

# Deterministic Factorization of Sparse Polynomials with Bounded Individual Degree

Vishwas Bhargava <sup>\*</sup>

Shubhangi Saraf <sup>†</sup>

Ilya Volkovich <sup>‡</sup>

## Abstract

In this paper we study the problem of deterministic factorization of sparse polynomials. We show that if  $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$  is a polynomial with  $s$  monomials, with individual degrees of its variables bounded by  $d$ , then  $f$  can be deterministically factored in time  $s^{\text{poly}(d) \log n}$ . Prior to our work, the only efficient factoring algorithms known for this class of polynomials were randomized, and other than for the cases of  $d = 1$  and  $d = 2$ , only exponential time deterministic factoring algorithms were known.

A crucial ingredient in our proof is a quasi-polynomial sparsity bound for factors of sparse polynomials of bounded individual degree. In particular we show if  $f$  is an  $s$ -sparse polynomial in  $n$  variables, with individual degrees of its variables bounded by  $d$ , then the sparsity of each factor of  $f$  is bounded by  $s^{\mathcal{O}(d^2 \log n)}$ . This is the first nontrivial bound on factor sparsity for  $d > 2$ . Our sparsity bound uses techniques from convex geometry, such as the theory of Newton polytopes and an approximate version of the classical Carathéodory's Theorem.

Our work addresses and partially answers a question of von zur Gathen and Kaltofen (JCSS 1985) who asked whether a quasi-polynomial bound holds for the sparsity of factors of sparse polynomials.

## 1 Introduction

Polynomial factorization is one of the most fundamental questions in computational algebra. The problem of multivariate polynomial factorization asks the following: Given  $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$  a multivariate polynomial over a field  $\mathbb{F}$ , compute each of the irreducible factors of  $f$ . Other than being natural and central, the problem has many applications in areas such as list decoding [29, 12], derandomization [13] and cryptography [3].

There has been a large body of research studying efficient algorithms for this problem (see e.g. [9]) and numerous *randomized* algorithms were designed [10, 14, 15, 17, 9, 16, 8]. However, the question of whether there exist *deterministic* algorithms for this problem remains an important and interesting open question (see [9, 18]).

Another fundamental question in algebraic complexity is the problem of Polynomial Identity Testing (PIT). The problem of PIT asks the following: Given a polynomial  $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$  represented

---

<sup>\*</sup>Department of Computer Science, Rutgers University, Piscataway, NJ 08854. Email: [vishwas1384@gmail.com](mailto:vishwas1384@gmail.com).

<sup>†</sup>Department of Mathematics & Department of Computer Science, Rutgers University, Piscataway, NJ 08854. Research supported in part by NSF grants CCF-1350572 and CCF-1540634. Email: [shubhangi.saraf@gmail.com](mailto:shubhangi.saraf@gmail.com).

<sup>‡</sup>Department of EECS, CSE Division, University of Michigan, Ann Arbor, MI 48109. Email: [ilyavol@umich.edu](mailto:ilyavol@umich.edu).

by a small arithmetic circuit, determine if the polynomial is identically 0. In a recent work, Kopparty et al [20] showed that the problem of derandomizing multivariate polynomial factorization is *equivalent* to the problem of derandomizing polynomial identity testing for general arithmetic circuits. They showed this result in both the white-box and the black-box settings. We already know deterministic PIT algorithms for several interesting classes of arithmetic circuits, and this raises the very natural question of whether we can derandomize *polynomial factoring* for these classes. Perhaps the most natural such class of polynomials is the class of *sparse* polynomials.

The sparsity of  $f$ , denoted  $\|f\|$ , is the number of monomials (with non zero coefficients) appearing in  $f$ . For instance, the sparsity of the polynomial  $x_1 + x_2^3 + x_3x_4 + 20$  is four.

Factoring of sparse polynomials has been studied for over three decades. It was initiated by the work of von zur Gathen and Kaltofen [10] that gives the first *randomized* algorithm for factorization of sparse multivariate polynomials. The runtime of this algorithm has polynomial dependence on the sparsity of the *factors* of the underlying polynomial, and thus, very naturally, this work raised the question of whether one can find efficient bounds on the sparsity of factors of a sparse polynomial.

In this paper, we consider the following two problems: (1) Prove efficient bounds on the sparsity of the factors of sparse polynomials. (2) Derandomize polynomial factorization for sparse polynomials<sup>1</sup>.

Indeed, these are extremely natural questions to study. However already for general fields, we know that one cannot hope to prove a strong sparsity bound for the factors of a sparse polynomial. (We discuss two interesting examples of polynomials whose factors have a big blow-up in the number of monomials in Section 4.1).

In this paper, we focus our attention on the class of sparse polynomials with bounded individual degree, i.e. for some parameter  $d$ , we limit the degree of each variable  $x_i$  to be at most  $d$ .

One very interesting such class of polynomials is the class of sparse multilinear polynomials ( $d = 1$ ). This is the simplest case of sparse polynomials with bounded degree. In [28], Shpilka and Volkovich gave a derandomization for the problem of polynomial factorization for this class. Factor sparsity bounds are fairly easy to show for this class of polynomials, and armed with the sparsity bound and a technique for derandomizing a certain PIT problem that arises, they were able to derandomize factoring in this case. This was extended to the case  $d = 2$  in the work of Volkovich [32], again by first showing a sparsity bound for the factors of polynomials of individual degree 2, and then showing how to derandomize the polynomial factorization problem. For  $d > 2$ , the techniques used by the above works for proving sparsity bounds on the factors of a polynomial seem to break down.

In a recent beautiful work, Oliveira [4] showed that the factors of sparse polynomials of bounded individual degree can be computed by small *depth-7* circuits. This again raises the very natural question: What is the size of the best *depth-2* circuit computing the factors of a sparse polynomial of bounded individual degree. This is precisely the problem of proving sparsity bounds for the factors of a sparse polynomial of bounded individual degree, which is a question we study in this paper.

The other question that we address in this work is the problem of deterministically factoring sparse polynomials of bounded individual degree. A bound on the sparsity of the factors of such a polynomial just implies that the factors will have an efficient representation as a sum of monomials. However in order to actually obtain the factors deterministically, there are several additional de-

---

<sup>1</sup>These questions were raised as important open questions in a recent survey by Forbes and Shpilka [6].

randomization hurdles to overcome.

## 1.1 Our Results

In this paper we give the first deterministic quasi-polynomial time algorithm for factoring sparse polynomials of bounded individual degree. Prior to our work, only efficient randomized factoring algorithms were known for this class of polynomials, and other than for the cases of  $d = 1$  [28] and  $d = 2$  [32] only exponential time deterministic factoring algorithms were known.

A crucial ingredient of our proof is a quasi-polynomial size sparsity bound for factors of sparse polynomials of bounded individual degree  $d$ . In particular, we show that if  $f$  is an  $s$ -sparse polynomial in  $n$  variables with individual degrees of its variables bounded by  $d$ , then  $f$  can be deterministically factored in time  $s^{\text{poly}(d) \log n}$ . This is the first nontrivial bound on factor sparsity for any  $d > 2$ . Our sparsity bound uses techniques from convex geometry, such as the theory of Newton polytopes and an approximate version of the classical Carathéodory's Theorem.

We say that a polynomial  $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$  has *sparsity*  $s$  if it has at most  $s$  nonzero monomials. We say that it has individual degree at most  $d$  if the maximum degree in each of its variables is bounded above by  $d$ .

We formally state below our factor sparsity bound and then our result on deterministic factoring.

**Theorem 1** (Factor Sparsity Bound). *Let  $\mathbb{F}$  be an arbitrary field (finite or otherwise) and let  $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$  be a polynomial of sparsity  $s$  and individual degrees at most  $d$ , then the sparsity of every factor of  $f$  is bounded by  $s^{\mathcal{O}(d^2 \log n)}$ .*

**Remark 1.1.** *Note that for  $d = \text{polylog}(n)$ , we obtain a quasi-polynomial sparsity bound on the factors of  $f$ . Indeed when  $s = \text{poly}(n)$ , for any  $d = o(\sqrt{n}/\log^2 n)$ , we obtain a nontrivial sparsity bound on the factors of  $f$ .*

Given a polynomial  $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$ , the *complete factorization* of  $f$  is a representation of  $f$  as a product  $h_1^{e_1} \cdots h_m^{e_m}$ , where  $h_1, h_2, \dots, h_m$ -s are pairwise coprime, irreducible polynomials, and  $e_1, e_2, \dots, e_m$  are positive integers. This representation is unique up to reordering of the  $h_i$ .

**Theorem 2** (Main). *There exists a deterministic algorithm that given a polynomial  $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$  of sparsity  $s$  and individual degrees at most  $d$ , computes the complete factorization of  $f$ , using  $s^{\mathcal{O}(d^7 \log n)} \cdot \text{poly}(c_{\mathbb{F}}(d^2))$  field operations, where:*

1.  $c_{\mathbb{F}}(d) = \text{poly}(\ell \cdot p, d)$ , if  $\mathbb{F} = \mathbb{F}_{p^\ell}$ .
2.  $c_{\mathbb{F}}(d) = \text{poly}(d, t)$ , where  $t$  is maximum bit-complexity of the coefficients of  $f$ , if  $\mathbb{F} = \mathbb{Q}$ .

**Remark 1.2.** *In the statement of Theorem 2,  $c_{\mathbb{F}}(d)$  denotes the time of the best known algorithm that factors a univariate polynomial of degree  $d$  over  $\mathbb{F}$ .*

**Remark 1.3.** *A more refined version of Theorem 2 is given in Theorem 5.7. The run time for the deterministic factoring algorithm in Theorem 5.7 gives the precise dependence on the sparsity bound for factors of sparse polynomials. In particular, if one could improve the sparsity bound, then one could plug it into the statement of Theorem 5.7 to get an improved run time for the deterministic factoring algorithm.*

## 1.2 Related Work

Over the last three decades, the question of derandomizing sparse polynomial factorization has seen only very partial progress.

The study of sparse polynomial factorization was initiated in [10], where the first *randomized* algorithm for the factorization of sparse polynomials was given. The runtime of this algorithm was polynomial in the sparsity of the factors, and in this work, von zur Gathen and Kaltofen explicitly raised the question of proving improved sparsity bounds for the factors of sparse polynomials.

In [5], Dvir and Oliveira gave an elegant approach for bounding the sparsity of factors of a general sparse polynomial by studying the Newton polytopes of the polynomial and its factors. This approach did not eventually lead to an efficient sparsity bound. However it did inspire our work and our approach of using techniques from convex geometry to bound the factors of sparse polynomials.

In [28], Shpilka and Volkovich gave efficient deterministic factoring algorithms for sparse multilinear polynomials. This result was extended in [31] to the model of sparse polynomials that split into multilinear factors. In [32], Volkovich gave an efficient deterministic factorization algorithm for sparse multiquadratic polynomials. The results [28, 32] correspond to the special case when the individual degree  $d$  equals 1 and 2, respectively. For  $d \geq 3$ , the proof techniques of both these works broke down, and a new approach was needed.

The problem of multivariate polynomial factorization for polynomials of bounded individual degree was also studied in [4]. In this work, among other things, it was shown that if  $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$  is an  $s$ -sparse polynomial of individual degree  $d$ , where  $\mathbb{F}$  is a field of characteristic 0, then any factor of  $f$  can be computed by a depth-7 circuit of size  $\text{poly}(dn^d, s)$ . In particular if  $d$  is constant, then this shows that any factor of  $f$  can be computed by a depth-7 circuit with only a polynomial blow-up in size. This is in contrast to our work, where we want to bound the number of *monomials* in the factors of  $f$ . In other words, we attempt to represent the factors of  $f$  by the more natural class of depth-2 circuits and then understand the size complexity (which we show is quasi-polynomial). We also would like to point out that our result holds over any field  $\mathbb{F}$ .

Another work that is relevant in this context is the work of Kopparty et al [20] which shows an equivalence between the problems of polynomial identity testing (PIT) and polynomial factorization. In particular, it shows that if one can derandomize PIT for the class of general arithmetic circuits, then one can derandomize polynomial factorization for that same class. Since there are several natural examples of classes of polynomials for which we know deterministic PIT algorithms, this naturally raises the question (which was indeed raised in [20]) of whether one can derandomize factoring for the corresponding classes of polynomials. Sparse polynomials are, perhaps, the most natural example of such a class, and our work makes the first significant advance in this direction.

## 1.3 Proof overview

Our proof of the deterministic factoring algorithm has two self-contained and independently interesting components. We first prove a sparsity bound on the factors of sparse polynomials with bounded individual degree (Theorem 1). We then show how such a sparsity bound can be used effectively to derandomize factoring of this same class of polynomials (Theorem 2).

We elaborate on both these components below.

### 1.3.1 Proof Overview for the Sparsity Bound: Theorem 1

Our proof uses tools from convex geometry such as the theory of Newton polytopes and an approximate version of Carathéodory's theorem.

Suppose that  $f, g, h \in \mathbb{F}[x_1, x_2, \dots, x_n]$  are polynomials such that  $f = g \cdot h$ . We want to show that if  $f$  is  $s$ -sparse and with bounded individual degree  $d$ , then  $g$  and  $h$  are both at most  $s'$  sparse, where  $s' = s^{\mathcal{O}(d^2 \log n)}$ .

We will show this by instead showing the following slightly more general result. For a polynomial  $f$ , let  $\|f\|$  denote the sparsity (i.e. the number of nonzero monomials) of  $f$ . Suppose that  $g$  is any polynomial of individual degree  $d$  such that  $\|g\| = s$ , and suppose that  $f = g \cdot h$  (with no assumptions on the degrees of  $f$  and  $h$ ), then  $\|f\| \geq s^{\frac{1}{\mathcal{O}(d^2 \log n)}}$ . In particular, there is no polynomial  $h$  that one can multiply  $g$  with, so that the product  $g \cdot h$  has an overwhelming cancellation of monomials.

**Newton Polytopes and Connection to the Sparsity Bound** Let  $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$  be a polynomial such that:

$$f = \sum a_{i_1 i_2 \dots i_n} x_1^{i_1} x_2^{i_2} \dots x_n^{i_n}.$$

One can consider the set

$$\text{Supp}(f) = \{(i_1, i_2, \dots, i_n) \mid a_{i_1 i_2 \dots i_n} \neq 0\} \subseteq \mathbb{R}^n$$

of exponent vectors of  $f$ . One can then associate a polytope  $P_f \subseteq \mathbb{R}^n$ , called the Newton polytope of  $f$ , which is the convex hull of points in  $\text{Supp}(f)$ .

A classic fact about Newton polytopes that was observed by Ostrowski [22] in 1921 states that if  $f = g \cdot h$ , then  $P_f$  is the Minkowski sum of  $P_g$  and  $P_h$ , where for two polytopes  $A$  and  $B$ , their Minkowski sum  $A + B$  is defined to be the set of points  $\{u + v \mid u \in A \text{ and } v \in B\}$ . Minkowski sums of polytopes are extremely well-studied and it is not difficult to show that the Minkowski sum of two polytopes is itself a polytope. Moreover, if we let  $V(P)$  denote the set of vertices (equivalently corner points) of a polytope  $P$ , then

$$|V(A + B)| \geq \max\{|V(A)|, |V(B)|\}.$$

Once we have these basic facts about Newton polytopes and Minkowski sums, it follows that a lower bound for  $\|f\|$  (in terms of  $\|g\|$ ), follows from a lower bound on  $|V(P_f)|$ , and in particular from a lower bound on  $|V(P_g)|$ . Thus, via the theory of Newton polytopes and Minkowski sums, we see that the monomials of  $g$  that correspond to the vertices of  $P_g$  are very *robust*. There is no way of multiplying  $g$  with any other polynomial and obtaining a cancellation of monomials that will make these special monomials corresponding to the vertices of  $P_g$  “disappear”.

Thus for  $f = g \cdot h$ , our task of lower bounding  $\|f\|$  in terms of  $\|g\|$  has reduced to lower bounding  $|V(P_g)|$ , where  $P_g$  is the Newton polytope of a polynomial  $g$  such that  $\|g\| = s$  and  $g$  has individual degree bounded by  $d$ . Showing a lower bound on  $|V(P_g)|$  will be the main technical core of our proof of the sparsity bound.

We note that this connection between Newton polytopes and sparsity bounds was first made in [5] and indeed it inspired the approach taken in this paper.

**Easy Example with Multilinear Polynomials** We demonstrate the approach of using Newton Polytopes for proving sparsity bounds via the following “toy” example of showing a sparsity bound for multilinear polynomials (i.e, when individual degree is bounded by 1). Suppose that  $f, g, h \in \mathbb{F}[x_1, x_2, \dots, x_n]$  and  $g$  is a multilinear polynomial such that  $f$  is nonzero and  $f = g \cdot h$ . One can give the following easy proof by induction on  $n$ , of the fact that  $\|f\| \geq \|g\|$ . Let  $x_1$  be variable that both  $g$  and  $h$  depend on. (If such a variable doesn’t exist then the sparsity bound is trivial.) Moreover assume WLOG that  $x_1$  doesn’t divide  $h$ , because if it did, then we could factor it out and work with the resulting polynomials. Since  $g$  is multilinear, we can express  $g$  as  $g = g_1 x_1 + g_0$ , where  $g_1$  and  $g_0$  are multilinear polynomials not depending on  $x_1$ , and  $g_1$  is nonzero. Let  $h = h_d x_1^d + \dots + h_0$ , where  $h_d$  and  $h_0$  are nonzero, and the  $h_i$  don’t depending on  $x_1$ . Now,

$$f = (g_1 x_1 + g_0) \cdot (h_d x_1^d + \dots + h_0) = (g_1 \cdot h_d) x_1^{d+1} + \dots + (h_0 \cdot g_0).$$

Thus,  $\|f\| \geq \|g_1 \cdot h_d\| + \|h_0 \cdot g_0\|$ . By the induction hypothesis,  $\|g_1 \cdot h_d\| \geq \|g_1\|$  and  $\|h_0 \cdot g_0\| \geq \|g_0\|$ . It follows that  $\|f\| \geq \|g_1\| + \|g_0\| = \|g\|$ .

Now let us give an alternate proof of the above bound using Newton polytopes. Let  $\text{Supp}(g) \subseteq \{0, 1\}^n$  be the set of exponent vectors of  $g$ . Then notice that no element of  $\text{Supp}(g)$  can be written as a nontrivial convex combination of any other points of  $\text{Supp}(g)$ . In particular, *every* element of  $\text{Supp}(g)$  is a vertex of  $P_g$ ! Thus  $\|f\| \geq |V(P_f)| \geq |V(P_g)| = |\text{Supp}(g)| = \|g\|$ .

**Sparsity Bound from Carathéodory’s Theorem** Note that in general, for an arbitrary polynomial  $g$ , there is no good bound on the number of vertices of  $P_g$  in terms of the number of monomials of  $g$ . For instance one can easily construct examples of polynomials  $g$  with exponential in  $n$  many monomials, and such that  $P_g$  has only  $n$  vertices. Here is an example : consider the polynomial  $P_g = (x_1 + x_2 + \dots + x_n)^n$ . It clearly has exponentially many monomials. However  $P_g$  has only  $n$  vertices, which are the scalings of the coordinate vectors by  $n$ .

In the case when  $g$  has individual degree bounded by  $d$ , we will show that a much nicer bound actually holds. Notice that in this case,  $\text{Supp}(g) \subseteq \{0, 1, \dots, d\}^n$ . We will show that if  $E \subseteq \{0, 1, \dots, d\}^n$  is an arbitrary subset of size  $s$ , then the convex hull of  $E$  (denoted  $\text{CS}(E)$ ) has at least  $s^{\frac{1}{d^2 \cdot \log n}}$  vertices. This will immediately imply our sparsity bound.

To show this bound, we will use an approximate version of Carathéodory’s theorem. The classic version of Carathéodory’s theorem is a fundamental result in convex geometry and it states that if a point  $\mu \in \mathbb{R}^n$  lies in the convex hull of a set  $V$ , then  $\mu$  can be written as the convex combination of at most  $n + 1$  points of  $V$ .

Now, for a set  $E \subseteq \{0, 1, \dots, d\}^n$ , let  $V(E)$  denote the vertices of the convex hull of  $E$ . It is easy to see that  $V(E) \subseteq E$ . Since every point  $\mu \in E$  is a convex combination of elements of  $V(E)$ , by Carathéodory’s theorem, it is a convex combination of at most  $n+1$  elements of  $V(E)$ . Now  $E$  is not an arbitrary collection of points. It is a subset of  $\{0, 1, \dots, d\}^n$ . Suppose we could show the following strengthened (and wishful) Carathéodory’s theorem in this setting: For  $E \subseteq \{0, 1, \dots, d\}^n$ , every point  $\mu \in E$  is a convex combination of at most  $k$  elements of  $V(E)$ , where  $k$  is some bound much smaller than  $n + 1$ . Not only this, the convex combination is a *k-uniform* convex combination, i.e. all the coefficients in the convex combination are equal to  $1/k$ . Notice that in such a case, we can immediately conclude that  $|E| \leq |V(E)|^k$ , since each subset of  $V(E)$  of size  $k$  would “recover” at most one element of  $E$  via a  $k$ -uniform convex combination, and each element of  $E$  must be

recovered by some subset of  $V(E)$  of size  $k$ . If such a result were true for  $k \leq d^2 \log n$  then it would imply our sparsity bound!

It unfortunately (and not too surprisingly) turns out that such a wishful theorem is not true. (Though one needs to work a little to find a counterexample.)

However, very fortunately, something very close does end up being true, and it suffices for our purpose! A suitable “approximate” version of Carathéodory’s theorem suitably applied implies the following: For  $E \subseteq \{0, 1, \dots, d\}^n$ , every point  $\mu \in E$  can be  $\varepsilon$ -approximated by a  $k$ -uniform convex combination of elements of  $V(E)$ , where  $k = \mathcal{O}(d^2 \log n)$ . Again, (and this time truly) one can conclude that  $|E| \leq |V(E)|^k$ , since each subset of  $V(E)$  of size  $k$  could “approximately recover” at most one element of  $E$  via a  $k$ -uniform convex combination (by the triangle inequality the same point cannot approximate two different points of  $E$ ), and each element of  $E$  must be approximately recovered by some subset of  $V(E)$  of size  $k$ . See Theorem 3.6 for the statement of the approximate Carathéodory theorem that we use.

### 1.3.2 Proof Overview for the Factoring Algorithm: Theorem 2

Let  $f \in \mathbb{F}[y, x_1, x_2, \dots, x_n]$  be a multivariate polynomial with individual degrees at most  $d$ . While in general  $f$  could have as many as  $d(n+1)$  factors, our starting point is an observation that if  $f$  is monic<sup>2</sup> in  $y$ , then every factor of  $f$  must also be monic in  $y$ . Consequently,  $f$  has at most  $d$  factors (total). This makes the monic case much easier to handle, and we first show how to factorize  $f$  when  $f$  is monic, and then we show how to extend our algorithm to the general non-monic case.

In the monic case, there are at most  $d$  factors. How would we identify these factors? The traditional approach [10, 15, 17] suggests projecting the polynomial into a low-dimensional space, where the factorization problem is easy. Yet, in order to recover the original factors, the factorization “pattern” of  $f$  should stay the same upon the projection. That is, every irreducible factor should remain irreducible upon the projection. This is typically achieved by the Hilbert Irreducibility Theorem, which shows that a random projection would achieve this goal. Nonetheless, derandomizing the Irreducibility Theorem appears to be a challenging task. Instead, we take a somewhat different approach.

**Finding a “good” Projection** First, we relax the requirement of maintaining the same factorization “pattern” to a requirement that different irreducible factors do not “overlap” upon projection (i.e. have no non-trivial gcd). This is a standard processing step in many factorization algorithms and it is usually taken care of by hitting the Discriminant of the polynomial  $f$  (i.e.  $\Delta_y(f)$ ). Yet this approach for obtaining our deterministic algorithm presents its challenges, and it is particularly tricky in the case that the characteristic of the ambient field  $\mathbb{F}$  is finite (i.e.  $\text{char}(\mathbb{F}) > 0$ ). We show how to go around these problems.

Formally, let  $f \in \mathbb{F}[y, x_1, x_2, \dots, x_n]$  be monic in  $y$  and let  $f(y, \bar{x}) = h_1^{e_1}(y, \bar{x}) \dots h_k^{e_k}(y, \bar{x})$  be the factorization of  $f(y, \bar{x})$ . We will project  $f$  to a univariate polynomial in  $y$  by setting all the variables in  $\bar{x}$  to elements of  $\mathbb{F}$ . In order to guarantee that different irreducible factors have no non-trivial

---

<sup>2</sup>a polynomial is *monic* in a variable  $x_i$  if the leading coefficient of highest degree of  $x_i$  in  $f$  is equal to 1. See definition 2.2 for more details.



gcd after projection, it suffices to find an assignment  $\bar{a} \in \mathbb{F}^n$  such that

$$\forall i \neq j : \gcd(h_i(y, \bar{a}), h_j(y, \bar{a})) = 1.$$

This condition translates into finding a single assignment  $\bar{a}$  that hits (i.e. is a nonzero assignment for) the *Resultant*,  $\text{Res}_y(h_i, h_j)$ , for all  $i \neq j$  (see Section 2.5 for more details). As  $f$  is an  $s$ -sparse polynomial, by Theorem 1, each  $h_i$  is an  $s^{\mathcal{O}(d^2 \log n)}$ -sparse polynomial. Hence, by the properties of the Resultant (Lemma 2.8),  $\text{Res}_y(h_i, h_j)$  is  $s^{\mathcal{O}(d^3 \log n)}$ -sparse polynomial. Consequently, hitting all the pairwise resultants corresponds to hitting their product, which is a (somewhat) sparse polynomial. We handle this in a “black-box” fashion. That is, we iterate over all the points in a hitting set for (somewhat) sparse polynomials (for example using the hitting set of [19]).

**Finding the “right” Partition** As the projection we obtain is no longer required to maintain the same factorization “pattern”, irreducible factors could split into “pieces” (i.e. further factorize upon projection) in a way that the same set of “pieces” can emerge from different polynomials. For example, consider the polynomials  $f(y, x) = (y^2 - x)y$  and  $g = y(y - x)(y + x)$ . These two polynomials have different factorization patterns. However observe that  $f(y, 1) = g(y, 1) = y(y - 1)(y + 1)$ . While in both cases, the different “pieces” of the irreducible factors of  $f$  and  $g$  do not overlap (i.e. no nontrivial gcd), it is not clear how to group the pieces together to recover the factorization pattern of the original polynomial. I.e. just by examining the pieces, we cannot determine what the right partition of the set of factors of  $f(y, 1)$  and  $g(y, 1)$  should be.

We address this problem by recalling and taking advantage of the fact that a monic polynomial of degree  $d$  can split into at most  $d$  pieces! Therefore, there are at most  $d^{\mathcal{O}(d)}$  possible partitions. We find the “right” partition by iterating over all of them till we find the right one.

**Reconstructing the Factors** As before, let  $f \in \mathbb{F}[y, x_1, x_2, \dots, x_n]$  be monic in  $y$  and let  $f(y, \bar{x}) = h_1^{e_1}(y, \bar{x}) \dots h_k^{e_k}(y, \bar{x})$ . Given a “good” projection  $\bar{a}$  and the “right” partition, we will show how to obtain oracle (i.e. “black-box”) access to the polynomials  $h_1, \dots, h_k$ . Once we can do this, as Theorem 1 provides us an upper bound on the sparsity of  $h_i$ -s, we can use a reconstruction algorithm for sparse polynomials to reconstruct  $h_1, \dots, h_k$ , given via an oracle access.

We obtain oracle access to  $h_1, \dots, h_k$  by mirroring the factorization algorithm of [17]. Given an input point  $\bar{b} \in \mathbb{F}^n$  at which we want to compute  $h_1(y, \bar{b}), \dots, h_k(y, \bar{b})$ , the algorithm uses  $\bar{a}$  as an anchor point and draws a line to  $\bar{b}$ . We then obtain a problem of bi-variate factorization, which we know how to solve efficiently. The non-overlapping property of the “pieces” makes it possible to group the pieces together in the same consistent way for every choice of  $\bar{b}$ . Once we can do this, this allows us to evaluate the individual factors at  $\bar{b}$ .

**Testing the Purported Factors** As was discussed earlier, given a polynomial  $f$ , the algorithm will proceed by trying to reconstruct the factors of  $f$  for every projection and every partition. Some of these projections and partitions will return valid factorizations of  $f$  and some might return garbage. We need to prune out the garbage solutions, which we can do as follows: As each factor of  $f$  is “somewhat” sparse (Theorem 1) and there are at most  $d$  of them, given a purported factorization, we can test if it is a good and valid factorization by explicitly multiplying out the polynomials. Clearly, this algorithm will pick up *any* valid factorization of  $f$  (not just the irreducible one). We will select the irreducible factorization using the simple characterization given in Lemma 2.4.



**Factoring General Sparse Polynomials** In order to extend the above algorithm that works in the monic case to the more general case of non-monic polynomials, we use a standard reduction that transforms a general polynomial  $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$  into a monic one  $\hat{f}$ .

More formally, write:  $f = \sum_{j=0}^k f_j \cdot x_n^j$  such that  $f_k \neq 0$  and the  $f_j$ -s do not depend on  $x_n$ . Consider the polynomial  $\hat{f}(y, x_1, \dots, x_{n-1}) = f_k^{k-1} \cdot f(x_1, \dots, x_{n-1}, \frac{y}{f_k})$ . We show that if  $f$  is an  $s$ -sparse polynomial with individual degrees at most  $d$ , then  $\hat{f}$  is an  $(s^d)$ -sparse polynomial, monic in  $y$ , with individual degrees at most  $d^2$ .

Finally, we show that  $\hat{f}$  contains all the factors of  $f$  that depend on  $x_n$ , while  $f_k$  contains the remaining factors. We recover these remaining factors by recursively factoring  $f_k$ . Observe that  $f_k$  depends on at most  $n - 1$  variables.

**Organization of Paper** In the next section, we recall some algebraic tools and algebraic algorithms that will be useful for us. In Section 3, we discuss properties of polytopes and their relation to factor sparsity. Section 4 contains the proof of the sparsity bound along with a discussion on its tightness. We present and analyze the deterministic factoring algorithm in Section 5. We conclude with some open questions in Section 6.

## 2 Preliminaries

### 2.1 Algebraic Tool Kit

Let  $\mathbb{F}$  denote a field, finite or otherwise, and let  $\overline{\mathbb{F}}$  denote its algebraic closure.

### 2.2 Polynomials

A polynomial  $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$  *depends* on a variable  $x_i$  if there are two inputs  $\overline{\alpha}, \overline{\beta} \in \overline{\mathbb{F}}^n$  differing only in the  $i^{\text{th}}$  coordinate for which  $f(\overline{\alpha}) \neq f(\overline{\beta})$ . We denote by  $\text{var}(f)$  the set of variables that  $f$  depends on. We say that  $f$  is *similar* to  $g$  and denote by it  $f \sim g$  if  $f = \alpha g$  for some  $\alpha \neq 0 \in \mathbb{F}$ .

For a polynomial  $f(x_1, \dots, x_n)$ , a variable  $x_i$  and a field element  $\alpha$ , we denote with  $f|_{x_i=\alpha}$  the polynomial resulting from substituting  $\alpha$  to  $x_i$ . Similarly given a subset  $I \subseteq [n]$  and an assignment  $\overline{a} \in \mathbb{F}^n$ , we define  $f|_{\overline{x}_I=\overline{a}_I}$  to be the polynomial resulting from substituting  $a_i$  to  $x_i$  for every  $i \in I$ .

**Definition 2.1** (Line). *Given  $\overline{a}, \overline{b} \in \mathbb{F}^n$  we define a line passing through  $\overline{a}$  and  $\overline{b}$  as  $\ell_{\overline{a}, \overline{b}} : \mathbb{F} \rightarrow \mathbb{F}^n$ ,  $\ell_{\overline{a}, \overline{b}}(t) \triangleq (1 - t) \cdot \overline{a} + t \cdot \overline{b}$ . In particular,  $\ell_{\overline{a}, \overline{b}}(0) = \overline{a}$  and  $\ell_{\overline{a}, \overline{b}}(1) = \overline{b}$ .*

**Definition 2.2** (Degrees, Leading Coefficients). *Let  $x_i \in \text{var}(f)$ . We can write:  $f = \sum_{j=0}^d f_j \cdot x_i^j$  such that  $\forall j : x_i \notin \text{var}(f_j)$  and  $f_d \neq 0$ . The leading coefficient of  $f$  w.r.t to  $x_i$  is defined as  $\text{lc}_{x_i}(f) \triangleq f_d$ . The individual degree of  $x_i$  in  $f$  is defined as  $\deg_{x_i}(f) \triangleq d$ . We say that  $f$  is monic in a variable  $x_i$  if  $\text{lc}_{x_i}(f) = 1$ . We say that  $f$  is monic if it is monic in some variable.*

It easy to see that for every  $f, g \in \mathbb{F}[x_1, x_2, \dots, x_n]$  and  $i \in [n]$  it holds:  $\text{lc}_{x_i}(f \cdot g) = \text{lc}_{x_i}(f) \cdot \text{lc}_{x_i}(g)$ .

## 2.3 Factors and Divisibility

Let  $f, g \in \mathbb{F}[x_1, x_2, \dots, x_n]$  be polynomials. We say that  $g$  *divides*  $f$ , or equivalently  $g$  is a factor of  $f$ , and denote it by  $g \mid f$  if there exists a polynomial  $h \in \mathbb{F}[x_1, x_2, \dots, x_n]$  such that  $f = g \cdot h$ . We say that  $f$  is *irreducible* if  $f$  is non-constant and cannot be written as a product of two non-constant polynomials.

Given the notion of divisibility, we define the gcd of a set of polynomials in the natural way: we define it to be the highest degree polynomial dividing them all (suitably scaled)<sup>3</sup>. Given the notion of irreducibility we can state the important property of the uniqueness of factorization.

**Lemma 2.3** (Uniqueness of Factorization). *Let  $h_1^{e_1} \cdot \dots \cdot h_k^{e_k} = g_1^{e'_1} \cdot \dots \cdot g_{k'}^{e'_{k'}}$  be two factorizations of the same non-zero polynomial into irreducible, pairwise coprime factors. Then  $k = k'$  and there exists a permutation  $\sigma : [k] \rightarrow [k]$  such that  $h_i \sim g_{\sigma(i)}$  and  $e_i = e'_{\sigma(i)}$  for  $i \in [k]$ .*

Suppose that  $f$  is monic in  $x_i$ . It is easy to see  $f$  can be written as a product of monic factors. Therefore, we can specialize Lemma 2.3 to consider the *unique monic factorization* of  $f$  as:  $f = h_1^{e_1} \cdot \dots \cdot h_k^{e_k}$  where  $h_i$ -s are irreducible, monic, pairwise coprime factors.

The following lemma provides a characterization of all irreducible, pairwise coprime factorizations of any polynomial.

**Lemma 2.4.** *Consider the function  $\Phi : \mathbb{N}^* \rightarrow \mathbb{N}$ : given  $\bar{e} = (e_1, \dots, e_k)$ ,  $\Phi(\bar{e}) \triangleq 2 \cdot \sum_{i=1}^k e_i - k$ . Let  $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$  be a polynomial and let  $f = h_1^{e_1} \cdot \dots \cdot h_k^{e_k}$  a factorization of  $f$  (not necessarily irreducible or coprime), where  $h_i$ -s are non-constant and  $e_i \geq 1$ . Then all irreducible, pairwise coprime factorizations of  $f$  correspond to those that maximize  $\Phi(\bar{e})$ .*

*Proof.* First, observe that by Uniqueness of Factorization, all the all irreducible, pairwise coprime factorizations of  $f$  result in the same value of  $\Phi(\bar{e})$ . Next, we show that in the factorization that maximize  $\Phi(\bar{e})$ , all the  $h_i$ -s must be irreducible and coprime. Assume the contrary. We have two possible cases:

- There exists  $i$  such that  $h_i$  is reducible. That is,  $h_i$  can be written as  $h_i = u_i \cdot v_i$ , where  $u_i, v_i$  are non-constant polynomials. Now, consider a different factorization of  $f$  where we replace  $h_i^{e_i}$  by  $u_i^{e_i}$  and  $v_i^{e_i}$ . The value of  $\Phi$  under the new factorization will increase by  $2e_i - 1 \geq 1$ .
- There exists  $i$  and  $j$  such that  $h_i$  and  $h_j$  are not coprime. We can assume w.l.o.g that both  $h_i$  and  $h_j$  are irreducible. Therefore,  $h_i = \alpha \cdot h_j$  for some  $\alpha \in \mathbb{F}$ . Consider a different factorization of  $f$  where we replace  $h_i^{e_i}$  and  $h_j^{e_j}$  by a single factor:  $(\alpha^{\frac{e_i}{e_i+e_j}} \cdot h_j)^{e_i+e_j}$ . The value of  $\Phi$  under the new factorization will increase by 1.

□

---

<sup>3</sup>Such a polynomial is unique up to scaling, and one can fix a canonical polynomial in this class for instance by requiring that the leading monomial has coefficient 1. With this definition, two polynomials are pairwise coprime if their gcd is of degree 0, and in particular the gcd equals 1.

## 2.4 Sparse Polynomials

In this section we discuss sparse polynomials, their properties and some related efficient algorithms which leverage these properties.

An  $s$ -sparse polynomial is polynomial with at most  $s$  (non-zero) monomials. We denote by  $\|f\|$  the sparsity of  $f$ . In this section we list several results related to sparse polynomials. We begin with an efficient reconstruction algorithm for sparse polynomials.

**Lemma 2.5** ([19]). *Let  $n, s, d \in \mathbb{N}$ . There exists a deterministic algorithm that given  $n, s, d$  and an oracle access to an  $s$ -sparse polynomial  $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$  of degree  $d$ , uses  $\text{poly}(n, s, d, \log |\mathbb{F}|)$  field operations and outputs  $f$  (in its monomial representation).*

In particular the above lemma shows the existence of an efficient hitting set for sparse polynomials. We now give a lemma that shows the existence of an efficient hitting set for a product of sparse polynomials. Indeed it was shown in [27] that if there is an efficient hitting set for any class of polynomials, then one can construct an efficient hitting set for a product of few polynomials from that class. Thus we immediately get the following lemma.

**Lemma 2.6** ([19, 27, 24]). *There exists a deterministic algorithm that given  $n, s, d, k \in \mathbb{N}$  outputs a set  $\mathcal{SP}_{(n,s,d,k)}$  of size  $\text{poly}(n, s, d, k)$  such that any set of (at most)  $k$  non-zero  $s$ -sparse polynomials  $f_1, \dots, f_k \in \mathbb{F}[x_1, x_2, \dots, x_n]$  with individual degrees at most  $d$  have a common non-zero in  $\mathcal{SP}_{(n,s,d,k)}$ . In other words, there exists  $\bar{a} \in \mathcal{SP}_{(n,s,d,k)}$  such that  $\forall i : f_i(\bar{a}) \neq 0$ .*

Note that in the above lemma, we could have replaced individual degree by total degree, and the result would have still held, since the total degree is at most a factor of  $n$  more than the individual degree. However, in our applications, we will usually use an individual degree bound, and hence we stated the lemma in terms of individual degree.

As another simple corollary of [19], we obtain an efficient algorithm for sparse polynomial division, given an upper bound on the sparsity of the quotient polynomial. In other words, if  $f, g$  are sparse polynomials such that  $f = g \cdot h$ , then given black-box access to  $f$  and  $g$ , one can recover  $h$  (as long as it is also sparse). This is because given black-box access to  $f$  and  $g$ , one can simulate black-box access to  $h$ . One can then use [19] to interpolate and recover  $h$ . If  $h$  ends up being not sparse, then this algorithm would just reject. Moreover, given a candidate sparse polynomial  $h$ , it is easy to verify whether it is indeed the quotient polynomial of  $f$  and  $g$ , but just multiplying out  $h \cdot g$  and comparing with  $f$ .

**Lemma 2.7** (Corollary of [19]). *Let  $n, s, d, t \in \mathbb{N}$ . Let  $f, g \in \mathbb{F}[x_1, x_2, \dots, x_n]$  be  $s$ -sparse polynomials of degree at most  $d$ . Then there exists an algorithm that given  $f, g$  and  $t$  uses  $\text{poly}(n, d, s, t)$  field operations and computes the quotient polynomial of  $f$  and  $g$ , if it is a  $t$ -sparse polynomial. That is, if  $f = gh$  for some  $h \in \mathbb{F}[x_1, x_2, \dots, x_n]$ ,  $\|h\| \leq t$ , then the algorithm outputs  $h$ . Otherwise, the algorithm rejects.*

## 2.5 GCD and Resultants

Let  $f = a_d y^d + a_{d-1} y^{d-1} + \dots + a_0$  and  $g = b_e y^e + b_{e-1} y^{e-1} + \dots + b_0$  be polynomials of  $y$ -degree exactly  $d$  and  $e$ , respectively. Consider the  $(d+e) \times (d+e)$  Sylvester Matrix whose first  $e$  columns contain  $e$  shifts of the vector of coefficients  $(a_d, \dots, a_0, 0, \dots, 0)$ , and next  $d$  columns contain  $d$  shifts of the vector of coefficients  $(b_e, \dots, b_0, 0, \dots, 0)$ . That is,

$$\text{Res}_y(f, g) = \begin{vmatrix} a_d & 0 & \dots & 0 & b_e & 0 & \dots & 0 \\ a_{d-1} & a_d & \dots & 0 & b_{e-1} & b_e & \dots & 0 \\ a_{d-2} & a_{d-1} & \ddots & 0 & b_{e-2} & b_{e-1} & \ddots & 0 \\ \vdots & \vdots & \ddots & a_d & \vdots & \vdots & \ddots & b_e \\ \vdots & \vdots & \dots & a_{d-1} & \vdots & \vdots & \dots & b_{e-1} \\ a_0 & a_1 & \dots & \vdots & b_0 & b_1 & \dots & \vdots \\ 0 & a_0 & \ddots & \vdots & 0 & b_0 & \ddots & \vdots \\ \vdots & \vdots & \ddots & a_1 & \vdots & \vdots & \ddots & b_1 \\ 0 & 0 & \dots & a_0 & 0 & 0 & \dots & b_0 \end{vmatrix}_{(d+e) \times (d+e)}$$

This representation of resultant ensures that if  $f$  and  $g$  are sparse polynomials in  $\mathbb{F}[x_1, x_2, \dots, x_n]$  with small individual degree (in  $y$ ), then sparsity of  $\text{Res}_y(f, g)$  is bounded. We will use the following properties of resultant (For more info, see [11, Chap. 7])

**Lemma 2.8** (Resultant Properties). *Let  $f, g \in \mathbb{F}[y, x_1, x_2, \dots, x_n]$  be monic in  $y$ ,  $s$ -sparse polynomial with individual degrees at most  $d$ . Then:*

1.  $\text{Res}_y(f, g)(\bar{x})$  is an  $(2ds)^{2d}$ -sparse polynomial over  $\mathbb{F}[x_1, x_2, \dots, x_n]$  with individual degrees at most  $2d^2$ .
2. For every  $\bar{a} \in \mathbb{F}^n$ :  $\text{Res}_y(f|_{\bar{x}=\bar{a}}, g|_{\bar{x}=\bar{a}}) = \text{Res}_y(f, g)(\bar{a})$ .
3.  $\gcd(f, g) \neq 1$  iff  $\text{Res}_y(f, g) \equiv 0$ .

**Definition 2.9.** For a field  $\mathbb{F}$  we denote by  $c_{\mathbb{F}}(d)$  the time of the best known algorithm that factors a univariate polynomial of degree  $d$  over  $\mathbb{F}$ .

**Lemma 2.10** (Univariate factoring). *Let  $f(x) \in \mathbb{F}[x]$  be a univariate polynomial of degree  $d$  then by the well known algorithms of Lenstra-Lenstra-Lovasz [21] and Berlekamp [2, 26, 9, 7],  $f$  can be factorized in time  $c_{\mathbb{F}}(d)$ . where:*

1.  $c_{\mathbb{F}}(d) = \text{poly}(\ell \cdot p, d)$ , if  $\mathbb{F} = \mathbb{F}_{p^\ell}$ .
2.  $c_{\mathbb{F}}(d) = \text{poly}(d, t)$ , where  $t$  is maximum bit-complexity of the coefficients of  $f$ , if  $\mathbb{F} = \mathbb{Q}$ .

The next result which is implicit in many factorization algorithms, exhibits an efficient factorization algorithm for certain regime of parameters. In particular, for polynomials with constantly-many variables and a polynomial degree.

**Lemma 2.11** (Implicit in [15], see also [30]). *There exists a deterministic algorithm that given a  $r$ -variate, degree  $d$  polynomial  $f$  over  $\mathbb{F}$  outputs its irreducible factors. The runtime of the algorithm is  $(c_{\mathbb{F}}(d))^{\mathcal{O}(r)}$ .*

We will use this lemma for  $r = 2$  (i.e. bivariate factoring) in our deterministic factorization algorithm.

### 3 Polytopes and Polynomials

In this section we will discuss various properties of polytopes, in particular the Newton polytope. These will be crucial ingredients in our proof of the sparsity bound for factors of sparse polynomials. The main results that we will discuss and develop are:

1. If  $f, g, h$  are polynomials such that  $f = g \cdot h$  then the sparsity of  $f$  is lower bounded by  $\max\{|V(P_g)|, |V(P_h)|\}$ , where  $P_g$  and  $P_h$  are the Newton polytopes of  $g$  and  $h$  respectively, and where for a polytope  $P$ ,  $V(P)$  denotes the set of vertices of  $P$ .
2. The convex hull of any subset of  $\{0, 1, \dots, d\}^n$  must have “many” vertices (i.e. corner points). We will prove this as a corollary of an approximate version of Carathéodory’s theorem.

Our approach to bounding the sparsity of factors of a polynomial using the theory of polytopes, and in particular Item 1 (as stated above) was inspired by a connection of the theory of polytopes to sparsity bounds that was observed by Dvir and Oliveira [5].

For a finite set of points  $v_1, v_2, \dots, v_k \in \mathbb{R}^n$ , their *convex span*, which we denote by  $CS(v_1, \dots, v_k)$  is the set defined by

$$CS(v_1, v_2, \dots, v_k) = \left\{ \sum_{i=1}^k \lambda_i v_i \mid \lambda_i \geq 0 \text{ and } \sum_{i=1}^k \lambda_i = 1 \right\}.$$

A set  $P \subseteq \mathbb{R}^n$  is called a *polytope* if there is a finite set of points  $v_1, v_2, \dots, v_k \in \mathbb{R}^n$  such that  $P = CS(v_1, v_2, \dots, v_k)$ . For a polytope  $P$ , and a point  $a \in P$ , we say that  $a$  is a *vertex* of  $P$  if it **cannot** be written as  $a = \lambda u + (1 - \lambda)v$  for any  $u, v \in P \setminus \{a\}$  and  $\lambda \in [0, 1]$ . Alternatively, a vertex of  $P$  is face of dimension 0. We let  $V(P)$  denote the set of vertices of  $P$ .

It is an easy to verify, and a basic fact about polytopes, that if  $P$  is a polytope, then  $P = CS(V(P))$ . Moreover, if  $P = CS(v_1, v_2, \dots, v_k)$  then  $V(P) \subseteq \{v_1, v_2, \dots, v_k\}$ .

(For more details see [33] Propositions 2.2 and 2.3)

### 3.1 The Newton Polytope and Minkowski Sum

**Definition 3.1.** Given two polytopes  $P_1$  and  $P_2$  in  $\mathbb{R}^n$ , we define their Minkowski Sum  $P_1 + P_2$  to be the set of points given by

$$P_1 + P_2 = \{v_1 + v_2 \mid v_1 \in P_1 \text{ and } v_2 \in P_2\}.$$

The following is a classic fact about the Minkowski sum of two polytopes. It basically says that the Minkowski sum of two polytopes is itself a polytope, and the number of vertices of each of the original polytopes is a lower bound on the number of vertices of the Minkowski sum. See [5] (Theorem 3.12, Corollary 3.13), and [25] for the formal details of a proof. After we state the result, we will provide an informal proof sketch which also gives some intuition for why the result holds.

**Proposition 3.2.** Let  $P_1$  and  $P_2$  be polytopes in  $\mathbb{R}^n$ . Then their Minkowski sum  $P_1 + P_2$  is a polytope and

$$|V(P_1 + P_2)| \geq \max\{|V(P_1)|, |V(P_2)|\}.$$

*Proof sketch.* Let  $u_1, u_2, \dots, u_{k_1}$  be the vertices of  $P_1$  and  $v_1, v_2, \dots, v_{k_2}$  be the vertices of  $P_2$ . Now, any element of  $P_1 + P_2$  is of the form  $\mu_1 + \mu_2$ , where  $\mu_1$  is a convex combination of  $u_1, u_2, \dots, u_{k_1}$  and  $\mu_2$  is a convex combination of  $v_1, v_2, \dots, v_{k_2}$ . It follows easily from this that  $\mu_1 + \mu_2$  is a convex combination of  $V(P_1) + V(P_2) = \{u + v \mid u \in V(P_1), v \in V(P_2)\}$ . Thus  $P_1 + P_2 \subseteq CS(V(P_1) + V(P_2))$  and it is also easily to see that  $CS(V(P_1) + V(P_2)) \subseteq P_1 + P_2$ . Thus  $P_1 + P_2 = CS(V(P_1) + V(P_2))$ , and hence it is a polytope.

We will now show that for every  $u \in V(P_1)$ , there exists  $v \in V(P_2)$  such that  $u + v \in V(P_1 + P_2)$ .

Fix  $u \in P_1$ . Since  $u \in V(P_1)$ , there exists a hyperplane  $H$  that passes through  $u$  and such that all the rest of  $P_1$  lies on one side of  $H$ . In particular there is a degree one polynomial  $h$  such that  $h(u) = 0$  and for every  $u' \in P_1$  such that  $u' \neq u$ ,  $h(u') > 0$ . (The hyperplane  $H$  is the zero set of  $h$ .) Moreover such an  $h$  and  $H$  can be chosen that are “generic” in the sense that none of the one-dimensional or higher faces of  $P_1$  or  $P_2$  can be translated to lie within  $H$ . (Such an  $H$  can be obtained by doing a small random perturbation to the original  $H$  about the point  $u$ .)

Now for any real number  $a$ , consider the polynomial  $h_a = h + a$ . Let  $H_a$  be the zero set of  $h_a$ . If  $a$  is a large enough real valued number, then for any  $v' \in P_2$ ,  $h_a(v') > 0$ . Now slowly decrease the value of  $a$  till for the first time, for some value  $b$ ,  $H_b$  touches  $P_2$  at a single point, which will be some vertex  $v$ . Since  $H$  was a generic hyperplane, this we can ensure that  $H_b$  only touches  $P_2$  at a single point. Thus we will have the property that  $h_b(v) = 0$  and for all  $v' \in P_2$  such that  $v' \neq v$ ,  $h_b(v') > 0$ .

We will now show that for this choice of  $v$ ,  $u + v \in V(P_1 + P_2)$ . Let  $c$  be the constant term of  $h$ . Then  $h = h' + c$ , where  $h'$  is a homogeneous degree one polynomial. Then  $h_b = h' + c + b$ . Consider the degree one polynomial  $h^* = h' + 2c + b$ , and observe that  $h^*(u + v) = h'(u + v) + 2c + b = (h'(u) + c) + (h'(v) + c + b) = 0$ . Moreover for any  $u' \in P_1$ ,  $v' \in P_2$  such that  $(u, v) \neq (u', v')$ , it must hold that  $h^*(u' + v') > 0$ . Thus it must be that  $u + v \in V(P_1 + P_2)$ .

Observe also that the vertex  $u + v$  of  $P_1 + P_2$  cannot be expressed as  $u' + v'$  for any other  $u' \in P_1$  and  $v' \in P_2$  such that  $(u', v') \neq (u, v)$ . This is because  $h^*(u + v) = 0$  but for  $(u', v') \neq (u, v)$ ,  $h^*(u' + v') > 0$ . Thus corresponding to the vertex  $u \in P_1$ , we have identified a vertex  $u + v$  of  $P_1 + P_2$  which cannot be expressed in any other way as a sum of a vertex of  $P_1$  and a vertex of  $P_2$ .

Since we can do this for each vertex of  $P_1$ , it follows that  $|V(P_1)| \leq |V(P_1 + P_2)|$ . By symmetry,  $|V(P_2)| \leq |V(P_1 + P_2)|$ , and the result follows.  $\square$

For a polynomial  $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$ , suppose that

$$f = \sum a_{i_1 i_2 \dots i_n} x_1^{i_1} x_2^{i_2} \cdots x_n^{i_n}.$$

For each coefficient  $a_{i_1 i_2 \dots i_n} \neq 0$ , we say that the exponent vector  $(i_1, i_2, \dots, i_n)$  is in the *support* of  $f$ , when viewed as a vector in  $\mathbb{R}^n$ . We define  $\text{Supp}(f)$  to be the set of all support vectors of  $f$ , i.e.

$$\text{Supp}(f) = \{(i_1, i_2, \dots, i_n) \mid a_{i_1 i_2 \dots i_n} \neq 0\}.$$

The convex hull of the set  $\text{Supp}(f)$  is defined to be the *Newton polytope* of  $f$ , which we denote by  $P_f$ .

The following classic fact was observed by Ostrowski [22] in 1921. It states that if a polynomial  $f$  factors as  $g \cdot h$ , then the Newton polytope of  $f$  is the Minkowski sum of the Newton polytopes of  $g$  and  $h$ . (See also [5] (Proposition 3.16) for a proof.)

**Proposition 3.3.** *Let  $f, g, h \in \mathbb{F}[x_1, x_2, \dots, x_n]$  be polynomials such that  $f = g \cdot h$ . Then*

$$P_f = P_g + P_h.$$

**Remark 3.4.** *This result will eventually play a crucial role in the proof of our sparsity bound. Note that we want to show that if a certain polynomial  $f$  is sparse, then  $g$  and  $h$  are also sparse. We will show that if  $g$  (or  $h$ ) is “dense”,  $f$  must also be “dense”. If we can show that  $P_f$  has many vertices (i.e. corner points), then this will give us a lower bound on the number of monomials in  $f$ . Since  $P_f = P_g + P_h$ , a lower bound on  $|V(P_g)|$  (or  $|V(P_h)|$ ) is a lower bound on  $|V(P_f)|$ . Thus we then only need to lower bound  $|V(P_g)|$ , which we will show how to do using the results of the next section.*

As an immediate corollary of the above two propositions, we easily recover the following basic bound relating the sparsity of polynomials to the Newton polytopes of its factors. (This bound was observed by Dvir and Oliveira in [5]).

**Corollary 3.5.** *Let  $f, g, h \in \mathbb{F}[x_1, x_2, \dots, x_n]$  be polynomials such that  $f = g \cdot h$ . Then*

$$\|f\| \geq |V(P_f)| \geq \max\{|V(P_g)|, |V(P_h)|\}.$$

*Proof.* By Proposition 3.3,  $P_f = P_g + P_h$ , and hence by Proposition 3.2,

$$|V(P_f)| \geq \max\{|V(P_g)|, |V(P_h)|\}.$$

Since  $P_f = \text{CS}(\text{Supp}(f))$ , thus  $V(P_f) \subseteq \text{Supp}(f)$ . Hence  $|V(P_f)| \leq \|f\|$  and the result follows.  $\square$

It is worth noting that if  $d = 1$ , that is when  $g$  (or  $h$ ) is multilinear, then every point in  $P_g$  is a corner point. Hence,  $|V(P_g)| = \|g\|$  and by Prop. 3.2  $\|f\| \geq \|g\|$ .



### 3.2 An approximate Carathéodory's Theorem

Carathéodory's theorem is a fundamental result in convex geometry, and it states that if a point  $\mu \in \mathbb{R}^n$  lies in the convex hull of a set  $U$ , then  $\mu$  can be written as the convex combination of at most  $n + 1$  points of  $U$ .

In order to prove our sparsity bound, we will be using an “approximate” version of Carathéodory's theorem. The version that we use appears in [1]<sup>4</sup>. It essentially states that if a point  $\mu \in \mathbb{R}^n$  lies in the convex hull of a set  $U$ , then  $\mu$  can be *uniformly*  $\varepsilon$ -approximated in the  $\ell_\infty$  norm by a vector that is the convex combination of only  $\frac{\log n}{\varepsilon^2}$  points of  $U$ .

We first introduce some notation that we will use. For a set of vectors  $U = \{u_1, u_2, \dots, u_m\} \subseteq \mathbb{R}^n$ , let  $CS(U)$  denote the convex hull of  $U$ . (Note that for a finite set, the convex span of a set of vectors is the same as the convex hull of the vectors. Since in the rest of the paper we will only be dealing with finite sets, we will use the terms convex span and convex hull interchangeably). A vector  $\mu \in CS(U)$  is defined to be  $k$ -uniform with respect to  $U$  if there exists a multiset  $S$  of  $[m]$  of size at most  $k$  such that  $\mu = \frac{1}{k} \sum_{i \in S} u_i$ .

We present proof of the approximate Carathéodory theorem for completeness after stating the theorem. There are other approximate versions of the Carathéodory Theorem that appear in the literature, often in terms of  $\ell_p$  norms where  $2 \leq p < \infty$ . The version below is for the  $\ell_\infty$  norm, and its proof is fairly straightforward.

**Theorem 3.6** ([1], Theorem 3). *Given a set of vectors  $U = \{u_1, u_2, \dots, u_m\} \subseteq \mathbb{R}^n$  with  $\max_{u \in U} \|u\|_\infty \leq 1$ , and  $\varepsilon > 0$ . For every  $\mu \in CS(U)$  there exists an  $\mathcal{O}\left(\frac{\log n}{\varepsilon^2}\right)$  uniform vector  $\mu' \in CS(U)$  such that  $\|\mu - \mu'\|_\infty \leq \varepsilon$ .*

*Proof.* Since  $\mu \in CS(U)$ , thus  $\mu = \sum_{i=1}^m a_i u_i$ , where for each  $i \in [m]$ ,  $a_i \geq 0$  and  $\sum_{i=1}^m a_i = 1$ . Now consider the following probability distribution on  $U$ , where the probability of sampling  $u_i$  is  $a_i$ . Pick  $t = \left(\frac{\log n}{\varepsilon^2}\right)$  samples independently from this distribution and let the resulting vectors be  $v_1, v_2, \dots, v_t$ . Let

$$\mu' = \frac{\sum_{i=1}^t v_i}{t}.$$

**Claim 3.7.** *For any coordinate  $j \in [n]$ ,  $\Pr[|\mu_j - \mu'_j| > \varepsilon] < 1/n$ .*

*Proof.* It follows immediately from the Chernoff-Hoeffding bounds applied to  $t$  independent samples  $Y_1, Y_2, \dots, Y_t$  of the random variable  $Y$ , where for each  $i \in [m]$ ,  $Y = (u_i)_j$  with probability  $a_i$ . Then clearly  $\mathbb{E}[Y] = \mu_j$ , and  $\mu'_j = \frac{\sum_{i=1}^t Y_i}{t}$ . Then by the Chernoff-Hoeffding inequality,

$$\Pr[|\mu_j - \mu'_j| > \varepsilon] < e^{-2\varepsilon^2 t} < 1/n.$$

□

Once we have the claim, then a simple union bound over the coordinates implies that with positive probability,  $\|\mu - \mu'\|_\infty \leq \varepsilon$ , and hence a suitable  $\frac{\log n}{\varepsilon^2}$  uniform vector  $\mu' \in CS(U)$  exists. □

---

<sup>4</sup>There is actually a small typo in the version of the theorem in [1], and the statement below fixes it.

## 4 Sparsity Bound

In this section we prove the sparsity bound.

**Theorem 4.1** (The Bound of Factor Sparsity). *There exists a non-decreasing function  $\xi(n, s, d) \leq s^{\mathcal{O}(d^2 \log n)}$  such that if  $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$  is a polynomial of sparsity  $s$  and individual degrees at most  $d$ , and if  $f = g \cdot h$ , for  $g, h \in \mathbb{F}[x_1, x_2, \dots, x_n]$ , then the sparsity of  $g$  is upper bounded by  $\xi(n, s, d)$ .*

Before presenting the proof of the sparsity bound, we first show how to apply the approximate Carathéodory's Theorem 3.6 to show that the convex hull of any subset of  $\{0, 1, \dots, d\}^n$  must have *many* vertices (i.e. corner points).

**Theorem 4.2.** *Let  $E \subseteq \{0, 1, \dots, d\}^n$ . Let  $t = |V(CS(E))|$ . Then there exists an absolute constant  $C$  such that  $t^{Cd^2 \log n} \geq |E|$ .*

*Proof.* Let  $E_d \subseteq [0, 1]^n$  be the set obtained by taking  $E$  and scaling every member of it down coordinate-wise by a factor of  $d$ . Let  $\varepsilon = 1/3d$ . Let  $U = V(CS(E)) \subseteq E$  be the set of vertices of the convex span of  $E$ . Similarly, let  $U_d = V(CS(E_d)) \subseteq E_d$ . Then clearly  $|U| = |U_d|$ .

By Theorem 3.6, for every  $u_d \in E_d$ , since  $u_d \in CS(U_d)$ , thus there exists an  $\mathcal{O}\left(\frac{\log n}{\varepsilon^2}\right) = \mathcal{O}(d^2 \log n)$  uniform vector  $u'_d \in CS(U_d)$  such that  $\|u_d - u'_d\|_\infty \leq 1/3d$ .

Observe that for two distinct vectors  $u_d, v_d \in E_d$ ,  $\|u_d - v_d\|_\infty \geq 1/d$ . Hence if  $u'_d \in CS(U_d)$  is an  $\mathcal{O}(d^2 \log n)$  uniform vector such that  $\|u_d - u'_d\|_\infty \leq 1/3d$  and if  $v'_d \in CS(U_d)$  is an  $\mathcal{O}(d^2 \log n)$  uniform vector such that  $\|v_d - v'_d\|_\infty \leq 1/3d$ , Then by the triangle inequality, we must have that  $u'_d \neq v'_d$ .

The total number of  $\mathcal{O}(d^2 \log n)$  uniform vectors that can be generated by the set  $U_d$  is  $|U_d|^{\mathcal{O}(d^2 \log n)}$ . Moreover we have just shown that one can generate  $|E_d|$  distinct  $\mathcal{O}(d^2 \log n)$  uniform vectors from  $U_d$ .

Thus there is an absolute constant  $C$  such that Thus,

$$|U_d|^{Cd^2 \log n} \geq |E_d|$$

and we thus conclude that  $t^{Cd^2 \log n} \geq |E|$ .

□

**Remark 4.3.** *In fact, the dependence on  $\log n$  in the theorem above is necessary. In particular, there is a set  $E \subseteq \{-1, 0, 1\}^n$  such that the number of corner points in the convex hull of  $E$  is  $n$ , but  $|E| = n^{\Omega(\log n)}$ . However, it is not clear if such polytopes yield a polynomial with  $\xi(n, s, d) = s^{\Theta(d^2 \log n)}$ . An example of such a set (and the resulting polytope) was shared with us in [23], and we describe it below.*

**Claim 4.4** ([23]). *There is a set  $E \subseteq \{-1, 0, 1\}^n$  s.t.  $|V(CS(E))| = n$  and  $|E| = n^{\Omega(\log n)}$ .*

*Proof.* Let  $m$  be a positive integer. Let  $n = 2^m$  and let  $H$  be the  $n \times n$  Hadamard matrix. More precisely,  $H$  is the matrix over the Reals with entries being 1 or  $-1$  such that if we index the rows and columns of  $H$  by the elements of  $\mathbb{F}_2^m$ , and then the  $(a, b)$  entry of  $H$  is  $(-1)^{\langle a, b \rangle}$ , for all  $a, b \in \mathbb{F}_2^m$ .

Let  $V \subseteq \{-1, +1\}^n$  be the set of column vectors of  $H$ . We will show that the convex span of  $V$  contains at least  $n^{\Omega(\log n)}$  distinct elements of  $\{-1, 0, 1\}^n$ , and this will suffice to prove the claim.

Recall that each element of  $V$  is indexed by an element of  $\mathbb{F}_2^m$ .

We will show that for each  $S \subseteq \mathbb{F}_2^m$  that is a linear subspace, if we take the uniform convex span of the elements of  $V$  that correspond to the elements of  $S$ , then we get an element of  $\{0, 1\}^n$ . Moreover, distinct subspaces give rise to distinct elements of  $\{0, 1\}^n$ . Since the number of subspaces is  $n^{\Omega(\log n)}$ , the result then follows.

Now, let  $S \subseteq \mathbb{F}_2^m$  be a linear subspace, and let  $u_S$  be the characteristic vector of this subspace. We need to show that,

$$\frac{1}{|S|} H \cdot u_S \subseteq \{0, 1\}^n.$$

Let  $T = \{b \in \mathbb{F}_2^m : \langle a, b \rangle = 0, \forall a \in S\}$ . With slight abuse of notation let  $(H \cdot u_S)_b$  corresponds to the entry in the  $b$ -th coordinate. Notice that, if  $b \in T$ , then  $(H \cdot u_S)_b = \sum_{a \in S} (-1)^{\langle a, b \rangle} = |S|$ .

On the other hand, if  $b \notin T$ , then  $\langle a_o, b \rangle = 1$  for some  $a_o \in S$ . Thus, for each  $a \in S$ , we have that  $(-1)^{\langle a+a_o, b \rangle}$  and  $(-1)^{\langle a, b \rangle}$  to have different signs. Hence,  $(H \cdot u_S)_b = 0$  in this case.

Thus,

$$\frac{1}{|S|} H \cdot u_S \subseteq \{0, 1\}^n,$$

and the coordinates that equal 1 are precisely those that correspond to the orthogonal subspace of  $S$ . Thus distinct subspaces  $S$  give rise to distinct vectors in  $\{0, 1\}^n$ .  $\square$

**Remark 4.5.** *In order to obtain a polytope with non-negative coordinates, one can simply shift all the coordinates by 1.*

We now prove Theorem 4.1.

*Proof of Theorem 4.1.* Let  $\|g\|$  denote the sparsity of  $g$ . Thus  $g$  has  $\|g\|$  monomials. Let  $\text{Supp}(f), \text{Supp}(g) \subseteq \{0, 1, \dots, d\}^n$  denote the sets of exponent vectors of  $f$  and  $g$ , respectively.

Let  $t_g = |V(CS(\text{Supp}(g)))|$ . Thus  $t_g$  denotes the number of vertices of the polytope which is the convex span of  $\text{Supp}(g)$ . Similarly let  $t_f = |V(CS(\text{Supp}(f)))|$ . By Theorem 4.2,

$$t_g \geq \|g\| \frac{1}{Cd^2 \log n}.$$

Now, by Corollary 3.5,  $t_g \leq t_f$ . Moreover, since  $V(CS(\text{Supp}(f))) \subseteq E_f$ , thus  $t_f \leq |E_f|$ , which equals the sparsity of  $f$ ,  $\|f\|$ . Hence

$$\|f\| \geq \|g\| \frac{1}{Cd^2 \log n}$$

and the theorem follows.  $\square$

## 4.1 Tightness of Sparsity Bound

In this section we see some examples of polynomials that have factors with a significantly larger number of monomials than the original polynomials. In the case where we do not bound the individual degree of the polynomials, the factors can have a superpolynomial number of monomials.

Interestingly, the examples we will see are also tight cases of Prop. 3.2.

The following example was noted by von zur Gathen and Kaltofen [10].

**Example 4.6** ([10]). *Let*

$$f(x) = \prod_{i=1}^n (x_i^d - 1),$$

$$g(x) = \prod_{i=1}^n (1 + x_i + \dots + x_i^{d-1})$$

Notice that,  $g$  is a factor of  $f$ , but  $\|f\| = 2^n$  and  $\|g\| = d^n$ . Thus letting  $s$  denote the sparsity of  $f$ , notice that  $\|g\| = s^{\log d}$ , where  $d$  is the individual degree of  $f$ .

Indeed, for fields of characteristic 0, this is the best “blow-up” of the sparsity that we are aware of.

Our next example works for fields of positive characteristic, say  $\mathbb{F}_p$ , and uses the Frobenius action of powering by  $p$ . In this example we see a much bigger “blow-up” than in the previous example.

**Example 4.7.** *Let  $f \in \mathbb{F}_p[x_1, \dots, x_n]$ ,  $p$ -prime and let  $0 < d < p$ .*

$$f(x) = x_1^p + x_2^p + \dots + x_n^p,$$

$$g(x) = (x_1 + x_2 + \dots + x_n)^d$$

Notice again that  $g$  is a factor of  $f$ , but  $\|f\| = n$  and  $\|g\| = \binom{n+d-1}{d} \approx n^d$ . Thus if  $s$  denotes sparsity of  $f$ , then  $\|g\| = s^d$ .

The above example is particularly interesting because it shows that for general sparse polynomials, with no bound on the individual degree (for instance if the individual degree can be as large as  $n$ ), the factors of a polynomial can have exponentially more monomials than the original polynomial! Thus there is no hope of proving an efficient sparsity bound for general sparse polynomials.

Note however, that this example only applies to fields of certain characteristics. For instance for fields of characteristic 0, the previous example might be the one with the worst possible blowup, and hence for such fields, an efficient sparsity bound for general sparse polynomials might still hold. However any proof of such a sparsity bound must be able to take advantage of the properties of the underlying field. The techniques for the sparsity bound proved in this paper are oblivious to the underlying field, and thus, given Example 4.7, the best possible sparsity bound for factors of a polynomial that one can hope to show with such techniques is of the form  $\mathcal{O}(s^d)$ .

## 5 Factoring Algorithm

In this section, we give our deterministic factorization algorithm for sparse polynomials with small individual degree, thus proving Theorem 2. The runtime of the algorithm strongly depends on the bound in Theorem 4.1. To emphasize this dependence, we state our results in terms of  $\xi(n, s, d)$ . Theorem 1 follows by instating the upper bound.

As outlined in Section 1.3.2, we first focus on monic polynomials. Then we show how to extend the algorithm to general polynomials.

### 5.1 Black-box Factoring of Sparse Monic Polynomials (given some advice)

In this section we give an algorithm that takes as input a sparse monic polynomial  $f(y, \bar{x})$  of bounded individual degree, as well as some additional information about its factorization pattern, and then outputs (in some sense) blackbox access to its factors.

The algorithm mirrors that black-box factorization algorithm of [17].

The algorithm assumes that it is given an assignment  $\bar{a} \in \mathbb{F}^n$  for which no two distinct coprime factors of  $f(y, \bar{x})$  have non-trivial gcd, when we set  $\bar{x} = \bar{a}$ , and it is given the correct partition of the factors of  $f(y, \bar{a})$  (i.e. the partition gives the grouping of the factors of  $f(y, \bar{a})$  that will correspond to the factors of  $f$ ). The algorithm outputs evaluations of the irreducible factors of  $f$  at any input  $(y_0, \bar{b}) \in \mathbb{F}^{n+1}$ . More precisely, for any  $\bar{b} \in \mathbb{F}^n$ , and any irreducible factor  $h_i(y, \bar{x})$  of  $f$ , the algorithm will output the univariate polynomial  $h_i(y, \bar{b})$  which can then be evaluated at  $y_0$ .

Given an input point  $\bar{b} \in \mathbb{F}^n$ , the algorithm uses  $\bar{a}$  as an anchor point and draws a line to  $\bar{b}$ . Next, the algorithm computes a bi-variate factorization of the polynomial  $f(y, \ell_{\bar{a}, \bar{b}}(t))$  (see Definition 2.1). Finally, the algorithm outputs the black-boxes for each factor of  $f$  by matching the factors of  $f(y, \ell_{\bar{a}, \bar{b}}(t))$  to the factors of  $f(y, \bar{a})$ . We will describe our black-box factoring algorithm below:

**Input:**  $s$ -sparse monic (in  $y$ ) polynomial  $f \in \mathbb{F}[y, x_1, x_2, \dots, x_n]$  of individual degree at most  $d$ .

Assignments:  $\bar{a}, \bar{b} \in \mathbb{F}^n$

Univariate Polynomials:  $g_1(y), g_2(y), \dots, g_r(y)$

Partition:  $A_1 \cup A_2 \cup \dots \cup A_m = [r]$

Exponent Vector:  $\bar{e} = (e_1, e_2, \dots, e_m) \in [d]^m$

**Output:** Univariate Polynomials:  $\varphi_1(y), \varphi_2(y), \dots, \varphi_m(y)$

```

1  $\tilde{f}(y, t) \leftarrow f(y, \ell_{\bar{a}, \bar{b}}(t));$ 
2 Compute the bi-variate factorization of  $\tilde{f}(y, t) = f_1^{v_1}(y, t) \cdot f_2^{v_2}(y, t) \cdots f_{r'}^{v_{r'}}(y, t);$  /* wlog the
   polynomials are monic in  $y$  */
3 for  $i \leftarrow 1$  to  $m$  do
4    $\tilde{h}_i(y, t) \leftarrow 1;$ 
5   for  $k \leftarrow 1$  to  $r'$  do
6     if there exists  $j \in A_i$  s.t  $g_j(y) \mid f_k(y, 0)$  then  $\tilde{h}_i(y, t) \leftarrow \tilde{h}_i(y, t) \cdot f_k^{v_k/e_i}(y, t);$ 
7   end
8 end
9 return  $\tilde{h}_1(y, 1), \tilde{h}_2(y, 1), \dots, \tilde{h}_m(y, 1);$ 
```

**Algorithm 1:** Black-Box Evaluation of Factors

**Lemma 5.1** (Black-box Factorization). *Let  $f(y, \bar{x}) \in \mathbb{F}[y, x_1, x_2, \dots, x_n]$  be a polynomial monic in  $y$  with individual degrees at most  $d$ . Suppose  $f(y, \bar{x})$  can be written as  $f(y, \bar{x}) = \prod_{i=1}^m h_i^{e_i}(y, \bar{x})$  such that  $\gcd(h_i, h_{i'}) = 1$  for  $i \neq i'$ . Then given:*

1. *a point  $\bar{a} \in \mathbb{F}^n$  such that  $\forall i \neq i' : \text{Res}_y(h_i, h_{i'}) (\bar{a}) \neq 0$*
2. *monic irreducible polynomials  $g_1(y), g_2(y), \dots, g_r(y)$*
3. *a partition  $A_1 \dot{\cup} A_2 \dot{\cup} \dots \dot{\cup} A_m = [r]$  such that for all  $i \in [m] : h_i(y, \bar{a}) = \prod_{j \in A_i} g_j(y)$*
4. *exponent vector  $\bar{e} = (e_1, e_2, \dots, e_m) \in [d]^m$*

*and a point  $\bar{b} \in \mathbb{F}^n$ , Algorithm 1 computes  $h_1(y, \bar{b}), h_2(y, \bar{b}), \dots, h_m(y, \bar{b})$ , using  $\text{poly}(n, c_{\mathbb{F}}(d))$  field operations.*

*Proof.* We claim that for each  $i \in [m] : \tilde{h}_i(y, t) = h_i(y, \ell_{\bar{a}, \bar{b}}(t))$  and hence

$$\tilde{h}_i(y, 1) = h_i(y, \ell_{\bar{a}, \bar{b}}(1)) = h_i(y, \bar{b}).$$

Since,  $f(y, \bar{x}) = \prod_{i=1}^m h_i^{e_i}(y, \bar{x})$ , substituting  $\bar{x} = \ell_{\bar{a}, \bar{b}}(t)$  implies that  $\tilde{f}(y, t) = \prod_{i=1}^m h_i^{e_i}(y, \ell_{\bar{a}, \bar{b}}(t))$ . Fix  $k \in [r']$ . By Uniqueness of Factorization (Lemma 2.3),

$$\exists i : f_k(y, t) \mid h_i(y, \ell_{\bar{a}, \bar{b}}(t)).$$

Thus setting  $t = 0$ , we get that

$$f_k(y, 0) \mid h_i(y, \bar{a}).$$

Now since  $h_i(y, \bar{a})$  is a product of irreducible polynomials  $g_j(y)$ , for  $j \in A_i$ , thus

$$\exists j \in A_i : g_j(y) \mid f_k(y, 0).$$

The last step follows from Pre-condition 3 and Uniqueness of Factorization (Lemma 2.3). Now suppose  $\exists i'$  and  $j' \in A_{i'} : g_{j'}(y) \mid f_k(y, 0)$ . It follows that  $g_{j'}(y) \mid h_{i'}(y, \bar{a})$ .

Thus,  $\gcd(h_i(y, \bar{a}), h_{i'}(y, \bar{a})) \neq 1$ . By Lemma 2.8:  $\text{Res}_y(h_i, h_{i'}) (\bar{a}) = \text{Res}_y(h_i|_{\bar{x}=\bar{a}}, h_{i'}|_{\bar{x}=\bar{a}}) = 0$ . By Pre-condition 1, this is only possible if  $i = i'$ .

Now, fix  $i$ . Let  $f_k(y, t)$  and  $u_k$  be an irreducible factor of  $h_i(y, \ell_{\bar{a}, \bar{b}}(t))$  and its degree in the latter, respectively. Now observe that  $f_k(y, t)$  doesn't divide any other  $h_{i'}(y, \ell_{\bar{a}, \bar{b}}(t))$ : suppose it did, then setting  $t = 0$  and repeating the previous argument we would get a contradiction. Consequently,  $v_k = u_k \cdot e_i$  and hence:

$$f_k^{u_k}(y, t) = f_k^{v_k/e_i}(y, t)$$

and

$$f_k^{v_k/e_i}(y, t) \mid \tilde{h}_i(y, t).$$

We conclude that  $h_i(y, \ell_{\bar{a}, \bar{b}}(t)) \mid \tilde{h}_i(y, t)$ . The claim follows by observing that :

$$\prod_{i=1}^m \tilde{h}_i^{e_i}(y, t) = f_1^{v_1}(y, t) \cdot f_2^{v_2}(y, t) \cdots f_{r'}^{v_{r'}}(y, t) = \tilde{f}(y, t) = \prod_{i=1}^m h_i^{e_i}(y, \ell_{\bar{a}, \bar{b}}(t)).$$

For the runtime, observe that  $m, r, r' \leq d$  and clearly  $d = \mathcal{O}(c_{\mathbb{F}}(d))$ . □

## 5.2 Factoring Sparse Monic Polynomials (without advice)

With the black-box factoring algorithm of the previous subsection, we get blackbox access to the irreducible factors of the input monic sparse polynomial, and we can use a reconstruction algorithm to reconstruct the actual factors. The caveat is that black-box factorization algorithm of the previous section assumes that it is given some additional information: an assignment  $\bar{a} \in \mathbb{F}^n$  for which no two distinct factors of  $f(y, \bar{x})$  have non-trivial gcd, when we set  $\bar{x} = \bar{a}$ , and the correct partition of the factors of  $f(y, \bar{a})$ .

In this section we show that the advice is actually a member of a small set that can be computed, and hence one can just “guess” the advice! Since  $f(y, \bar{a})$  has at most  $d$  factors, the number of possible partition is  $d^{\mathcal{O}(d)}$ . Hence we can “guess” the correct partition by trying out all the possibilities. In terms of finding  $\bar{a}$  as above, the following lemma shows that there exists a small set of points  $S \subseteq \mathbb{F}^n$  that contain a point  $\bar{a}$  with the required properties for every monic sparse polynomial of degree  $d$ .

**Lemma 5.2.** *Let  $f \in \mathbb{F}[y, x_1, x_2, \dots, x_n]$  be monic in  $y$ ,  $s$ -sparse polynomial with individual degrees at most  $d$  and let  $f(y, \bar{x}) = h_1^{e_1}(y, \bar{x}) \dots h_k^{e_k}(y, \bar{x})$  be the unique monic factorization of  $f(y, \bar{x})$ . Then there exists a set  $S$  of size  $|S| = (n \cdot \xi(n, d, s))^{\mathcal{O}(d)}$  such that for any  $f$  as above there exists an assignment  $\bar{a} \in \mathbb{F}^n$  satisfying  $\forall i \neq i' : \text{Res}_y(h_i, h_{i'}) (\bar{a}) \neq 0$ .*

We defer the proof of the lemma to the end of the section.

Given a polynomial  $f$ , the algorithm will proceed by trying to reconstruct the factors of  $f$  for every projection in  $S$  and every partition. Given a purported factorization, we can test it by explicitly multiplying out the polynomials. Clearly, the algorithm will pick up *any* valid factorization of  $f$  (not just the irreducible one). We will select the irreducible factorization using the simple characterization given in Lemma 2.4.

**Input:**  $s$ -sparse polynomial  $f \in \mathbb{F}[y, x_1, x_2, \dots, x_n]$ , monic in  $y$ , of individual degree at most  $d$ .

**Output:** monic irreducible factors  $h_1, h_2, \dots, h_m$ , and  $e_1, e_2, \dots, e_m$  such that  $f = h_1^{e_1} \dots h_m^{e_m}$

1. For each  $\bar{a} \in S$  (from Lemma 5.2), subset  $I \subseteq [d]$ ,  $m' \in [d]$ , a non-empty *partition* of  $I$ :  $A_1 \cup A_2 \cup \dots \cup A_{m'} = I$ , and exponent vector  $\bar{e}' = (e'_1, e'_2, \dots, e'_{m'}) \in [d]^{m'}$ :

(a) Compute the monic univariate factorization  $f(y, \bar{a}) = \prod_{j=1}^r g_j(y)$  (Using Lemma 2.10)

(b) Call Algorithm 1 with  $f, \bar{a}, \{A_i\}_{i \in [m]}, \bar{e}$  and  $\{g_j(y)\}_{j \in I}$ .

(c) Invoke the reconstruction algorithm from Lemma 2.5 with  $n' = n, s' = \xi(n, d, s), d' = d$  using the above as an oracle to reconstruct the polynomials  $h'_1(y, \bar{x}), \dots, h'_{m'}(y, \bar{x})$ .

(d) Test that  $f \equiv h'^{e'_1}_1 \cdot h'^{e'_2}_2 \dots h'^{e'_{m'}}_{m'}$  factorization. (Via explicit multiplication)

2. Return a factorization that maximizes the expression  $\Phi(\bar{e}) \triangleq 2 \cdot \sum_{i=1}^{m'} e'_i - m'$ . /\* Pick the most “refined” factorization \*/

**Algorithm 2:** Sparse Monic Polynomial Factorization



**Lemma 5.3.** *Let  $f(y, \bar{x}) \in \mathbb{F}[y, x_1, x_2, \dots, x_n]$  be a polynomial, monic in  $y$ , with individual degrees at most  $d$ . Given  $f$ , Algorithm 2 computes the unique monic factorization of  $f$ . That is, the algorithm outputs coprime, monic irreducible polynomials  $h_1, h_2, \dots, h_m$ , and  $e_1, e_2, \dots, e_m$  such that  $f = h_1^{e_1} \cdots h_m^{e_m}$ , using at most  $(n \cdot \xi(n, d, s))^{\mathcal{O}(d)} \cdot \text{poly}(\text{c}_{\mathbb{F}}(d))$  field operations.*

*Proof.* Let  $f = h_1^{e_1} \cdots h_m^{e_m}$  be the unique monic factorization of  $f$ . By definition,  $\forall i \neq i' : \gcd(h_i, h_{i'}) = 1$ ,  $\forall i : e_i \leq d$  and  $m \leq d$ . We first claim that as the algorithm iterates over all settings of  $\bar{a}, I, m', \bar{e}'$  and the partition, one of these “guesses” satisfies the pre-conditions of Lemma 5.1.

By Lemma 5.2, there exists  $\bar{a} \in S$  (where  $S$  is the set from Lemma 5.2) such that  $\forall i \neq i' : \text{Res}_y(h_i, h_{i'}) (\bar{a}) \neq 0$ . Consider the monic univariate factorization:

$$\prod_{j=1}^r g_j(y) = f(y, \bar{a}) = \prod_{i=1}^m h_i^{e_i}(y, \bar{a}).$$

Clearly,  $r \leq d$ , and the claim follows from the uniqueness of factorization (Lemma 2.10). Therefore, by Lemma 5.1, given this guess, Algorithm 1 will produce oracle access for the polynomials  $h_1, \dots, h_m$ . By Theorem 4.1,  $\forall i : \|h_i\| \leq \xi(n, s, d)$ . Therefore, the reconstruction algorithm from Lemma 2.5 will, indeed, output the polynomials  $h'_1, \dots, h'_m$  such that  $\forall i : h'_i \equiv h_i$ , which will pass the subsequent tests.

Let  $f \equiv h_1^{e'_1} \cdots h_m^{e'_m}$  be the factorization returned by the algorithm. By Lemma 2.4, the polynomials  $h'$  are irreducible and pairwise coprime. The final claim follows by uniqueness of factorization.

**Runtime Analysis:** By Lemma 5.2, there are  $\text{poly}(n \cdot \xi(n, d, s))^{\mathcal{O}(d)} \cdot d^{\mathcal{O}(d)}$  iterations. We outline the runtime of each step in a iteration:

1. By Lemma 2.10 -  $\text{c}_{\mathbb{F}}(d)$ .
2. By Lemma 5.1 -  $\text{poly}(n, \text{c}_{\mathbb{F}}(d))$  per query.
3. By Lemma 2.5 -  $\text{poly}(n, \xi(n, d, s), d)$  time and queries.
4. By Theorem 4.1 -  $\xi(n, d, s)^{\mathcal{O}(d)}$ .

Putting all together:  $n^{\mathcal{O}(d)} \cdot \xi(n, d, s)^{\mathcal{O}(d)} \cdot \text{poly}(\text{c}_{\mathbb{F}}(d))$ . □

We now give the proof of Lemma 5.2.

*Proof of Lemma 5.2.* By Theorem 4.1, for each  $i \in [k] : \|h_i\| \leq \xi(n, d, s)$ . For  $i, i' \in [k]$ , consider the polynomials:

$$f_{(i, i')}(\bar{x}) \triangleq \text{Res}_y(h_i, h_{i'}) (\bar{x}).$$

Fix  $i, i' \in [k]$  such that  $i \neq i'$ . By definition,  $f_{(i, i')} \not\equiv 0$ . Moreover, by Lemma 2.8,  $f_{(i, i')}$  is  $(2d \cdot \xi(n, d, s))^{2d}$ -sparse polynomial with individual degrees at most  $2d^2$ . As  $k \leq d$ , by Lemma 2.6,  $\mathcal{SP}_{(n, (2d \cdot \xi(n, d, s))^{2d}, 2d^2, d^2)}$  contains a common non-zero for all  $f_{(i, i')}$ -s. The claim about the size follows from Lemma 2.6. □

### 5.3 Factoring General Sparse Polynomials

In this section we show how to extend the factorization algorithm for monic sparse polynomials to general sparse polynomials. We begin by showing how to convert a (general) sparse polynomial with “small” individual degrees into a “somewhat” sparse monic polynomial of a “slightly larger” individual degrees.

**Definition 5.4.** Let  $f(x_1, \dots, x_n, x_{n+1}) \in \mathbb{F}[x_1, x_2, \dots, x_{n+1}]$  and let  $k \leq d$  denote the degree of  $x_{n+1}$  in  $f$ . Let  $f_k \triangleq \text{lc}_{x_{n+1}}(f)$ . We define:  $\hat{f}(y, x_1, \dots, x_n) \triangleq f_k^{k-1} \cdot f(x_1, \dots, x_n, \frac{y}{f_k})$ .

**Lemma 5.5.** Suppose  $f$  is an  $s$ -sparse polynomial with individual degrees at most  $d$ . Then function  $\hat{f}$  is an  $(s^d)$ -sparse polynomial in  $\mathbb{F}[y, x_1, x_2, \dots, x_n]$ , monic in  $y$  with individual degrees at most  $d^2$ .

*Proof.* Write:  $f = \sum_{j=0}^k f_j \cdot x_{n+1}^j$  such that  $\forall j, x_{n+1} \notin \text{var}(f_j)$ . Then

$$\hat{f} = \sum_{j=0}^k f_j \cdot f_k^{k-1} \cdot y^j / f_k^j = y^k + \sum_{j=0}^{k-1} f_j \cdot f_k^{k-1-j} \cdot y^j.$$

Observe that for every  $x_i : \deg_{x_i}(f_j \cdot f_k^{k-1-j}) \leq d + d(k-1) \leq d^2$ . For the sparsity of  $\hat{f}$ :

$$\|\hat{f}\| = 1 + \sum_{j=0}^{k-1} \|f_j \cdot f_k^{k-1-j}\| \leq \sum_{j=0}^k \|f_j\| \cdot \|f_k\|^{k-1} \leq \sum_{j=0}^k \|f_j\| \cdot s^{k-1} = \|f\| \cdot s^{k-1} \leq s^k \leq s^d.$$

□

In addition to the question regarding the sparsity of the polynomial  $\hat{f}$ , there are two follow-up questions we need to address:

1. How are the factors of  $\hat{f}$  related the original factors of  $f$ ?
2. As the degree of  $y$  in  $\hat{f}$  is at most  $d$ , we can recover at most  $d$  factors, while  $f$  could potentially have  $dn$  factors! How can we recover the remaining factors?

The following lemma provides the answers to both questions.

**Lemma 5.6.** Let  $f(\bar{x}, x_{n+1}) = \prod_{i=1}^{m'} h_i^{e_i}(\bar{x}, x_{n+1}) \cdot \prod_{l=m'+1}^m h_l^{e_l}(\bar{x})$  and  $f_k(\bar{x}) = \prod_{j=1}^r w_j^{\beta_j}(\bar{x})$  be pair-wise coprime, irreducible factorizations of  $f$  and  $f_k$ , respectively such that  $x_{n+1} \in \text{var}(h_i)$  iff  $i \in [m']$ . Furthermore, let  $\hat{f}(y, \bar{x}) = \prod_{j=1}^{\hat{m}} \hat{h}_j^{\hat{e}_j}(y, \bar{x})$  be the unique monic factorization of  $\hat{f}$ . Then

1.  $\hat{m} = m'$  and there exist polynomials  $u_1(\bar{x}), \dots, u_{m'}(\bar{x}) \in \mathbb{F}[x_1, x_2, \dots, x_n]$  and a permutation  $\sigma : [m'] \rightarrow [m']$  such that:  $\hat{h}_i(f_k \cdot x_{n+1}, \bar{x}) = h_{\sigma(i)}(\bar{x}, x_{n+1}) \cdot u_i(\bar{x})$  and  $\hat{e}_i = e_{\sigma(i)}$  for  $i \in [m']$ .
2.  $m - m' \leq r$ . Moreover, there exists an injective map  $\tau : \{m' + 1, \dots, m\} \rightarrow [r]$  such that  $h_l$  and  $w_{\tau(l)}$  are nonzero scalar multiples of each other (i.e.  $h_l \sim w_{\tau(l)}$ ), for  $l \in \{m' + 1, \dots, m\}$ .

We defer the proof of the lemma to the end of the section.

In light of the above, the algorithm proceeds by first converting a given polynomial  $f$  into a monic polynomial  $\hat{f}$  to recover the factors that depend on  $x_{n+1}$ . Next, the algorithm recursively factors  $f_k$  (that does not depend on  $x_{n+1}$ ) to recover the factors that *do not* depend on  $x_{n+1}$  (if any).

**Input:**  $s$ -sparse polynomial  $f \in \mathbb{F}[x_1, x_2, \dots, x_{n+1}]$  with individual degrees at most  $d$ .  
**Output:** irreducible factors  $h_1, h_2, \dots, h_m$ , and  $e_1, e_2, \dots, e_m$  such that  $f = h_1^{e_1} \dots h_m^{e_m}$

1. **if**  $n \leq 1$  **then** Return the bi-variate factorization of  $f$ ;
2.  $k = \deg_{x_{n+1}}(f)$ ;  $f_k \leftarrow \text{lc}_{x_{n+1}}(f)$ ;
3. Compute  $\hat{f}(y, \bar{x})$  (Using Definition 5.4)
4. Compute the unique monic factorization  $\hat{f}(y, \bar{x}) = \prod_{i=1}^{\hat{m}} \hat{h}_i^{e_i}(y, \bar{x})$  (Using Algorithm 2)
5. **foreach**  $i \in [\hat{m}]$  **do**  $h_i(x_1, \dots, x_n, x_{n+1}) \leftarrow \hat{h}_i(f_k \cdot x_{n+1}, \bar{x})$ ;
6. Recursively compute a factorization of  $f_k(x_1, \dots, x_n) = \prod_{j=1}^r w_j^{\beta_j}(x_1, \dots, x_n)$
7. **for**  $j \leftarrow 1$  **to**  $r$  **do**  
 $\alpha_j \leftarrow -\beta_j \cdot (k - 1)$ ;  
**for**  $i \leftarrow 1$  **to**  $\hat{m}$  **do**  

Find the maximal  $d_{ij}$  such that  $w_j^{d_{ij}} \mid h_i$ ; /\* By iteratively applying Lemma 2.7 with  
 $t = \xi(n, d^2, s^d)$  \*/  
 $\alpha_j \leftarrow \alpha_j + d_{ij} \cdot e_i$ ;  $h_i \leftarrow h_i / w_j^{d_{ij}}$ ;
8. **return**  $h_1, \dots, h_{\hat{m}}, w_1, \dots, w_r$  and  $e_1, \dots, e_{\hat{m}}, \alpha_1, \dots, \alpha_r$ ; /\* Return only those where  $\alpha_j > 0$  \*/

**Algorithm 3:** Main Algorithm: overview

**Theorem 5.7.** *Let  $f(\bar{x}) \in \mathbb{F}[x_1, x_2, \dots, x_n]$  be a polynomial with individual degrees at most  $d$ . Given  $f$ , Algorithm 3 outputs pairwise coprime, irreducible polynomials  $h_1, h_2, \dots, h_m$ , and  $e_1, e_2, \dots, e_m$  such that  $f = h_1^{e_1} \dots h_m^{e_m}$ , using  $(n \cdot \xi(n, d^2, s^d))^{\mathcal{O}(d^2)} \cdot \text{poly}(\text{c}_{\mathbb{F}}(d^2))$  field operations.*

*Proof.* The correctness of the algorithm follows from Lemmas 5.5 and 5.6. In particular, by Theorem 4.1:  $\|\hat{h}_i\|, \|h_i\| \leq \xi(n, d^2, s^d)$ .

**Runtime Analysis:** Let  $T(n, s, d)$  denote the number of field operations of the algorithm given an  $s$ -sparse,  $n$ -variate polynomial of individual degrees at most  $d$ . We get that  $T(1, s, d), T(2, s, d) =$

$\text{poly}(\text{c}_{\mathbb{F}}(d))$ . For  $n \geq 3$ , we outline the runtime of each step.

1.  $T(1, s, d), T(2, s, d) = \text{poly}(\text{c}_{\mathbb{F}}(d))$ .
2.  $\text{poly}(n, s, d)$ .
3. By Lemma 5.5 -  $\text{poly}(n, s^d)$ .
4. By Lemmas 5.5 and 5.3 -  $(n \cdot \xi(n, d^2, s^d))^{\mathcal{O}(d^2)} \cdot \text{poly}(\text{c}_{\mathbb{F}}(d^2))$ .
5. By Lemma 5.5 and Theorem 4.1 -  $\text{poly}(\xi(n, d^2, s^d))$ .
6. Since  $\|f_k\| \leq \|f\| \leq s - T(n-1, s, d)$ .
7. By Lemma 2.7 -  $\text{poly}(nd, \xi(n, d^2, s^d))$ .

Consequently:  $T(n, s, d) = T(n-1, s, d) + (n \cdot \xi(n, d^2, s^d))^{\mathcal{O}(d^2)} \cdot \text{poly}(\text{c}_{\mathbb{F}}(d^2))$  implying that  $T(n, s, d) = (n \cdot \xi(n, d^2, s^d))^{\mathcal{O}(d^2)} \cdot \text{poly}(\text{c}_{\mathbb{F}}(d^2))$ .  $\square$

We now give the proof of Lemma 5.6.

*Proof of Lemma 5.6.* Part 1. Observe that:

$$\prod_{i=1}^{m'} h_i^{e_i}(\bar{x}, x_{n+1}) \cdot \prod_{l=m'+1}^m h_l^{e_l}(\bar{x}) \cdot f_k^{k-1}(\bar{x}) = f(\bar{x}, x_{n+1}) \cdot f_k^{k-1}(\bar{x}) = \hat{f}(f_k \cdot x_{n+1}, \bar{x}) = \prod_{j=1}^{\hat{m}} \hat{h}_j^{\hat{e}_j}(f_k \cdot x_{n+1}, \bar{x})$$

Let us view the above as univariate polynomials in  $x_{n+1}$  over  $(\mathbb{F}(x_1, \dots, x_n))[x_{n+1}]$ . Given this view, the term  $\prod_{l=m'+1}^m h_l^{e_l}(\bar{x}) \cdot f_k^{k-1}(\bar{x})$  is a field element in the field of rational functions:  $\mathbb{F}(x_1, \dots, x_n)$ . Therefore, by Uniqueness of Factorization (Lemma 2.3)  $\hat{m} = m'$  and there exist polynomials  $u_1(\bar{x}), \dots, u_{m'}(\bar{x}) \in \mathbb{F}(x_1, \dots, x_n)$  and a permutation  $\sigma : [m'] \rightarrow [m']$  such that:  $\hat{h}_i(f_k \cdot x_{n+1}, \bar{x}) = h_{\sigma(i)}(\bar{x}, x_{n+1}) \cdot u_i(\bar{x})$  and  $\hat{e}_i = e_{\sigma(i)}$  for  $i \in [m']$ . Finally, as  $\hat{h}_i(f_k \cdot x_{n+1}, \bar{x})$ -s are polynomials (and not rational functions) and  $h_{\sigma(i)}(\bar{x}, x_{n+1}) \cdot u_i(\bar{x})$ -s are irreducible polynomials, it follows that  $u_1(\bar{x}), \dots, u_{m'}(\bar{x}) \in \mathbb{F}[x_1, x_2, \dots, x_n]$ .

Part 2. Observe that:

$$\prod_{j=1}^r w_j^{\beta_j}(\bar{x}) = f_k(\bar{x}) = \text{lc}_{x_{n+1}}(f) = \text{lc}_{x_{n+1}} \left( \prod_{i=1}^{m'} h_i^{e_i}(\bar{x}, x_{n+1}) \right) \cdot \prod_{l=m'+1}^m h_l^{e_l}(\bar{x}).$$

and the claim follows by Uniqueness of Factorization (Lemma 2.3).  $\square$

## 6 Open Questions

We conclude by listing some open problems.

Perhaps the most immediate and natural question left open by this work is to understand whether one can obtain an improved sparsity bound on the factors of  $s$ -sparse polynomials of bounded individual degree. As we discussed in Section 4.1, the best lower bound for we know for the sparsity of factors of  $s$ -sparse polynomials of individual degree  $d$  is  $s^{\log d}$  over fields of characteristic 0 and about  $s^d$  over general fields. Thus there is a considerable gap between these lower bounds and the upper bound that we prove, and it is a very interesting question to close to gap.

Another more ambitious goal is to obtain a non trivial sparsity bound with no restriction on individual degree. As we noted in Section 4.1, such a result would not be possible for all fields, and any such proof would have to use the properties of the underlying field to obtain a better bound.

One could also study the algorithmic implications of a general sparsity bound. It seems challenging to derandomize polynomial factoring, even if we assume that factors of a given sparse polynomial are sparse (without assuming any individual degree bound). We leave this as an interesting open problem.

Given the result of [20] which shows an equivalence between the problems of polynomial identity testing and polynomial factorization, this also naturally raises the question (and indeed it was raised in [20]) of whether one can derandomize factoring for the classes of polynomials for which we know how to derandomize PIT. Sparse polynomials are a natural example of such a class, but there are several other natural classes that one could consider.

## Acknowledgments

The authors would like to thank Ramprasad Saptharishi [23] for sharing with us the example presented in Claim 4.4 and for letting us include it in this paper. In addition, the authors would like to thank the anonymous referees for useful comments that improved the presentation of the results.

## References

- [1] S. Barman. Approximating nash equilibria and dense bipartite subgraphs via an approximate version of carathéodory’s theorem. In *Proceedings of the forty-seventh Annual ACM Symposium on Theory of Computing (STOC)*, pages 361–369, 2015. 16
- [2] E. Berlekamp. Factoring polynomials over large finite fields. *Mathematics of Computation*, 24(111):713–335, 1970. 12
- [3] 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. 1
- [4] R. M. de Oliveira. Factors of low individual degree polynomials. In *Proceedings of the 30th Conference on Computational Complexity (CCC)*, pages 198–216, 2015. 2, 4

- [5] Z. Dvir and R. M. de Oliveira. Factors of sparse polynomials are sparse. *CoRR*, abs/1404.4834, 2014. [4](#), [5](#), [13](#), [14](#), [15](#)
- [6] M. A. Forbes and A. Shpilka. Complexity theory column 88: Challenges in polynomial factorization. *SIGACT News*, 46(4):32–49, 2015. [2](#)
- [7] S. Gao, E. Kaltofen, and A. G. B. Lauder. Deterministic distinct-degree factorization of polynomials over finite fields. *J. Symb. Comput.*, 38(6):1461–1470, 2004. [12](#)
- [8] J. v. z. Gathen. Who was who in polynomial factorization:. In *ISSAC*, page 2, 2006. [1](#)
- [9] J. v. z. Gathen and J. Gerhard. *Modern computer algebra*. Cambridge University Press, 1999. [1](#), [12](#)
- [10] J. v. z. Gathen and E. Kaltofen. Factoring sparse multivariate polynomials. *Journal of Computer and System Sciences*, 31(2):265–287, 1985. [1](#), [2](#), [4](#), [7](#), [19](#)
- [11] K. O. Geddes, S. R. Czapor, and G. Labahn. *Algorithms for computer algebra*. Kluwer, 1992. [12](#)
- [12] 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. [1](#)
- [13] V. Kabanets and R. Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Computational Complexity*, 13(1-2):1–46, 2004. [1](#)
- [14] E. Kaltofen. Single-factor hensel lifting and its application to the straight-line complexity of certain polynomials. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing (STOC)*, pages 443–452, 1987. [1](#)
- [15] 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., Greenwich, Connecticut, 1989. [1](#), [7](#), [13](#)
- [16] E. Kaltofen. Polynomial factorization: a success story. In *ISSAC*, pages 3–4, 2003. [1](#)
- [17] 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. [1](#), [7](#), [8](#), [20](#)
- [18] N. Kayal. *Derandomizing some number-theoretic and algebraic algorithms*. PhD thesis, Indian Institute of Technology, Kanpur, India, 2007. [1](#)
- [19] 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. [8](#), [11](#)
- [20] S. Kopparty, S. Saraf, and A. Shpilka. Equivalence of polynomial identity testing and deterministic multivariate polynomial factorization. In *Proceedings of the 29th Annual IEEE Conference on Computational Complexity (CCC)*, pages 169–180, 2014. [2](#), [4](#), [27](#)

- [21] A. Lenstra, H. Lenstr, and L. Lovász. Factoring polynomials with rational coefficients. *Mathematische Annalen*, 261(4):515–534, 1982. 12
- [22] A. Ostrowski. "U on the meaning of the theory of convex polyhedra for the formal algebra. *Annual Reports German Math. Association*, 20:98–99, 1921. 5, 15
- [23] R. Saptharishi. Private communication, 2018. 17, 18, 27
- [24] S. Saraf and I. Volkovich. Blackbox identity testing for depth-4 multilinear circuits. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 421–430, 2011. Full version at <https://eccc.weizmann.ac.il/report/2011/046>. 11
- [25] A. Schinzel. *Polynomials with special regard to reducibility*, volume 77. Cambridge University Press, 2000. 14
- [26] V. Shoup. A fast deterministic algorithm for factoring polynomials over finite fields of small characteristic. In *ISSAC*, pages 14–21, 1991. 12
- [27] A. Shpilka and I. Volkovich. Improved polynomial identity testing for read-once formulas. In *APPROX-RANDOM*, pages 700–713, 2009. Full version at <https://eccc.weizmann.ac.il/report/2010/011>. 11
- [28] 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>. 2, 3, 4
- [29] M. Sudan. Decoding of reed solomon codes beyond the error-correction bound. *Journal of Complexity*, 13(1):180–193, 1997. 1
- [30] M. Sudan. Algebra and computation. <http://people.csail.mit.edu/madhu/FT98/course.html>, 1998. Lecture notes. 13
- [31] I. Volkovich. Deterministically factoring sparse polynomials into multilinear factors and sums of univariate polynomials. In *APPROX-RANDOM*, pages 943–958, 2015. 4
- [32] I. Volkovich. On some computations on sparse polynomials. In *APPROX-RANDOM*, pages 48:1–4:21, 2017. 2, 3, 4
- [33] G. M. Ziegler. *Lectures on polytopes*, volume 152. Springer Science & Business Media, 2012. 13