

Factoring Products of Sparse Irreducibles of Bounded Individual Degree via Rational Interpolation

Aminadav Chuyoon* Amir Shpilka*

Abstract

We design a deterministic algorithm that, given blackbox access to the product $f = \prod_{i=1}^{\ell} h_i$ of ℓ irreducible s -sparse n -variate polynomials of bounded individual degree d , recovers the h_i 's and their multiplicities. The algorithm makes $\text{poly}(n, (s\ell)^d)$ field and bit operations. For any constant $d > 2$, this is the first *polynomial-time* algorithm for this problem, resolving for the bounded-individual-degree regime an open question of Dutta, Sinhababu, and Thierauf (Random 2024). The previous best deterministic algorithm, due to the authors, runs in $\text{poly}(n, d^d, s^{d \log \ell}, \ell^d)$ time, which is quasi-polynomial in ℓ .

The improvement is enabled by a new sparse rational interpolation theorem in the bounded-individual-degree setting. Given blackbox access to rational functions $a_1/b, \dots, a_N/b$ where the a_j 's and b are s -sparse of individual degree at most d with $\gcd(a_1, \dots, a_N, b) = 1$, we recover the a_j 's and b in time $\text{poly}(n, d, s^d, N)$. In contrast to the rational interpolation theorem of Chuyoon-Shpilka (CCC 2026), our algorithm reconstructs the denominator directly and does not require a precomputed list of its irreducible factors.

*Blavatnik School of Computer Science and AI, Tel Aviv University, chuyoon1@mail.tau.ac.il, shpilka@tauex.tau.ac.il. 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.

Contents

1	Introduction	1
1.1	Background and motivation	1
1.2	Our results	2
1.3	Proof overview	3
1.4	Related work	6
1.5	AI disclosure	6
1.6	Organization	7
2	Preliminaries	7
2.1	Basic polynomial algorithms	7
2.2	Reverse-monic Polynomials and Normalization	9
2.3	The generator of [CS26b]	9
3	Coprimality certificate	10
4	Bounded-individual-degree sparse rational interpolation	12
4.1	Bounded-individual-degree sparse rational interpolation in characteristic zero	15
4.1.1	Computing the logarithmic derivatives	15
4.1.2	Reconstruction from the logarithmic derivatives	16
4.1.3	Putting it together	16
5	Algorithmic tools from [CS26b], and improvements	17
6	Factoring products of sparse irreducibles	19
	Bibliography	22

1 Introduction

1.1 Background and motivation

This paper continues the line of work, going back to von zur Gathen and Kaltofen [vzGK85], on factorization of *sparse* polynomials. An n -variate polynomial is called s -sparse if it is a sum of at most s monomials, and has *bounded individual degree* d if each variable appears in it with degree at most d . We write $\mathcal{P}(n, s, d)$ for the class of n -variate s -sparse polynomials of individual degree at most d , and $\mathcal{P}_{\text{Factors}}(n, s, d)$ for the class of their factors. Sparse polynomials form one of the most basic subclasses of polynomials in algebraic complexity: they are precisely those polynomials computed by polynomial-size $\Sigma\Pi$ circuits, and have received considerable attention from the perspectives of polynomial identity testing [KS01], reconstruction [BT88, GKS90], and factorization [vzGK85, Kal85a, Vol15, Vol17, BSV20, KRS24, DST24, BV25, BKR⁺25, CS26b]. A central problem in this area is the *efficient deterministic factorization* of multivariate polynomials whose irreducible factors are sparse. In fact, this problem is interesting even in the bounded-individual-degree case. Concretely, Dutta, Sinhababu, and Thierauf [DST24] ask the following.

Question 1.1 ([DST24], Section 6, item 4). *Given a blackbox computing the product of ℓ sparse irreducible polynomials $f_1, \dots, f_\ell$ of bounded individual degree d , find the f_i 's in deterministic polynomial time.*

This question is open even when $d = 3$. Several special cases have been resolved. Volkovich [Vol15] gave a polynomial-time algorithm for the case where all f_i 's are multilinear, and in [Vol17] he gave an efficient deterministic algorithm for factoring multiquadratic sparse polynomials (i.e., when the product itself is multiquadratic and sparse). In [CS26b] the authors solve the problem for $d = 2$. In [BSV20] the authors proved a sparsity bound on factors of $\mathcal{P}(n, s, d)$ polynomials and used it to obtain a deterministic quasi-polynomial-time algorithm for factoring sparse polynomials of bounded individual degree. Kumar, Ramanathan, and Saptharishi [KRS24] and Dutta, Sinhababu, and Thierauf [DST24] gave efficient deterministic reconstruction of constant degree factors of sparse polynomials. Recently, in [CS26b] the authors obtained the following.

Theorem 1.2 ([CS26b], Theorem 1.10). *Let $\mathbb{F}$ be a field of characteristic zero or larger than $2d$. There is a deterministic $\text{poly}(n, s^{d \log \ell}, (\ell d)^d)$ -time algorithm which, given blackbox access to a product $f = \prod_{r=1}^{\ell} h_r$, where $h_1, \dots, h_\ell \in \mathcal{P}(n, s, d)$ are irreducible, returns the h_r 's. Over arbitrary fields the algorithm runs in $\text{poly}(n, (d^2)!, s^{d \log \ell + d^3}, \ell^d)$ time.*

Observe that the running time of Theorem 1.2 is *quasi-polynomial* as it contains the factor $s^{d \log \ell}$. Theorem 1.2 therefore answers Question 1.1 only when $\ell = O(1)$. One of the main contributions of this paper is a deterministic polynomial-time algorithm for Question 1.1 when the individual degree d is constant.

An important ingredient in the algorithm of [CS26b] is a solution to the sparse rational interpolation problem in the bounded-individual-degree case. This problem concerns the reconstruction of a rational function from blackbox access, given that its numerator and denominator are sparse polynomials. Surprisingly, the best known deterministic algorithm for the unrestricted degree version of this question runs in exponential time [GKS94]. In [CS26b], the authors solved the following variants, where a list of candidate irreducible factors of the denominator is supplied as input:

Theorem 1.3 ([CS26b], Theorem 5.2). *Let $\mathbb{F}$ be a field of characteristic zero or larger than $2d$. Suppose we have blackbox access to rational functions $\{a_j f^j / b\}_{1 \leq j \leq N}$, where:*

- (1) $a_j \in \mathcal{P}(n, s, d)$ for every j .

- (2) f is a product of ℓ polynomials in $\mathcal{P}_{\text{Factors}}(n, s, d)$, accessible as a blackbox.
- (3) $\mathcal{F}$ is a set of s -sparse irreducible factors of f , supplied as input.
- (4) b is an s -sparse factor of f whose irreducible factors are in $\mathcal{F}$.
- (5) $\gcd(a_1, \dots, a_N, b) = 1$.

Then there is a deterministic algorithm that recovers $a_1, \dots, a_N$ and b in time $\text{poly}(n, d!, s^d, N, |\mathcal{F}|, \ell)$.

The factorization algorithm of [CS26b] applies a somewhat weaker version of Theorem 1.3 recursively: at each level, a precomputed list $\mathcal{F}$ of *all* the s -sparse divisors of an auxiliary free-term polynomial f_0 is produced using recursion, and a divisibility test is then used to produce the denominator b . This enumeration is the source of the $s^{d \log \ell}$ factor in Theorem 1.2.

The second main result of this work is an efficient algorithm for the bounded-individual-degree sparse rational interpolation problem that requires no precomputed information about the factors of b .

1.2 Our results

We first make the following convention to simplify the presentation.

Remark 1.4. We count field operations and auxiliary bit operations, and account separately for coefficient bit complexity. We assume that sufficiently many evaluation elements are available. If $\mathbb{F}$ is too small, we work over an explicitly represented finite extension $\mathbb{E}/\mathbb{F}$ of sufficiently large size and assume that the blackboxes support evaluations over $\mathbb{E}$.

Our first main result resolves Question 1.1 in the bounded-individual-degree regime for any constant d .

Theorem 1.5 (Factoring products of sparse irreducibles). *Let $\mathbb{F}$ be an arbitrary field. There is a deterministic algorithm which, given blackbox access to a product $f = \prod_{i=1}^{\ell} h_i$, where $h_1, \dots, h_{\ell} \in \mathcal{P}(n, s, d)$ are (not necessarily distinct) irreducible polynomials, returns (scalar multiples of) the h_r 's together with their multiplicities. The algorithm performs $\text{poly}(n, (s\ell)^d)$ many field and bit operations.*

Comparing with Theorem 1.2, the $s^{d \log \ell}$ factor is replaced by s^d (by [CS26b, Theorem 1.8], an (n, s, d) -sparse polynomial may have as many as s^d sparse divisors, which may explain this factor). In particular, this resolves Question 1.1 for constant d .

Theorem 1.5 is obtained from a sparse rational interpolation theorem for rational functions in which both numerator and denominator are from $\mathcal{P}(n, s, d)$. The rational interpolation theorem of [CS26b, Theorem 5.2] (see Theorem 1.3) assumes as input a list $\mathcal{F}$ of s -sparse irreducible polynomials containing every irreducible factor of the denominator b . Our new theorem removes the assumption entirely.

Theorem 1.6 (Bounded-individual-degree sparse rational interpolation). *Let $\mathbb{F}$ be an arbitrary field. Suppose we have blackbox access¹ to rational functions of the form*

$$\frac{a_j(\mathbf{x})}{b(\mathbf{x})}, \quad 1 \leq j \leq N,$$

¹We assume that the blackbox returns $\perp$ on zeroes of b .

where $a_1, \dots, a_N, b \in \mathcal{P}(n, s, d)$ and $\gcd(a_1, \dots, a_N, b) = 1$. Then there is a deterministic algorithm that performs $\text{poly}(n, s^d, N)$ bit and field operations and recovers $\lambda a_1, \dots, \lambda a_N, \lambda b$ for some $\lambda \in \mathbb{F} \setminus \{0\}$.

In particular, the case $N = 1$ solves the sparse rational interpolation problem in the case of coprime sparse polynomials of bounded individual degree.

The common scalar λ in [Theorem 1.6](#) can be avoided if we assume some normalization of the input polynomials.

1.3 Proof overview

We give two different proofs of [Theorem 1.6](#).² One holds for fields of characteristic zero and the other for arbitrary fields. Since we think that the characteristic zero proof contains an idea that may be useful in other settings, we decided to give both proofs. We next describe the idea common to both proofs and how each of them handles the resulting difficulties.

The basic observation is that if all the a_j and b were univariate, then the rational interpolation problem could be solved using simple linear algebra and univariate polynomial computations. Indeed, suppose we are given rational functions $a_1(u)/b(u), \dots, a_N(u)/b(u)$, with $\gcd(a_1, \dots, a_N, b) = 1$. For each function a_j/b , we set up a linear system $\phi_j \cdot (a_j/b) = \psi_j$ by evaluating at sufficiently many points where the fraction is defined. Here ϕ_j and ψ_j are univariates of appropriate degrees: for ϕ_j we use an upper bound on $\deg(b)$, and for ψ_j we use an upper bound on $\deg(a_j)$. Finding a nonzero solution and cancelling the gcd of ϕ_j and ψ_j lets us recover each rational function in reduced form. It is not hard to see that the monic least common multiple of their denominators is $b/\text{lc}(b)$. Indeed, every irreducible factor of b is absent from at least one numerator, so its full multiplicity survives in at least one reduced denominator.

For multivariate polynomials, we would like to apply this observation one variable at a time by substituting for $n - 1$ coordinates using an appropriate generator that allows interpolation, such as a suitably parameterized Klivans–Spielman generator. We also need the generator to preserve the relevant joint coprimality when one coordinate is left free. The difficulty is that the generator variables become part of the coefficient field. Consequently, univariate reconstruction determines the denominator only up to an unknown rational function of those variables.

More precisely, consider a generator $\Gamma(\mathbf{z}) = (\Gamma_1(\mathbf{z}), \dots, \Gamma_n(\mathbf{z})) : \mathbb{F}^k \rightarrow \mathbb{F}^n$, where k is constant. For each i , leave the i th coordinate free and define

$$b_i(\mathbf{z}, u) := b(\Gamma_1(\mathbf{z}), \dots, \Gamma_{i-1}(\mathbf{z}), u, \Gamma_{i+1}(\mathbf{z}), \dots, \Gamma_n(\mathbf{z})) \in \mathbb{F}(\mathbf{z})[u].$$

Assuming the generator Γ preserves coprimality of the input polynomials, the idea described above lets us reconstruct the polynomials b_i up to their leading coefficients $\text{lc}_u(b_i)$, that is, to reconstruct

$$B_i(\mathbf{z}, u) := \frac{b_i(\mathbf{z}, u)}{\text{lc}_u(b_i)}.$$

The remaining task is to combine these locally normalized polynomials to recover $b \circ \Gamma$ up to a scalar in $\mathbb{F}$. This is where the two proofs diverge.

Characteristic zero approach: logarithmic derivatives. Here, the key observation is that taking the logarithmic derivative with respect to the selected variable, is oblivious to whether we

²See [Section 1.5](#) for an account of the development of the proofs.

divided by the leading coefficient or not. Ideally, we would like to learn $B(\mathbf{z}) := b \circ \Gamma$. We instead have access to $B_i(\mathbf{z}, u) = b_i / \text{lc}_u(b_i)$. Since $\text{lc}_u(b_i)$ is independent of u ,

$$\frac{\partial_u B_i}{B_i} = \frac{\partial_u b_i}{b_i}.$$

Substituting $u = \Gamma_i(\mathbf{z})$ therefore gives access to

$$L_i(\mathbf{z}) := \left. \frac{\partial_u B_i}{B_i} \right|_{u=\Gamma_i(\mathbf{z})} = \frac{(\partial_{x_i} b)(\Gamma(\mathbf{z}))}{b(\Gamma(\mathbf{z}))}.$$

By the chain rule, the rational functions L_i determine every logarithmic derivative of B :

$$\mathcal{L}_\zeta := \frac{\partial_\zeta B}{B} = \sum_{i=1}^n L_i(\mathbf{z}) \partial_\zeta \Gamma_i(\mathbf{z}), \quad \zeta \in \{z_1, \dots, z_k\}.$$

Thus, although we do not know the leading coefficients of the b_i 's, we can recover all logarithmic derivatives of the polynomial we wish to reconstruct.

Once the $\mathcal{L}_\zeta$'s are known, we reconstruct $b(\Gamma)$ by solving the linear system

$$\partial_\zeta P - \mathcal{L}_\zeta \cdot P = 0 \quad \text{for all } \zeta \in \{z_1, \dots, z_k\}$$

in the space of polynomials in $\mathbf{z}$ of degree at most δ , where δ is a known upper bound on $\deg(b \circ \Gamma)$. After clearing denominators, this is a finite linear system in the coefficients of P . In characteristic zero, its solution space is one-dimensional and spanned by $b(\Gamma)$: any solution satisfies $\partial_\zeta(P/B) = 0$ for every ζ , and the common kernel of these partial derivations on $\mathbb{F}(\mathbf{z})$ is exactly $\mathbb{F}$.

If the characteristic of the field is $p > 0$, then the common kernel is instead $\mathbb{F}(z_1^p, \dots, z_k^p)$, so this argument does not give uniqueness without a degree restriction. Uniqueness still holds when $p > \delta$, since both P and B have degree at most δ . This is why this idea works in characteristic zero or sufficiently high characteristic.

Arbitrary-characteristic approach: ratios. The characteristic zero proof removes the unknown normalization by taking a logarithmic derivative. An alternative argument that works in arbitrary characteristic instead evaluates the same local denominator at two values of u and takes their ratio. Here, the leading coefficient cancels since we take the ratio.

Let $\mathbf{y}$ be an independent tuple of seed variables. Define the hybrid generators

$$\begin{aligned} H_i(\mathbf{y}, \mathbf{z}) &= (\Gamma_1(\mathbf{y}), \dots, \Gamma_i(\mathbf{y}), \Gamma_{i+1}(\mathbf{z}), \dots, \Gamma_n(\mathbf{z})), & 0 \leq i \leq n, \\ R_i(\mathbf{y}, \mathbf{z}, u) &= (\Gamma_1(\mathbf{y}), \dots, \Gamma_{i-1}(\mathbf{y}), u, \Gamma_{i+1}(\mathbf{z}), \dots, \Gamma_n(\mathbf{z})), & 1 \leq i \leq n. \end{aligned}$$

Thus $H_0 = \Gamma(\mathbf{z})$, $H_n = \Gamma(\mathbf{y})$, and the two substitutions $u = \Gamma_i(\mathbf{z})$ and $u = \Gamma_i(\mathbf{y})$ in R_i give H_{i-1} and H_i , respectively.

For our choice of Γ , the hybrid substitutions also preserve the required joint coprimality, and $b(H_i) \neq 0$ for every i , as we prove below. We can therefore apply the basic idea to reconstruct

$$\tilde{B}_i(\mathbf{y}, \mathbf{z}, u) := \frac{b \circ R_i}{\text{lc}_u(b \circ R_i)}.$$

We note that

$$\frac{\tilde{B}_i(\mathbf{y}, \mathbf{z}, \Gamma_i(\mathbf{z}))}{\tilde{B}_i(\mathbf{y}, \mathbf{z}, \Gamma_i(\mathbf{y}))} = \frac{b(H_{i-1})}{b(H_i)}.$$

Multiplying these ratios gives the telescoping product

$$\prod_{i=1}^n \frac{\tilde{B}_i(\mathbf{y}, \mathbf{z}, \Gamma_i(\mathbf{z}))}{\tilde{B}_i(\mathbf{y}, \mathbf{z}, \Gamma_i(\mathbf{y}))} = \prod_{i=1}^n \frac{b(H_{i-1})}{b(H_i)} = \frac{b(H_0)}{b(H_n)} = \frac{b(\Gamma(\mathbf{z}))}{b(\Gamma(\mathbf{y}))}.$$

We now have an explicit rational function whose numerator and denominator involve disjoint sets of variables and are therefore coprime. We can learn this rational function using the usual linear algebra approach and get $b(\Gamma(\mathbf{z}))$ up to a scalar in $\mathbb{F}$, from which we can recover b .

Factorization given rational interpolation: With the bounded-individual-degree sparse rational interpolation result at hand, the factorization follows the template of the meta-algorithm of [CS26b], which we sketch next.

For this application, we use the arbitrary-characteristic proof. We therefore need a generator whose specializations include all the hybrid maps R_i used in that proof.

Given the input product $f = \prod_{j=1}^\ell h_j$ of irreducible polynomials in $\mathcal{P}(n+1, s, d)$, we first remove its maximal monomial factor. We then pick a distinguished variable x_0 and view f as a polynomial in x_0 with coefficients in $\mathbb{F}[\mathbf{x}]$, $f = \sum_j f_j \cdot x_0^j$. In particular, $f_0 \neq 0$. We repeat the procedure over all choices of the distinguished variable so that every nonconstant factor is recovered.

We next define $\tilde{f}(X, \mathbf{x}) := f(X \cdot f_0, \mathbf{x})/f_0$, which is reverse-monic in X and has coefficients in $\mathbb{F}[\mathbf{x}]$: its constant coefficient is 1, and its j th coefficient is $f_j f_0^{j-1}$ for $j \geq 1$. By [CS26b, Claim 2.5] (see Claim 2.11), each irreducible factor $h = \sum_{j=0}^{d_h} c_j \cdot x_0^j$ of f with $x_0 \in \text{var}(h)$ corresponds to an irreducible factor

$$\tilde{h}(X, \mathbf{x}) = \sum_{j=0}^{d_h} \frac{c_j \cdot f_0^j}{c_0} \cdot X^j$$

of $\tilde{f}$. These coefficients are also polynomials, since $c_0 \mid f_0$.

Let $A_1(v), \dots, A_n(v)$ be Lagrange interpolation polynomials at distinct points $\eta_1, \dots, \eta_n$ in the field of computation, and define

$$\Psi(\mathbf{y}, \mathbf{z}, u, v) := \sum_{i=1}^n A_i(v) R_i(\mathbf{y}, \mathbf{z}, u).$$

Thus $\Psi|_{v=\eta_i} = R_i$, and a further specialization gives Γ . We compose $\tilde{f}$ with Ψ , obtaining

$$\hat{f}(X, \mathbf{y}, \mathbf{z}, u, v) := \tilde{f}(X, \Psi(\mathbf{y}, \mathbf{z}, u, v)),$$

a polynomial in 11 variables, which we recover explicitly by dense interpolation and factor. Each irreducible factor $\tilde{h}$ of $\tilde{f}$ as above corresponds to a divisor of $\hat{f}$. After normalizing this divisor to have constant coefficient 1 as a polynomial in X , its coefficients provide us with blackbox access to the rational functions $(c_j \cdot f_0^j / c_0)(\Psi)$. Dividing the j th coefficient by $f_0(\Psi)^j$ and specializing Ψ yields the fractions $c_j(R_i)/c_0(R_i)$ and $c_j(\Gamma)/c_0(\Gamma)$.

Since h is irreducible and depends on x_0 , Gauss's lemma gives $\gcd(c_0, c_1, \dots, c_{d_h}) = 1$, and since $h \in \mathcal{P}(n+1, s, d)$, all c_j 's are in $\mathcal{P}(n, s, d)$. We can therefore invoke the arbitrary-characteristic proof of Theorem 1.6, with $b = c_0$ and $a_j = c_j$, to recover the c_j 's, and hence h , up to a common scalar. Since $\hat{f}(0, \mathbf{y}, \mathbf{z}, u, v) = 1$, every nonconstant irreducible factor of $\hat{f}$ has positive X -degree. Enumerating divisors of $\hat{f}$ of X -degree at most d , normalized to have constant coefficient 1, therefore yields a polynomially bounded list of candidate factors for fixed d . If we work over an extension, we normalize the candidates and keep only those with coefficients in $\mathbb{F}$. We then use the coprimality certificate to improve the divisibility and multiplicity procedures of [CS26b], prune out any reducible or non-dividing polynomial from the list, and compute the multiplicity of each irreducible factor.

1.4 Related work

We elaborate on the work most directly relevant to the present paper.

Sparsity bounds for factors. Volkovich [Vol15, Vol17] showed that factors of multilinear, respectively multiquadratic, s -sparse polynomials are again s -sparse. Bhargava, Saraf, and Volkovich [BSV20] extended this to give the first nontrivial sparsity bound for factors of (n, s, d) -sparse polynomials: $s^{O(d^2 \log n)}$. Whether the true bound is polynomial in s and n is a major open problem. Bisht and Saxena [BS25] improved this to $s^{\text{poly}(d)}$ under the assumption that the polynomial is also symmetric. Bisht and Volkovich [BV25] obtained an $s^{O(d \log s)}$ bound on the sparsity of a quotient of an s -sparse polynomial of bounded individual degree d by a multilinear factor. From the work of Forbes [For15] and [BV25] it follows that the sparsity of the quotient of $f \in \mathcal{P}(n, s, d)$ by a degree r polynomial is bounded by $(ns)^{O(dr)}$.

Deterministic factorization. For a single (n, s, d) -sparse polynomial, [BSV20] gave a $\text{poly}(n, s^{d^7 \log n})$ -time deterministic factorization algorithm, later improved by [CS26b] to $\text{poly}(n, s^{d^3 \log n})$.

Very recently, Bhattacharjee, Kothary, Rai and Saraf [BKRS26] obtained a complementary list-factorization result for sparse polynomials of bounded individual degree. They show that, given an explicit n -variate s -sparse polynomial of individual degree at most d , one can deterministically output in time $\text{poly}(n, d!, s^d)$ a list of s^d constant-depth circuits containing every factor not divisible by a monomial. The list may contain spurious circuits. They also consider the more general setting in which the input sparse polynomial need not have bounded individual degree. For an s -sparse polynomial of individual degree at most D , they give a deterministic quasipolynomial-time algorithm that outputs a list of size $\binom{D}{\leq d}^{\log s}$ containing all factors of individual degree at most d that are not divisible by a monomial. The list may contain spurious elements.

For products of sparse polynomials, the situation has been more limited: Volkovich’s algorithm [Vol15] handles products with multilinear factors, while [Vol17] handles multiquadratic inputs. In [CS26b], the authors solve the problem for multiquadratic factors and give the quasipolynomial-time algorithm of Theorem 1.2 for general d . Theorem 1.5 of the present paper is the first polynomial-time algorithm, for constant $d > 2$, for factoring products of irreducible polynomials from $\mathcal{P}(n, s, d)$ over arbitrary fields, thereby resolving Question 1.1 for every constant d .

Rational interpolation. Deterministic interpolation of a rational function a/b from blackbox access, where a and b are sparse, is a long-standing open problem; the best deterministic algorithm of Grigoriev, Karpinski, and Singer [GKS94] runs in time exponential in the number of variables. Randomized polynomial-time algorithms are known [CL11, vdHL21].

[CS26b, Theorems 5.1, 5.2] solve the bounded-individual-degree case, where the algorithm also receives a precomputed list containing all irreducible factors of b . Theorem 1.6 settles a bounded-individual-degree special case: the numerators $a_1, \dots, a_N$ and common denominator b all lie in $\mathcal{P}(n, s, d)$ and satisfy $\gcd(a_1, \dots, a_N, b) = 1$. Compared with [CS26b, Theorems 5.1, 5.2], the present theorem removes the need for a precomputed list containing all irreducible factors of b .

1.5 AI disclosure

Originally, in the first version of this paper [CS26a], we obtained the proof for the characteristic zero case (the one given in Section 4.1). When giving the paper to ChatGPT 6 asking for comments

on the writing, among some comments it responded with the technique for arbitrary characteristic. The proof written here for arbitrary fields is based on this idea.

1.6 Organization

Section 2 sets up notation and basic facts. In Section 3 we give a new polynomial certificate for coprimality which will help us improve the running time of some of the algorithms of [CS26b] (we do that in Section 5). In Section 4 we give our rational reconstruction algorithm, and prove Theorem 1.6. The characteristic zero proof is given in Section 4.1. In Section 6 we deduce Theorem 1.5 from Theorem 1.6.

2 Preliminaries

We use bold letters to denote tuples. Specifically, we let $\mathbf{x} = (x_1, \dots, x_n)$ denote an n -tuple of indeterminates. Two nonzero polynomials are *associate* if one is a scalar multiple of the other.

We use the computational conventions of Remark 1.4 stated above. Whenever we speak of blackbox access to a fraction $R = f/g \in \mathbb{F}(\mathbf{x})$, with $g \neq 0$, the blackbox returns $f(\alpha)/g(\alpha)$ when $g(\alpha) \neq 0$, and returns $\perp$ when $g(\alpha) = 0$.

Definition 2.1. A polynomial $f \in \mathbb{F}[\mathbf{x}]$ is s -sparse if it is a sum of at most s monomials. It is of *bounded individual degree* d if $\deg_{x_i}(f) \leq d$ for every i . We denote the class of n -variate, s -sparse polynomials of bounded individual degree d by $\mathcal{P}(n, s, d)$, and the class of factors of polynomials in $\mathcal{P}(n, s, d)$ by $\mathcal{P}_{\text{Factors}}(n, s, d)$.

For a nonconstant $\phi \in \mathcal{P}_{\text{Factors}}(n, s, d)$ and a polynomial $f \in \mathbb{F}[\mathbf{x}]$, we denote by $v_\phi(f)$ the multiplicity of ϕ as a factor of f (i.e., the largest $k \geq 0$ such that $\phi^k \mid f$, with $v_\phi(f) = 0$ if $\phi \nmid f$). If $f = 0$ then we set $v_\phi(f) = \infty$. We write $\text{var}(f) \subseteq \{x_1, \dots, x_n\}$ for the set of variables on which f depends nontrivially.

2.1 Basic polynomial algorithms

The following well-known facts on univariate factorization and dense polynomial interpolation will be used throughout. For univariate factorization over $\mathbb{Q}$ we use the algorithm of Lenstra, Lenstra, and Lovász [LLL82]; over finite fields we use the algorithm of von zur Gathen and Shoup [vzGS92]. For multivariate factorization we use the reduction of [Kal85b, Kal85a] to the univariate case.

Theorem 2.2 (Univariate factorization over $\mathbb{Q}$). *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 $\text{poly}(d, B)$.*

We next state a theorem concerning deterministic multivariate factorization. See e.g., [Len85, Kal85b].

Theorem 2.3 (Multivariate polynomial factorization). *There exists a deterministic algorithm that given an n -variate polynomial $f \in \mathbb{F}[\mathbf{x}]$ of total degree d , and bit complexity B , outputs its factorization into irreducible polynomials. Its running time is $\text{poly}(d^n) \cdot T(d, B, \mathbb{F})$, where $T(d, B, \mathbb{F})$ is the running time of univariate factorization of degree d polynomials, with bit complexity B , over $\mathbb{F}$.*

Lemma 2.4 (Dense multivariate interpolation, folklore). *There is a deterministic algorithm that, given blackbox access to an n -variate polynomial $f \in \mathbb{F}[\mathbf{x}]$ of total degree at most d , returns its monomial representation in time $\text{poly}((d+1)^n)$.*

We shall also rely on rational interpolation, with the blackbox convention above.

Lemma 2.5 (Rational interpolation). *There is a deterministic algorithm that, given blackbox access to an n -variate rational function $R(\mathbf{x}) \in \mathbb{F}(\mathbf{x})$, with the promise that $R(\mathbf{x}) = f(\mathbf{x})/g(\mathbf{x})$ for some $f, g \in \mathbb{F}[\mathbf{x}]$ of total degree at most d , returns the unique (up to a common nonzero scalar) coprime pair (f^*, g^*) with $R = f^*/g^*$, in time $\text{poly}((d+1)^n)$.*

For completeness, we sketch the proof.

Proof. Let $S \subset \mathbb{F}$ be a set of size $3d+1$. Let $\phi, \psi \in \mathbb{F}[\mathbf{x}]$ be unknown polynomials of total degree at most d . We construct the linear system $R(\alpha)\phi(\alpha) = \psi(\alpha)$, where α ranges over all points in S^n for which $R(\alpha)$ is defined, and find a nonzero solution (ϕ, ψ) ; one exists since (g, f) is a solution. For every such α , we have $g(\alpha) \neq 0$ and

$$f(\alpha)\phi(\alpha) - g(\alpha)\psi(\alpha) = 0.$$

Thus, the polynomial $g(\mathbf{x})(f(\mathbf{x})\phi(\mathbf{x}) - g(\mathbf{x})\psi(\mathbf{x}))$ evaluates to zero on the entire grid S^n . The total degree of this polynomial is at most $3d$. By the polynomial identity lemma, since $|S| > 3d$, it follows that $g(\mathbf{x})(f(\mathbf{x})\phi(\mathbf{x}) - g(\mathbf{x})\psi(\mathbf{x})) = 0$. Because $g(\mathbf{x})$ is not the zero polynomial, $f(\mathbf{x})\phi(\mathbf{x}) - g(\mathbf{x})\psi(\mathbf{x}) = 0$. In particular $\phi \neq 0$, since otherwise $g\psi = 0$ and hence $\psi = 0$. Therefore $f/g = \psi/\phi$. Dividing ψ and ϕ by their greatest common divisor yields the unique coprime pair (f^*, g^*) with $f^*/g^* = R$, up to a common nonzero scalar.

To bound the running time, observe that S^n contains $(3d+1)^n$ points. The polynomials ϕ and ψ each have $\binom{n+d}{n} \leq (d+1)^n$ unknown coefficients. Consequently, the linear system consists of at most $(3d+1)^n$ equations in $2\binom{n+d}{n}$ variables. Thus, the dimensions of the system are polynomial in $(d+1)^n$. Evaluating the black box, solving the linear system, and computing the greatest common divisor of the resulting dense multivariate polynomials all take time polynomial in the size of the system. $\square$

In our application, all uses of [Theorem 2.3](#) and [Lemma 2.5](#) will be on polynomials and rational functions in a *constant* number of variables (the generator variables), so the running times will be polynomial in the relevant parameters.

We next recall the definition of Lagrange interpolation.

Definition 2.6. Let $\alpha_1, \dots, \alpha_r \in \mathbb{F}$ be different field elements. Then the polynomials

$$A_i(x) = \prod_{1 \leq j \leq r, j \neq i} \frac{x - \alpha_j}{\alpha_i - \alpha_j} \tag{1}$$

satisfy $\deg(A_i) = r-1$ and $A_i(\alpha_j) = \delta_{i,j}$, where $\delta_{i,j}$ is Kronecker's delta. The A_i -s are called the Lagrange interpolating polynomials of $\alpha_1, \dots, \alpha_r$.

We will also need the notion of a reduced form of a rational function.

Definition 2.7 (Reduced rational function). A rational function $R(u) \in \mathbb{F}(u)$ is said to be in *reduced form* if it is expressed as the fraction

$$R(u) = \frac{f(u)}{g(u)},$$

such that the following conditions hold:

- (1) $f(u), g(u) \in \mathbb{F}[u]$ with $g(u) \neq 0$.
- (2) $f(u)$ and $g(u)$ are coprime in $\mathbb{F}[u]$ (that is, $\gcd(f, g) = 1$).
- (3) $g(u)$ is a monic polynomial.

Under these conditions, the polynomials $f(u)$ and $g(u)$ are uniquely determined by $R(u)$. The polynomial $g(u)$ is referred to as the *reduced denominator* of $R(u)$.

2.2 Reverse-monic Polynomials and Normalization

We shall need the following notions and claims from [CS26b].

Definition 2.8 ([CS26b, Definition 2.2]). A polynomial $f \in \mathbb{F}[x_0, \mathbf{x}]$ is said to be reverse-monic with respect to x_0 if its free term as a polynomial in $\mathbb{F}(\mathbf{x})[x_0]$ is 1. Equivalently, f is reverse-monic with respect to x_0 if it is of the form $f = 1 + \sum_{i \geq 1} f_i x_0^i$ where $f_i \in \mathbb{F}[\mathbf{x}]$.

Claim 2.9 ([CS26b, Claim 2.3]). *Let $f \in \mathbb{F}[x_0, \mathbf{x}]$ be a reverse-monic polynomial with respect to x_0 . Then, the number of irreducible factors of f is at most $\deg_{x_0}(f)$.*

We next define a *normalization* operation that turns a polynomial into a reverse-monic one.

Definition 2.10 ([CS26b, Definition 2.4]). Let $f(x_0, \mathbf{x}) = \sum_{i=0}^d f_i(\mathbf{x}) x_0^i \in \mathbb{F}[x_0, \mathbf{x}]$, such that $x_0 \nmid f$. The normalization of f with respect to x_0 is the reverse-monic polynomial

$$\tilde{f}(y, \mathbf{x}) = f(y f_0, x_1, \dots, x_n) / f_0 = 1 + \sum_{i=1}^d f_i(\mathbf{x}) f_0^{i-1}(\mathbf{x}) y^i. \quad (2)$$

Claim 2.11 ([CS26b, Claim 2.5]). *Let f be as in Definition 2.10. Suppose $f = \prod_{i=0}^N h_i \in \mathbb{F}[x_0, \mathbf{x}]$ is the factorization of f into irreducible factors h_i , and denote each factor by $h_i = \sum_{j=0}^{d_i} c_{i,j} x_0^j$ (where $c_{i,j} \in \mathbb{F}[\mathbf{x}]$). Then,*

$$\tilde{f} = \prod_{i=0}^N \tilde{h}_i, \text{ where } \tilde{h}_i = \sum_{j=0}^{d_i} \frac{c_{i,j}}{c_{i,0}} f_0^j y^j.$$

The factors $\tilde{h}_i$ with $d_i > 0$ are irreducible, and give the factorization of $\tilde{f}$ into nonunit factors. Moreover, every factor (not necessarily irreducible) of f of the form $\sum_{j=0}^d \zeta_j x_0^j$ (where $\zeta_j \in \mathbb{F}[\mathbf{x}]$) corresponds to a factor of $\tilde{f}$ of the form $\sum_{j=0}^d \frac{\zeta_j}{\zeta_0} f_0^j y^j$. In particular, every multiquadratic factor of f of the form $a_2 x_0^2 + a_1 x_0 + b$ corresponds to a factor of $\tilde{f}$ of the form $1 + \frac{a_1 f_0}{b} y + \frac{a_2 f_0^2}{b} y^2$. In addition, we have blackbox access to $\tilde{f}$ from blackbox access to f .

2.3 The generator of [CS26b]

We follow the formulation in [CS26b, §3]. For a positive integer m (which will be taken to be a polynomial in the input parameters), let q be a prime in $[m, 2m]$, and fix sets $\mathcal{A} = \{\alpha_1, \dots, \alpha_m\}$, $\mathcal{B} = \{\beta_1, \dots, \beta_n\}$, $\mathcal{C} = \{\gamma_1, \dots, \gamma_n\} \subseteq \mathbb{F}$ of distinct field elements. Let $\{A_i(y)\}_{i=1}^m$ be the Lagrange interpolation polynomials for $\mathcal{A}$, and similarly $\{B_i(z)\}_{i=1}^n$ for $\mathcal{B}$ and $\{C_i(v)\}_{i=1}^n$ for $\mathcal{C}$.

Definition 2.12 (KS generator). The Klivans–Spielman generator (KS-generator) is the map $G_{(m)}^{\text{KS}} : \mathbb{F}^4 \rightarrow \mathbb{F}^n$ defined by

$$G_{(m)}^{\text{KS}}(t, y, z, w) = \sum_{i=1}^m A_i(y) \cdot ((1 + B_1(z)(w - 1)) t^{i \bmod q}, \dots, (1 + B_n(z)(w - 1)) t^{i^n \bmod q}).$$

Observation 2.13. *The degree of (each coordinate of) $G_{(m)}^{KS}$ can be upper bounded by*

$$\deg(G_{(m)}^{KS}) \leq (m-1) + (n-1) + 1 + (q-1) < 3m + n.$$

Theorem 2.14 ([KS01], see also [CS26b, §3]). *Let $f \in \mathcal{P}(n, s, d)$ and $m = (nds)^2$. Then f can be recovered, in deterministic time $\text{poly}(m)$, from the values of $f(G_{(m)}^{KS}(t, y, z, w))$ as a polynomial in the four variables (t, y, z, w) .*

The KS-generator does not allow one to “revive” a single original variable x_i in a controlled way. To remedy this, [CS26b], following [SV14], adds a second component that uses two further variables (u, v) :

$$G^{SV}(u, v) = (C_1(v)u, \dots, C_n(v)u).$$

Definition 2.15 ([CS26b, Definition 3.3]). The generator $G_{(m)} : \mathbb{F}^6 \rightarrow \mathbb{F}^n$ is

$$G_{(m)}(t, y, z, w, u, v) = G_{(m)}^{KS}(t, y, z, w) + G^{SV}(u, v).$$

We shall refer to (t, y, z, w, u, v) as the *generator variables*, and write $\xi = (t, y, z, w, u, v) \in \mathbb{F}^6$ for short. Thus $G_{(m)} = G_{(m)}(\xi)$.

For each $1 \leq i \leq n$, the *i*th revived generator $G_{(m,i)}$ is obtained from $G_{(m)}$ by the substitution $u \leftarrow u - G_{(m)}^{KS}(t, y, z, w)_i$ and $v \leftarrow \gamma_i$, where $G_{(m)}^{KS}(\cdot)_i$ denotes the *i*th coordinate of $G_{(m)}^{KS}$. Since $C_j(\gamma_i) = \delta_{j,i}$, we get

$$\begin{aligned} G_{(m,i)}(\xi) = & (G_{(m)}^{KS}(t, y, z, w)_1, \dots, G_{(m)}^{KS}(t, y, z, w)_{i-1}, u, \\ & G_{(m)}^{KS}(t, y, z, w)_{i+1}, \dots, G_{(m)}^{KS}(t, y, z, w)_n). \end{aligned} \quad (3)$$

In particular, the *i*-th revived generator $G_{(m,i)}$ depends on the five generator variables (t, y, z, w, u) , and the variable u plays the role of the “revived” original variable x_i .

We will repeatedly use the following basic properties of $G_{(m)}$, all established in [CS26b].

Lemma 2.16 ([CS26b, Lemma 3.4, Corollaries 3.5, 3.6]). *Let $f \in \mathcal{P}_{\text{Factors}}(n, s, d)$ and let $m = (nds)^2$. Then:*

- (1) $f(G_{(m)}) \neq 0$. In fact, if $f \in \mathcal{P}(n, s, d)$, then f can be recovered from $f(G_{(m)})$ in deterministic time $\text{poly}(m)$.
- (2) For every $1 \leq i \leq n$, $\deg_u(f(G_{(m,i)})) = \deg_{x_i}(f)$.
- (3) In particular, if $x_i \notin \text{var}(f)$ then $f(G_{(m,i)})$ does not depend on u , while if $x_i \in \text{var}(f)$ then $\deg_u(f(G_{(m,i)})) \geq 1$.

3 Coprimality certificate

In this section we give a new certificate for coprimality of sparse polynomials. This replaces the certificate of [BV25, CS26b] that required hitting high powers of the underlying polynomials.

Lemma 3.1 (Coprimality certificate). *Let $\mathbb{E}$ be any field, and let $f_1, \dots, f_N \in \mathbb{E}[\mathbf{x}, u]$ each have at most s monomials and individual degree at most d . Suppose their joint gcd in $\mathbb{E}(\mathbf{x})[u]$ is 1, and let $\delta = \max_j \deg_u f_j \geq 1$. There is a nonzero $\Phi(\mathbf{x})$ with at most $s^{2\delta}$ monomials and individual degree at most $2\delta d$ such that every substitution of the $\mathbf{x}$ variables into a field which keeps Φ nonzero preserves the trivial joint gcd in u .*

Proof. Over $\mathbb{K} = \mathbb{E}(\mathbf{x})$, form the coefficient matrix M that has 2δ rows, indexed by the coefficients of $1, u, \dots, u^{2\delta-1}$, and $N\delta$ columns equal to the coefficient vectors of

$$u^k f_j, \quad 1 \leq j \leq N, \quad 0 \leq k < \delta.$$

Claim 3.2. M has rank 2δ .

Proof. Let $V \subseteq \mathbb{K}[u]_{<2\delta}$ be the column span. A common divisor of positive degree divides every member of V , so in that case $1 \notin V$.

Conversely, suppose $\gcd(f_1, \dots, f_N) = 1$. Assume without loss of generality that $\deg(f_N) = \delta$.

Let $g \in \mathbb{K}[u]$ be arbitrary of degree smaller than 2δ . Since $\gcd(f_1, \dots, f_N) = 1$ there are polynomials h_i such that $g = \sum_{i=1}^N f_i h_i$. Let q_j, r_j be such that $h_j = q_j f_N + r_j$, with $\deg(r_j) < \delta$. Then,

$$g = \sum_{j=1}^N h_j f_j = \sum_{j=1}^N r_j f_j + \left(\sum_{j=1}^N q_j f_j \right) f_N.$$

Since $\deg(g), \deg(r_j f_j) < 2\delta$ it follows that $\deg\left(\left(\sum_{j=1}^N q_j f_j\right) f_N\right) < 2\delta$ and hence $\deg\left(\sum_{j=1}^N q_j f_j\right) < \delta$. Consequently, the coefficient vector of g is in the column span of M . In particular, the columns of M span all polynomials of degree smaller than 2δ and the claim follows. $\square$

Let Φ be a nonzero $2\delta \times 2\delta$ minor. Consider a column in this minor corresponding to $u^k f_j$. It can be expressed as a sum of at most s vectors of the form

$$c\mathbf{x}^\alpha \mathbf{e}_{e+k+1},$$

where $\mathbf{e}_{e+k+1}$ is the elementary basis vector with 1 in coordinate $e+k+1$ and zeros elsewhere, corresponding to each original term $c\mathbf{x}^\alpha u^e$ of f_j . By Claim 3.3 below, with $r = 2\delta$, the minor Φ has the required sparsity and individual degree. Every substitution keeping Φ nonzero preserves the full row rank of M . Hence 1 remains in its column span, and no nonconstant polynomial in u can divide all the substituted f_j 's. $\square$

Claim 3.3 (Sparse minors and rank preservation). *Let $\mathbb{F}$ be an arbitrary field, and let $M \in \mathbb{F}[\mathbf{x}]^{a \times b}$. Suppose that each column of M can be written as*

$$\sum_{\alpha \in A_j} \mathbf{v}_{j,\alpha} \mathbf{x}^\alpha, \quad |A_j| \leq s, \quad \mathbf{v}_{j,\alpha} \in \mathbb{F}^a,$$

where every monomial $\mathbf{x}^\alpha$ has individual degree at most d . Then every $k \times k$ minor of M has at most s^k monomials and individual degree at most kd .

Moreover, if M has rank $r \geq 1$ over $\mathbb{F}(\mathbf{x})$, then there is a nonzero polynomial Φ with at most s^r monomials and individual degree at most rd such that every $\mathbb{F}$ -algebra homomorphism $\sigma : \mathbb{F}[\mathbf{x}] \rightarrow K$ into a field satisfying $\sigma(\Phi) \neq 0$ preserves the rank of M .

Proof. By multilinearity in the columns, a $k \times k$ minor is a sum of at most s^k terms. Each term is the product of k monomials and the determinant of a matrix over $\mathbb{F}$. This gives the claimed sparsity and degree bounds.

For the second assertion, choose a nonzero $r \times r$ minor Φ of M . If $\sigma(\Phi) \neq 0$, then $\text{rank}_K \sigma(M) \geq r$. On the other hand, all larger minors of M are identically zero, and remain zero after applying σ . Thus $\text{rank}_K \sigma(M) = r$. $\square$

Set

$$S := s^{2d}, \quad D := 2d^2, \quad m := (nDS)^2, \quad \Gamma(\mathbf{z}) := G_{(m)}^{\text{KS}}(z_1, z_2, z_3, z_4), \quad (4)$$

where we rename the four variables (t, y, z, w) of [Definition 2.12](#) to $\mathbf{z} = (z_1, z_2, z_3, z_4)$. These parameters are fixed for the rest of the paper. By [Theorem 2.14](#), every polynomial in $\mathcal{P}(n, S, D)$ can be recovered from its composition with Γ . For $i \in [n]$, define

$$\widehat{R}_i(\mathbf{z}, u) := (\Gamma_1(\mathbf{z}), \dots, \Gamma_{i-1}(\mathbf{z}), u, \Gamma_{i+1}(\mathbf{z}), \dots, \Gamma_n(\mathbf{z})). \quad (5)$$

Corollary 3.4 (Preservation of gcd degree). *Let $p, q \in \mathcal{P}(n, s, d)$ be nonzero, and fix $i \in [n]$. Let $K_i := \mathbb{F}(\mathbf{x}_{\neq i})$, and set $P := p(\widehat{R}_i)$ and $Q := q(\widehat{R}_i)$. Then*

$$\deg_u \gcd(P, Q) = \deg_{x_i} \gcd(p, q).$$

$\mathbb{F}(\mathbf{z})[u] \qquad K_i[x_i]$

In particular, the substitution preserves coprimality over the respective coefficient fields.

Proof. Note that $\deg_u P = \deg_{x_i} p =: a$ and $\deg_u Q = \deg_{x_i} q =: b$ since the leading coefficients of p and q with respect to x_i are nonzero sparse polynomials independent of x_i and the KS-generator keeps both nonzero.

Let M be the Sylvester matrix of p and q , viewed as univariates in x_i . Its columns are the coefficient vectors of

$$p, x_i p, \dots, x_i^{b-1} p, \quad q, x_i q, \dots, x_i^{a-1} q.$$

Each column satisfies the hypothesis of [Claim 3.3](#) with parameters s, d . Since $a + b \leq 2d$, every minor of M has at most S monomials and individual degree at most D . The KS-generator therefore preserves a nonzero minor of maximum order, and hence preserves the rank of M .

The degree assertions imply that the substituted matrix is the Sylvester matrix of P and Q . By [\[CS26b, Lemma 2.20\(4\)\]](#),

$$\deg \gcd(p, q) = a + b - \text{rank } M.$$

Applying this identity before and after substitution proves the claim. $\square$

4 Bounded-individual-degree sparse rational interpolation

In this section we prove [Theorem 1.6](#). We follow the high level idea described in [Section 1.3](#).

Throughout, let $b, a_1, \dots, a_N \in \mathcal{P}(n, s, d)$, with $b \neq 0$ and $\gcd(b, a_1, \dots, a_N) = 1$, and suppose that we have blackbox access to the fractions a_j/b . Let $\mathbb{F}$ be the original coefficient field. Following the conventions of [Remark 1.4](#), we work over a computation field $\mathbb{E}/\mathbb{F}$ with sufficiently many available evaluation elements if needed.

We first give a generator that when composed with $a_1, \dots, a_N, b$ preserves their coprimality, using the certificate given in [Lemma 3.1](#).

Let S, D, m and $\Gamma = \Gamma(\mathbf{z})$ be as in (4), and let $\mathbf{y}$ be an independent copy of the four seed variables $\mathbf{z}$. Define

$$H_i(\mathbf{y}, \mathbf{z}) = (\Gamma_1(\mathbf{y}), \dots, \Gamma_i(\mathbf{y}), \Gamma_{i+1}(\mathbf{z}), \dots, \Gamma_n(\mathbf{z})), \quad 0 \leq i \leq n, \quad (6)$$

$$R_i(\mathbf{y}, \mathbf{z}, u) = (\Gamma_1(\mathbf{y}), \dots, \Gamma_{i-1}(\mathbf{y}), u, \Gamma_{i+1}(\mathbf{z}), \dots, \Gamma_n(\mathbf{z})), \quad 1 \leq i \leq n. \quad (7)$$

Thus $H_0 = \Gamma(\mathbf{z})$ and $H_n = \Gamma(\mathbf{y})$, and

$$R_i|_{u=\Gamma_i(\mathbf{z})} = H_{i-1}, \quad R_i|_{u=\Gamma_i(\mathbf{y})} = H_i. \quad (8)$$

Using the coprimality certificate Φ of [Lemma 3.1](#) we next show that composing with the hybrid generators R_i maintains coprimality in our setting.

Lemma 4.1. Suppose $b, a_1, \dots, a_N \in \mathcal{P}(n, s, d)$, $b \neq 0$, and $\gcd(b, a_1, \dots, a_N) = 1$. Then $b(H_i) \neq 0$ for every i , and, for each $1 \leq i \leq n$,

$$\gcd(b(R_i), a_1(R_i), \dots, a_N(R_i)) = 1 \quad \text{in } \mathbb{F}(\mathbf{y}, \mathbf{z})[u].$$

Proof. By Theorem 2.14, $B(\mathbf{z}) = b(\Gamma(\mathbf{z})) \neq 0$. Setting $\mathbf{y} = \mathbf{z}$ in any $b(H_i)$ gives $B(\mathbf{z})$, so all $b(H_i)$ are nonzero. Since $b(R_i)|_{u=\Gamma_i(\mathbf{z})} = b(H_{i-1})$, also $b(R_i) \neq 0$.

Fix i . By Gauss's lemma, the original tuple has joint gcd 1 in $\mathbb{F}(\mathbf{x}_{\neq i})[x_i]$. If every polynomial has x_i -degree zero, then $b(R_i)$ is a nonzero element of the coefficient field, and there is nothing to prove. Otherwise, Lemma 3.1 supplies a nonzero $\Phi_i(\mathbf{x}_{\neq i}) \in \mathcal{P}(n, S, D)$.

Consider $\Phi_i \circ R_i$. On the diagonal $\mathbf{y} = \mathbf{z}$, the resulting polynomial becomes $\Phi_i(\Gamma(\mathbf{z}))$, which is nonzero by Theorem 2.14 (recall that Φ_i does not depend on the i th coordinate). The hybrid substitution therefore keeps the minor nonzero. The claim follows from Lemma 3.1. $\square$

Lemma 4.1 reduces the number of variables to a constant while keeping coprimality. Our next lemma shows how to compute $b(R_i)(\mathbf{y}, \mathbf{z}, u)$ up to a polynomial factor independent of u , given blackbox access to the $(a_j/b)(R_i)$. We state it in a general form and then apply it in our setting.

Lemma 4.2 (Local denominator reconstruction). Let $\mathbf{t} = (t_1, \dots, t_k)$, where k is constant, and let $A_1, \dots, A_N, Q \in \mathbb{F}[\mathbf{t}, u]$ have total degree at most δ , with $Q \neq 0$ and

$$\gcd(A_1, \dots, A_N, Q) = 1 \quad \text{in } \mathbb{F}(\mathbf{t})[u].$$

Given blackbox access to $A_1/Q, \dots, A_N/Q$, we can compute a nonzero polynomial $C \in \mathbb{F}[\mathbf{t}, u]$ of total degree at most δ such that

$$Q(\mathbf{t}, u) = h(\mathbf{t})C(\mathbf{t}, u) \quad \text{for some nonzero } h \in \mathbb{F}[\mathbf{t}]. \quad (9)$$

The running time is $\text{poly}(N, \delta)$ field operations and blackbox queries, with the exponent depending on k .

Proof. Apply Lemma 2.5 to each fraction to obtain a reduced pair $P_j, Q_j \in \mathbb{F}[\mathbf{t}, u]$ with $P_j/Q_j = A_j/Q$. Each polynomial has degree at most δ . Since the pair is reduced, $Q_j \mid Q$. Compute

$$C = \text{lcm}_{j \in [N]} Q_j \quad \text{in } \mathbb{F}[\mathbf{t}, u].$$

Every intermediate lcm divides Q and therefore has degree at most δ .

Over $\mathbb{F}(\mathbf{t})$, each irreducible factor of Q is absent from at least one A_j . Its full multiplicity therefore occurs in at least one reduced denominator. Consequently, C and Q are associates in $\mathbb{F}(\mathbf{t})[u]$. Since $C \mid Q$ in the polynomial ring, the quotient Q/C is a polynomial independent of u , giving (9). The interpolation and gcd computations all involve the constant number $k + 1$ of variables. $\square$

Corollary 4.3. Let m be as in (4). Set $\delta = (3m + n)nd$. Given blackbox access to $a_1/b, \dots, a_N/b$, we can learn polynomials $C_i(\mathbf{y}, \mathbf{z}, u)$ such that

$$b(R_i(\mathbf{y}, \mathbf{z}, u)) = h_i(\mathbf{y}, \mathbf{z})C_i(\mathbf{y}, \mathbf{z}, u). \quad (10)$$

The running time is $\text{poly}(N, \delta)$ field operations and blackbox queries.

Proof. Apply Lemma 4.2 to $A_j = a_j(R_i)$ and $Q = b(R_i)$. Note that since the individual degrees are bounded by d , Observation 2.13 guarantees that the degree of each $a_j(R_i)$ and of $b(R_i)$ is smaller than δ . $\square$

We next show how to reconstruct an associate polynomial to $B(\mathbf{z})$. We give the proof that works over arbitrary fields here, and in [Section 4.1](#) we give a different argument tailored for characteristic zero.

Lemma 4.4. *Given blackbox access to the a_j/b we can recover an associate polynomial to $B(\mathbf{z}) = b(\Gamma)$ in time $\text{poly}(n, N, m)$.*

Proof. Apply [Corollary 4.3](#) and form the explicit polynomials

$$U(\mathbf{y}, \mathbf{z}) = \prod_{i=1}^n C_i(\mathbf{y}, \mathbf{z}, \Gamma_i(\mathbf{z})), \quad V(\mathbf{y}, \mathbf{z}) = \prod_{i=1}^n C_i(\mathbf{y}, \mathbf{z}, \Gamma_i(\mathbf{y})).$$

By (8) and (10), each factor of U or V , multiplied by h_i , equals $b(H_{i-1})$ or $b(H_i)$. These polynomials are nonzero by [Lemma 4.1](#). Moreover,

$$\frac{U(\mathbf{y}, \mathbf{z})}{V(\mathbf{y}, \mathbf{z})} = \prod_{i=1}^n \frac{b(H_{i-1})}{b(H_i)} = \frac{B(\mathbf{z})}{B(\mathbf{y})}. \quad (11)$$

Compute $g = \gcd(U, V)$. Since $B(\mathbf{z})$ and $B(\mathbf{y})$ involve disjoint variables, they are coprime, and hence

$$\tilde{B}(\mathbf{z}) := U/g = \lambda B(\mathbf{z}) \quad \text{for some } \lambda \in \mathbb{F}^\times.$$

For the running time we note that the computations involve nine variables and have total cost $\text{poly}(n, N, \delta)$. Since C_i divides $b(R_i)$, we have $\deg_u C_i \leq d$. Furthermore,

$$\deg U, \deg V \leq n(\delta + d(3m + n)).$$

Their gcd is computed in eight variables. The total number of field operations and blackbox queries is

$$\text{poly}(n, N, m) = \text{poly}(n, N, d, s^d).$$

□

Finally, we show how to reconstruct b and the a_j given the associate polynomial to $B(\mathbf{z})$.

Lemma 4.5 (Completing the reconstruction). *Given an explicit polynomial $\tilde{B} = \lambda B$, for some $\lambda \in \mathbb{F}^\times$, we can recover $\lambda b, \lambda a_1, \dots, \lambda a_N$ in $\text{poly}(n, N, d, m)$ field operations and blackbox queries.*

Proof. Apply [Theorem 2.14](#) to $\tilde{B}$ to recover λb . For each j , use [Lemma 2.5](#) to recover a reduced pair $g_j, q_j \in \mathbb{F}[\mathbf{z}]$ satisfying

$$\frac{g_j(\mathbf{z})}{q_j(\mathbf{z})} = \frac{a_j(\Gamma(\mathbf{z}))}{B(\mathbf{z})}.$$

These are rational functions in four variables whose numerator and denominator have degree at most $nd(3m + n)$, by [Observation 2.13](#). Since $q_j \mid B$, the exact polynomial quotient

$$\frac{\tilde{B} g_j}{q_j} = \lambda a_j(\Gamma(\mathbf{z}))$$

is well-defined. Apply [Theorem 2.14](#) once more to recover λa_j . All the numerators have the same scalar λ . □

Proof of Theorem 1.6. Apply Lemma 4.4 to recover $\tilde{B} = \lambda B$ for some $\lambda \in \mathbb{F}^\times$, and then apply Lemma 4.5 to recover $\lambda b, \lambda a_1, \dots, \lambda a_N$. The total cost is $\text{poly}(n, N, d, s^d)$ field operations and blackbox queries. Since $s > 1$, this is $\text{poly}(n, N, s^d)$. $\square$

Remark 4.6. The arbitrary-characteristic proof only requires blackbox access to the fractions on the images of $R_1, \dots, R_n$ and Γ . Indeed, Lemma 4.4 uses the fractions only after composition with the maps R_i , and Lemma 4.5 uses them only after composition with Γ . Thus access to these substituted fractions suffices for the reconstruction, with the same convention concerning undefined evaluations.

4.1 Bounded-individual-degree sparse rational interpolation in characteristic zero

In this section we give a different proof of Lemma 4.4 for fields of characteristic zero. We use the same m , Γ , and $B(\mathbf{z}) = b(\Gamma(\mathbf{z}))$ as above, and set $\delta = (3m + n)nd$. We first show how to compute the logarithmic derivatives of b composed with Γ . We then use them to set up a linear system whose solution space is spanned by B . Finally, Lemma 4.5 gives the required reconstruction of b and the a_j 's. The argument also works in characteristic larger than δ , as we explain at the end of the section.

4.1.1 Computing the logarithmic derivatives

We use the substitutions $\hat{R}_i$ of (5), and note that $\hat{R}_i(\mathbf{z}, u) = R_i(\mathbf{z}, \mathbf{z}, u)$. We apply Lemma 4.2 directly to these substitutions.

Lemma 4.7. *Given blackbox access to $a_1/b, \dots, a_N/b$, for each $1 \leq i \leq n$ we can compute the rational function*

$$L_i(\mathbf{z}) := \frac{(\partial_{x_i} b)(\Gamma(\mathbf{z}))}{B(\mathbf{z})} \in \mathbb{F}(\mathbf{z})$$

in time $\text{poly}(N, \delta)$.

Proof. Fix i . Substituting $u = \Gamma_i(\mathbf{z})$ into $b(\hat{R}_i(\mathbf{z}, u))$ gives $B(\mathbf{z}) \neq 0$, so $b(\hat{R}_i)$ is nonzero. The certificate used in the proof of Lemma 4.1 also shows that

$$\gcd(b(\hat{R}_i), a_1(\hat{R}_i), \dots, a_N(\hat{R}_i)) = 1 \quad \text{in } \mathbb{F}(\mathbf{z})[u].$$

Indeed, if the original polynomials are all independent of x_i , then $b(\hat{R}_i)$ is a nonzero element of the coefficient field. Otherwise, the certificate Φ_i from that proof remains nonzero after substituting the coordinates of Γ for the variables other than x_i . The claim follows from Lemma 3.1.

All the substituted polynomials have total degree at most δ . By Lemma 4.2, we can therefore compute $\hat{C}_i \in \mathbb{F}[\mathbf{z}, u]$ such that

$$b(\hat{R}_i(\mathbf{z}, u)) = \hat{h}_i(\mathbf{z}) \hat{C}_i(\mathbf{z}, u), \tag{12}$$

where $\hat{h}_i \in \mathbb{F}[\mathbf{z}]$ is nonzero. Since $\hat{h}_i$ does not depend on u , we have

$$\frac{\partial_u \hat{C}_i}{\hat{C}_i} = \frac{\partial_u b(\hat{R}_i)}{b(\hat{R}_i)} = \frac{(\partial_{x_i} b)(\hat{R}_i)}{b(\hat{R}_i)}.$$

The last equality follows because only the i th coordinate of $\hat{R}_i$ depends on u , and this coordinate is u itself. Now substitute $u = \Gamma_i(\mathbf{z})$ to obtain

$$L_i(\mathbf{z}) = \left. \frac{\partial_u \hat{C}_i(\mathbf{z}, u)}{\hat{C}_i(\mathbf{z}, u)} \right|_{u=\Gamma_i(\mathbf{z})}. \tag{13}$$

This substitution is well-defined: by (12), $\widehat{h}_i(\mathbf{z})\widehat{C}_i(\mathbf{z}, \Gamma_i(\mathbf{z})) = B(\mathbf{z}) \neq 0$.

The local reconstruction involves five variables and costs $\text{poly}(N, \delta)$. Also, $\deg \widehat{C}_i \leq \delta$ and $\deg_u \widehat{C}_i \leq d$, since $\widehat{C}_i$ divides $b(\widehat{R}_i)$. The numerator and denominator in (13) therefore have degree at most $\delta + d(3m + n) \leq 2\delta$, and the remaining computations take $\text{poly}(\delta)$ time. $\square$

4.1.2 Reconstruction from the logarithmic derivatives

We next show how to reconstruct B from the rational functions $L_1, \dots, L_n$ by solving a homogeneous linear system.

Lemma 4.8. *Suppose $\mathbb{F}$ has characteristic zero. Given the rational functions L_i computed in Lemma 4.7, we can recover an associate polynomial to $B(\mathbf{z})$ by solving a homogeneous linear system over $\mathbb{F}$ of size $\text{poly}(n, m, \delta)$.*

Proof. For every $\zeta \in \{z_1, z_2, z_3, z_4\}$, the chain rule gives

$$\mathcal{L}_\zeta(\mathbf{z}) := \frac{\partial_\zeta B(\mathbf{z})}{B(\mathbf{z})} = \sum_{i=1}^n L_i(\mathbf{z}) \partial_\zeta \Gamma_i(\mathbf{z}). \quad (14)$$

The right-hand side is explicitly known, so we can compute each $\mathcal{L}_\zeta$. We set up the following homogeneous linear system in the coefficients of an unknown polynomial $P \in \mathbb{F}[\mathbf{z}]$ of total degree at most δ :

$$\partial_\zeta P - \mathcal{L}_\zeta P = 0, \quad \zeta \in \{z_1, z_2, z_3, z_4\}. \quad (15)$$

As in the proof of Lemma 2.5, after clearing denominators each identity gives linear constraints on the coefficients of P . There are $\binom{\delta+4}{4}$ unknowns. By the degree bounds in Lemma 4.7, the numerators and denominators of the $\mathcal{L}_\zeta$'s have degree $\text{poly}(n, m, \delta)$, so the number of equations is also $\text{poly}(n, m, \delta)$.

It is clear that $P = B$ is a nonzero solution. Conversely, if P is a nonzero solution, then for every ζ ,

$$\partial_\zeta(P/B) = \frac{B\partial_\zeta P - P\partial_\zeta B}{B^2} = 0.$$

Since $\mathbb{F}$ has characteristic zero, the common kernel of the four partial derivations on $\mathbb{F}(\mathbf{z})$ is exactly $\mathbb{F}$. Thus $P/B \in \mathbb{F}^\times$, and the solution space is one-dimensional. Solving the system gives $\tilde{B} = \lambda B$ for some $\lambda \in \mathbb{F}^\times$. $\square$

4.1.3 Putting it together

We now combine the preceding lemmas with Lemma 4.5 to obtain the characteristic-zero proof of Theorem 1.6.

Proof of Theorem 1.6 in characteristic zero. Use the same m and Γ as in (4), and set $\delta = (3m + n)nd$. Apply Lemma 4.7 for each i to compute $L_1, \dots, L_n$. Next, apply Lemma 4.8 to recover $\tilde{B} = \lambda B$. Finally, Lemma 4.5 recovers λb and all the λa_j 's with the same scalar λ . The total running time is $\text{poly}(n, N, m) = \text{poly}(n, N, d, s^d)$, which gives the bound in Theorem 1.6. $\square$

Remark 4.9. The characteristic-zero proof only requires blackbox access to the fractions on the image of $G_{(m)}$. Indeed, the maps $\widehat{R}_i$ are precisely the revived generators from (3), and Γ is obtained by setting the SV variable u to zero in $G_{(m)}$. These are the only substitutions used in Lemma 4.7 and Lemma 4.5.

Remark 4.10 (The requirement in positive characteristic). The only use of characteristic zero is in the uniqueness argument in [Lemma 4.8](#). In characteristic $p > 0$, the common kernel of the partial derivations on $\mathbb{F}(\mathbf{z})$ is $\mathbb{F}(z_1^p, z_2^p, z_3^p, z_4^p)$, so that argument does not give uniqueness in general.

Nevertheless, uniqueness still holds when $p > \delta$. To see this, let P be a nonzero solution of (15) and write $P/B = A/C$ in reduced form. Then $\deg A, \deg C \leq \delta$, and for every ζ we have

$$C\partial_\zeta A = A\partial_\zeta C.$$

Since A and C are coprime, $A \mid \partial_\zeta A$ and $C \mid \partial_\zeta C$. A nonzero derivative has smaller degree, so both derivatives must be zero. In characteristic p , a polynomial whose partial derivatives all vanish has every variable exponent divisible by p . Since $\deg A, \deg C \leq \delta < p$, both A and C are constants. Thus $P/B \in \mathbb{F}^\times$, as required. This proves that the same algorithm works whenever $\text{char}(\mathbb{F}) > \delta$.

5 Algorithmic tools from [\[CS26b\]](#), and improvements

We first recall the algorithmic tools from [\[CS26b\]](#) that hold over arbitrary fields. The numbering of the cited results refers to the full version. We use the computational conventions of [Remark 1.4](#) stated above.

The first result computes the monomial factors of a product of factors of sparse polynomials.

Corollary 5.1 (Finding monomial divisors [\[CS26b, Corollary 3.8\]](#)). *There is a deterministic $\text{poly}(n, d, s, \ell)$ -time algorithm that, given blackbox access to a product f of ℓ polynomials in $\mathcal{P}_{\text{Factors}}(n, s, d)$, computes the multiplicity of each x_i as a factor of f . If M is the maximal monomial divisor of f , the algorithm also provides blackbox access to f/M .*

To state the multiplicity result in its full generality, we recall the notion of a primitive divisor.

Definition 5.2 ([\[CS26b, Definition 4.1\]](#)). A nonconstant polynomial ϕ is a *primitive divisor* for a class $\mathcal{A}$ of nonzero polynomials if the following conditions hold:

- (1) Every $p \in \mathcal{A}$ can be written as $p = \phi^k q$, where $k \geq 0$ and $\gcd(\phi, q) = 1$.
- (2) No multiple of ϕ that is not associate to ϕ satisfies the first condition.

For a product f of polynomials in $\mathcal{A}$, the multiplicity $v_\phi(f)$ is the exponent of ϕ in its factorization into primitive divisors; equivalently, it is the largest k such that $\phi^k \mid f$. An irreducible polynomial belonging to $\mathcal{A}$ is a primitive divisor for $\mathcal{A}$.

Lemma 5.3 (Multiplicity of factors [\[CS26b, Lemma 4.13\]](#)). *There is a deterministic $\text{poly}(n, (d^2)!, s^{d^3}, \ell)$ -time algorithm that, given blackbox access to a product f of ℓ polynomials in $\mathcal{P}(n, s, d)$ and blackbox access to a primitive divisor ϕ for $\mathcal{P}(n, s, d)$, returns $v_\phi(f)$. For some explicit $m = \text{poly}(n, (d^2)!, s^{d^3})$, evaluations of f on the image of $G_{(m)}$ suffice.*

Moreover, if ϕ is an irreducible polynomial in $\mathcal{P}(n, s, d)$, the algorithm also applies when f is a product of ℓ polynomials in $\mathcal{P}_{\text{Factors}}(n, s, d)$.

The next result tests divisibility of products of sparse polynomials.

Theorem 5.4 (Testing divisibility of products [\[CS26b, Theorem 4.14\]](#)). *There is a deterministic $\text{poly}(n, (d^2)!, s^{d^3}, \ell)$ -time algorithm that, given blackbox access to two polynomials f, g , each a product of ℓ polynomials in $\mathcal{P}(n, s, d)$, decides whether $f \mid g$.*

We shall need the following result of Volkovich. For completeness, we give a proof of the stated arithmetic bound using [Corollary 3.4](#).

Lemma 5.5 (Sparse divisibility [[Vol17](#), Lemma 42]). *Let $f, g \in \mathcal{P}(n, s, d)$ be nonzero, where $n, s, d \geq 1$. Given f and g explicitly, one can deterministically test whether $f \mid g$ using $\text{poly}(n, d, s^d)$ field operations, over any field.*

Proof. Let $h = \gcd(f, g)$ in $\mathbb{F}[\mathbf{x}]$. For each $i \in [n]$, compute the explicit polynomials $F_i = f(\widehat{R}_i)$ and $G_i = g(\widehat{R}_i)$, using the substitutions (5). (If $\mathbb{F}$ is too small, we compute over an extension $\mathbb{E}$ as in [Remark 1.4](#); this changes neither h nor the gcd degrees below, since gcds are unchanged by field extension.) By [Corollary 3.4](#) and Gauss's lemma,

$$\deg_u \gcd_{\mathbb{F}(\mathbf{z})[u]}(F_i, G_i) = \deg_{x_i} h.$$

Thus $f \mid g$ if and only if these degrees equal $\deg_{x_i} f$ for every i . Indeed, the equalities say that the polynomial f/h has degree zero in every variable. The substituted polynomials have five variables and total degree $\text{poly}(n, m, d)$, where $m = \text{poly}(n, d, s^d)$. Computing them and their univariate gcd degrees over $\mathbb{F}(\mathbf{z})$ takes $\text{poly}(n, d, s^d)$ field operations. This proves the claim. $\square$

We next improve the dependence on d for testing whether a sparse polynomial divides a product under a promise and for computing multiplicities, using [Corollary 3.4](#). For both results, we use the substitutions $\widehat{R}_i$ defined in (5), with $m = \text{poly}(n, d, s^d)$.

Theorem 5.6 (Divisibility under a promise). *Let $f = \prod_{r=1}^{\ell} h_r$, where the h_r 's are nonzero polynomials in $\mathcal{P}(n, s, d)$, and let $\phi \in \mathcal{P}(n, s, d)$ be nonconstant. Suppose that either $\phi \mid f$ or $\gcd(\phi, f) = 1$. Then, for any $x_i \in \text{var}(\phi)$,*

$$\phi \mid f \iff \phi(\widehat{R}_i) \mid f(\widehat{R}_i) \text{ in } \mathbb{F}(\mathbf{z})[u].$$

Given ϕ explicitly and blackbox access to f , one can test this divisibility deterministically using $\text{poly}(n, d, s^d, \ell)$ field operations, in arbitrary characteristic.

Proof. We follow the divisibility procedure of [[CS26b](#), Theorem 4.14], using [Corollary 3.4](#) to obtain the improved bound. Write $P_\phi := \phi(\widehat{R}_i)$ and $F_i := f(\widehat{R}_i)$. As in the proof of [Corollary 3.4](#)

$$\deg_u P_\phi = \deg_{x_i} \phi > 0.$$

If $\phi \mid f$, then clearly $P_\phi \mid F_i$. Conversely, suppose $\phi \nmid f$. Then the promise gives $\gcd(\phi, h_r) = 1$ for every r . By [Corollary 3.4](#),

$$\gcd(P_\phi, h_r(\widehat{R}_i)) = 1 \text{ in } \mathbb{F}(\mathbf{z})[u]$$

for every r . Consequently, $\gcd(P_\phi, F_i) = 1$ in this ring. Since P_ϕ has positive u -degree, it cannot divide F_i .

We next explain how to test whether $P_\phi \mid F_i$ in $\mathbb{F}(\mathbf{z})[u]$. We compute P_ϕ from ϕ and recover F_i using dense interpolation. Both are polynomials in five variables of total degree $\text{poly}(n, m, \ell d)$. To avoid computing with rational functions in $\mathbf{z}$ we use pseudo-division (see e.g., [[Knu97](#), §4.6.1]). That is, we let $c = \text{lc}_u(P_\phi)$ and $k = \max(0, \deg_u F_i - \deg_u P_\phi + 1)$, and compute $Q, r \in \mathbb{F}[\mathbf{z}][u]$, with $\deg_u r < \deg_u P_\phi$, such that

$$c^k F_i = Q \cdot P_\phi + r.$$

Since c is invertible in $\mathbb{F}(\mathbf{z})$, it follows that $P_\phi \mid F_i$ in $\mathbb{F}(\mathbf{z})[u]$ if and only if $r = 0$. It is not hard to see that all the polynomials computed in this process have total degree $\text{poly}(n, m, \ell d)$, and thus the whole computation requires $\text{poly}(n, m, \ell d)$ field operations. As $m = \text{poly}(n, d, s^d)$, the claimed bound follows. $\square$

The same corollary improves the multiplicity computation when all the irreducible factors are sparse.

Lemma 5.7 (Computing multiplicities of sparse irreducibles). *There is a deterministic algorithm which, given an explicit irreducible polynomial $\phi \in \mathcal{P}(n, s, d)$ and blackbox access to $f = \prod_{r=1}^{\ell} h_r$, where all the h_r 's are irreducible polynomials in $\mathcal{P}(n, s, d)$, computes $v_{\phi}(f)$ using $\text{poly}(n, d, s^d, \ell)$ field operations, in arbitrary characteristic.*

Proof. We follow the multiplicity procedure of [CS26b, Lemma 4.13], using Corollary 3.4. Choose $x_i \in \text{var}(\phi)$, and write $P_{\phi} := \phi(\widehat{R}_i)$ and $F_i := f(\widehat{R}_i)$. As above, $\deg_u P_{\phi} = \deg_{x_i} \phi > 0$.

Let $e := v_{\phi}(f)$ and write $f = \phi^e H$. Every irreducible factor of H is sparse and coprime to ϕ . Applying Corollary 3.4 to ϕ and each of these factors gives

$$F_i = P_{\phi}^e H(\widehat{R}_i), \quad \gcd(P_{\phi}, H(\widehat{R}_i)) = 1 \quad \text{in } \mathbb{F}(\mathbf{z})[u].$$

Since $\deg_u P_{\phi} > 0$, we obtain

$$e = \max\{k \geq 0 : P_{\phi}^k \mid F_i \text{ in } \mathbb{F}(\mathbf{z})[u]\}.$$

Compute P_{ϕ} from ϕ , recover F_i by dense interpolation, and repeatedly divide F_i by P_{ϕ} . The divisions can be performed in $\mathbb{F}[\mathbf{z}, u]$: P_{ϕ}^e divides F_i since $f = \phi^e H$, whereas P_{ϕ}^{e+1} does not divide even over $\mathbb{F}(\mathbf{z})[u]$. There are at most $\ell + 1$ attempts. Each division is an exact division in $\mathbb{F}[\mathbf{z}, u]$ of five-variate polynomials of degree $\text{poly}(n, m, \ell d)$. With five variables and $m = \text{poly}(n, d, s^d)$, these computations take $\text{poly}(n, d, s^d, \ell)$ field operations. $\square$

6 Factoring products of sparse irreducibles

In this section we deduce Theorem 1.5 from Theorem 1.6. The proof follows the meta-algorithm of [CS26b, §6], with the key ingredient (the rational interpolation theorem [CS26b, Theorem 5.2]) replaced by Theorem 1.6. We shall need a slightly different generator than the one in the first version of this paper [CS26a] to apply Theorem 1.6. Specifically, we require a generator that specializes to each of the hybrid maps R_i of Section 4. We also modify the pruning and multiplicity computations to avoid characteristic restrictions.

Throughout this section, we assume that $\mathbb{F}$ has sufficiently many elements. As usual, the displayed running-time bounds suppress the dependence on coefficient bit complexity and the cost of univariate factorization over $\mathbb{F}$; the latter is accounted for by the factor T in Theorem 2.3. We recall the statement of Theorem 1.5.

Theorem 1.5 (Factoring products of sparse irreducibles). *Let $\mathbb{F}$ be an arbitrary field. There is a deterministic algorithm which, given blackbox access to a product $f = \prod_{i=1}^{\ell} h_i$, where $h_1, \dots, h_{\ell} \in \mathcal{P}(n, s, d)$ are (not necessarily distinct) irreducible polynomials, returns (scalar multiples of) the h_r 's together with their multiplicities. The algorithm performs $\text{poly}(n, (s\ell)^d)$ many field and bit operations.*

Proof. For convenience, we assume $f \in \mathbb{F}[x_0, \mathbf{x}]$ is a product of ℓ irreducible polynomials in $\mathcal{P}(n + 1, s, d)$, with $\mathbf{x} = (x_1, \dots, x_n)$. The univariate case follows directly from univariate factorization, so we assume $n \geq 1$.

We first compute the multiplicity of each x_i ($0 \leq i \leq n$) as a factor of f , and replace f by $f/(\text{maximal monomial divisor})$, using Corollary 5.1. We retain the variable factors and their multiplicities for the output. From now on f is not divisible by any x_i . Assume, without loss of generality, that f depends on x_0 after this step. Write $f = \sum_{j=0}^{\ell d} f_j x_0^j$, where $f_j \in \mathbb{F}[\mathbf{x}]$ and $f_0 \neq 0$.

Consider $\tilde{f}(X, \mathbf{x}) := f(Xf_0(\mathbf{x}), \mathbf{x})/f_0(\mathbf{x})$, which is reverse-monic in X (meaning its constant term with respect to X is exactly 1). By [Claim 2.11](#), $\tilde{f} \in \mathbb{F}[X, \mathbf{x}]$ and we have blackbox access to it. Let $m, \Gamma = G_{(m)}^{\text{KS}}$, and $R_1(\mathbf{y}, \mathbf{z}, u), \dots, R_n(\mathbf{y}, \mathbf{z}, u)$ be as in (4) and (7). Thus $m = \text{poly}(n, d, s^d)$, and $\mathbf{y}, \mathbf{z}$ are independent tuples of four seed variables each. Choose distinct $\eta_1, \dots, \eta_n \in \mathbb{F}$, and let $A_1(v), \dots, A_n(v)$ be their Lagrange interpolation polynomials. Define

$$\Psi(\mathbf{y}, \mathbf{z}, u, v) := \sum_{i=1}^n A_i(v) R_i(\mathbf{y}, \mathbf{z}, u). \quad (16)$$

Then

$$\Psi(\mathbf{y}, \mathbf{z}, u, \eta_i) = R_i(\mathbf{y}, \mathbf{z}, u), \quad \Psi(\mathbf{z}, \mathbf{z}, \Gamma_1(\mathbf{z}), \eta_1) = \Gamma(\mathbf{z}). \quad (17)$$

In particular, Ψ has ten seed variables, its degree is smaller than $3m + 2n$, and every nonzero polynomial in $\mathcal{P}(n, s, d)$ remains nonzero after composition with Ψ .

Let

$$\hat{f}(X, \mathbf{y}, \mathbf{z}, u, v) := \tilde{f}(X, \Psi(\mathbf{y}, \mathbf{z}, u, v)).$$

Using [Lemma 2.4](#) and [Theorem 2.3](#), obtain the factorization of $\hat{f}$ into irreducibles, as a polynomial in eleven variables of total degree $\text{poly}(m, \ell d)$. Since $\hat{f}$ has $O(1)$ variables, this takes $\text{poly}(m, \ell d)$ arithmetic operations and univariate factorization calls over $\mathbb{F}$, on polynomials of degree $\text{poly}(m, \ell d)$. We also recover the explicit polynomial $f_0(\Psi)$ by dense interpolation.

Since $\tilde{f}$ is reverse-monic in X , so is $\hat{f}$. By [Claim 2.9](#), $\hat{f}$ has at most ℓd nonconstant irreducible factors, counted with multiplicity, each of positive X -degree. Hence the number of reverse-monic divisors of $\hat{f}$ of X -degree at most d is at most $\binom{\ell d + d}{d} \leq (e(\ell + 1))^d$.

For each such divisor $\hat{h}$ of positive X -degree, we attempt to recover an irreducible factor $h = \sum_{j=0}^{d_h} c_j(\mathbf{x}) x_0^j \in \mathcal{P}(n + 1, s, d)$ of f corresponding to $\hat{h}$, as follows.

By [Claim 2.11](#), every irreducible factor $h = \sum_{j=0}^{d_h} c_j(\mathbf{x}) x_0^j$ of f with $x_0 \in \text{var}(h)$ gives rise to an irreducible factor $\tilde{h}$ of $\tilde{f}$ over $\mathbb{F}$:

$$\tilde{h}(X, \mathbf{x}) = \sum_{j=0}^{d_h} \frac{c_j(\mathbf{x}) f_0(\mathbf{x})^j}{c_0(\mathbf{x})} X^j \in \mathbb{F}[\mathbf{x}, X].$$

Composing with Ψ , the polynomial $\hat{h} := \tilde{h}(X, \Psi)$ is a reverse-monic divisor of $\hat{f}$. Its X -degree is d_h : the polynomials $c_0(\Psi)$, $c_{d_h}(\Psi)$, and $f_0(\Psi)$ are nonzero. For the last claim, note that f_0 is a product of nonzero polynomials in $\mathcal{P}(n, s, d)$, and specialize Ψ to Γ . Since $d_h \leq d$, $\hat{h}$ is one of the enumerated divisors. It need not be irreducible, which is why we enumerate divisors rather than irreducible factors.

If $\hat{h}$ is correctly identified as the candidate factor, then for $1 \leq j \leq d_h$ the coefficient of X^j in $\hat{h}$ equals

$$e_j = \frac{c_j(\Psi) f_0(\Psi)^j}{c_0(\Psi)}.$$

Form the explicit rational functions

$$Q_j := \frac{e_j}{f_0(\Psi)^j} = \frac{c_j(\Psi)}{c_0(\Psi)}, \quad 1 \leq j \leq d_h.$$

By (17), specializing $v = \eta_i$ in Ψ gives the fractions $c_j(R_i)/c_0(R_i)$, and the further specialization gives $c_j(\Gamma)/c_0(\Gamma)$. These are the substitutions used in the proof of [Theorem 1.6](#). We reduce the

explicit fractions after each specialization before using them. Note that the denominators are nonzero: specializing $\mathbf{y} = \mathbf{z}$ and $u = \Gamma_i(\mathbf{z})$ in $f_0(R_i)$ gives $f_0(\Gamma) \neq 0$.

Since h is irreducible and depends on x_0 , Gauss's lemma gives $\gcd(c_0, c_1, \dots, c_{d_h}) = 1$. Moreover, all the c_j 's belong to $\mathcal{P}(n, s, d)$. The reconstruction of [Theorem 1.6](#) therefore returns $\lambda c_0, \lambda c_1, \dots, \lambda c_{d_h}$ for some $\lambda \in \mathbb{F} \setminus \{0\}$. Form the candidate

$$h^* := \sum_{j=0}^{d_h} (\lambda c_j) x_0^j = \lambda h,$$

and add it to a candidate list $\mathcal{F}(f)$, provided h^* is nonconstant and $(n+1, s, d)$ -sparse, and no polynomial associate to it is already in the list.

A divisor $\widehat{h}$ that does not come from a factor of f may violate the hypotheses of [Theorem 1.6](#). We therefore abort a reconstruction attempt if it exceeds the running time of [Theorem 1.6](#) or if one of its steps fails, for instance a division that should be exact is not. We keep h^* only if it is nonconstant and lies in $\mathcal{P}(n+1, s, d)$. A genuine factor h passes, since all hypotheses of [Theorem 1.6](#) hold for it. Spurious candidates that survive cause no harm: the pruning step below only uses that every candidate lies in $\mathcal{P}(n+1, s, d)$ and that every irreducible factor of f is in the list.

Repeat the procedure with each of the $n+1$ variables $x_0, x_1, \dots, x_n$ playing the role of distinguished variable, skipping variables on which f does not depend. Since every non-monomial irreducible factor of f depends on at least one variable, each such factor appears in $\mathcal{F}(f)$ for the corresponding choice.

Pruning the list. The list $\mathcal{F}(f)$ contains at most $(n+1)(e(\ell+1))^d$ candidates, each nonconstant and $(n+1, s, d)$ -sparse. We first discard every candidate properly divisible by another candidate in the list. Pairwise divisibility of explicit sparse polynomials can be tested in arbitrary characteristic using $\text{poly}(n, d, s^d)$ field operations by [Lemma 5.5](#).

Every irreducible factor of f remains in the list. Moreover, every remaining candidate ϕ either is an associate of an irreducible factor of f , or satisfies $\gcd(\phi, f) = 1$. Indeed, otherwise some irreducible factor h_r of f properly divides ϕ , and h_r is itself present in the original candidate list.

We discard every remaining candidate ϕ that does not divide f , using [Theorem 5.6](#) in $\text{poly}(n, d, s^d, \ell)$ field operations per candidate. The surviving candidates are exactly the irreducible factors of f , up to associates.

Computing multiplicities. For each surviving candidate ϕ , apply [Lemma 5.7](#) to compute $v_\phi(f)$ using $\text{poly}(n, d, s^d, \ell)$ field operations. Together with the variable factors recorded in the first step, these give the complete factorization of the original input, up to scalar multiples.

Running-time analysis. As shown above, the number of reconstruction attempts is at most

$$K := (n+1) \binom{\ell d + d}{d} \leq (n+1)(e(\ell+1))^d,$$

and $|\mathcal{F}(f)| \leq K$. The dense interpolation and factorization of the constant-variate polynomial $\widehat{f}$ take $\text{poly}(n, d, s^d, \ell)$ arithmetic operations and univariate factorization calls over $\mathbb{F}$, with the degrees of the univariate inputs bounded by $\text{poly}(n, d, s^d, \ell)$. Each reconstruction attempt takes $\text{poly}(n, d, s^d, \ell)$ arithmetic operations. The pruning step uses at most $|\mathcal{F}(f)|^2$ sparse divisibility tests, followed by one constant-variate divisibility test for each minimal candidate. The multiplicity computation has the same polynomial overhead per surviving factor. Thus the total number of field

and bit operations, excluding the cost of univariate factorization calls and coefficient bit complexity, is at most

$$K^2 \text{poly}(n, d, s^d, \ell) = \text{poly}(n, d, s^d, (\ell + 1)^d) = \text{poly}(n, (s\ell)^d).$$

The number of univariate factorization calls and their input degrees obey the same bound. This proves the theorem. $\square$

Bibliography

- [BKR⁺25] Somnath Bhattacharjee, Mrinal Kumar, Shanthanu S Rai, Varun Ramanathan, Ramprasad Satharishi, and Shubhangi Saraf. Closure under factorization from a result of Furstenberg. *arXiv preprint arXiv:2506.23214*, 2025. 1
- [BKRS26] Somnath Bhattacharjee, Rishabh Kothary, Shanthanu S. Rai, and Shubhangi Saraf. Deterministic algorithms for low individual degree factors of sparse polynomials, 2026. 6
- [BS25] Pranav Bisht and Nitin Saxena. Derandomization via symmetric polytopes: Poly-time factorization of certain sparse polynomials. *ACM Transactions on Computation Theory*, 17(2):1–20, 2025. 6
- [BSV20] Vishwas Bhargava, Shubhangi Saraf, and Ilya Volkovich. Deterministic factorization of sparse polynomials with bounded individual degree. *Journal of the ACM (JACM)*, 67(2):1–28, 2020. 1, 6
- [BT88] Michael Ben-Or and Prason 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. 1
- [BV25] Pranav Bisht and Ilya Volkovich. On solving sparse polynomial factorization related problems. *Computational Complexity*, 34(1):7, 2025. 1, 6, 10
- [CL11] Annie Cuyt and Wen-shin Lee. Sparse interpolation of multivariate rational functions. *Theoretical Computer Science*, 412(16):1445–1456, 2011. 6
- [CS26a] Aminadav Chuyoon and Amir Shpilka. Factoring products of sparse irreducibles of bounded individual degree via rational interpolation. *Electron. Colloquium Comput. Complex.*, TR26-108, 2026. First version of this paper. 6, 19
- [CS26b] Aminadav Chuyoon and Amir Shpilka. On factorization of sparse polynomials of bounded individual degree. In Dana Moshkovitz, editor, *41st Computational Complexity Conference, CCC 2026, Lisbon, Portugal, August 3-6, 2026*, volume 383 of *LIPIcs*, pages 20:1–20:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2026. Full version in <https://arxiv.org/abs/2603.07589>. 2, 1, 5, 6, 7, 9, 10, 12, 17, 18, 19
- [DST24] Pranjal Dutta, Amit Sinhababu, and Thomas Thierauf. Derandomizing multivariate polynomial factoring for low degree factors. In Amit Kumar and Noga Ron-Zewi, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2024, London School of Economics, London, UK, August 28-30, 2024*, volume 317 of *LIPIcs*, pages 75:1–75:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024. 1

- [For15] Michael A Forbes. Deterministic divisibility testing via shifted partial derivatives. In *2015 IEEE 56th Annual Symposium on Foundations of Computer Science*, pages 451–465. IEEE, 2015. [6](#)
- [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. [1](#)
- [GKS94] Dima Grigoriev, Marek Karpinski, and Michael F Singer. Computational complexity of sparse rational interpolation. *SIAM Journal on Computing*, 23(1):1–11, 1994. [1](#), [6](#)
- [Kal85a] Erich Kaltofen. Computing with polynomials given by straight-line programs I: greatest common divisors. In *Proceedings of the seventeenth annual ACM symposium on Theory of computing*, pages 131–142, 1985. [1](#), [7](#)
- [Kal85b] Erich Kaltofen. Polynomial-time reductions from multivariate to bi- and univariate integral polynomial factorization. *SIAM J. Comput.*, 14(2):469–489, 1985. [7](#)
- [Knu97] Donald E. Knuth. *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*. Addison-Wesley, third edition, 1997. [18](#)
- [KRS24] Mrinal Kumar, Varun Ramanathan, and Ramprasad Saptharishi. Deterministic algorithms for low degree factors of constant depth circuits. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3901–3918. SIAM, 2024. [1](#)
- [KS01] Adam R. Klivans and Daniel Spielman. Randomness efficient identity testing of multivariate polynomials. STOC ’01, page 216–223, New York, NY, USA, 2001. Association for Computing Machinery. [1](#), [10](#)
- [Len85] Arjen Klaas Lenstra. Factoring multivariate polynomials over finite fields. *J. Comput. System Sci.*, 30(2):235–248, 1985. [7](#)
- [LLL82] Arjen Klaas Lenstra, Hendrik Willem Jr. Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261:515–534, 1982. [7](#)
- [SV14] Amir Shpilka and Ilya Volkovich. On reconstruction and testing of read-once formulas. *Theory Comput.*, 10:465–514, 2014. [10](#)
- [vdHL21] Joris van der Hoeven and Grégoire Lecerf. On sparse interpolation of rational functions and gcds. *ACM Commun. Comput. Algebra*, 55(1):1–12, 2021. [6](#)
- [Vol15] Ilya Volkovich. Deterministically factoring sparse polynomials into multilinear factors and sums of univariate polynomials. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2015)*, pages 943–958. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2015. [1](#), [6](#)
- [Vol17] Ilya Volkovich. On some computations on sparse polynomials. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2017)*, pages 48:1–48:21. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2017. [1](#), [6](#), [18](#)

- [vzGK85] Joachim von zur Gathen and Erich Kaltofen. Factoring sparse multivariate polynomials. *Journal of Computer and System Sciences*, 31(2):265–287, 1985. [1](#)
- [vzGS92] Joachim von zur Gathen and Victor Shoup. Computing Frobenius maps and factoring polynomials. *Comput. Complexity*, 2(3):187–224, 1992. [7](#)