. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, October 25-27, 2009, Atlanta, Georgia, USA*, pages 198–207. IEEE Computer Society, 2009. 4

[KS09c] Adam R. Klivans and Alexander A. Sherstov. Cryptographic hardness for learning intersections of halfspaces. *Journal of Computer and System Sciences*, 75(1):2–12, 2009. 1

[KS11] Zohar S. 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. 4

[KS19] Mrinal Kumar and Ramprasad Saptharishi. Hardness-randomness tradeoffs for algebraic computation. *Bulletin of EATCS*, 3(129), 2019. 1

[KST23] Mrinal Kumar, Ramprasad Saptharishi, and Anamay Tengse. Near-optimal bootstrapping of hitting sets for algebraic models. *Theory Comput.*, 19:1–30, 2023. 1

[KX25] Pravesh K Kothari and Jeff Xu. Smooth trade-off for tensor pca via sharp bounds for kikuchi matrices. *arXiv preprint arXiv:2510.03061*, 2025. 2

[LLL82] H.W. jr. Lenstra, A.K. Lenstra, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:515–534, 1982. 15

[LM21] Allen Liu and Ankur Moitra. Settling the robust learnability of mixtures of gaussians. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 518–531, 2021. 2

[LST25] Nutan Limaye, Srikanth Srinivasan, and Sébastien Tavenas. Superpolynomial lower bounds against low-depth algebraic circuits. *Journal of the ACM*, 72(4):1–35, 2025. 3

[Mas84] Richard C. Mason. *Diophantine Equations over Function Fields*. London Mathematical Society Lecture Note Series. Cambridge University Press, 1984. 5, 7

[MR14] Andrea Montanari and Emile Richard. A statistical model for tensor pca. *Advances in neural information processing systems*, 27, 2014. 2

[MSS16] Tengyu Ma, Jonathan Shi, and David Steurer. Polynomial-time tensor decompositions with sum-of-squares. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 438–446. IEEE, 2016. 2

[Ore22] Øystein Ore. Über höhere kongruenzen. *Norske Videnskaps-Akademi i Oslo. Forhandlinger (Proceedings of the Norwegian Academy of Science and Letters)*, 1922(12):1–8, 1922. Contains the original root bound for nonzero multivariate polynomials over finite fields, later used in polynomial identity testing (Schwartz–Zippel lemma). 1

[PS21] Shir Peleg and Amir Shpilka. Polynomial time deterministic identity testing algorithm for σ [3] πσπ [2] circuits via edelstein–kelly type theorem for quadratic polynomials. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 259–271, 2021. 4

[PSV24] Shir Peleg, Amir Shpilka, and Ben Lee Volk. Tensor Reconstruction Beyond Constant Rank. In Venkatesan Guruswami, editor, *15th Innovations in Theoretical Computer Science Conference (ITCS 2024)*, volume 287 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 87:1–87:20, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. 1, 5

[RS05] Ran Raz and Amir Shpilka. Deterministic polynomial identity testing in non-commutative models. *Computational Complexity*, 14(1):1–19, 2005. 4

[RV17] Oded Regev and Aravindan Vijayaraghavan. On learning mixtures of well-separated gaussians. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 85–96. IEEE, 2017. 2

[Sax08] Nitin Saxena. Diagonal circuit identity testing and lower bounds. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part I: Tack A: Algorithms, Automata, Complexity, and Games*, volume 5125 of *Lecture Notes in Computer Science*, pages 60–71. Springer, 2008. 4

[Sax09] Nitin Saxena. Progress on polynomial identity testing. *Bull. EATCS*, 99:49–79, 2009. 1

[Sax14] Nitin Saxena. Progress on polynomial identity testing-ii. In *Perspectives in Computational Complexity: The Somenath Biswas Anniversary Volume*, pages 131–146. Springer, 2014. 1

[Sch80] Jacob T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, 1980. 1

[Shp09] Amir Shpilka. Interpolation of depth-3 arithmetic circuits with two multiplication gates. *SIAM J. Comput.*, 38(6):2130–2161, 2009. 1, 4

[Sin16] Gaurav Sinha. Reconstruction of real depth-3 circuits with top fan-in 2. In Ran Raz, editor, *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, volume 50 of *LIPIcs*, pages 31:1–31:53. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. 1, 4

[Sin22] Gaurav Sinha. Efficient reconstruction of depth three arithmetic circuits with top fan-in two. In Mark Braverman, editor, *13th Innovations in Theoretical Computer Science Conference, ITCS 2022, January 31 - February 3, 2022, Berkeley, CA, USA*, volume 215 of *LIPIcs*, pages 118:1–118:33. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022. 1, 4

[SK01] Arora Sanjeev and Ravi Kannan. Learning mixtures of arbitrary gaussians. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 247–257, 2001. 2

[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. 4

[SS13] Nitin Saxena and Comandur Seshadhri. From sylvester-gallai configurations to rank bounds: Improved blackbox identity test for depth-3 circuits. *J. ACM*, 60(5):33, 2013. 4

[SŠ18] Marcus Schaefer and Daniel Štefankovič. The complexity of tensor rank. *Theory of Computing Systems*, 62(5):1161–1174, 2018. 1

[SS25] Shubhangi Saraf and Devansh Shringi. Reconstruction of depth 3 arithmetic circuits with top fan-in 3. In *40th Computational Complexity Conference (CCC 2025)*, pages 21–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2025. 4

[SSV25] Shubhangi Saraf, Devansh Shringi, and Narmada Varadarajan. Reconstruction of depth-3 arithmetic circuits with constant top fan-in. *Electron. Colloquium Comput. Complex.*, TR25-222, 2025. 1, 4

[Sto81] Walter W. Stothers. Polynomial identities and Hauptmoduln. *The Quarterly Journal of Mathematics*, 32(3):349–370, 1981. 5, 7

[SV14] Amir Shpilka and Ilya Volkovich. On reconstruction and testing of read-once formulas. *Theory Comput.*, 10:465–514, 2014. 1

[SV18] Shubhangi Saraf and Ilya Volkovich. Black-box identity testing of depth-4 multilinear circuits. *Comb.*, 38(5):1205–1238, 2018. 4

[Swe18] Joseph Swernofsky. Tensor rank is hard to approximate. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2018)*, pages 26–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2018. 1

[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. 1

[Syl51] James Joseph Sylvester. Lx. on a remarkable discovery in the theory of canonical forms and of hyperdeterminants. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(12):391–410, 1851. 4

[Tav15] Sébastien Tavenas. Improved bounds for reduction to depth 4 and depth 3. *Inf. Comput.*, 240:2–11, 2015. 1

[Tes12] Gerald Teschl. *Ordinary Differential Equations and Dynamical Systems*, volume 140 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2012. 14

[VW03] Leonid N Vaserstein and Ethel R Wheland. Vanishing polynomial sums. *Communications in Algebra*, 31(2):751–772, 2003. 5, 8, 9

[vzGS92] Joachim von zur Gathen and Victor Shoup. Computing Frobenius maps and factoring polynomials. *Comput. Complexity*, 2(3):187–224, 1992. 15

[Zip79] Richard Zippel. Probabilistic algorithms for sparse polynomials. In Edward W. Ng, editor, *Symbolic and Algebraic Computation, EUROSAM '79, An International Symposiumon Symbolic and Algebraic Computation, Marseille, France, June 1979, Proceedings*, volume 72 of *Lecture Notes in Computer Science*, pages 216–226. Springer, 1979. 1

A Linear Independence and Wronskian

We prove Theorem 2.11. We recall its statement.

Theorem 2.11. *Let $g_1, \dots, g_n$ be polynomials over a field $\mathbb{F}$ of characteristic p. If the maximum degree of any g_i is d and $p = 0$ or $p > d$, then $g_1, \dots, g_n$ are linearly dependent over $\mathbb{F}$ if and only if their Wronskian $W(g_1, \dots, g_n)$ is identically zero.*

Proof. The 'if' direction is immediate. For the converse, assume that $g_1, \dots, g_n$ are linearly independent. Since the determinant is invariant under elementary column operations, we may assume, without loss of generality, that $g_1, \dots, g_n$ have distinct degrees.

Let their degrees be $d_1 > d_2 > \cdots > d_n \geqslant 0$, and write $g_j = c_j x^{d_j} +$ (lower terms) with $c_j \neq 0$. By multilinearity of the determinant, the leading term of the Wronskian equals the Wronskian of the leading monomials, provided its coefficient is nonzero:

$$W(\mathrm{LM}(g_1), \dots, \mathrm{LM}(g_n)) = W(c_1 x^{d_1}, \dots, c_n x^{d_n}).$$

The determinant of the matrix of derivatives of $c_j x^{d_j}$ is

$$\det\bigl(c_j (d_j)_i \, x^{d_j - i}\bigr)_{\substack{i \in [[n-1]], \\ j \in [n]}}$$

where $(d_j)_i = d_j(d_j-1)\cdots(d_j-i+1)$ is the falling factorial (defined to be 1 for $i=0$). Factoring out c_j from each column and collecting powers of x yields

$$W(c_1x^{d_1},\dots,c_nx^{d_n}) = \left(\Pi_{j=1}^{n} c_j\right)\left(\Pi_{1\leqslant i<j\leqslant n}(d_j-d_i)\right)\left(\Pi_{j=1}^{n}(d_j)_{n-1}\right)x^{\sum_j d_j-\binom{n}{2}}.$$

Because $p=0$ or $p>d$, all integers d_j and differences (d_j-d_i) are nonzero in $\mathbb{F}$, and each falling factorial $(d_j)_{n-1}$ is nonzero. Since each $c_j \neq 0$, the product above is nonzero, so $W(g_1,\dots,g_n)$ is not identically zero. □

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il