Symmetry in polytopes: Deterministic poly-time factorization of certain sparse polynomials with bounded individual degree

Pranav Bisht * Nitin Saxena †

Abstract

Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ be a polynomial with s monomials and having individual degrees of its variables bounded by d. Bhargava, Saraf and Volkovich (FOCS'18;JACM'20) designed a deterministic algorithm to factor f in $s^{\mathrm{poly}(d)\log n}$ time, which has the best time complexity for this class of polynomials. In this work, we show that if f is a symmetric polynomial, then it can be deterministically factored in time $(sn)^{\mathrm{poly}(d)}$. For constants $d > 2$, this is the *first* factoring algorithm in deterministic polynomial-time. Our technique is field independent, i.e. we efficiently reduce to the univariate factoring algorithms in that field.

The main component for our factoring algorithm is a new sparsity bound of $s^{O(d^2 \log d)}$, which we prove for any factor of the symmetric polynomial f. Our sparsity bound uses techniques from convex geometry and exploits symmetry (only) in the Newton polytope of f. Symmetric support often gives 'dense' factors, for eg.: In finite fields (resp. characteristic zero), there is a famous example of an s-sparse polynomial having s^d-dense (resp. $s^{\log d}$-dense) factors. These extremal examples are symmetric, so, our method characterizes them tightly.

We prove a crucial structural result about convex polytopes (that have 'low entropy'), which is behind all the results in this work. It is of independent interest in the area of convex geometry.

Keywords: Sparse polynomials, symmetric polynomials, factor-sparsity, multivariate polynomial factorization, derandomization, convex polytopes, hitting-set.

Contents

*Department of CSE, IIT Kanpur, India, pbisht@cse.iitk.ac.in

†Department of CSE, IIT Kanpur, India, nitin@cse.iitk.ac.in. Both authors contributed equally to this manuscript.

1 Introduction

Polynomial factorization is one of the most fundamental and central problems studied in computational algebra. In this paper, we concern ourselves with the problem of *multi*variate polynomial factorization which, given a polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ over some field $\mathbb{F}$ as an input, asks to compute all the irreducible factors of f. The problem admits an efficient randomized algorithm [Kal89, KT90] in the blackbox circuit model. Finding an efficient *deterministic* algorithm continues to be a challenging open problem. See [Kal03, vzG06, vzGG13, Sud98] for a detailed exposition. Polynomial factorization also has interesting connections with other problems such as circuit lower bounds [KI04], list decoding [Sud97, GS98] and cryptography [CR88].

Another fundamental problem is that of *Polynomial identity testing* (PIT). Given an input polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ as an algebraic circuit (See Appendix A for definition of algebraic circuits), it asks to determine if f is identically zero. This problem also admits a simple and efficient randomized algorithm due to PIT Lemma [Sch80, Zip79, DL77, Ore22] and like factoring, finding a deterministic efficient algorithm is still open. In fact, [KSS14] showed that the problem of derandomizing multivariate polynomial factoring is equivalent to derandomizing PIT, for general algebraic circuits. Showing this equivalence for other interesting circuit classes, was left as an open problem by [KSS14]. For a particular class, before showing this equivalence, one also needs to solve the problem of *Factor Closure*. Given an input polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ in a particular representation, it asks for an upper bound on the size of its factors in the same representation.

A very natural interesting class of algebraic circuits, to be considered for the factor closure problem, is that of *sparse* polynomials. Sparse polynomials are also known as *lacunary* polynomials, or *fewnomials*, in the diverse maths literature. Sparsity of a polynomial f is defined as number of monomials in f with non-zero coefficients. Sparse polynomials can also be seen as *depth*-2 ($\Sigma\Pi$) algebraic circuits. *Individual degree* of a polynomial f is defined as the maximum degree of some variable appearing in f (See Section 2). [vzGK85] first raised the following important question:

Question 1.1. Let f be an s-sparse polynomial with *constant* individual degree. Then, what is the best upper bound on sparsity for an arbitrary factor of f?

There are known examples where a significant blowup in the sparsity of factors occur. For fields of characteristic zero, there is example of an s-sparse polynomial having a factor with sparsity $s^{\log d}$, where d is the individual degree (Example 4.9). For finite fields, there is an example where the sparsity of a factor is s^d (Example 4.10). Hence, the answer of Question 1.1 has to be at least s^d, but how much larger could the exponent be? Addressing Question 1.1, [BSV20] proved a sparsity bound of $s^{O(\log n)}$, significantly better than the trivial d^n bound. Although [Vol17] has conjectured that the true bound is $\text{poly}(s)$, a better sparsity bound than that of [BSV20] is yet to be found.

We make definite progress in this direction and establish the conjectured bound for the class of symmetric polynomials. Symmetric polynomials are defined as polynomials such that if any of the variables are interchanged, we still obtain the same polynomial (See Section 2 for formal definition). *We prove* $\text{poly}(s)$ *upper bound on the sparsity of factors of an* s*-sparse symmetric polynomial with constant individual degree, thus getting rid of the* $O(\log n)$ *term present in the exponent of sparsity bound by [BSV20].* Symmetric polynomials is a natural class to consider as they are studied extensively in both computer science and mathematics. It is intriguing to explore the effect of symmetry in the polynomial factorization problem. Many multivariate polynomials $f \in \mathbb{F}[x_1, \ldots, x_n]$ are constructed by 'boosting' a univariate polynomial $a(X)$ by product or addition as shown below:

$$f = \prod_{i=1}^{n} a(x_i) \quad \text{or} \quad f = \left(\sum_{i=1}^{n} a(x_i) \right)^d \quad \text{for some } d \geq 1.$$

Some of the famous polynomials that exhibit highest known factor-sparsity blowups (Example 4.9 and Example 4.10) are of this type. Such polynomials are actually symmetric by construction and are exactly captured by the results of this work. Therefore, the symmetric setting is important and non-trivial. Our bound works over all fields and is fairly tight considering that s^d is a lower bound on factor-sparsity as evident in Example 4.10.

For a general (possibly non-symmetric) input polynomial f, we show that any symmetric factor g of f (if it exists) also has a $\text{poly}(s)$ sparsity upper bound. Moreover, our proof techniques do *not* require symmetry in the coefficients of g, but merely in the support of g. Besides symmetric polynomials, our technique also proves polynomial sparsity bounds for factors of a more general class of polynomials, which we call *low-entropy* polynomials and will be defined later.

As a result of our new sparsity bounds, we also get poly-time deterministic factorization algorithms for these classes of symmetric and low-entropy polynomials.

1.1 Our results: Leveraging symmetry & entropy

Theorem 1 (Symmetric sparsity bound)**.** *Let $\mathbb{F}$ be an arbitrary field and let $f \in \mathbb{F}[x_1, \ldots, x_n]$ be an s-sparse, symmetric polynomial with individual degree at most d. Then, every factor of f has its sparsity bounded by $s^{O(d^2 \log d)}$.*

Remark. (1) For constant individual degree d, we get $\text{poly}(s)$ sparsity bound for the factors of a symmetric polynomial.
(2) Previous best sparsity upper bound was $s^{O(d^2 \log n)}$ due to [BSV20], which is super polynomial even for constant d.
(3) We can also work with a general (possibly *non*-symmetric) f and get the same sparsity bound for any *symmetric* factor of f (if it exists). We prove this formally in Theorem 4. This yields a surprising *incompressibility result* in Corollary 3: a symmetric bounded-individual-degree polynomial g, that is s-dense, has $s^{\Omega(1)}$-dense multiples gh, for an arbitrary nonzero polynomial h !

As a direct consequence of the factor-sparsity bound in Theorem 1, we get a deterministic factorization algorithm for s-sparse symmetric polynomials in Corollary 1 below. Given a polynomial f, the *complete factorization* of f is a representation of f as $f_1^{e_1} f_2^{e_2} \cdots f_k^{e_k}$, where $f_1, \ldots, f_k$ are coprime irreducible polynomials and $e_1, \ldots, e_k$ are positive integers. This representation is unique up to a reordering of f_i's. We use $c_{\mathbb{F}}(d)$ below to denote the best known time complexity for factoring a univariate polynomial of degree d over $\mathbb{F}$.

Corollary 1 (Symmetric factoring algorithm)**.** *Let $\mathbb{F}$ be an arbitrary field and let $f \in \mathbb{F}[x_1, \ldots, x_n]$ be an s-sparse, symmetric polynomial with individual degree at most d. Then, there is a deterministic algorithm that computes the complete factorization of f in at most $\text{poly}(s^{d^7 \log d} \cdot n^{d^2} \cdot c_{\mathbb{F}}(d^2))$ field operations.*

Remark. (1) Observe that for constant d, the time complexity is only $\text{poly}(s, n, c_{\mathbb{F}}(d))$.
(2) For a finite field $\mathbb{F} = \mathbb{F}_{p^l}$, $c_{\mathbb{F}}(d) \leq \text{poly}(l \cdot p, d)$. For $\mathbb{F} = \mathbb{Q}$, $c_{\mathbb{F}}(d) \leq \text{poly}(d, t)$, where t is maximum bit-complexity of the coefficients of f.
(3) Throughout this paper (specifically in Theorem 1, Corollary 1 and Theorem 4), we get the same results if we replace symmetric polynomials with a much more general class of polynomials, which we call *symmetric-support* polynomials. Let $\text{supp}(f)$ denote the set of monomials in f with non-zero coefficients. We call $f \in \mathbb{F}[x_1, \ldots, x_n]$ a symmetric-support polynomial if for each monomial $x_1^{e_1} \cdots x_n^{e_n} \in \text{supp}(f)$, we also have that $x_1^{e_{\sigma(1)}} \cdots x_n^{e_{\sigma(n)}} \in \text{supp}(f)$ for every permutation $\sigma \in S_n$. For eg., $f = x_1^2 x_2 x_3 + 2x_1 x_2^2 x_3 - x_1 x_2 x_3^2$ is a symmetric-support polynomial that is not symmetric. (While $f = x_1^2 x_2 x_3 + x_1 x_2^2 x_3$ is not symmetric-support.)

We say that an n-dimensional vector $\mathbf{v}$ is of *δ-entropy* if it has $(n - \delta)$ equi-valued coordinates. We call a set of vectors A, a *δ-entropy set*, if for every vector $\mathbf{v} \in A$, $\mathbf{v}$ is of $(\leq \delta)$-entropy. We

can identify the support of polynomial f, with the subset $\text{supp}(f) \subseteq \{0,1,\ldots,d\}^n$, as the set of exponent vectors corresponding to each monomial in f. We call f a *δ-entropy polynomial*, if $\text{supp}(f)$ is of δ-entropy. For eg., $f = x_1x_2x_3^3x_4^5 + 2x_1^2x_2^4x_3^6x_4^6 + 3x_1^7x_2^9x_3^8x_4^9$ is a 2-entropy polynomial (also, (≥ 3)-entropy); it does *not* have symmetric-support; moreover, it is *not* a 1-entropy polynomial.

Theorem 2 (Factors of low entropy polynomials)**.** *Let $\mathbb{F}$ be an arbitrary field and let $f \in \mathbb{F}[x_1,\ldots,x_n]$ be a δ-entropy polynomial of individual degree at most d. Then, every factor of f has its sparsity bounded by $(nd)^{O(d\delta)}$.*

Remark. (1) For constant $d\delta$, we get $\text{poly}(s)$-sparsity bound for the factors. Further, our bound beats the $s^{O(d^2 \log n)}$ bound [BSV20], as long as f has entropy δ as low as $o(d \log n)$.
(2) We do *not* claim that the factors are 'low'-entropy as well. For eg., $f = (x_1 \cdots x_{n/2})(x_{n/2+1} \cdots x_n)$ has both the factors with high entropy $n/2$, while f is itself a 0-entropy polynomial.

It may not be immediately obvious, but Theorem 1 is an application of Theorem 2; this will be clear in the proofs. We also get a factoring algorithm below as a corollary of Theorem 2. For a polynomial f with constant individual degree d and constant entropy δ, it can deterministically factor f in $\text{poly}(n, c_{\mathbb{F}}(d))$ field operations.

Corollary 2 (Low entropy factoring algorithm)**.** *Let $\mathbb{F}$ be an arbitrary field and let $f \in \mathbb{F}[x_1,\ldots,x_n]$ be a δ-entropy polynomial of individual degree at most d. Then, there is a deterministic algorithm that computes the complete factorization of f in at most $\text{poly}((nd)^{d^4\delta} \cdot c_{\mathbb{F}}(d^2))$ field operations.*

As another consequence of our improved factor-sparsity bound in Theorem 1, we get poly-time blackbox PIT for a special case of depth-4 circuits in Corollary 4. We also prove a new structural result on polytopes in Theorem 3 which is at the core of all the results stated above. It states: *if we have a δ-entropy set of exponent vectors, then the integral-points in the convex hull, have entropy at most $O(d\delta)$.* Trivially, such a thing is false for $\mathbb{Z}$-linear-span (resp. $\mathbb{Q}$-linear-span) of vertices. Yet, surprisingly, a *convex*-span preserves the low-entropy of its vertices!

1.2 Related works

The problem of sparse polynomial factorization was first studied in [vzGK85], where a randomized algorithm for the same was provided. The time complexity of this algorithm was polynomial in the sparsity of factors and thus [vzGK85] raised the question of proving better sparsity bounds for factors of sparse polynomials. [SV10] gave an efficient deterministic algorithm for factorization of sparse multilinear polynomials ($d = 1$). [Vol15] generalized this result to sparse polynomials that factorize into multilinear factors. [Vol17] solved the model of sparse multiquadratic polynomials ($d = 2$). Another motivation for studying polynomial factorization of special classes is that improving factoring methods often lead to improvements in blackbox PIT. For eg. [LST21] used factoring of special circuits to give the first subexponential PIT for constant-depth circuits.

The (retracted) paper [DdO14] provided a new interesting approach of connecting sparsity bounds of f with Newton polytopes of the polynomial and its factors. [BSV20] were able to successfully utilize this approach and proved a sparsity bound of $s^{O(d^2 \log n)}$ for factors of s-sparse, individual degree $\leq d$ polynomials. For symmetric polynomials in Theorem 1, we improve this bound considerably as all these works are in the setting of bounded individual degree ($d = O(1)$). [BSV20] also gave a deterministic factorization algorithm for sparse polynomials with constant individual degree which runs in $s^{O(\log n)}$ time. With our new $\text{poly}(s)$ sparsity bound, we use their algorithm to get a deterministic $\text{poly}(s)$-time factorization algorithm for symmetric, s-sparse polynomials, in this regime ($d = O(1)$).

A related, and somewhat subsumed, problem in polynomial factorization is that of factor closure. Given an input f in a particular representation, what is the size of factors in the same representation? The foundational work of [Kal86, Kal87, Kal89] showed that if f is a size-s algebraic *circuit*, then its factors also have $\text{poly}(s)$-sized algebraic circuits, i.e. VP is closed under factoring. [DSS18] studied the factor closure for the classes of VF, VBP, VNP with a quasi-polynomial blowup in size, and gave analogous whitebox algorithms (see Appendix A for definitions of these classes). [CKS19] were able to show that VNP is properly closed under factoring (with only poly blowup in size). Recently, [ST20] showed that size-s ABPs have factors of size $\text{poly}(s)$, thus proving factor closure for VBP class.

For algebraic formulas, [Oli16] showed that if f is computed by a depth-Δ, size-s algebraic formula, then its factors can be computed by depth-$(\Delta + 5)$ and size $\text{poly}(s)$ formulas, provided that individual degree of f is constant. Observe that sparse polynomials can also be seen as depth-2 algebraic formulas, thus [Oli16] showed that factors of bounded individual degree sparse polynomials can be computed by depth-7 formulas. [BSV20] showed that these factors can be computed in depth-2 itself with a quasi-polynomial blowup in size. We show that if f is also symmetric, then the factors can be computed in depth-2, with only a polynomial blowup in size.

Besides being studied extensively in classical mathematics, symmetric polynomials have been investigated in the area of algebraic complexity theory as well, mostly in the context of lower bounds. [Shp02] defines depth-2 symmetric circuits and studies complexity of determinant for this model. [HY11] study complexity of elementary symmetric polynomials in the algebraic formula model and [CKL+20] study a class of symmetric polynomials called Schur polynomials, also in the formula model. [FLMS15] prove strong lower bound for elementary symmetric polynomials in the model of depth-4 algebraic circuits with bounded bottom fan-in. [BJ18] study the complexity of symmetric polynomials with respect to its expression in terms of elementary symmetric polynomials. [DW20] define a new class of algebraic circuits, which they call symmetric algebraic circuits and show separation of determinant from permanent in this model. In this work, we study symmetric polynomials in the context of sparse polynomial factorization.

1.3 The sparsity connection with Newton polytopes

Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ be a polynomial of individual degree d such that $f =: \sum c_{e_1,e_2,\ldots,e_n} \cdot x_1^{e_1} x_2^{e_2} \cdots x_n^{e_n}$. We define support of f as the set of exponent vectors: $\text{supp}(f) := \{(e_1, \ldots, e_n) \mid c_{e_1,\ldots,e_n} \neq 0\}$. Let us denote sparsity of f as $\|f\|$, which is the same as $|\text{supp}(f)|$. We define the *Newton Polytope* of f, denoted by P_f, as the convex hull (or convex-span) of all the points in $\text{supp}(f)$.

For two polytopes A and B, their *Minkowski sum* $A + B$ is defined as the set of points $\{a + b \mid a \in A, b \in B\}$. Minkowski sum is well studied in polytope literature and comes with some nice properties. The Minkowski sum $A + B$ is itself a convex polytope. Let $V(P)$ denote the set of *vertices* (corner points) of a polytope P, then one can show a certain 'incompressibility' property: $|V(A + B)| \geq \max\{|V(A)|, |V(B)|\}$. See the exposition in [BSV20, Prop. 3.2] or [DdO14, Cor. 3.13] for a proof of this.

A classical fact about Newton polytopes, first observed by [Ost21] a century ago, states that if $f = g \cdot h$, then $P_f = P_g + P_h$, where $P_g + P_h$ is the Minkowski sum of Newton polytopes of g and h. Moreover, we know that $\|f\| \geq |V(P_f)|$; since $\text{supp}(f)$ is in the convex hull of $V(P_f)$. Therefore, we are able to connect sparsity of f with a factor g as follows:

$$\|f\| \geq |V(P_f)| \geq |V(P_g)|. \tag{1.2}$$

Another way to think about sparsity bound problem for $f = g \cdot h$ is to determine how many monomials of g survive on multiplication with h. Equation (1.2) tells us that at least $|V(P_g)|$ many monomials survive. These vertices of P_g are in a sense *extremal* monomials in g that never get canceled on multiplication (with any h).

1.4 Sparsity bound using symmetric polytopes

Let f be an s-sparse polynomial and suppose g is a symmetric factor of f with individual degree d. In Theorem 4, we show that $\|g\| \leq s^{O(d^2 \log d)}$. The proof technique exploits symmetry in Newton polytope of g and is described below. Our proof focuses only on the *integral-points* in the polytopes, so we define the object $P'_g := P_g \cap \mathbb{Z}^n$ (similarly P'_f). Since $V(P_g) \subseteq \text{supp}(g)$, the polytope vertices lie in $\{0, \ldots, d\}^n$. Thus, it easily follows due to the convex-span that $P'_g \subseteq \{0, \ldots, d\}^n$.

We say that a polytope P is symmetric if for each point $\mathbf{v} = (v_1, \ldots, v_n)$ in P, every permutation of $\mathbf{v}$ is also in P (See Section 2 for formal definition). Consider Newton polytope P_g of g. Since g is symmetric, we note that P_g is a symmetric polytope. In that case, we observe that if a monomial (or its exponent vector) is a vertex of P_g, then all its distinct permutations must also be vertices of P_g (See Lemma 4.1). In other words, the vertex set $V(P_g)$ is also symmetric. This property helps us show that the gap between $\|g\|$ and $|V(P_g)|$ is low, which we explain now.

We first note that $|V(P_g)| \leq \|f\| = s$ from (1.2). Now, recall the notion of entropy, defined before Theorem 2 (or refer to Section 2). Let entropy of $V(P_g)$ be δ, for some $\delta \geq 0$. Since $V(P_g)$ is symmetric and its size is upper bounded by s, every vector in it must contain a coordinate

of 'very high' frequency, otherwise $V(P_g)$ will contain a lot of distinct permutations and its size will blow up (i.e. *symmetry* $\Rightarrow$ *low entropy*). Formally, $V(P_g)$ must be of low entropy. With some careful counting arguments in Lemma 4.4 and Lemma 4.5, we indeed show that $\delta \leq O(d \log s)$ (Remember that we treat d as a constant in the bounded individual degree setting).

At this point, our Structure Theorem 3 comes into play. It acts as a bridge that connects entropy of $V(P_g)$ to entropy of P'_g, showing that there is only an $O(d)$ blowup in entropy on moving from a set to its convex-span (integral-points). Thus, entropy of P'_g is at most $O(d^2 \log s)$. Using an intricate counting argument in Lemma 4.3, we deduce that for P'_g with this low entropy, sparsity of g is nicely bounded. In particular, for P'_g with $O(d^2 \log s)$-entropy and $|V(P_g)| \leq s$, we get $\|g\| \leq s^{O(d^2 \log d)}$. Careful inspection of the argument above shows that we in fact prove a slightly stronger statement, that $|V(P_g)| \geq \|g\|^{\frac{1}{O(d^2 \log d)}}$ (See Corollary 3). In other words, using (1.2), *any* (nonzero) multiple of g has this sparsity *lower* bound.

1.5 Limitations of previous techniques & our new structure

The fascinating approach of connecting sparsity with Newton polytopes was first presented in [DdO14], which was summarized in Equation (1.2) earlier. The more important part of actually getting a non-trivial factor-sparsity bound using this polytope approach was done in [BSV20]. For $f = gh$, [BSV20] proved a lower bound on $|V(P_g)|$ in terms of $\|g\|$ by using an approximate version of Carathéodory's theorem [BSV20, Theorem 3.6] and showed that $|V(P_g)| \geq \|g\|^{\frac{1}{O(d^2 \log n)}}$. Thus using (1.2), they get $\|g\| \leq \|f\|^{O(d^2 \log n)}$. Although the use of approximate Carathéodory's theorem gives the first non-trivial factor-sparsity bound, it also brings in an $O(\log n)$ term in the exponent of their bound. In this work, we make use of the sparsity connection in [DdO14] but replace this second component of [BSV20] with our new techniques of designing a set of clever *hyperplane equations* in Structure Theorem 3, and exploiting the symmetry of polytopes in Theorem 4. These new techniques help us get rid of the unwanted $\log n$ term in exponent of the factor-sparsity bound by [BSV20]. If g is symmetric, we show a much better lower bound of $|V(P_g)| \geq \|g\|^{\frac{1}{O(d^2 \log d)}}$. This gives us $\|g\| \leq \|f\|^{O(d^2 \log d)}$ using (1.2).

We basically construct a set of hyperplanes which 'enclose' the vertices $V(P_g)$ in Claim 3.2, and hence the integral-points P'_g of the Newton polytope P_g, of a symmetric factor g. These hyperplanes are described by a set of vectors consisting of all distinct S_n-permutations of an 'almost-balanced' vector in $\{-1, +1\}^n$, defined in Equation (3.1). These vectors are specifically designed to 'witness' the low-entropy of any internal integral-point in P_g (Claim 3.4). Thus, by ensuring low-entropy of all points in P'_g, these hyperplanes are able to nicely bound the number of points in $\text{supp}(g) \subseteq P'_g$, as shown in Lemma 4.3.

The underlying proof template of [BSV20] and Theorem 4 in our work is similar, in the sense that both show a lower bound on $|V(P_g)|$. Unfortunately [BSV20, Remark 4.3 and Claim 4.4] show that this particular proof strategy cannot get a sparsity upper bound better than $s^{O(\log n)}$,

for general factors of a general sparse polynomial. We note that the polytope example in [BSV20, Claim 4.4] is not symmetric and is therefore not a hurdle for Theorem 4 in our work. Moreover, the proofs (of Theorem 1 and Theorem 2) in our work are schematically different. Instead of showing lower bound on $|V(P_g)|$, we utilize structural properties like symmetry, or low-entropy, of input f to get sparsity upper bound for its factors; even though the factors themselves might not be symmetric or of low entropy. We now discuss this new proof approach of ours.

1.6 Proof ideas

Proof idea for Theorem 2. In Theorem 2, we are given a low-entropy input polynomial f, say of entropy δ. If f factorizes as $f = gh$, then using the Minkowski sum property of Newton polytopes, we know that $P_f = P_g + P_h$. Note that the vertices $V(P_f) \subseteq \text{supp}(f) \subseteq P'_f$, and all we know is: $\text{supp}(f)$ has entropy $\leq \delta$. Fix a point $\mathbf{v} \in \text{supp}(h)$. Observe that $V(P_g) + \mathbf{v}$ are vertices of the polytope $P_g + \mathbf{v}$, which itself is a *sub*-polytope of P_f. By convexity, the integral-points in all these polytopes are bound to be in $\{0, \ldots, d\}^n$. Now, we use our Structure Theorem 3 cleverly to deduce that there is only an $O(d)$ blowup in entropy, when we go from the vertices $V(P_f)$ to this special sub-polytope's integral-part, namely $(P'_g + \mathbf{v})$. Now, simple counting (as in Lemma 4.3) helps us upper bound the total number of points in P'_g to be low, namely $(nd)^{O(d\delta)}$ at most. As, $\text{supp}(g) \subseteq P'_g$, we conclude that $\|g\| \leq (nd)^{O(d\delta)}$.

We remark that in Theorem 2, we are showing sparsity upper bound for factors of a low entropy f even though the factors themselves might be of *high* entropy. If factors were of low entropy, then we trivially get sparsity bound for them. But that is not always the case.

Proof idea for Theorem 1. We first point out that a symmetric polynomial f may not have symmetric factors, otherwise Theorem 1 would have followed directly from Theorem 4. For example consider symmetric $f = x^3 + x^2y^2 + xy + y^3$ which factorizes as $(x^2 + y)(x + y^2)$. Each factor considered individually, is not symmetric. We instead use a combination of ideas described above in Theorem 4 and Theorem 2, in order to prove Theorem 1.

We are given an input polynomial f which is symmetric and s-sparse. Now consider its Newton polytope's integral-points P'_f; say, its vertex set $V(P_f)$ has entropy δ. Since $V(P_f)$ is a symmetric set by Lemma 4.1 and $|V(P_f)| \leq s$, we deduce that its entropy must be low, i.e. $\delta = O(d \log s)$ using Lemma 4.4 and the proof analysis of Lemma 4.5. Now, we again use our Structure Theorem 3 to observe that P'_f is of entropy $O(d\delta) = O(d^2 \log s)$. In other words, our input symmetric polynomial f is of sufficiently low entropy as $\text{supp}(f) \subseteq P'_f$. Together with this we utilize $|V(P_f)| \leq s$, in an intricate calculation, in the proof of Theorem 2, to derive the desired factor-sparsity upper bound of $s^{O(d^2 \log d)}$.

Proof idea for Corollary 1 and Corollary 2. In the bounded individual degree regime, [BSV20] showed that one can get a deterministic factoring algorithm for sparse polynomials, which has

time complexity same as the bound (up to a polynomial blowup) that one can prove for factors of sparse polynomials. Thus, with our new polynomial sparsity bounds proved in Theorem 1 and Theorem 2 in the constant individual degree regime, we get poly-time factoring algorithms for symmetric resp. low-entropy sparse polynomials in Corollary 1 resp. Corollary 2.

2 Preliminaries

Notations: We use shorthand $[n]$ for the set $\{1, 2, \ldots, n\}$. We denote a vector $v = (v_1, \ldots, v_n)$ in short by $\mathbf{v}$ (as a column vector). We will use the terms vector or *point* interchangeably. We will sometimes use $\mathbb{F}[\mathbf{x}]$ as short for $\mathbb{F}[x_1, \ldots, x_n]$. The finite *symmetric group* on n elements, which contains all the permutations of n elements, is denoted by S_n. For a vector $\mathbf{v}$ and a permutation $\sigma \in S_n$, we denote σ-permutation of $\mathbf{v}$ by $\sigma \circ \mathbf{v} := (v_{\sigma(1)}, \ldots, v_{\sigma(n)})$. We call a set of points symmetric, if for each point $\mathbf{v}$ in it, $\sigma \circ \mathbf{v}$ is also in the set, for every permutation $\sigma \in S_n$. We denote the n-fold Cartesian product of a set H by H^n. We will use $\log x$ for $\log_2 x$ and $\ln x$ for $\log_e x$.

Let $f \in \mathbb{F}[\mathbf{x}]$ be an n-variate polynomial. Individual degree of a variable x_i, denoted by $\deg_{x_i}(f)$ is defined as the maximum degree of that variable in f, while *individual degree* of a polynomial is the maximum among all the individual degrees, $\max_{i\in[n]} \deg_{x_i}(f)$. We will use $\mathbf{x}^{\mathbf{e}}$ to denote the monomial $x_1^{e_1} x_2^{e_2} \cdots x_n^{e_n}$. We define $\mathrm{coeff}(\mathbf{x}^{\mathbf{e}})(f)$ as the coefficient of monomial $\mathbf{x}^{\mathbf{e}}$ in polynomial f. We define *support* of f as $\mathrm{supp}(f) = \{\mathbf{e} \mid \mathrm{coeff}(\mathbf{x}^{\mathbf{e}})(f) \neq 0\}$. Let us denote *sparsity* of f as $\|f\|$, which is the same as $|\mathrm{supp}(f)|$.

Polytopes: For a finite set of points $\mathbf{v}_1, \ldots, \mathbf{v}_k \in \mathbb{R}^n$, their *convex combination* is defined as an $\mathbb{R}$-linear combination of the points: $\alpha_1\mathbf{v}_1 + \ldots + \alpha_k\mathbf{v}_k$, such that $\alpha_i \geq 0$ for each $i \in [k]$ and $\sum_{i=1}^k \alpha_i = 1$. We define *convex-span* (or convex hull) $CS(\mathbf{v}_1, \ldots, \mathbf{v}_k)$ as the set of all possible convex combinations of $\mathbf{v}_i$, $i \in [k]$. A set $P \subseteq \mathbb{R}^n$ is called a (bounded) *polytope* if there is a finite set of points $\mathbf{v}_1, \ldots, \mathbf{v}_k$ such that $P = CS(\mathbf{v}_1, \ldots, \mathbf{v}_k)$. A point $\mathbf{a} \in P$ is called a *vertex* of P if it cannot be written as $\mathbf{a} = \alpha\mathbf{u} + (1-\alpha)\mathbf{v}$ for any $\mathbf{u}, \mathbf{v} \in P \setminus \{\mathbf{a}\}$ and $\alpha \in [0, 1]$. It is equivalent to saying that vertices are *corner* points of a polytope P which cannot be expressed as convex combination of any other set of points in P. We use $V(P)$ to denote the set of vertices of P. It is easy to verify that for a polytope P, $P = CS(V(P))$. Moreover, if $P = CS(\mathbf{v}_1, \ldots, \mathbf{v}_k)$ then $V(P) \subseteq \{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$.

Minkowski sum of two polytopes $A, B \in \mathbb{R}^n$ is defined as the following set of points $A + B = \{\mathbf{a} + \mathbf{b} \mid \mathbf{a} \in A, \mathbf{b} \in B\}$. A basic fact is that Minkowski sum of two polytopes is itself a polytope. The vertices of Minkowski sum have some very useful properties. It is known that every vertex of $A + B$ can be expressed *uniquely* as a sum $\mathbf{u} + \mathbf{v}$, where $\mathbf{u}$ is a vertex of A and $\mathbf{v}$ is a vertex of B. Additionally, one can show that $|V(A + B)| \geq \max\{|V(A)|, |V(B)|\}$ (See [BSV20, Prop. 3.2] or [DdO14, Cor. 3.13]). We refer the readers to [Zie12, Sch00] for a detailed discussion on polytopes.

For a polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$, the *Newton polytope* of f is defined as $P_f := CS(\mathrm{supp}(f))$. In this paper, we crucially exploit its *integral-points* $P'_f := P_f \cap \mathbb{Z}^n$. We denote the vertex set $V(P_f)$

by V_f. We also note that $V_f \subseteq \text{supp}(f) \subseteq \{0,\ldots,d\}^n$ (integral-points). The following two facts form the backbone for the Newton polytope approach of bounding factor-sparsity.

Proposition 2.1. *[Ost21] Let $f, g, h \in \mathbb{F}[x_1,\ldots,x_n]$ be polynomials such that $f = g \cdot h$. Then*

$$P_f = P_g + P_h.$$

Proposition 2.2. *[BSV20, DdO14] Let $f, g, h \in \mathbb{F}[x_1,\ldots,x_n]$ be polynomials such that $f = g \cdot h$. Then*

$$\|f\| \geq |V_f| \geq \max\{|V_g|, |V_h|\}.$$

Hyperplanes and Halfspaces: A hyperplane is the generalization of a line in higher dimensions. A hyperplane H is defined as $H := \{\mathbf{y} \in \mathbb{R}^n \mid \mathbf{a}^\intercal \mathbf{y} = b\}$, for some vector $\mathbf{a} \in \mathbb{R}^n \setminus \{\mathbf{0}\}$ and a number $b \in \mathbb{R}$. The hyperplane divides the space into two parts $\mathbf{a}^\intercal \mathbf{y} \geq b$ and $\mathbf{a}^\intercal \mathbf{y} \leq b$. These are called halfspaces.

Symmetric polynomials and polytopes: We call $f \in \mathbb{F}[x_1,\ldots,x_n]$ a *symmetric* polynomial if $f(x_1,\ldots,x_n) = f(x_{\sigma(1)},\ldots,x_{\sigma(n)})$, for any permutation $\sigma \in S_n$. It is equivalent to saying: f is symmetric if and only if $\text{coeff}(x_1^{e_1}\cdots x_n^{e_n})(f) = \text{coeff}(x_1^{e_{\sigma(1)}}\cdots x_n^{e_{\sigma(n)}})(f)$, for all $\sigma \in S_n$. We call $f \in \mathbb{F}[x_1,\ldots,x_n]$ a *symmetric-support* polynomial if for each monomial $x_1^{e_1}\cdots x_n^{e_n}$, we have $\text{coeff}(x_1^{e_1}\cdots x_n^{e_n})(f) \neq 0 \Rightarrow \text{coeff}(x_1^{e_{\sigma(1)}}\cdots x_n^{e_{\sigma(n)}})(f) \neq 0$, for every $\sigma \in S_n$. Note that all symmetric polynomials are also symmetric-support polynomials.

We say that a polytope P is symmetric if for each point $(v_1,\ldots,v_n)$ in P, $(v_{\sigma(1)},\ldots,v_{\sigma(n)})$ is also in P, for every permutation $\sigma \in P$. Note that Newton polytopes of symmetric-support polynomials are in fact symmetric!

δ-entropy: We say that a vector $\mathbf{v} \in \{0,1,\ldots,d\}^n$ is of δ-entropy if it has $(n-\delta)$ equi-valued coordinates. More generally, we define $\mathbf{v}$ to be of $(\delta_1,\ldots,\delta_d)$-entropy if it has d distinct values from $\{0,1,\ldots,d\}$ occurring with $\delta_1 \geq \ldots \geq \delta_d$ frequencies in $\mathbf{v}$ respectively, and the remaining value with $\delta_0 := n - \sum_{i=1}^d \delta_i$ frequency. A δ-entropy vector in the generalized notation is, thus, a $(\delta_1,\ldots,\delta_d)$-entropy vector with $\sum_{i=1}^d \delta_i \leq \delta$. We also extend this definition of entropy to sets and polynomials. We call $A \subseteq \{0,1,\ldots,d\}^n$ a δ-entropy set, if for every vector $\mathbf{v} \in A$, $\mathbf{v}$ is of $(\leq \delta)$-entropy. We call f a δ-entropy polynomial, if its support set $\text{supp}(f)$ is of δ-entropy. Example 4.10 gives a polynomial f with 1-entropy. In contrast, consider $f = x_1\cdots x_{n/3} + x_{n/3+1}\cdots x_n$ which is a polynomial with high entropy of $(n/3)$; and is not $(< n/3)$-entropy.

3 Polytope Entropy– Theorem 3

We first discuss Theorem 3 as it is the core structural observation driving all the results in this work. Suppose we are given a δ-entropy set V of exponents from $\{0,1,\ldots,d\}^n$. The motivating

question is that if all vertices are of low-entropy, then how large can the entropy of internal points in their convex-span be? In this theorem, we prove that the entropy blows up only by a factor of $O(d)$ for the integral-points, which is a small factor in the bounded individual degree regime. Intuitively, internal integral-points inherit nice properties of vertices as they can be expressed as their convex combination, the nice property being low-entropy in this theorem. In order to prove this, we first design a set of symmetric hyperplane equations such that for each hyperplane, the entire δ-entropy set V lies on one side of its halfspace (Claim 3.2). Thus, every point in convex-span of V must also lie on the same side (Lemma 3.7). Secondly, the hyperplane equations are so designed that any *integral*-point lying on that side must have entropy at most $O(d\delta)$ (Claim 3.4).

The clever design of the hyperplane equations is the novelty of this work and we present it as a new approach for factor-sparsity bounds. Besides the factor-sparsity implications, we believe that Theorem 3 is of interest independently also and might have other applications in convex geometry and polytope theory.

Let P_f be the Newton polytope of f and V_f be its set of vertices such that $P_f = CS(V_f)$. We prove few claims below which will help us prove Theorem 3. We are given V_f to be a δ-entropy set. Let $m := n - \delta$. We first design a hyperplane $\mathbf{u}^\intercal \cdot \mathbf{y} + d\delta = 0$ that will help us prove very useful properties in Claim 3.2 and Claim 3.4 below. Define column vector $\mathbf{u} \in \{-1, +1\}^n$ such that,

$$u_i := \begin{cases} +1 & \text{if } i \leq \delta + m/2 \\ -1 & \text{otherwise.} \end{cases} \tag{3.1}$$

The first $\delta + m/2$ coordinates of $\mathbf{u}$ are $+1$ and the remaining $n - (\delta + m/2) = m/2$ coordinates are -1.

Claim 3.2 (Hyperplane cover)**.** *Let $V \subseteq \{0, 1, \ldots, d\}^n$ be a δ-entropy set and $\mathbf{u}$ be as defined in (3.1). Then, every point $\mathbf{y} \in V$ satisfies each of the following symmetric inequalities:*

$$(\sigma \circ \mathbf{u})^\intercal \cdot \mathbf{y} + d\delta \;\geq\; 0, \quad \textit{for each } \sigma \in S_n \,. \tag{3.3}$$

Proof. Let $\sigma \in S_n$ be any permutation. Define $l := (\sigma \circ \mathbf{u})^\intercal \cdot \mathbf{y} + d\delta$. Minimum value of l is attained when coordinates corresponding to -1 in $\sigma \circ \mathbf{u}$ are filled with the largest possible value in $\mathbf{y}$, which is d, as $\mathbf{y} \in V \subseteq \{0, 1, \ldots, d\}^n$.

Fix a $\leq \delta$-entropy point $\mathbf{y}$; so assume that $\mathbf{y}$ has $\geq m = n - \delta$ coordinates equal to i for some $0 \leq i \leq d$. To minimize l, by varying $(\sigma \circ \mathbf{u})$, we can place at most δ many d's in the coordinates corresponding to -1 and rest of the $\geq m$ coordinates must be filled by i. Since we have a total of $(\delta + m/2)$ coordinates equal to $+1$ and remaining $(m/2)$ coordinates equal to -1, the minimum value of l in this case is $(\delta + m/2) \cdot i - (m/2 - \delta) \cdot i - (\delta) \cdot d \; + d\delta \;=\; 2\delta i \;\geq 0$. Thus, $l \geq 0$ for our $\mathbf{y} \in V$ and $\forall \sigma \in S_n$. Implying $(\sigma \circ \mathbf{u})^\intercal \cdot \mathbf{y} + d\delta \geq 0$ in all cases, as promised. □

The symmetry in the above hyperplane-cover inequalities helps us to argue about $\mathbf{y}$, by first *sorting its coordinates*. This a powerful trick, as our following central claim demonstrates.

Claim 3.4 (Integral-point in the cover)**.** *Let* $\mathbf{u}$ *be as defined in* (3.1)*. Let* $\mathbf{y} \in \{0, 1, \ldots, d\}^n$ *be a point with* $0 \leq y_1 \leq y_2 \leq \ldots \leq y_n \leq d$ *such that* $\mathbf{u}^\intercal \cdot \mathbf{y} + d\delta \geq 0$ *holds. Then,* $\mathbf{y}$ *is a* $(2d\delta)$*-entropy point.*

Proof. Recall $n = \delta + m$. Let us call, expectedly, the first $(\delta + m/2)$ coordinates, the 'positive zone' of $\mathbf{y}$ and the remaining $(m/2)$ coordinates, the 'negative zone' of $\mathbf{y}$; this corresponds to positions of $+1$'s and -1's in $\mathbf{u}$ respectively. By hypothesis, the positive zone has coordinates at most as large as in the negative zone. Let $\mathbf{y} =: \mathbf{y}_+ + \mathbf{y}_-$, where we define $\mathbf{y}_+$ and $\mathbf{y}_-$ as follows,

$$(\mathbf{y}_+)_j \coloneqq \begin{cases} \mathbf{y}_j & \text{if } j \leq \delta + m/2 \\ 0 & \text{otherwise.} \end{cases}$$

$$(\mathbf{y}_-)_j \coloneqq \begin{cases} \mathbf{y}_j & \text{if } j > \delta + m/2 \\ 0 & \text{otherwise.} \end{cases}$$

Suppose the first coordinate in the negative zone of $\mathbf{y}$ is i for some $i \in \{0, 1, \ldots, d\}$. We then claim that $\mathbf{y}$ has at least $n - 2d\delta$ coordinates with this same value i, proving $\mathbf{y}$ to be a $2d\delta$-entropy point. Let p_+ and p_- denote the frequency of i in positive and negative zones of $\mathbf{y}$ respectively, for some integers $p_+, p_- \geq 0$. Note that $\mathbf{u}^\intercal \cdot \mathbf{y} = \|\mathbf{y}_+\|_1 - \|\mathbf{y}_-\|_1$, where $\|\cdot\|_1$ is the $\mathbf{L^1}$-norm. We first upper bound $\|\mathbf{y}_+\|_1$. Since $\mathbf{y}$ is sorted, the last p_+ coordinates in positive zone must be i and all coordinates preceding it are of value at most $i - 1$. This gives us

$$\|\mathbf{y}_+\|_1 \leq \ (i-1) \cdot (\delta + m/2 - p_+) \ + \ i \cdot (p_+) \ = \ i\delta + im/2 - \delta - m/2 + p_+ \ . \tag{3.5}$$

Now, we lower bound $\|\mathbf{y}_-\|_1$. Since $\mathbf{y}$ is sorted, the first p_- coordinates in negative zone must be i and all subsequent coordinates are of value at least $(i+1)$. This gives us

$$\|\mathbf{y}_-\|_1 \geq \ i \cdot (p_-) \ + \ (i+1) \cdot (m/2 - p_-) \ = \ im/2 + m/2 - p_- \ . \tag{3.6}$$

Let $l \coloneqq \mathbf{u}^\intercal \cdot \mathbf{y} + d\delta$. By hypothesis, $l \geq 0$. Then using (3.5) and (3.6) together, we observe that

$$\begin{aligned} 0 \leq \ l \ &= \ \mathbf{u}^\intercal \cdot \mathbf{y} + d\delta \ = \ \|\mathbf{y}_+\|_1 - \|\mathbf{y}_-\|_1 + d\delta \\ 0 \leq \ & i\delta + im/2 - \delta - m/2 + p_+ \ - \ (im/2 + m/2 - p_-) \ + \ d\delta \\ &= \ (i+d)\delta - \delta - m + p_+ + p_- \\ &\leq \ 2d\delta - n + p_+ + p_- \qquad (\text{since } i \leq d \text{ and } n = m + \delta) \\ n - 2d\delta \leq \ & p_+ + p_- \ . \end{aligned}$$

Recall that total frequency of the value i in $\mathbf{y}$ is $p_+ + p_- \geq n - 2d\delta$. This proves that $\mathbf{y}$ is a $(2d\delta)$-entropy point; finishing the proof. (We note that the calculations in eq. (3.5) and eq. (3.6) are technically for $i \in [d-1]$. For the corner cases of $i = 0$ or $i = d$, the same logic will work and in fact with a better lower bound on frequency of i in $\mathbf{y}$.) □

We now show that if for a set of points, each point lies on one side of a hyperplane, then all the points in their convex-span also lie on that side.

Lemma 3.7 (Halfspace is convex)**.** *Let $V = \{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k\} \subseteq \mathbb{R}^n$ be a set of k vectors. Suppose each of these vectors lie in the same halfspace, i.e. for some $(\mathbf{a}, b) \in \mathbb{R}^n \times \mathbb{R}$, for each $i \in [k]$, $\mathbf{a}^\intercal \cdot \mathbf{v}_i + b \geq 0$. Then, any vector in the convex-span of V also lies in the same halfspace, i.e. for each $\mathbf{v} \in CS(V)$, $\mathbf{a}^\intercal \cdot \mathbf{v} + b \geq 0$.*

Proof. This follows because $\mathbf{v}$ is a convex combination of vectors in V. Let $\mathbf{v} = \alpha_1 \cdot \mathbf{v}_1 + \ldots + \alpha_k \cdot \mathbf{v}_k$, where $\sum_{i=1}^k \alpha_i = 1$ and $\alpha_i \in \mathbb{R}_{\geq 0}$ for each $i \in [k]$. Thus,

$$\begin{aligned}
\mathbf{a}^\intercal \cdot \mathbf{v} + b &= \mathbf{a}^\intercal \cdot \left(\sum_{i=1}^k \alpha_i \cdot \mathbf{v}_i \right) + b \\
&= \left(\sum_{i=1}^k \alpha_i \cdot \mathbf{a}^\intercal \cdot \mathbf{v}_i \right) + \sum_{i=1}^k \alpha_i \cdot b \\
&= \sum_{i=1}^k \alpha_i \cdot (\mathbf{a}^\intercal \cdot \mathbf{v}_i + b) \geq 0.
\end{aligned}$$

In the second step, we use $\sum_{i=1}^k \alpha_i = 1$; while in the last step we use the hypothesis and $\alpha_i \geq 0$ for each $i \in [k]$. □

We are now ready to prove the main theorem of this section.

Theorem 3 (Polytope Entropy Theorem)**.** *Let $V \subseteq \{0, 1, \ldots, d\}^n$ be a δ-entropy set, then $CS(V) \cap \mathbb{Z}^n$ is a $(2d\delta)$-entropy set.*

Proof. In Claim 3.2, we showed that every point $\mathbf{v} \in V$ belongs to the halfspace $\mathbf{u}^\intercal \cdot \mathbf{v} + d\delta \geq 0$. In fact, we proved it for every permutation $\sigma \in S_n$ of $\mathbf{u}$. Let $\mathbf{y}$ be any integral-point in $CS(V)$. Thus by Lemma 3.7, $(\sigma \circ \mathbf{u})^\intercal \cdot \mathbf{y} + d\delta \geq 0$, for every $\sigma \in S_n$.

Let $\pi \in S_n$ be the permutation which sorts $\mathbf{y}$, i.e. $y_{\pi(1)} \leq y_{\pi(2)} \leq \ldots \leq y_{\pi(n)}$. These are integers in $\{0, \ldots, d\}$ due to convexity. Since the hyperplane cover holds for every permutation, in particular it holds for $\pi^{-1} \in S_n$ also, i.e. $(\pi^{-1} \circ \mathbf{u})^\intercal \cdot \mathbf{y} + d\delta \geq 0$. Thus,

$$(\pi^{-1} \circ \mathbf{u})^\intercal \cdot \mathbf{y} + d\delta \;=\; \mathbf{u}^\intercal \cdot (\pi \circ \mathbf{y}) + d\delta \;\geq\; 0\,.$$

Now by Claim 3.4, as $\pi \circ \mathbf{y}$ is sorted, we deduce that $\pi \circ \mathbf{y}$ is a $(2d\delta)$-entropy vector. Since entropy of a vector does not change on permuting it, we deduce that $\mathbf{y}$ is also a $(2d\delta)$-entropy point. □

Remark. Note that Theorem 3 is fairly tight, i.e. a blow-up in entropy of internal integral points by a factor of d is inevitable. To observe this, consider $V = \{\sigma \circ (d, 0, \ldots, 0) \mid \sigma \in S_n\}$, with entropy $\delta = 1$. It is easy to see that the integral vector $\mathbf{v} = \sum_{i=1}^d e_i$ is in $CS(V)$, where e_i is the standard unit vector with a single 1 in position i and rest all 0. Thus, $\mathbf{v}$ is a vector with 1 in first d coordinates and remaining all 0, and it has entropy exactly d, for $d \leq n/2$. Therefore, $CS(V) \cap \mathbb{Z}^n$ is of entropy at least $d\delta$. However, in this work we are concerned with $d = O(1)$, so even $2d\delta$ blowup is fine.

4 Factor sparsity bounds: Proof of Theorems 1 & 2

If f is a symmetric polynomial, then its Newton polytope P_f is a symmetric polytope. We prove that a monomial is a vertex of P_f if and only if all its S_n permutations are also vertices of P_f.

Lemma 4.1 (Vertices of symmetric polytope)**.** *Let P be a symmetric Newton polytope with $V(P)$ being its vertex set. Then, for $\sigma \in S_n$: $\mathbf{v} \in V(P)$ if and only if $\sigma \circ \mathbf{v} \in V(P)$.*

Proof. ($\Rightarrow$) Suppose $V(P) =: \{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$. Let $\mathbf{v} \in V(P)$. Consider any permutation $\sigma \in S_n$. Since P is a symmetric polytope, we at least know that $\sigma \circ \mathbf{v} \in P$. For the sake of contradiction, suppose $\sigma \circ \mathbf{v} \notin V(P)$. Thus, it is an internal point which can be expressed as a nontrivial convex combination of vertices. There exist, for $i \in [k]$, $0 \leq \alpha_i < 1$, $\sum_{i=1}^{k} \alpha_i = 1$ such that:

$$\begin{aligned} \sigma \circ \mathbf{v} &= \alpha_1 \cdot \mathbf{v}_1 + \ldots + \alpha_k \cdot \mathbf{v}_k \\ \sigma^{-1} \circ (\sigma \circ \mathbf{v}) &= \sigma^{-1} \circ (\alpha_1 \cdot \mathbf{v}_1 + \ldots + \alpha_k \cdot \mathbf{v}_k) \\ \mathbf{v} &= \alpha_1 \cdot \sigma^{-1} \circ \mathbf{v}_1 + \ldots + \alpha_k \cdot \sigma^{-1} \circ \mathbf{v}_k \end{aligned}$$

Observe that $\sigma^{-1} \in S_n$ and since P is symmetric $\sigma^{-1} \circ \mathbf{v}_i \in P$ for all $i \in [k]$. This means, $\mathbf{v}$ is a nontrivial convex combination of other points in P, which contradicts the fact that $\mathbf{v}$ is a vertex (Section 2). Therefore, $\sigma \circ \mathbf{v}$ must also be a vertex.

($\Leftarrow$) Now we wish to prove that all permutations of a non-vertex point must also be non-vertices. Let $\mathbf{v} \notin V(P)$. Then, for any $\sigma \in S_n$, we get a nontrivial convex combination:

$$\begin{aligned} \mathbf{v} &= \alpha_1 \cdot \mathbf{v}_1 + \ldots + \alpha_k \cdot \mathbf{v}_k \\ \sigma \circ \mathbf{v} &= \alpha_1 \cdot \sigma \circ \mathbf{v}_1 + \ldots + \alpha_k \cdot \sigma \circ \mathbf{v}_k \end{aligned}$$

Again, since P is symmetric $\sigma \circ \mathbf{v}_i \in P$ for all $i \in [k]$. Thus, $\sigma \circ \mathbf{v}$ is a nontrivial convex combination of other points, hence it must be a non-vertex. □

We state few standard bounds in the lemma below which will be useful in this section.

Lemma 4.2 (Counting estimates)**.** 1. *For positive integers a, b, c, with $a \geq bc$: $(a/b)^b \leq \binom{a}{b}$, and*

2. $\binom{a}{bc} \leq \binom{a}{c}^b$.

3. *For positive real x :* $\log(1+x) > \ln(1+x) > x - \frac{x^2}{2}$.

Proof. For (1), see that $\binom{a}{b} = \frac{a(a-1)\cdots(a-b+1)}{b(b-1)\cdots 1} \geq \left(\frac{a}{b}\right)^b$. For (2), we make use of $\binom{a}{c+b'} \leq \binom{a}{b'} \cdot \binom{a-b'}{c} \leq \binom{a}{b'} \cdot \binom{a}{c}$. Use this b times to get $\binom{a}{bc} \leq \binom{a}{c}^b$. For (3), show that derivative of f is positive for $x > 0$, where $f = \ln(1+x) - (x - \frac{x^2}{2})$, and for $x = 0$, $f(0) = 0$. □

Given a δ-entropy set, we now bound the total number of integral-points in its convex-span. Below, we state it specifically for the vertex set V_g of Newton polytope P_g of a polynomial g.

Lemma 4.3 (Entropy-sparsity upper bound)**.** *Let $g \in \mathbb{F}[\mathbf{x}]$ be any polynomial of individual degree $\leq d$, such that V_g is a δ-entropy set. Then, g has sparsity at most $\binom{n}{\delta}^{2d} \cdot d^{O(d\delta+1)}$.*

Proof. By Theorem 3, we have that $P'_g := CS(V_g) \cap \mathbb{Z}^n$ is a $2d\delta$-entropy set. Thus, for a point, in the worst case, we have to choose $n - 2d\delta$ positions for the equal coordinates, and the integral-value to be chosen for those coordinates has $(d+1)$ choices. Remaining $2d\delta$ positions have at most d choices of values each. Thus, we get

$$\begin{aligned} |P'_g| \;&\leq\; \binom{n}{n-2d\delta} \cdot (d+1) \cdot d^{2d\delta} \\ &\leq\; \binom{n}{2d\delta} \cdot d^{O(d\delta)} \;\leq\; \binom{n}{\delta}^{2d} \cdot d^{O(d\delta)} \,. \end{aligned}$$

The last step above follows from Lemma 4.2. Finally, we note that $\mathrm{supp}(g) \subseteq P'_g$ and hence, $\|g\| \leq |P'_g|$. The stated bound is then valid for any $\delta \geq 1$. For the special case of $\delta = 0$, we observe that the bound is simply $d+1$ then. □

Given a symmetric set of δ-entropy, we now show a simple lower bound on its cardinality.

Lemma 4.4 (Entropy-Sparsity lower bound for symmetric)**.** *Let $g \in \mathbb{F}[x_1, \ldots, x_n]$ be symmetric of individual degree $\leq d$, such that V_g is a δ-entropy set, for smallest possible δ. Then, $|V_g| \geq \binom{n}{\delta}$.*

Proof. Since V_g is exactly a δ-entropy set, it contains some vector $\mathbf{v}$ with maximum-frequency of coordinate-values exactly $n - \delta$. By Lemma 4.1, V_g must also contain all the distinct S_n-permutations of this vector. Therefore, in the minimal size case, V_g contains all the distinct S_n-permutations of $\mathbf{v}$ which has only two different values, one with frequency $(n-\delta)$ while the other with frequency δ. Thus, $|V_g| \geq \binom{n}{\delta}$. □

Lemma 4.4 helps us upper bound the entropy of a symmetric vertex set as follows.

Lemma 4.5 (symmetry ⇒ low entropy)**.** *Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ be an s-sparse polynomial. Let g be any symmetric factor of f with individual degree $\leq d$ such that V_g is a δ-entropy set, for smallest possible δ. Then, $\delta = O(d \log s)$.*

Proof. Since g has individual degree $\leq d$, we know that $V_g \subseteq \{0, 1, \ldots, d\}^n$. Thus by a simple averaging argument, for any point in V_g, there exists a coordinate with frequency $\geq n/(d+1)$ in that value. Therefore,

$$\delta \;\leq\; n - \frac{n}{d+1} \;=\; n\left(1 - \frac{1}{d+1}\right) . \tag{4.6}$$

We know: $s \geq |V_f| \geq |V_g|$. Further, $|V_g| \geq \binom{n}{\delta}$, by Lemma 4.4. Hence,

$$s \;\geq\; \binom{n}{\delta} . \tag{4.7}$$

We now show that (4.7) constrains the value of δ to be low with the additional help of (4.6). Using Lemma 4.2, we deduce the following, starting from (4.7):

$$\begin{aligned} s &\geq \left(\frac{n}{\delta}\right)^{\delta} . \\ \text{Thus, } \log s &\geq \delta \cdot \log\left(\frac{n}{\delta}\right) \\ &\geq \delta \cdot \log\left(\frac{n}{n\left(1-\frac{1}{d+1}\right)}\right) \quad \text{[Using (4.6)]} \\ &= \delta \cdot \log\left(1+\frac{1}{d}\right) \\ &> \delta \cdot \left(\frac{1}{d}-\frac{1}{2d^2}\right) \quad \text{[Using Lemma 4.2]} \\ &\geq \frac{\delta}{2d} \quad \text{[As } d \geq 1\text{]} . \end{aligned}$$

This proves that $\delta = O(d \log s)$. □

The above 'low'-entropy upper-bound is non-constant; so it does not directly give a good sparsity-bound on g. Yet, cleverly using all the machinery developed till now, we will prove factor-sparsity bounds.

Theorem 4 (Symmetric factor sparsity bound)**.** *Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ be an s-sparse polynomial. Let g be any symmetric factor of f, with individual degree at most d. Then, g has sparsity at most $s^{O(d^2 \log d)}$.*

Proof. Let V_g be a δ-entropy set, for the smallest possible $\delta \geq 0$. By Lemma 4.3,

$$\|g\| \leq \binom{n}{\delta}^{2d} \cdot d^{O(d\delta+1)} .$$

Also by Lemma 4.4, we know that $|V_g| \geq \binom{n}{\delta}$. Moreover, $|\text{supp}(f)| \geq |V_f| \geq |V_g|$. Thus, $\binom{n}{\delta} \leq s$. This gives us:

$$\begin{aligned} \|g\| &\leq s^{2d} \cdot d^{O(d\delta)} \\ &\leq s^{2d} \cdot d^{O(d^2 \log s)} \quad \text{[Using Lemma 4.5]} \\ &= s^{2d} \cdot s^{O(d^2 \log d)} \leq s^{O(d^2 \log d)} . \end{aligned}$$

□

The above proof analysis actually proves something stronger than Theorem 4: the convex hull of any symmetric subset of $\{0, 1, \ldots, d\}^n$ must have *many* vertices (corner points). We state this *incompressibility result* formally as Corollary 3 below.

Corollary 3 (Symmetric polytope count)**.** *Let $E \subseteq \{0, 1, \ldots, d\}^n$ be a symmetric set of points. Let $t := |V(CS(E))|$, then $t \geq |E|^{\Omega(1/(d^2 \log d))}$.*

Proof. Since E is a symmetric set of points, $CS(E)$ is a symmetric polytope and its vertex set $V(CS(E))$ is also symmetric by Lemma 4.1. By Lemma 4.5, the entropy of $V(CS(E))$ is $\delta := O(d \log t)$. Thus, the set $CS(E) \cap \mathbb{Z}^n$ is of $O(d^2 \log t)$-entropy by Theorem 3. Therefore, $|E| \leq |CS(E) \cap \mathbb{Z}^n| \leq t^{O(d^2 \log d)}$ by Lemma 4.3 and Lemma 4.4. □

Remark. In Theorem 4 (resp. Corollary 3), we do not require the whole of g (resp. E) to be symmetric, rather we only need its vertex set V_g (resp. $V(CS(E))$) to be symmetric, which is a much weaker requirement. Similarly, as will be clear below, we don't need f to be symmetric in Theorem 1 but only V_f to be symmetric. In these cases, we will not require use of Lemma 4.1.

Proof of Main Theorems. We first prove Theorem 2 showing factor-sparsity bound for low-entropy polynomials.

Proof of Theorem 2. Consider the $\mathbf{u}$ defined in (3.1). Denote the integral-points contained in the hyperplane-cover as $T := \{\mathbf{y} \in \{0, 1, \ldots, d\}^n \mid (\sigma \circ \mathbf{u})^\intercal \cdot \mathbf{y} + d\delta \geq 0, \text{ for each } \sigma \in S_n\}$. We now claim that $|T| \leq (nd)^{O(d\delta)}$. As done in proof of Theorem 3, without loss of generality, we can assume $\mathbf{y}$ to be sorted and focus on a single halfspace,

$$\mathbf{u}^\intercal \cdot \mathbf{y} + d\delta \geq 0 .$$

Then by Claim 3.4, $\mathbf{y}$ is a $2d\delta$-entropy point. Thus, we can bound $|T|$ on the same lines as in Lemma 4.3,

$$|T| \leq \binom{n}{2d\delta} \cdot (d+1) \cdot d^{2d\delta} \leq (nd)^{O(d\delta)} .$$

By hypothesis, f is a δ-entropy polynomial. In particular, $V_f \subseteq \text{supp}(f)$ is a δ-entropy set. By Claim 3.2, we note that (3.3) holds for each $\mathbf{a} \in V_f$ and consequently also for each $\mathbf{b} \in P_f$ by Lemma 3.7. That is,

$$(\sigma \circ \mathbf{u})^\intercal \cdot \mathbf{b} + d\delta \geq 0 \tag{4.8}$$

for each $\mathbf{b} \in P_f$ and $\sigma \in S_n$. Consider a fixed point $\mathbf{v} \in \text{supp}(h)$. For each point $\mathbf{z} \in \text{supp}(g)$, translate it by $+\mathbf{v}$. Observe that, $\mathbf{z} + \mathbf{v} \in P_f$, since $P_f = P_g + P_h$ and $\mathbf{z} \in \text{supp}(g) \subseteq P_g, \mathbf{v} \in \text{supp}(h) \subseteq P_h$. This means that $\mathbf{z} + \mathbf{v}$ satisfies (4.8),

$$(\sigma \circ \mathbf{u})^\intercal \cdot (\mathbf{z} + \mathbf{v}) + d\delta \geq 0 .$$

Moreover, $\mathbf{z} + \mathbf{v} \in P_f \cap \{0, 1, \ldots, d\}^n$ by convexity. Therefore, for each $\mathbf{z} \in \text{supp}(g)$, we get a distinct point $\mathbf{z} + \mathbf{v} \in T$. Hence, $|\text{supp}(g)| \leq |T|$ gives the desired bound. □

We finally show how to deduce Theorem 1 from Theorem 2 in order to get factor-sparsity bound for a symmetric, sparse polynomial.

Proof of Theorem 1. Since f is symmetric, consider the symmetric polytope $P_f = CS(\text{supp}(f))$. Let V_f be the vertex set of P_f such that it is a δ-entropy set, for smallest possible δ. Since f is symmetric, $s \geq |V_f| \geq \binom{n}{\delta}$ by Lemma 4.4. Now with the exact same analysis as done in the proof of Lemma 4.5, we deduce that $\delta = O(d \log s)$.

Thus, f is a sufficiently low-entropy polynomial and we run the same proof as Theorem 2, except that we bound $|T|$ a bit differently as follows:

$$\begin{aligned} |T| &\leq \binom{n}{2d\delta} \cdot (d+1) \cdot d^{2d\delta} \\ &\leq \binom{n}{\delta}^{2d} \cdot d^{O(d\delta)} \quad \text{[Using Lemma 4.2]} \\ &\leq s^{2d} \cdot d^{O(d\delta)} = s^{2d} \cdot d^{O(d^2 \log s)} \leq s^{O(d^2 \log d)} . \end{aligned}$$

Thus, we get the desired sparsity bound for any factor g of f, using the proof of Theorem 2. □

Remark. In all the sparsity-bound theorems above, for the corner case of entropy $\delta = 0$, sparsity of factor g is simply, $\|g\| \leq d+1$. This is because of the $d+1$ upper bound in Lemma 4.3 for $\delta = 0$.

4.1 Tightness of sparsity bounds

We give two examples below where factors of a sparse polynomial have significantly high sparsity, unless the individual degree is bounded.

Example 4.9. [vzGK85] Consider the following polynomial f with individual degree d, which factorizes as $f = gh$, where

$$\begin{aligned} f &= \prod_{i=1}^{n} (x_i^d - 1) , \\ g &= \prod_{i=1}^{n} (1 + x_i + \ldots + x_i^{d-1}) , \\ h &= \prod_{i=1}^{n} (x_i - 1) . \end{aligned}$$

Observe that $\|f\| = 2^n$, while $\|g\| = d^n$. If we let $s := \|f\|$, then $\|g\| = s^{\log d}$. Over fields of characteristic 0, this is an example which exhibits highest known blowup in sparsity. Moreover, it also shows that $s^{\log d}$ is a lower bound on factor-sparsity. Note that f is a symmetric polynomial and Theorem 1 shows that $\|g\| \leq s^{O(d^2 \log d)}$.

Example 4.10. [BSV20] Over finite fields we have the following polynomial which has a higher exponential blowup in factor-sparsity. Consider the following polynomial $f \in \mathbb{F}_p[x_1, \ldots, x_n]$ with individual degree p, for some prime p and let $0 < d < p$. We have $f = gh$, where

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

$$g = (x_1 + x_2 + \ldots + x_n)^d ,$$
$$h = (x_1 + x_2 + \ldots + x_n)^{p-d} .$$

Observe that $\|f\| = n$, while $\|g\| = \binom{n+d-1}{d} \approx n^d$. If we let $s := \|f\|$, then $\|g\| \approx s^d$. The factor-sparsity bounds in this work hold for any field $\mathbb{F}$, finite or otherwise. Moreover, f is also symmetric and falls under the purview of Theorem 1, which shows that $\|g\| \leq s^{O(d^2 \log d)}$. Hence, this example shows that in Theorem 1, we cannot do better than s^d.

Note that this f is also a low-entropy polynomial, in fact with $\delta = 1$ as all the exponent vectors in $\mathrm{supp}(f)$ have $n-1$ coordinates having the same value 0. Thus, we can use Theorem 2 for this f to get a factor-sparsity bound of $(np)^{O(p)}$ which is really close to the actual sparsity n^d.

5 Factoring algorithms: Proof of Corollaries 1 & 2

A nice feature of the results in [BSV20] is that once you prove a nice factor-sparsity bound, they also show how to get a deterministic factoring algorithm for sparse polynomials which runs in time polynomial in the sparsity bound proven. We restate this as Lemma 5.1 below.

Lemma 5.1. *[BSV20, Theorem 5.8] Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ be an s-sparse polynomial with individual degrees at most d. Let $\xi(n, d, s)$ be the upper bound on sparsity for every factor of f. Then given f, there is a deterministic algorithm that computes complete factorization of f in $(n \cdot \xi(n, d^2, s^d))^{O(d^2)} \cdot \mathrm{poly}(c_{\mathbb{F}}(d^2))$ field operations.*

Here $c_{\mathbb{F}}(d)$ is the best known time complexity for factoring a univariate polynomial of degree d over the field $\mathbb{F}$. In [BSV20], they showed that $\xi(n, d, s) \leq s^{O(d^2 \log n)}$ and then used Lemma 5.1 to get a factoring algorithm of $s^{\mathrm{poly}(d) \log n}$ time complexity, which is quasi-polynomial in the bounded individual degree setting. In this work, we deal with symmetric and constant-entropy input polynomials, for which we show that $\xi(n, d, s) \leq (ns)^{\mathrm{poly}(d)}$. Thus, we can use Lemma 5.1 to get polynomial time factoring algorithms in the bounded individual degree setting. The finer time complexities are discussed in the proofs of Corollary 1 and Corollary 2 below. Also, note that $\xi(n, d, s)$ is a lower bound on time complexity for sparse factoring algorithms as we output each factor as an explicit list of monomials.

Proof of Corollary 1. Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ be an s-sparse, symmetric polynomial with individual degree d. In Theorem 1, we show that $\xi(n, d, s) \leq s^{O(d^2 \log d)}$. Plugging this value in Lemma 5.1, we get a deterministic factoring algorithm for f. The time complexity $T(n, s, d)$ for this algorithm is:

$$\begin{aligned} T(n, s, d) &= (n \cdot \xi(n, d^2, s^d))^{O(d^2)} \cdot \mathrm{poly}(c_{\mathbb{F}}(d^2)) \\ &= \left(n \cdot s^{d \cdot O(d^4 \log d^2)}\right)^{O(d^2)} \cdot \mathrm{poly}(c_{\mathbb{F}}(d^2)) \\ &= s^{O(d^7 \log d)} \cdot n^{O(d^2)} \cdot \mathrm{poly}(c_{\mathbb{F}}(d^2)) \end{aligned}$$

□

Proof of Corollary 2. Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ be a δ-entropy polynomial with individual degree d. In Theorem 1, we show that $\xi(n, d, s) \leq (nd)^{O(d\delta)}$. Plugging this value in Lemma 5.1, we get a deterministic factoring algorithm for f. The time complexity $T(n, s, d)$ for this algorithm is:

$$\begin{aligned} T(n,s,d) &= (n \cdot \xi(n, d^2, s^d))^{O(d^2)} \cdot \text{poly}(c_{\mathbb{F}}(d^2)) \\ &= \left(n \cdot (nd^2)^{O(d^2\delta)}\right)^{O(d^2)} \cdot \text{poly}(c_{\mathbb{F}}(d^2)) \\ &= (nd)^{O(d^4\delta)} \cdot \text{poly}(c_{\mathbb{F}}(d^2)) \end{aligned}$$

□

6 Future directions

In this work, we show an $s^{O(d^2 \log d)}$ upper bound on the factor-sparsity of an s-sparse, symmetric polynomial with individual degree at most d. The crucial reason for why we were able to get a nice bound was that symmetric polynomials have low-entropy (Lemma 4.5 and Theorem 3). In general, an s-sparse polynomial may not be symmetric and could have high entropy like $n/2$, for which Theorem 2 would give a bound with an $n/2$-term in its exponent, which is undesirable. The main open problem is to prove $s^{\text{poly}(d)}$ upper bound for a general (possibly non-symmetric) s-sparse polynomial. The multilinear ($d = 1$) and multiquadratic cases ($d = 2$) are solved already in the works of [SV10], [Vol17] respectively. The next step in this direction would be to prove this upper bound when input polynomial is multi*cubic* ($d = 3$).

In this work, we also show an $(nd)^{O(d\delta)}$ sparsity upper bound for factors of δ-entropy polynomials. This bound is efficient when δ is constant. We leave it as an open question to study factors of sparse polynomials with high-entropy. Can one even show an $(snd)^{\text{poly}(d)}$ factor-sparsity bound for s-sparse, $(\log n)$-entropy polynomial of individual degree d? We note that the example in [BSV20, Claim 4.4] is a polynomial with its vertices having exactly $n/2$-entropy such that its Newton polytope has at least $n^{\Omega(\log n)}$-many integral points. Therefore, a new technique which goes beyond counting number of internal points in a polytope is required, in case a $(snd)^{\text{poly}(d)}$ factor-sparsity bound does exist for high-entropy polynomials.

Another interesting problem in this context is the irreducibility testing problem which can be thought of as the decision version of factorization problem. Let f be an s-sparse polynomial with individual degree d, can we test whether f is reducible in $s^{\text{poly}(d)}$ time? Can this problem be solved without proving the best factor-sparsity bounds?

Acknowledgments

Pranav thanks Ilya Volkovich and Vishwas Bhargava for useful discussions. Nitin thanks the funding support from DST-SERB (CRG/2020/000045) and N. Rama Rao Chair.

References

[AGS19] Manindra Agrawal, Sumanta Ghosh, and Nitin Saxena. Bootstrapping variables in algebraic circuits. *Proceedings of the National Academy of Sciences*, 116(17):8107–8118, 2019. 27

[AV08] Manindra Agrawal and V Vinay. Arithmetic circuits: A chasm at depth four. In *2008 49th Annual IEEE Symposium on Foundations of Computer Science*, pages 67–75. IEEE, 2008. 27

[BJ18] Markus Bläser and Gorav Jindal. On the complexity of symmetric polynomials. In *10th Innovations in Theoretical Computer Science Conference (ITCS 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018. 6

[BS21] Pranav Bisht and Nitin Saxena. Blackbox identity testing for sum of special ROABPs and its border class. *computational complexity*, 30(1):1–48, 2021. 27

[BSV20] Vishwas Bhargava, Shubhangi Saraf, and Ilya Volkovich. Deterministic factorization of sparse polynomials with bounded individual degree. *Journal of the ACM (JACM)*, 67(2):1–28, 2020. (Preliminary version in FOCS 2018). 3, 4, 5, 6, 7, 8, 9, 10, 11, 19, 20, 21, 28, 29

[CKL+20] Prasad Chaugule, Mrinal Kumar, Nutan Limaye, Chandra Kanta Mohapatra, Adrian She, and Srikanth Srinivasan. Schur polynomials do not have small formulas if the determinant doesn't. In *Proceedings of the 35th Computational Complexity Conference*, pages 1–27, 2020. 6

[CKS19] Chi-Ning Chou, Mrinal Kumar, and Noam Solomon. Closure results for polynomial factorization. *Theory of Computing*, 15(1):1–34, 2019. 6

[CR88] Benny Chor and Ronald L Rivest. A knapsack-type public key cryptosystem based on arithmetic in finite fields. *IEEE Transactions on Information Theory*, 34(5):901–909, 1988. 2

[DdO14] Zeev Dvir and Rafael Mendes de Oliveira. Factors of Sparse Polynomials are Sparse, 2014. Pre-print available at `arXiv:1404.4834`. 6, 7, 8, 10, 11

[DDS21] Pranjal Dutta, Prateek Dwivedi, and Nitin Saxena. Deterministic identity testing paradigms for bounded top-fanin depth-4 circuits. In *36th Conference on Computational Complexity (CCC 2021)*, volume 5, page 9, 2021. 27

[DL77] Richard A DeMillo and Richard J Lipton. A Probabilistic Remark on Algebraic Program Testing. Technical report, Georgia Inst of Tech Atlanta School of Information and Computer science, 1977. 2, 26

[DSS18] Pranjal Dutta, Nitin Saxena, and Amit Sinhababu. Discovering the roots: Uniform closure results for algebraic classes under factoring. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 1152–1165, 2018. 6

[DW20] Anuj Dawar and Gregory Wilsenach. Symmetric Arithmetic Circuits. In *47th International Colloquium on Automata, Languages, and Programming (ICALP 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020. 6

[FLMS15] Hervé Fournier, Nutan Limaye, Meena Mahajan, and Srikanth Srinivasan. The shifted partial derivative complexity of elementary symmetric polynomials. In *International Symposium on Mathematical Foundations of Computer Science*, pages 324–335. Springer, 2015. 6

[GS98] V. Guruswami and M. Sudan. Improved decoding of Reed-Solomon and algebraic-geometric codes. In *Proceedings 39th Annual Symposium on Foundations of Computer Science (Cat. No.98CB36280)*, pages 28–37, 1998. 2

[HY11] Pavel Hrubeš and Amir Yehudayoff. Homogeneous formulas and symmetric polynomials. *Computational Complexity*, 20(3):559–578, 2011. 6

[Kal86] Erich Kaltofen. Uniform closure properties of p-computable functions. In *Proceedings of the eighteenth annual ACM symposium on Theory of computing*, pages 330–337, 1986. 6

[Kal87] Erich Kaltofen. Single-factor Hensel lifting and its application to the straight-line complexity of certain polynomials. In *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, pages 443–452, 1987. 6

[Kal89] Erich Kaltofen. Factorization of Polynomials Given by Straight-Line Programs. *Adv. Comput. Res.*, 5:375–412, 1989. 2, 6

[Kal03] Erich Kaltofen. Polynomial factorization: a success story. In *Proceedings of the 2003 international symposium on Symbolic and algebraic computation*, pages 3–4, 2003. 2

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

[KS01] Adam R Klivans and Daniel Spielman. Randomness efficient identity testing of multivariate polynomials. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 216–223, 2001. 27, 28, 29

[KSS14] Swastik Kopparty, Shubhangi Saraf, and Amir Shpilka. Equivalence of polynomial identity testing and deterministic multivariate polynomial factorization. In *2014 IEEE 29th Conference on Computational Complexity (CCC)*, pages 169–180. IEEE, 2014. 2, 28

[KT90] Erich Kaltofen and Barry M Trager. Computing with polynomials given byblack boxes for their evaluations: Greatest common divisors, factorization, separation of numerators and denominators. *Journal of Symbolic Computation*, 9(3):301–320, 1990. 2

[LST21] Nutan Limaye, Srikanth Srinivasan, and Sébastien Tavenas. Superpolynomial lower bounds against low-depth algebraic circuits. In *FOCS'21*, 2021. 5

[Oli16] Rafael Oliveira. Factors of low individual degree polynomials. *computational complexity*, 25(2):507–561, 2016. 6

[Ore22] Øystein Ore. Über höhere kongruenzen. *Norsk Mat. Forenings Skrifter*, 1(7):15, 1922. 2, 26

[Ost21] AM Ostrowski. Über die Bedeutung der Theorie der konvexen Polyeder für die formale Algebra. *Jahresberichte Deutsche Math. Verein*, 20:98–99, 1921. (English translated version republished in ACM SIGSAM Bulletin, 33(1):5, 1999). 7, 11

[PS21] Shir Peleg and Amir Shpilka. Polynomial time deterministic identity testing algorithm for Σ [3] $\Pi\Sigma\Pi$ [2] circuits via Edelstein–Kelly type theorem for quadratic polynomials. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 259–271, 2021. 27

[Sch80] Jacob T Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM (JACM)*, 27(4):701–717, 1980. 2, 26

[Sch00] Andrzej Schinzel. *Polynomials with special regard to reducibility*, volume 77. Cambridge University Press, 2000. 10

[Shp02] Amir Shpilka. Affine projections of symmetric polynomials. *Journal of Computer and System Sciences*, 65(4):639–659, 2002. 6

[ST20] Amit Sinhababu and Thomas Thierauf. Factorization of polynomials given by arithmetic branching programs. In *35th Computational Complexity Conference (CCC 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020. 6

[Sud97] Madhu Sudan. Decoding of Reed Solomon codes beyond the error-correction bound. *Journal of complexity*, 13(1):180–193, 1997. 2

[Sud98] Madhu Sudan. Algebra and Computation. Lecture notes, 1998. 2

[SV09] Amir Shpilka and Ilya Volkovich. Improved polynomial identity testing for read-once formulas. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 700–713. Springer, 2009. 26

[SV10] Amir Shpilka and Ilya Volkovich. On the relation between polynomial identity testing and finding variable disjoint factors. In *International Colloquium on Automata, Languages, and Programming*, pages 408–419. Springer, 2010. 5, 21

[SY10] Amir Shpilka and Amir Yehudayoff. *Arithmetic circuits: A survey of recent results and open questions*. Now Publishers Inc, 2010. 25

[Vol15] Ilya Volkovich. Deterministically factoring sparse polynomials into multilinear factors and sums of univariate polynomials. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2015)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2015. 5

[Vol17] Ilya Volkovich. On some computations on sparse polynomials. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2017)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2017. 3, 5, 21

[vzG06] Joachim von zur Gathen. Who was who in polynomial factorization: 1. In *Proceedings of the 2006 international symposium on Symbolic and algebraic computation*, page 2, 2006. 2

[vzGG13] Joachim von zur Gathen and Jürgen Gerhard. *Modern computer algebra*. Cambridge university press, 2013. 2

[vzGK85] Joachim von zur Gathen and Erich Kaltofen. Factoring sparse multivariate polynomials. *Journal of Computer and System Sciences*, 31(2):265–287, 1985. 3, 5, 19

[Zie12] Günter M Ziegler. *Lectures on polytopes*, volume 152. Springer Science & Business Media, 2012. 10

[Zip79] Richard Zippel. Probabilistic algorithms for sparse polynomials. In *International symposium on symbolic and algebraic manipulation*, pages 216–226. Springer, 1979. 2, 26

A Basics of algebraic complexity

Below, we define various algebraic models for computation of a polynomial. For details, we refer the reader to the excellent survey of [SY10].

Algebraic computational models: An *algebraic circuit* is a directed acyclic graph where computation is done bottom-up, with input leaves at the bottom and a single output node at top. The leaves are labeled with variables or field constants while rest of the nodes are either addition or multiplication nodes. The directed edges $u \to v$ are labeled with field constants, which get multiplied to the polynomial computed at node u before feeding it to node v. The in-degree of a node

is called its *fan-in* and out-degree is called *fan-out*. *Size* of the circuit is simply size of the directed graph. *Depth* of the circuit is length of the longest path from a leaf to the output node. *Degree* of the circuit is maximum degree of a polynomial computed at any node in the circuit.

A size s *depth-2* $\Sigma\Pi$ *circuit* computes a sum of s-many monomials. Thus, depth-2 circuits compute the class of sparse polynomials. The much more general class of poly(n)-sized and poly(n) degree algebraic circuits is called VP, which is considered the algebraic analog of complexity class P. The class VNP is considered the algebraic analog of complexity class NP. It is the class of polynomials which can be expressed as an exponential sum of a projection of a VP circuit family.

An algebraic circuit where fan-out of every node is one is called an *algebraic formula*. The class of polynomial sized formulas is called VF.

An *algebraic branching program (ABP)* is defined using a layered directed graph with a unique source and sink vertex. Each edge is directed from one layer to the next and has a linear polynomial as its weight. The weight of a path is product of edge weights along the path. The polynomial computed by the ABP is then simply the sum of all weighted paths from source to sink. The *length* of an ABP is the length of the longest path from source to sink and *width* of an ABP is the maximum possible number of vertices in a layer. The *size* of ABP is the product of its length and width. VBP is the class of all polynomial sized ABPs.

Hitting set and generators: In blackbox PIT for a class of n-variate polynomials $\mathcal{C}$, we are asked to provide a set $\mathcal{H} \in \mathbb{F}^n$ such that for any non-zero $f \in \mathcal{C}$, there exists at least one point $\boldsymbol{\alpha} \in \mathcal{H}$ such that $f(\boldsymbol{\alpha}) \neq 0$. Such a set $\mathcal{H}$ is called *hitting-set* for class $\mathcal{C}$.

In literature, we also have the notion of a hitting set generator (HSG) or simply generator in short, which is equivalent to a hitting set but is easier to work with PIT algorithms, especially when using or composing a known PIT algorithm into your own PIT.

Definition A.1 (Generator)**.** *Let $\mathcal{C}$ be a class of n-variate polynomials. Consider $\Phi = (f_1, f_2, \ldots, f_n)$, a tuple of k-variate polynomials where for each $i \in [n]$, $f_i \in \mathbb{F}[t_1, t_2, \ldots, t_k]$. Let $g(x_1, \ldots, x_n)$ be an n-variate polynomial. Define $\Phi : \mathbb{F}[x_1, \ldots, x_n] \to \mathbb{F}[t_1, \ldots, t_k]$ as $\Phi(g) = g(f_1, \ldots, f_n)$. We call Φ a k-seeded generator for class $\mathcal{C}$ if for every non-zero $g \in \mathcal{C}$, $\Phi(g) \neq 0$. Degree of generator Φ is defined as* $\deg(\Phi) := \max\{\deg(f_i)\}_{i=1}^{n}$.

For poly-time PIT, k is kept constant. A generator $\Phi \in \mathbb{F}[\mathbf{t}]^n$ acts as a variable reduction map which converts an input polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ to $\Phi(f) \in \mathbb{F}[t_1, \ldots, t_k]$ such that $f = 0$ if and only if $\Phi(f) = 0$. Let D be the degree of Φ, which makes $\Phi(f)$ a polynomial of individual degree dD, where d is the individual degree of f. Thus, Φ gives us a hitting-set of size $(dD+1)^k$ by brute-force derandomization for $\Phi(f)$ using PIT Lemma [Sch80, Zip79, DL77, Ore22]. In other words, we get a poly-time blackbox PIT for f when k is constant, Φ can be designed in poly-time and its degree is also polynomially bounded. See [SV09] for more on equivalence between hitting-sets and generators.

Sometimes we have a collection of maps for a class $\mathcal{C}$ such that for a given $f \in \mathcal{C}$, one of the maps in the collection acts as generator for f. In such cases, we can *interpolate* the generators to form a single generator.

Lemma A.2 (Generator Interpolation). *Let $G = \{\Phi_1, \ldots, \Phi_k\}$ where each $\Phi_i \in \mathbb{F}[t]^n$. Suppose G is a set of candidate generators for a class of n-variate polynomials $\mathcal{C}$ such that for any non-zero $f \in \mathcal{C}$, there exists $i \in [k]$, $\Phi_i(f) \neq 0$. Then, there exists a single generator $\Psi \in \mathbb{F}[t, y]^n$ such that for every non-zero $f \in \mathcal{P}$, $\Psi(f) \neq 0$. Moreover, $\deg_t(\Psi) = \max\{\deg(\Phi)\}_{\Phi \in G}$ and $\deg_y(\Psi) = |G| - 1 = k - 1$.*

See [BS21, Lemma 2.6] for a proof of this folklore trick. Below, we mention the sparse PIT map of [KS01] which gives efficient deterministic blackbox PIT for the class of sparse polynomials. We state the version presented in [BS21, Lemma 2.9].

Lemma A.3 (Sparse HSG; [KS01]). *Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ be a non-zero polynomial of individual degree at most d, such that $\text{sparsity}(f) \leq m$. Let p be a prime larger than $\max(d, mn + 1)$. Then, there is some $k \in [mn + 1]$ such that the univariate polynomial $f'(y) := f(y, y^{k^1 \bmod p}, \ldots, y^{k^{n-1} \bmod p})$ is non-zero. This yields an HSG $\Psi \in \mathbb{F}[t]^n$ for the class of m-sparse polynomials such that $\deg(\Psi) = \text{poly}(m, n, d)$.*

B PIT for special depth-4 circuits

As another application of the sparsity bound for factors of a symmetric polynomial, we get an efficient blackbox PIT for a special case of depth-4 $\Sigma\Pi\Sigma\Pi$ circuits. A size s, $\Sigma\Pi\Sigma\Pi$ circuit computes a polynomial of the form $f = \sum_{i=1}^{k} \prod_{j=1}^{m} f_{ij}$, where f_{ij} are s-sparse polynomials for each $i \in [k], j \in [m]$. This is a very important model to solve for PIT as it is known that a poly-time blackbox PIT for depth-4 circuits implies a quasi-polynomial time blackbox PIT for general circuits [AV08, AGS19]. Various restricted versions of this model have been attacked and in particular even the model of depth-4 circuit with constant top and bottom fan-in is highly non-trivial. We have a poly-time blackbox PIT when top fan-in is 3 and bottom fan-in is 2, i.e. $k = 3$ and f_{ij}'s are quadratic polynomials [PS21]. Recently, [DDS21] gave a quasi-polynomial time blackbox PIT for any constant top and bottom fan-in depth-4 circuits.

In Corollary 4 below, we give a poly-time blackbox PIT for depth-4 circuits with top fan-in $k = 2$ where the sparse polynomials f_{ij}'s are symmetric and of constant individual degree. Note that constant individual degree is a much weaker restriction than constant total degree restriction arising in the constant bottom fan-in model. In the constant individual degree restriction, the total degree of f_{ij}'s can be as large as $O(n)$. However, we do require these sparse polynomials to be symmetric. We also note that our PIT algorithm works for any field, while works of [PS21, DDS21] do have certain field restrictions.

For the PIT below, we exploit the classical tool of *resultant*. Suppose $u, v \in \mathbb{F}[x_1, \ldots, x_n, y]$ are two polynomials which are factors of two distinct sparse polynomials say g and h respectively. We

will use the fact that $\text{res}_y(u,v) \neq 0$ if and only if $\gcd_y(u,v)$ is a polynomial with y-degree zero. In other words, when $\text{res}_y(u,v) \neq 0$, then u and v do not share a common factor having variable y in its support. In general when g and h are not symmetric, we do not have a polynomial bound for sparsities of their factors u and v and therefore $\text{res}_y(u,v)$ may not be sparse. However, in our setting g and h will be sparse as well as symmetric polynomials, thus their factors u and v will also be sparse in the constant individual degree regime, because of our Theorem 1. Therefore, their resultant polynomial will also be sparse in the symmetric setting. See [BSV20, Lemma 2.8] or [KSS14] for a quick recap on resultants and its sparsity bound.

Corollary 4. *Let $\mathcal{C}$ be the class of polynomials of the form $f = \prod_{i=1}^{r} g_i + \prod_{j=1}^{m} h_j$, where $g_i, h_j \in \mathbb{F}[\mathbf{x}]$ are s-sparse* symmetric *polynomials with individual degree d, for each $i \in [r]$ and $j \in [m]$. Then, there is a deterministic black-box PIT algorithm for $\mathcal{C}$ that runs in $(s^{\text{poly}(d)} \cdot \text{poly}(dn))$-time.*

Proof. We wish to design a generator for f such that $f = 0$ if and only if $\Phi(f) = 0$. Note that for any map acting on variables, $f = 0 \Rightarrow \Phi(f) = 0$. Thus, we wish to design Φ such that $f \neq 0 \Rightarrow \Phi(f) \neq 0$.

If $f \neq 0$, then $\prod_{i=1}^{r} g_i \neq -\prod_{j=1}^{m} h_j$. Consider the complete factorization of both the sides $\prod_{i=1}^{r} g_i$ and $\prod_{j=1}^{m} h_j$, to get:

$$\prod_{i=1}^{k} u_i^{e_i} \neq -\prod_{j=1}^{t} v_j^{f_j}, \tag{B.1}$$

where u_i, v_j's are irreducible polynomials, $e_i, f_j \geq 1$ for each $i \in [k], j \in [t]$ for some $k \geq r$ and $t \geq m$. Then observe that either: (1) there exists some $i \in [k]$ such that $u_i \neq v_j$ for all $j \in [t]$, or (2) there exists some $j \in [t]$ such that $v_j \neq u_i$ for all $i \in [k]$, or (3) we have that $k = t$ with $u_i = v_i$ for each $i \in [k]$ (after reordering). The first two cases are similar and we will only show how to handle the first and last case. In the last case, find the first $i \in [k]$ such that $e_i \neq f_i$. Without loss of generality, suppose $e_i > f_i$. Then we can factor out $u_i^{f_i}$ from both sides and we will be left with only $u_i^{e_i - f_i}$ in LHS of Equation (B.1), such that $u_i \neq u_j$ for all $j \in [k]$ on RHS. Now we use Theorem 1 to exploit the fact that u_i is $s^{\text{poly}(d)}$-sparse since it is a factor of some s-sparse and symmetric polynomial. Therefore, we can use generator of [KS01] with appropriate parameters to preserve non-zeroness of u_i or equivalently $u_i^{f_i}$. We can then combine this with generator for $f/(u_i^{f_i})$ (which we design below) by interpolation of generators. See Lemma A.3 and Lemma A.2 for technical details.

Now, let us safely assume that there exists some $i \in [k]$ such that $u_i \neq v_j$ for all $j \in [t]$. Since u_i and v_j's are irreducible polynomials, this means $\gcd(u_i, v_j) = 1$ for all $j \in [t]$. Let x_l be a variable in $\text{supp}(u_i)$. First we will design a generator $\Phi : \mathbb{F}[x_1, \ldots, x_n] \to \mathbb{F}[y, x_l]$ which preserves co-primality of u_i with each v_j, with respect to variable x_l. We will later show that this is sufficient to preserve non-zeroness of f.

Using Theorem 1, observe that both u_i and v_j are $s^{\text{poly}(d)}$-sparse as they are factors of some s-sparse symmetric polynomials g_a and h_b respectively, for some $a \in [r]$ and $b \in [m]$. Now consider

the resultant polynomial $\mathrm{res}_{x_l}(u_i, v_j)$ which is a $(n-1)$-variate polynomial, free from variable x_l. Since $\gcd(u_i, v_j) = 1$, this resultant polynomial is non-zero. Moreover, it is an $(2ds)^{\mathrm{poly}(d)}$-sparse polynomial with individual degree at most $2d^2$ using [BSV20, Lemma 2.8]. Thus, we can design a generator Φ using Lemma A.3 that preserves non-zeroness of the $(n-1)$-variate resultant polynomial, i.e. $\Phi(\mathrm{res}_{x_l}(u_i, v_j)) = \mathrm{res}_{x_l}(\Phi(u_i), \Phi(v_j)) \neq 0$. Therefore, $\gcd(\Phi(u_i), \Phi(v_j)) = c$ for any $j \in [t]$, where c is some polynomial which is free of variable x_l, i.e. $\deg_{x_l}(c) = 0$. By uniqueness of factorization, $\Phi(u_i)$ has an irreducible factor with variable x_l in its support, such that it does not divide $\Phi(v_j)$, for any $j \in [t]$. This is because Φ left variable x_l untouched and mapped rest of the variables while preserving non-zeroness of resultant. Thus, the existence of such an irreducible factor of u_i which does not divide any v_j is a certificate for inequality of both sides, which gives:

$$\prod_{i=1}^{k} \Phi(u_i)^{e_i} \neq -\prod_{j=1}^{t} \Phi(v_j)^{f_j} . \tag{B.2}$$

Using this, we can prove non-zeroness of $\Phi(f)$ as follows:

$$\begin{aligned}
\Phi(f) &= \Phi\left(\prod_{i=1}^{r} g_i + \prod_{j=1}^{m} h_j\right) \\
&= \prod_{i=1}^{r} \Phi(g_i) + \prod_{j=1}^{m} \Phi(h_j) \\
&= \prod_{i=1}^{k} \Phi(u_i)^{e_i} + \prod_{j=1}^{t} \Phi(v_j)^{f_j} \neq 0 .
\end{aligned}$$

The second and third step follow because a variable map Φ is a ring homomorphism, i.e. $\Phi(a \cdot b) = \Phi(a) \cdot \Phi(b)$ and $\Phi(a + b) = \Phi(a) + \Phi(b)$, for any two polynomials a, b. Moreover, generator Φ is efficient as it is generator of [KS01] for $s^{\mathrm{poly}(d)}$-sparse polynomials of degree $\leq dn$. Thus, $\deg(\Phi) = s^{\mathrm{poly}(d)} \cdot \mathrm{poly}(dn)$ using Lemma A.3, which also gives an efficient blackbox PIT algorithm for f in the same time complexity. □