Derandomization via Symmetric Polytopes: Poly-Time Factorization of Certain Sparse Polynomials

Pranav Bisht
Department of Computer Science & Engineering, IIT Kanpur, India

Nitin Saxena
Department of Computer Science & Engineering, IIT Kanpur, India

Abstract

More than three decades ago, after a series of results, Kaltofen and Trager (J. Symb. Comput. 1990) designed a randomized polynomial time algorithm for factorization of multivariate circuits. Derandomizing this algorithm, even for restricted circuit classes, is an important open problem. In particular, the case of s-sparse polynomials, having individual degree $d = O(1)$, is very well-studied (Shpilka, Volkovich ICALP'10; Volkovich RANDOM'17; Bhargava, Saraf and Volkovich FOCS'18, JACM'20). We give a complete derandomization for this class assuming that the input is a *symmetric* polynomial over rationals. Generally, we prove an $s^{\mathrm{poly}(d)}$-sparsity bound for the factors of symmetric polynomials over any field. This characterizes the known worst-case examples of sparsity blow-up for sparse polynomial factoring.

To factor f, we use techniques from convex geometry and exploit symmetry (only) in the Newton polytope of f. We prove a crucial result about convex polytopes, by introducing the concept of "low min-entropy", which might also be of independent interest.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory; Theory of computation → Pseudorandomness and derandomization; Mathematics of computing → Combinatoric problems

Keywords and phrases Multivariate polynomial factorization, derandomization, sparse polynomials, symmetric polynomials, factor-sparsity, convex polytopes

Digital Object Identifier 10.4230/LIPIcs.FSTTCS.2022.9

Related Version *Full Version*:
`https://www.cse.iitk.ac.in/users/nitin/papers/symmetricSparse.pdf`

Funding *Nitin Saxena*: DST-SERB (CRG/2020/000045) and N. Rama Rao Chair.

Acknowledgements Pranav thanks Ilya Volkovich and Vishwas Bhargava for useful discussions. The authors would also like to thank the anonymous reviewers for their useful comments that improved the presentation of results.

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. In a famous line of work, it was shown that this problem admits a polynomial time randomized algorithm [18, 20]. Finding an efficient *deterministic* algorithm continues to be a challenging open problem[1].

[1] We only want an efficient deterministic algorithm to reduce multivariate circuit (or even sparse) factorization to univariate factorization.

See [19, 38, 39, 35] for a detailed exposition. Polynomial factorization also has interesting connections with other problems such as circuit lower bounds [15], list decoding [34, 13] and cryptography [6].

The representation of the input polynomial is significant in the complexity of this problem. The randomized algorithms of [18, 20] are efficient when both the input polynomial and output factors are represented as general algebraic circuits (see Appendix B for definition). It is imperative to ask if derandomizing polynomial factorization will be easier for restricted circuit classes. One may believe the problem to be easier since the input polynomial is weak as it belongs to a restricted class. However, note that the demand is correspondingly stronger; we want output factors also to be given in this restricted class. A very natural interesting class of algebraic circuits, which is considered for this problem, is that of *sparse* polynomials. 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. If we give sparse polynomial as input to Kaltofen's factorization algorithm, the output will be in the form of general algebraic circuit, but in *sparse polynomial factorization*, we want output factors also to be in sparse representation.

There is another motivation for studying sparse polynomials for the factorization problem, which arises from another fundamental problem called the *Polynomial identity testing* (PIT). Given an input polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ as an algebraic circuit, PIT asks to determine if f is identically zero. This problem also admits a simple and efficient randomized algorithm due to PIT Lemma [29, 42, 10, 26] and like factoring, finding a deterministic efficient algorithm is still open. It is easy to show that PIT reduces to factorization[2]. In fact, [22] 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 [22]. The classes, which admit an efficient deterministic PIT algorithm are particularly interesting because if we show the equivalence for these classes, we also derandomize factoring for them. Sparse polynomials is one such natural class, for which we do have poly-time deterministic PIT algorithms [21].

Note that the run-time of sparse factorization algorithm will be lower bounded by the sparsity of factors, just for writing the output alone. In general, the sparse model is not closed under factors, i.e. sparse polynomials can have dense factors as shown in Examples 4.8 and 4.9. Therefore, the bounded *individual degree* setting is studied for sparse polynomial factorization [31, 36, 37, 2]. Individual degree of a polynomial is defined as the maximum degree of some variable appearing in it (See Section 2). This raises the following important question:

▶ **Question 1.1.** *Let f be an s-sparse polynomial with constant individual degree. Can f be deterministically factored in* $\mathrm{poly}(s)$*-time?*

We make progress towards a positive answer to this question by giving a $\mathrm{poly}(s)$-time deterministic factoring algorithm, when the sparse polynomial f is also symmetric. Symmetric polynomials are ones that remain the same even if any of the variables are interchanged (See Section 2 for formal definition). To the best of our knowledge, symmetry was not exploited before our work and the previous best deterministic algorithm for this class was a super-polynomial time algorithm by [2] with run-time $s^{O(\log n)}$, for general s-sparse polynomials.

[2] Observe that the polynomial $x^2 + y \cdot f$ factorizes if and only if $f = 0$.

The work of [2] partially derandomizes the factoring question for sparse polynomials having constant individual degree. The crucial derandomization step in their algorithm is proving an efficient sparsity bound for factors of the input polynomial f, which dominates the run-time of their algorithm. They were able to prove an $s^{O(\log n)}$ sparsity bound for factors of f. Our main contribution is to prove factor-sparsity bound of $s^{O(1)}$, when f is symmetric. The key combinatorial object associated with derandomization of sparse factoring or equivalently factor-sparsity, is the *Newton polytope* of f. We prove our bound by giving a completely new line of attack on the symmetric Newton polytope of f.

Symmetric polynomials are studied extensively both in 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 are known to have dense factors (Example 4.8 and Example 4.9) 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. Moreover, our proof techniques do *not* require symmetry in the coefficients of f, but merely in the support of f. Besides symmetric polynomials, our techniques also give polynomial-time deterministic factoring algorithm for another class of polynomials, which we call *low min-entropy* polynomials, that we define later.

1.1 Our results

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 co-prime, irreducible polynomials and $e_1, \ldots, e_k$ are positive integers. This representation is unique up to a reordering of f_i's. Let $\mathbb{Q}$ denote the field of rational numbers. We get a poly-time deterministic reduction from multivariate to univariate factoring over any field $\mathbb{F}$. In particular, this gives us a complete poly-time factoring algorithm over $\mathbb{Q}$ below, by using the univariate factoring algorithm of [23].

▶ **Theorem 1.2** (Symmetric factoring algorithm). *Let $f \in \mathbb{Q}[x_1, \ldots, x_n]$ be an s-sparse, symmetric polynomial with constant individual degree d. Let t be the maximum bit-complexity of the coefficients of f. Then, there is a deterministic algorithm that computes the complete factorization of f in at most* $\text{poly}(s, n, t)$*-time.*

▶ Remark.

1. The exact complexity is $\text{poly}(s^{d^7 \log d} \cdot n^{d^2} \cdot t)$-time.
2. For a finite field $\mathbb{F} = \mathbb{F}_{p^\ell}$, we get a deterministic $\text{poly}(s^{d^7 \log d} \cdot n^{d^2} \cdot \ell \log p)$-time reduction to univariate factoring.
3. Throughout this paper (specifically in Theorems 1.2, 1.3 & 4.6), we get the same results if we replace symmetric polynomials with a more general class of polynomials, which we call *symmetric-support* polynomials. Let 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.

The factoring algorithm in Theorem 1.2 is a direct consequence of our new sparsity bound below.

▶ **Theorem 1.3** (Symmetric sparsity bound). *Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ be an s-sparse, symmetric polynomial with constant individual degree d, over any field $\mathbb{F}$. Then, every factor of f has its sparsity bounded by* $\mathrm{poly}(s)$.

▶ Remark.

1. The exact factor-sparsity bound that we prove is $s^{O(d^2 \log d)}$. From Example 4.9, we note that s^d is a lower bound on factor-sparsity and hence also a lower bound on the time-complexity of any factoring algorithm for sparse polynomials.
2. Previous best sparsity upper bound was $s^{O(d^2 \log n)}$ due to [2], 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 4.6. This yields a surprising *incompressibility result*: a symmetric bounded-individual-degree polynomial g, that is s-dense, has $s^{\Omega(1)}$-dense multiples gh, for an arbitrary nonzero polynomial h! (See Remark 4.5)

We now give a factoring algorithm for another class of polynomials, which we call "low min-entropy" polynomials. We say that an n-dimensional vector $\mathbf{v}$ is of *δ-min-entropy* if $(n - \delta)$ of its coordinates have the same value, where we consider minimum δ across all distinct elements in $\mathbf{v}$. We call a set of vectors A, a *δ-min-entropy set*, if every vector in A has min-entropy $\leq \delta$. We can identify the support of polynomial f, with the subset $\mathrm{supp(f)} \subseteq \{0, 1, \ldots, d\}^n$, as the set of exponent vectors corresponding to each monomial in f. We call f a *δ-min-entropy polynomial*, if supp(f) is of δ-min-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-min-entropy polynomial (also, 3-min-entropy but not 1-min-entropy). Moreover, it does *not* have symmetric-support. See Section 2 for more details on δ-min-entropy polynomials.

▶ **Theorem 1.4** (Low min-entropy factoring algorithm). *Let $f \in \mathbb{Q}[x_1, \ldots, x_n]$ be a δ-min-entropy polynomial with individual degree d, for some constants δ, d. Let t be the maximum bit-complexity of the coefficients of f. Then, there is a deterministic algorithm that computes the complete factorization of f in at most* $\mathrm{poly}(n, t)$*-time.*

▶ Remark. The exact complexity is $\mathrm{poly}((nd)^{d^4\delta} \cdot t)$-time. For a finite field $\mathbb{F} = \mathbb{F}_{p^\ell}$, we get a deterministic $\mathrm{poly}((nd)^{d^4\delta} \cdot \ell \log p)$-time reduction to univariate factoring.

The factoring algorithm in Theorem 1.4 is also a direct consequence of the following sparsity bound for factors of δ-min-entropy polynomials.

▶ **Theorem 1.5** (Factors of low min-entropy polynomials). *Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ be a δ-min-entropy polynomial with individual degree d over any field $\mathbb{F}$, such that d, δ are constants. Then, every factor of f has its sparsity bounded by* $\mathrm{poly}(n)$.

▶ Remark.

1. The exact factor-sparsity bound that we prove is $(nd)^{O(d\delta)}$, for any field $\mathbb{F}$. Further, our bound beats the $s^{O(d^2 \log n)}$ bound in [2], as long as f has min-entropy δ as low as $o(d \log n)$.
2. We do *not* claim that the factors are "low" min-entropy as well. For example, $f = (x_1 \cdots x_{n/2}) \cdot (x_{n/2+1} \cdots x_n)$ has both the factors with high min-entropy $n/2$, while f is itself of 0-min-entropy.

All the factor-sparsity results mentioned above crucially rely on our structure theorem stated below. It shows that if we have a δ-min-entropy set of exponent vectors, then the integral-points in its convex hull, have min-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 min-entropy of its vertices!

▶ **Theorem 1.6** (Polytope entropy Theorem). *Let $V \subseteq \{0, 1, \ldots, d\}^n$ be a δ-min-entropy set, then $CS(V) \cap \mathbb{Z}^n$ is a $(2d\delta)$-min-entropy set.*

1.2 Related works

The problem of sparse polynomial factorization was first studied in [40], where a randomized algorithm for the same was provided. The time complexity of this algorithm was polynomial in the sparsity of factors and exponential in the number of factors. Naturally, [40] raised the question of proving better sparsity bounds for factors of sparse polynomials. Even designing factoring algorithms for sparse polynomials of individual degree one or two required non-trivial insights. An efficient deterministic algorithm for factorization of sparse multilinear polynomials ($d = 1$) was provided in [31]. This result was further generalized to sparse polynomials that factorize into multilinear factors in [36]. The model of sparse multiquadratic polynomials ($d = 2$) was solved in [37].

The work of [2] utilized the interesting connection of factor-sparsity with Newton polytopes. They proved a sparsity bound of $s^{O(\log n)}$ for factors of s-sparse polynomials with constant individual degree. For symmetric polynomials in Theorem 1.3, we improve this bound considerably. The work of [2] also gives a deterministic factorization algorithm for sparse polynomials with constant individual degree which runs in $s^{O(\log n)}$ time. With our new poly(s)-sparsity bound, we use their algorithm to get a deterministic poly(s)-time factorization algorithm for symmetric, s-sparse polynomials, in the bounded individual degree regime.

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 [16, 17, 18] showed that if f is a size-s, degree-poly(n) algebraic *circuit*, then its factors also have poly(s, n)-sized algebraic circuits, i.e. VP is closed under factoring. [11] proved factor closure for the classes of VF, VBP, VNP with a quasi-polynomial blowup in size, and gave analogous whitebox algorithms (see Appendix B for definitions of these classes). VNP was shown to be properly closed under factoring (with only poly blowup in size) in [7]. Recently, [33] showed that size-s ABPs have factors of size poly(s), thus proving factor closure for VBP class.

For algebraic formulas, [25] 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 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 [25] showed that factors of bounded individual degree sparse polynomials can be computed by depth-7 formulas. However, [2] 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 even be computed in depth-2 efficiently, with only a polynomial blowup in size.

Another motivation for studying polynomial factorization of special classes is that improving factoring methods often lead to improvements in blackbox PIT. For eg. [24] used factoring of special circuits to give the first subexponential PIT for constant-depth circuits.

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. A new model of depth-2 symmetric circuits was defined in [30], which studies

the complexity of determinant for this model. The complexity of elementary symmetric polynomials in the algebraic formula model is studied in [14] while [5] study a class of symmetric polynomials called Schur polynomials, also in the formula model. A strong lower bound for elementary symmetric polynomials in the model of depth-4 algebraic circuits with bounded bottom fan-in is proved in [12]. The complexity of symmetric polynomials with respect to its expression in terms of elementary symmetric polynomials is studied in [3]. A new class of algebraic circuits called symmetric algebraic circuits is studied in [8], which shows a separation of determinant from permanent in this model. Recently in [9], under a more stringent symmetry restriction in this model, lower bounds for even the determinant polynomial are shown. In this work, symmetric polynomials are studied in the context of sparse polynomial factorization.

1.3 Proof Techniques

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: $\text{supp(f)} \triangleq \{(e_1, \ldots, e_n) \mid c_{e_1,\ldots,e_n} \neq 0\}$. 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 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: $|V(A + B)| \geq \max\{|V(A)|, |V(B)|\}$.

A classical fact about Newton polytopes, first observed by [27] 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 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:

$$\|f\| \geq |V(P_f)| \geq |V(P_g)|. \tag{1}$$

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) 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).

Polytope entropy Theorem

We introduce the notion of δ-min-entropy sets and polynomials in this work. In our structure theorem (Theorem 1.6), we analyze how min-entropy behaves with respect to polytopes. Suppose we are given a δ-min-entropy set V of exponent vectors from $\{0, 1, \ldots, d\}^n$. The motivating question is that if all vertices (elements of V) have low min-entropy, then how large can be the min-entropy of internal points in the polytope? In this theorem, we show that min-entropy blows up only by a factor of $O(d)$ for the internal *integral-points*, which is a small factor in the bounded individual degree regime.

In order to prove this, we first design a set of symmetric hyperplane equations in Section 3 such that for each hyperplane, the entire set V lies on one side of it. Then by convexity, one can show that every point in its convex-span also lies on the same side. Secondly,

the hyperplane equations are so designed that any *integral*-point "enclosed" within these hyperplanes must have min-entropy at most $O(d\delta)$. We present this designing of "certifying" hyperplanes as a new approach for bounding sparsity of factors. Besides the factor-sparsity implications, Theorem 1.6 could be of independent interest in its own right.

Symmetric polytopes

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. If f is a symmetric polynomial, then its Newton polytope P_f will be a symmetric polytope. Further, we observe that if a monomial (or its exponent vector) is a vertex of a symmetric polytope, then all its distinct permutations are also vertices. In other words, the vertex set $V(P_f)$ is also symmetric. Moreover, if f is sparse, then cardinality of $V(P_f)$ is also bounded. Since the vertex set is symmetric, every vector in it must contain a coordinate of "very high" frequency, otherwise $V(P_f)$ will contain a lot of distinct permutations and its size will blow up (i.e. *symmetry* $\Rightarrow$ *low min-entropy*). Thus, $V(P_f)$ will have somewhat low min-entropy. Now, we can use our Theorem 1.6 to deduce that all the integral points in P_f have low min-entropy. In Section 4, we exploit this with additional counting arguments to prove that total number of integral points in this symmetric polytope P_f is few. In particular, we show that $|P_f \cap \mathbb{Z}^n| \leq |V(P_f)|^{O(d^2 \log d)}$, where d is the individual degree of f. If f factorizes as $f = gh$, then we have $P_f = P_g + P_h$. This means that P_f contains a translated copy of P_g. Moreover, $\mathrm{supp(g)} \subseteq P_g$ contains only integral points and therefore, $\|g\| \leq |P_f \cap \mathbb{Z}^n| \leq |V(P_f)|^{O(d^2 \log d)}$. This implies $\|g\| \leq \|f\|^{O(d^2 \log d)}$, showing that any factor of a sparse, symmetric polynomial is also sparse (Theorem 1.3). Sparsity bound for factors of δ-min-entropy polynomials follows similar trajectory. Once we have these nice sparsity bounds, we derive efficient deterministic factoring algorithms in Section 5 using a result of [2] (Lemma 5.1).

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 use the symbol $\triangleq$ for definitions. 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 $\sigma \circ \mathbf{v} \triangleq (v_{\sigma(1)}, \ldots, v_{\sigma(n)})$. 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 [2, Prop. 3.2]). We refer the readers to [41, 28] 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 $P_f \triangleq CS(\text{supp(f)})$. 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** ([27]). *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** ([2]). *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 $H \triangleq \{\mathbf{y} \in \mathbb{R}^n \mid \mathbf{a}^\intercal \mathbf{y} = b\}$, 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! (See Lemma A.1)

δ-min-entropy

We say that a vector $\mathbf{v} \in \{0, 1, \ldots, d\}^n$ is of δ-min-entropy if a single value $c \in \{0, \ldots, d\}$ appears in exactly $(n - \delta)$ coordinates of $\mathbf{v}$. We consider c with the highest frequency in $\mathbf{v}$ and hence the minimum δ. In coding theory language, δ-min-entropy vector $\mathbf{v}$ has *Hamming distance* δ from the constant vector $(c, \ldots, c)$. We also extend this definition of min-entropy to sets and polynomials. We call $A \subseteq \{0, 1, \ldots, d\}^n$ a δ-min-entropy set, if for every vector $\mathbf{v} \in A$, $\mathbf{v}$ has min-entropy $\leq \delta$. We call f a δ-min-entropy polynomial, if its support set

supp(f) is of δ-min-entropy. Example 4.9 gives a polynomial f with 1-min-entropy. In contrast, consider $f = x_1 \cdots x_{n/3} + x_{n/3+1} \cdots x_n$, which is a polynomial of $(n/3)$-min-entropy but it's not $(< n/3)$-min-entropy.

3 Polytope entropy – Theorem 1.6

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 1.6. We are given V_f to be a δ-min-entropy set. Let $m \triangleq n - \delta$. We first design a hyperplane $\mathbf{u}^\intercal \cdot \mathbf{y} + d\delta = 0$ that will help us prove very useful properties in Lemma 3.1 and Lemma 3.2 below. Define column vector $\mathbf{u} \in \{-1, +1\}^n$ such that,

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

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

▶ **Lemma 3.1** (Hyperplane cover). *Let $V \subseteq \{0, 1, \ldots, d\}^n$ be a δ-min-entropy set and $\mathbf{u}$ be as defined in (2). Then, every point $\mathbf{y} \in V$ satisfies each of the following symmetric inequalities:*

$$(\sigma \circ \mathbf{u})^\intercal \cdot \mathbf{y} + d\delta \;\geq\; 0, \quad \textit{for each } \sigma \in S_n\,. \tag{3}$$

Proof. Let $\mathbf{y}$ be any $\leq \delta$-min-entropy point from V; so assume that $\mathbf{y}$ has some majority element i with frequency $\geq m = n - \delta$, where $i \in \{0, \ldots, d\}$. Let $\sigma \in S_n$ be any permutation. Define $\ell \triangleq (\sigma \circ \mathbf{u})^\intercal \cdot \mathbf{y}$. It suffices to show that $\ell \geq -d\delta$. We claim that 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 corresponding to $+1$ and remaining $(m/2)$ coordinates corresponding to -1, the minimum value of ℓ in this case is $(\delta + m/2) \cdot i - (m/2 - \delta) \cdot i - (\delta) \cdot d \;=\; (2i - d)\delta \;\geq -d\delta$, since $i \geq 0$. Thus, $\ell \geq -d\delta$ in this configuration.

We now show why this is an optimal configuration to minimize ℓ. Note that the majority element i will always have a non-negative contribution in ℓ, since its frequency in the positions corresponding to $+1$ will always be greater than or equal to its frequency in the positions corresponding to -1. This is because there are only $m/2$ positions corresponding to -1 and i has frequency $\geq m$. Therefore to minimize its contribution, majority element i should be fixed to 0. To get lowest possible value from the remaining elements in $\mathbf{y}$, we should set all of them to d and place them in any δ positions corresponding to -1. This will give minimum value of ℓ to be $-d\delta$. Rest of the configurations can only produce higher values for ℓ. Thus, $\ell + d\delta \geq 0$ for each $\mathbf{y} \in V$ and for every $\sigma \in S_n$. ◀

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.

▶ **Lemma 3.2** (Integral-point in the cover). *Let $\mathbf{u}$ be as defined in (2). 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 $(2d\delta)$-min-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

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

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

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 $2d\delta$-min-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

$$\|\mathbf{y}_+\|_1 \leq \ (i-1) \cdot (\delta + m/2 - p_+) \ + \ i \cdot (p_+) \ = \ i\delta + im/2 - \delta - m/2 + p_+ \ . \tag{4}$$

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

$$\|\mathbf{y}_-\|_1 \geq \ i \cdot (p_-) \ + \ (i+1) \cdot (m/2 - p_-) \ = \ im/2 + m/2 - p_- \ . \tag{5}$$

Let $l \triangleq \mathbf{u}^\intercal \cdot \mathbf{y} + d\delta$. By hypothesis, $l \geq 0$. Then using (4) and (5) 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 $(2d\delta)$-min-entropy point; finishing the proof. (We note that the calculations in Equation (4) and Equation (5) 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 $\mathbf{y}$.) ◀

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. We prove this lemma below.

▶ **Lemma 3.3** (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 for some $(\mathbf{a}, b) \in \mathbb{R}^n \times \mathbb{R}$ and for each $i \in [k]$, $\mathbf{a}^\intercal \cdot \mathbf{v}_i + b \geq 0$. Then, for any $\mathbf{v} \in CS(V)$, $\mathbf{a}^\intercal \cdot \mathbf{v} + b \geq 0$.*

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,

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

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 our structure theorem, which is restated below.

▶ **Theorem 1.6** (Polytope entropy Theorem). *Let $V \subseteq \{0, 1, \ldots, d\}^n$ be a δ-min-entropy set, then $CS(V) \cap \mathbb{Z}^n$ is a $(2d\delta)$-min-entropy set.*

Proof. In Lemma 3.1, 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 3.3, $(\sigma \circ \mathbf{u})^\intercal \cdot \mathbf{y} + d\delta \geq 0$, 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 Lemma 3.2, as $\pi \circ \mathbf{y}$ is sorted, we deduce that $\pi \circ \mathbf{y}$ is a $(2d\delta)$-min-entropy vector. Since min-entropy of a vector does not change on permuting it, we deduce that $\mathbf{y}$ is also a $(2d\delta)$-min-entropy point. ◀

▶ Remark. Note that Theorem 1.6 is fairly tight, i.e. a blow-up in min-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 min-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 min-entropy exactly d, for $d \leq n/2$. Therefore, $CS(V) \cap \mathbb{Z}^n$ is of min-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 1.3 & 1.5

We refer the reader to Appendix A for some basic observations that will be used for the results in this section.

4.1 Polytope bounds

Using a simple counting argument below, we bound the total number of points in a given δ-min-entropy set of integral points.

▶ **Lemma 4.1** (min-entropy sparsity upper bound). *Let $V \subseteq \{0, 1, \ldots, d\}^n$ be a δ-min-entropy set. Then, $|V| \leq \binom{n}{\delta} \cdot (d+1)^{\delta+1}$.*

Proof. Any $\mathbf{v} \in V$ has min-entropy $\leq \delta$ and thus has atleast $n - \delta$ coordinates that are equal. To specify $\mathbf{v}$, one needs to specify the indices of these repeated coordinates ($\leq \binom{n}{\delta}$ possibilities) and specify a total of $\delta + 1$ values for all the coordinates of $\mathbf{v}$ ($d + 1$ possibilities for each). Hence, we deduce that $|V| \leq \binom{n}{\delta} \cdot (d+1)^{\delta+1}$. ◀

We now show that a symmetric set of integral points is of somewhat low min-entropy.

▶ **Lemma 4.2** (symmetry ⇒ low min-entropy). *Let $V \subseteq \{0,1,\ldots,d\}^n$ be any symmetric set. Then V is a δ-min-entropy set, for some $\delta < 2d \cdot \log|V|$.*

Proof. Consider the smallest possible δ, for which V can be a δ-min-entropy set. By pigeonhole principle, for any point in V, there exists a coordinate-value with frequency $\geq n/(d+1)$. Therefore,

$$\delta \;\leq\; n - \frac{n}{d+1} \;=\; n\left(1 - \frac{1}{d+1}\right). \tag{6}$$

Since δ is chosen to be the minimum possible, a non-empty V contains a vector $\mathbf{v}$ having some integral value with maximum-frequency exactly $n-\delta$. Since V is symmetric, it must also contain all the distinct S_n-permutations of this vector $\mathbf{v}$, which are at least $\binom{n}{\delta}$ many. This implies that $|V| \geq \binom{n}{\delta}$ and further using Lemma A.3, we get that $|V| \geq \left(\frac{n}{\delta}\right)^{\delta}$. Then,

$$\begin{aligned}
\log|V| \;&\geq\; \delta \cdot \log\left(\frac{n}{\delta}\right) \;\geq\; \delta \cdot \log\left(\frac{n}{n\left(1-\frac{1}{d+1}\right)}\right) \quad \text{[Using (6)]} \\
&=\; \delta \cdot \log\left(1+\frac{1}{d}\right) \;>\; \delta \cdot \left(\frac{1}{d} - \frac{1}{2d^2}\right) \quad \text{[Using Lemma A.3]} \\
&\geq\; \frac{\delta}{2d} \quad \text{[As } d \geq 1\text{]}\,.
\end{aligned}$$

This proves that $\delta < 2d \cdot \log|V|$. ◀

▶ Remark. We note that if we are promised $\delta \leq n/2$, then in the proof of Lemma 4.2, we get $|V| \geq \binom{n}{\delta} \geq 2^{\delta}$ which implies $\delta \leq \log|V|$. This will lead to an improved bound of $|V|^{O(d\log d)}$ in Theorem 4.4. In general however, δ can be higher than $n/2$. Consider for example the set of all S_n-permutations of the string $0^{n/3} \cdot 1^{n/3} \cdot 2^{n/3}$, which has $\delta = n - n/3 = 2n/3$.

We now show that convex span of a low min-entropy set has low number of integral points.

▶ **Theorem 4.3** (Low min-entropy polytope count). *Let $V \subseteq \{0,1,\ldots,d\}^n$ be a δ-min-entropy set. Then, $|CS(V) \cap \mathbb{Z}^n| \leq \binom{n}{\delta}^{2d} \cdot (d+1)^{2d\delta+1}$.*

Proof. Theorem 1.6 proves that $CS(V)\cap\mathbb{Z}^n$ is a $(2d\delta)$-min-entropy set. Note that $CS(V)\cap\mathbb{Z}^n$ is also a subset of $\{0,\ldots,d\}^n$, since V is. Then by Lemma 4.1, $|CS(V)\cap\mathbb{Z}^n| \leq \binom{n}{2d\delta}\cdot(d+1)^{2d\delta+1}$. The conclusion then follows from Lemma A.3 by observing $\binom{n}{2d\delta} \leq \binom{n}{\delta}^{2d}$. ◀

Using the counting arguments above, we now show that the gap between number of vertices and total number of integral points in a symmetric polytope, is low.

▶ **Theorem 4.4** (Symmetric polytope count). *Let $V \subseteq \{0,1,\ldots,d\}^n$ be the vertices of a symmetric polytope P. Then, $|P \cap \mathbb{Z}^n| \leq |V|^{O(d^2\log d))}$.*

Proof. Consider the smallest possible δ, for which V can be a δ-min-entropy set. Since P is a symmetric polytope, its vertex set V is also symmetric by Lemma A.2. By Lemma 4.2, the min-entropy of V is at most $\delta \triangleq O(d \cdot \log|V|)$. Then by Theorem 4.3, $|CS(V)\cap\mathbb{Z}^n| \leq \binom{n}{\delta}^{2d}\cdot(d+1)^{2d\delta+1}$. Observe that $(d+1)^{2d\delta+1} \leq d^{O(d^2\log|V|)} = |V|^{O(d^2\log d)}$. In Lemma 4.2, we also proved that $|V| \geq \binom{n}{\delta}$, since V is symmetric. Thus, $\binom{n}{\delta}^{2d} \leq |V|^{2d}$ and we get the desired conclusion. ◀

▶ Remark 4.5. We also prove that a symmetric polytope formed by taking convex hull of a symmetric subset E of $\{0,1,\ldots,d\}^n$ must have *many* vertices (corner points), at least $|E|^{\Omega(1/(d^2\log d))}$-many, to be precise. We get this *incompressibility result* by substituting $P = CS(E)$ in Theorem 4.4 and observing that $E \subseteq P \cap \mathbb{Z}^n$.

4.2 Factor-sparsity bounds

We start by proving the sparsity bound for a symmetric factor g of some sparse polynomial f, which may itself not be symmetric. To get a poly(s) sparsity bound, we only require individual degree of g to be constant, while f may have arbitrary individual degree.

▶ **Theorem 4.6** (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 P_g be the Newton polytope of g and V_g be its vertex set. Since g is symmetric, P_g is a symmetric polytope. Now invoke Theorem 4.4 with $P = P_g$ to note that $|P_g \cap \mathbb{Z}^n| \leq |V_g|^{O(d^2\log d)}$. Observe that supp(g) $\subseteq P_g \cap \mathbb{Z}^n$ and $|V_g| \leq |V_f| \leq s$. Therefore, $\|g\| \leq s^{O(d^2\log d)}$. ◀

▶ Remark. In Theorem 4.6 (resp. Theorem 4.4), we do not require the whole of g (resp. P) to be symmetric, rather we only need its vertex set V_g (resp. V) 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 1.3 but only V_f to be symmetric. In these cases, we will not require use of Lemma A.2.

We will be needing the following observation for our main proofs below.

▶ **Observation 4.7.** *Let $f,g,h \in \mathbb{F}[x_1,\ldots,x_n]$ be polynomials such that $f = g\cdot h$. Let P_f be the Newton polytope of f. Then, $\|g\| \leq |P_f \cap \mathbb{Z}^n|$.*

Proof. By Minkowski sum property, we have that $P_f = P_g + P_h$. For every $\mathbf{u} \in$ supp(g), consider a fixed point $\mathbf{v} \in$ supp(h). Observe that $\mathbf{u}+\mathbf{v} \in P_f$, since $\mathbf{u} \in P_g$ and $\mathbf{v} \in P_h$. Moreover, since the support vectors are integral, $\mathbf{u}+\mathbf{v} \in P_f \cap \mathbb{Z}^n$. In other words, $P_f \cap \mathbb{Z}^n$ contains a translated copy of supp(g). This means that $\|g\| \leq |P_f \cap \mathbb{Z}^n|$. ◀

We have sufficient tools now to prove our main theorems below.

▶ **Theorem 1.3** (Symmetric sparsity bound). *Let $f \in \mathbb{F}[x_1,\ldots,x_n]$ be an s-sparse, symmetric polynomial with constant individual degree d, over any field $\mathbb{F}$. Then, every factor of f has its sparsity bounded by* poly(s).

Proof. Since f is symmetric, its Newton polytope P_f will also be symmetric. Let V_f be the vertex set of P_f. Then, invoke Theorem 4.4 with $P = P_f$ to deduce that $|P_f \cap \mathbb{Z}^n| \leq |V_f|^{O(d^2\log d)}$. The conclusion then follows from Observation 4.7 and $|V_f| \leq s$. ◀

We note that Theorem 1.3 does not follow from Theorem 4.6 as a symmetric polynomial f may not have symmetric factors. 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 also note that factors of a low min-entropy polynomial might have *high* min-entropy. But we still get a nice sparsity bound for them below.

▶ **Theorem 1.5** (Factors of low min-entropy polynomials). *Let $f \in \mathbb{F}[x_1,\ldots,x_n]$ be a δ-min-entropy polynomial with individual degree d over any field $\mathbb{F}$, such that d,δ are constants. Then, every factor of f has its sparsity bounded by* poly(n).

Proof. Let P_f be the Newton polytope of f and V_f be its vertex set. Since f is of δ-min-entropy, so is V_f, as $V_f \subseteq \text{supp(f)}$. Then by Theorem 4.3, we deduce that $|P_f \cap \mathbb{Z}^n| \leq \binom{n}{\delta}^{2d} \cdot (d+1)^{2d\delta+1} \leq (nd)^{O(d\delta)}$. The conclusion then follows from Observation 4.7. ◀

▶ Remark. We note that the example in Claim 4.4 of [2] is a non-symmetric polynomial with its vertices having exactly $(n/2)$-min-entropy such that its Newton polytope has at least $n^{\Omega(\log n)}$-many integral points. Therefore for high-entropy polynomials, a new technique is required which goes beyond counting number of internal points in the Newton polytope.

4.3 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 4.8** ([40]). Consider the following polynomial f with individual degree d, which factorizes as $f = gh$, where

$$f \;=\; \prod_{i=1}^{n}(x_i^d - 1)\;, \qquad g \;=\; \prod_{i=1}^{n}(1 + x_i + \ldots + x_i^{d-1})\;, \qquad h \;=\; \prod_{i=1}^{n}(x_i - 1)\;.$$

Observe that $\|f\| = 2^n$, while $\|g\| = d^n$. If we let $s \triangleq \|f\|$, 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 1.3 shows that $\|g\| \leq s^{O(d^2 \log d)}$.

▶ **Example 4.9** ([2]). 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

$$\begin{aligned} f \;&=\; x_1^p + x_2^p + \ldots + x_n^p \;=\; (x_1 + x_2 + \ldots + x_n)^p\;, \\ g \;&=\; (x_1 + x_2 + \ldots + x_n)^d\;, \qquad h \;=\; (x_1 + x_2 + \ldots + x_n)^{p-d}\;. \end{aligned}$$

Observe that $\|f\| = n$, while $\|g\| = \binom{n+d-1}{d} \approx n^d$. If we let $s \triangleq \|f\|$, 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 1.3, which shows that $\|g\| \leq s^{O(d^2 \log d)}$. Hence, this example shows that in Theorem 1.3, we cannot do better than s^d.

Note that this f is also a low min-entropy polynomial, in fact with $\delta = 1$ as all the exponent vectors in supp(f) have $n-1$ coordinates having the same value 0. Thus, we can use Theorem 1.5 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 Theorems 1.2 & 1.4

A nice feature of the results in [2] 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** (Theorem 5.8 in [2]). *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 $(n \cdot \xi(n, d^2, s^d))^{O(d^2)} \cdot \text{poly}(c_{\mathbb{F}}(d^2))$ field operations.*

Here $c_{\mathbb{F}}(d)$ is the best known time complexity for factoring a univariate polynomial of degree d over the field $\mathbb{F}$. For $\mathbb{F} = \mathbb{Q}$, $c_{\mathbb{F}}(d) \leq \text{poly}(d, t)$ where t is the maximum bit-complexity of the coefficients of f [23]. For a finite field $\mathbb{F} = \mathbb{F}_{p^\ell}$, $c_{\mathbb{F}}(d) \leq \text{poly}(\ell \cdot p, d)$ [1, 4]. In other words, the above theorem shows a deterministic reduction from multivariate to univariate factoring over any field $\mathbb{F}$. In [2], 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 $s^{\text{poly}(d) \log n}$ time complexity, which is quasi-polynomial in the bounded individual degree setting. In this work, we deal with symmetric and constant-min-entropy input polynomials, for which we show that $\xi(n, d, s) \leq (ns)^{\text{poly}(d)}$. 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 Theorem 1.2 and Theorem 1.4 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.

▶ **Theorem 1.2** (Symmetric factoring algorithm). *Let $f \in \mathbb{Q}[x_1, \ldots, x_n]$ be an s-sparse, symmetric polynomial with constant individual degree d. Let t be the maximum bit-complexity of the coefficients of f. Then, there is a deterministic algorithm that computes the complete factorization of f in at most* $\text{poly}(s, n, t)$*-time.*

Proof. Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ be an s-sparse, symmetric polynomial with individual degree d. In Theorem 1.3, 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:

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

This time complexity is $\text{poly}(s, n, t)$ over the field of rationals $\mathbb{Q}$, when d is a constant. ◀

▶ **Theorem 1.4** (Low min-entropy factoring algorithm). *Let $f \in \mathbb{Q}[x_1, \ldots, x_n]$ be a δ-min-entropy polynomial with individual degree d, for some constants δ, d. Let t be the maximum bit-complexity of the coefficients of f. Then, there is a deterministic algorithm that computes the complete factorization of f in at most* $\text{poly}(n, t)$*-time.*

Proof. Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ be a δ-min-entropy polynomial with individual degree d. In Theorem 1.3, 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:

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

This time complexity is $\text{poly}(n, t)$ over the field of rationals $\mathbb{Q}$, when d, δ are constants. ◀

6 Future directions

In this work, we show a positive evidence for resolving Question 1.1 by solving it for symmetric and constant min-entropy polynomials with bounded individual degree. As discussed, the constant individual degree restriction here is necessary for sparse factorization (Examples 4.8, 4.9). We leave it as an open question to study factors of sparse polynomials with high min-entropy. For example, can one show an $(snd)^{\mathrm{poly}(d)}$ factor-sparsity bound for s-sparse, $(\log n)$-min-entropy polynomial of individual degree d?

References

1 Elwyn R Berlekamp. Factoring polynomials over finite fields. *Bell System Technical Journal*, 46(8):1853–1859, 1967.

2 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 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.

4 David G Cantor and Hans Zassenhaus. A new algorithm for factoring polynomials over finite fields. *Mathematics of Computation*, pages 587–592, 1981.

5 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 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.

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

8 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.

9 Anuj Dawar and Gregory Wilsenach. Lower Bounds for Symmetric Circuits for the Determinant. In Mark Braverman, editor, *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*, volume 215 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 52:1–52:22, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2022.52.

10 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.

11 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.

12 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.

13 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. doi:10.1109/SFCS.1998.743426.

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

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

16 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.

17 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.

18 Erich Kaltofen. Factorization of polynomials given by straight-line programs. *Adv. Comput. Res.*, 5:375–412, 1989.

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

20 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.

21 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.

22 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.

23 Arjen K Lenstra, Hendrik Willem Lenstra, and László Lovász. Factoring polynomials with rational coefficients. *Mathematische annalen*, 261(ARTICLE):515–534, 1982.

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

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

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

27 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).

28 Andrzej Schinzel. *Polynomials with special regard to reducibility*, volume 77. Cambridge University Press, 2000.

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

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

31 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.

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

33 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.

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

35 Madhu Sudan. Algebra and computation. lecture notes, 1998.

36 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.

37 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.

38 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.

39 Joachim von zur Gathen and Jürgen Gerhard. *Modern computer algebra*. Cambridge university press, 2013.

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

41 Günter M Ziegler. *Lectures on polytopes*, volume 152. Springer Science & Business Media, 2012.

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

A Preliminary observations for Section 4

We start by proving that a symmetric polynomial will have a symmetric Newton polytope.

▶ **Lemma A.1** (Symmetric polynomials ⇒ symmetric polytopes). *Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ be a symmetric-support polynomial. Then the Newton polytope P_f of f is also symmetric.*

Proof. Let $\text{supp(f)} = \{\mathbf{v}_1, \ldots, \mathbf{v}_k\}$. Let $\mathbf{v} \in P_f$ be an arbitrary point. We need to show that $\sigma \circ \mathbf{v} \in P_f$, for every $\sigma \in S_n$. Since $P_f = CS(\text{supp(f)})$, we can express $\mathbf{v}$ as:

$$\mathbf{v} = \sum_{i=1}^{k} \alpha_i \cdot \mathbf{v}_i.$$

Then for an arbitrary $\sigma \in S_n$, observe that:

$$\sigma \circ \mathbf{v} = \sum_{i=1}^{k} \alpha_i \cdot \sigma \circ \mathbf{v}_i.$$

Now, since f is a symmetric-support polynomial, $\sigma \circ \mathbf{v}_i$ is also in supp(f), for each $i \in [k]$. The above equation then implies that $\sigma \circ \mathbf{v} \in CS(\text{supp(f)})$, which means that $\sigma \circ \mathbf{v} \in P_f$. ◀

Next, we observe that a point is a vertex in a symmetric polytope P if and only if all its S_n permutations are also vertices in P.

▶ **Lemma A.2** (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. ◀

The lemma below mentions few standard bounds that we make use of.

▶ **Lemma A.3** (Counting estimates)**.**

1. *For positive integers a, b with $a \geq b$,* $(a/b)^b \leq \binom{a}{b}$.
2. *For positive integers a, b, c with $a \geq bc$,* $\binom{a}{bc} \leq \binom{a}{c}^b$.
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), observe that derivative of f is positive for $x > 0$, where $f = \ln(1+x) - (x - \frac{x^2}{2})$, and for $x = 0$, $f(0) = 0$. ◀

B Algebraic computational models

Below, we define various algebraic models for computation of a polynomial. For details, we refer the reader to the excellent survey of [32].

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.