On Solving Sparse Polynomial Factorization Related Problems

Pranav Bisht
Computer Science Department, Boston College, Chestnut Hill, MA, USA

Ilya Volkovich
Computer Science Department, Boston College, Chestnut Hill, MA, USA

Abstract

In a recent result of Bhargava, Saraf and Volkovich [FOCS'18; JACM'20], the first factor sparsity bound for constant individual degree polynomials was shown. In particular, it was shown that any factor of a polynomial with at most s terms and individual degree bounded by d can itself have at most $s^{O(d^2 \log n)}$ terms. It is conjectured, though, that the "true" sparsity bound should be polynomial (i.e. $s^{\mathrm{poly}(d)}$). In this paper we provide supporting evidence for this conjecture by presenting polynomial-time algorithms for several problems that would be implied by a polynomial-size sparsity bound. In particular, we give efficient (deterministic) algorithms for identity testing of $\Sigma^{[2]}\Pi\Sigma\Pi^{[\text{ind-deg } d]}$ circuits and testing if a sparse polynomial is an exact power. Hence, our algorithms rely on different techniques.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory; Theory of computation → Pseudorandomness and derandomization

Keywords and phrases Sparse Polynomials, Identity Testing, Derandomization, Factor-Sparsity, Multivariate Polynomial Factorization

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2022.10

Related Version *Full Version*: `https://eccc.weizmann.ac.il/report/2022/070/` [6]

Acknowledgements The authors would like to thank the anonymous referees for their detailed comments and suggestions on the previous version of the paper.

1 Introduction

Polynomial Factorization is one of the core problems in algebraic complexity: given a multivariate polynomial $f \in \mathbb{F}[x_1, x_2 \ldots, x_n]$ over a field $\mathbb{F}$, output all its irreducible factors. In addition to being a natural problem, its importance is highlighted by various applications such as: list decoding [28, 16], derandomization [17], cryptography [7] and others. In the seminal works of [18, 19], efficient randomized factorization algorithms were presented. Yet, coming up with an efficient *deterministic* factorization algorithm remains a long-standing open question.

Indeed, one aspect of the computational problem is the representation of the input polynomial. One natural way to represent a polynomial is by listing all its terms and coefficients. This is known as *dense* representation. Yet, even if the individual degree of every variable is bounded by a small constant d, the total number of terms can be exponentially large, reaching $(d+1)^n$. Nonetheless, in many applications [31, 13, 3, 15] the actual number of non-zero terms in a polynomial is much smaller - $\mathrm{poly}(n)$. Such polynomials are referred to as *sparse* polynomials, which will be the focus of our paper.

A key question that precedes the design of efficient factorization algorithms for sparse polynomial is whether a factor of a sparse polynomial is (itself) sparse. Indeed, this question was first studied by von zur Gathen and Kaltofen in [13] that gave a randomized factorization algorithm where the runtime depends on the number of terms in the output factors. In the

same paper they provided an example inspired by geometric series (see below) of a family of polynomials that have factors with a super-polynomial (quasi-polynomial) number of terms. We denote by $\|f\|$ the *sparsity* of f. That is, the number of non-zero terms in f.

▶ **Example 1** ([13]). Let $n \geq 1$. Consider the polynomial $f(\overline{x}) = \prod_{i\in[n]} (x_i^n - 1)$ which can be written as a product of $g(\overline{x}) = \prod_{i\in[n]} (1 + x_i + \ldots + x_i^{n-1})$ and $h(\overline{x}) = \prod_{i\in[n]} (x_i - 1)$.
Observe that $\|f\| = \|h\| = 2^n$ while $\|g\| = n^n$, resulting in a quasi-polynomial blow-up[1].

Furthermore, for fields with finite characteristics the blow-up can be significantly larger:

▶ **Example 2** ([29]). For a prime p, let $f \in \mathbb{F}_p[x_1, \ldots x_n]$, and let $0 < d < p$. Consider: $f(\overline{x}) = (x_1 + x_2 + \ldots + x_n)^p = x_1^p + x_2^p + \ldots + x_n^p$, $g(\overline{x}) = (x_1 + x_2 + \ldots + x_n)^d$. Notice that g is a factor of f, but $\|f\| = n$ and $\|g\| = \binom{n+d-1}{d} = n^{\Omega(d)}$.

Based on the above, we should first try to obtain a "sparsity-bound" on factors of sparse polynomials with constant (i.e. bounded) individual degree. More formally, for some fixed d, we require that $\deg_{x_i} \leq d$, for all variables x_i. The simplest case (when $d = 1$) corresponds to the so-called *multilinear* polynomials. In [26], it was shown that a factor of an s-sparse[2] multilinear polynomial is itself s-sparse. Subsequently, in [30], this result was extended to the case of *multiquadratic* polynomials (i.e. when $d = 2$). In a recent work of [4], a quasi-polynomial-size sparsity bound was given for *any* fixed d. Specifically, it was shown that a factor of an s-sparse polynomial with individual degree bounded by d is $s^{O(d^2 \log n)}$-sparse. In addition, [4] designed a factorization algorithm whose runtime is efficient in terms of the sparsity bound. As a result they obtained a deterministic quasi-polynomial-time factorization algorithm for sparse polynomials with bounded individual degree. In the same paper it was also conjectured that the "true" sparsity bound should be polynomial rather than quasi-polynomial. More formally:

▶ **Conjecture 3.** *There exists a universal constant $k \in \mathbb{N}$ such that for any $s, d \in \mathbb{N}$, any factor of an s-sparse polynomial with individual degree bounded by d has at most s^{d^k} terms.*

In this paper we provide supporting evidence for this conjecture by presenting deterministic polynomial-time algorithms for some problems that reduce to sparse polynomial factorization. It is to be noted that invoking the aforementioned factorization algorithm of [4] with a polynomial-size sparsity bound would imply a (deterministic) *polynomial-time* algorithm for sparse polynomial factorization and hence polynomial-time algorithms for these problems. In the absence of a polynomial-size sparsity bound, we design our algorithms using new techniques.

1.1 Our Results

We will now describe our main results. In what follows, $\mathbb{F}$ is an *arbitrary* field. (finite or otherwise).

---

[1] Although g is not irreducible, this issue can be resolved using standard techniques. For example, by considering the product $f + yh = (g + y)h$ for a new variable y.
[2] A polynomial is s-sparse, it if contains at most s non-zero terms.

1.1.1 Identity Testing for $\Sigma^{[2]}\Pi\Sigma\Pi^{[\text{ind-deg } d]}$ Circuits

The Polynomial Identity Testing (PIT) problem asks to decide whether a given input polynomial is identically zero. The input is usually given in the form of an algebraic circuit. The PIT algorithm is called *white-box* if one can look "inside" the circuit. The algorithm is called *black-box* if the circuit is given via an oracle access, where one is only allowed to evaluate the polynomial on a chosen set of input points. PIT is one of the few natural problems which have a simple efficient randomized algorithm [9, 25, 31] but lack a deterministic one. Indeed, it has been a long standing open question to come up with an efficient deterministic algorithm for this problem. We refer the reader to the full version of the paper [6] for more details.

Our first result is an efficient (deterministic) identity testing algorithm for the class of $\Sigma^{[2]}\Pi\Sigma\Pi^{[\text{ind-deg } d]}$ circuits, where a $\Sigma^{[2]}\Pi\Sigma\Pi^{[\text{ind-deg } d]}$ circuit C of size s computes a polynomial of the form:

$$C = \prod_{i=1}^{r} g_i + \prod_{j=1}^{m} h_j$$

where each polynomial (g_i and h_j) is an s-sparse polynomial with individual degree at most d (for some fixed d). Note, though, that r and m, and hence the total degree of C, can be arbitrary (i.e. polynomially) large. In particular, the polynomial computed by C may not itself be sparse. This class generalizes the model considered in [30], where $m = 1$ and the g_i-s are irreducible polynomials. For the formal definition of our circuit model and further discussion, see Section 4.1.

Observe that the identity testing problem for this circuit class reduces to polynomial factorization of sparse polynomials with bounded individual degree. Therefore, by invoking the factorization algorithm of [4], we can get an algorithm whose runtime is efficient in terms of the sparsity bound. Plugging in the best bound of [4] results in a quasi-polynomial-time algorithm. Our next result gives a *polynomial-time* algorithm for this model. In addition, our algorithm operates in the *black-box* setting, whereas the described factorization-based algorithm is a white-box algorithm.

▶ **Theorem 1.** *There exists a deterministic algorithm that given a black-box access to a* $\Sigma^{[2]}\Pi\Sigma\Pi^{[\text{ind-deg } d]}$ *circuit* C *of size* s *determines if* $C \equiv 0$*, in time* $\text{poly}((sd)^{d^3}, n)$.

An important ingredient in our algorithm is a result that links the gcd of two polynomials, their subresultant and the resultant of their coprime parts - in the **multivariate** setting. See Section 3 for the formal definitions.

▶ **Theorem 2.** *Let* $A, B \in \mathbb{F}[x_1, x_2, \ldots, x_\ell]$ *be two polynomials such that* $A = f \cdot g$ *and* $B = h \cdot g$ *and let* x_i *be a variable. Then*

$$S_{x_i}(d, A, B) = g \cdot \text{Res}_{x_i}(f, h) \cdot \text{lc}_{x_i}(g)^{m'+n'-1}$$

here $m = \deg_{x_i}(A)$, $n = \deg_{x_i}(B)$, $d = \deg_{x_i}(g)$, $m' = \deg_{x_i}(f) = m - d$ *and* $n' = \deg_{x_i}(h) = n - d$. *In addition:*

- $\text{Res}_{x_i}(f, h)$ *is the resultant of* f *and* h *w.r.t the variable* x_i.
- $\text{lc}_{x_i}(g)$ *is the leading coefficient of* g *when written as a polynomial in* x_i
- *And finally,* $S_{x_i}(d, A, B)$ *is the* d*-th subresultant of* A *and* B.

To put the result in context, consider two univariate polynomials $A, B \in \mathbb{F}[x]$. A classical result in the Theory of Resultants (see e.g. [14, 12, 8]) states that:

1. $\mathrm{Res}(A,B) \equiv 0$ if and only if $\gcd(A,B)$ is non-trivial.
2. If u and v are field elements (i.e. $u, v \in \mathbb{F}$) then $\mathrm{Res}(uA, vB) = \mathrm{Res}(A,B) \cdot u^{\deg B} \cdot v^{\deg A}$.
3. The *j-th Subresultant* $S(j,A,B) \equiv 0$ whenever $j < \deg(\gcd(A,B))$.
4. There exists a non-zero field element $\alpha \in \mathbb{F}$ such that $S(j,A,B) = \alpha \cdot \gcd(A,B)$, when $j = \deg(\gcd(A,B))$.

In the multivariate setting one can always regard multivariate polynomials as polynomials in a single variable with coefficients being rational functions in the remaining variables. Yet, in this case α is no longer a mere "field element" as it can now be an arbitrary rational function in the remaining variables! From that perspective, our result can be seen as explicitly expressing α as a polynomial (and not even a rational function) in the remaining variables. We believe that this explicit relation could be of interest in its own right.

To illustrate the aforementioned problem and provide more intuition on our result, let us write the polynomials A and B in the statement of Theorem 2 as $A = (uf) \cdot (g/u)$ and $B = (uh) \cdot (g/u)$, where u is a rational function that **does not** depend on x_i. Observe that the introduction of u does not affect the degrees of x_i. We obtain the following invariant:

$$S_{x_i}(d,A,B) \;=\; \frac{g}{u} \cdot \mathrm{Res}_{x_i}(uf, uh) \cdot \mathrm{lc}_{x_i}\left(\frac{g}{u}\right)^{m'+n'-1} \;=$$
$$\frac{g}{u} \cdot \mathrm{Res}_{x_i}(f,h) \cdot u^{m'+n'} \cdot \frac{\mathrm{lc}_{x_i}(g)^{m'+n'-1}}{u^{m'+n'-1}} = \; g \cdot \mathrm{Res}_{x_i}(f,h) \cdot \mathrm{lc}_{x_i}(g)^{m'+n'-1}.$$

1.1.2 Exact Powers

Our next result pertains to exact powers of polynomials. A polynomial $f \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ is an *exact power* if there exists (another) polynomial $g \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ and $e \in \mathbb{N}$ such that $f = g^e$. We note that despite the rich structure, the best known sparsity bound for exact roots (i.e. $\|g\|$ in terms of $\|f\|$) is the general sparsity bound of size $s^{O(d^2 \log n)}$ by [4]. Hence, one can use the factorization algorithm of [4] to test if a given sparse polynomial is an exact power, in quasi-polynomial time. Similarly, a polynomial-size sparsity bound, even for the case of exact roots, would imply a polynomial-time algorithm for exact-power testing problem. We provide a polynomial-time algorithm for exact-power testing that **does not** rely on this sparsity bound.

▶ **Theorem 3.** *There is a deterministic algorithm that given a sparse polynomial $f \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ of individual degree d as an input, decides whether $f = g^e$ for some polynomial $g \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ and $e \in \mathbb{N}$, in time* $\mathrm{poly}(s^{d^2}, n)$.

We remark that the algorithm only performs exact-power testing and **does not** output a "witness" polynomial g. Indeed, a polynomial-time algorithm that actually outputs g would imply a polynomial-size sparsity bound on exact roots! In addition, the runtime of our algorithm is polynomial in the bit-complexity of the field elements since it does not rely on univariate polynomial factorization. For instance, for finite fields we get the runtime of $\mathrm{poly}(\log|\mathbb{F}|)$ vs $\mathrm{poly}(|\mathbb{F}|)$.

We defer the details and proof of Theorem 3 in Section 5 of the full version of the paper [6].

1.1.3 Improved Sparsity Bounds for Co-factors of Multilinear Polynomials

Given two polynomials $f, h \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ such that $f = gh$, g is called a *quotient polynomial* or a *co-factor* of h. We study the problem of multilinear co-factor sparsity: suppose f is s-sparse and h is multilinear. How sparse/dense can g be? We remark that

any (even non-constructive) efficient upper bound on the sparsity of g allows us to compute g efficiently by interpolating the ratio f/h using a reconstruction algorithm for sparse polynomials (e.g. [20]) and verifying the result.

The motivation to study this problem is two-fold: first of all, by previous results (see e.g. [4]) a multilinear factor of an s-sparse polynomial (of any degree) is itself s-sparse. This suggests more structure for multilinear co-factors we could potentially exploit. Second, a polynomial-size sparsity bound on multilinear co-factors g (even when the individual degree of g is $d = 2$) would imply a polynomial-size sparsity bound for (**all** factors of) polynomials with individual degree $d = 3$. We note that the multicubic ($d = 3$) case is the first instance where we do not have a polynomial-size factor-sparsity bound yet. Indeed, multilinear co-factors can be seen as the "bottle-neck" for this case. The formal argument is given in Section 1.4.

To state our next result we need the following technical definition. We say that a polynomial $h \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ has a *unique projection of length* k if there exist k variables $x_{i_1}, x_{i_2}, \ldots, x_{i_k}$ and k corresponding exponents $e_1, e_2, \ldots, e_k$ such that h has a unique monomial that contains the pattern $x_{i_1}^{e_1} x_{i_2}^{e_2} \cdot \ldots \cdot x_{i_k}^{e_k}$

▶ **Theorem 4.** *Let $f \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ be a polynomial of sparsity s and individual degree at most d such that $f = gh$. Suppose, in addition, that h is a multilinear polynomial with a unique projection of length k. Then the sparsity of g is bounded by $s^{O(dk)}$.*

We remark that Example 2 with $d = p - 1$ (resulting in a lower bound of $n^{\Omega(p)}$) showcases the tightness of our result as here f is n-sparse and $h = x_1 + \ldots + x_n$ has a unique projection of length 1 (e.g. x_1) which results in an upper bound of $n^{O(p)}$ for g. We can also extend Theorem 4 to the case of a co-factor of a power of a multilinear polynomial. Subsequently, we show that every multilinear s-sparse polynomial has a unique projection of length $O(\log s)$ (see Theorem 6.25 & Lemma 6.9 in the full version of the paper [6]). By plugging this result into Theorem 4, we obtain a new sparsity bound of size $s^{O(d \log s)}$ for all multilinear co-factors.

▶ **Corollary 4.** *Let $f \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ be a polynomial of sparsity s and individual degree at most d such that $f = gh$. Suppose, in addition, that h is a multilinear polynomial. Then the sparsity of g is bounded by $s^{O(d \log s)}$.*

The obtained bound is slightly better than the general sparsity bound of size $s^{O(d^2 \log n)}$ by [4] when $s = \text{poly}(n)$. Although our overall improvement may seem incremental (e.g. it does not allow us to "get rid" of the $\log n$ in the exponent) our main contribution here is conceptual: identifying a combinatorial property – the length of the shortest unique projection – that governs the bound on the sparsity of multilinear co-factors.

1.2 Related Work

For the sparse polynomial factorization problem, [4] have shown that factors of an s-sparse polynomial of individual degree d, have their sparsity bounded by $s^{O(d^2 \log n)}$. Currently, this is the best known bound for factor-sparsity when $d \geq 3$. For restricted classes of symmetric polynomials, Bisht and Saxena [5] recently improved this bound to $s^{O(d^2 \log d)}$.

In [13], another problem was posed alongside the sparse factorization problem, in the hope that it might be easier. This problem is referred to as *testing sparse factorization*. Given $m + 1$ sparse polynomials $f, g_1, \ldots, g_m$, it asks to test whether $f = g_1 \cdot \ldots \cdot g_m$. The work of [23] gives a polynomial-time algorithm for this problem, in the special case where every g_i is a sum of univariate polynomials. [30] gives a polynomial-time algorithm when f (and therefore every g_i) has constant individual degree and each g_i is an irreducible polynomial.

Our PIT result is connected to this problem. In Theorem 1, we give a polynomial-time algorithm to test whether $\prod_{i=1}^{r} f_i = \prod_{j=1}^{m} g_j$, where each f_i, g_j is a sparse polynomial with constant individual degree. Note that now LHS is also a product of polynomials. Moreover, there is no restriction placed on g_j-s except that they have bounded individual degree.

The depth-4 $\Sigma\Pi\Sigma\Pi$ circuit class is extremely important in the context of the PIT problem, as it is known that a polynomial-time black-box PIT for this class implies a quasi-polynomial-time black-box PIT for general VP circuits [2, 1]. For a long time, no PIT algorithm better than the trivial $d^{O(n)}$ time algorithm was known for this class, until the recent breakthrough result of Limaye et al. [21], which gives a sub-exponential time algorithm. Various restricted versions of depth-4 circuits are studied to get close to polynomial-time PIT algorithms. For example, Peleg and Shpilka [22] give a polynomial-time PIT algorithm for $\Sigma^{[3]}\Pi\Sigma\Pi^{[2]}$ circuits, where the top fan-in is 3 and the bottom fan-in is 2. Recently, Dutta et al. [10] gave a quasi-polynomial-time PIT for $\Sigma^{[k]}\Pi\Sigma\Pi^{[d]}$ circuits, where the top fan-in k and bottom fan-in d are allowed to be any fixed constants. In this model, the restriction on bottom fan-in implies that the bottom $\Sigma\Pi$ computes polynomials of total degree at most d. We give polynomial-time PIT algorithm for $\Sigma^{[2]}\Pi\Sigma\Pi^{[\mathsf{ind\text{-}deg}\ d]}$ model, where the top fan-in is 2 and the bottom $\Sigma\Pi$ computes polynomials with *individual* degree at most d. We note that the individual degree restriction is much weaker than the total degree restriction. Indeed, even for the case of individual degree bounded by 1 (i.e. multilinear polynomials) the total degree can still be $\Omega(n)$! [24] gave a polynomial-time PIT algorithm for the class of multilinear $\Sigma^{[k]}\Pi\Sigma\Pi$ circuits, with constant top fan-in k, where every gate in the circuit computes a multilinear polynomial. Yet, even a white-box polynomial-time PIT for *general* $\Sigma^{[2]}\Pi\Sigma\Pi$ circuits is still open.

Another related problem is that of *divisibility testing*, which gives two multivariate polynomials f and h and asks to decide whether h divides f. [11] gives a quasi-polynomial-time algorithm when f is sparse and h is a quadratic polynomial (and hence also sparse). We note that the quadratic restriction on h is much stronger than a constant individual degree restriction, although there is no constant degree restriction for f here. [30] gives a polynomial-time algorithm when both f, h are sparse and have constant individual degree. In the proof of Corollary 4, we solve a "search" version of the divisibility testing problem, i.e. we actually compute f/h in quasi-polynomial time, when f is sparse with constant individual degree and h is a multilinear factor of f.

1.3 Our Techniques & Proof Ideas

Let $C = \prod_{i=1}^{r} g_i + \prod_{j=1}^{m} h_j$ where g_i-s and h_j-s are s-sparse polynomials in $\mathbb{F}[x_1, x_2, \ldots, x_n]$ of individual degree at most d. Clearly, if $C \equiv 0$ then it will evaluate to zero on any input. Now suppose $C \not\equiv 0$. Our goal is to find a point $\mathbf{a} \in \mathbb{F}^n$ such that $C(\mathbf{a}) \neq 0$. Our approach relies on the uniqueness of factorization property of the ring of multivariate polynomials. Specifically, we have that $\prod_{i=1}^{r} g_i \neq -\prod_{j=1}^{m} h_j$. Consequently, wlog there exists an irreducible polynomial (factor) u and $\ell > 0$ such that u^ℓ divides the LHS but does not divide the RHS. Our goal is to preserve this "situation" while reducing the number of variables. Clearly, a random projection will be sufficient. However, we wish to obtain a deterministic algorithm. To this end, we are looking for a projection that does not introduce new dependencies between factors. That is, for every i, j: if $v \mid g_i$ and $u \mid h_j$ satisfying $\gcd(u, v) = 1$ we need to ensure that $\gcd(u', v') = 1$, when u' and v' are the projections of u and v, respectively. The main tool for that is the *Resultant*. Indeed, one of the fundamental properties of the resultant is $\mathrm{Res}(A, B) \not\equiv 0$ if and only if $\gcd(A, B) = 1$. In the multivariate setting, this condition roughly translates into: $[\forall x_k : \mathrm{Res}_{x_k}(u, v) \not\equiv 0] \implies \gcd(u', v') = 1$. In other words, we

need to hit all the resultants of the form $\text{Res}_{x_k}(u, v)$ when $v \mid g_i$ and $u \mid h_j$. By definition, $\text{Res}_{x_k}(u, v)$ is a determinant of $2d \times 2d$ matrix where each entry is a coefficient of u or v. Hence, $\text{Res}_{x_k}(u, v)$ is $t^{O(d)}$-sparse polynomial with individual degree at most $O(d^2)$, where t is an upper bound on the sparsities of u and v. Consequently, we can use a hitting set generator for sparse polynomials (e.g. [20]) to hit the resultant. As u and v are factors of s-sparse polynomials of individual degree d, the best upper by [4] will be $t = s^{O(d^2 \log s)}$. This will result in a quasi-polynomial-time algorithm.

Another idea would be to use the multiplicative properties of the resultant and hit $\text{Res}_{x_k}(h_j, g_i)$ instead. Indeed, $\text{Res}_{x_k}(h_j, g_i) \not\equiv 0 \implies \text{Res}_{x_k}(u, v) \not\equiv 0$ and since g_i and h_j are s-sparse, $\text{Res}_{x_k}(h_j, g_i)$ is $s^{O(d)}$-sparse and this would get a polynomial-time algorithm. The main issue is that we could have $\text{Res}_{x_k}(u, v) \not\equiv 0$ while $\text{Res}_{x_k}(h_j, g_i) \equiv 0$. For example, if $h_j = uf$ and $g_i = vf$ for the same polynomial f. Going back, one may ask whether we could show a better sparsity bound on $\text{Res}_{x_k}(u, v)$. While we do not quite do that, we instead show that $\text{Res}_{x_k}(u, v)$ is a *factor* of *some* $s^{O(d)}$-sparse polynomial of individual degree at most $O(d^2)$.

As the ring of polynomials forms an integral domain, this allows us to use a polynomial-size hitting set generator for sparse polynomials.

To achieve the above goal, suppose for simplicity that $g_i = u^{a_1} \cdot v^{b_1}$ and $h_j = u^{a_2} \cdot v^{b_2}$, for some non-negative integers a_1, b_1, a_2, b_2. If all these numbers are strictly positive, we run into the same issue we have encountered earlier. That is, $\text{Res}_{x_k}(u, v) \not\equiv 0$ while $\text{Res}_{x_k}(h_j, g_i) \equiv 0$. To address that, we apply Theorem 2 (our key technical contribution) which allows us to "extract" the gcd. For example, if $g_i = uv^2$ and $h_j = u^2 v$, we can write $g_i = v \cdot uv$ and $h_j = u \cdot uv$ and obtain that $\text{Res}_{x_k}(u, v)$ is a factor of $S_{x_k}(\deg_{x_k}(uv), g_i, h_j)$, which is an $s^{O(d)}$-sparse polynomial (see Observation 15). However, a sole gcd extraction may be insufficient. Consider the case when $g_i = uv^2$ and $h_j = uv$. Repeating the same argument will just yield a trivial statement that $\text{Res}_{x_k}(v, 1) = 1$ is a factor of a sparse polynomial. To overcome this difficulty, we apply the previous argument on powers of g_i and h_j. That is, on $g_i^z = u^{za_1} \cdot v^{zb_1}$ and $h_j^t = u^{ta_2} \cdot v^{tb_2}$. The idea now would be to isolate the powers of u from the powers of v. Within the same example, consider $g_i^2 = u^2 v^4 = v \cdot u^2 v^3$ and $h_j^3 = u^3 v^3 = u \cdot u^2 v^3$. Now, by Theorem 2, $\text{Res}_{x_k}(u, v)$ is a factor of $S_{x_k}(\deg_{x_k}(u^2 v^3), g_i^2, h_j^3)$. More generally, we show how to find appropriate "small" z and t using linear algebra.

Unfortunately, though, this could be made possible only when h_j and g_i satisfy certain "non-degeneracy" condition w.r.t u and v. More formally, when the matrix $E \triangleq \begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \end{bmatrix}$ has full rank (see Lemma 20).

Our final crucial observation is that we can actually ignore "degenerate" pairs u, v. To this end, we prove a technical lemma (Lemma 7) which could be of independent interest.

1.3.1 Exact Power Testing

Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ be an s-sparse polynomial of constant individual degree d. We show how to test whether $f = g^e$, for some other polynomial $g \in \mathbb{F}[x_1, \ldots, x_n]$ and some $e \in \mathbb{N}$. We utilize the notion of *reverse-monic* polynomials for this result. We call a polynomial h reverse-monic, if there exists some $i \in [n]$, such that $h|_{x_i=0} = 1$. If our input polynomial f is reverse-monic, we show that g is $s^{O(d)}$-sparse. Moreover, we also get an algorithm to compute this exact root g. We prove this in Lemmas 5.4 & 5.9 of the full version of the paper [6], using a formal expansion that can be thought of as a generalization of the Binomial Expansion: $(1 + x)^{\frac{1}{e}} = \sum_{i=0}^{\infty} \binom{\frac{1}{e}}{i} x^i$. In general though, our input polynomial f may not be reverse-monic. We first convert f into a reverse-monic polynomial $\hat{f}$ with respect to some

variable x_i, using a known standard transformation. This step only incurs a slight sparsity blow-up of s^d. One important property of this transformation is that it preserves the "exact power" structure. That is, if $f = g^e$, then $\hat{f} = h^e$, for some polynomial h. We then compute this e-th root of the reverse-monic $\hat{f}$, as mentioned previously.

However, we are still not quite done. It can happen that a polynomial f which was not an exact power, may become an exact power after the reverse-monic transformation. We need an additional condition to get the converse implication. We show that if both $\hat{f}$ and $f_{|_{x_i=0}}$ are exact powers, then we can correctly conclude that f is also an exact power. This gives us a recursive algorithm, as $f_{|_{x_i=0}}$ is a polynomial in $(n-1)$ variables. This procedure is described formally in Algorithm 3 of the full version of the paper [6].

1.3.2 Co-Factor Sparsity Bound

For the co-factor bounds, our results build on the division elimination techniques of [27]. Let us outline our approach. To this end, let $f, h \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ be s-sparse polynomials such that $h(0, \ldots, 0) = 1$ and suppose that $f = gh$ for some polynomial $g \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ with individual degree at most d. Consider the following formal expansion: $\frac{1}{(1-x)} = \sum_{j=0}^{\infty} x^j$. Then we have: $g = \frac{f}{h} = \frac{f}{(1-(1-h))} = \sum_{j=0}^{\infty} f(1-h)^j$ when the equality is an equality of formal sums of monomials. The key observation is that $(1-h)$ does not contain any constants, hence total degree of every monomial in $(1-h)^j$ (and hence in every summand $f(1-h)^j$) is at least j. Consequently, we can "discard" the tail $\sum_{j=dn+1}^{\infty} f(1-h)^j$ since every monomial in g has a total degree of at most dn. Indeed, g will be formed by a subset of monomials of $\sum_{j=0}^{dn} f(1-h)^j$. This allows us to obtain an upper bound on the sparsity of $g : \|g\| \leq \sum_{j=0}^{dn} s^{j+1} \leq s^{dn+2}$. Clearly, the outlined approach has two major flaws:

1. It requires that $h(0, \ldots, 0) = 1$ (or more generally, $h(0, \ldots, 0) \neq 0$). And even then:

2. The obtained bound is exponential in n.

One way to address the former is by a random shift to the variable. However, this may significantly increase **both** the sparsity and the individual degree! We take a different approach. Our main observation is that the argument still works if we treat the polynomials as polynomials in "fewer" variables.Formally, let $I \subseteq [n]$ of size $|I| = k$. We can regard the polynomials as polynomials in the variables x_I with coefficient in the remaining variables. In particular, suppose that $h|_{x_I=0_I} = 1$. In this case we say that h is *I-reverse monic*. Observe that every monomial in $(1-h)^j$ contains at least one variable from x_I. That is, the total x_I-degree of $(1-h)^j$ is at least j and hence (as before) we can discard the tail. Yet now, g depends "only" on k variables and thus its "total" degree is kd (and not nd). This way we obtain a better upper bound on the sparsity of g, if k is "small": $\|g\| \leq \sum_{j=0}^{kd} s^{j+1} \leq s^{kd+2}$. Of course, our approach still relies on the assumption that h is I-reverse monic for a "small" subset I. Although we are unable to lift this assumption, we can weaken it. As was noted earlier, if $h(0, \ldots, 0) = \alpha \neq 0$ (i.e. when $I = [n]$) we can just divide by α as it is a field element. However, this is no longer possible for an arbitrary I (especially, if I is a small set). Yet, we observe that if $h|_{x_I=0_I} = \alpha$ and α is a non-zero *single monomial* (in the remaining variables) we can transform h into an I-reverse monic polynomial $\hat{h}$ with the exact same sparsity. The idea is to apply the transformation $x_i = \alpha \cdot x_i$ for all $i \in I$. Note that since α is a single monomial, this transformation is reversible. Indeed, there is an 1-1 correspondence between the monomials of h and $\hat{h}$. Given this connection, we refer to such h as I-reverse *pseudo*-monic.

Our final ingredient is (yet) another observation that for multilinear polynomials we can weaken the assumption that h is I-reverse *pseudo*-monic further by considering *unique projections*. That is, monomials that have a "unique pattern". Formally, we want h to have

exactly one monomial that contains the submonomial: $x_{i_1}^{e_1}x_{i_2}^{e_2}\cdots x_{i_k}^{e_k}$. We show that by "flipping" the variables in h we can transform it into another multilinear polynomial $\tilde{h}$ which is $\{i_1, i_2, \ldots, i_k\}$-reverse pseudo-monic. As a result, $\|g\| \leq s^{kd+2}$.

This is our main conceptual contribution: the upper bound on the sparsity of a multilinear co-factor g is governed by a *combinatorial property* of the set of monomials of h: the length of the shortest unique projection. As an application, we show that every s-sparse polynomial has a unique projection of length at most $\log s + 1$, thus we obtain a new, slightly stronger, sparsity bound on co-factors of multilinear polynomials.

1.4 Multilinear co-Factor Motivation

Theorem 4 and Corollary 4 in this paper apply to the factorization scenario of $f = gh$ where f is s-sparse and h is multilinear. First of all, note that by previous results (see [4] and references within) h itself is s-sparse. So we are looking to bound the sparsity of g. As it turns out, this pattern is the "bottleneck" case for multicubic polynomials. In other words, showing a polynomial-size sparsity bound on g in this scenario would imply a polynomial-size sparsity bound on factors of general multicubic polynomials! In fact, it is sufficient to consider the case when the degree of g in every variable is exactly 2! We remark that getting polynomial-size sparsity bound is open for $d \geq 3$. The following lemma summarizes this formally.

▶ **Lemma 5.** *Suppose there exists an absolute constant $a \geq 1$ such that for any multicubic polynomial f: if $g \mid f$ **and** f/g is multilinear then $\|g\| \leq \|f\|^a$. Then for any multicubic polynomial f if $g \mid f$ then $\|g\| \leq \|f\|^a$.*

We defer the proof to Section A.

2 Preliminaries

Notations

We use the shorthand $[n]$ for the set $\{1, 2, \ldots, n\}$. We denote a vector $v = (v_1, \ldots, v_n)$ in short by $\mathbf{v}$ (as a column vector). We denote the n-fold Cartesian product of a set H by H^n. The set of non-negative real numbers is denoted by $\mathbb{R}_{\geq 0}$ and $\mathbb{R}^n_{\geq 0}$ denotes the space of n-dimensional non-negative real vectors.

Let $f \in \mathbb{F}[\mathbf{x}] = \mathbb{F}[x_1, x_2, \ldots, x_n]$ be an n-variate polynomial. The *individual degree* of a variable x_i in f, denoted by $\deg_{x_i}(f)$, is defined as the maximum degree of that variable in f, while the *individual degree* of f is the maximum among all the individual degrees, $\max_{i\in[n]} \deg_{x_i}(f)$. We define the *sparsity* of f as the number of non-zero terms in f. Let us denote *sparsity* of f as $\|f\|$. We say that f *depends* on a variable x_i if there exist $\mathbf{a}, \mathbf{b} \in \mathbb{F}^n$ which differ only in the i-th coordinate such that $f(\mathbf{a}) \neq f(\mathbf{b})$. We define the set $\mathrm{var}(f) \triangleq \{i \mid f \text{ depends on } x_i\}$. For $i \in [n]$, we denote by $\mathrm{lc}_{x_i}(f)$ the *leading coefficient* of f when written as a polynomial in x_i. Formally, let $f = \sum_{j=0}^{d} f_j \cdot x_i^j$ such that $\forall j$, f_j is a polynomial in rest of the variables and $f_d \not\equiv 0$. Then $\mathrm{lc}_{x_i}(f) \triangleq f_d$. Observe that if $f = g \cdot h$ then $\forall i \in [n] : \mathrm{lc}_{x_i}(f) = \mathrm{lc}_{x_i}(g) \cdot \mathrm{lc}_{x_i}(h)$.

For a set $I \subseteq [n]$, we use x_I to denote the set of variables $\{x_i \mid i \in I\}$ and $x_{[n]\setminus I}$ to denote the set of remaining variables. We use the symbol $f|_{x_I=0_I}$ to denote the polynomial resulting from substituting 0 at all the x_I variables in f. Let $\mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{F}^n$. For $i \in [n]$ we define a partial assignment $\mathbf{a}_{-i} \triangleq (a_1, \ldots, a_{i-1}, a_{i+1}, \ldots, a_n)$ and $f(x_i, \mathbf{a}_{-i}) \triangleq f(a_1, \ldots, a_{i-1}, x_i, a_{i+1}, \ldots, a_n)$.

Let $f \sim g$ denote equality of f and g up to multiplication by non-zero a field constant, i.e. $f = c \cdot g$, for some $c \neq 0 \in \mathbb{F}$.

2.1 Vectors and Interlacing

In this section we will state some useful properties of vectors that may be of independent interest. We defer the proof of Lemma 7 to Section A.2.

▶ **Definition 6** (Interlacing). *Let $a, b, c, d \in \mathbb{R}$. We say that two pair of points (a, b) and (c, d)* interlace *if $(a - c)(b - d) < 0$. Equivalently, $(a - c)$ and $(b - d)$ have opposite signs.*

▶ Remark. In particular, interlacing implies that at least two of a, b, c, d are non-zero.
The following result relates the interlacing property with linear dependence.

▶ **Lemma 7.** *Let $\mathbf{a}, \mathbf{b} \in \mathbb{R}^n_{\geq 0}$ and $\mathbf{c}, \mathbf{d} \in \mathbb{R}^m_{\geq 0}$ such that $\left(\sum_{i=1}^{n} a_i, \sum_{i=1}^{n} b_i\right)$ interlaces with $\left(\sum_{j=1}^{m} c_j, \sum_{j=1}^{m} d_j\right)$. Then there exists $i \in [n], j \in [m]$ s.t. the matrix $\begin{bmatrix} a_i & c_j \\ b_i & d_j \end{bmatrix}$ has full rank.*

3 GCD, Resultants and Subresultants

In this section we discuss the classical tool of resultants and subresultants. In Theorem 14 we prove an interesting connection which is a somewhat more general version of Theorem 2. This result will be crucial in the design of our PIT algorithm and could be of interest in its own right.

The polynomial ring $\mathbb{F}[x_1, \ldots, x_n]$ is a unique factorization domain (UFD). Hence, the gcd of two polynomials is well defined up to a multiplication by field element. We can also define gcd with respect to a single variable x_i, where we treat rest of the variables as field elements. That is, $\gcd_{x_i}(f, g)$ is well defined up to multiplication by a rational function depending on the remaining variables. By convention, we choose the normalized gcd whenever we consider $\gcd_{x_i}(f, g)$ in this work. For example, let $f = x_1^2 x_2 + x_1 x_2^2$ and $g = x_1^2 x_2^2$, then $\gcd(f, g) = x_1 x_2$ while $\gcd_{x_2}(f, g) = x_2$. Technically, the former is gcd in $\mathbb{F}[x_1, x_2]$ and the latter is normalized gcd in $\mathbb{F}(x_1)[x_2]$. We now consider a somewhat more general scenario. Let $A(y), B(y) \in \mathcal{R}[y]$ be two non-zero polynomials of y-degree d and e, respectively in an arbitrary UFD $\mathcal{R}$. Suppose $A(y) = \sum_{i=0}^{d} a_i \cdot y^i$ and $B(y) = \sum_{j=0}^{e} b_j \cdot y^j$. Consider the $(d + e) \times (d + e)$ *Sylvester matrix* M whose first e rows are the e shifts of the row vector $(a_d, \ldots, a_0, 0, \ldots, 0)$ and next d rows are the d shifts of the row vector $(b_e, \ldots, b_0, 0, \ldots, 0)$.

$$M = \begin{bmatrix} a_d & a_{d-1} & \ldots & a_1 & a_0 & & & \\ & a_d & a_{d-1} & \ldots & a_1 & a_0 & & \\ & & \ldots & \ldots & \ldots & \ldots & & \\ & & & a_d & a_{d-1} & \ldots & a_1 & a_0 \\ b_e & b_{e-1} & \ldots & b_1 & b_0 & & & \\ & b_e & b_{e-1} & \ldots & b_1 & b_0 & & \\ & & \ldots & \ldots & \ldots & \ldots & & \\ & & & b_e & b_{e-1} & \ldots & b_1 & b_0 \end{bmatrix}.$$

▶ **Definition 8** (Resultant). *The resultant $\mathrm{Res}_y(A, B) \in \mathcal{R}$ is defined to be the determinant of this Sylvester matrix. That is, $\mathrm{Res}_y(A, B) = \det(M)$.*

In our setting, $\mathcal{R}$ will be a polynomial ring, say $\mathbb{F}[x_1, \ldots, x_n]$ and $\text{Res}_y(A, B)$ will be a polynomial free of y-variable. The following is the most fundamental property of the Resultant:

▶ **Lemma 9** (See e.g. [14, 12, 8]). *Let $A, B \in \mathbb{F}[y, x_1, \ldots, x_n]$ be two polynomials. Then $\gcd_y(A, B) \neq 1$ if and only if $\text{Res}_y(A, B) \equiv 0$. That is, A and B have a non-trivial factor that depends on the variable y iff the Resultant of A, B w.r.t. y is the identically zero polynomial.*

We state the following important connection between the projection of the resultant and the GCD of projections.

▶ **Lemma 10.** *Let $f(y, \mathbf{x})$ and $g(y, \mathbf{x})$ be two polynomials in $\mathbb{F}[y, \mathbf{x}]$. Let $\mathbf{a} \in \mathbb{F}^n$. Then, $\text{Res}_y(f, g)(\mathbf{a}) \neq 0 \implies \gcd_y(f(y, \mathbf{a}), g(y, \mathbf{a})) = 1$.*

The proof can be found in Section A. We also require multiplicative property of the Resultant that essentially follows from the definition:

▶ **Lemma 11.** *Let $A, B, u, v \in \mathbb{F}[y, x_1, \ldots, x_n]$ be polynomials. Then $\text{Res}_y(A, B) \mid \text{Res}_y(uA, vB)$.*

We now study few useful sub-matrices of the Sylvester matrix below.

▶ **Definition 12** (j-th principal resultant). *Let M_j be the submatrix of M formed by deleting last j rows of A terms, last j rows of B terms and the last $2j$ columns. We call M_j to be the j-th principal resultant of A and B. Note that $\text{Res}_y(A, B) = M = M_0$.*

We can now define the subresultant polynomial as follows.

▶ **Definition 13** (Subresultant). *Let M_{ij} be the $(d + e - 2j) \times (d + e - 2j)$ submatrix of Sylvester matrix M formed by deleting:*

- *rows $e - j + 1$ to e (each having coefficients of $A(y)$),*
- *rows $d + e - j + 1$ to $d + e$ (each having coefficients of $B(y)$),*
- *columns $d + e - 2j$ to $d + e$, except for column $d + e - i - j$.*

Note that the j-th principal resultant M_j is exactly M_{jj}.
For $0 \leq j \leq e$, the j-th subresultant of $A(y), B(y) \in \mathcal{R}[y]$ is the polynomial in $\mathcal{R}[y]$ of degree j defined by $S_y(j, A, B) = \det(M_{0j}) + \det(M_{1j}) \cdot y + \ldots + \det(M_{jj}) \cdot y^j$.

We now prove our main technical result which links the gcd of two polynomials, their subresultant and the resultant of their coprime parts. Theorem 2 is a special case of this result which, we believe, could be interesting in its own right.

▶ **Theorem 14.** *Let $A(x), B(x) \in \mathcal{R}[x]$ be two polynomials over an arbitrary UFD $\mathcal{R}$. Suppose $A(x) = f(x) \cdot g(x)$ and $B(x) = h(x) \cdot g(x)$ with $\deg_x(A) = m$, $\deg_x(B) = n$, $\deg_x(g) = d$, $\deg_x(f) = m' = m - d$ and $\deg_x(h) = n' = n - d$. Then*

$$S_x(d, A, B) = g \cdot \text{Res}_x(f, h) \cdot \text{lc}_x(g)^{m'+n'-1}.$$

Due to its length and space limitation, we defer the proof of Theorem 14 to Section A.1 in the Appendix. We conclude this section making an important observation that any subresultant (and hence the Resultant) of two sparse polynomials of individual degree at most d is a sum of at most $d + 1$ determinants of $2d \times 2d$ matrices where each entry is a coefficient of a sparse polynomial and, hence is itself a (somewhat) sparse polynomial of a small individual degree.

▶ **Observation 15.** *Let $A, B \in \mathbb{F}[y, x_1, \ldots, x_n]$ be two s-sparse polynomials with individual degrees at most d. Then for any j, $S_y(j, A, B)$ is an $(2ds)^{2d+1}$-sparse polynomial with individual degrees at most $2d^2$.*

4 PIT for $\Sigma^{[2]}\Pi\Sigma\Pi^{[\text{ind-deg } d]}$ Circuits

In this section we prove our main result Theorem 1. We refer the reader to the full version of the paper [6] for the formal definition of an algebraic circuit and PIT algorithm. For the purpose of black-box PIT algorithm, we require the notion of hitting set generators (HSG) or simply generators.

▶ **Definition 16** (Generator). *Let $\mathcal{C}$ be a class of n-variate polynomials. Consider $\mathcal{G} = (\mathcal{G}_1, \mathcal{G}_2, \ldots, \mathcal{G}_n) : \mathbb{F}^k \to \mathbb{F}^n$, an n-tuple of k-variate polynomials where for each $i \in [n]$, $\mathcal{G}_i \in \mathbb{F}[t_1, t_2, \ldots, t_k]$. Let $f(x_1, \ldots, x_n)$ be an n-variate polynomial. We define action of $\mathcal{G}$ on polynomial f by $f(\mathcal{G}) = f(\mathcal{G}_1, \ldots, \mathcal{G}_n) \in \mathbb{F}[t_1, \ldots, t_k]$. We call $\mathcal{G}$ a k-seeded generator for class $\mathcal{C}$ if for every non-zero $f \in \mathcal{C}$, $f(\mathcal{G}) \not\equiv 0$. Degree of generator $\mathcal{G}$ is defined as $\deg(\mathcal{G}) \triangleq \max\{\deg(\mathcal{G}_i)\}_{i=1}^n$. We define $\mathcal{G}_{-i} \triangleq (\mathcal{G}_1, \ldots, \mathcal{G}_{i-1}, \mathcal{G}_{i+1}, \ldots, \mathcal{G}_n)$ and $f(x_i, \mathcal{G}_{-i}) \triangleq f(\mathcal{G}_1, \ldots, \mathcal{G}_{i-1}, x_i, \mathcal{G}_{i+1}, \ldots, \mathcal{G}_n)$.*

For a polynomial-time PIT algorithm, k is kept constant. A generator $\mathcal{G}$ acts as a variable reduction map which converts an input polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ to $f(\mathcal{G}) \in \mathbb{F}[t_1, \ldots, t_k]$ such that $f \equiv 0$ if and only if $f(\mathcal{G}) \equiv 0$. Let D be the degree of $\mathcal{G}$ and d be the individual degree of f. Then $\mathcal{G}$ gives us a brute-force hitting-set of size $(ndD)^k$ (Lemma A.2 in the full version of the paper [6]). In other words, we get a polynomial-time black-box PIT algorithm for f when k is constant, $\mathcal{G}$ can be designed in polynomial time and its degree is also polynomially bounded.

4.1 The $\Sigma^{[k]}\Pi\Sigma\Pi^{[\text{ind-deg } d]}$ Model

A size s, depth-4 $\Sigma\Pi\Sigma\Pi$ circuit computes a polynomial of the form $f = \sum_{i=1}^{k} \prod_{j=1}^{m_i} f_{ij}$, where f_{ij} are s-sparse polynomials for each $i \in [k], j \in [m_i]$.

For $k = 2$, even white-box PIT for $\Sigma^{[2]}\Pi\Sigma\Pi$ circuits is still open. A more restricted model is the class of $\Sigma^{[k]}\Pi\Sigma\Pi^{[d]}$ circuits, where the top fan-in k and the bottom fan-in d are constants. For a size-s circuit of this class, f_{ij}'s are s-sparse polynomials of constant total degree at most d. For $k = 3$ and $d > 2$, coming up with a polynomial PIT algorithm remains an open question here. We now introduce, what we call the $\Sigma^{[k]}\Pi\Sigma\Pi^{[\text{ind-deg } d]}$ model. In the $\Sigma^{[k]}\Pi\Sigma\Pi^{[d]}$ model, the sparse polynomials f_{ij}'s have constant total degree $\leq d$. We relax this restriction to f_{ij}'s being constant *individual* degree $\leq d$ polynomials in $\Sigma^{[k]}\Pi\Sigma\Pi^{[\text{ind-deg } d]}$ model. This is a more general model, since f_{ij}'s can now have much higher total degree, like $\Omega(n)$. In Section 4.2, we give a deterministic polynomial-time black-box PIT algorithm for this model when $k = 2$ and d is any constant. We also note that our PIT algorithm works for any field $\mathbb{F}$, while the works of [22, 10] do have certain field restrictions.

4.2 The PIT Algorithm

For a polynomial f and an irreducible polynomial u, let $\mathrm{e}_u(f)$ denote the highest power of u in f. In other words, $f = u^{\mathrm{e}_u(f)} \cdot g$, such that $u \nmid g$. If $u \nmid f$, then $\mathrm{e}_u(f) = 0$. We define a polynomial Φ with respect to two non-zero polynomials P, Q as follows:

▶ **Definition 17.** *Let $P, Q \in \mathbb{F}[x_1, \ldots, x_n]$ be two non-zero polynomials. Define the polynomial $\Phi_{P,Q} \in \mathbb{F}[x_1, \ldots, x_n]$ as: $\Phi_{P,Q} \triangleq \prod_{\substack{u,v \mid PQ \\ i \in [n]}} \mathrm{Res}_{x_i}(u, v) \cdot \prod_{i \in [n]} \mathrm{lc}_{x_i}(P) \cdot \prod_{i \in [n]} \mathrm{lc}_{x_i}(Q)$, where u, v are irreducible factors of P or Q such that $(\mathrm{e}_u(P), \mathrm{e}_v(P))$ interlaces with $(\mathrm{e}_u(Q), \mathrm{e}_v(Q))$. Moreover, we only consider non-zero multiplicands.*

The next Lemma shows that a non-zero of Φ preserves non-similarity of polynomials.

▶ **Lemma 18.** *Let $P, Q \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ be two polynomials such that $P \not\sim Q$ and let $\mathbf{a} \in \mathbb{F}^n$ such that $\Phi_{P,Q}(\mathbf{a}) \neq 0$. Then, there exists an $i \in [n]$ such that $P(x_i, \mathbf{a}_{-i}) \not\sim Q(x_i, \mathbf{a}_{-i})$.*

Proof. By our premise, we have $P \not\sim Q$. By uniqueness of factorization, without loss of generality, there exists an irreducible factor u of P, appearing with higher power in P than in Q. That is, $k \triangleq \mathrm{e}_u(P) > \ell \triangleq \mathrm{e}_u(Q) \geq 0$. Let $P = u^k \cdot G$ and $Q = u^\ell \cdot H$, for some polynomials G, H such that u does not divide either of them. Define the set $T \triangleq \{v \mid v \text{ is an irreducible factor of } H \text{ and } \mathrm{e}_v(P) \geq \mathrm{e}_v(Q)\}$. Then let $P = u^k \cdot \left(\prod_{v \in T} v^{\mathrm{e}_v(P)}\right) \cdot G'$, $Q = u^\ell \cdot \left(\prod_{v \in T} v^{\mathrm{e}_v(Q)}\right) \cdot H'$, where G', H' are the product of remaining polynomials from G and H respectively. Pick $i \in \mathrm{var}(u)$, i.e. u depends on x_i. Note that $\mathrm{lc}_{x_i}(P) \not\equiv 0$, since u is a factor of P which depends on x_i. Since lc is multiplicative, we get that $\mathrm{lc}_{x_i}(P(x_i, \mathbf{a}_{-i})) = \mathrm{lc}_{x_i}(u(x_i, \mathbf{a}_{-i}))^k \cdot \mathrm{lc}_{x_i}(G(x_i, \mathbf{a}_{-i}))$. From our premise, we also know that $\Phi_{P,Q}(\mathbf{a}) \neq 0$. Then by the definition of $\Phi_{P,Q}$, we get that $\mathrm{lc}_{x_i}(P(x_i, \mathbf{a}_{-i})) \neq 0$, which implies that $\mathrm{lc}_{x_i}(u(x_i, \mathbf{a}_{-i})) \neq 0$. Together with the fact that u has x_i-degree at least one, we conclude that $u(x_i, \mathbf{a}_{-i})$ also has x_i-degree at least one. Suppose for the sake of contradiction that $P(x_i, \mathbf{a}_{-i}) \sim Q(x_i, \mathbf{a}_{-i})$. Then:

$$u(x_i, \mathbf{a}_{-i})^{k-\ell} \cdot \left(\prod_{v \in T} v^{\mathrm{e}_v(P) - \mathrm{e}_v(Q)}(x_i, \mathbf{a}_{-i})\right) \cdot G'(x_i, \mathbf{a}_{-i}) \sim H'(x_i, \mathbf{a}_{-i}).$$

Since $k > \ell$ and $\forall v \in T : \mathrm{e}_v(P) \geq \mathrm{e}_v(Q)$, LHS is a proper polynomial in the above equation. Moreover, $u(x_i, \mathbf{a}_{-i})$ divides LHS. Now since LHS $\sim H'(x_i, \mathbf{a}_{-i})$ and $u(x_i, \mathbf{a}_{-i})$ depends on x_i, we deduce that $H'(x_i, \mathbf{a}_{-i})$ also depends on x_i. By uniqueness of factorization, we also deduce that $u(x_i, \mathbf{a}_{-i})$ divides $H'(x_i, \mathbf{a}_{-i})$. Let $H' = v_1^{e_1} \cdot \ldots \cdot v_m^{e_m}$ be the irreducible factorization of H', for some $m \geq 1$ and $e_j \geq 1$ for all $j \in [m]$, where each v_j is irreducible. Here $e_j = \mathrm{e}_{v_j}(Q)$ for each $j \in [m]$. Then $H'(x_i, \mathbf{a}_{-i}) = v_1(x_i, \mathbf{a}_{-i})^{e_1} \cdot \ldots \cdot v_m(x_i, \mathbf{a}_{-i})^{e_m}$, where $v_j(x_i, \mathbf{a}_{-i})$'s may not be irreducible anymore due to substitution. Recall that u does not divide H and hence it does not divide H' either. Since u is irreducible, we get that $\gcd_{x_i}(u, v_j) = 1$, for all $j \in [m]$. At the same time, recall that $u(x_i, \mathbf{a}_{-i})$ divides $H'(x_i, \mathbf{a}_{-i})$. Since $H'(x_i, \mathbf{a}_{-i})$ depends on x_i, this implies that $u(x_i, \mathbf{a}_{-i})$ shares a non-trivial factor with some $v_j(x_i, \mathbf{a}_{-i})$ which depends on x_i. Thus, there exists some $j \in [m]$ such that $\gcd_{x_i}(u(x_i, \mathbf{a}_{-i}), v_j(x_i, \mathbf{a}_{-i})) \neq 1$. By definition of H', $v_j \notin T$ and hence $\mathrm{e}_{v_j}(P) < \mathrm{e}_{v_j}(Q) = e_j$ for all $j \in [m]$. Recall that $\mathrm{e}_u(P) > \mathrm{e}_u(Q)$. Hence $(\mathrm{e}_u(P), \mathrm{e}_{v_j}(P))$ interlaces with $(\mathrm{e}_u(Q), \mathrm{e}_{v_j}(Q))$. By our premise, $\Phi_{P,Q}(\mathbf{a}) \neq 0$. Then by the definition of $\Phi_{P,Q}$, we get that $\mathrm{Res}_{x_i}(u, v_j)(\mathbf{a}_{-i}) \neq 0$. By further applying Lemma 10, we deduce that $\gcd_{x_i}(u(x_i, \mathbf{a}_{-i}), v_j(x_i, \mathbf{a}_{-i})) = 1$, which gives us a contradiction. Hence, $P(x_i, \mathbf{a}_{-i}) \not\sim Q(x_i, \mathbf{a}_{-i})$. ◀

The following is a technical lemma for future use. The proof is deferred to Section A.

▶ **Lemma 19.** *Let $u, v \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ be two coprime and irreducible polynomials such that $\mathrm{var}(u) \cap \mathrm{var}(v)$ is non-empty. And suppose we have two polynomials $g = u^{a_1} \cdot v^{b_1}$ and $h = u^{a_2} \cdot v^{b_2}$, for some non-negative integers a_1, b_1, a_2, b_2. Define $z \triangleq a_2 + b_2$ and $t \triangleq a_1 + b_1$. For any $i \in \mathrm{var}(u) \cap \mathrm{var}(v)$, let $W \triangleq \gcd_{x_i}(g^z, h^t)$. Finally, let E be the following matrix: $E \triangleq \begin{bmatrix} a_1 & a_2 \\ b_1 & b_2 \end{bmatrix}$. Then $\frac{g^z}{W} = \begin{cases} u^{\det(E)} & \text{if } \det(E) \geq 0 \\ v^{-\det(E)} & \text{otherwise.} \end{cases}$ $\frac{h^t}{W} = \begin{cases} v^{\det(E)} & \text{if } \det(E) \geq 0 \\ u^{-\det(E)} & \text{otherwise.} \end{cases}$*

We now show that under certain non-degeneracy condition, a resultant of two factors of sparse polynomials is itself a factor of a (somewhat) sparse polynomial.

▶ **Lemma 20.** *Let $u \nsim v \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ be two irreducible polynomials. Suppose there exist s-sparse, individual degree-d polynomials $g, h \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ such that the matrix $E \triangleq \begin{bmatrix} \mathrm{e}_u(g) & \mathrm{e}_u(h) \\ \mathrm{e}_v(g) & \mathrm{e}_v(h) \end{bmatrix}$ has full rank. Then for any $i \in [n]$: $\mathrm{Res}_{x_i}(u, v)$ is a factor of a non-zero $(sd)^{\mathcal{O}(d^3)}$-sparse, $\mathcal{O}(d^4)$-individual degree polynomial.*

Proof. Consider any $i \notin \mathrm{var}(u) \cup \mathrm{var}(v)$. Then $\mathrm{Res}_{x_i}(u, v)$ is defined to be 1, which is trivially a factor of any sparse polynomial. Now consider any $i \in \mathrm{var}(u) \setminus \mathrm{var}(v)$. Then by definition, $\mathrm{Res}_{x_i}(u, v) = v^{\deg_{x_i}(u)}$. Note that both $e_v(g)$ and $e_v(h)$ cannot be zero, as E has full rank. Therefore, v is factor of g or h, which are both s-sparse. Similarly, u is also a factor of g or h which implies $\deg_{x_i}(u) \leq d$. We deduce that $\mathrm{Res}_{x_i}(u, v)$ is a factor of an s^d-sparse polynomial. Similarly, we get the same conclusion for any $i \in \mathrm{var}(v) \setminus \mathrm{var}(u)$. We are now left with $i \in \mathrm{var}(u) \cap \mathrm{var}(v)$, for which we shall prove below.

Let us write $g = u^{\mathrm{e}_u(g)} \cdot v^{\mathrm{e}_v(g)} \cdot A$ and $h = u^{\mathrm{e}_u(h)} \cdot v^{\mathrm{e}_v(h)} \cdot B$, for some polynomials $A, B \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ co-prime to both u and v. Let $g' = u^{\mathrm{e}_u(g)} \cdot v^{\mathrm{e}_v(g)}$ and $h' = u^{\mathrm{e}_u(h)} \cdot v^{\mathrm{e}_v(h)}$. Further, let $z = \mathrm{e}_u(h) + \mathrm{e}_v(h)$ and $t = \mathrm{e}_u(g) + \mathrm{e}_v(g)$. Consider polynomials $g^z = (g')^z \cdot A^z$ and $h^t = (h')^t \cdot B^t$. Since both g, h have individual degree d, we know that $\mathrm{e}_u(g), \mathrm{e}_v(g), \mathrm{e}_u(h), \mathrm{e}_v(h) \leq d$ and hence $s, t \leq 2d$. Pick any $i \in \mathrm{var}(u) \cap \mathrm{var}(v)$ and consider $\gcd_{x_i}(g^z, h^t)$. Define $W \triangleq \gcd_{x_i}((g')^z, (h')^t)$ and $Y \triangleq \gcd_{x_i}(A^z, B^t)$. Since g', h' are co-prime to both A and B, we deduce that $\gcd_{x_i}(g^z, h^t) = W \cdot Y$. By our premise, we have $\det(E) \neq 0$. Without loss of generality, let us assume $\det(E) > 0$. The other case follows similarly. Using Lemma 19, we get that $(g')^z / W = u^{\det(E)}$ and $(h')^t / W = v^{\det(E)}$. Therefore, we can write $g^z = W \cdot Y \cdot u^{\det(E)} \cdot \frac{A^z}{Y}, h^t = W \cdot Y \cdot v^{\det(E)} \cdot \frac{B^t}{Y}$. Note that A^z / Y and B^t / Y are proper polynomials by definition of Y. Let $\ell = \deg_{x_i}(\gcd(g^z, h^t))$. By Theorem 14, there exists $k \geq 0$ such that:

$$S_{x_i}(\ell, g^z, h^t) = W \cdot Y \cdot \mathrm{Res}_{x_i}\left(u^{\det(E)} \cdot \frac{A^z}{Y},\ v^{\det(E)} \cdot \frac{B^t}{Y}\right) \cdot \mathrm{lc}_{x_i}(W \cdot Y)^k.$$

Since both g^z and h^t are s^{2d}-sparse with individual degree at most $2d^2$, by Observation 15, $S_{x_i}(\ell, g^z, h^t)$ is $(sd)^{\mathcal{O}(d^3)}$-sparse with individual degree at most $8d^4$. Furthermore, observe that by definition: $\gcd_{x_i}\left(u^{\det(E)} \cdot \frac{A^z}{Y},\ v^{\det(E)} \cdot \frac{B^t}{Y}\right) = 1 \implies \mathrm{Res}_{x_i}\left(u^{\det(E)} \cdot \frac{A^z}{Y},\ v^{\det(E)} \cdot \frac{B^t}{Y}\right) \not\equiv 0 \implies S_{x_i}(\ell, g^z, h^t) \not\equiv 0$. Finally, since $\det(E)$ is a positive integer, we use Lemma 11 to deduce that $\mathrm{Res}_{x_i}(u, v) \mid \mathrm{Res}_{x_i}\left(u^{\det(E)} \cdot \frac{A^z}{Y},\ v^{\det(E)} \cdot \frac{B^t}{Y}\right)$. Hence, we conclude that $\mathrm{Res}_{x_i}(u, v)$ is a factor of a non-zero $(sd)^{\mathcal{O}(d^3)}$-sparse sub-resultant polynomial. ◀

Using the above, we conclude that while the multiplicands in the polynomial $\Phi_{P,Q}$ may not themselves be sparse, they are factors of (some) sparse polynomials. Consequently, $\Phi_{P,Q}$ can be hit by a hitting set generator for sparse polynomials.

▶ **Lemma 21.** *Let both $P, Q \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ be products of s-sparse, individual degree-d polynomials and let $\mathcal{G}$ be a generator for $(sd)^{\mathcal{O}(d^3)}$-sparse, $\mathcal{O}(d^4)$-individual degree polynomials. Then $\Phi_{P,Q}(\mathcal{G}) \not\equiv 0$.*

Proof. Let $P = \prod_{j \in [r]} g_j$ and $Q = \prod_{k \in [m]} h_k$, where g_j, h_k are s-sparse polynomials of individual degree d, for all $j \in [r], k \in [m]$. By definition, $\Phi_{P,Q}$ has two types of multiplicands. We will show that $\mathcal{G}$ hits both types.

For the first type, let u, v be any irreducible factors of P or Q such that $(\mathrm{e}_u(P), \mathrm{e}_v(P))$ interlaces with $(\mathrm{e}_u(Q), \mathrm{e}_v(Q))$. We wish to show that $\mathrm{Res}_{x_i}(u,v) \not\equiv 0 \implies \mathrm{Res}_{x_i}(u,v)(\mathcal{G}) \not\equiv 0$. Observe: $\mathrm{e}_u(P) = \sum_{j=1}^{r} \mathrm{e}_u(g_j), \mathrm{e}_u(Q) = \sum_{k=1}^{m} \mathrm{e}_u(h_k), \mathrm{e}_v(P) = \sum_{j=1}^{r} \mathrm{e}_v(g_j), \mathrm{e}_v(Q) = \sum_{k=1}^{m} \mathrm{e}_v(h_k)$. By definition, each of these e-values is a non-negative integer. Therefore by Lemma 7, there exists $j \in [r], k \in [m]$ such that $E \triangleq \begin{bmatrix} \mathrm{e}_u(g_j) & \mathrm{e}_u(h_k) \\ \mathrm{e}_v(g_j) & \mathrm{e}_v(h_k) \end{bmatrix}$ has full rank. Then by Lemma 20, for any $i \in [n] : \mathrm{Res}_{x_i}(u,v)$ is factor of some $(sd)^{\mathcal{O}(d^3)}$-sparse, $\mathcal{O}(d^4)$-individual degree polynomial. Since, $\mathcal{G}$ is a generator for such polynomials, we deduce that $\mathrm{Res}_{x_i}(u,v)(\mathcal{G}) \not\equiv 0$.

For the second type, by multiplicative property of lc, we know that for any $i \in [n] : \mathrm{lc}_{x_i}(P) = \prod_{j\in[r]} \mathrm{lc}_{x_i}(g_j)$ and $\mathrm{lc}_{x_i}(P)(\mathcal{G}) = \prod_{j\in[r]} \mathrm{lc}_{x_i}(g_j)(\mathcal{G})$. Note that $\mathrm{lc}_{x_i}(g_j)$ is also s-sparse with individual degree d. Hence, $\mathrm{lc}_{x_i}(g_j)(\mathcal{G}) \not\equiv 0$, for all $j \in [r], i \in [n]$. This implies $\mathrm{lc}_{x_i}(P)(\mathcal{G}) \not\equiv 0$, for all $i \in [n]$ (whenever $\mathrm{lc}_{x_i}(P) \not\equiv 0$). Similarly, we can show $\mathrm{lc}_{x_i}(Q)(\mathcal{G}) \not\equiv 0$, for all $i \in [n]$. We conclude that $\Phi_{P,Q}(\mathcal{G}) \not\equiv 0$. ◀

By combining the result with Lemma 18 we obtain the following corollary.

▶ **Corollary 22.** *Let $P, Q \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ be products of s-sparse, individual degree-d polynomials such that $P \nsim Q$. Let $\mathcal{G}$ be a generator for $(sd)^{\mathcal{O}(d^3)}$-sparse, $\mathcal{O}(d^4)$-individual degree polynomials. Then there exists an $i \in [n]$ such that $P(x_i, \mathcal{G}_{-i}) \nsim Q(x_i, \mathcal{G}_{-i})$.*

Finally, we describe the black-box PIT algorithm for $\Sigma^{[2]}\Pi\Sigma\Pi^{[\text{ind-deg } d]}$ circuits. The correctness and the time complexity follow from Corollary 22. Hence, Theorem 1 follows from this Lemma. Due to space limitation we give the proof of Corollary 22 and the detailed analysis of Algorithm 1 in Section A.3.

▶ **Lemma 23.** *There exists a deterministic algorithm that given n, d, s and a black-box access to a $\Sigma^{[2]}\Pi\Sigma\Pi^{[\text{ind-deg } d]}$ circuit C of size s determines if $C \equiv 0$, in time $\mathrm{poly}((sd)^{d^3}, n)$. Algorithm 1 provides the outline.*

Algorithm 1 Black-box PIT algorithm for class $\Sigma^{[2]}\Pi\Sigma\Pi^{[\text{ind-deg } d]}$.

Input: A black-box access to a polynomial $f(x_1, \ldots, x_n) \in \Sigma^{[2]}\Pi\Sigma\Pi^{[\text{ind-deg } d]}$
Output: "ZERO", if f is identically zero and "NON-ZERO", otherwise.

```
Invoke [20] to get generator G of seed-length 1 for n-variate polynomials of sparsity
  ≤ (sd)^{O(d^3)} and individual degree ≤ O(d^4).
for i ← 1 to n do
    Compute the bivariate polynomial f(x_i, G_{-i}).
    Do brute-force black-box PIT for f(x_i, G_{-i}).
    if f(x_i, G_{-i}) ≢ 0 then return "NON-ZERO".
return "ZERO".
```

5 Future Directions

A lot of interesting open problems arise in the context of this work:

- Design a polynomial-time PIT algorithm for $\Sigma^{[k]}\Pi\Sigma\Pi^{[\text{ind-deg } d]}$ circuits with bounded k and d, for $k \geq 3$. To the best of our knowledge, the smallest open case is $k = 3$ and $d = 1$!
- Prove a polynomial-size sparsity bound (Conjecture 3) even for the special cases like exact-roots, multilinear co-factors.
  - In particular, improve the sparsity bound in Corollary 4. Ideally, get rid of the $\log s$ term in the exponent. One can start by studying the structure of polynomials with non-constant or log-sized unique projections.

- Can we show that $\mathrm{Res}_{x_i}(u, v)$ is actually sparse (or "somewhat sparse") under the premises of Lemma 20? This claim will be implied by a polynomial-size sparsity bound.

References

1. M. Agrawal, S. Ghosh, and N. Saxena. Bootstrapping variables in algebraic circuits. *Proceedings of the National Academy of Sciences*, 116(17):8107–8118, 2019.
2. M. Agrawal and V. Vinay. Arithmetic circuits: A chasm at depth four. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 67–75, 2008.
3. M. Ben-Or and P. Tiwari. A deterministic algorithm for sparse multivariate polynominal interpolation. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing (STOC)*, pages 301–309, 1988.
4. V. Bhargava, S. Saraf, and I. Volkovich. Deterministic factorization of sparse polynomials with bounded individual degree. *J. ACM*, 67(2):8:1–8:28, 2020.
5. P. Bisht and N. Saxena. Derandomization via symmetric polytopes: Poly-timefactorization of certain sparse polynomials. In *Proceedings of the 42nd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, 2022. URL: `https://www.cse.iitk.ac.in/users/nitin/papers/symmetricSparse.pdf`.
6. P. Bisht and I. Volkovich. On solving sparse polynomial factorization related problems. *Electron. Colloquium Comput. Complex.*, TR22-070, 2022. URL: `https://eccc.weizmann.ac.il/report/2022/070`.
7. B. Chor and R. L. Rivest. A knapsack-type public key cryptosystem based on arithmetic in finite fields. *IEEE Transactions on Information Theory*, 34(5):901–909, 1988.
8. D. A. Cox, J. Little, and D. O'Shea. *Ideals, varieties, and algorithms - an introduction to computational algebraic geometry and commutative algebra (4. ed.)*. Undergraduate texts in mathematics. Springer, 2015.
9. R. A. DeMillo and R. J. Lipton. A probabilistic remark on algebraic program testing. *Inf. Process. Lett.*, 7(4):193–195, 1978.
10. P. Dutta, P. Dwivedi, and N. Saxena. Deterministic identity testing paradigms for bounded top-fanin depth-4 circuits. In *36th Conference on Computational Complexity (CCC 2021)*, volume 5, page 9, 2021.
11. M. A. Forbes. Deterministic divisibility testing via shifted partial derivatives. In *FOCS*, 2015.
12. J. von zur Gathen and J. Gerhard. *Modern computer algebra*. Cambridge University Press, 1999.
13. J. von zur Gathen and E. Kaltofen. Factoring sparse multivariate polynomials. *Journal of Computer and System Sciences*, 31(2):265–287, 1985. `doi:10.1016/0022-0000(85)90044-3`.
14. K. O. Geddes, S. R. Czapor, and G. Labahn. *Algorithms for computer algebra*. Kluwer, 1992.
15. E. Grigorescu, K. Jung, and R. Rubinfeld. A local decision test for sparse polynomials. *Inf. Process. Lett.*, 110(20):898–901, 2010. `doi:10.1016/j.ipl.2010.07.012`.
16. V. Guruswami and M. Sudan. Improved decoding of reed-solomon codes and algebraic-geometry codes. *IEEE Transactions on Information Theory*, 45(6):1757–1767, 1999.
17. V. Kabanets and R. Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Computational Complexity*, 13(1-2):1–46, 2004.
18. E. Kaltofen. Factorization of polynomials given by straight-line programs. In S. Micali, editor, *Randomness in Computation*, volume 5 of *Advances in Computing Research*, pages 375–412. JAI Press Inc., Greenwhich, Connecticut, 1989.
19. E. Kaltofen and B. M. Trager. Computing with polynomials given by black boxes for their evaluations: Greatest common divisors, factorization, separation of numerators and denominators. *J. of Symbolic Computation*, 9(3):301–320, 1990.

20 A. Klivans and D. Spielman. Randomness efficient identity testing of multivariate polynomials. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 216–223, 2001.

21 N. Limaye, S. Srinivasan, and S. Tavenas. Superpolynomial lower bounds against low-depth algebraic circuits. In *FOCS 2021*, 2022.

22 S. Peleg and A. Shpilka. Polynomial time deterministic identity testing algorithm for $\Sigma^{[3]}\Pi\Sigma\Pi^{[2]}$ 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.

23 C. Saha, R. Saptharishi, and N. Saxena. A case of depth-3 identity testing, sparse factorization and duality. *Computational Complexity*, 22(1):39–69, 2013. doi:10.1007/s00037-012-0054-4.

24 S. Saraf and I. Volkovich. Blackbox identity testing for depth-4 multilinear circuits. *Combinatorica*, 38(5):1205–1238, 2018.

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

26 A. Shpilka and I. Volkovich. On the relation between polynomial identity testing and finding variable disjoint factors. In *Automata, Languages and Programming, 37th International Colloquium (ICALP)*, pages 408–419, 2010. Full version at https://eccc.weizmann.ac.il/report/2010/036.

27 V. Strassen. Vermeidung von divisionen. *J. of Reine Angew. Math.*, 264:182–202, 1973.

28 M. Sudan. Decoding of reed solomon codes beyond the error-correction bound. *Journal of Complexity*, 13(1):180–193, 1997.

29 I. Volkovich. Deterministically factoring sparse polynomials into multilinear factors and sums of univariate polynomials. In *APPROX-RANDOM*, pages 943–958, 2015.

30 I. Volkovich. On some computations on sparse polynomials. In *APPROX-RANDOM*, pages 48:1–4:21, 2017.

31 R. Zippel. Probabilistic algorithms for sparse polynomials. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, pages 216–226, 1979.

A Missing Proofs

Proof of Lemma 5. We prove the following claim: Let $f \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ be a multicubic polynomial such that $f = uv$ and $v \not\equiv 0$. Then $\|u\| \leq \|f\|^a$. Note that the claim also covers the case when $f = u \equiv 0$. The proof is by induction on n (the number of variables in f). The base case is when $n = 0$ (i.e. $u, f, v \in \mathbb{F}$) where the claim follows trivially. Suppose $n \geq 1$. We have the following cases to consider:

- There exists a variable x_i s.t. $\deg_{x_i}(u) \geq 1$ but $\deg_{x_i}(v) = 0$. Let $1 \leq d \leq 3$ be the degree of x_i in u. In this case we can write:

  $$(u_d x_i^d + \ldots + u_0)v = uv = f = f_d {x_i}^d + \ldots + f_0.$$

  Here, u_j, f_j and v do not depend on x_i. Formally: $f_j = u_j v$ for $j \in \{0, \ldots, d\}$. By the induction hypothesis, we have that $\|u_j\| \leq \|f_j\|^a$ for $j \in \{0, \ldots, d\}$ and hence:

  $$\|u\| = \sum_{j=0}^{d} \|u_j\| \leq \sum_{j=0}^{d} \|f_j\|^a \leq \left(\sum_{j=0}^{d} \|f_j\| \right)^a = \|f\|^a.$$

- There exists a variable x_i s.t. $\deg_{x_i}(v) \geq 1$, but $\deg_{x_i}(u) = 0$. Pick $\alpha \in \mathbb{F}$ such that $v|_{x_i=\alpha} \not\equiv 0$. We have that:

  $$u \cdot v|_{x_i=\alpha} = u|_{x_i=\alpha} \cdot v|_{x_i=\alpha} = f|_{x_i=\alpha}.$$

  By the induction hypothesis: $\|u\| \leq \|f|_{x_i=\alpha}\|^a \leq \|f\|^a$.

- There exists a variable x_i s.t. $\deg_{x_i}(u) = 1$. Wlog $\deg_{x_i}(v) \geq 1$. We can write

$$(u_1 x_i + u_0)(v_d x_i^d + \ldots + v_e x_i^e) = uv = f = (f_{d+1} x_i^{d+1} + \ldots + f_e x_i^e).$$

  Here, $d > e$ and $v_d, v_e \not\equiv 0$. In particular, we have that $u_1 v_d = f_{d+1}$ and $u_0 v_e = f_e$. By the induction hypothesis: $\|u\| = \|u_1\| + \|u_0\| \leq \|f_{d+1}\|^a + \|f_e\|^a \leq \|f\|^a$.
- WLOG we are left with the case that for each $i \in [n]$ we have that: $\deg_{x_i}(u) = 2$ and $\deg_{x_i}(v) = 1$. Based on our assumption, in this case $\|u\| \leq \|f\|^a$ and we are done. ◀

We state the following important connection between projection of resultant and resultant of projections.

▶ **Lemma 24.** *Let $f, g \in \mathbb{F}[y, \mathbf{x}]$ be two polynomials and let $\mathbf{a} \in \mathbb{F}^n$. Then,*

$$\mathrm{Res}_y(f, g)(\mathbf{a}) \neq 0 \implies \mathrm{Res}_y(f(\mathbf{a}), g(\mathbf{a})) \neq 0.$$

Proof. Let $d \triangleq \deg_y(f)$, $e \triangleq \deg_y(g)$, $r \triangleq \deg_y(f(\mathbf{a}))$ and $t \triangleq \deg_y(g(\mathbf{a}))$. Then with some easy determinant calculations, one can show that:

$$\mathrm{Res}_y(f, g)(\mathbf{a}) = \begin{cases} \mathrm{Res}_y(f(\mathbf{a}), g(\mathbf{a})) & r = d, t = e \\ (\mathrm{lc}_y(f)(\mathbf{a}))^{e-t} \cdot \mathrm{Res}_y(f(\mathbf{a}), g(\mathbf{a})) & r = d, t < e \\ (-1)^{e(d-r)} \cdot (\mathrm{lc}_y(g)(\mathbf{a}))^{d-r} \cdot \mathrm{Res}_y(f(\mathbf{a}), g(\mathbf{a})) & r < d, t = e \\ 0 & r < d, t < e \end{cases}$$

Note that if $\mathrm{Res}_y(f, g)(\mathbf{a}) \neq 0$, then $\mathrm{Res}_y(f(\mathbf{a}), g(\mathbf{a}))$ divides it and hence the conclusion follows. ◀

Lemma 10 follows from Lemma 9 and Lemma 24.

Proof of Lemma 19. We have that: $g^z = (u^{a_1} \cdot v^{b_1})^z = u^{a_1 a_2 + a_1 b_2} \cdot v^{a_2 b_1 + b_1 b_2}$ and $h^t = (u^{a_2} \cdot v^{b_2})^t = u^{a_1 a_2 + a_2 b_1} \cdot v^{a_1 b_2 + b_1 b_2}$. If $\det(E) \geq 0$, then $a_1 b_2 \geq a_2 b_1$ and consequently $W = u^{a_1 a_2 + a_2 b_1} \cdot v^{a_2 b_1 + b_1 b_2}$. In that case, $g^z / W = u^{a_1 b_2 - a_2 b_1} = u^{\det(E)}$ and $h^t / W = v^{a_1 b_2 - a_2 b_1} = v^{\det(E)}$. Otherwise, if $\det(E) < 0$, then $a_2 b_1 > a_1 b_2$ and consequently $W = u^{a_1 a_2 + a_1 b_2} \cdot v^{a_1 b_2 + b_1 b_2}$. Then $g^z / W = v^{a_2 b_1 - a_1 b_2} = v^{-\det(E)}$ and $h^t / W = u^{a_2 b_1 - a_1 b_2} = u^{-\det(E)}$. ◀

A.1 Proof of Theorem 14

In this section we give the proof of Theorem 14, from which Theorem 2 follows. We start by stating below some known results in the theory of subresultants, which will be useful for us.

▶ **Lemma 25** (Lem 7.1 of [14]). *Let $A(x), B(x) \in \mathcal{R}[x]$ be two polynomials over an arbitrary UFD $\mathcal{R}$. Let $\mathcal{K}$ be the field of fractions of $\mathcal{R}$. Suppose $A(x) = Q(x) \cdot B(x) + R(x)$, for some polynomials $Q, R \in \mathcal{K}[x]$ such that $\deg_x(A) = m$, $\deg_x(B) = n$, $\deg_x(Q) = m - n$, $\deg_x(R) = k$ and $m \geq n > k$. Let b and r denote the leading coefficients of $B(x)$ and $R(x)$ respectively. Then*

$$S_x(j, A, B) = (-1)^{(m-j)(n-j)} \times \begin{cases} b^{m-k} \cdot S_x(j, B, R) & 0 \leq j < k \\ b^{m-k} \cdot r^{n-k-1} \cdot R(x) & j = k \\ 0 & k < j < n-1 \\ b^{m-n+1} \cdot R(x) & j = n-1. \end{cases}$$

That is, $S_x(j, A, B)$ equals to one of the above four expressions multiplied by the corresponding sign $(-1)^{(m-j)(n-j)}$.

For the sake of completeness, we restate Theorem 14 below.

▶ **Theorem.** *Let $A(x), B(x) \in \mathcal{R}[x]$ be two polynomials over an arbitrary UFD $\mathcal{R}$. Suppose $A(x) = f(x) \cdot g(x)$ and $B(x) = h(x) \cdot g(x)$ with $\deg_x(A) = m$, $\deg_x(B) = n$, $\deg_x(g) = d$, $\deg_x(f) = m' = m - d$ and $\deg_x(h) = n' = n - d$. Then*

$$S_x(d, A, B) = g \cdot \mathrm{Res}_x(f, h) \cdot \mathrm{lc}_x(g)^{m'+n'-1}.$$

Proof of Theorem 14. Let $\mathcal{K}$ be the field of fractions of UFD $\mathcal{R}$. Consider Euclidean division of A by B in $\mathcal{K}[x]$ so that we get $A(x) = Q(x) \cdot B(x) + R(x)$, for some polynomials $Q, R \in \mathcal{K}[x]$ such that $\deg_x(R) < \deg_x(B)$. Note that since g divides both A and B, it must also divide R. Therefore, $R = g \cdot p$ for some polynomial $p(x) \in \mathcal{K}[x]$. Thus, we also get

$$f(x) = Q(x) \cdot h(x) + p(x) \tag{A.1}$$

Let $\deg_x(R) = k$ for some $k < n$ and let $\deg_x(p) = k' = k - d$. Now, we prove the theorem by induction on $\deg_x(p)$.

Base case. $\deg_x(p) = k' = 0$. In other words, $\deg_x(R) = k = d$. Thus using second case of Lemma 25, we get that:

$$\begin{aligned}
S_x(d, A, B) &= (-1)^{(m-d)(n-d)} \cdot b^{m-k} \cdot r^{n-k-1} \cdot R \\
&= (-1)^{m'.n'} \cdot \mathrm{lc}_x(h)^{m-k} \cdot \mathrm{lc}_x(g)^{m-k} \cdot \mathrm{lc}_x(p)^{n-k-1} \cdot \mathrm{lc}_x(g)^{n-k-1} \cdot pg \\
&= (-1)^{m'.n'} \cdot g \cdot \mathrm{lc}_x(h)^{m-k} \cdot \mathrm{lc}_x(p)^{n-k} \cdot \mathrm{lc}_x(g)^{m+n-2k-1} \\
S_x(d, A, B) &= (-1)^{m'.n'} \cdot g \cdot \mathrm{lc}_x(h)^{m-k} \cdot \mathrm{lc}_x(p)^{n-k} \cdot \mathrm{lc}_x(g)^{m'+n'-1}
\end{aligned} \tag{A.2}$$

The second last step above follows because $p = \mathrm{lc}_x(p)$ when $\deg_x(p) = 0$. Now, we shall compute $\mathrm{Res}_x(f, h)$. Note that $\mathrm{Res}_x(f, h) = S_x(0, f, h)$ by definition of subresultant. Considering (A.1) with $\deg_x(p) = 0$, we can use second case of Lemma 25 to get:

$$\begin{aligned}
S_x(0, f, h) &= (-1)^{(\deg_x(f)-0).(\deg_x(h)-0)} \cdot \mathrm{lc}_x(h)^{\deg_x(f)-\deg_x(p)} \cdot \mathrm{lc}_x(p)^{\deg_x(h)-\deg_x(p)-1} \cdot p \\
&= (-1)^{m'.n'} \cdot \mathrm{lc}_x(h)^{m'} \cdot \mathrm{lc}_x(p)^{n'-1} \cdot p \quad [\text{as } \deg_x(p) = 0] \\
&= (-1)^{m'.n'} \cdot \mathrm{lc}_x(h)^{m'} \cdot \mathrm{lc}_x(p)^{n'} \quad [\text{as } p = \mathrm{lc}_x(p)] \\
\mathrm{Res}_x(f, h) &= (-1)^{m'.n'} \cdot \mathrm{lc}_x(h)^{m-k} \cdot \mathrm{lc}_x(p)^{n-k}
\end{aligned} \tag{A.3}$$

(A.2) and (A.3) together yield $S_x(d, A, B) = g \cdot \mathrm{Res}_x(f, h) \cdot \mathrm{lc}_x(g)^{m'+n'-1}$ for the base case.

Induction step. Now, we assume $\deg_x(p) = k' > 1$. In other words, $\deg_x(R) = k > d$. Therefore, by first case of Lemma 25:

$$\begin{aligned}
S_x(d, A, B) &= (-1)^{(m-d)(n-d)} \cdot b^{m-k} \cdot S_x(d, B, R) \\
&= (-1)^{m'.n'} \cdot \mathrm{lc}_x(h)^{m-k} \cdot \mathrm{lc}_x(g)^{m-k} \cdot S_x(d, B, R)
\end{aligned} \tag{A.4}$$

Now consider Euclidean division of B by R in $\mathcal{K}[x]$ to get

$$B(x) = Q'(x) \cdot R(x) + R'(x) \tag{A.5}$$

for some polynomial $R'(x) \in \mathcal{K}[x]$ with $\deg_x(R') < \deg_x(R)$. Since g divides both B and R, we deduce that g must also divide R'. Let $R' = g \cdot p'$ for some polynomial $p' \in \mathcal{K}[x]$. Thus from (A.5), we also get

$$h(x) = Q'(x) \cdot p(x) + p'(x) \tag{A.6}$$

In (A.5) since $\deg_x(R') < \deg_x(R)$ or equivalently $\deg_x(p') < \deg_x(p)$, we can use induction hypothesis to deduce that,

$$S_x(d, B, R) = g \cdot \mathrm{Res}_x(h, p) \cdot \mathrm{lc}_x(g)^{n'+k'-1} \tag{A.7}$$

Note that $\deg_x(p) = k' > 0$ in induction step, thus we can use first case of Lemma 25 on (A.1) to get

$$\begin{aligned}
\mathrm{Res}_x(f, h) &= S_x(0, f, h) \\
&= (-1)^{(\deg_x(f)-0)(\deg_x(h)-0)} \cdot \mathrm{lc}_x(h)^{\deg_x(f)-\deg_x(p)} \cdot S_x(0, h, p) \\
&= (-1)^{m'.n'} \cdot \mathrm{lc}_x(h)^{m'-k'} \cdot \mathrm{Res}_x(h, p) \\
\mathrm{Res}_x(h, p) &= \frac{\mathrm{Res}_x(f, h)}{(-1)^{m'.n'} \cdot \mathrm{lc}_x(h)^{m'-k'}}.
\end{aligned} \tag{A.8}$$

Substituting (A.8) in (A.7), we get:

$$S_x(d, B, R) = g \cdot \frac{\mathrm{Res}_x(f, h)}{(-1)^{m'.n'} \cdot \mathrm{lc}_x(h)^{m'-k'}} \cdot \mathrm{lc}_x(g)^{n'+k'-1} \tag{A.9}$$

Substituting (A.9) back into (A.4), we get

$$\begin{aligned}
S_x(d, A, B) &= (-1)^{m'.n'} \cdot \mathrm{lc}_x(h)^{m-k} \cdot \mathrm{lc}_x(g)^{m-k} \cdot g \cdot \frac{\mathrm{Res}_x(f, h)}{(-1)^{m'.n'} \cdot \mathrm{lc}_x(h)^{m'-k'}} \cdot \mathrm{lc}_x(g)^{n'+k'-1} \\
&= \mathrm{lc}_x(g)^{m-k} \cdot g \cdot \mathrm{Res}_x(f, h) \cdot \mathrm{lc}_x(g)^{n'+k'-1} \quad [\text{as } m-k = m'-k'] \\
&= g \cdot \mathrm{Res}_x(f, h) \cdot \mathrm{lc}_x(g)^{m-k+n'+k'-1} \\
&= g \cdot \mathrm{Res}_x(f, h) \cdot \mathrm{lc}_x(g)^{m'+n'-1} \quad [\text{as } m-k+k' = m-d = m']
\end{aligned}$$

This completes the proof of induction step, as well as that of the theorem. ◀

A.2 Proof of Lemma 7

Proof of Lemma 7. Without loss of generality, suppose $\sum_{i=1}^{n} a_i > \sum_{j=1}^{m} c_j$ and $\sum_{i=1}^{n} b_i < \sum_{j=1}^{m} d_j$. In particular, there exists $s \in [n], t \in [m]$ such that $a_s, d_t > 0$. Consider the $2 \times (n+m)$ matrix,

$$E \triangleq \begin{bmatrix} a_1 & \cdots & a_n & c_1 & \cdots & c_m \\ b_1 & \cdots & b_n & d_1 & \cdots & d_m \end{bmatrix}.$$

Suppose E is not full-rank. Since $\mathbf{a}$ and $\mathbf{d}$ are non-zero vectors, E is not the zero matrix and hence it must have rank 1. In that case, the first row is linearly dependent on the second row. Since all E's entries are non-negative, there exist $\alpha, \beta > 0$ such that $\alpha \cdot \mathbf{a} = \beta \cdot \mathbf{b}$ and $\alpha \cdot \mathbf{c} = \beta \cdot \mathbf{d}$. This implies:

$$\alpha \cdot \left(\sum_{i=1}^{n} a_i \right) = \beta \cdot \left(\sum_{i=1}^{n} b_i \right), \quad \alpha \cdot \left(\sum_{j=1}^{m} c_j \right) = \beta \cdot \left(\sum_{j=1}^{m} d_j \right).$$

Which in turn implies:

$$\beta \cdot \left(\sum_{i=1}^{n} b_i \right) = \alpha \cdot \left(\sum_{i=1}^{n} a_i \right) > \alpha \cdot \left(\sum_{j=1}^{m} c_j \right) = \beta \cdot \left(\sum_{j=1}^{m} d_j \right) \implies \sum_{i=1}^{n} b_i > \sum_{j=1}^{m} d_j$$

This contradicts the interlacing property. Hence, E must be full-rank. Let E' be a 2×2 rank-2 minor of E. If E' is of the form $E' = \begin{bmatrix} a_i & c_j \\ b_i & d_j \end{bmatrix}$ for some i, j we are done. Otherwise, suppose if E' is of the form $E' = \begin{bmatrix} a_i & a_j \\ b_i & b_j \end{bmatrix}$ or $E' = \begin{bmatrix} c_i & c_j \\ d_i & d_j \end{bmatrix}$ then by the exchange property, we can exchange one of the columns in E' with non-zero columns $\begin{bmatrix} c_t \\ d_t \end{bmatrix}$ or $\begin{bmatrix} a_s \\ b_s \end{bmatrix}$, respectively, to get a rank-2 minor of the required form. ◀

A.3 Proof of Theorem 1

In this section we formally prove our main result - Theorem 1. We begin by restating and proving Corollary 22.

▶ **Corollary** (Corollary 22). *Let $P, Q \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ be products of s-sparse, individual degree-d polynomials such that $P \nsim Q$. Let $\mathcal{G} = (\mathcal{G}_1, \ldots, \mathcal{G}_n) : \mathbb{F}^t \to \mathbb{F}^n$ be a generator for $(sd)^{\mathcal{O}(d^3)}$-sparse, $\mathcal{O}(d^4)$-individual degree polynomials. Then there exists an $i \in [n]$ such that $P(x_i, \mathcal{G}_{-i}) \nsim Q(x_i, \mathcal{G}_{-i})$.*

Proof of Corollary 22. By Lemma 21, we get that $\Phi_{P,Q}(\mathcal{G}) \not\equiv 0$. We deduce that there exists a set $W \subseteq \mathbb{F}$ of large enough size such that $\mathcal{G}(W^t) \subseteq \mathbb{F}^n$ is a hitting set for $\Phi_{P,Q}$. In particular, there exists $\mathbf{b} \in W^t$ such that for $\mathbf{a} \triangleq \mathcal{G}(\mathbf{b})$, we have $\Phi_{P,Q}(\mathbf{a}) \neq 0$. By Lemma 18, there exists an $i \in [n]$ such that $P(x_i, \mathbf{a}_{-i}) \nsim Q(x_i, \mathbf{a}_{-i})$. Now suppose $P(x_i, \mathcal{G}_{-i}) \sim Q(x_i, \mathcal{G}_{-i})$. Then have that $P\big(\mathcal{G}_1(\mathbf{b}), \ldots, \mathcal{G}_{i-1}(\mathbf{b}), x_i, \mathcal{G}_{i+1}(\mathbf{b}), \ldots, \mathcal{G}_n(\mathbf{b})\big) \sim Q\big(\mathcal{G}_1(\mathbf{b}), \ldots, \mathcal{G}_{i-1}(\mathbf{b}), x_i, \mathcal{G}_{i+1}(\mathbf{b}), \ldots, \mathcal{G}_n(\mathbf{b})\big)$. This implies that $P(x_i, \mathbf{a}_{-i}) \sim Q(x_i, \mathbf{a}_{-i})$, which is a contradiction. Hence, $P(x_i, \mathcal{G}_{-i}) \nsim Q(x_i, \mathcal{G}_{-i})$. ◀

For completeness, we restate Theorem 1.

▶ **Theorem.** *There exists a deterministic algorithm that given n, d, s and a black-box access to a $\Sigma^{[2]}\Pi\Sigma\Pi^{[\text{ind-deg } d]}$ circuit C of size s determines if $C \equiv 0$, in time $\text{poly}((sd)^{d^3}, n)$. The algorithm below provides the outline.*

Algorithm 2 Black-box PIT algorithm for class $\Sigma^{[2]}\Pi\Sigma\Pi^{[\text{ind-deg } d]}$.

Input: A polynomial $f(x_1, \ldots, x_n) \in \Sigma^{[2]}\Pi\Sigma\Pi^{[\text{ind-deg } d]}$, where bottom $\sum\prod$ computes s-sparse polynomials of individual degrees $\leq d$.
Output: ZERO, if f is identically zero and NON-ZERO, otherwise.

```
Invoke [20] to get generator G of seed-length 1 for n-variate polynomials of sparsity
  ≤ (sd)^{O(d^3)} and individual degree ≤ O(d^4).
for i ← 1 to n do
    Compute the bivariate polynomial f(x_i, G_{-i}).
    Do brute-force black-box PIT for f(x_i, G_{-i}).
    if f(x_i, G_{-i}) ≢ 0 then
        return NON-ZERO.
return ZERO.
```

Proof. We now analyze the correctness and runtime complexity of the Algorithm.

Correctness. Note that $f \equiv 0 \implies f(x_i, \mathcal{G}_{-i}) \equiv 0$ trivially, for all $i \in [n]$. Thus, the algorithm outputs ZERO in this case, as desired. Now suppose $f \not\equiv 0$. Let $f = P + Q$, where both P, Q are product of s-sparse, individual degree d polynomials. If $P \nsim Q$, then Corollary 22 implies that there exists an $i \in [n]$ such that $P(x_i, \mathcal{G}_{-i}) \nsim Q(x_i, \mathcal{G}_{-i})$. In particular, $P(x_i, \mathcal{G}_{-i}) \neq -Q(x_i, \mathcal{G}_{-i})$. Since $f(x_i, \mathcal{G}_{-i}) = P(x_i, \mathcal{G}_{-i}) + Q(x_i, \mathcal{G}_{-i})$, we deduce that $f(x_i, \mathcal{G}_{-i}) \not\equiv 0$ in this case. Now suppose $f \not\equiv 0$ but $P \sim Q$. Let $P = cQ$, for some $c \in \mathbb{F}$. Then $f = (c+1)Q$, where $c \neq -1$ and $Q \not\equiv 0$. This means that there exists an $i \in [n]$ such that $\mathrm{lc}_{x_i}(Q) \not\equiv 0$. Using Lemma 21, we know that $\Phi_{P,Q}(\mathcal{G}) \not\equiv 0$ and thus by definition of $\Phi_{P,Q}$, we get $\mathrm{lc}_{x_i}(Q)(\mathcal{G}) \not\equiv 0$. This implies that $Q(x_i, \mathcal{G}_{-i}) \not\equiv 0$. We conclude that $f(x_i, \mathcal{G}_{-i}) = (c+1)Q(x_i, \mathcal{G}_{-i}) \not\equiv 0$. Thus whenever $f \not\equiv 0$, the algorithm outputs NON-ZERO.

Time complexity. By [20], degree of generator $\mathcal{G}$ is $\mathrm{poly}((sd)^{d^3}, n)$. Note that f has individual degree at most sd and thus $f(x_i, \mathcal{G}_{-i})$ has individual degree $\leq sd.\deg(\mathcal{G})$. Testing non-zeroness of the bivariate polynomial $f(x_i, \mathcal{G}_{-i})$ takes only $\mathrm{poly}((sd)^{d^3}, n)$ time. The n iterations only add a factor of n. ◀