

When Are Monomial Symmetric Polynomials Hard?

Abstract

Symmetric polynomials play a central role in algebraic complexity theory. In this paper, we make progress toward classifying the complexity of one of the most natural families of symmetric polynomials, the *monomial symmetric polynomials*. For a partition $\lambda = (\lambda_1, \dots, \lambda_r)$ and $n \geq r$, the monomial symmetric polynomial $m_\lambda(x_1, \dots, x_n)$ is defined as

$$m_\lambda(x_1, \dots, x_n) := \sum x_{i_1}^{\lambda_1} x_{i_2}^{\lambda_2} \cdots x_{i_r}^{\lambda_r},$$

where the sum ranges over all distinct monomials obtained by assigning the parts of λ to pairwise distinct variables.

Our main result establishes an almost-complete dichotomy for the complexity of the monomial symmetric polynomials m_λ . Informally speaking, if a family of partitions $\lambda = (\lambda_1, \dots, \lambda_r)$ contains polynomially many distinct parts, then the corresponding monomial symmetric polynomials are VNP-complete under c -reductions. On the other hand, if the number of distinct parts is constant, then the family admits polynomial-size circuits. Thus the only remaining gap concerns intermediate regimes where the number of distinct parts grows, but only subpolynomially.

As part of our proof, we use combinatorial properties of matching polynomials to establish VNP-hardness of the perfect matching polynomial for *blow-up graphs*; these are graphs with non-generic edge-weights that are obtained by “blowing up” a smaller graph. This blow-up step is what lets us handle arbitrary multiplicities of equal parts in λ . We reduce these matching polynomials to monomial-symmetric polynomials by combining classical techniques from algebraic complexity theory with structural properties of Sidon sets.

1 Introduction

Symmetry is a fundamental concept in mathematics and computer science, appearing across algebra, combinatorics, and theoretical computer science. In algebra, it appears prominently with the class of *symmetric polynomials*, i.e., polynomials which are invariant under any permutation of their variables. Symmetric polynomials have been extensively studied in algebraic combinatorics and invariant theory, and they naturally arise in counting problems and algebraic computation. Understanding the computational complexity of natural families of symmetric polynomials is therefore a central question in algebraic complexity theory.

In algebraic complexity theory, the underlying complexity measure is usually the size of *arithmetic circuits* computing the polynomial. An arithmetic circuit is a directed acyclic graph whose internal nodes are labeled by addition or multiplication gates, and whose leaves are labeled by input variables or field constants. Every node (gate) in the circuit computes a polynomial in a natural way, and the output node computes the final polynomial of interest. The size of a circuit is the number of nodes in its underlying graph. The arithmetic complexity of a polynomial $f \in \mathbb{F}[x_1, \dots, x_n]$ is defined as the minimum size of an arithmetic circuit computing it, denoted by $L(f)$. We refer the reader to the surveys [Mah13, SY10, Sap15, Bür16] for background on algebraic complexity theory.

Monomial symmetric polynomials. In this paper, we investigate the circuit complexity of *monomial symmetric polynomials*. Let $\lambda = (\lambda_1, \dots, \lambda_r)$ be a partition with $\lambda_1 \geq \dots \geq \lambda_r > 0$. The monomial symmetric polynomial m_λ associated with λ in n variables is obtained by symmetrizing the monomial

$x_1^{\lambda_1} x_2^{\lambda_2} \cdots x_r^{\lambda_r}$, i.e., by taking all permutations of variables and keeping each resulting distinct monomial once:

$$m_{\lambda}(x_1, \dots, x_n) := \sum x_{i_1}^{\lambda_1} x_{i_2}^{\lambda_2} \cdots x_{i_r}^{\lambda_r},$$

where the sum ranges over all *distinct* monomials obtained by assigning the parts of λ to pairwise distinct variables; equivalently, we sum over the S_n -orbit but keep each resulting monomial only once. A monomial of the form $x_{i_1}^{\lambda_1} x_{i_2}^{\lambda_2} \cdots x_{i_r}^{\lambda_r}$ is obviously a very simple polynomial which can be computed by a small arithmetic circuit, and it is natural to ask how the arithmetic complexity is affected when a monomial is symmetrized as above.

The family of monomial symmetric polynomials forms one of the standard bases of the ring of symmetric polynomials and plays a fundamental role in algebraic combinatorics and representation theory. In an ordinary polynomial ring, monomials are the simplest basis elements. It is therefore natural, but potentially misleading, to expect the analogous basis elements in the symmetric world to remain easy. Our results show that this expectation indeed fails sharply: once the defining partition has sufficiently many distinct parts, the resulting monomial symmetric polynomial is already VNP-complete.

Classical symmetric polynomials, such as elementary and power symmetric polynomials, can be viewed as special cases of monomial symmetric polynomials for specific partitions. For instance, the k -th elementary symmetric polynomial e_k is the monomial symmetric polynomial corresponding to the partition $(1, 1, \dots, 1)$ with k parts, while the k -th power sum symmetric polynomial p_k corresponds to the partition (k) . Both elementary and power sum symmetric polynomials admit efficient arithmetic circuit representations of polynomial size. Thus the hard instances are not obtained by adding complicated local structure to the terms; rather, hardness is controlled by the global pattern of exponents in the partition. This makes monomial symmetric polynomials a natural test case for the complexity of large, structured sums: the local pattern is just a monomial, yet the symmetrized expression can still be hard, and repeated exponents start to matter.

Another natural example is the Vandermonde permanent:

$$\text{Per} \left((x_i^j)_{i,j \in [n]} \right) = m_{(n, n-1, \dots, 1)}(x_1, \dots, x_n).$$

This family has only n variables but degree $1 + 2 + \cdots + n = \binom{n+1}{2}$, placing it outside the variable-count regime used in our hardness theorem below.

Another less obvious interpretation of monomial symmetric polynomials comes from (weighted) pattern counting problems: It is known that counting *small* subgraph patterns H in large host graphs G is hard if and only if some *quotient* of H (i.e., a graph obtained from H by identifying vertices) has large treewidth [CM14, CDM17]; treewidth is a particular graph parameter that measures the tree-likeness of graphs. For $\lambda = (\lambda_1, \dots, \lambda_r)$, the monomial symmetric polynomial $m_{\lambda}(x_1, \dots, x_n)$ can be viewed as the weighted count of H_{λ} -subgraph copies in a graph G , where the pattern H_{λ} contains r isolated vertices with self-loops of multiplicities $\lambda_1, \dots, \lambda_r$, and the host graph G contains n vertices, with self-loops of edge-weights $x_1, \dots, x_n$, respectively. Edge-weights are multiplicative, so that the count becomes a polynomial in $x_1, \dots, x_n$. From a graph-theoretic perspective, the graph H_{λ} is extremely simple: It merely consists of disjoint self-loops, so the treewidth of H_{λ} and all of its quotients is at most 1. But since the weighted count is the monomial-symmetric polynomial m_{λ} and thus hard for some partitions λ , it follows that the graph-theoretic intuition from the complexity of counting small patterns can break down fundamentally once the pattern and large graph have comparable size; see also [DPS26a, DPS26b] for more general considerations of this kind in the context of algebraic complexity theory.

1.1 Complexity of Symmetrized Polynomials

Looking beyond monomial-symmetric polynomials, more general symmetric polynomials have been studied in algebraic complexity theory for decades. As a first question, one can ask whether there exist *any* families of

symmetric polynomials that require super-polynomial-size arithmetic circuits. Since we lack even superlinear circuit lower bounds, these results either need to address restricted models or require complexity-theoretic assumptions such as $VP \neq VNP$. Even then, this question is non-trivial; in the Boolean setting, the ability to take negations means that every symmetric Boolean function can be computed by a polynomial-size Boolean circuit. Specifically, every symmetric Boolean function can be expressed as a function of the Hamming weight of the input, and thus can be computed by a polynomial-size circuit. In fact, every symmetric Boolean function can be computed by a polynomial-size *threshold* circuit, placing it in the complexity class TC^0 .

Unconditional lower bounds in restricted models. A seminal result of Nisan and Wigderson [NW96] established that elementary symmetric polynomials require exponential-size homogeneous depth-three circuits over fields of characteristic zero. Their proof introduced the powerful technique of analyzing the dimension of the space of partial derivatives, now a standard tool for proving circuit lower bounds. For non-homogeneous depth-three circuits, Shpilka and Wigderson [SW99] prove a quadratic lower bound for computing elementary symmetric polynomials of linear degree, which is tight. The upper bound is due to Ben-Or, see [SW99, Thm 4.2]. Hrubes and Yehudayoff [HY11] investigate formulas of arbitrary depth. They prove that every multilinear homogeneous formula for e_k has size $k^{\Omega(\log k)}n$ and construct homogeneous formulas of matching size, which are, however, not multilinear. Fournier *et al.* [FLMS17] study the shifted partial derivative complexity of elementary symmetric polynomials, strengthening the results of Nisan and Wigderson in certain parameter regimes. Subsequent work has extended these results to other models of computation and to various families of symmetric polynomials. Despite this rich history, the complexity of monomial symmetric polynomials beyond elementary and power sum cases has remained largely open.

Conditional lower bounds from effective fundamental theorems. The question of superpolynomial lower bounds for general arithmetic circuits computing symmetric polynomials (conditional on $VP \neq VNP$) was finally settled by Bläser and Jindal [BJ19]. They constructed families of symmetric polynomials that require super-polynomial-size arithmetic circuits assuming $VP \neq VNP$. More precisely, they proved that certain families of symmetric polynomials are VNP-complete under c -reductions, which implies they cannot be computed by polynomial-size arithmetic circuits unless $VP = VNP$. Here, c -reductions are the algebraic analogue of Turing reductions in classical complexity theory, allowing for oracle access to the polynomial being reduced to, see Section A and Section 2 for a formal definition.

The main tool of Bläser and Jindal is an effective version of the fundamental theorem for symmetric polynomials (Theorem A.1) in terms of circuit complexity. Specifically, Bläser and Jindal showed that if (f_n) is a family of VNP-complete polynomials, then the polynomial family g_n defined by $g_n := f_n(e_1, e_2, \dots, e_{v(n)})$ is a family of symmetric polynomials that is VNP-complete under c -reductions, where e_i is the i -th elementary symmetric polynomial and $v(n)$ is the number of variables in f_n . Chaugule *et al.* [CKL⁺23] prove a similar result for formulas. More recently, Bhattacharjee *et al.* [BKR⁺26] proved a constant-depth analogue of this type of result and used it to obtain piecewise constant-depth algebraic circuits for univariate polynomial gcd over sufficiently large fields.

Similar techniques also apply to Schur polynomials, another basis family of symmetric polynomials that play a central role in algebraic combinatorics and representation theory. They can be defined as the determinant of a matrix with complete homogeneous symmetric polynomials as entries. Chaugule *et al.* [CKL⁺23] show that for certain families of partitions, Schur polynomials are hard for VBP in the sense that if they have algebraic formulas of polynomial size, then so does the ordinary determinant family.

Known results for monomial-symmetric polynomials. In the wake of the above results based on effective fundamental theorems for symmetric polynomials, Curticapean *et al.* [CLS22] constructed an explicit family of monomial-symmetric polynomials that are VNP-complete under c -reductions. Their hard family is

specifically constructed to enable the hardness result. For example, the chosen partition has only distinct parts, and more importantly, these parts essentially enjoy the highly non-generic property of forming a *Sidon set*, i.e., a set of integers S such that $x, y \in S$ can be recovered uniquely from $x + y$.

The result in [CLS22] was shown through a c -reduction from the VNP-complete perfect matching polynomials to monomial symmetric polynomials. The starting point is the Binet–Minc identity (Theorem 2.1), which expresses monomial symmetric polynomials in terms of power sum symmetric polynomials as

$$m_{\lambda}(x_1, x_2, \dots, x_n) = f(p_1, p_2, \dots, p_d),$$

for some “witnessing polynomial” f , where d is the degree of m_{λ} and p_i is the i -th power sum symmetric polynomial. By crucially using the Sidon properties of λ and additional technical conditions on λ , a relatively direct reduction from perfect matching polynomials to this witnessing polynomial f can be established. Then, using results from [CKL⁺23, BJ19], the witnessing polynomial $f(y_1, y_2, \dots, y_d)$ is c -reduced to the monomial-symmetric polynomial m_{λ} , giving the overall VNP-hardness result.

1.2 Our Results

In this paper, we establish an almost-complete dichotomy for the arithmetic complexity of monomial symmetric polynomials in terms of the number of distinct parts in the defining partition: At one end, a *constant* number of distinct parts leads to polynomial-size circuits. At the other end, *polynomially many* distinct parts lead to VNP-completeness under c -reductions, provided that the number of variables in m_{λ} is large enough. More formally, we prove the following VNP-hardness result for monomial symmetric polynomials; for the definition of p -families, see Section A.

Theorem 1.1. *Let $(f_n)_{n \in \mathbb{N}}$ be a p -family of monomial symmetric polynomials such that the number of distinct parts in the defining partition of f_n is at least n^{ϵ} for some $\epsilon > 0$. Then $(f_n)_{n \in \mathbb{N}}$ is VNP-complete under c -reductions, assuming the number of variables in f_n is at least the degree of f_n .*

Let us emphasize that the last condition is a genuine restriction: it excludes, for example, the Vandermonde permanent discussed above, which has degree $\binom{n+1}{2}$ but only n variables. We return to this limitation in Section 5.

For the tractable side, we also show that if the defining partition of a monomial symmetric polynomial contains only a constant number of distinct parts, then the corresponding monomial symmetric polynomial can be computed by a polynomial-size arithmetic circuit. This is a folklore result, which we state below for completeness; a proof can be found in Section B.

Theorem 1.2. *The monomial symmetric polynomial $m_{\lambda}(x_1, x_2, \dots, x_n)$ of degree d can be computed by an arithmetic circuit of size at most $\text{poly}(d) \cdot n^{O(t)}$ if the number of distinct parts in the partition λ is t .*

The remaining blind spot is the intermediate regime where the number of distinct parts grows with n but remains sub-polynomial. A similar situation occurred in the classification of *immanants*, another family of polynomials that are indexed by partitions λ and can be viewed as generalized determinants. Bürgisser [Bür00] proved the first hardness results for certain immanants, and Curticapean [Cur21] later proved a full dichotomy for immanants. Curiously, this dichotomy result *does* address regimes of sub-polynomially growing hardness parameters by using complexity assumptions from parameterized complexity theory. In our setting of monomial symmetric polynomials, both this issue and the overall hardness proof appear significantly more difficult: Immanants have matrix inputs, which directly give a very useful graph-theoretic structure that allows for a reduction from matchings, whereas hardness proofs for monomial symmetric polynomials have to recover this matching structure from additive relations among these exponents. We return to this in Section 5.

1.3 Proof Ideas

At first, our setup is similar to the result by Curticapean *et al.* [CLS22] for specifically constructed hard families of λ : We construct a c -reduction from VNP-complete perfect matching polynomials to monomial symmetric polynomials. Our starting point is again the Binet–Minc identity (Theorem 2.1) that expresses m_λ of degree d in terms of the power sum symmetric polynomials p_i , as sketched above for [CLS22]. That is,

$$m_\lambda(x_1, x_2, \dots, x_n) = f(p_1, p_2, \dots, p_d).$$

For arbitrary partitions λ with many distinct parts, two new technical issues arise:

- First, for general λ , we have no chance of controlling the additive relations among all distinct parts of λ , so we must ensure carefully that we can extract a well-behaved subset.
- Second, repeated parts are a nontrivial issue that was invisible in the previous approaches, where λ could simply be chosen to *not* have repeating parts: For our reduction from perfect matching polynomials, repeated parts have the effect of turning a generic graph into a *blow-up graph*, where each selected part μ_i contributes a block of size equal to its multiplicity w_i . Proving that the corresponding blow-up matching polynomials remain hard is a central new ingredient of the paper, and it is also a source of structured, non-generic permanent-like polynomial families.

Our proof handles both issues through three steps. First, we extract a large sum-free Sidon subset from the distinct parts of λ . Second, we use Binet–Minc together with homogeneous extraction and substitutions to force the surviving set partitions to be pairings inside this subset. Third, repeated parts are handled by interpreting these pairings as perfect matchings in a blow-up graph, and then reducing ordinary perfect matching to perfect matching in arbitrary blow-ups. We now spell this out. Suppose

$$\lambda = (\lambda_1, \lambda_2, \dots, \lambda_k) = (\mu_1^{w_1}, \mu_2^{w_2}, \dots, \mu_t^{w_t}),$$

where $\mu_1 > \mu_2 > \dots > \mu_t$ are the distinct parts of λ , and w_i is the multiplicity of μ_i in λ . The Binet–Minc identity is a Möbius inversion formula on the partition lattice of $[k]$, so monomials of f correspond to set partitions of $[k]$. From the set $\{\mu_1, \dots, \mu_t\}$ we extract a sum-free Sidon subset S of size $\Omega(\sqrt{t})$. The Sidon property makes pairwise sums of elements of S uniquely decodable, and these sums will become edge variables of a complete graph.

We then take a carefully chosen homogeneous component of f and set $y_\mu = 0$ for every $\mu \in S$. This removes singleton blocks whose labels lie in S . A counting argument forces every surviving set partition to use all elements outside S as singleton blocks, while the copies of elements in S must be paired into blocks of size two; larger blocks would leave too few blocks in total. After dividing out the forced singleton contribution from the elements outside S , the remaining monomials are exactly encoded by pairings among the copies of elements of S .

If every selected part occurs once, these pairings are perfect matchings of a complete graph on $\Omega(\sqrt{t})$ vertices, recovering the situation closest to [CLS22]. With multiplicities, however, each selected part μ_i contributes w_i indistinguishable copies, and the extracted polynomial is the perfect matching polynomial of a blow-up of the complete graph. The blow-up graph replaces vertex i by a clique of size w_i ; see Fig. 2 for an example and Section 2 for the formal definition.

It remains to show that these blow-up matching polynomials are hard for arbitrary positive multiplicities. This is the role of Section 3. There we reduce the ordinary perfect matching polynomial to the perfect matching polynomial of any blow-up of a complete graph. The reduction attaches private gadget blocks and chooses algebraic edge-weights so that, after taking the right homogeneous component, every original block is used exactly once. Thus the blow-up polynomial still contains the ordinary perfect matching polynomial in a form accessible by c -reductions.

1.4 Organization of the Paper

The remainder of the paper is organized as follows. In [Section 2](#), we introduce the necessary background and notation. In [Section 3](#), we reduce the perfect matching polynomial to its blown-up version for arbitrary multiplicities. In [Section 4](#), we reduce the blown-up perfect matching polynomial to the monomial symmetric polynomial m_λ . Finally, we conclude with a discussion of open problems in [Section 5](#). In the appendix, [Section A](#) and [Section B](#) contain background material and omitted proofs.

2 Preliminaries

Throughout this paper, unless stated otherwise, we assume that the underlying field $\mathbb{F}$ is $\mathbb{C}$.

2.1 The Ring of Symmetric Polynomials

A polynomial $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$ is called *symmetric* if, for every permutation $\sigma \in S_n$, we have

$$f(x_1, x_2, \dots, x_n) = f(x_{\sigma(1)}, x_{\sigma(2)}, \dots, x_{\sigma(n)}).$$

The ring of symmetric polynomials in n variables is a fundamental object in algebraic combinatorics and admits multiple natural bases, e.g., the *elementary symmetric polynomials* $e_k(x_1, \dots, x_n)$ for $k \in [n]$ and power sum symmetric polynomials $p_k(x_1, \dots, x_n)$:

$$e_k(x_1, \dots, x_n) := \sum_{1 \leq i_1 < i_2 < \dots < i_k \leq n} x_{i_1} x_{i_2} \cdots x_{i_k} \quad \text{and} \quad p_k(x_1, \dots, x_n) := \sum_{i=1}^n x_i^k.$$

Every symmetric polynomial can be uniquely expressed as a polynomial in the elementary symmetric polynomials, and also as a polynomial in the power sum symmetric polynomials. See [Section A.1](#) for details. For a symmetric polynomial f , we define f^{elem} and f^{pow} to be the unique polynomials in $\mathbb{F}[y_1, \dots, y_n]$ such that

$$f = f^{\text{elem}}(e_1, \dots, e_n) \quad \text{and} \quad f = f^{\text{pow}}(p_1, \dots, p_n).$$

2.2 Matching Polynomials

We refer to [Section A.2](#) for a basic introduction to algebraic complexity. We will be working with c -reductions, which are the algebraic analogue of Turing reductions from classical complexity theory. The oracle complexity of a polynomial f with respect to another polynomial g is the minimum size of an arithmetic circuit computing f that is allowed to query an oracle for g , denoted by $L^g(f)$. For more details on c -reductions, see [Section A.2](#).

Let G be an n -vertex graph whose edges are labelled by variables from a set $\mathbf{X}$ or constants from $\mathbb{F}$. For a matching M of G , we denote its weight by $\text{wt}(M)$, which is defined as the product of the weights of the edges in M . The *matching polynomial* of G , denoted $\text{Match}(G) \in \mathbb{F}[\mathbf{X}]$, is defined as

$$\text{Match}(G) := \sum_M \text{wt}(M),$$

where the sum ranges over all matchings M of G . In a similar way, the *perfect matching polynomial* of G is defined by

$$\text{PerfMatch}(G) := \sum_M \text{wt}(M),$$

where the sum ranges over all perfect matchings M of G .

Now assume that $n \in \mathbb{N}$ is even. Let $\mathbf{X}_n := \{x_{i,j} \mid 1 \leq i < j \leq n\}$ be the set of $\binom{n}{2}$ variables corresponding to the edges of the complete graph K_n . We consider the edge $(i, j) \in E(K_n)$ to be weighted by $x_{i,j}$. It is well known that

$$\text{PerfMatch}_n := \text{PerfMatch}(K_n)$$

is VNP-complete under p -projections. This follows from the VNP-completeness of the permanent ([Val79]) together with the fact that $\text{Per}(A) = \text{PerfMatch}(G)$ for the complete bipartite graph G whose edge (i, j) has weight $A_{i,j}$.

We will also consider the so-called *blow-up* matching polynomials. Let G be an edge-weighted graph on n vertices. For a vector $\mathbf{w} = (w_1, w_2, \dots, w_n)$ of positive integers, the $\mathbf{w}$ -*blow-up* of G , denoted $G^{(\mathbf{w})}$, is the edge-weighted graph obtained by replacing each vertex i with a clique of size w_i , and replacing each edge $\{i, j\}$ with a complete bipartite graph between the corresponding cliques. The edges in this bipartite graph inherit the weight of the edge $\{i, j\}$ in G . The edges inside the clique corresponding to vertex i are assigned weight z_i , where z_i is a new variable. For a vector $\mathbf{w} \in \mathbb{Z}_{>0}^n$, we define the polynomials $\text{Match}_n^{(\mathbf{w})}$ and $\text{PerfMatch}_n^{(\mathbf{w})}$ to be the matching polynomial and the perfect matching polynomial, respectively, of the blow-up graph $K_n^{(\mathbf{w})}$. Throughout the paper, blow-up vectors have positive integer entries. A zero entry would simply delete the corresponding vertex and can be removed in advance.

2.3 Monomial Symmetric Polynomials and the Rectangular Permanent

We now define the permanent of rectangular matrices. Fix a $k \times n$ matrix X where $k \leq n$ and the (i, j) -th entry of X is a variable $x_{i,j}$. For a positive integer m , we denote by $[m]$ the set $\{1, 2, \dots, m\}$. For a subset $J \subseteq [n]$ of size k , we denote by X_J the submatrix obtained by keeping only the columns indexed by J .

The *rectangular permanent* $\text{rPer}_{k,n}$ is defined by

$$\text{rPer}_{k,n}(X) = \sum_{J \in \binom{[n]}{k}} \text{Per}(X_J).$$

The following polynomial identity will play a key role in our arguments. For a positive integer k , let $\mathcal{P}_k$ denote the set of all partitions of $[k]$ into non-empty subsets.

Theorem 2.1 (Binet–Minc Identity [Min84]). *Let $\mathbb{F}$ be any field and let X be a $k \times n$ matrix as above. Then*

$$\text{rPer}_{k,n}(X) = \sum_{\mathcal{I} \in \mathcal{P}_k} (-1)^{k-|\mathcal{I}|} \prod_{I \in \mathcal{I}} (|I| - 1)! S_I \quad \text{with} \quad S_I := \sum_{j=1}^n \prod_{i \in I} x_{i,j}.$$

A partition λ of a positive integer d is a tuple of positive integers $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_k)$ such that $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_k > 0$ and $\sum_{i=1}^k \lambda_i = d$. We write $\lambda \vdash d$. For a partition λ , we define:

$$\begin{aligned} d(\lambda) &:= \sum_{i \in [k]} \lambda_i && \text{(degree),} \\ \ell(\lambda) &:= k && \text{(number of parts),} \\ D(\lambda) &:= |\{\lambda_1, \dots, \lambda_k\}| && \text{(number of distinct parts).} \end{aligned}$$

We also use a shorthand notation for partitions with repeated parts. If $\mu_1 > \mu_2 > \dots > \mu_t$ (with $t = D(\lambda)$) are the distinct parts of λ with multiplicities $w_1, w_2, \dots, w_t$ respectively, we write

$$\lambda = (\mu_1^{w_1}, \mu_2^{w_2}, \dots, \mu_t^{w_t}).$$

Recall that the monomial symmetric polynomial associated with the partition λ in n variables is denoted by

$$m_\lambda(x_1, x_2, \dots, x_n)$$

and defined as:

$$m_\lambda(x_1, x_2, \dots, x_n) := \sum x_{i_1}^{\lambda_1} x_{i_2}^{\lambda_2} \cdots x_{i_k}^{\lambda_k}.$$

Here the sum ranges over all distinct monomials obtained by assigning the parts of λ to pairwise distinct variables; equivalently, over all monomials whose nonzero exponent multiset is $\{\lambda_1, \dots, \lambda_k\}$. When the number of variables is clear from the context, we simply write m_λ . Observe that the monomial symmetric polynomial corresponding to the partition $\lambda = (\lambda_1, \dots, \lambda_k)$ is the rectangular permanent of the following $k \times n$ matrix, up to the factor caused by repeated rows:

$$\begin{pmatrix} x_1^{\lambda_1} & x_2^{\lambda_1} & \cdots & x_n^{\lambda_1} \\ x_1^{\lambda_2} & x_2^{\lambda_2} & \cdots & x_n^{\lambda_2} \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{\lambda_k} & x_2^{\lambda_k} & \cdots & x_n^{\lambda_k} \end{pmatrix}.$$

More precisely, if $\lambda = (\mu_1^{w_1}, \dots, \mu_t^{w_t})$, then this rectangular permanent is

$$\left(\prod_{j=1}^t w_j! \right) m_\lambda(x_1, \dots, x_n),$$

because permuting equal rows does not change the resulting monomial.

Applying the Binet–Minc identity yields the following corollary.

Corollary 2.2. *Let $\lambda = (\lambda_1, \dots, \lambda_k)$ be a partition of degree $d(\lambda)$, and write $\lambda = (\mu_1^{w_1}, \dots, \mu_t^{w_t})$. Put $\kappa(\lambda) := \prod_{j=1}^t w_j!$. Then*

$$m_\lambda = \frac{1}{\kappa(\lambda)} \sum_{I \in \mathcal{P}_k} (-1)^{k-|I|} \prod_{I \in \mathcal{I}} (|I| - 1)! p_{\lambda_I}, \quad (1)$$

where $\lambda_I := \sum_{i \in I} \lambda_i$.

2.4 Tools from Additive Combinatorics

To extract the perfect matching polynomial of complete graphs from the power-sum representation of m_λ , we also need the notion of Sidon sets. A set of positive integers S is called a *Sidon set* if for any four elements $a, b, c, d \in S$, the equality

$$a + b = c + d$$

implies that the unordered pairs $\{a, b\}$ and $\{c, d\}$ are equal. In other words, unordered pairwise sums are distinct. We will use the following classical result on the existence of large Sidon subsets of finite sets of positive integers.

Theorem 2.3 (Existence of large Sidon sets [Abb90]). *For every finite set $A \subseteq \mathbb{N}$ of sufficiently large cardinality, there exists a Sidon subset $S \subseteq A$ such that*

$$|S| \geq 0.08 \sqrt{|A|}.$$

We shall need a slightly stronger version of the above theorem, in the sense that we want Sidon subsets $S \subseteq A$ such that the sum of two elements of S is not equal to any element of S . This stronger version (Corollary 2.5) follows from the following result of Erdős [Erd65].

Theorem 2.4 (Erdős, [Erd65, Theorem 2]). *For every finite set A of non-zero real numbers, there exists a subset $S \subseteq A$ with $|S| \geq |A|/3$ such that*

$$a + b \neq c \quad \text{for any } a, b, c \in S.$$

Sets which satisfy the above condition are called *sum-free* sets. The above theorem states that every finite set of non-zero real numbers contains a large sum-free subset. The following corollary is an immediate consequence of Theorem 2.3 and Theorem 2.4.

Corollary 2.5. *For every finite set A of positive integers of sufficiently large cardinality, there exists a sum-free Sidon subset $S \subseteq A$ with $|S| \geq 0.04\sqrt{|A|}$. More specifically, S satisfies the following two properties:*

$$\begin{aligned} a + b = c + d &\implies \{a, b\} = \{c, d\} \quad \text{for any } a, b, c, d \in S, \\ a + b &\neq c \quad \text{for any } a, b, c \in S. \end{aligned}$$

Proof. First apply Theorem 2.4 to obtain a sum-free subset $A' \subseteq A$ of size at least $|A|/3$. If $|A|$ is sufficiently large, then so is $|A'|$, so we can apply Theorem 2.3 to A' and obtain a Sidon subset $S \subseteq A'$ with

$$|S| \geq 0.08\sqrt{|A'|} \geq \frac{0.08}{\sqrt{3}}\sqrt{|A|} > 0.04\sqrt{|A|}.$$

Since $S \subseteq A'$, the sum-free property is preserved. □

3 Hardness of Blown-Up Matching Polynomials

In this section, we reduce the multiplicity-free perfect matching polynomial to its blown-up version. This will allow us later to handle m_λ for partitions λ with repeating parts; the repetitions correspond to blow-ups.

Lemma 3.1. *Let $r \geq 4$ and $\mathbf{w} = (w_1, \dots, w_r) \in \mathbb{Z}_{>0}^r$ be such that $\sum_i w_i$ is even. For $t := 2\lfloor r/4 \rfloor$, we have*

$$\text{PerfMatch}_t \leq_c \text{PerfMatch}_r^{(\mathbf{w})}.$$

Throughout this section, fix an integer $r \geq 4$ and arbitrary vertex-multiplicities $\mathbf{w} \in \mathbb{Z}_{>0}^r$ whose total sum $n := \sum_i w_i$ is even. Without loss of generality, we assume that $\mathbf{w}$ is non-increasing. Define $t := 2\lfloor r/4 \rfloor$ as above. Our reduction proceeds along the following chain whose parts are established in Lemma 3.3 and Lemma 3.4:

$$\text{PerfMatch}_t \leq_c \text{Match}_r^{(\mathbf{w})} \leq_c \text{PerfMatch}_r^{(\mathbf{w})}. \quad (2)$$

Towards the first reduction, let us write $a_i = w_i$ and $b_i = w_{t+i}$ for $i \in [t]$. Because $\mathbf{w}$ is sorted, we have $a_i \geq b_i$ for all $i \in [t]$. We construct an edge-weighted graph G on r vertices as follows:

- Introduce a new indeterminate z and set $x_{ij} \leftarrow z \cdot x_{ij}$ for all $1 \leq i < j \leq t$.
- Set $x_{i,t+i} \leftarrow \xi_i$ for $1 \leq i \leq t$, where ξ_i will be determined later. The choice of ξ_i is crucial.
- Set $x_{i,j} \leftarrow 0$ for all $1 \leq i < j \leq r$ not covered in the previous two steps.
- Set every intra-block variable z_i of $\text{Match}_r^{(\mathbf{w})}$ to 0.

Note that only the first $2t = 4\lfloor r/4 \rfloor$ vertices are relevant, while the other (up to three) vertices are isolated in G .

We are given access to $\text{Match}(G^{(\mathbf{w})})$. For $i \in [t]$, let A_i be the vertex block (clique) in $G^{(\mathbf{w})}$ corresponding to vertex i of G , and let B_i be the block (clique) corresponding to vertex $t+i$. We will call $A_1 \cup \dots \cup A_t$ the “original” vertices of $G^{(\mathbf{w})}$ and $B_1 \cup \dots \cup B_t$ the “gadget” vertices. Note that each block A_i of original

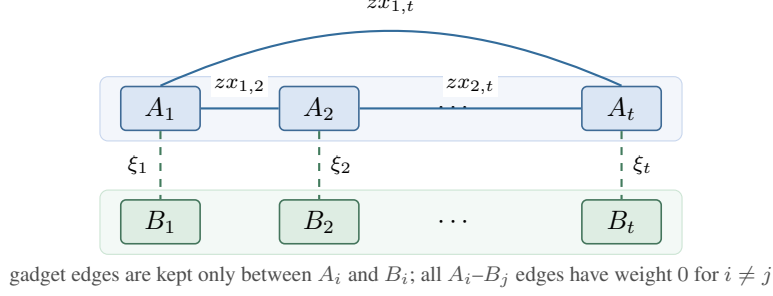


Figure 1: Schematic view of the graph used in Section 3. Each displayed block is a clique in the blow-up graph. Solid edges between original blocks represent complete bipartite edge sets of weights $zx_{i,j}$; dashed edges connect each original block A_i only to its private gadget block B_i with weight ξ_i .

vertices has a private block B_i of gadget vertices; there are no edges (i.e., zero-weight edges) between A_i and B_j for $i \neq j$.

The reduction from PerfMatch_t to $\text{Match}_r^{(w)}$ proceeds as follows: Every matching M in $G^{(w)}$ is the union of an “original matching” O that lives within the original blocks and, for each $i \in [t]$, some matching between the original block A_i and its gadget block B_i . In other words, each matching O on the original part can be extended in several ways to a matching of the overall graph $G^{(w)}$ by adding gadget edges. For each $i \in [t]$, the edge-weight ξ_i on the gadget edges between A_i and B_i is chosen carefully to ensure that the weighted count of extensions vanishes if O leaves all vertices of A_i unmatched. The existence of such an edge-weight ξ_i follows from properties of matching polynomials:

Given a graph H , let m_k for $k \in \mathbb{N}$ denote the number of matchings with k edges in H . The matching-generating polynomial of H is defined as

$$M(H, x) := \sum_k m_k x^k.$$

The following lemma uses properties of the matching polynomial of complete bipartite graphs (i.e., the graphs between original blocks and gadget blocks) to find a useful gadget edge-weight ξ_i . Due to space limitations, the proof is presented in Section C of the appendix.

Lemma 3.2. *For all integers $1 \leq b \leq a$, there is $\xi = \xi(a, b) \in \mathbb{C}$ such that*

$$M(K_{a,b}, \xi) = 0, \tag{3}$$

$$M(K_{a-1,b}, \xi) \neq 0. \tag{4}$$

For nonuniform c -reductions, the existence of the constants $\xi(a, b)$ is enough. Moreover, these constants can be chosen with controlled algebraic complexity: $\xi(a, b)$ is a root of

$$M(K_{a,b}, x) = \sum_{k=0}^b \binom{a}{k} \binom{b}{k} k! x^k,$$

so it has algebraic degree at most b over $\mathbb{Q}$, and the defining polynomial has integer coefficients whose bit lengths are polynomially bounded in a and b .

Throughout our construction, we choose $\xi_i = \xi(a_i, b_i)$ for $i \in [t]$ via Lemma 3.2. If we then count matchings M in $G^{(w)}$ with exactly $t/2$ original edges, then no original block A_i can be entirely avoided by the original edges: If all vertices of A_i were avoided, then the extensions between the $a = |A_i|$ original and

$b = |B_i|$ gadget vertices amount to $M(K_{a,b}, \xi_i) = 0$ by (3). This implies that each original block is hit at least once, and since only $t/2$ edges are available, exactly once. The contributions from gadget edges do not vanish due to (4). This allows us to show that matchings M in $G^{(\mathbf{w})}$ correspond to perfect matchings in K_t in a well-controlled way.

Lemma 3.3. *For $r \geq 4$ and $\mathbf{w} \in \mathbb{Z}_{>0}^r$ with even $n = \sum_i w_i$, we have $\text{PerfMatch}_t \leq_c \text{Match}_r^{(\mathbf{w})}$ with $t = 2\lfloor r/4 \rfloor$.*

Proof. As above, set $a_i := w_i$ and $b_i := w_{t+i}$ for $i \in [t]$, and define blocks A_i, B_i . Let $\xi_i := \xi(a_i, b_i)$ be the values obtained from Lemma 3.2, and abbreviate

$$\eta_i := M(K_{a_i-1, b_i}, \xi_i) \neq 0.$$

We construct the graph G as above with edge-weight ξ_i on all edges between A_i and B_i , and we set all intra-block variables to 0. To obtain the required c -reduction, it suffices to show that

$$\text{PerfMatch}_t = \frac{[z^{t/2}] \text{Match}(G^{(\mathbf{w})})}{\prod_{i=1}^t a_i \eta_i}. \quad (5)$$

We establish (5) in the remainder of the proof. Let $\mathcal{O}$ be the set of matchings with exactly $t/2$ edges in the graph $G^{(\mathbf{w})}[A_1 \cup \dots \cup A_t]$ induced by original blocks. We call these the *original matchings*. For every original matching $O \in \mathcal{O}$, define its *type* θ_O to be the map with domain $[t]$ that maps $i \mapsto |V(O) \cap A_i|$. An original matching $O \in \mathcal{O}$ is a *lift* if $\theta_O(i) = 1$ for all $i \in [t]$. For an original edge e between blocks A_i and A_j , let $\bar{w}(e) := x_{ij}$ be its weight after removing the auxiliary factor z .

Each lift corresponds to a perfect matching in K_t by consolidating each block A_i of $G^{(\mathbf{w})}$ to a single vertex i in K_t . Conversely, each perfect matching in K_t corresponds to $\prod_{i=1}^t a_i$ lifts in $G^{(\mathbf{w})}$ by independently choosing, for each block i , one of a_i vertices in block A_i . The correspondence preserves the weights $\bar{w}(e)$ of the original edges, so the weighted count of lifts equals that of perfect matchings in K_t , up to the factor $\prod_{i=1}^t a_i$.

In view of this, we aim at restricting the matchings counted in $[z^{t/2}] \text{Match}(G^{(\mathbf{w})})$ to lifts. Every matching M in $G^{(\mathbf{w})}$ of z -degree $t/2$ has the form $M = O \cup F$, where $O \in \mathcal{O}$ is an original matching and F is a matching on gadget edges. For $O \in \mathcal{O}$, let $\mathcal{E}_O$ be the set of matchings on gadget edges that extend O to a matching M in $G^{(\mathbf{w})}$. As the sets $\mathcal{E}_O$ for $O \in \mathcal{O}$ partition the matchings of z -degree $t/2$, we obtain

$$[z^{t/2}] \text{Match}(G^{(\mathbf{w})}) = \sum_{O \in \mathcal{O}} \prod_{e \in O} \bar{w}(e) \underbrace{\sum_{F \in \mathcal{E}_O} \prod_{e \in F} w(e)}_{=q(O)}. \quad (6)$$

We show that the gadget extensions $q(O)$ restrict the summation to lifts O . To abbreviate, let us write $\theta := \theta_O$ and write Q_i^θ for the complete bipartite graph with $a_i - \theta(i)$ left and b_i right vertices. We observe that

$$q(O) = \prod_{i=1}^t M(Q_i^\theta, \xi_i) = \begin{cases} 0 & O \text{ is not a lift,} \\ \prod_{i=1}^t \eta_i & O \text{ is a lift.} \end{cases} \quad (7)$$

For the first equation, note that the sum defining $q(O)$ in (6) ranges over all matchings in the disjoint union of complete bipartite graphs $Q_1^\theta \cup \dots \cup Q_t^\theta$, where Q_i^θ has weight ξ_i on all edges. For the second equation:

- If O is not a lift, we show that $q(O) = 0$. Indeed, if O is not a lift, then $\theta(i) = 0$ for some $i \in [t]$, because O is a matching with $t/2$ edges by definition, so an index j with $\theta(j) > 1$ forces some index i with $\theta(i) = 0$. Then $Q_i^\theta \cong K_{a_i, b_i}$ and our choice of ξ_i by Lemma 3.2 ensures that the i -th factor in (7) is $M(Q_i^\theta, \xi_i) = 0$, which overall implies $q(O) = 0$.

- If O is a lift, then $Q_i^\theta \cong K_{a_i-1, b_i}$ for all i . This means that the factors in (7) for $i \in [t]$ are $M(Q_i^\theta, \xi_i) = \eta_i \neq 0$.

Substituting (7) into (6), we obtain

$$[z^{t/2}] \text{Match} \left(G^{(\mathbf{w})} \right) = \left(\prod_{i=1}^t \eta_i \right) \sum_{O \in \mathcal{O}} \prod_{\text{lift } e \in O} \bar{w}(e) = \left(\prod_{i=1}^t a_i \eta_i \right) \text{PerfMatch}_t, \quad (8)$$

where the second equation follows from the observation made at the beginning of the proof that every perfect matching in K_t corresponds to $\prod_i a_i$ lifts $O \in \mathcal{O}$ of the same $\bar{w}$ -weight. This proves (5) and concludes the proof. $\square$

Next, we reduce from the blown-up matching polynomial to the blown-up perfect matching polynomial. This proof is standard: We substitute each blow-up edge variable $\theta \leftarrow \lambda\theta + 1$ and interpolate on λ . The details are given in the appendix.

Lemma 3.4. *For $\mathbf{w} \in \mathbb{Z}_{>0}^r$ with even $n = \sum_i w_i$, we have $\text{Match}_r^{(\mathbf{w})} \leq_c \text{PerfMatch}_r^{(\mathbf{w})}$.*

The proof of Lemma 3.1 is now immediate:

Proof of Lemma 3.1. The proof follows by combining Lemma 3.3 and Lemma 3.4 with the transitivity of c -reductions, as outlined in (2). $\square$

4 Blown-Up Matching Polynomial to Monomial Symmetric Polynomials

In this section, we extend ideas from [CLS22] to prove VNP-completeness of a large class of monomial symmetric polynomial families under c -reductions: if the defining partitions contain sufficiently many distinct parts and the number of variables is at least the degree, then the corresponding family of monomial symmetric polynomials is VNP-complete under c -reductions.

For a partition λ with degree $d(\lambda)$, define m_λ^{pow} to be the unique polynomial in the power-sum variables $y_1, \dots, y_{d(\lambda)}$ such that, when the number of variables is at least $d(\lambda)$,

$$m_\lambda(x_1, \dots, x_v) = m_\lambda^{\text{pow}}(p_1(x), \dots, p_{d(\lambda)}(x)).$$

We now state the key lemma underlying our main result.

Lemma 4.1. *There is an absolute constant $\alpha > 0$ such that the following holds for all sufficiently large t . Suppose $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_k)$ is a partition with $D(\lambda) = t$. Then there exist $r > \alpha\sqrt{t}$ and a vector $\mathbf{w} \in \mathbb{Z}_{>0}^r$ with even $\|\mathbf{w}\|_1$ such that*

$$\text{PerfMatch}_r^{(\mathbf{w})} \leq_c m_\lambda(x_1, \dots, x_v)$$

for all $v \geq d(\lambda)$.

Proof. Write the partition λ as $\lambda = (\mu_1^{w_1}, \mu_2^{w_2}, \dots, \mu_t^{w_t})$, where $\mu_1 > \mu_2 > \dots > \mu_t > 0$ are the distinct parts of λ and w_i denotes the multiplicity of μ_i for each $i \in [t]$. By Corollary 2.5, there exists a sum-free Sidon subset $S_0 \subseteq \{\mu_1, \mu_2, \dots, \mu_t\}$ of size at least $0.04\sqrt{t}$. If necessary, remove one element from S_0 so that the total multiplicity of the remaining set is even, and call the resulting set S . For all sufficiently large t , this leaves $r := |S| > \alpha\sqrt{t}$ for a fixed constant $\alpha > 0$. Suppose we have

$$S = \{\mu_{i_1}, \mu_{i_2}, \dots, \mu_{i_r}\} \text{ for some } 1 \leq i_1 < i_2 < \dots < i_r \leq t.$$

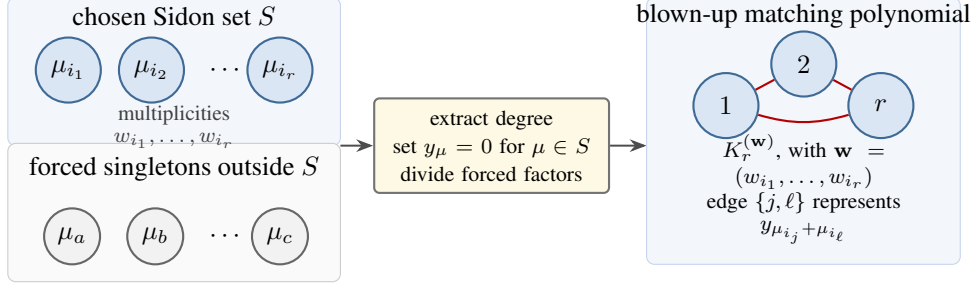


Figure 2: Schematic view of [Lemma 4.1](#). After the homogeneous extraction and substitutions, elements outside the Sidon set S contribute forced singleton factors, while copies of elements in S must be paired. Because pairwise sums in S are uniquely decodable, these pairings are exactly perfect matchings in the blow-up graph $K_r^{(\mathbf{w})}$.

$W := w_{i_1} + \dots + w_{i_r}$ is even. By [Lemma A.6](#), we have

$$L^{m_{\lambda}(x_1, \dots, x_v)}(m_{\lambda}^{\text{pow}}) \leq \tilde{O}(v^4(d(\lambda))^3). \quad (9)$$

For brevity, let $\mathbf{y} = \{y_1, y_2, \dots, y_{d(\lambda)}\}$. By definition of m_{λ}^{pow} and [Corollary 2.2](#), we have:

$$m_{\lambda}^{\text{pow}}(\mathbf{y}) = \frac{1}{\kappa(\lambda)} \sum_{\mathcal{I} \in \mathcal{P}_k} (-1)^{k-|\mathcal{I}|} \prod_{I \in \mathcal{I}} (|I| - 1)! y_{\lambda_I},$$

where $\lambda_I = \sum_{i \in I} \lambda_i$ for any nonempty $I \subseteq [k]$. We consider the homogeneous component of degree $k - W/2$ in m_{λ}^{pow} and we substitute $y_{\mu_{i_1}} = y_{\mu_{i_2}} = \dots = y_{\mu_{i_r}} = 0$ in this homogeneous component to obtain a new polynomial $g(\mathbf{y})$. Note that since S is a sum-free Sidon set, this substitution cannot make any variable of the form $y_{\mu_{i_j} + \mu_{i_\ell}}$ vanish in $g(\mathbf{y})$. By [Lemma A.7](#) we obtain:

$$L^{m_{\lambda}^{\text{pow}}}(g(\mathbf{y})) \leq O(d(\lambda)^2). \quad (10)$$

Each monomial of m_{λ}^{pow} corresponds to a partition $\mathcal{I}$ of $[k]$, and the degree of this monomial is exactly $|\mathcal{I}|$. Thus a partition $\mathcal{I}$ contributes to $g(\mathbf{y})$ only if $|\mathcal{I}| = k - W/2$. Such a partition must contain at least $k - W$ singleton parts. Since we set

$$y_{\mu_{i_1}} = y_{\mu_{i_2}} = \dots = y_{\mu_{i_r}} = 0,$$

singletons corresponding to $\mu_{i_1}, \dots, \mu_{i_r}$ do not contribute. Hence every contributing partition $\mathcal{I}$ must contain exactly $k - W$ singleton parts corresponding to μ_j with $j \in [t] \setminus \{i_1, i_2, \dots, i_r\}$.

After these $k - W$ forced singletons are fixed, the remaining W elements are arranged into $W/2$ non-singleton parts. Since every non-singleton part has size at least two, all of these parts have size exactly two. Thus every remaining part pairs two elements from $S = \{\mu_{i_1}, \dots, \mu_{i_r}\}$, with μ_{i_j} appearing exactly w_{i_j} times across these pairs. Because $S = \{\mu_{i_1}, \dots, \mu_{i_r}\}$ is a sum-free Sidon set, all sums

$$\mu_{i_j} + \mu_{i_\ell} \quad (1 \leq j \leq \ell \leq r)$$

are distinct. Moreover, since S is sum-free, none of these sums is equal to any element of S , so the substitution $y_{\mu_{i_1}} = \dots = y_{\mu_{i_r}} = 0$ does not delete a pair variable. It might still happen that $\mu_{i_j} + \mu_{i_\ell} = \mu_q$ for some $q \in [t] \setminus \{i_1, i_2, \dots, i_r\}$. But all copies of μ_q outside S occur as forced singletons, hence

$$\prod_{\substack{j \in [t] \\ j \notin \{i_1, i_2, \dots, i_r\}}} y_{\mu_j}^{w_j} \text{ divides } g(\mathbf{y}).$$

Hence, using [Lemma A.8](#), we can divide $g(\mathbf{y})$ by this monomial to obtain a new polynomial $h(\mathbf{y})$ such that:

$$L^{m_{\lambda}^{\text{pow}}}(h(\mathbf{y})) \leq \text{poly}(d(\lambda)). \quad (11)$$

Now, in $h(\mathbf{y})$, we perform the following substitutions:

- Each variable $y_{\mu_{i_j} + \mu_{i_k}}$ is substituted by a variable $x_{j,k}$ for $1 \leq j < k \leq r$.
- Each variable $y_{2\mu_{i_j}}$ is substituted by a new variable z_j for all $j \in [r]$.

This yields a polynomial H whose variables are precisely the variables $x_{i,j}$ for $1 \leq i < j \leq r$ and z_j for $j \in [r]$. From [Eq. \(11\)](#) it follows immediately that

$$L^{m_{\lambda}^{\text{pow}}}(H) \leq \text{poly}(d(\lambda)).$$

Combining [Eq. \(9\)](#), [Eq. \(11\)](#), and [Lemma A.5](#), we obtain

$$L^{m_{\lambda}(x_1, \dots, x_v)}(H) \leq \text{poly}(v, d(\lambda)).$$

Finally, observe that each monomial of H corresponds to a partition $\mathcal{I}$ of $[k]$ with exactly $W/2$ parts of size two arising from elements in S with μ_{i_j} appearing w_{i_j} many times. Hence each monomial of H corresponds to a perfect matching of the blow-up graph $K_r^{(\mathbf{w})}$, and vice versa. See [Fig. 2](#) for a visualization. Therefore the polynomial H is a nonzero scalar multiple of $\text{PerfMatch}_r^{(\mathbf{w})}$, where $\mathbf{w} = (w_{i_1}, w_{i_2}, \dots, w_{i_r})$; the scalar is $(-1)^{W/2}/\kappa(\lambda)$. Since we work over $\mathbb{C}$, this scalar can be inverted in the reduction. Hence $L^{m_{\lambda}(x_1, \dots, x_v)}(\text{PerfMatch}_r^{(\mathbf{w})})$ is polynomially bounded in v and $d(\lambda)$, which proves the lemma. $\square$

To prove our main result claimed in [Theorem 1.1](#), we use the results proved in [Section 3](#).

Proof of Theorem 1.1. Let $(f_n)_{n \in \mathbb{N}}$ be a p -family of monomial symmetric polynomials. Write

$$f_n = m_{\lambda_n}(x_1, \dots, x_{v_n})$$

for some partition λ_n and positive integers v_n with $v_n \geq d(\lambda_n)$ for all $n \in \mathbb{N}$. By assumption, $D(\lambda_n) \geq n^\epsilon$ for some constant $\epsilon > 0$. The family (f_n) is in VNP. Writing $\lambda_n = (\mu_1^{w_1}, \dots, \mu_t^{w_t})$, consider the product

$$\prod_{i=1}^{v_n} \left(1 + \sum_{j=1}^t y_j x_i^{\mu_j} \right)$$

which is a VP family (and hence a VNP family), since v_n , t , and $d(\lambda_n)$ are polynomially bounded for a p -family. The coefficient of $y_1^{w_1} \cdots y_t^{w_t}$ in this product is exactly f_n . By the standard closure of VNP under taking coefficients of monomials in polynomially many auxiliary variables [[Bür00](#), Theorem 2.19(3)], (f_n) is a VNP family.

By [Lemma 4.1](#), there exist constants $\alpha > 0$ and n_0 , and, for each $n \geq n_0$, parameters $r_n > \alpha \sqrt{D(\lambda_n)}$ and a vector $\mathbf{w}_n \in \mathbb{Z}_{>0}^{r_n}$ with even total sum such that

$$\text{PerfMatch}_{r_n}^{(\mathbf{w}_n)} \leq_c f_n.$$

Since $D(\lambda_n) \geq n^\epsilon$, we have $r_n > \alpha n^{\epsilon/2}$. By [Lemma 3.1](#), setting $s_n = 2\lfloor r_n/4 \rfloor$, we obtain

$$\text{PerfMatch}_{s_n} \leq_c \text{PerfMatch}_{r_n}^{(\mathbf{w}_n)}.$$

Composing the two reductions via [Lemma A.5](#) yields

$$\text{PerfMatch}_{s_n} \leq_c f_n.$$

Thus $s_n \geq cn^{\epsilon/2}$ for some constant $c > 0$ and all sufficiently large n . To reduce an arbitrary even-size perfect matching polynomial PerfMatch_m to the family (f_n) , choose an index $N = \max\{n_0, \lceil (m/c)^{2/\epsilon} \rceil\}$. Then $s_N \geq m$, and PerfMatch_m is a projection of PerfMatch_{s_N} by fixing the extra $s_N - m$ vertices to a predetermined perfect matching of weight 1 and setting all other incident extra edges to 0. Since N is polynomially bounded in m , the perfect matching family c -reduces to (f_n) . The perfect matching polynomial family is VNP-complete under p -projections, hence $(f_n)_{n \in \mathbb{N}}$ is VNP-hard under c -reductions. Together with membership in VNP, this proves VNP-completeness. $\square$

5 Conclusion and Open Problems

We have established an almost-complete dichotomy for the algebraic complexity of monomial symmetric polynomials in terms of the number of distinct parts in the partition λ . In our proofs, we introduced the blow-up matching polynomial and showed its hardness under arbitrary blowups; this may be of independent interest toward robust hardness proofs of perturbed versions of VNP-hard problems.

Our results suggest several directions for future work, some of which we list below.

Removal of variable count restriction: All our lower bounds and VNP-hardness results require that the number of variables in the monomial symmetric polynomial is at least its degree. Can we show a lower bound for the case when the number of variables is smaller than the degree? This requirement arises because we rely on the algebraic independence of power symmetric polynomials via [Theorem A.2](#), which only holds when $n \geq d(\lambda)$. Removing this constraint would significantly strengthen our results and extend the dichotomy to a broader class of monomial symmetric polynomials.

Vandermonde permanents: The Vandermonde permanent discussed in the introduction is

$$\text{Per} \left((x_i^j)_{i,j \in [n]} \right) = m_{(n, n-1, \dots, 1)}(x_1, \dots, x_n).$$

This family is not covered by our results, since it has only n variables but degree $\binom{n+1}{2}$. It is a natural open problem whether the Vandermonde permanent family is VNP-complete. Our techniques for proving VNP-completeness require the number of variables to be at least the degree, because the various power symmetric polynomials appearing in the Binet–Minc ([Theorem 2.1](#)) formula for m_λ are required to be algebraically independent.

Fine-grained complexity in the middle regime: What is the complexity of the monomial symmetric polynomial family m_λ when the number of distinct parts in λ is $\omega(1)$ but not too large for it to be VNP-complete, e.g., $\text{poly}(\log n)$? Can we show a lower bound for this case? Treating the number of distinct parts as a parameter might be useful; parameterized complexity was introduced in algebraic complexity by [\[BE19\]](#).

For immanants, Curticapean [\[Cur21\]](#) has shown such a fine-grained result: If a certain parameter is bounded by a constant, then the corresponding immanant family is in VP, whereas if this parameter grows unboundedly, the family is hard for the parameterized class VW[1]. In a similar spirit, can we for instance show that m_λ is fixed parameter tractable when parameterized by the number of distinct parts in λ , without requiring any restriction on the number of variables? This would provide a more nuanced understanding of the complexity landscape beyond the dichotomy.

Permanents of non-generic matrices: In general, what other families of matrix permanents are VNP-complete where the entries of the matrix are not generic variables $x_{i,j}$? Our techniques in this paper, particularly the use of the Binet–Minc identity and Sidon sets, might extend to other structured matrices with algebraic or combinatorial patterns.

References

- [Abb90] H. L. Abbott. Sidon sets. *Canadian Mathematical Bulletin*, 33(3):335–341, 1990.
- [BE19] Markus Bläser and Christian Engels. Parameterized Valiant’s Classes. In Bart M. P. Jansen and Jan Arne Telle, editors, *14th International Symposium on Parameterized and Exact Computation (IPEC 2019)*, volume 148 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 3:1–3:14, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.
- [BJ19] Markus Bläser and Gorav Jindal. On the Complexity of Symmetric Polynomials. In Avrim Blum, editor, *10th Innovations in Theoretical Computer Science Conference (ITCS 2019)*, volume 124 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 47:1–47:14, Dagstuhl, Germany, 2019. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.
- [BKR⁺26] Somnath Bhattacharjee, Mrinal Kumar, Shanthanu S. Rai, Varun Ramanathan, Ramprasad Satharishi, and Shubhangi Saraf. Constant-depth circuits for polynomial GCD over any characteristic. In Dana Moshkovitz, editor, *41st Computational Complexity Conference, CCC 2026*, volume 383 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 16:1–16:21. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2026.
- [BSC17] Ben Blum-Smith and Samuel Coskey. The fundamental theorem on symmetric polynomials: History’s first whiff of galois theory. *The College Mathematics Journal*, 48(1):pp. 18–29, 2017.
- [Bür00] Peter Bürgisser. *Completeness and Reduction in Algebraic Complexity Theory*, volume 7 of *Algorithms and computation in mathematics*. Springer, 2000.
- [Bür16] Peter Bürgisser. Permanent versus determinant, obstructions, and kronecker coefficients. *Séminaire Lotharingien de Combinatoire*, 75:B75a, 2016. 19 pp.
- [CDM17] Radu Curticapean, Holger Dell, and Dániel Marx. Homomorphisms are a good basis for counting small subgraphs. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 210–223. ACM, 2017.
- [CKL⁺23] Prasad Chaugule, Mrinal Kumar, Nutan Limaye, Chandra Kanta Mohapatra, Adrian She, and Srikanth Srinivasan. Schur polynomials do not have small formulas if the determinant does not. *Comput. Complex.*, 32(1):3, 2023.
- [CLS22] Radu Curticapean, Nutan Limaye, and Srikanth Srinivasan. On the VNP-Hardness of Some Monomial Symmetric Polynomials. In Anuj Dawar and Venkatesan Guruswami, editors, *42nd IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2022)*, volume 250 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 16:1–16:14, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.
- [CM14] Radu Curticapean and Dániel Marx. Complexity of counting subgraphs: Only the boundedness of the vertex-cover number counts. In *55th IEEE Annual Symposium on Foundations of Computer*

- Science, FOCS 2014, Philadelphia, PA, USA, October 18-21, 2014*, pages 130–139. IEEE Computer Society, 2014.
- [Cur21] Radu Curticapean. A full complexity dichotomy for immanant families. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 1770–1783. ACM, 2021.
 - [DPS26a] Anuj Dawar, Benedikt Pago, and Tim Seppelt. Symmetric algebraic circuits and homomorphism polynomials. In Shubhangi Saraf, editor, *17th Innovations in Theoretical Computer Science Conference, ITCS 2026, Bocconi University, Milan, Italy, January 27-30, 2026*, volume 362 of *LIPIcs*, pages 46:1–46:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2026.
 - [DPS26b] Prateek Dwivedi, Benedikt Pago, and Tim Seppelt. Lower bounds in algebraic complexity via symmetry and homomorphism polynomials. In Aditya Bhaskara and Artur Czumaj, editors, *Proceedings of the 58th Annual ACM Symposium on Theory of Computing, STOC 2026, Salt Lake City, UT, USA, June 22-26, 2026*, pages 631–640. ACM, 2026.
 - [Erd65] Paul Erdős. Extremal problems in number theory. In *Proceedings of Symposia in Pure Mathematics*, volume 8, pages 181–189. American Mathematical Society, 1965.
 - [FLMS17] Hervé Fournier, Nutan Limaye, Meena Mahajan, and Srikanth Srinivasan. The shifted partial derivative complexity of elementary symmetric polynomials. *Theory Comput.*, 13(1):1–34, 2017.
 - [HY11] Pavel Hrubes and Amir Yehudayoff. Homogeneous formulas and symmetric polynomials. *Comput. Complex.*, 20(3):559–578, 2011.
 - [Kal85] Erich Kaltofen. Polynomial-time reductions from multivariate to bi- and univariate integral polynomial factorization. *SIAM Journal on Computing*, 14(2):469–489, 1985.
 - [Kal86] E Kaltofen. Uniform closure properties of p-computable functions. In *Proceedings of the Eighteenth Annual ACM Symposium on Theory of Computing, STOC '86*, page 330–337, New York, NY, USA, 1986. Association for Computing Machinery.
 - [Kal87] E. 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, STOC '87*, page 443–452, New York, NY, USA, 1987. Association for Computing Machinery.
 - [Kal89] Erich L. Kaltofen. Factorization of polynomials given by straight-line programs. *Adv. Comput. Res.*, 5:375–412, 1989.
 - [Mah13] Meena Mahajan. Algebraic complexity classes. *ArXiv*, abs/1307.3863, 2013.
 - [Min84] H. Minc. *Permanents*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 1984.
 - [NW96] Noam Nisan and Avi Wigderson. Lower bounds on arithmetic circuits via partial derivatives. *Computational Complexity*, 6(3):217–234, September 1996.
 - [Sap15] Ramprasad Saptharishi. A survey of lower bounds in arithmetic circuit complexity. *Github survey*, 95, 2015.
 - [Str73] Volker Strassen. Vermeidung von divisionen. *Journal für die reine und angewandte Mathematik*, 264:184–202, 1973.

- [SW99] Amir Shpilka and Avi Wigderson. Depth-3 arithmetic formulae over fields of characteristic zero. In *Proceedings of the 14th Annual IEEE Conference on Computational Complexity, Atlanta, Georgia, USA, May 4-6, 1999*, page 87. IEEE Computer Society, 1999.
- [SY10] Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Found. Trends Theor. Comput. Sci.*, 5(3–4):207–388, March 2010.
- [Sze75] Gábor Szegő. *Orthogonal Polynomials*. American Mathematical Society, Providence, RI, 1975.
- [Val79] Leslie G. Valiant. Completeness classes in algebra. In Michael J. Fischer, Richard A. DeMillo, Nancy A. Lynch, Walter A. Burkhard, and Alfred V. Aho, editors, *Proceedings of the 11th Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1979, Atlanta, Georgia, USA*, pages 249–261. ACM, 1979.

A Background

A.1 Symmetric Polynomials

Theorem A.1 (Fundamental Theorem of Symmetric Polynomials, see [BSC17]). *Every symmetric polynomial in $\mathbb{F}[x_1, \dots, x_n]$ can be uniquely expressed as a polynomial in the elementary symmetric polynomials $e_1, e_2, \dots, e_n$. Namely, for every symmetric polynomial $f \in \mathbb{F}[x_1, \dots, x_n]$, there exists a unique polynomial $g \in \mathbb{F}[y_1, y_2, \dots, y_n]$ such that*

$$f(x_1, \dots, x_n) = g(e_1(x_1, \dots, x_n), \dots, e_n(x_1, \dots, x_n)).$$

An alternative basis is provided by the *power sum symmetric polynomials*. The following theorem follows from the Newton identities, which relate the power sum symmetric polynomials to the elementary symmetric polynomials.

Theorem A.2 (Power sum basis). *For n variables over $\mathbb{C}$, the power sum symmetric polynomials*

$$p_1, p_2, \dots, p_n$$

are algebraically independent and generate the ring of symmetric polynomials. Every symmetric polynomial can be uniquely expressed as a polynomial in $p_1, \dots, p_n$. Namely, for every symmetric polynomial $f \in \mathbb{C}[x_1, \dots, x_n]$, there exists a unique polynomial $g \in \mathbb{C}[y_1, y_2, \dots, y_n]$ such that

$$f(x_1, \dots, x_n) = g(p_1(x_1, \dots, x_n), \dots, p_n(x_1, \dots, x_n)).$$

A.2 Algebraic Complexity Classes and Reductions

In algebraic complexity theory, reductions between polynomials are typically defined via *projections*. A polynomial $f \in \mathbb{F}[x_1, x_2, \dots, x_n]$ is said to be a projection of another polynomial $g \in \mathbb{F}[y_1, y_2, \dots, y_m]$ if f can be obtained from g by substituting each variable y_i with either a variable x_j or a constant from the field $\mathbb{F}$. This notion of reduction is quite restrictive, since it does not allow adaptive computations or multiple queries to the polynomial being reduced to.

A more flexible notion is given by *c-reductions*. To define it, we introduce the oracle complexity of a polynomial. The oracle complexity of a polynomial f with respect to another polynomial g is the minimum size of an arithmetic circuit computing f that is allowed to query an oracle for g . Following the notation of [CLS22], we denote this quantity by $L^g(f)$. If $L^g(f)$ is polynomially bounded in the number of variables

and degree of f and g , we simply write $f \leq_c g$ to simplify the notation. Throughout the paper, the complexity notation $\tilde{O}$ hides polylogarithmic factors.

Now we define the notion of VNP-completeness under c -reductions. A sequence $(u) = (u_n)_{n \in \mathbb{N}}$ of non-negative integers is called p -bounded if there exists a polynomial function p such that for all $n \in \mathbb{N}$, we have $u_n \leq p(n)$.

A sequence $(f_n)_{n \in \mathbb{N}}$ of polynomials is called a p -family if for each $n \in \mathbb{N}$, the number of variables in f_n and degree of f_n are p -bounded.

Definition A.3. *The class VP consists of all p -families $(f_n)_{n \in \mathbb{N}}$ such that $(L(f_n))_{n \in \mathbb{N}}$ is p -bounded.*

Definition A.4. *The class VNP consists of all p -families $(f_n)_{n \in \mathbb{N}}$ where $f_n \in \mathbb{F}[x_1, \dots, x_{v_n}]$ such that there exists a polynomial family $(g_n)_{n \in \mathbb{N}}$ in VP with $g_n \in \mathbb{F}[x_1, \dots, x_{v_n}, y_1, \dots, y_{u_n}]$ satisfying:*

$$f_n(x_1, \dots, x_{v_n}) = \sum_{b \in \{0,1\}^{u_n}} g_n(x_1, \dots, x_{v_n}, b_1, \dots, b_{u_n}).$$

A p -family $(f_n)_{n \in \mathbb{N}}$ is said to be a c -reduction of $(g_n)_{n \in \mathbb{N}}$, denoted by $(f_n) \preceq_c (g_n)$, if there exists a p -bounded sequence $(t_n)_{n \in \mathbb{N}}$ such that $L^{g_{t_n}}(f_n)$ is p -bounded. It is well known that the permanent polynomial family $(\text{Per}_n)_{n \in \mathbb{N}}$ defined as

$$\text{Per}_n(X) = \sum_{\sigma \in S_n} \prod_{i=1}^n x_{i, \sigma(i)},$$

is VNP-complete under c -reductions. In fact, $(\text{Per}_n)_{n \in \mathbb{N}}$ is VNP-complete under the more restrictive notion of p -projections, which is a special case of c -reductions where the reduction is given by a single substitution of variables. The following lemma is an immediate consequence of the definition of c -reductions.

Lemma A.5. *If f, g, h are polynomials such that $L^f(g) \leq a$ and $L^g(h) \leq b$, then $L^f(h) \leq O(ab)$.*

A.3 Tools From Algebraic Complexity Theory

Corollary 2.2 expresses the monomial symmetric polynomial m_λ as a polynomial in the power sum symmetric polynomials. Assume that the number of variables n is at least the degree of m_λ , i.e., $n \geq d(\lambda)$. Then by **Theorem A.2**, the power sum symmetric polynomials $p_1, p_2, \dots, p_n$ are algebraically independent and generate the ring of symmetric polynomials. Theorem 4 of [BJ19], together with the efficient Newton identities, implies that the arithmetic complexity of m_λ^{pow} with respect to the oracle m_λ is quite small. This result is stated more explicitly in [CKL⁺23, Theorem 4.16].

Lemma A.6 ([BJ19, CKL⁺23]). *For any partition λ and all $n \geq d(\lambda)$, we have*

$$L^{m_\lambda(x_1, \dots, x_n)}(m_\lambda^{\text{pow}}) \leq \tilde{O}(n^4(d(\lambda))^3).$$

These results prove a more general statement: for any symmetric polynomial, the complexity of expressing it as a polynomial in the power sums is small relative to an oracle for the original polynomial. The above lemma is the special case corresponding to monomial symmetric polynomials.

We will also need the following folklore result concerning the arithmetic complexity of homogeneous components of polynomials. For a polynomial $f \in \mathbb{F}[x_1, \dots, x_n]$ and a non-negative integer m , we denote by $\text{Hom}_m(f)$ the homogeneous component of f of degree m , i.e., the sum of all monomials in f that have total degree m .

Lemma A.7 (Folklore, Lemma 12 in [BJ19]). *Let $f \in \mathbb{F}[x_1, \dots, x_n]$ be a polynomial of degree d . Then for any $m \in \{0, 1, \dots, d\}$, we have:*

$$L^f(\text{Hom}_m(f)) \leq O(nd).$$

We also need a result on the arithmetic complexity of dividing polynomials by a monomial factor. This follows immediately from results in [Str73, Kal86, Kal87, Kal85, Kal89], but we give a self-contained proof below for completeness.

Lemma A.8. *Let $f \in \mathbb{F}[x_1, \dots, x_n]$ be a polynomial of degree d . Suppose m is a monomial factor of f . Then*

$$L^f(f/m) \leq \text{poly}(n, d).$$

Proof. Suppose $f \in \mathbb{F}[x_1, \dots, x_n]$ is a polynomial of degree d . Suppose m is a monomial factor of f . Then $f = gm$, where g is a polynomial of degree at most d . Define $f_{\text{shift}}(x) := f(x_1 + 1, \dots, x_n + 1)$ and $g_{\text{shift}}(x) := g(x_1 + 1, \dots, x_n + 1)$. The oracle for f computes f_{shift} by querying f at the shifted inputs. This substitution does not change the degree of the polynomial, so f_{shift} is still a polynomial of degree d . Moreover, under this substitution, the monomial m is mapped to a polynomial of the form $1 - h$ with constant term of h being zero. The polynomial h is explicitly computable from the monomial m , and hence so are its powers. Thus we can write $f_{\text{shift}} = g_{\text{shift}}(1 - h)$. Now we have $g_{\text{shift}} = f_{\text{shift}}/(1 - h) = f_{\text{shift}}(1 + h + h^2 + \dots)$. We use F to denote the polynomial $f_{\text{shift}}(1 + h + h^2 + \dots + h^d)$. Note that g_{shift} is a polynomial of degree at most d , hence we have:

$$g_{\text{shift}} = \sum_{i=0}^d \text{Hom}_i(F).$$

Indeed, $F = g_{\text{shift}}(1 - h^{d+1})$, and the term $g_{\text{shift}}h^{d+1}$ has no homogeneous component of degree at most d because h has zero constant term. Since $\deg(F) = O(d^2)$, the sum of the homogeneous components of F of degree at most d can be recovered by the usual interpolation in an auxiliary scaling variable using polynomially many evaluations of F . Finally substituting $x_i \leftarrow x_i - 1$ recovers g from g_{shift} . Thus $L^f(g) \leq \text{poly}(n, d)$. $\square$

B Omitted Proofs

Proof sketch of Theorem 1.2. Let $\lambda = (\mu_1^{w_1}, \mu_2^{w_2}, \dots, \mu_t^{w_t})$ be a partition with at most t distinct parts. Then the monomial symmetric polynomial m_λ can be written as the coefficient of $y_1^{w_1} y_2^{w_2} \dots y_t^{w_t}$ in the polynomial G defined below:

$$G(x_1, x_2, \dots, x_n, y_1, y_2, \dots, y_t) := \prod_{i=1}^n \left(1 + \sum_{j=1}^t y_j x_i^{\mu_j} \right).$$

Note that the individual degree of each y_j in G is at most n . We now think of G as a polynomial in the variables $y_1, y_2, \dots, y_t$ with coefficients in $\mathbb{F}[x_1, x_2, \dots, x_n]$. We use the well-known Kronecker substitution and replace y_i by $y^{(n+1)^{i-1}}$ for all $i \in [t]$ to obtain a univariate polynomial $H(y)$ in y with coefficients in $\mathbb{F}[x_1, x_2, \dots, x_n]$. This mapping ensures that each monomial $y_1^{a_1} y_2^{a_2} \dots y_t^{a_t}$ of G is mapped to a unique monomial $y^{a_1 + a_2(n+1) + \dots + a_t(n+1)^{t-1}}$ of $H(y)$. Define $W := w_1 + w_2(n+1) + \dots + w_t(n+1)^{t-1}$.

In particular, the monomial $y_1^{w_1} y_2^{w_2} \dots y_t^{w_t}$ is mapped to y^W . Hence, we can extract the coefficient of y^W in $H(y)$ to obtain m_λ . Since the degree of $H(y)$ is at most $(n+1)^t - 1$, we can evaluate $H(y)$ at $(n+1)^t$ many points and interpolate to extract the coefficient of y^W . For one evaluation point $\alpha \in \mathbb{F}$, computing all powers $x_i^{\mu_j}$ with $\mu_j \leq d$, forming the n factors, and multiplying them costs $O(nt \cdot \text{poly}(d))$ gates. Since λ is a partition of degree d into positive parts, we have $t \leq d$, so this per-point cost is at most $n \cdot \text{poly}(d)$. The $(n+1)^t = n^{O(t)}$ interpolation points and final linear combination give the claimed $\text{poly}(d) \cdot n^{O(t)}$ bound. $\square$

Proof of Lemma 3.4. Let λ be a new variable and let q be the polynomial obtained from $\text{PerfMatch}_r^{(w)}$ by substituting $\theta \leftarrow \lambda\theta + 1$ for every edge variable θ of the blow-up graph, including the inter-block variables

x_{ij} and the intra-block variables z_i . To prove the lemma, we establish the identity

$$\text{Match}_r^{(\mathbf{w})} = \sum_{k=0}^{n/2} \frac{[\lambda^k]q}{(n-2k-1)!!}, \quad (12)$$

where, for even m , $(m-1)!! = (m-1)(m-3)\cdots 1$ is the number of perfect matchings in K_m , with the convention $(-1)!! = 1$. The required c -reduction follows by summing over k and performing interpolation on λ , similar to [Lemma A.7](#).

To prove (12), we take a graph perspective: The substitution above amounts to multiplying each original edge-weight in $K_r^{(\mathbf{w})}$ by λ and adding an additional edge of weight 1 between each pair of vertices. Let $Q_r^{(\mathbf{w})}$ denote the resulting multigraph. Thus, for every pair of vertices u, v in $K_r^{(\mathbf{w})}$, the multigraph $Q_r^{(\mathbf{w})}$ contains one “original” edge of weight $\lambda\theta_{uv}$, where θ_{uv} is the corresponding inter-block or intra-block variable, and one “neutral” edge of weight 1.

Every k -matching M in $K_r^{(\mathbf{w})}$ can be extended to exactly $(n-2k-1)!!$ perfect matchings in $Q_r^{(\mathbf{w})}$ by adding a perfect matching on neutral edges on the $n-2k$ vertices not matched by M . Each such extension has the same weight as M , up to a factor λ^k . The coefficient of λ^k in q sums over all such extensions of k -matchings in $K_r^{(\mathbf{w})}$. $\square$

C Proof of [Lemma 3.2](#)

Recall that the matching-generating polynomial of H is $M(H, x) := \sum_k m_k x^k$. Throughout this section, fix $1 \leq b \leq a$ and abbreviate

$$\begin{aligned} f(x) &= M(K_{a,b}, x) \\ g(x) &= M(K_{a-1,b}, x). \end{aligned}$$

By definition,

$$f(x) = \sum_{k=0}^b \binom{a}{k} \binom{b}{k} k! x^k,$$

so $\deg f = b$, $\deg g \leq b$, and $f(0) = g(0) = 1$. Moreover $f \not\equiv g$, for instance because the coefficient of x is ab in f and $(a-1)b$ in g .

We will use the standard relation between matchings in complete bipartite graphs and generalized Laguerre polynomials. For $\alpha > -1$,

$$L_n^{(\alpha)}(y) := \sum_{k=0}^n (-1)^k \binom{n+\alpha}{n-k} \frac{y^k}{k!}.$$

A direct coefficient comparison gives

$$M(K_{a,b}, x) = b! x^b L_b^{(a-b)}(-x^{-1}).$$

By [[Sze75](#), Section 5.1, (5.1.1)], $L_n^{(\alpha)}$ is orthogonal on $[0, \infty)$ with weight $e^{-x}x^\alpha$ for $\alpha > -1$. Since $a-b \geq 0$, Szegő’s zero theorem [[Sze75](#), Theorem 3.3.1] implies that $L_b^{(a-b)}$ has b distinct positive real zeros. Hence the roots of $f = M(K_{a,b}, x)$ are simple and nonzero.

Proof of [Lemma 3.2](#). The preceding discussion shows that f has b distinct nonzero roots. If g vanished on all these roots, then f and g would agree on $b+1$ points, namely on 0 and the b roots. This would imply $f \equiv g$, because f and g are of degree at most b , contradicting $f \not\equiv g$. It follows that there is a root ξ of f that is not a root of g ; this is our desired value $\xi = \xi(a, b)$. $\square$