Electronic Colloquium on Computational Complexity, Report No. 61 (2011)

Affine projections of polynomials

Neeraj Kayal *

April 18, 2011

Abstract

An m-variate polynomial f is said to be an *affine projection* of some n-variate polynomial g if there exists an $n \times m$ matrix A and an n-dimensional vector $\mathbf{b}$ such that $f(\mathbf{x}) = g(A\mathbf{x} + \mathbf{b})$. In other words, if f can be obtained by replacing each variable of g by an affine combination of the variables occurring in f, then it is said to be an affine projection of g. Given f and g can we determine whether f is an affine projection of g? Some well known problems (such as the determinant versus permanent and matrix multiplication for example) are instances of this problem.

The intention of this paper is to understand the complexity of the corresponding computational problem: given polynomials f and g find A and b such that $f = g(A\mathbf{x} + \mathbf{b})$, if such an $(A, \mathbf{b})$ exists. We first show that this is an NP-hard problem. We then focus our attention on instances where g is a member of some fixed, well known family of polynomials so that the input consists only of the polynomial $f(\mathbf{x})$ having m variables and degree d. We consider the situation where $f(\mathbf{x})$ is given to us as a blackbox (i.e. for any point $\mathbf{a} \in \mathbb{F}^m$ we can query the blackbox and obtain $f(\mathbf{a})$ in one step) and devise randomized algorithms with running time $\mathsf{poly}(mnd)$ in the following special cases:

(1) when $f = \mathsf{Perm}_n(A\mathbf{x} + \mathbf{b})$ and A satisfies $\text{rank}(A) = n^2$. Here Perm_n is the permanent polynomial.

(2) when $f = \mathsf{Det}_n(A\mathbf{x} + \mathbf{b})$ and A satisfies $\text{rank}(A) = n^2$. Here Det_n is the determinant polynomial.

(3) when $f = \mathsf{Pow}_{n,d}(A\mathbf{x} + b)$ and A is a *random* $n \times m$ matrix with $d = n^{\Omega(1)}$. Here $\mathsf{Pow}_{n,d}$ is the power-symmetric polynomial of degree d.

(4) when $f = \mathsf{SPS}_{n,d}(A\mathbf{x} + b)$ and A is a *random* $(nd) \times m$ matrix with n constant. Here $\mathsf{SPS}_{n,d}$ is the sum-of-products polynomials of degree d with n terms.

Acknowledgements. The author would like to thank : Ketan Mulmuley and Milind Sohoni for suggesting the use of the corresponding lie algebras for equivalence to the determinant, K V Subrahmanyam for his nice lectures on representation theory, Michael Forbes for pointing out the relationship between symmetric rank and tensor rank, Shubhangi Saraf and Srikanth Srinivasan for discussions pertaining to projections of the sum of products polynomial.

*Microsoft Research India, `neeraka@microsoft.com`

ISSN 1433-8092

Contents

1 Introduction

The topic of interest here is the notion of an *affine projection* of a polynomial. Intuitively, a polynomial f is an affine projection of a polynomial g, denoted $f \leq_{\text{aff}} g$, if f is obtained from g via an affine change of variables. More formally, an m-variate polynomial f is said to be an affine projection of some n-variate polynomial g if there exist m-variate affine forms (i.e. degree one polynomials) $\ell_1, \ell_2, \ldots, \ell_n$ such that

$$f(\mathbf{x}) = g(\ell_1, \ell_2, \ldots, \ell_n),$$

written compactly as $f(\mathbf{x}) = g(A \cdot \mathbf{x} + \mathbf{b})$ where A is an $n \times m$ matrix $\mathbf{b}$ is an n-dimensional vector. In this paper, we study the computational complexity of finding an affine projection given the polynomials f and g (if it exists). Let us state this formally.

Name: PolyProj
Input: Polynomials $f(x_1, x_2, \ldots, x_m)$ and $g(x_1, x_2, \ldots, x_n)$ over the field $\mathbb{F}$.
Output: An $n \times m$ matrix A and a vector $\mathbf{b} \in \mathbb{F}^n$ such that $f(\mathbf{x}) = g(A\mathbf{x} + \mathbf{b})$, if such an A and $\mathbf{b}$ exist. Else output `'No such projection exists'`.

1.1 Motivation.

The motivation for this study is that some well-known open problems/conjectures (and also some not so well-known conjectures) in arithmetic complexity are instances of this problem. We now introduce the reader to some popular families of polynomials and then mention how PolyProj encompasses an apparenly diverse collection of problems.

(1). $\mathsf{Sym}_{n,d}$: the elementary symmetric polynomials.

$$\mathsf{Sym}_{n,d}(x_1, x_2, \ldots, x_n) := \sum_{S \subseteq [n], |S| = d} \prod_{i \in S} x_i$$

(2). $\mathsf{Pow}_{n,d}$: the power symmetric polynomials.

$$\mathsf{Pow}_{n,d}(x_1, x_2, \ldots, x_n) := \sum_{i \in [n]} x_i^d$$

(3). $\mathsf{SPS}_{n,d}$: the sum of products polynomial.

$$\mathsf{SPS}_{n,d}(x_{11}, \ldots, x_{1d}, x_{21}, \ldots, x_{2d}, \ldots, x_{nd}) := \sum_{i=1}^{n} \prod_{j=1}^{d} x_{ij}$$

(4). Det_n: the determinant polynomial.

$$\mathsf{Det}_n(x_{11}, x_{12}, \ldots, x_{nn}) := \sum_{\pi \in S_n} \text{sign}(\pi) \prod_{i=1}^{n} x_{i\pi(i)}$$

(5). Perm_n: the permanent polynomial.

$$\mathsf{Perm}_n(x_{11}, x_{12}, \ldots, x_{nn}) := \sum_{\pi \in S_n} \prod_{i=1}^{n} x_{i\pi(i)}$$

(6). TrMat_n: the trace of matrix multiplication.

$$\mathsf{TrMat}_n(\mathbf{x}, \mathbf{y}, \mathbf{z}) := \sum_{i,j,k \in [n]} x_{ij} \cdot y_{jk} \cdot z_{ki}$$

In talking about these polynomials, when the parameters n and d are clear from context we will drop these - so for example we will often refer to Det_n simply as Det. For concreteness, for the rest of this paper we fix the underlying field to be $\mathbb{C}$, the field of complex numbers. [1] Let us now see how some open problems and/or interesting results in arithmetic complexity can equivalently be stated in terms of some polynomial being a projection of some other polynomial. We begin with two well known instances of PolyProj :

(1) The determinant versus permanent problem (cf. the works of [Sze92,Pól71,vzG87,Cai90,Mes89, MR04, CCL08] and the survey [Agr06]). It is conjectured that

$$\mathsf{Perm}_m \not\leq_{\mathrm{aff}} \mathsf{Det}_n \quad \text{for any } n = m^{O(1)}.$$

(2) The arithmetic complexity of matrix multiplication (cf. [Str69, BI11]). It is conjectured that

$$\mathsf{TrMat}_n \leq_{\mathrm{aff}} \mathsf{SPS}_{m,3} \quad \text{for some } m = \tilde{O}(n^2).$$

For example Strassen's 1969 discovery [Str69] that the product of two 2×2 matrices can be computed with 7 multplications can be restated as TrMat_2 is a projection of $\mathsf{SPS}_{7,3}$ in the following manner:

$$\begin{aligned} \mathsf{TrMat}_2 \quad = \quad & ((x_{11} + x_{22}) \cdot (y_{11} + y_{22}) \cdot (z_{11} + z_{22})) + ((x_{21} + x_{22}) \cdot y_{11} \cdot (z_{21} - z_{22})) \\ & + (x_{11} \cdot (y_{12} - y_{22}) \cdot (z_{12} + z_{22})) + (x_{22} \cdot (y_{21} - y_{11}) \cdot (z_{11} + z_{21})) \\ & + ((x_{11} + x_{12}) \cdot y_{22} \cdot (-z_{11} + z_{12})) + ((x_{21} - x_{11}) \cdot (y_{11} + y_{12}) \cdot z_{22}) \\ & + ((x_{12} - x_{22}) \cdot (y_{21} + y_{22}) \cdot z_{11}) \end{aligned}$$

Some of the lesser known conjectures/problems include:

(3) Lower bounds for depth-three arithmetic formulas (cf. the survey [SY10]): in our terminology the problem is to find an explicit low-degree polynomial f such that $f \not\leq_{\mathrm{aff}} \mathsf{SPS}_{n,d}$ for any $(n \cdot d) = m^{O(1)}$. A closely related problem that will be relevant for us is the reconstruction problem for depth-three arithmetic circuits [Shp07, KS09] which in our terminology is the following: given a polynomial f and integers n, d find $A, \mathbf{b}$ such that

$$f(\mathbf{x}) = \mathsf{SPS}_{n,d}(A \cdot \mathbf{x} + \mathbf{b}).$$

[1] The discussion in this paper will carry over with some minor changes as long as the characteristic of the field $\mathbb{F}$ is large enough.

(4) Waring problem for polynomials (cf. [Ell69]): for an m-variate polynomial f of degree d, what is the smallest n such that $f \leq_{\text{aff}} \mathsf{Pow}_{n,d}$? For more on the Waring problem for polynomials see the works of Ellison [Ell69], Ehrenborg and Rota [ER93], Kleppe [Kle99] and the references therein. The number n is also sometimes called the rank of the symmetric tensor f (any m-variate polynomial f can be viewed as a symmetric tensor of order m). Thus this problem is sometimes referred to as the problem of determining the symmetric rank of symmetric tensors [BGI09, CGLM08].

(5) Lower bounds for affine projections of symmetric polynomials [Shp02]: find an explicit m-variate polynomial f of degree $m^{O(1)}$ such that

$$f \not\leq_{\text{aff}} \mathsf{Sym}_{n,d} \quad \text{whenever } (n \cdot d) = m^{O(1)}$$

(6) A conjecture of Scott Aaronson [Aar08]. Random m-variate affine projections of Det_n are *pseudorandom polynomials* in the sense that they are indistinguishable (via $\mathsf{poly}\left(\binom{m+n}{n}\right)$-time algorithms) from truly random m-variate polynomials of degree n.

With these open problems/conjectures and the related upper bounds/algorithms forming the backdrop, one is naturally compelled to ask the following question – given polynomials f and g, can we determine if f is an affine projection of g? In this paper we make an attempt to understand this question by examining it under the lens of computational complexity. We show that this problem is NP-hard in general but admits randomized polynomial-time algorithms in some natural special cases. All the special cases that we look at will have g fixed and a member of one of the families of polynomials listed above. Even with g fixed, we do not know how to solve PolyProj efficiently so we impose some further restrictions on the kind of projections that are allowed.
At this point we should specify the representation used to encode the input polynomials. The affine projection problem is interesting whatever be the representation used. Here we will typically deal with an input polynomial f given as a blackbox - i.e. we have access to an oracle "holding" the polynomial f so that for any point $\mathbf{a} \in \mathbb{F}^m$, we can query this oracle and obtain the value of $f(\mathbf{a})$ in one step. For the hardness result we will use the sparse representation for polynomials wherein a polynomial f with t nonzero monomials is given as a list of t elements containing the monomials and their coefficients.

1.2 Affine equivalence.

Our first set of results concern the restriction where A is invertible, i.e. when f is affinely equivalent to g. Let us motivate our study of projections under this restriction with an example - quadratic polynomials. For simplicity let us consider the case where f and g are *homogeneous* quadratic polynomials. [2] It is a classic result that every homogeneous quadratic polynomial is equivalent (under invertible linear transformations) to $\mathsf{Pow}_{r,2}$ for some integer $r \geq 0$. So let f be equivalent to $\mathsf{Pow}_{r_f,2}$ and g be equivalent to $\mathsf{Pow}_{r_g,2}$. So then f is an affine projection of g if and only if $r_f \leq r_g$. This observation is effective and can be generalized suitably to inhomogeneous quadratic polynomial so that we have –

Fact 1. PolyProj *can be solved in polynomial-time for quadratic polynomials.*

[2] Similar remarks apply when f and g are inhomogeneous quadratic polynomials – one just needs to consider a few additional cases in that situation.

This example suggests that in order to solve PolyProj a first step might be to determine/characterize all the polynomials which are equivalent to a given polynomial g. Unfortunately, this is a quite difficult problem in general - it was shown by Agrawal and Saxena [AS06] that determining whether two polynomials are equivalent under invertible *linear transformations* is at least as difficult as Graph Isomorphism. Indeed there is a cryptosystem [Pat96] based on the presumed *average-case hardness* of polynomial equivalence. The first set of results presented here builds on previous work of the present author [Kay11] and shows that for g belonging to any of the families of polynomials listed above, one can efficiently determine whether a given polynomial is affinely equivalent to g. [3] The main cases that are tackled here are the cases of the permanent and the determinant. Specifically we show:

Theorem 2. *There exists a randomized algorithm that given integers n, d, m and blackbox access to an m-variate polynomial f of degree d determines whether there exists a matrix $A \in \mathbb{F}^{n^2 \times m}$ of rank n^2 and a vector $\mathbf{b} \in \mathbb{F}^{n^2}$ such that*

$$f(\mathbf{x}) = \mathsf{Perm}_n(A \cdot \mathbf{x} + \mathbf{b}).$$

Moreover the running time of the algorithm is $(mnd)^{O(1)}$.

Remark 3. (1) The theorem as stated here apparently tackles a problem more general than affine equivalence; however as we will see in section 5.1, it easily reduces to it.

(2) Note that for $m = n^2$ ($d = n$ without loss of generality), our running time of $\mathsf{poly}(n)$ is much smaller than $n!$, the number of monomials in the permanent.

A similar result holds for the determinant as well.

Theorem 4. *There exists a randomized algorithm that given integers n, d, m and blackbox access to an m-variate polynomial f of degree d determines whether there exists a matrix $A \in \mathbb{F}^{n^2 \times m}$ of rank n^2 and a vector $\mathbf{b} \in \mathbb{F}^{n^2}$ such that*

$$f(\mathbf{x}) = \mathsf{Det}_n(A \cdot \mathbf{x} + \mathbf{b})$$

Moreover the running time of the algorithm is $(mnd)^{O(1)}$.

A very rough overview of the main ingredients used in the algorithms of theorems 2 and 4 above is as follows. We use the structure of the lie algebra of the group of symmetries of a given polynomial f to determine most of the "continuous part" of the affine map from Perm_n (respectively Det_n) to f. We then use the second partial derivatives of f to determine the "discrete map" of this map while the residual "continuous part" is determined using some well-chosen substitutions. We refer the reader to section 3.1 for an overview and to sections 6.1 and 6.2 for the full details. In particular the two theorems presented above solve open problems posed by the same author in an earlier work [Kay11].

[3] Actually we do not present the algorithm for equivalence to TrMat_n here. This will be done in a forthcoming note.

1.3 Random Projections.

We then turn our attention to general affine projections, i.e. PolyProj instances where the rank of the matrix A is typically much less than the number of variables in g. The next pair of results we present here pertain to random projections of certain families of polynomials. One motivation for looking at random projections is the usage of PolyProj -like problems in cryptography. Specifically, Patarin [Pat96] uses the assumption that it is difficult to find the linear transformation given a random n-variate projection f of a random g. Another motivation is the conjecture by Scott Aaronson concerning random projections of the determinant mentioned in section 1.1. [4] But perhaps the most important reason (one born out of the author's laziness?) is that using this assumption we avoid several degenerate cases which might otherwise have bogged us down severely. This allows the algorithm and its analysis to be stated simply and cleanly and brings out well the main point of this line of work - namely that the mathematical ideas underlying lower bound proofs for affine projections of certain families of polynomials can be used to design efficient algorithms for solving (nondegenerate instances of) PolyProj for these families.
Background. It turns out that there is a common theme underlying most (all?) of the lower bound proofs for affine projections over algebraically closed fields - that these lower bound proofs have been through the discovery of what we call an affinely invariant property. Let us make precise the notion we have in mind.

Definition 5. *An* affinely invariant property *of polynomials is a map $\Pi : \mathbb{F}[\mathbf{x}] \mapsto \mathbb{R}$ such that for any two polynomials f and g in $\mathbb{F}[\mathbf{x}]$, if*

$$f \leq_{\text{aff}} g \quad \text{then } \Pi(f) \leq \Pi(g).$$

In particular if f is affinely equivalent to g then $\Pi(f) = \Pi(g)$.

The degree of a polynomial is one natural affinely invariant property. See section 5.2 for many more examples of affinely invariant properties and their applications to lower bounds. We show that the mathematical ideas underlying these affinely invariant properties can be used to prove that PolyProj has a unique solution (in the sense made precise below) in certain situations. These two ingredients can in turn be combined with a "project and lift" technique of Kaltofen [Kal89] and Shpilka [Shp07] to design efficient algorithms for PolyProj in these situations. Specifically we show:

Theorem 6. *There exists a randomized algorithm A whose input consists of integers n, m, d and blackbox access to an m-variate polynomial f of degree d. It does the following computation:*

(1) If $d > 2n$ and $f = \mathsf{Pow}_{n,d}(\ell_1, \ell_2, \ldots, \ell_n)$ then the algorithm always computes the ℓ_i's in $\mathsf{poly}(nd)$ time.

(2) If $d \leq 2n$ and f is of the form

$$f = \mathsf{Pow}_{n,d}(\ell_1, \ldots, \ell_n)$$

[4] To the best of our knowledge there is no complexity-theoretic evidence for Aaronson's conjecture. Specifically we do not know any (widely believed) complexity-theoretic hypothesis whose truth would imply Aaronson's conjecture; however if such evidence were to be found then its implication would be somewhat stunnning - it would give the first known "natural proof"-like barrier (in the sense of Razborov and Rudich [RR94]) for proving arithmetic circuit lower bounds.

then with probability at least $\left(1 - \frac{2dn}{|S|}\right)$*, the algorithm correctly computes the* ℓ_i*'s (over the random choice of* ℓ_i*'s with coefficients from a set* S*). Furthermore, the running time of the algorithm in this case is* $(d \cdot n)^{O(t)}$*, where* t *is the smallest integer satisfying*

$$\binom{t + d/2 - 1}{d/2} \geq n.$$

In particular, if $d \geq n^{\epsilon}$ *for some constant* $\epsilon > 0$ *then the algorithm has running time* $(n \cdot d)^{O(\epsilon^{-1})}$*.*

Remark 7.

1. The algorithm above is interesting only when d is relatively large. When d is small, say when $d = 3$ then the algorithm is no better than a brute force algorithm. Michael Forbes has noted that in this case, the problem is closely related to tensor rank (of order three tensors). See proposition 75 for the precise statement. It was shown by Hastad [Hås90] that computing the rank of order three tensors is NP-complete. [5]

2. When $d = n^{\Omega(1)}$ the algorithm has running time $\mathsf{poly}(nd)$. Note that in this case the number of monomials in such a f is typically exponential in (nd) so that the running time of our algorithm is much less than the number of (nonzero) monomials in f.

Theorem 8. *There exists a randomized algorithm* A *whose input consists of integers* n, m, d *and blackbox access to an* m*-variate polynomial* f *of degree* d *with* $d, m > n^2 + n$*. If* f *is of the form*

$$f = \sum_{i \in [n]} \prod_{j \in [d]} \ell_{ij}$$

and if every subset of the ℓ_{ij}*'s of size* $(n^2 + n)$ *is linearly independent then the algorithm* A *correctly computes the* ℓ_{ij}*'s . Furthermore, the running time of the algorithm is* $\mathsf{poly}(m \cdot d^{n^2})$*.*

Remark 9.

1. The algorithm above is interesting only when n is very small say n bounded. When f is set-multilinear, the quantity n equals (upto a contant factor) the tensor rank of f, a quantity which is known to be NP-hard to compute (cf. [Hås90], [Raz10] or [BI11])

2. If the ℓ_{ij}'s are chosen at random with coefficients from a large enough set S then with high probability (see fact 77 for a more precise statement), every subset of $(n^2 + n)$ ℓ_{ij}'s will be linearly independent. Thus this algorithm in particular solves PolyProj for random projections of $\mathsf{SPS}_{n,d}$ with n bounded.

3. When n is bounded the algorithm has running time $\mathsf{poly}(md)$. Note that in this case the number of monomials in such an f is $\binom{m+d}{d}$ so that when m and d are comparable then the running time of our algorithm is typically much less than the number of (nonzero) monomials in f.

4. Closely related is the work of Shpilka [Shp07] and Karnin and Shpilka [KS09] who give algorithms of running time $m \cdot |\mathbb{F}|^{(\log d)^{n^3}}$ for affine projections of $\mathsf{SPS}_{n,d}$ over finite fields. Note that their algorithm works so long as the ℓ_{ij}'s satisfy a relatively mild condition while

[5] Our understanding of tensor rank is quite poor - unlike symmetric tensors, we do not know the rank of even *generic* order three tensors. The best known lower bound for an $n \times n \times n$ tensor is $3n$ due to Alexeev, Forbes and Tsimmerman [AFT11].

we impose the much more stringent condition of $O(n^2)$-wise independence. While this is a significant disadvantage of our algorithm, the benefit we obtain is the significant improvement in the running time and the relative simplicity of the algorithm and its analysis.

Uniqueness of solutions of PolyProj **.** One important component of theorems 6 and 8 above is the uniqueness of solutions of PolyProj . Let us motivate and make precise the notion of uniqueness we have in mind. For an n-variate polynomial g, let

$$\mathscr{G}_g := \{B \in \mathbb{F}^{(n\times n)*} \;:\; g(B\mathbf{x}) = g(\mathbf{x})\}$$

be *the group of symmetries* of g. (It turns out that for all the families of polynomials listed in section 1.1, their groups of symmetries are well understood and completely characterized). Observe that if $f(\mathbf{x}) = g(A\cdot\mathbf{x}+\mathbf{b})$ then for any $B \in \mathscr{G}_g$ we also have $f(\mathbf{x}) = g(B\cdot A\cdot\mathbf{x}+B\cdot\mathbf{b})$. Let us ask the following question - are these all the ways in which f can be expressed as a projection of g? Let us introduce some terminology to capture this.

Definition 10. *We will say that*

$$f(\mathbf{x}) = g(A\mathbf{x}+\mathbf{b}),$$

is a projection of g in an essentially unique way [6] *if*

$$f(\mathbf{x}) = g(A'\mathbf{x}+\mathbf{b}'),$$

is any other projection from g to f then it holds that there exists a $B \in \mathscr{G}_g$ such that

$$A' = B\cdot A, \quad \text{and } \mathbf{b}' = B\cdot\mathbf{b}.$$

In particular, if f is affinely equivalent to a regular polynomial g (see section 4 for definition of regularity) then this happens in an essentially unique way. As part of the proof of theorems 6 and 8, we show that *random projections* of $\mathsf{Pow}_{n,d}$ and $\mathsf{SPS}_{n,d}$ are essentially unique. We combine this observation with a "project-and-lift technique" from Kaltofen [Kal89], Shpilka [Shp07] while exploiting the appropriate affinely invariant property to get efficient algorithms for solving PolyProj in the situation where the input polynomial f is a random projection of $\mathsf{SPS}_{n,d}$ (for bounded n) or of $\mathsf{Pow}_{n,d}$ (for $d = n^{\Omega(1)}$).

2 NP-hardness of PolyProj

In this section we show that the PolyProj problem is NP-hard under polynomial-time Turing reductions. We give the reduction from the Graph 3-colorability problem. We shall use the following slightly modified definition of the Graph 3-colorability problem.

Definition 11. *Let $G = (V,E)$ be a graph. Let $n_1, n_2, n_3, m_{12}, m_{13}, m_{23} \geq 0$ be nonnegative integers. We will say that G is $(n_1, n_2, n_3, m_{12}, m_{13}, m_{23})$-3-colorable if there is an assignment of a unique color $c_i \in \{1,2,3\}$ to each vertex $i \in V$ satisfying the following conditions.*

(i) $n_1+n_2+n_3 = |V|$ *and* $m_{12}+m_{13}+m_{23} = |E|$

(ii) No two vertices of the same color are adjacent. i.e. if $\{i,j\} \in E$ then $c_i \neq c_j$.

[6]See the remarks at the beginning of section 3.2.1 for some examples and discussion of this notion.

(iii) For each $i \in [3]$, there are n_i vertices of color i. That is, for each $i \in [3]$, we have $|\{j : c_j = i\}| = n_i$

(iv) For each $1 \le i < j \le 3$ there are m_{ij} edges whose two endpoints have colors i and j. That is, for $1 \le i < j \le 3$ $|\{\{k, \ell\} \in E : c_k = i \text{ and } c_\ell = j\}| = m_{ij}$

The corresponding computational problem is the following.

> **Name:** GRAPH 3-COLORABILITY
> **Input:** A graph $G = (V, E)$ and integers $n_1, n_2, n_3, m_{12}, m_{13}, m_{23} \ge 0$.
> **Output:** ACCEPT if and only if G is $(n_1, n_2, n_3, m_{12}, m_{13}, m_{23})$-3-colorable.

This version of GRAPH 3-COLORABILITY is easily seen to be equivalent (under polynomial-time Turing reductions) to the usual definition as in [Kar72]. In particular this is an NP-hard problem. We now give the reduction from POLYPROJ to GRAPH 3-COLORABILITY . Let the graph G have $n = |V|$ vertices. Consider the two polynomials

$$g := \sum_{\{i,j\} \in E} x_i x_j \quad \text{and}$$

$$f := m_{12} \cdot x_1 \cdot x_2 + m_{13} \cdot x_1 \cdot x_3 + m_{23} \cdot x_2 \cdot x_3$$

Suppose the graph has a three coloring satisfying the appropriate constraints and where the i-th vertex gets the color $c_i \in [3]$. Then the natural map $x_i \mapsto x_{c_i}$ gives an affine projection from g to f. Now if we could somehow ensure that any projection map from g to f sent every x_i ($i \in [n]$) to some x_j ($j \in [3]$), then such a map would give a 3-coloring of G as well. We will add some extra higher degree terms to these polynomials in order to ensure that any affine projection from g to f has the desired form.

Theorem 12. *Let $G = (V, E)$ be a graph. Let*

$$g := \left(\sum_{i \in [n]} x_i^{n^2+4n+4} \right) + \left(\sum_{k \in [n]} \sum_{i \in [n]} x_i^{k(n+3)} \right) + \left(\sum_{\{i,j\} \in E} x_i x_j \right)$$

and

$$f := \left(\sum_{i \in [3]} n_i x_i^{n^2+4n+4} \right) + \left(\sum_{k \in [n]} \sum_{i \in [3]} n_i x_i^{k(n+3)} \right) + \left(\sum_{1 \le i < j \le 3} m_{ij} x_i x_j \right)$$

Then the graph G is $(n_1, n_2, n_3, m_{12}, m_{13}, m_{23}) - 3$-colorable if and only if the polynomial f is an affine projection of g.

One direction is easy to see. If G is $(n_1, n_2, n_3, m_{12}, m_{13}, m_{23})$-3-colorable then f is an affine projection of g via the natural map $x_i \mapsto x_{c_i}$ where c_i is the color of vertex i. The converse is the interesting direction. In section 7.1 we prove the that any projection from g to f corresponds to a 3-coloring of the graph G by showing that such a projection has the desired form.

3 Overview of Algorithms

3.1 Overview of algorithms for Polynomial Equivalence

In section 5.1 we show that affine equivalence (and also a slightly more general variant that we call a full rank) reduces to equivalence of polynomials under invertible *linear* transformations. We now undertake the task of devising algorithms for polynomial equivalence in some special cases. We first define a notion that will be useful towards this end.

Definition 13. *Let $f, g \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ be n-variate polynomials. Let $G \leq \mathrm{GL}(n, \mathbb{F})$ be a subgroup of the general linear group. We will say that f is G-*equivalent *to g if there exists an $A \in G$ such that*

$$f(\mathbf{x}) = g(A \cdot \mathbf{x}).$$

We now define three subgroups of $\mathrm{GL}(n, \mathbb{F})$ that will be particularly useful. The first one is group of invertible diagonal matrices which we call $\mathrm{SC}(n, \mathbb{F})$ (simply SC in short). We refer to this subgroup as the group of scaling matrices. Another important subgroup of $\mathrm{GL}(n, \mathbb{F})$ is the group of permutation matrices which we denote by $\mathrm{PM}(n, \mathbb{F})$ (simply PM in short). Finally we denote by $\mathrm{PS}(n, \mathbb{F})$ the subgroup of $\mathrm{GL}(n, \mathbb{F})$ generated by PM and SC.

Overview of equivalence algorithms. Suppose we are given as input an n^2-variate polynomial f which is equivalent to the permanent (respectively the determinant) under the action of $\mathrm{GL}(n^2, \mathbb{F})$ group and we want to determine this equivalence. We will solve this problem in three steps.

Step (1): **Reduction to** PS**-equivalence.** In this step we will exploit the fact that the permanent (resp. the determinant)has a nontrivial lie algebra associated to it (see section 4.3 for the relevant definitions). By analyzing the lie algebra of f we shall compute a linear transformation $A_1 \in \mathrm{GL}(n^2, \mathbb{F})$ such that the polynomial

$$f_1(\mathbf{x}) := f(A_1 \mathbf{x})$$

is PS-equivalent to the permanent (resp. the determinant).

Step (2): **Reduction to** SC**-equivalence.** Given $f_1(\mathbf{x})$ as above we exploit the second-order partial derivatives of the permanent (resp. the determinant) to computation a permutation matrix A_2 such that

$$f_2(\mathbf{x}) := f_1(A_2 \mathbf{x})$$

is SC-equivalent to the permanent (resp. the determinant).

Step (3): **Solving** SC**-equivalence.** In this step, we perform some simple substitutions to determine $\lambda_{11}, \lambda_{12}, \ldots, \lambda_{nn}$ such that

$$f_2(\lambda_{11} x_{11}, \lambda_{12} x_{12}, \ldots, \lambda_{nn} x_{nn})$$

equals the permanent (resp. the determinant) polynomial. Let

$$f_3(\mathbf{x}) := f_2(\lambda_{11} x_{11}, \lambda_{12} x_{12}, \ldots, \lambda_{nn} x_{nn})$$

Step (4): **Verification.** In the last step we verify that the polynomial $f_3(\mathbf{x})$ obtained above does indeed equal the permanent (resp. the determinant) polynomial. In the case of the determinant this step is accomplished easily using the DeMillo-Lipton-Schwarz-Zippel identity testing algorithm. In the case of the permanent a randomized algorithm was obtained by Impagliazzo and Kabanets by exploiting the downward self-reducibility of the permanent [KI04, AvM10].

3.2 Overview of Projection algorithms.

Consider the PolyProj problem: given an m-variate polynomial f and an n-variate polynomial g we want to find affine forms $\ell_1, \ell_2, \ldots, \ell_n$ (if they exist) such that

$$f = g(\ell_1, \ell_2, \ldots, \ell_n). \tag{1}$$

Let

$$\ell_i = \sum_{j \in [m]} a_{ij} x_i + a_{i0}$$

We can think of the a_{ij}'s as unknowns and use equation (1) to write down a set polynomial equations in the a_{ij} which we can then solve to obtain the a_{ij}'s. If $d = \max(\deg(f), \deg(g))$, this will give a $\mathsf{poly}(d^{n(m+1)})$-time algorithm to find the a_{ij}'s. But this of course is much more than polynomial time. If we could somehow obtain a system of polynomial equations in a *constant* number of unknowns, we would be in business. The first idea which has been used quite often in the literature is to effectively ensure that m is a constant by considering a random affine projection of f onto a constant-dimensional space. We will describe this in more detail shortly, but for now assume that m is a constant. But this means that we still have $O(n)$ unknowns so solving a system of polynomial equations is still not feasible. The second step will involve using one of the affinely invariant properties from section 5.2.1 to write down a system of polynomial equations *in constantly many variables* and use the solution of this system to recover the ℓ_i's. We now give some more details.

Step (1): **Projection to t dimensions.** Pick a random invertible matrix $A \in \mathbb{F}^{n \times n}$. Let $\hat{f}(\mathbf{x}) := f(A \cdot \mathbf{x})$. Pick a suitable integer $t \geq 1$ (t is typically a constant). For $k \in [t..n]$, let

$$\pi_k(\hat{f})(y_1, \ldots, y_t) := \hat{f}(\pi_k(x_1), \pi_k(x_2), \ldots, \pi_n(x_n))$$

where $\pi_k : \mathbb{F}[x_1, x_2, \ldots, x_n] \mapsto \mathbb{F}[y_1, \ldots, y_t]$ is a homomorphism defined in the following way:

$$\pi_k(x_i) = \begin{cases} y_i & \text{if } i \in [t-1] \\ y_t & \text{if } i = k \\ 0 & \text{otherwise} \end{cases}$$

Let $f_k(\mathbf{y}) := \pi_k(\hat{f})$. Use the algorithm of proposition 24 to obtain a representation of f_k as a list of $\binom{t+d}{d}$ coefficients.

Step (2): **Solving the t-dimensional problem.** For each $k \in [t..[n]]$, find $\ell_1^{[k]}, \ldots, \ell_n^{[k]}$ such that

$$\pi_k(\hat{f})(y_1, y_2, \ldots, y_t) = g(\ell_1^{[k]}, \ldots, \ell_n^{[k]}).$$

This will typically be done by using a suitable affinely invariant property to formulate a system of equations in constantly many variables whose solutions correspond to the $\ell_i^{[k]}$'s.

Step (3): **'Lifting' the $\ell_i^{[k]}$'s to the ℓ_i's.** In this step, one typically shows that the $\ell_i^{[k]}$'s are unique (say maybe upto scalar multiples and reindexing). Once this is established it is relatively easy to compute the ℓ_i's given the $\ell_i^{[k]}$'s.

Step (4): **Verification.** In this step one uses the DeMillo-Lipton-Schwarz-Zippel lemma to test that the ℓ_i's computed above are a valid solution. That is one verifies the identity

$$f(\mathbf{x}) = g(\ell_1, \ldots, \ell_n).$$

The first and fourth steps of this algorithmic strategy are easy. The third step above can be accomplished by a lemma implicit in the works of Kaltofen [Kal89], Shpilka [Shp07] and Karnin and Shpilka [KS09].

Lemma 14. *Let $\mathscr{G}_g$ be the group of symmetries of g. If $\mathscr{G}_g$ is a subgroup of* $\mathrm{PS}(n, \mathbb{F})$ *(see section 3.1 for the definition of the subgroup* $\mathrm{PS}(n, \mathbb{F})$*) and each $\pi_k(\hat{f})$ is a projection of g in an essentially unique way (in the sense of definition 10) then given the $\ell_i^{[k]}$'s as above one can recover the ℓ_i's in* $\mathsf{poly}(n)$ *time.*

Our contribution here is to show that in certain cases, random projections satisfy the prerequisites of this lemma and that the computations involved in the second step of the overall algorithm given above can also be done efficiently. We will now make a few remarks on the role of uniqueness in these algorithms.

3.2.1 Uniqueness of random Projections

Recall the notion of uniqueness of solutions from definition 10. It turns out that projections of polynomials are usually not unique except for some rare cases. For example, consider $g = \mathsf{SPS}_{1,d} = x_1 \cdot x_2 \cdot \ldots \cdot x_d$. Then an affine projection of g is of the form

$$f = \ell_1 \cdot \ell_2 \cdot \ldots \cdot \ell_d.$$

If f can be expressed as a projection of g in some other way say

$$f = \ell'_1 \cdot \ell'_2 \cdot \ldots \cdot \ell'_d$$

then by unique factorization of polynomials we have that there exists a permutation $\pi \in S_d$ and scalars $\lambda_1, \ldots \lambda_d \in \mathbb{F}$ with $\prod_{i \in [d]} \lambda_i = 1$ such that

$$\ell'_i = \lambda_i \ell_{\pi(i)}.$$

This means that any affine projection of $\mathsf{SPS}_{1,d}$ is in an essentially unique manner. We now make some remarks about the algorithm of section 3.2 and the role of affinely invariant properties therein. To make the ensuing discussion concrete let us assume that g is a member of the power-symmetric family of polynomials, i.e. $g = \mathsf{Pow}_{n,d}$. Recall that lemma 14 allowed us to accomplish step three in polynomial time *assuming uniqueness.* For $\mathsf{Pow}_{n,d}$, the group of symmetries is generated by the permutation matrices and diagonal matrices whose diagonal entries are the d-th roots of unity. Thus if projections of $\mathsf{Pow}_{n,d}$ were essentially unique then it would have meant that whenever

$$\sum_{i\in[n]} \ell_i^d = \sum_{i\in[n]} p_i^d$$

then there exists a permutation $\pi \in S_n$ and integers $e_1, e_2, \ldots, e_n$ such that

$$\forall i \in [n] \quad p_i = \omega^{e_i} \cdot \ell_{\pi(i)},$$

where ω is a primitive d-th root of unity. Unfortunately however this is not true in general and it is easy to find counterexamples. A more valiant work might have characterized algebraically all the situations where uniqueness holds and used that characterization to solve the PolyProj for projections of Pow. Here we do not give such a characterization but take a somewhat cowardly alternative – we show that when the ℓ_i's are *random affine forms* then uniqueness holds. Specifically we show,

Theorem 15. *Let $S \subseteq \mathbb{F}$ be a finite set. If we pick a set of n affine forms $\ell_1, \ldots, \ell_n$ with each coefficient being chosen independently and uniformly at random from S then with probability at least*

$$\left(1 - \frac{2dn}{|S|}\right),$$

the expression

$$f = \ell_1^d + \ell_2^d + \ldots + \ell_n^d$$

is unique in the sense that if f can also be written as

$$f = p_1^d + p_2^d + \ldots + p_n^d$$

then there exists a permutation $\pi \in S_n$ and integers $e_1, e_2, \ldots, e_n$ such that

$$p_i = \omega^{e_i} \ell_{\pi(i)}.$$

Here $\omega \in \mathbb{F}$ is a primitive d-th root of unity and n is any integer satisfying

$$n < \binom{m + d/2 - 1}{d/2}.$$

This theorem combined with an algorithmic solution to Step (2) will give us the required polynomial-time algorithm. Similar comments apply to projections of $\mathsf{SPS}_{n,d}$.

4 Preliminaries

4.1 Notation and terminology

$[n]$ denotes the set $\{1, 2, \ldots, n\}$ while $[m..n]$ denotes $\{m, m+1, \ldots, n\}$.
Homogeneous polynomial. Recall that a polynomial $f(x_1, \ldots, x_n)$ is said to be *homogeneous* of degree d if every monomial with a nonzero coefficient is of degree d. Now, any polynomial $f \in \mathbb{F}[\mathbf{x}]$ of degree d can be uniquely written as

$$f = f^{[d]} + f^{[d-1]} + \ldots + f^{[0]},$$

where each $f^{[i]}$ is homogeneous of degree i. We call $f^{[i]}$ the *homogeneous component of degree* i of f.

Linear Dependence among polynomials: A very useful notion will be the notion of *linear dependencies among polynomials.* We now define this notion.

Definition 16. *Let* $\mathbf{f}(\mathbf{x}) \stackrel{\text{def}}{=} (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x})) \in (\mathbb{F}[\mathbf{x}])^m$ *be an* m*-tuple of polynomials over a field* $\mathbb{F}$*. The set of* $\mathbb{F}$-linear dependencies *in* $\mathbf{f}$*, denoted* $\mathbf{f}^\perp$*, is the set of all vectors* $\mathbf{v} \in \mathbb{F}^m$ *whose inner product with* $\mathbf{f}$ *is the zero polynomial, i.e.,*

$$\mathbf{f}^\perp \stackrel{\text{def}}{=} \Big\{(a_1, \ldots, a_m) \in \mathbb{F}^m : a_1 f_1(\mathbf{x}) + \ldots + a_m f_m(\mathbf{x}) = 0\Big\}$$

If $\mathbf{f}^\perp$ *contains a nonzero vector, then the* f_i*'s are said to be* $\mathbb{F}$*-linearly dependent.*

Note that the set $\mathbf{f}^\perp$ is a linear subspace of $\mathbb{F}^m$. A polynomial $f(x_1, \ldots, x_n) \in \mathbb{F}[x_1, \ldots, x_n]$ is said to be *regular* if its n first order partial derivatives namely

$$\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \ldots, \frac{\partial f}{\partial x_n}$$

are $\mathbb{F}$-linearly independent.

Matrices: $\mathbf{1}_n$ shall denote the $n \times n$ identity matrix. M^T shall denote the transpose of the matrix M. $\mathrm{GL}(n, \mathbb{F})$ (abbreviated simply as $\mathrm{GL}(n)$ when the field $\mathbb{F}$ is clear from context) denotes the general linear group of order n over $\mathbb{F}$ (i.e. the group of invertible $n \times n$ matrices over the field $\mathbb{F}$). Similarly, $\mathrm{SL}(n, \mathbb{F})$ (abbreviated $\mathrm{SL}(n)$) denotes the special linear group, i.e. the group of *unimodular* $n \times n$ matrices (i.e. matrices with determinant 1) over the field $\mathbb{F}$.

4.2 Algorithmic preliminaries

Throughout the rest of this article we will assume that an input polynomial is given to us as a 'black box' – we have access to an oracle "holding" the polynomial $f(\mathbf{x})$ so that for any point $\mathbf{a} \in \mathbb{F}^n$, we can obtain $f(\mathbf{a})$ in a single step by querying this oracle. This representation of an input polynomial is in some sense the weakest representation for which one can hope to have efficient algorithms and it subsumes all other representations such as arithmetic circuits. We now recall a few preliminary algorithmic tasks that can be accomplished on a polynomial given as a black box.

4.2.1 Linear dependencies among polynomials

In many of our applications, we will want to efficiently compute a basis of $\mathbf{f}^\perp$ for a given tuple $\mathbf{f} = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))$ of polynomials. Let us capture this as a computational problem.

Definition 17. *The problem of computing linear dependencies between polynomials, denoted* PolyDep *, is defined to be the following computational problem: given as input* m *polynomials* $f_1(\mathbf{x}), \ldots, f_m(\mathbf{x})$ *respectively, output a basis for the subspace* $\mathbf{f}^\perp = (f_1(\mathbf{x}), \ldots, f_m(\mathbf{x}))^\perp \subseteq \mathbb{F}^m$*.*

PolyDep admits an efficient randomized algorithm (see for example [Kay11] for a proof). This randomized algorithm will form a basic building block of our algorithms

Lemma 18. *Let* $\mathbf{f} = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x}))$ *be an* m*-tuple of* n*-variate polynomials. Let*

$$\mathcal{P} := \{\mathbf{a}_i \ : \ i \in [m]\} \subset \mathbb{F}^n$$

be a set of m *points in* $\mathbb{F}^n$*. Consider the* $m \times m$ *matrix*

$$M := (f_j(\mathbf{a}_i))_{i,j \in [m]} .$$

With high probability over a random choice of $\mathcal{P}$*, the nullspace of* M *consists precisely of all the vectors* $(\alpha_1, \alpha_2, \ldots, \alpha_m) \in \mathbb{F}^m$ *such that*

$$\sum_{i \in [m]} \alpha_i f_i(\mathbf{x}) = 0.$$

We get the algorithmic consequence as a corollary.

Corollary 19. *Given a vector of* m *polynomials* $\mathbf{f} = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_m(\mathbf{x}))$*, we can compute a basis for the space* $\mathbf{f}^{\perp}$ *in randomized polynomial time.*

4.2.2 Eliminating redundant variables

Definition 20. *Let* $f(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ *be a polynomial. We will say that* $f(\mathbf{x})$ *is independent of a variable* x_i *if no monomial of* $f(\mathbf{x})$ *contains* x_i*. We will say that* the number of essential variables *in* $f(\mathbf{x})$ *is* t *if we can make an invertible linear* $A \in \mathbb{F}^{(n \times n)*}$ *transformation on the variables such that* $f(A \cdot \mathbf{x})$ *depends on only* t *variables* $x_1, \ldots, x_t$*. The remaining* $(n - t)$ *variables* $x_{t+1}, \ldots, x_n$ *are said to be* redundant *variables. We will say that* $f(\mathbf{x})$ *is* regular *if it has no redundant variables.*

We have:

Lemma 21. *Given a polynomial* $f(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ *with* m *essential variables, we can compute in randomized polynomial time an invertible linear transformation* $A \in \mathbb{F}^{(n \times n)*}$ *such that* $f(A \cdot \mathbf{x})$ *depends on the first* m *variables only.*

4.2.3 Obtaining the derivatives

Proposition 22. *Let*

$$f(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$$

be an n*-variate polynomial of degree* d*. Given black box access to* f*, in time* $\mathsf{poly}(dn)$*, we obtain black box access to any derivative* $\frac{\partial f}{\partial x_i}$ *of* f*.*

See section 7.2 for a proof.

4.2.4 Obtaining the homogeneous components

The following observation says that given black box access to a polynomial, we can obtain black box access to all its homogeneous components in randomized polynomial time.

Proposition 23. *Let*

$$f(\mathbf{x}) = f^{[d]}(\mathbf{x}) + f^{[d-1]}(\mathbf{x}) + \ldots + f^{[0]}(\mathbf{x})$$

be a polynomial of degree d*. Given blackbox access to* $f(\mathbf{x})$ *and a point* $\mathbf{a} \in \mathbb{F}^n$*, we can compute* $f^{[i]}(\mathbf{a})$ *for each* $i \in [0..d]$ *in polynomial time.*

See section 7.2 for a proof.

4.2.5 Interpolating on a constant dimensional affine subspace

The following well-known proposition is used extensively in dealing with low-degree multivariate polynomials.

Proposition 24. *Given blackbox access to a polynomial f defined on a vector space V, one can express the restriction of f to any constant-dimensional subspace $U \subseteq V$ as a sum of coefficients in polynomial-time.*

4.3 Stabilizers and Lie Algebras

We now discuss some basic concepts that will be required later. We give an overview of the basic notions about lie algebras and fix the relevant notation. Let $f(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ be a polynomial. The *group of symmetries of* f denoted $\mathscr{G}_f$ is the set of all *invertible* $n \times n$ matrices A such that $f(A\mathbf{x}) = f(\mathbf{x})$. [7] When the polynomial f is clear from context we denote $\mathscr{G}_f$ simply by $\mathscr{G}$.
It turns out that for any polynomial f, its group of symmetries $\mathscr{G}_f$ is a closed subgroup of $\mathbb{F}^{(n\times n)*}$, in other words it is a *matrix lie group* (cf. [Kir08], theorem 3.26). We refer the interested reader to the texts by [Kir08, Hal07] for more information about lie groups and proper formal definitions. It means that we can view the group $\mathscr{G}_f$ as a manifold in the space $\mathbb{F}^{n\times n}$. The corresponding lie algebra, denoted $\mathfrak{g}_f$ is the subspace of $\mathbb{F}^{n\times n}$ tangent to $\mathscr{G}_f$ at the point $\mathbf{1}_n$ (the identity matrix). For most polynomials f, $\mathscr{G}_f$ is trivial, i.e. $\mathscr{G}_f$ consists of only the identity matrix. For some polynomials $\mathscr{G}_f$ is nontrivial but the lie algebra $\mathfrak{g}_f$ is trivial (i.e. it consists only of the zero matrix). Such a $\mathscr{G}_f$ is said to be a discrete group. For example, the polynomials $\mathsf{Pow}_{n,d}$ and $\mathsf{Sym}_{n,d}$ have a nontrivial discrete symmetry group. [8] In this paper we will be concerned with polynomials f for which $\mathscr{G}_f$ is continuous, i.e. polynomials f for which $\mathfrak{g}_f$ is nontrivial. In particular, the symmetries of the permanent (determinant) form a continous group. We will exploit the rich structure of the associated lie algebras to determine the equivalence of a given polynomial to the permanent (determinant). We shall be using the following equivalent definition of $\mathfrak{g}_f$.

Definition 25. *Let $f(x_1, \ldots, x_n) \in \mathbb{F}[x_1, \ldots x_n]$ be an n-variate polynomial. Let ϵ be a formal variable with $\epsilon^2 = 0$. Then $\mathfrak{g}_f$ is defined to be the set of all matrices $A \in \mathbb{F}^{n\times n}$ such that*

$$f((\mathbf{1}_n + \epsilon A)\mathbf{x}) = f(\mathbf{x}). \tag{2}$$

The reader is referred to [Hal07], definition 2.15 for a more proper definition of the lie algebra and then to [Hal07], theorem 2.27 for equivalence of the two definitions. We begin our algorithmic quest by noting that (a basis for) the lie algebra of any given polynomial can be computed efficiently.

Lemma 26. *Given an n-variate polynomial $f(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ (as a blackbox), a basis for the lie algebra of its group of symmetries can be computed in randomized polynomial time.*

See section 7.3 for a proof. In sections 6.1 and 6.2 we will see an explcit set of basis elements for the lie algebras of the permanent and determinant respectively. For now, let us return to the general description. Now suppose that $g(\mathbf{x}) = f(B \cdot \mathbf{x})$, for some invertible matrix B. Then the lie algebra of g is a conjugate of the lie algebra of f via the matrix B (Proposition 58). That is,

$$\mathfrak{g}_g = \{B^{-1}AB : A \in \mathfrak{g}_f\} \tag{3}$$

[7] The group of symmetries of a polynomial is also referred to in the literature as the stabilizer, the group of automorphisms, the group of isomorphisms or as the isotropy subgroup of f.

[8] For these polynomials, a randomized polynomial time algorithm for testing equivalence was presented in [Kay11].

It is easy to see that the converse however is not true in general. That is, it can happen that (3) holds for some matrix B but $f(B\mathbf{x})$ does not equal $g(\mathbf{x})$. We will see that understanding the structure of $\mathfrak{g}_f$ for a given f will nevertheless help us in determining the equivalence of f to the permanent and/or the determinant. Let us now recall the following important fact about lie algebras.

Fact 27. *Let $A, B \in \mathfrak{g}$. Then $[A, B] := (AB - BA) \in \mathfrak{g}$.*

Lie Algebraic Concepts. Let us now fix the notation for some basic concepts from the theory of lie algebras. Let $\mathfrak{g} \subseteq \mathbb{F}^{m\times m}$ be a lie algebra. The *centralizer* of an element $A \in \mathfrak{g}$, denoted $\mathrm{Cent}(A)$ is the set of all elements $X \in \mathfrak{g}$ such that

$$[A, X] = 0.$$

Fact 28. *For every $A \in \mathfrak{g}$, Cent(A) is a subspace of $\mathfrak{g}$ and can be computed efficiently, i.e. in* $\mathsf{poly}(m)$*-time.*

We say that the lie algebra $\mathfrak{g}$ is the *direct sum* of two subalgebras $\mathfrak{g}_1$ and $\mathfrak{g}_2$, denoted

$$\mathfrak{g} = \mathfrak{g}_1 \oplus \mathfrak{g}_2,$$

if $\mathfrak{g}$ is direct sum of $\mathfrak{g}_1$ and $\mathfrak{g}_2$ as a vector space and moreover that

$$[A, B] = 0 \quad \forall A \in \mathfrak{g}_1, B \in \mathfrak{g}_2.$$

$\mathfrak{g}$ is said to be *nilpotent* if the lower central series namely

$$\mathfrak{g} > [\mathfrak{g}, \mathfrak{g}] > [[\mathfrak{g}, \mathfrak{g}], \mathfrak{g}] > \ldots$$

becomes zero eventually. The normalizer of a subalgebra $\mathfrak{h}$ of $\mathfrak{g}$ is the set of all $X \in \mathfrak{g}$ such that $[X, \mathfrak{h}] \subseteq \mathfrak{h}$ A Cartan subalgebra of $\mathfrak{g}$ is a nilpotent subalgebra equal to its own normalizer.
Properties of $\mathrm{SL}(n, \mathbb{F})$ and $\mathfrak{sl}_n$. The lie algebra of the special linear group $\mathrm{SL}(n, \mathbb{F})$ we denote by $\mathfrak{sl}_n$. It consists of all $n \times n$ matrices with trace zero. We shall need the following fact about the Cartan subalgebras of $\mathfrak{sl}_n$.

Fact 29. *The subalgebra $\mathfrak{h}$ consisting of traceless diagonal matrices is a Cartan subalgebra of $\mathfrak{sl}_n$. Every other Cartan subalgebra of $\mathfrak{sl}_n$ is a conjugate (via an element of SL_n) of this Cartan subalgebra $\mathfrak{h}$.*

5 Preliminary Observations

5.1 Full rank projections versus polynomial equivalence

In this section consider a special case of the polynomial projection problem which we refer to as the 'full rank projection problem'. It is defined as follows.

> **Name:** FullRankProj
> **Input:** Polynomials $f(x_1, x_2, \ldots, x_m)$ and $g(x_1, x_2, \ldots, x_n)$ over the field $\mathbb{F}$.
> **Output:** An $n \times m$ matrix A *of rank* n and a vector $\mathbf{b} \in \mathbb{F}^n$ such that $f(\mathbf{x}) = g(A\mathbf{x} + \mathbf{b})$, if such an A and $\mathbf{b}$ exist. Else output `'No such projection exists'`.

A further special case of FullRankProj which has been studied much more extensively is the polynomial equivalence problem defined below.

> **Name:** PolyEquiv
> **Input:** Polynomials $f(x_1, x_2, \ldots, x_n)$ and $g(x_1, x_2, \ldots, x_n)$ over the field $\mathbb{F}$.
> **Output:** An invertible $n \times n$ matrix A such that $f(\mathbf{x}) = g(A\mathbf{x})$, if such an A exists. Else output `'No such equivalence exists'`.

We will first show that if the polynomial g satisfies some (relatively mild) conditions, then the full rank projection problem for g reduces to the equivalence problem for g. Specifically,

Theorem 30. *Let $g(x_1, \ldots, x_n)$ be a polynomial. Suppose that*

1. *g is homogeneous of degree $d \geq 3$.*
2. *g is a regular polynomial. Moreover, we have access to a set*

$$\{\mathbf{a}_i \in \mathbb{F}^n \ : \ i \in [n]\}$$

such that

   (a) *The matrix*

$$M := \left(\frac{\partial g}{\partial x_i}(\mathbf{a}_j) \right)_{n \times n}$$

   is of full rank.

   (b) *The entries of the matrix M are known to us. In other words, we have access to*

$$\frac{\partial g}{\partial x_i}(\mathbf{a}_j) \quad \text{for each } i, j \in [n].$$

Then determining whether a given m-variate polynomial f is a full rank projection of g is equivalent (under randomized polynomial-time Turing reductions) to determining whether a given f is equivalent to g.

In proving this reduction, it will help us conceptually to introduce one more special case of FullRankProj .

> **Name:** Translation
> **Input:** Polynomials $f(x_1, x_2, \ldots, x_n)$ and $g(x_1, x_2, \ldots, x_n)$ over the field $\mathbb{F}$.
> **Output:** A vector $\mathbf{a} = (a_1, a_2, \ldots, a_n) \in \mathbb{F}^n$ such that
>
> $$f(x_1, x_2, \ldots, x_n) = g(x_1 + a_1, x_2 + a_2, \ldots, x_n + a_n),$$
>
> ($f(\mathbf{x}) = g(\mathbf{x} + \mathbf{a})$ in short) if such a vector $\mathbf{a}$ exists. Else output `'No such translation exists'`.

An affine map $\mathbf{x} \mapsto A \cdot \mathbf{x} + \mathbf{b}$ with $\text{rank}(A) = n$ has roughly has three 'components':

(i) An invertible linear transformation

(ii) A translation

(iii) The introduction of 'redundant variables'

The proof of theorem 30 then goes as follows: the redundant variables are eliminated by lemma 21, and the translation is obtained by applying the algorithm of lemma 19 to the first-order partial derivatives of g. In ths way the full rank projection problem boils down to the polynomial equivalence problem. The details of the proof of theorem 30 is in section 7.3.
Most of the families of polynomials listed in section 1 are easily seen to satisfy the assumptions of the above theorem so that for these families of polynomials, the full rank projection problem reduces to the polynomial equivalence problem. We first give the proof for the determinant.

Corollary 31. *The full rank projection problem for the determinant reduces to the equivalence problem for the determinant.*

Proof. By definition the determinant Det_n is homogeneous of degree n. Every first order derivative of the determinant is (upto a sign) the determinant of an $(n-1) \times (n-1)$ sized minor. The set of monomials occuring in the minors of these $(n-1) \times (n-1)$ determinants are disjoint so that the first order derivatives are $\mathbb{F}$-linearly independent. Hence Det_n is a regular polynomial. Finally, by lemma 18, a random set of n^2 points in $\mathbb{F}^{n \times n}$ satisfies property 2(a). Finally, we can evaluate the subdeterminants at these n^2 points in polynomial time and satisfy property 2(b) as well so that all the conditions of theorem 30 are fulfilled. □

In section 6.2 we will present a randomized polynomial time algorithm for determinantal equivalence. In a manner similar to above, the full rank projection problem for the power symmetric polynomials and the sum of products polynomial reduces to the corresponding equivalence problem. For the elementary symmetric polynomial $\mathsf{Sym}_{n,d}$ the $\mathbb{F}$-linear independence of its first order derivatives follows from the nonsingularity of an appropriate matrix [KN97](pp. 22-23). For each of $\mathsf{Pow}, \mathsf{SPS}$ and Sym, a randomized polynomial time algorithm for determining equivalence is given in [Kay11]. We thus have:

Corollary 32. *There is a randomized algorithm with running time* $\mathsf{poly}(mnd)$ *to determine whether a given m-variate polynomial is a full rank projection of* $\mathsf{Sym}_{n,d}$.

Corollary 33. *There is a randomized algorithm with running time* $\mathsf{poly}(mnd)$ *to determine whether a given m-variate polynomial is a full rank projection of* $\mathsf{Pow}_{n,d}$.

Corollary 34. *There is a randomized algorithm with running time* $\mathsf{poly}(mnd)$ *to determine whether a given m-variate polynomial is a full rank projection of* $\mathsf{SPS}_{n,d}$.

The case of the permanent is a little more involved because we do not know how to compute the permanent efficiently at a randomly chosen point. Fortunately, we can overcome this hurdle. Note that it is easy to compute the permanent of a diagonal matrix - the value of the permanent is simply the product of the entries along the diagonal. Moreover, we can also compute the permanent of a matrix which is obtained from a diagonal matrix by permuting the rows or columns in some arbitary way. It turns out that such matrices yiels a set of points satisfying the requirement 2(b) of theorem 30.

Proposition 35. *Let $\lambda_1, \lambda_2, \ldots, \lambda_n$ be some n distinct nonzero elements of $\mathbb{F}$. For $1 \le i, j \le n$ let the matrix $P_{ij} \in \mathbb{F}^{n\times n}$ be defined as*

$$(P_{ij})_{k\ell} \stackrel{\text{def}}{=} \begin{cases} \lambda_i^k & \text{if } \ell - k = j - 1 \\ 0 & \text{otherwise} \end{cases}$$

Then the matrix $M \in \mathbb{F}^{n^2\times n^2}$,

$$M_{(i,j),(k,\ell)} = \frac{\partial \mathsf{Perm}}{\partial x_{k\ell}} (P_{ij})$$

has rank n^2.

The matrix M in the proposition above is essentially a block-diagonal matrix with the blocks being Vandermonde matrices so that it has full rank. The details are in section 7.3. As discussed above this proposition implies that the full rank projection problem for the permanent reduces to the permanent equivalence problem.

Corollary 36. *The full rank projection problem for the permanent reduces to the equivalence problem for the permanent.*

In section 6.1 we will present a randomized polynomial time algorithm for permanental equivalence. In order to do this we shall follow the overall strategy given in section 3.1 reducing GL-equivalence to PS-equivalence to SC-equivalence and then finally solving this last problem by making some well-chosen substitutions. Let us record a fact about the group PS.

Fact 37. *The group* PS *is a semidirect product of* SC *and* PM *with* SC *being the normal subgroup. In particular every element $A \in$ PS can uniquely be written as*

$$A = B \cdot C \quad \text{for some } B \in \mathrm{SC} \text{ and } C \in \mathrm{PM}.$$

5.2 Affinely invariant properties and lower bounds

Over the years there have been several notable proofs of the form $f \not\le_{\text{aff}} g$ for certain pairs f and g. Typically this has been done through the discovery of an appropriate "affinely invariant property". We first give a quick survey of such results pointing out some of the important affinely invariant properties that have been discovered. We then show how the mathematical ideas underlying some of these affinely invariant properties can be exploited to design efficient algorithms for two special cases of PolyProj - random projections of the $\mathsf{Pow}_{n,d}$ and $\mathsf{SPS}_{n,d}$ polynmials. In this paper we shall show:

1. Given blackbox access to a random m-variate affine projection f of $\mathsf{Pow}_{n,d}$, we can find the projection efficiently *assuming* $d = n^{\Omega(1)}$
2. Given blackbox access to a random m-variate affine projection f of $\mathsf{SPS}_{n,d}$, we can find the projection efficiently *assuming* $n = O(1)$ *and* $d, m > n^2 + n$.

We refer the reader to section 6.3 (respectively section 6.4) respectively for more precise statements of these results, some background and a discussion of the hardness of these problems if the relevant restrictions on d and n are removed. We now give an overview of these algorithms focusing on the two major ingredients involved - exploitation of a relevant affinely invariant property and uniqueness of solutions.

5.2.1 Affinely Invariant Properties and lower bounds.

Recall the notion of an affinely invariant property from definition 5. One example is the degree of a polynomial. Another example is the number of essential variables in a polynomial (see definition 20) . Yet another example of an affinely invariant property is as follows.

$$\mathrm{Ir}(f) := \deg(f) - \text{number of irreducible factors of } f$$

We now list some more examples of known affinely invariant properties which have found applications to lower bound proofs.

(I) **Dimension of k-th order Partial Derivatives:** denoted $\dim(\partial^k(f))$, it is the number of $\mathbb{F}$-linearly independent polynomials in $\partial^k(f)$, where $\partial^k(f) \subseteq \mathbb{F}[\mathbf{x}]$ is the set of k-th order partial derivatives of f. First discovered/used by Nisan and Wigderson [NW97], it has the following applications.

(1) (cf. the survey by Wigderson [Wig02]):

$$\mathsf{Det}_n \not\leq_{\mathrm{aff}} \mathsf{SPS}_{t,d} \;\text{ unless } (t2^d) \geq \binom{2n}{n}$$

(2) (cf. the survey by Chen, Kayal, Wigderson [CKW11])

$$\mathsf{SPS}_{1,n} \not\leq_{\mathrm{aff}} \mathsf{Pow}_{t,d} \;\text{ unless } (t \cdot d) \geq 2^n$$

(II) **Minimal codimension of a vanishing subspace:** denoted by $\mathrm{Va}(f)$, it is defined as

$$\mathrm{Va}(f) := \max\{\mathrm{codim}(H) \;:\; H \text{ is a vanishing subspace of } f\},$$

where an affine subspace H of $\mathbb{F}^m$ is said to be a vanishing subspace $f(\mathbf{x})$ if $f(\mathbf{a}) = 0$ for every $\mathbf{a} \in H$. It has the following applications.

(1) (Shpilka and Wigderson [SW01]):

$$\mathsf{Sym}_{n,\frac{n}{2}} \not\leq_{\mathrm{aff}} \mathsf{SPS}_{t,d} \;\text{ unless } t \geq n \;\text{ (for any } d)$$

(2) (Folklore)[9]:

$$\mathsf{Det}_n \not\leq_{\mathrm{aff}} \mathsf{SPS}_{t,d} \;\text{ unless } t \geq n \;\text{ (for any } d)$$

(III) **Rank of the Hessian at a zero:** denoted $\mathrm{Hz}(f)$, it is defined as

$$\mathrm{Hz}(f) := \min\{\mathrm{rank}(H_f(\mathbf{a})) : \mathbf{a} \in \mathbb{F}^n \;\text{ satisfies } f(\mathbf{a}) = 0\}$$

where $H_f(\mathbf{x}) \in (\mathbb{F}[\mathbf{x}])^{n \times n}$ is the *Hessian* of f defined as follows:

$$H_f(\mathbf{x}) \stackrel{\mathrm{def}}{=} \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \cdot \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_1 \cdot \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \cdot \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n \cdot \partial x_n} \end{bmatrix}.$$

It has the following application.

[9] A proof was comunicated to the author by Srikanth Srinivasan

(1) (Mignon and Ressayre [MR04]):

$$\mathsf{Perm}_n \not\leq_{\text{aff}} \mathsf{Det}_m \text{ unless } m \geq \frac{n^2}{2}$$

In sections 6.3 and 6.4 we will see how the first two affinely invariant properties mentioned above can be exploited to find projections of the power symmetric polynomial and the sum-of-products polynomial respectively. In order to do this we follow the overall algorithm given in section 3.2.

6 Algorithms for the special cases

6.1 The case of the Permanent polynomial

In this section we flesh out the details of the steps outlined in section 3.1 for the problem of deciding whether a given polynomial is equivalent to the permanent. Our goal is to prove theorem 2. As we have already noted in section 3.1, step 4 (verification) of the algorithm outline can be accomplished in randomized polynomial time using the downward self-reducibility of the permanent as given by Impagliazzo and Kabanets [KI04]. We now describe how to accomplish the first three steps of the algorithm outline of section 3.1. Towards this end, let us recall the characterization of $\mathscr{G}_{\mathsf{Perm}}$ proved by Marcus and May [MM62] [10].

Theorem 38. *If $T \in \mathscr{G}_{\mathsf{Perm}_n}$ and if $n > 2$ then there exist permutation matrices P and Q and diagonal matrices D and L such that $\mathsf{Perm}(D) = \mathsf{Perm}(L) = 1$ and either $T \cdot X = DPXQL$ or $T(X) = DP(X^T)QL$.*

While the proof techniques of [MM62,Bot67] do not give an algorithm for the polynomial equivalence problem, nevertheless this theorem yields a great deal of insight into the lie algebra of the permanent and guides the design of our algorithm. Recall that the lie algebra corresponds to the tangent space of the manifold $\mathscr{G}_{\mathsf{Perm}}$ at the identity. Thus the dimension of the lie algebra $\mathfrak{g}_{\mathsf{Perm}}$ is the dimension of the manifold $\mathscr{G}_{\mathsf{Perm}}$, which intuitively is the number of "continuous degrees of freedom" in $\mathscr{G}_{\mathsf{Perm}}$. As far the continuous part is concerned the permutation matrices dont matter so that the dimension is essentially determined by the diagonal matrices D and L. These satisfy $\mathsf{Perm}(D) = \mathsf{Perm}(L) = 1$ so that we have $(2n-2)$ degrees of freedom. Thus we have

Proposition 39. *The lie algebra of the permanent, $\mathfrak{g}_{\mathsf{Perm}}$ has a basis of size $(2n-2)$ consisting of the following matrices:*

- *$(n-1)$ matrices $R_2, R_3, \ldots, R_n$ where for each $k \in [2..n]$,*

$$(R_k)_{(i_1,j_1),(i_2,j_2)} := \begin{cases} 1 & if\ (i_1, j_1) = (i_2, j_2) \text{ and } i_1 = i_2 = 1 \\ -1 & if\ (i_1, j_1) = (i_2, j_2) \text{ and } i_1 = i_2 = k \\ 0 & otherwise \end{cases}$$

  Intuitively each R_k corresponds to multiplying the first row by some scalar λ and multiplying the k-th row by λ^{-1}.

[10] a simpler proof is given by Peter Botta [Bot67]

- $(n-1)$ *matrices* $C_2, C_3, \ldots, C_n$ *where for each* $k \in [2..n]$,

$$(C_k)_{(i_1,j_1),(i_2,j_2)} := \begin{cases} 1 & if\ (i_1,j_1) = (i_2,j_2) \text{ and } j_1 = j_2 = 1 \\ -1 & if\ (i_1,j_1) = (i_2,j_2) \text{ and } j_1 = j_2 = k \\ 0 & otherwise \end{cases}$$

Intuitively each C_k *corresponds to multiplying the first column by some scalar* λ *and multiplying the* k*-th column by* λ^{-1}.

It is readily verified that the set of $(2n-2)$ matrices described in the statement above are linearly independent and indeed are in $\mathfrak{g}_{\mathsf{Perm}}$. The argument sketched above (which can be made more formal and precise) saying that dimension of $\mathfrak{g}_{\mathsf{Perm}}$ is $(2n-2)$ means that the matrices described above for a basis of $\mathfrak{g}_{\mathsf{Perm}}$ as well. [11] Note that all the basis elements described above are diagonal matrices. This will help us accomplish step (1) of the outline described in section 3.1.

6.1.1 Step 1: Reduction to PS-equivalence

This is the most important step in determining whether a given polynomial is equivalent to the permanent. Here is the description of the algorithm of this step.

Input: Blackbox access to an n^2-variate polynomial f.
Output: A matrix $D \in \mathrm{GL}(n^2, \mathbb{F})$ having the property that if $f(\mathbf{x})$ is $\mathrm{GL}(n^2, \mathbb{F})$-equivalent to Perm_n then $f(D \cdot \mathbf{x})$ is PS-equivalent to Perm_n.
Algorithm:

Step (i) Using the algorithm of lemma 26 compute a basis $A_1, A_2, \ldots A_k \in \mathrm{GL}(n^2, \mathbb{F})$ for $\mathfrak{g}_f$. If the dimension k of this lie algebra is different from $(2n-2)$ then output 'f is not equivalent to Perm'.

Step (ii) Compute a matrix D which simultaneously diagonalizes $A_1, A_2, \ldots, A_{2n-2}$. Specifically compute a matrix D such that

$$D^{-1} A_i D$$

is a diagonal matrix for each $i \in [2n-2]$. If no such D exists then output 'f is not equivalent to Perm' else output D.

The second step of this algorithm, viz. simultaneous diagonalization of a set of (commuting) matrices is a standard linear algebra computation and can easily be accomplished in $\mathsf{poly}(n)$ time. For example, in this case one can pick a random matrix $A \in \mathfrak{g}_f$ and diagonalize it, i.e. find D such that $D^{-1} \cdot A \cdot D$ is diagonal. Overall therefore the time complexity is clearly $\mathsf{poly}(n)$. The correctness of the algorithm above is encapsulated in the following proposition whose proof is in section 7.4.

Proposition 40. *If* $f(\mathbf{x})$ *is* $\mathrm{GL}(n^2, \mathbb{F})$*-equivalent to* Perm *then* $f(D \cdot \mathbf{x})$ *is* PS*-equivalent to* Perm.

[11] A more direct proof of proposition 39 can be had by following the proof of lemma 26, claim 59 in particular, and doing the relevant computations.

6.1.2 Step 2: Reduction to SC-equivalence

In this step, our problem is the following: given a polynomial $f(\mathbf{x})$ which is PS-equivalent to the permanent, we want to find a permutation matrix $P \in \mathbb{F}^{n^2 \times n^2}$ such that $f(P\mathbf{x})$ is SC-equivalent to the permanent. In other words f is of the form

$$f(\mathbf{x}) = \mathsf{Perm}_n(\lambda_{11}x_{\pi(1,1)}, \lambda_{12}x_{\pi(1,2)}, \ldots, \lambda_{nn}x_{\pi(n,n)})$$

for some unknown permutation $\pi : [n] \times [n] \mapsto [n] \times [n]$ and some unknown nonzro scalars λ_{ij}. We will now use the following fact about second order partial derivatives of the permanent.

$$\frac{\partial^2 \mathsf{Perm}}{\partial x_{ij} \cdot \partial x_{k\ell}} \begin{cases} = 0 & \text{if } i = k \quad \text{or } j = \ell \\ \neq 0 & \text{otherwise} \end{cases} \tag{4}$$

In other words the second order derivative of Perm with respect to variables x_{ij} and $x_{k\ell}$ is zero if and only if (i, j) and (k, ℓ) agree on at least one coordinate. It immediately implies that

$$\frac{\partial^2 f}{\partial x_{ij} \cdot \partial x_{k\ell}} \tag{5}$$

is zero precisely when $\pi^{-1}(i, j)$ and $\pi^{-1}(k, \ell)$ agree on at least one coordinate. This observation can be used to rearrange the n^2 variables of f into an $n \times n$ matrix.

Proposition 41. *Let*

$$\delta_{ij,k\ell} = \begin{cases} 0 & \text{if } \frac{\partial^2 f}{\partial x_{ij} \cdot \partial x_{k\ell}} = 0 \\ 1 & \text{otherwise} \end{cases} \tag{6}$$

Then given the set $\{\delta_{ij,k\ell} \ : \ i, j, k, \ell \in [n]\}$ we can determine (in $O(n^4)$ time) a permutation $\sigma : ([n] \times [n]) \mapsto ([n] \times [n])$ such that the polynomial

$$f_2(\mathbf{x}) := f(x_{\sigma(1,1)}, x_{\sigma(1,2)}, \ldots, x_{\sigma(n,n)})$$

is SC-*equivalent to the permanent.*

6.1.3 Step 3: Solving SC-equivalence

In this step, our problem is the following: given a polynomial $f(\mathbf{x}) \in \mathbb{F}[x_{11}, \ldots, x_{nn}]$ find $\lambda_{11}, \lambda_{12}, \ldots, \lambda_{nn}$ such that

$$f(\mathbf{x}) = \mathsf{Perm}_n(\lambda_{11}x_{11}, \lambda_{12}x_{12}, \ldots, \lambda_{nn}x_{nn}). \tag{7}$$

The idea is that the λ_{ij}'s can be obtained by evaluating f on certain well-chosen points (matrices). Consider $f(\mathbf{1}_n)$ (recall that $\mathbf{1}$ is the $n \times n$ identity matrix). From (7) we have

$$f(\mathbf{1}_n) = \lambda_{11} \cdot \lambda_{22} \cdot \ldots \cdot \lambda_{nn}.$$

Thus evaluating f at $\mathbf{1}_n$ gives us the product of the λ_{ii}'s. More generally, evaluating f at a permutation matrix will give us the product of some subset of λ_{ij}'s. We will now see that evaluating f on a small, well-chosen set of permutation matrices helps us recover all the λ_{ij}'s.

Proposition 42. *There exists an explicit set S of $O(n^2)$ permutation matrices in $\mathbb{F}^{n \times n}$ such that knowing $f(\mathbf{a})$ for each $\mathbf{a} \in S$ allows us to determine the λ_{ij}'s in $O(n^2)$ time.*

6.2 The case of the Determinant polynomial

In this section we consider the problem of deciding whether a given polynomial is equivalent to the determinant. Our goal is to prove theorem 4. We follow the steps outlined in section 3.1 for this problem. As one might expect, steps two to four are very similar for the permanent as well as the determinant and we avoid repetition by omitting the details. We now focus only on the first step. Towards this end, let us recall the characterization of $\mathscr{G}_{\mathsf{Det}_n}$.

Theorem 43. *If $T \in \mathscr{G}_{\mathsf{Det}_n}$ then there exist matrices P and Q in $SL(n, \mathbb{F})$ such that either $T(X) = PXQ$ or $T(X) = P(X^T)Q$.*

An accessible proof can be found in Marcus and Moyls [MM59]. This theorem was first proved by Frobenius [Fro97] and was subsequently rediscovered, sometimes with easier proofs by Kantor [Kan97], Schur [Sch25], Morita [Mor44], Dieudonné [Die48], Marcus and Purves [MP59], Marcus and May [MM62]. The techniques of these results do not appear to be directly applicable for our algorithmic purposes [12]. Nevertheless it lays bare the structure of $\mathscr{G}_{\mathsf{Det}}$, and therefore also of $\mathfrak{g}_{\mathsf{Det}}$, and in doing so it guides the design of the algorithm.

Corollary 44. *The group of symmetries of the determinant polynomial, $\mathscr{G}_{\mathsf{Det}}$ is isomorphic to a semidproduct of S_2 with $\mathrm{SL}(n, \mathbb{F}) \times \mathrm{SL}(n, \mathbb{F})$. The corresponding lie algebra $\mathfrak{g}_{\mathsf{Det}}$ is isomorphic to $\mathfrak{sl}_n \oplus \mathfrak{sl}_n$.*

6.2.1 Reduction to PS-equivalence

Here is the description of the algorithm of this step.

Input: Blackbox access to an n^2-variate polynomial f.
Output: A matrix $D \in \mathrm{GL}(n^2, \mathbb{F})$ having the property that if $f(\mathbf{x})$ is $\mathrm{GL}(n^2, \mathbb{F})$-equivalent to Det_n then $f(D \cdot \mathbf{x})$ is PS-equivalent to Det_n.
Algorithm:

Step (i) Using the algorithm of lemma 26 compute a basis $A_1, A_2, \ldots A_k \in \mathrm{GL}(n^2, \mathbb{F})$ for $\mathfrak{g}_f$. If the dimension k of this lie algebra is different from $(2n^2 - 2)$ then output 'f is not equivalent to Det'.

Step (ii) Pick a random element $B \in \mathfrak{g}_f$. Compute a basis for $\mathrm{Cent}(B)$ (by solving a system of homogeneous linear equations, see fact 28). Let

$$X_1, X_2, \ldots, X_k$$

be a basis of $\mathrm{Cent}(B)$. If k is different from $(2n - 2)$ then output 'f is not equivalent to Det'.

Step (iii) Compute a matrix D which simultaneously diagonalizes $X_1, X_2, \ldots, X_{2n-2}$. Specifically compute a matrix D such that

$$D^{-1} X_i D$$

is a diagnoal matrix for each $i \in [2n - 2]$. If no such D exists then output 'f is not equivalent to Perm' else output D.

[12] the notion of rank and the characterization of rank-one matrices are very "basis-dependent"

All the steps of the algorithm above involve straightforward linear algebra and can easily be accomplished in $\mathsf{poly}(n)$ time. The correctness of the algorithm above is encapsulated in the following proposition.

Proposition 45. *Assume that $f(\mathbf{x})$ is $\mathrm{GL}(n^2, \mathbb{F})$-equivalent to Det. Then with high probability over the random choice of the matrix B in step (ii), $f(D \cdot \mathbf{x})$ is PS-equivalent to Det.*

The proof of this proposition is in section 7.5.

6.3 The case of the Power Symmetric polynomial

In this section we look at instances of PolyProj where the input polynomial $f(\mathbf{x})$ is an affine projection of $\mathsf{Pow}_{n,d}$, i.e.

$$f(\mathbf{x}) = \mathsf{Pow}_{n,d}(\ell_1, \ell_2, \ldots, \ell_n),$$

where the ℓ_i's are m-variate affine forms. Our task is to recover the ℓ_i's given blackbox access to f. We follow the algorithm outline given in section 5.2 to design the algorithm of theorem 6.

Proof of theorem 6

We follow the outline given in section 5.2 but handle the two cases ($d > 2n$ and $d \leq 2n$) separately.
Case I: $d > 2n$ In this case we pick $t = 1$ so that our problem essentially becomes the following: given a *univariate* polynomial $f(x)$ find the smallest integer n such that

$$f(x) = (a_1 x + b_1)^d + \ldots (a_n x + b_n)^d$$

This problem can then be solved using the work of Kleppe [Kle99]. Specifically we show.

Proposition 46. *There is a randomized polynomial time-algorithm to determine the smallest n such that a given univariate polynomial f of degree d can be expressed as a sum of n d-th powers of affine forms. Moreover, if $d > 2n$ then such an expression for f, if it exists, is essentially unique.*

The proof of this proposition is given in section 7.6.
Case II: $d \leq 2n$
We follow the algorithm outline given in section 5.2 and choose $t = 2^{\frac{\log n}{\log d}}$. For concreteness let us give the give the exposition assuming $d = n^{\Omega(1)}$ whence the t becomes a constant. To prove the theorem above we just need to prove the uniqueness of random projections of $\mathsf{Pow}_{n,d}$ and show to accomplish the second step of the overall algorithm of section 5.2 in polynomial time. Thus our problem effectively is the same as the problem that we started out with but the number of variables m has reduced to a constant. Also note that if

$$f = \sum_{i \in [n]} \ell_i^d$$

then we can homogenize this expression and assume without loss of generality that f is homogeneous of degree d and the ℓ_i's are linear forms (rather than affine forms). The uniqueness is captured in the following proposition whose proof is given in section 7.6.

Proposition 47. *With probability at least*

$$\left(1 - \frac{2dn}{|S|}\right)$$

over the random choice of the ℓ_i's, the expression

$$f = \ell_1^d + \ell_2^d + \ldots + \ell_n^d$$

is unique in the sense that if we also have

$$f = p_1^d + p_2^d + \ldots + p_n^d$$

then exists a permutation $\pi \in S_n$ and integers $e_1, e_2, \ldots, e_n$ such that

$$p_i = \omega^{e_i} \ell_{\pi(i)}.$$

Here $\omega \in \mathbb{F}$ is a primitive d-th root of unity and n is any integer satisfying

$$n < \binom{m + d/2}{d/2}.$$

We now show how this constant-dimensional version of our problem can be solved by solving an appropriate system of polynomial equations.

Solving the small dimensional problem. The algorithm is as follows.

> **Input:** Blackbox access to a m-variate polynomial f of degree d and an integer $n \geq 1$.
> **Output:** If f is a projection of $\mathsf{Pow}_{n,d}$ then a set of n linear forms $\ell_1, \ldots, \ell_n$ such that
>
> $$f = \mathsf{Pow}_{n,d}(\ell_1, \ell_2, \ldots, \ell_n)$$
>
> **Algorithm:**
>
> Step (i) By solving an appropriate set of polynomial equations find the set L of all m-variate linear forms ℓ such that $\dim(\partial^{d/2}(f - \ell^d)) \leq (n-1)$.
>
> Step (ii) Let $\ell_1, \ell_2, \ldots, \ell_t$ be all the distinct (upto scalar multiples) members of L. If $t = n$ then output L else output 'Fail.'

Correctness of the algorithm and the running time. The correctness of this algorithm (with high probability over the random choice of the ℓ_i's) is captured in the following proposition whose proof is given in section 7.6.

Proposition 48. *With probability at least*

$$\left(1 - \frac{2dn}{|S|}\right)$$

over the random choice of the ℓ_i's, any linear form p with the property that

$$\dim(\partial^{k+1}(f - p^d)) \leq (n-1) \tag{8}$$

is of the form $\omega \cdot \ell_i$ for some $i \in [n]$. Here $\omega \in \mathbb{F}$ is a d-th root of unity and $k \in [d]$ is any integer such that

$$n < \min\left(\binom{m+d-k-2}{d-k-1}, \binom{m+k}{k+1}\right)$$

For the running time, note that we are solving a system of polynomial equations of degree at most dn in m variables. This can be done in randomized time $(dn)^m$ which is $(dn)^{O(1)}$ for our choice of parameters.

□

6.4 The case of the Sum of Products polynomial

In this section we look at instances of PolyProj where the input polynomial $f(\mathbf{x})$ is an affine projection of $\mathsf{SPS}_{n,d}$, i.e.

$$f(\mathbf{x}) = \sum_{i\in[n]} \prod_{j\in[d]} \ell_{ij}$$

where the ℓ_{ij}'s are m-variate affine forms. Our task is to recover the ℓ_{ij}'s given blackbox access to f.

We follow the algorithm outline given in section 5.2 to design the algorithm of theorem 6.

Proof of theorem 8

We follow the algorithm outline given in section 5.2 and choose $t = n^2 + n + 1$. To prove theorem 8 above we need to prove the uniqueness of projections of $\mathsf{SPS}_{n,d}$ and show how to accomplish the second step of the overall algorithm of section 5.2 in polynomial time. Thus our problem effectively is the same as the problem that we started out with but the number of variables m has reduced to $n^2 + n + 1$. Also note that if

$$f = \sum_{i\in[n]} \prod_{j\in[d]} \ell_{ij}$$

then we can homogenize this expression and assume without loss of generality that f is homogeneous of degree d and the ℓ_{ij}'s are linear forms (rather than affine forms). The uniqueness is captured in the following proposition whose proof is given in section 7.7.

Proposition 49. *Let n, d, m be inetegrs with $d, m > n^2 + n$. If every subset of $n^2 + n$ of the ℓ_{ij}'s is linearly independent then the expression*

$$f = \sum_{i\in[n]} \prod_{j\in[d]} \ell_{ij}$$

is unique in the sense that if we also have

$$f = \sum_{i\in[n]} \prod_{j\in[d]} p_{ij}$$

then there exists a permutation $\pi : ([n] \times [d]) \mapsto (n \times [d])$ such that:

(i) p_{ij} is a scalar multiple of $\ell_{\pi(i,j)}$

(ii) $\pi(i_1, j_1)$ and $\pi(i_2, j_2)$ agree on their first coordinates if and only if $i_1 = i_2$.

We now show how this constant-dimensional version of our problem can be solved by solving an appropriate system of polynomial equations.

Solving the small dimensional problem.

Terminology. We will be looking at subspaces of $\mathbb{F}^m$. We will say that a subspace H of codimension t is *defined by* some t linear forms $p_1, \ldots, p_t$ if the p_i's are $\mathbb{F}$-linearly independent and H is the set of common zeroes of $p_1, p_2, \ldots, p_t$. i.e. if

$$H = \{\mathbf{a} \in \mathbb{F}^m \ : \ p_1(\mathbf{a}) = \ldots = p_t(\mathbf{a}) = 0\}.$$

For a polynomial f we will say that f vanishes on H, denoted

$$f \equiv 0 \pmod{H} \quad \text{iff } f(\mathbf{a}) = 0 \ \ \forall \mathbf{a} \in H.$$

A subspace of codimension 1 will be called a hyperplane (note that a hyperplane corresponds to a linear form by which it is defined). We will say that a set of linear forms is t-wise independent if every subset of size t (and smaller) is linearly idependent. We are now ready to formally state the algorithm.

Input: Integers n, m and d (with $d, m > n^2 + n$) and blackbox access to a homogeneous m-variate polynomial f of degree d.
Output: If f is a projection of $\mathsf{SPS}_{n,d}$ then a set of nd linear forms over m variables $\{\ell_{ij} : i \in [n], j \in [d]\}$ such that

$$f = \sum_{i \in [n]} \prod_{j \in [d]} \ell_{ij}$$

Algorithm:

Step (i) By solving an appropriate set of polynomial equations find the set S of all subspaces $H \subset \mathbb{F}^m$ of codimension n such that

$$f \equiv 0 \pmod{H}.$$

If $|S| \neq d^n$ then output 'Fail.'

Step (ii) Compute the set L of all linear forms ℓ such that there exists a pair of subspaces $H_1, H_2 \in S$ satisfying:

(a) $\mathrm{codim}(\mathrm{Span}(H_1, H_2)) = 1$
(b) $\mathrm{Span}(H_1, H_2)$ is defined by the linear form ℓ.

If $|L| \neq (dn)$ then output 'Fail.'

Step (iii) Form a graph G whose vertices correspond to the nd linear forms in L and where the nodes corresponding to two linear forms $\ell, p \in L$ are adjacent if and only if there does not exist any subspace H in S properly contained in the subspace defined by $p(\mathbf{x}) = \ell(\mathbf{x}) = 0$. Find the connected components of G. If the number of connected components of G is different from n or if the number of nodes in any connected of G is different from d then output 'Fail.'

Step (iv) For each $i \in [n]$ let $T_i(\mathbf{x})$ be the product of the linear forms corresponding to the nodes in the i-th connected component of G. Using the algorithm of lemma 18 find scalars $\alpha_1, \ldots, \alpha_n$ such that

$$f = \alpha_1 \cdot T_1 + \ldots + \alpha_n \cdot T_n.$$

Output the linear forms in each T_i (appropriately scaled).

Correctness of the algorithm and the running time. For the running time, note that in the first step we are solving a system of polynomial equations of degree at most d in n^3 variables. This can be done in time $(d)^{n^3}$. The rest of the steps take only $\mathsf{poly}(dn)$ time.

The correctness of this algorithm is captured in the following proposition whose proof is given in section 7.6.

Proposition 50. *If the ℓ_{ij}'s are (n^2+n)-wise independent then the computations done in the above algorithm satisfy the following properties:*

1. *$|S| = d^n$. Moreover for every subspace H in S there exist $j_1, j_2, \ldots, j_n \in [d]$ such that H is defined by $\ell_{1j_1}, \ldots, \ell_{nj_n}$.*

2. The set L computed in step (ii) consists of scalar multiples of the ℓ_{ij}'s.

3. In the graph G each node corresponds to a unique ℓ_{ij}. Moreover the nodes corresponding to $\ell_{i_1j_1}$ and $\ell_{i_2j_2}$ are adjacent if and only if i_1 equals i_2.

□

7 Proofs of technical claims

7.1 Proofs of technical claims from section 2

In this section we prove theorem 12 from section 2. It has already been noted that if the graph is 3-colorable then f is an affine rojection of g. Our aim is to prove the converse. Henceforth, we will assume that f is an affine projection of g via a map that sends x_i to $\ell_i(x_1, x_2, x_3) + a_i$, where ℓ_i is a linear form. In other words

$$f(x_1, x_2, x_3) = g(\ell_1 + a_1, \ell_2 + a_2, \ldots, \ell_n + a_n) \tag{9}$$

$$\begin{aligned}
\text{where } g &:= \left(\sum_{i\in[n]} x_i^{n^2+4n+4}\right) + \left(\sum_{k\in[n]}\sum_{i\in[n]} x_i^{k(n+3)}\right) + \left(\sum_{\{i,j\}\in E} x_i x_j\right) \\
\text{and } f &:= \left(\sum_{i\in[3]} n_i x_i^{n^2+4n+4}\right) + \left(\sum_{k\in[n]}\sum_{i\in[3]} n_i x_i^{k(n+3)}\right) + \left(\sum_{1\le i<j\le 3} m_{ij} x_i x_j\right)
\end{aligned}$$

We prove the correctness of the reduction (theorem 12) through a sequence of propositions. Our first proposition is an easy consequence of the nonzeroness of the Vandermonde determinant.

Proposition 51. *Let $d \geq 0$ be an integer. If*

$$\sum_{i=1}^{n} \beta_i \alpha_i^k = 0 \quad \text{for } k \in [d..d + (n-1)] \tag{10}$$

then

$$\sum_{i=1}^{n} \beta_i \alpha_i^k = 0$$

for **all** *$k \geq 1$. Moreover, if the α_i's are all nonzero then $\sum \beta_i$ is zero as well.*

Proof. The proof goes via induction on n and uses the properties of the Vandermonde matrix. Equation (10) implies that the vector $(\beta_1, \beta_2, \ldots, \beta_n)$ is in the nullspace of a Vandermonde matrix M whose determinant is

$$\left(\prod_{i=1}^{n} \alpha_i^d \cdot \prod_{i<j} (\alpha_i - \alpha_j)\right).$$

If $(\beta_1, \beta_2, \ldots, \beta_n)$ is the zero vector then

$$\sum_{i=1}^{n} \beta_i \alpha_i^k = 0 \quad \forall k \geq 0.$$

Otherwise either some $\alpha_i = 0$ or some $\alpha_i = \alpha_j$. In both these cases, the conclusion follows by induction. $\square$

Corollary 52. *Let $d > n$ be an integer. Let $\beta_1, \ldots, \beta_n$ be elements of $\mathbb{F}$ each of which is nonzero. If*

$$\sum_{i=1}^{n} \alpha_i^{d-k} \beta_i^k = 0 \quad \forall k \in [n] \tag{11}$$

then

$$\sum_{i=1}^{n} \alpha_i^{d-k} \beta_i^k = 0 \quad \forall k \in [0..d-1]$$

Proof. Rewriting equation (11) as

$$\sum_{i=1}^{n} \gamma_i^{d-k} \beta_i^d = 0 \quad \forall k \in [n]$$

where $\gamma_i := \frac{\alpha_i}{\beta_i}$ and applying proposition 51 above, we get that

$$\sum_{i=1}^{n} \gamma_i^{d-k} \beta_i^d = 0 \quad \forall k \geq 0.$$

The conclusion follows. $\square$

Corollary 53. *Let $\ell_1, \ell_2, \ldots, \ell_n$ be linear forms. Let $a_1, a_2, \ldots, a_n \in \mathbb{F}$ be field elements each of which is nonzero. Let $d > n$ be an integer. If*

$$\sum_{i \in [n]} \ell_i^{d-k} a_i^k = 0 \quad \forall k \in [n]$$

then

$$\sum_{i \in [n]} \ell_i^{d-k} a_i^k = 0 \quad \forall k \in [0..d-1].$$

In particular,

$$\sum_{i \in [n]} \ell_i^d = 0.$$

The proof of this corollary follows if we think of the ℓ_i's and the a_i's as elements of the appropriate rational function field and apply corollary 52. We will now need the following proposition dating back to the time of Newton relating the power symmetric polynomials to the elementary symmetric polynomials.

Proposition 54.

$$\mathsf{Sym}_{n,k} := \frac{1}{k}\left(\mathsf{Sym}_{n,k-1}\mathsf{Pow}_{n,1} - \mathsf{Sym}_{n,k-2}\mathsf{Pow}_{n,2} + \ldots + (-1)^{k-1}\mathsf{Pow}_{n,k}\right)$$

See for example [Mea] for a proof. It yields the following insight into the common solution of a particular system of equations involving the power symmetric polynomials.

Lemma 55. *Let $\alpha_1, \alpha_2, \ldots, \alpha_n$ be field elements. Suppose that for some integer $m \in [0..n]$ we have*

$$\sum_{i\in[n]} \alpha_i^k = m \quad \forall k \in [n]$$

then there exists a subset $S \subseteq [n]$ of size m such that

$$\alpha_i = \begin{cases} 1 & \text{if} \quad i \in S \\ 0 & \text{otherwise} \end{cases}$$

In particular, if

$$\sum_{i\in[n]} \alpha_i^k = 0 \quad \forall k \in [n]$$

then $\alpha_i = 0 \;\; \forall i \in [n]$.

Proof. We first derive a nice expression for $\mathsf{Sym}_{n,k}(\alpha_1, \alpha_2, \ldots, \alpha_n)$.

Claim 56.

$$\mathsf{Sym}_{n,k}(\alpha_1, \alpha_2, \ldots, \alpha_n) = \binom{m}{k}.$$

Proof of Claim 56: The proof is by induction on k. For the base case of $k = 1$ we have

$$\begin{aligned} \mathsf{Sym}_{n,1}(\alpha_1, \ldots, \alpha_n) &= \sum_{i\in[n]} \alpha_i \\ &= m \\ &= \binom{m}{1} \end{aligned}$$

Let us now look at the general case. By Proposition 54 we get

$$\begin{aligned} \mathsf{Sym}_{n,k+1}(\alpha_1, \ldots, \alpha_n) &= \frac{1}{k+1}\left(\mathsf{Sym}_{n,k}\mathsf{Pow}_{n,1} - \mathsf{Sym}_{n,k-1}\mathsf{Pow}_{n,2} + \ldots + (-1)^k\mathsf{Pow}_{n,k+1}\right) \\ &= \frac{1}{k+1}\left(m\binom{m}{k} - m\binom{m}{k-1} + \ldots + (-1)^k m\right) \\ &= \frac{m}{k+1}\left((-1)^k \sum_{j=0}^{k} (-1)^j \binom{m}{j}\right) \\ &= \frac{m}{k+1}\binom{m-1}{k} \\ &= \binom{m}{k+1} \end{aligned}$$

This proves the claim. $\square$

Now consider the univariate polynomial

$$\begin{aligned}(t+\alpha_1)\cdot(t+\alpha_2)\cdot\ldots\cdot(t+\alpha_n) &= \sum_{j\in[0..n]} \mathsf{Sym}_{n,j}(\alpha_1,\ldots,\alpha_n)t^{n-j} \\ &= \sum_{j\in[0..n]} \binom{m}{j} t^{n-j} \\ &= (t+1)^m\cdot t^{n-m}\end{aligned}$$

The statement of the lemma then follows by using unique factorization of (univariate) polynomials. $\square$

We are now ready to give thr proof of theorem 12.

Proof of theorem 12 : Our first aim is to show that the a_i's are all zero. Towards this end, our first step is to show that for each $i\in[n]$, either a_i is zero or ℓ_i is zero. Let $S\subseteq[n]$ consist of those indices $i\in[n]$ such that a_i is zero.

Claim 57. *For each $i\in S$, $\ell_i=0$.*

Proof of claim 57 : For $k\in[(n^2+3n+1)..(n^2+4n)]$, comparing the homogenous parts of degree k on the l.h.s and r.h.s of equation (9) we get that

$$\binom{n^2+4n+4}{k}\left(\sum_{i\in S}\ell_i^k a_i^{n^2+4n+4-k}\right)=0.$$

Since for each $i\in S$, a_i is nonzero, we can apply corollary 53 and obtain

$$\sum_{i\in S}\ell_i^k a_i^{n^2+4n+4-k}=0\quad\forall k\in[1..(n^2+4n+4)]. \tag{12}$$

In particular,

$$\sum_{i\in S}\ell_i^{n^2+4n+4}=0 \tag{13}$$

Now for $k\in[(n^2+2n)..(n^2+3n-1)]$ comparing the coefficient of homogeneous parts of degree k on l.h.s and r.h.s of equation (9) we get that

$$\binom{n^2+4n+4}{k}\left(\sum_{i\in S}\ell_i^k a_i^{n^2+4n+4-k}\right)+\binom{n^2+3n}{k}\left(\sum_{i\in S}\ell_i^k a_i^{n^2+3n-k}\right) \;=\; 0$$

which using (12) in turn means that

$$\sum_{i\in S}\ell_i^k a_i^{n^2+3n-k}=0\quad\forall k\in[(n^2+2n)..(n^2+3n-1)]$$

Applying corollary 53 again, we get

$$\sum_{i\in S}\ell_i^k a_i^{n^2+3n-k}=0\quad\forall k\in[1..(n^2+3n)] \tag{14}$$

In particular, we get

$$\sum_{i \in S} \ell_i^{n^2+3n} = 0 \tag{15}$$

Continuing in this way we get that $k \in [n]$

$$\sum_{i \in S} \ell_i^{k(n+3)} = 0 \tag{16}$$

By lemma 55 we get that $\ell_i = 0$ for each $i \in S$. This proves the claim. $\square$

In the rest of the proof, we will be comparing coefficients of monomials of degree at least one on the two sides of equation (9). This claim above means that we can pretty much forget all the affine forms for which a_i is nonzero because the corresponding ℓ_i is zero and hence such affine forms contribute only to the constant term of r.h.s of equation (9) and not to any higher degree term. Let $\bar{S}$ be the complement of S, i.e. $\bar{S} = [n] \setminus S$. Now let $\ell_i = \alpha_i x_1 + \beta_i x_2 + \gamma_i x_3$. Comparing the coefficient of $x_1^{k(n+3)}$ on the two sides of equation (9) we get

$$\sum_{i \in \bar{S}} \alpha_i^{k(n+3)} = n_1 \quad \forall k \in [n].$$

By lemma 55 there must exist a subset T_1 of $\bar{S}$ of size n_1 such that

$$\alpha_i^{n+3} = \begin{cases} 1 & \text{if} \quad i \in T_1 \\ 0 & \text{if} \quad i \in \bar{S} \setminus T_1 \end{cases} \tag{17}$$

Comparing the coefficient of $x_1^{n^2+4n+4}$ on the two sides of equation (9) we get

$$\sum_{i \in \bar{S}} \alpha_i^{(n+1)(n+3)+1} = n_1$$

which means that

$$\sum_{i \in \bar{S}} \alpha_i = n_1 \quad (\text{as } \alpha_i^{n+3} = \alpha_i^{(n+3)(n+1)} = 1).$$

Combined with (17) we get that in fact

$$\alpha_i = \begin{cases} 1 & \text{if} \quad i \in T_1 \\ 0 & \text{if} \quad i \in \bar{S} \setminus T_1 \end{cases} \tag{18}$$

In a similar we get that there exists a subsets T_2 and T_3 of $\bar{S}$ of sizes n_2 and n_3 respectively such that

$$\beta_i = \begin{cases} 1 & \text{if} \quad i \in T_2 \\ 0 & \text{if} \quad i \in \bar{S} \setminus T_2 \end{cases} \tag{19}$$

and

$$\gamma_i = \begin{cases} 1 & \text{if} \quad i \in T_3 \\ 0 & \text{if} \quad i \in \bar{S} \setminus T_3 \end{cases} \tag{20}$$

Let us compare the coefficient of $x_1^{n^2+4n+3}x_2$ on the two sides of equation (9). We get

$$\begin{aligned} 0 &= \sum_{i\in\bar{S}} \alpha_i^{n^2+4n+3}\beta_i \\ &= \sum_{i\in T_1\cap T_2} \alpha_i^{n^2+4n+3}\beta_i \\ &= \sum_{i\in T_1\cap T_2} \alpha_i\beta_i \\ &= |T_1\cap T_2| \end{aligned}$$

Thus the sets T_1 and T_2 are disjoint. Applying the same argument for other pairs we get that T_1, T_2 and T_3 are pairwise disjoint subsets of $\bar{S} \subseteq [n]$. The union of T_1, T_2, T_3 has size $n_1 + n_2 + n_3 = n$ so that $\bar{S} = [n]$ and S is the empty set. This also means that each ℓ_i equals either x_1 or x_2 or x_3 (depending on which T_j i belongs to). Let $\ell_i = x_{c_i}$ for $c_i \in [3]$. Finally comparing the coefficients of the quadratic terms on the two sides of equation (9) we get that the map

$$\phi : [n] \mapsto [3], \quad i \mapsto c_i$$

is a $(n_1, n_2, n_3, m_{12}, m_{13}, m_{23}) - 3$-coloring of the graph G. This completes the proof of the NP-hardness of PolyProj . □

7.2 Proofs of technical claims from section 4

Proof of Proposition 22. Without loss of generality we can assume $i = 1$. Let $\mathbf{a} = (a_1, \ldots, a_n) \in \mathbb{F}^n$ be the point at which we want the value of $\frac{\partial f}{\partial x_1}$. Consider

$$\hat{f}(x_1) := f(x_1 + a_1, a_2, \ldots, a_n).$$

Then $\hat{f}(x_1)$ can be computed via interpolation. Finally

$$\frac{\partial f}{\partial x_1}(\mathbf{a}) = \frac{\partial \hat{f}}{\partial x_1}(0, 0, \ldots, 0)$$

□

Proof of Proposition 23. Let $\mathbf{a} = (a_1, a_2, \ldots, a_n)$ and for $\lambda \in \mathbb{F}$ let

$$\lambda \cdot \mathbf{a} = (\lambda \cdot a_1, \lambda \cdot a_2, \ldots, \lambda \cdot a_n).$$

Then we have

$$f(\lambda \cdot \mathbf{a}) = \lambda^d \cdot f^{[d]}(\mathbf{a}) + \lambda^{d-1} \cdot f^{[d-1]}(\mathbf{a}) + \ldots + \lambda^0 \cdot f^{[0]}(\mathbf{a})$$

so that by plugging in $(d+1)$ different values for λ in the above equation, using the oracle for $f(\mathbf{x})$ to obtain each $f(\lambda \cdot \mathbf{a})$ and solving the resulting system of linear equations we obtain $f^{[i]}(\mathbf{a})$ in polynomial time. (The matrix corresponding to this system of linear equations is a Vandermonde matrix so that it always has an inverse.) □

Proposition 58. *If* $f(\mathbf{x}) = g(A \cdot \mathbf{x})$ *then*

$$\mathfrak{g}_f = A^{-1} \cdot \mathfrak{g}_g \cdot A$$

Proof. Suppose $B \in \mathfrak{g}_g$, i.e.

$$f(\mathbf{x}) = f((1 + \epsilon \cdot B) \cdot \mathbf{x}).$$

Then

$$g(A \cdot \mathbf{x}) = g(A \cdot (1 + \epsilon \cdot B) \cdot \mathbf{x})$$

so that

$$\begin{aligned} g(\mathbf{x}) &= g(A \cdot (1 + \epsilon \cdot B) \cdot A^{-1} \cdot \mathbf{x}) \\ &= g((1 + \epsilon \cdot (A \cdot B \cdot A^{-1})) \cdot \mathbf{x}) \end{aligned}$$

Thus $\mathfrak{g}_f \subset A^{-1} \cdot \mathfrak{g}_g \cdot A$. Similarly $\mathfrak{g}_g \subset A \cdot \mathfrak{g}_g \cdot A^{-1}$. Thus

$$\mathfrak{g}_f = A^{-1} \cdot \mathfrak{g}_g \cdot A$$

□

We now give the proof of lemma 26 showing that the lie algebra of a polynomial given as a blackbox can be computed efficiently.

Proof of lemma 26. We will obtain the generators of $\mathfrak{g}_f$ by solving a system of homogeneous linear equations. Recall that a matrix $A \in \mathfrak{g}_f$ if and only if

$$f((\mathbf{1}_n + \epsilon A)\mathbf{x}) = f(\mathbf{x}). \tag{21}$$

Let the (i, j)-th entry of A be a_{ij}. A simple computation yields

Claim 59.

$$f((\mathbf{1} + \epsilon A)\mathbf{x}) - f(\mathbf{x}) = \epsilon \cdot \left(\sum_{i,j \in [n]} a_{ij} x_j \frac{\partial f}{\partial x_i} \right) \tag{22}$$

Proof of claim 59. By linearity of derivatives, it suffices to verify (22) for the case when f is a monomial, in which case this is routine. □

Thus the computation of a basis of $\mathfrak{g}_f$ boils down to computing a basis for the $\mathbb{F}$-linear dependencies among the set of polynomials

$$\{x_j \frac{\partial f}{\partial x_i} \ : \ i, j \in [n]\}.$$

By proposition 22, given blackbox access to f, we can obtain blackbox access to its derivatives and therefore also to $x_j \frac{\partial f}{\partial x_i}$ in random polynomial time. We can subsequently compute the $\mathbb{F}$-linear dependencies among these polynomials by the algorithm of lemma 18. □

7.3 Proofs of technical claims from section 5.1

The following is the multivariate analog of Taylor expansion.

Fact 60. *Let $g(x_1, \ldots, x_n) \in \mathbb{F}[x_1, \ldots, x_n]$ be a polynomial. Then*

$$g(x_1 + a_1, \ldots, x_n + a_n) = g(\mathbf{x}) + \frac{1}{1!} \sum_{i \in [n]} a_i \frac{\partial g}{\partial x_i} + \frac{1}{2!} \sum_{i,j \in [n]} a_i a_j \frac{\partial^2 g}{\partial x_i \cdot \partial x_j} + \ldots,$$

where the '$\ldots$' consists of terms with higher order derivatives of g.

Proposition 61. *If $g(\mathbf{x})$ is a regular homogeneous n-variate polynomial of degree d and if*

$$g(A \cdot \mathbf{x} + \mathbf{b}) = g(\mathbf{x})$$

then $\mathbf{b} = 0$ and $A \in \mathscr{G}_g$. In other words, if g is regular and homogeneous then its symmetries under the affine group is the same as its symmetries under the general linear group.

Proof. Comparing the homogeneous parts of degree d on the two sides of

$$g(A \cdot \mathbf{x} + \mathbf{b}) = g(\mathbf{x}) \tag{23}$$

we see that $g(A \cdot \mathbf{x}) = g(\mathbf{x})$ so that $A \in \mathscr{G}_g$. Applying the transformation $A^{-1} \in \mathscr{G}_g$ to the variables in equation (23) we get that

$$g(\mathbf{x} + A^{-1} \cdot \mathbf{b}) = g(A^{-1} \cdot \mathbf{x})$$

so that

$$g(\mathbf{x}) = g(\mathbf{x} + \mathbf{c}),$$

where $\mathbf{c} = (c_1, c_2, \ldots, c_n) = A^{-1} \cdot \mathbf{b}$. Applying Taylor expansion (fact 60) and comparing the homogeneous parts of degree $(d-1)$ on the two sides we get

$$\sum_{i \in [n]} c_i \frac{\partial g}{\partial x_i} = 0.$$

By regularity of g, the first order partial derivatives are $\mathbb{F}$-linearly independent and therefore $c_i = 0$ for each $i \in [n]$. Thus $\mathbf{b} = A \cdot \mathbf{c}$ is also zero. $\square$

Proof of Theorem 30. The interesting direction is the reduction of FullRankProj to PolyEquiv . So let us assume that we have an oracle that given an n-vraiate polynomial h determines an invertible matrix A such that $h(\mathbf{x}) = g(A \cdot \mathbf{x})$, if such an A exists. Now we are given an m-variate polynomial f and suppose there exists A, $\mathbf{b}$ such that

$$f(\mathbf{x}) = g(A\mathbf{x} + b). \tag{24}$$

If m is larger than n then f contains redundant variables and these can be eliminated by the algorithm of lemma 21. So we can assume $m = n$. Using the algorithm of proposition 23, we verify that f has degree d and obtain blackbox access to $f^{[d]}$, the degree d homogeneous component of f. Since g is homogeneous of degree d, comparing the homogeneous parts of degree d on the two sides of equation 24 we have

$$f^{[d]}(\mathbf{x}) = g(A \cdot \mathbf{x}).$$

Using the oracle for g-equivalence, we find a matrix C such that

$$f^{[d]}(\mathbf{x}) = g(C \cdot \mathbf{x}).$$

Then $A \cdot C^{-1} \in \mathscr{G}_g$ and

$$f(C^{-1} \cdot \mathbf{x}) = g(\mathbf{x} + C \cdot A^{-1} \cdot \mathbf{b}).$$

So if we denote by $h(\mathbf{x})$ the polynomial $f(C^{-1} \cdot \mathbf{x})$, then our problem boils down to expressing h as a translation of g. So suppose $h(\mathbf{x}) = g(\mathbf{x} + \mathbf{c})$. By Taylor expansion (Fact 60) we have

$$g(\mathbf{x} + \mathbf{c}) = g(\mathbf{x}) + \sum_{i=1}^{n} c_i \frac{\partial g}{\partial x_i} + \text{lower degree terms}$$

so that

$$h^{[d-1]}(\mathbf{x}) = \sum_{j=1}^{n} c_j \frac{\partial g}{\partial x_j}(\mathbf{x}).$$

If we now plug in $\mathbf{x} = \mathbf{a}_i$ for each $i \in [n]$ then we obtain a system of linear equations with the c_j's as unknowns which we can solve in polynomial time to obtain the c_j's. $\square$

Proof of Proposition 35. Let $L = \prod_{i \in [n]} \lambda_i$. Then we have

$$\mathsf{Perm}(P_{ij}) = L^k$$

and that

$$\left(\frac{\partial \mathsf{Perm}}{\partial x_{k\ell}}\right)(P_{ij}) = \begin{cases} L^k \cdot \lambda_i^{-k} & \text{if } \ell - k = j - 1 \\ 0 & \text{otherwise} \end{cases}$$

Recall that the matrix M is defined as

$$M_{(i,j),(k,\ell)} = \frac{\partial \mathsf{Perm}}{\partial x_{k\ell}}(P_{ij})$$

Thus M is a block diagonal matrix with n blocks $B_1, B_2, \ldots, B_n$ where the t-th block ($t \in [n]$) B_t has n rows with indices of the form (i, t) ($i \in [n]$) and n columns with indices of the form $(k, k + t - 1)$ ($k \in [n]$). To show that M is invertible it suffices to show that each block B_t is invertible. Now the entry of B_t at the i-th row and k-th column is

$$L^k \lambda_i^{-k}$$

so that

$$\mathsf{Det}(B_t) = L^{\frac{(n-1)(n+2)}{2}} \cdot \prod_{i<k} (\lambda_i^{-1} - \lambda_k^{-1})$$

Thus each B_t is invertible. $\square$

7.4 Proofs of technical claims from section 6.1

Proof of proposition 40. By assumption f is $GL(n^2, \mathbb{F})$-equivalent to Perm so let

$$f(\mathbf{x}) = \mathsf{Perm}_n(A \cdot \mathbf{x}) \quad \text{for some } A \in GL(n^2, \mathbb{F}).$$

By proposition 58, we have

$$\mathfrak{g}_f = A^{-1}\mathfrak{g}_{\mathsf{Perm}} \cdot A.$$

So if $D^{-1} \cdot \mathfrak{g}_f \cdot D$ consists of diagonal matrices only then

$$D^{-1} \cdot A^{-1} \cdot \mathfrak{g}_{\mathsf{Perm}} \cdot A \cdot D$$

also consists only of diagonal matrices. We first show that $\mathfrak{g}_{\mathsf{Perm}}$ contains a matrix B all of whose eigenvalues are distinct.

Claim 62. *For $n \geq 3$, $\mathfrak{g}_{\mathsf{Perm}}$ contains matrices all of whose eigenvalue are distinct.*

Proof of claim 62. It suffices to show that a random linear combination of the basis elements of $\mathfrak{g}_{\mathsf{Perm}}$ gives a diagonal matrix with all diagonal entries distinct. Proposition 39 gives an explicit basis of $\mathfrak{g}_{\mathsf{Perm}}$. Let us take a take a formal linear combination of these basis elements, i.e. let us consider the matrix

$$T := \alpha_2 R_2 + \alpha_3 R_3 + \ldots + \alpha_n R_n + \beta_2 C_2 + \beta_3 C_3 + \ldots + \beta_n C_n.$$

Since the R_i's aand the C_j's are diagonal matrices, therefore T is also a diagonal matrix. Moreover

$$T_{ij,ij} = \begin{cases} \sum_{k \in [2..n]} (\alpha_k + \beta_k) & \text{if } i = j = 1 \\ (\sum_{k \in [2..n]} \alpha_k) - \beta_j & \text{if } i = 1, \text{ and } j \geq 2 \\ (\sum_{k \in [2..n]} \beta_k) - \alpha_i & \text{if } i \geq 2, \text{ and } j = 1 \\ -\alpha_i - \beta_j & \text{otherwise} \end{cases}$$

The entries on the diagonal are all distinct when viewed as formal polynomials in the α_i's and the β_j's. By the DeMillo-Lipton-Schwarz-Zippel lemma, the diagonal entries of T will be distinct with high probability if the α_i's and the β_j's are chosen independently at random from a large enough subset of $\mathbb{F}$. This proves the claim. □

Fix such a matrix $B \in \mathfrak{g}_{\mathsf{Perm}}$ all of whose eigenvalues are distinct. Since $(D^{-1} \cdot A^{-1})$ diagonalizes $\mathfrak{g}_{\mathsf{Perm}}$ we have that $C := D^{-1} \cdot A^{-1}$ diagonalizes B. We now claim that $C \in \mathrm{PS}(n^2, \mathbb{F})$.

Claim 63.

$$C \in \mathrm{PS}(n^2, \mathbb{F}).$$

Proof of claim 63. Since the matrix CBC^{-1} is a diagonal matrix, the columns of C must be the eigenvectors of B. Since B itself is diagonal and has distinct eigenvalues, the eigenvectors of B are precisely the elementary unit vectors $e_1, e_2, \ldots, e_{n^2}$ and scalar multiples thereof. In turn this means that the columns of C are scalar multiples of some permutation of $e_1, e_2, \ldots, e_{n^2}$. This in turn means that $C \in \mathrm{PS}(n^2, \mathbb{F})$. □

Since PS is a subgroup of $GL(n^2, \mathbb{F})$ C^{-1} also belongs to PS. Finally we have

$$\begin{aligned} f(D \cdot \mathbf{x}) &= \mathsf{Perm}(A \cdot D \cdot \mathbf{x}) \\ &= \mathsf{Perm}(C^{-1} \cdot \mathbf{x}) \end{aligned}$$

and hence $f(D \cdot \mathbf{x})$ is PS$(n^2, \mathbb{F})$-equivalent to Perm_n, as required.

$\square$

Proof of proposition 41. Recall that by assumption the given f is of the form

$$f(\mathbf{x}) = \mathsf{Perm}_n(\lambda_{11}x_{\pi(1,1)}, \lambda_{12}x_{\pi(1,2)}, \ldots, \lambda_{nn}x_{\pi(n,n)})$$

We will use the observation in equation 5 to compute one such π as follows. Since the permanent is invariant under permuting the rows and columns, we can assume without loss of generality that $\pi(1,1) = (1,1)$. We find a set of size $(2n-2)$ say $A \subset ([n] \times [n] \setminus \{(1,1)\})$ such that

$$\frac{\partial^2 f}{\partial x_{11} \cdot \partial x_{ij}} = 0$$

for each $(i,j) \in A$ (if no such set A exists then f is not PS-equivalent to Perm). We then partition A into two sets R and C of size $(n-1)$ each such that

$$\frac{\partial^2 f}{\partial x_{ij} \cdot \partial x_{k\ell}} \begin{cases} = 0 & \text{if } (i,j) \in R \text{ and } (k,\ell) \in R \\ = 0 & \text{if } (i,j) \in C \text{ and } (k,\ell) \in C \\ \neq 0 & \text{if } (i,j) \in R \text{ and } (k,\ell) \in C \end{cases} \tag{25}$$

(if no such partition is found then f is not PS-equivalent to the permanent). Clearly, such a partition of A can be found efficiently using the property above. Let

$$R = \{(i_1, j_1), (i_2, j_2), \ldots, (i_{n-1}, j_{n-1})\}.$$

Define

$$\pi^{-1}(i_1, j_1) = (2,1),\ \pi^{-1}(i_2, j_2) = (3,1),\ \ldots \pi^{-1}(i_{n-1}, j_{n-1}) = (n,1).$$

Similarly let

$$C = \{(k_1, \ell_1), (k_2, \ell_2), \ldots, (k_{n-1}, \ell_{n-1})\}.$$

Define

$$\pi^{-1}(k_1, \ell_1) = (1,2),\ \pi^{-1}(k_2, \ell_2) = (1,3),\ \ldots \pi^{-1}(k_{n-1}, \ell_{n-1}) = (1,n).$$

Finally for $(i,j) \in ([n] \times [n]) \setminus (\{(1,1)\} \cup R \cup C)$ there must exist a unique pair $(i_r, j_r) \in R$ and $(k_s, \ell_s) \in C$ such that

$$\frac{\partial^2 f}{\partial x_{ij} \cdot \partial x_{i_r j_r}} = \frac{\partial^2 f}{\partial x_{ij} \cdot \partial x_{k_s \ell_s}} = 0$$

(if not then f is not PS-equaivalent to permanent). Define $\pi^{-1}(i,j) = (r,s)$. In this way we have obtained the permutation π. Let $\sigma : ([n] \times [n]) \mapsto ([n] \times [n])$ be the inverse of π. Then the polynomial

$$f_2(\mathbf{x}) := f(x_{\sigma(1,1)}, x_{\sigma(1,2)}, \ldots, x_{\sigma(n,n)})$$

is SC-equivalent to the permanent.

$\square$

The following proposition shows how to do the appropriate scaling and thereby recover the equivalence between f_2 above and the permanent.

Proof of proposition 42. We first note that the stabilizer/automorphism group of the Permanent polynomial itself has a nontrivial intersection with $\mathrm{SC}(n^2, \mathbb{F})$ (theorem 38). This allows us to deduce that we can assume without loss of generality that some $(2n-2)$ λ_{ij}'s are 1. More specifcally, we can assume without loss of generality that

$$\lambda_{11} = \lambda_{12} = \ldots = \lambda_{1n} = \lambda_{21} = \lambda_{31} = \ldots = \lambda_{(n-1)1} = 1.$$

We will compute the rest of the λ_{ij}'s. Now by substituting some variables to zero and some others to one in equation (7) we get

$$f\begin{pmatrix} x_{11} & x_{12} & 0 & \ldots & 0 \\ x_{21} & x_{22} & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \end{pmatrix} = \mathsf{Perm}\begin{pmatrix} x_{11} & x_{12} & 0 & \ldots & 0 \\ x_{21} & \lambda_{22}x_{22} & 0 & \ldots & 0 \\ 0 & 0 & \lambda_{33} & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & \lambda_{nn} \end{pmatrix}.$$

In particular,

$$f(\mathbf{a}) = \lambda_{22}\lambda_{33} \cdot \ldots \cdot \lambda_{nn}, \quad \text{where } \mathbf{a} := \begin{pmatrix} 1 & 0 & 0 & \ldots & 0 \\ 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \end{pmatrix}$$

while

$$f(\mathbf{b}) = \lambda_{33} \cdot \ldots \cdot \lambda_{nn} \quad \text{where } \mathbf{b} := \begin{pmatrix} 0 & 1 & 0 & \ldots & 0 \\ 1 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \end{pmatrix}$$

so that

$$\lambda_{22} = \frac{f(\mathbf{a})}{f(\mathbf{b})}.$$

In a similar way the other λ_{ij}'s can be obtained for $i \in [n-1]$ and $j \in [n]$. Now λ_{nn} can also be obtained as

$$\lambda_{nn} = \frac{f(\mathbf{b})}{\prod_{i \in [3..(n-1)]} \lambda_{ii}}.$$

Finally all the other λ_{nj}'s can be obtained similarly. In this way we have determined whether a given polynomial f is SC-equivalent to the permanent polynomial by evaulating f at $O(n^2)$ points with the overall arithmetic operations also being $O(n^2)$.

□

7.5 Proofs of technical claims from section 6.2

Proposition 64. *With high probability, a random element $B \in \mathfrak{g}_{\mathsf{Det}}$ has the property that all its eigenvalues are distinct.*

Proof. It suffices to show that there exists an $X \in \mathfrak{g}_{\mathsf{Det}}$ all of whose eigenvalues are distinct. The conclusion would then follow by an application of the DeMillo-Lipton-Schwarz-Zippel lemma. Note that the elements of $\mathfrak{g}_{\mathsf{Perm}}$ are also elements of $\mathfrak{g}_{\mathsf{Det}}$ so that such an element exists by claim 62. $\square$

Proposition 65. *With high probability over the random choice of the element $B \in \mathfrak{g}_{\mathsf{Det}}$ there exists an $S \in \mathscr{G}_{\mathsf{Det}}$ such that*

$$S^{-1} \cdot \mathit{Cent}(B) \cdot S$$

consists of diagonal matrices.

Proof. $\mathfrak{g}_{\mathsf{Det}}$ is isomorphic to $\mathfrak{sl}_n \oplus \mathfrak{sl}_n$ (by corollary 44) so that we can deduce structural properties of $\mathfrak{g}_{\mathsf{Det}}$ by proving that $\mathfrak{sl}_n \oplus \mathfrak{sl}_n$ has these properties. Now a random traceless matrix has distinct eigenvalues (with high probability) so that a random element of $\mathfrak{sl}_n$ is is a "locally regular element" of $\mathfrak{sl}_n$ (cf. Graaf [dG97] for the definition of a locally regular element). Thus with high probability a random element $B \in \mathfrak{g}_{\mathsf{Det}}$ is also a locally regular element of $\mathfrak{g}_{\mathsf{Det}}$. This means that $\mathrm{Cent}(B)$ is a Cartan subalgebra of $\mathfrak{g}_{\mathsf{Det}}$ (by Proposition 3.13 in [dG97]). All Cartan subalgebras are conjugate (via automorphisms of the lie algebra). In our case, the lie algebra is isomorphic to the direct product $\mathfrak{sl}_n \oplus \mathfrak{sl}_n$. The Cartan subalgebras of $\mathfrak{sl}_n$ are well understood and it is known (cf. [Kir08]) that the Cartan subalgebras of $\mathfrak{sl}_n$ are in fact all conjugate under $\mathrm{SL}(n, \mathbb{F})$ (fact 29). Now $\mathfrak{g}_{\mathsf{Det}}$ has a 'canonical' Cartan subalgebra consisting of diagonal matrices (corresponding to scaling of rows and columns). Thus there exists an $S \in \mathscr{G}_{\mathsf{Det}}$ such that

$$S^{-1} \cdot \mathrm{Cent}(B) \cdot S$$

consists of diagonal matrices. $\square$

Proof of Proposition 45. Suppose that

$$f(\mathbf{x}) = \mathsf{Det}_n(A \cdot \mathbf{x}) \quad \text{for some } A \in \mathrm{GL}(n^2, \mathbb{F}).$$

By proposition 58 we have

$$\mathfrak{g}_f = A^{-1} \mathfrak{g}_{\mathsf{Det}} A.$$

Thus picking a random $B \in \mathfrak{g}_f$ is the same as picking a random $C \in \mathfrak{g}_{\mathsf{Det}}$ and then computing $B := A^{-1} \cdot C \cdot A$. Since D diagonalizes $\mathrm{Cent}(B) \subset \mathfrak{g}_f$, we have that

$$D^{-1} \cdot A^{-1} \mathrm{Cent}(C) \cdot A \cdot D$$

is a set of diagonal matrices. Now by Proposition 65 there exists an $S \in \mathscr{G}_(\mathsf{Det})$ such that

$$S^{-1} \cdot \mathrm{Cent}(C) \cdot S$$

is a set of diagonal matrices. Proceeding as in the case of the permanent and using claim 63 we get that

$$S^{-1} \cdot (AD) = Z$$

for some matrix $Z \in \mathrm{PS}(n^2, \mathbb{F})$. Therefore $AD = S \cdot Z$. No we have

$$\begin{aligned} f(D \cdot \mathbf{x}) &= \mathsf{Det}(A \cdot D \cdot \mathbf{x}) \\ &= \mathsf{Det}(S \cdot Z \cdot \mathbf{x}) \\ &= \mathsf{Det}(S \cdot (Z \cdot \mathbf{x})) \\ &= \mathsf{Det}(Z \cdot \mathbf{x}) \quad (\text{since } S \in \mathscr{G}_{\mathsf{Det}}) \end{aligned}$$

Thus $f(D \cdot \mathbf{x})$ is PS-equivalent to Det_n. □

7.6 Proofs of technical claims from section 6.3

Terminology: the ring of differential operators. We denote by ∂_i the map from $\mathbb{F}[\mathbf{x}]$ to itself given by $f(\mathbf{x}) \mapsto \frac{\partial f}{\partial x_i}$. Notice that each ∂_i is an $\mathbb{F}$-linear map from $\mathbb{F}[\mathbf{x}]$ to itself. We will denote the linear combinations and compositions of these basic linear operators in the natural way. Thus $\partial_i \partial_j$ is a shorthand for the map that sends $f(\mathbf{x})$ to $(\partial_i(\partial_j f))(\mathbf{x})$, while $\partial_i + \partial_j$ is a shorthand for the map that sends $f(\mathbf{x})$ to $(\partial_i f + \partial_j f)(\mathbf{x})$. Continuing in this way, one can look at all polynomial expressions in $\partial_1, \ldots, \partial_n$. They form a commutative ring which we denote by $\mathbb{F}[\partial_1, \ldots, \partial_n]$. We call it the ring of differential operators.

7.6.1 Representing a univariate polynomial as a sum of like powers of affine forms.

In this subsection we will give a proof of proposition 46. Consider a univariate polynomial $f(x) \in \mathbb{F}[x]$ of degree d as in the statement of proposition 46. Consider the smallest n such that f can be written as the sum of n d-th powers of affine forms, i.e.

$$f = (a_1 x + b_1)^d + (a_2 x + b_2)^d + \ldots + (a_n x + b_n)^d.$$

Let $g(x_1, x_2) = x_2^d f(\frac{x_1}{x_2})$ be the homogenization of f so that

$$g = (a_1 x_1 + b_1 x_2)^d + (a_2 x_1 + b_2 x_2)^d + \ldots + (a_n x_1 + b_n x_2)^d.$$

Johannes Kleppe [Kle99] related this to the vanishing of n-th order derivatives of g in the following manner. Note that we can assume without loss of generality that $(a_i x + b_i)$'s are pairwise coprime. Consider the differential operator

$$D = (b_1 \partial_1 - a_1 \partial_2) \cdot (b_2 \partial_1 - a_2 \partial_2) \cdot \ldots \cdot (b_n \partial_1 - a_n \partial_2)$$

Note that D is square-free and that $D \circ g = 0$. It turns out that the converse is also true.

Lemma 66. *([Kle99], theorem 1.2) Let*

$$D = \prod_{i \in [n]} (\partial_1 - \alpha_i \partial_2)$$

be a square-free *differential operator (i.e. the α_i's are all distinct) of order n such that $D(\partial_1, \partial_2) \circ g(x_1, x_2) = 0$. Then $g(x_1, x_2)$ can be written as an $\mathbb{F}$-linear combination of $(\alpha_1 x_1 + x_2)^d, (\alpha_2 x_1 + x_2)^d, \ldots, (\alpha_n x_1 + x_2)^d$. That is, there exist constants $\beta_1, \beta_2, \ldots, \beta_n$ such that*

$$g(x_1, x_2) = \sum_{i \in [n]} \beta_i (\alpha_i x_1 + x_2)^d.$$

Lemma 67. *([Kle99], lemma 1.1) Let $D_1, D_2, \ldots, D_t \in \mathbb{F}[\partial_1, \partial_2]$ be homogeneous differential operaors of order n. Then the linear space of differential operators generated by the D_i's contains a squarefree operator if and only if a random linear combination of the D_i's gives a square-free operator.*

Proof. An operator $D(\partial_1, \partial_2) \in \mathbb{F}[\partial_1, \partial_2]$ is squarefree if and only if a certain polynomial expression in the coefficients of the $(\partial_1^i \cdot \partial_2^j)$'s is nonzero. The conclusion follows by an application of the DeMillo-Lipton-Schwarz-Zippel lemma. $\square$

Proof of Proposition 46. The uniqueness part of the theorem statement follows from a simple application of the invertibility of a Vandermonde matrix. The algorithm itself follows from lemmas 66 and 67 in the following way. Compute a basis of all differential operators of degree n which make g vanish and take a random linear combination of these operators to determine whether there exists a squarefree operator D in this linear space. Factoring such a squarefree operator D gives us $\alpha_1, \alpha_2, \ldots, \alpha_n$ such that

$$g(x_1, x_2) = \sum_{i \in [n]} \beta_i (\alpha_i x_1 + x_2)^d$$

for some $\beta_1, \ldots, \beta_n \in \mathbb{F}$. Finally the β_i's can be computed by solving an appropriate system of linear equations. The running time is clearly $\mathsf{poly}(n \cdot d)$. $\square$

7.6.2 Sum of like powers of random linear forms.

In the rest of this subsection we consider the following scenario. Let $S \subseteq \mathbb{F}$ be a "very large" finite set. Let m, d, n be positive integers. We pick a collection of n linear forms

$$\ell_1, \ell_2, \ldots, \ell_n, \quad \text{where } \ell_i = \sum_{j \in [m]} a_{ij} x_j$$

with a_{ij} being chosen independently and uniformly at random from $S \subseteq \mathbb{F}$. Here we will analyze linear combinations of d-th powers of such forms. In particular, we will be interested in properties of the polynomial

$$f = \sum_{i \in [n]} \ell_i^d.$$

We begin by recalling a lemma from Ellison [Ell69].

Lemma 68. *Let m, d be positive integers. For any $n \leq \binom{m+d-1}{d}$, there exists a collection of n linear forms $p_1, p_2, \ldots, p_n$ such that the polynomials $p_1^d, p_2^d, \ldots, p_n^d$ are $\mathbb{F}$-linearly independent.*

We now recall one well-known proposition regarding projections of a set of pairwise coprime linear forms (cf. [Kal89] for a proof).

Proposition 69. *Let*

$$\ell_1(\mathbf{x}), \ell_2(\mathbf{x}), \ldots, \ell_n(\mathbf{x}) \in \mathbb{F}[x_1, \ldots, x_m]$$

be a collection of linear forms which is pairwise coprime, i.e. no ℓ_i is a scalar multiple of an ℓ_j ($j \neq i$). Let $A \in \mathbb{F}^{(m \times m)*}$ be a random invertible linear transformation. For $i \in [n]$, let $\hat{\ell}_i := \ell_i(A \cdot \mathbf{x})$. Then with high probability (over the random choice of A), the set of bivariate linear forms $\{\hat{\ell}_i(x_1, x_2, 0, \ldots, 0)\}$ is pairwise coprime.

Proposition 70. *If* $n < \binom{m+d-1}{d}$ *then the collection of polynomials*

$$\{\ell_i^d : i \in [n]\}$$

is $\mathbb{F}$*-linearly independent with probability at least*

$$\left(1 - \frac{dn}{|S|}\right).$$

Proof. Let the coefficient of x_j in ℓ_i be a_{ij}. Consider the $n \times \binom{m+d-1}{d}$ matrix M whose (i,j)-th entry is the coefficient of the j-th monomial in ℓ_i^d. This is a polynomial of degree d in the a_{ij}'s. Then the ℓ_i^d's are $\mathbb{F}$-linearly independent if and only if the rank of this matrix is n. By lemma 68, there exists a set of a_{ij}'s such that M has rank n. The conclusion follows by an application of the DeMillo-Lipton-Schwarz-Zippel lemma. $\square$

Corollary 71. *For* any *set of* n *nonzero field elements* $\alpha_1, \alpha_2, \ldots, \alpha_n$ *the collection of polynomials*

$$\{\alpha_i \ell_i^d : i \in [n]\}$$

is $\mathbb{F}$*-linearly independent with probability at least*

$$\left(1 - \frac{dn}{|S|}\right).$$

(Note that the α_i*'s can be chosen in an arbitrary way, possibly depending on the choice of the* a_{ij}*'s.)*

Proof. The proof is basically the same as the proof of proposition 70. The matrix M' in this case is the same as matrix M we got in the proof of proposition 70 but where the i-th row has been scaled by a factor of α_i. Thus

$$\text{rank}(M') = \text{rank}(M)$$

which inturn equals n with probability at least

$$\left(1 - \frac{dn}{|S|}\right).$$

$\square$

Proposition 72. Dimension of k-th order partial derivatives. *Let* $f = \sum_{i \in [n]} \ell_i^d$. *Consider the set* $\partial^k(f)$ *of* k*-th order partial derivatives of* f. *If* k *is such that*

$$n < \min\left(\binom{m+d-k-1}{d-k}, \binom{m+k-1}{k}\right)$$

then

$$\dim(\partial^k(f)) = n$$

with probability at least

$$\left(1 - \frac{dn}{|S|}\right).$$

Proof. Let $\mathbf{e} = (e_1, e_2, \ldots, e_m) \in \mathbb{Z}^m_{\geq 0}$ with

$$|\mathbf{e}| := e_1 + e_2 + \ldots + e_m = k.$$

Consider the following k-th order partial derivative of f:

$$(\partial_1^{e_1} \cdot \partial_2^{e_2} \cdot \ldots \cdot \partial_m^{e_m}) \circ f$$

We have

$$(\partial_1^{e_1} \cdot \partial_2^{e_2} \cdot \ldots \cdot \partial_m^{e_m}) \circ f = \sum_{i \in [n]} \mathbf{a}_i^{\mathbf{e}} \cdot \ell_i^{d-k}.$$

By proposition 70 the polynomials ℓ_i^{d-k} are $\mathbb{F}$-linearly independent with probability at least $\left(1 - \frac{(d-k)n}{|S|}\right)$ so that it suffices to show that the set of vectors

$$\{(\mathbf{a}_1^{\mathbf{e}_j}, \ldots, \mathbf{a}_n^{\mathbf{e}_j}) \in \mathbb{F}^n \ : |\mathbf{e}_j| = k\} \subseteq \mathbb{F}^m$$

has dimension n with high probability. By lemma 18, the corresponding $r \times n$ matrix

$$M := \begin{pmatrix} \mathbf{a}_1^{\mathbf{e}_1} & \mathbf{a}_2^{\mathbf{e}_1} & \ldots & \mathbf{a}_n^{\mathbf{e}_1} \\ \mathbf{a}_1^{\mathbf{e}_2} & \mathbf{a}_2^{\mathbf{e}_2} & \ldots & \mathbf{a}_n^{\mathbf{e}_2} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{a}_1^{\mathbf{e}_r} & \mathbf{a}_2^{\mathbf{e}_r} & \ldots & \mathbf{a}_n^{\mathbf{e}_r} \end{pmatrix}$$

has rank n with probability at least $(1 - \frac{kn}{|S|})$ (here $r = \binom{m+k-1}{k}$ is the number of possible monomials of degree k in m variables). Overall therefore the set of k-th order partial derivatives $\partial^k(f)$ has dimension n with probability at least

$$\left(1 - \frac{kn + (d-k)n}{|S|}\right) = \left(1 - \frac{dn}{|S|}\right).$$

$\square$

Corollary 73. *Let $\alpha_1, \alpha_2, \ldots, \alpha_n \in \mathbb{F}$ be any set of n nonzero field elements. Let $f = \sum_{i \in [n]} \alpha_i \ell_i^d$. Consider the set $\partial^k(f)$ of k-th order partial derivatives of f. If k is such that*

$$n < \min\left(\binom{m+d-k-1}{d-k}, \binom{m+k-1}{k}\right)$$

then

$$\dim(\partial^k(f)) = n$$

with probability at least

$$\left(1 - \frac{dn}{|S|}\right).$$

Proof. We follow the proof of Proposition 72 above replacing the matrix M with another matrix M' whose columns are scaled by the α_i's so that

$$\text{rank}(M) = \text{rank}(M').$$

The rest of the proof is identical. $\square$

We are now ready to give the proof of Proposition 48.

Proof of Proposition 48. Let $p(\mathbf{x}) = b_1 \cdot x_1 + b_2 \cdot x_2 + \ldots + b_m \cdot x_m$. Let $g := f - p^d$.

Claim 74. *p is a scalar multiple of some ℓ_i.*

Proof of claim. The proof is by reducio ad absurdum. Suppose if possible that p is not a scalar multiple of any ℓ_i. Then by making a suitable change of variables if neccessary, we can assume without loss of generality that $p(x_1, x_2, 0, \ldots, 0)$ is not a scalar multiple of $\ell_i(x_1, x_2, 0, \ldots, 0)$ for every $i \in [n]$ (using proposition 69). In other words we can assume without loss of generality that

$$(b_2 \cdot a_{i1} - b_1 \cdot a_{i2}) \neq 0 \quad \forall i \in [n].$$

Consider the polynomial

$$\begin{aligned} h &:= b_2(\partial_1 \circ g) - b_1(\partial_2 \circ g) \\ &= b_2 \cdot \frac{\partial g}{\partial x_1} - b_1 \cdot \frac{\partial g}{\partial x_2} \\ &= \left(\sum_{i \in [n]} b_2 a_{i1} \ell_i^{d-1} + b_2 b_1 m^{d-1} \right) - \left(\sum_{i \in [n]} b_1 a_{i2} \ell_i^{d-1} + b_1 b_2 m^{d-1} \right) \\ &= \sum_{i \in [n]} (b_2 a_{i1} - b_1 a_{i2}) \ell_i^{d-1} \end{aligned}$$

By corollary 73,

$$\dim(\partial^k(h)) = n$$

(with high probability). On the other hand, the k-th order derivatives of h are linear combinations of the $(k+1)$-th order derivatives of g so that from the assumption that

$$\dim(\partial^{k+1}(f - p^d)) \leq (n-1)$$

we have

$$\dim(\partial^k(h)) \leq (n-1).$$

This is a contradiction. Therefore p must be scalar multiple of some ℓ_i. □

So suppose that $p = \beta \cdot \ell_1$ (by reindexing the ℓ_i's if necessary). Then

$$g = (1 - \beta^d)\ell_1^d + \sum_{i \in [2..n]} \ell_i^d.$$

By corollary 73 the $(k+1)$-th order derivatives of g will have rank n unless $1 - \beta^d = 0$. From equation (8) it now follows that β is a d-th root of unity. □

We are now ready to prove Proposition 47.

Proof of theorem 47. Let us choose $k = \frac{d}{2} - 1$. We have

$$f - p_1^d = \sum_{i \in [2..n]} p_i^d$$

so that

$$\dim(\partial^k(f - p_1^d)) \leq (n-1).$$

Thus by proposition 48, $p_1 = \omega^{e_1} \ell_{\pi(1)}$ for integer $\pi(i) \in [n]$. In a similar way we get that for each $i \in [n]$, $p_i = \omega^{e_i} \ell_{\pi(i)}$ for integer $\pi(i) \in [n]$. It remains to show that π is a permutation. Suppose not, then there exists $i, j \in [n]$ with $i \neq j$ such that $\pi(i) = \pi(j)$. this implies that p_i is a scalar multiple of p_j. In turn, this means that f can be written as the sum of $(n-1)$ d-th powers of linear forms (going to the algebraic closure of $\mathbb{F}$ if necessary). This would mean that

$$\dim(\partial^k(f)) \leq (n-1),$$

but this contradicts corollary 73. Thus π must be permutation. $\square$

Relationship of Symmetric rank with tensor rank. Let $f(\mathbf{x}, \mathbf{y}, \mathbf{z}) \in \mathbb{F}[\mathbf{x}, \mathbf{y}, \mathbf{z}]$ be a homogeneous set-multilinear polynomial over the 3 sets $\mathbf{x}, \mathbf{y}$ and $\mathbf{z}$ (of m variables each). Thus f is of degree 3 and every monomial in f is of the form $x_i y_j z_k$ for some $i, j, k \in [m]$. The tensor rank of f is the smallest integer r such that

$$f = \sum_{i \in [r]} \ell_{i1}(\mathbf{x}) \cdot \ell_{i2}(\mathbf{y}) \cdot \ell_{i3}(\mathbf{z}), \tag{26}$$

where the ℓ_{ij}'s as usual denote linear forms over the relevant set of variables. The *symmetric rank* of f is the smallest integer n such that f is a linear projection of $\mathsf{Pow}_{n,3}$.

Proposition 75. *Let $f(\mathbf{x}, \mathbf{y}, \mathbf{z}) \in \mathbb{F}[\mathbf{x}, \mathbf{y}, \mathbf{z}]$ be a set-multilinear polynomial with tensor rank r and symmetric rank n. Then:*

(1) $n \leq 4r$.

(2) $r \leq n$.

Proof. The second inequality is immediate from the definition of tensor rank and symmetric rank. The first inequality follows from the identity

$$xyz = \frac{1}{24}\left((x+y+z)^3 - (x+y-z)^3 - (x-y+z)^3 + (x-y-z)^3\right)$$

as it allows us to write each summand in equation (26) as a sum of 4 cubes. $\square$

7.7 Proofs of technical claims from section 6.4

Fact 76. *Let H be a subspace defined by the linear forms $p_1, p_2, \ldots, p_t$. A polynomial $f \equiv 0 \pmod{H}$ if and only if there exist polynomials $f_1, f_2, \ldots, f_t$ such that*

$$f = \sum_{i \in [t]} p_i f_i.$$

In the rest of this subsection we will be looking at the polynomial

$$f = \sum_{i\in[n]} \prod_{j\in[d]} \ell_{ij},$$

where the ℓ_{ij}'s are n^2+n-wise independent. For convenience we denote $\prod_{j\in[d]} \ell_{ij}$ by T_i. We will be looking at the situation where $m \geq (n^2+n+1)$. We first record a simple consequence of the DeMillo-Lipton-Schwarz-Zippel lemma which says that if the ℓ_{ij}'s are chosen randomly then with high probability they are (n^2+n)-wise independent.

Fact 77. *If the coefficient of every ℓ_{ij} is chosen uniformly and independently at random from a set $S \subseteq \mathbb{F}$ then with probability at least*

$$\left(1 - \binom{dn}{n^2+n}\frac{n^2+n}{|S|}\right),$$

the ℓ_{ij}'s are (n^2+n)-wise independent.

We will follow the notation used in the algorithm in section 6.4. Recall that S was the set of subspaces of codimension n with the property that for every $H \in S$ we have $f \equiv 0 \pmod{H}$.

Proposition 78. *For any subspace $H \in S$ there exists a unique set $\{j_1, j_2, \ldots, j_n\} \in [d]^n$ such that*

$$H = \{\mathbf{a} \in \mathbb{F}^m : \ell_{1j_1}(\mathbf{a}) = \ell_{2j_2}(\mathbf{a}) = \ldots = \ell_{nj_n}) = 0\}.$$

Proof. Let the subspace H be defined by the n linear forms $h_1, h_2, \ldots, h_n$.

Claim 79. *For each $i \in [n]$ we have*

$$\prod_{j\in[d]} \ell_{ij} \equiv 0 \quad (\mathit{mod}\ H)$$

Proof of claim 79: The proof is by contradiction. We will obtain the contradiction by showing that there exists a set of $(n+1)$ linearly independent vectors in $\mathrm{Span}(h_1, h_2, \ldots, h_n)$. Assume without loss of generality that $T_1, T_2, \ldots, T_r$ are nonzero modulo H. Consider a tuple $\mathbf{j} = (j_1, j_2, \ldots, j_{r-1}) \in [d]^{r-1}$. Then for every such tuple $\mathbf{j}$ there must exist a $j_r \in [d]$ such that

$$b_1\ell_{1j_1} + b_2\ell_{2j_2} + \ldots + b_r\ell_{rj_r} = 0 \pmod{H}$$

for some $b_1, b_2, \ldots, b_r \in \mathbb{F}$ not all zero. Let $p_1 := \sum_{i\in[r]} b_i \cdot \ell_{ij_i}$. These ℓ_{ij_i}'s are n-wise independent so that p_1 is a nonzero vector in $\mathrm{Span}(h_1, h_2, \ldots, h_n)$. Continuing in this way and choosing $(n+1)$ different tuples in $[d]^{r-1}$ we get a set of $(n+1)$ nonzero vectors $p_1, \ldots p_{n+1}$ in $\mathrm{Span}(h_1, h_2, \ldots, h_n)$. Moreover we can ensure that $p_1, \ldots, p_{n+1}$ are linearly independent in the following manner. Each p_k is a linear combination of some $r \leq n$ ℓ_{ij}'s. We can choose our tuples $\mathbf{j}$ such that the ℓ_{ij}'s which span distinct p_k's are mutually disjoint. We need at most (n^2+n) such ℓ_{ij}'s. By assumption these are linearly independent so that the p_k's are linearly independent. □

From the claim above we have

$$\prod_{j\in[d]} \ell_{1j} \equiv 0 \pmod{h_1, h_2, \ldots, h_n}.$$

Since the h_i's are linear forms the ring $\mathbb{F}[\mathbf{x}]/(h_1, h_2, \ldots, h_n)$ is an integral domain so it must happen that there exists some $j \in [d]$ such that

$$\ell_{1j} \equiv 0 \pmod{h_1, h_2, \ldots, h_n}.$$

By reindexing the ℓ_{1j}'s if necessary we can assume without loss of generality that that $j = 1$. In a similar manner we obtain $\ell_{i1} \equiv 0 \pmod{h_1, h_2, \ldots, h_n}$ for each $i \in [n]$ (by reindexing the ℓ_{ij}'s if necessary). But $\ell_{11}, \ell_{21}, \ldots, \ell_{n1}$ are linearly independent. This means that the hyperplane H can equivalently be defined as the set of common zeroes of $\ell_{11}, \ell_{21}, \ldots, \ell_{n1}$. Finally the uniqueness also follows from the $(n+1)$-wise independence of the ℓ_{ij}'s. □

Corollary 80. *The map I from $[d]^n$ to S given by*

$$I(j_1, j_2, \ldots, j_n) = \text{subspace defined by } \ell_{1j_1}, \ell_{2j_2}, \ldots, \ell_{nj_n}$$

is a bijection.

Proof. By linear independence of $\ell_{1j_1}, \ell_{2j_2}, \ldots, \ell_{nj_n}$, the subspace H defined by these linear forms is of codimension n. Also $T_i = \prod_{j \in [d]} \ell_{ij}$ vanishes modulo ℓ_{ij_i} so that

$$f = \sum_{i \in [n]} T_i \equiv 0 \pmod{H}.$$

Thus $I(j_1, j_2, \ldots, j_n)$ is in S. If $(k_1, \ldots, k_n) \in [d]^n$ is another n-tuple of indices then the subspace defined by $\{\ell_{1j_1}, \ldots, \ell_{nj_n}\}$ is distinct from the subspace defined by $\{\ell_{1k_1}, \ldots, \ell_{nk_n}\}$ because of linear independence of these forms. Thus I is a one-one map. Finally by the proposition above, every subspace in S is in the image of I so that I is a bijection. □

Proposition 81. *If H_1 and H_2 are in S and if* $\mathrm{codim}(\mathrm{Span}(H_1, H_2)) = 1$ *then there exists an ℓ_{ij} such that* $\mathrm{Span}(H_1, H_2)$ *is the same as the hyperplane defined by ℓ_{ij}. Moreover every ℓ_{ij} can be obtained in this manner.*

Proof. By proposition 78 above and by reindexing the ℓ_{ij}'s if necessary we can assume without loss of generality that

$$H_1 \equiv \ell_{11} = \ell_{21} = \ldots = \ell_{n1} = 0.$$

Let

$$H_2 \equiv \ell_{1j_1} = \ell_{2j_2} = \ldots = \ell_{nj_n} = 0.$$

We then have

$$\mathrm{codim}(\mathrm{Span}(H_1, H_2)) = 2n - \mathrm{rank}(\ell_{11}, \ldots, \ell_{n1}, \ell_{1j_1}, \ldots, \ell_{nj_n}).$$

From the $2n$-wise linear independence of the ℓ_{ij}'s we have

$$\mathrm{rank}(\ell_{11}, \ldots, \ell_{n1}, \ell_{1j_1}, \ldots, \ell_{nj_n}) = 2n - |\{i : j_i = 1\}|.$$

Thus we have

$$|\{i : j_i = 1\}| = 1$$

so that $\mathrm{Span}(H_1, H_2)$ is the same as the hyperplane defined by such an ℓ_{i1}. For the converse consider a linear form ℓ_{ij}, say ℓ_{11}. Let H_1 be he hyperplane defined by

$$\ell_{11} = \ell_{21} = \ldots = \ell_{n1} = 0.$$

Let H_2 be the hyperplane defined by

$$\ell_{11} = \ell_{22} = \ell_{32} \ldots = \ell_{n2} = 0.$$

Then we have

$$f \equiv 0 (\mathrm{mod}\ H_1) \quad \text{and } f \equiv 0 (\mathrm{mod}\ H_2).$$

Moreover from the assumption that these linear forms are linearly independent we get that

$$\mathrm{codim}(\mathrm{Span}(H_1, H_2)) = 1.$$

Thus every ℓ_{ij} is obtained as the linear form defining a hyperplane spanned by some pair $H_1, H_2 \in S$. □

We are now ready to give the proofs of propositions 49 and 50.

Proof of Proposition 50. 1. From corollary 80 it follows that the size of S is d^n and it consists only of subspaces defined by some set $\ell_{1j_1}, \ldots, \ell_{nj_n}$ of linear forms.

2. By Proposition 81, we get that some scalar multiple of each ℓ_{ij} is an element of L and that all elements of L arise in this way. Moreover by pairwise linear independence of the ℓ_{ij}'s, L has exactly dn distinct elements.

3. Consider the two nodes in the graph G corresponding to a $\ell_{i_1j_1}$ and $\ell_{i_2j_2}$. We have the following two cases.

   I: $i_1 \neq i_2$. Without loss of generality we can assume

   $$i_1 = 1, i_2 = 2, j_1 = j_2 = 1.$$

   Then the subspace H in S defined by $\ell_{11}, \ell_{21}, \ldots, \ell_{n1}$ has the property that it is properly contained in the space defined by $\ell_{11} = \ell_{21} = 0$. Thus the nodes corresponding to ℓ_{11} and ℓ_{21} are not adjacent.

   II: $i_1 = i_2$. Without loss of generality we may assume

   $$i_1 = i_2 = 1, j_1 = 1, j_2 = 2.$$

   By corollary 80 every subspace H in S is defined by a set of linear forms $\ell_{1j_1}, \ell_{2j_2}, \ldots, \ell_{nj_n}$. By linear independence of

   $$\ell_{11}, \ell_{12}, \ell_{2j_2}, \ldots, \ell_{nj_n},$$

   we get that the subspace H cannot be contained in the subspace defined by ℓ_{11} and ℓ_{12}.

□

Proof of Proposition 49. Suppose that

$$f = \sum_{i \in [n]} \prod_{j \in [d]} \ell_{ij} = \sum_{i \in [n]} \prod_{j \in [d]} p_{ij}. \tag{27}$$

For $i \in [0..(n-1)]$, define

$$S_i := \{\text{Span}(H_1, H_2) : H_1, H_2 \in S \text{ and codim}(\text{Span}(H_1, H_2)) = (n-i)\}$$

Then S_0 equals S and S_{n-1} is the set of subspaces defined by linear forms in L. Proceeding as above and using the n^2-wise linear independence of the ℓ_{ij}'s we get that S_i has the following three propoerties:

1. Every subspace $J \in S_i$ is defined by some $(n-i)$ linear forms $\ell_{k_1 j_1}, \ell_{k_2 j_2}, \ldots, \ell_{k_{n-i} j_{n-i}}$ where $k_1, k_2, \ldots, k_{n-i}$ are all distinct.

2. There are exactly $\binom{n}{n-i} d^{n-i}$ distinct subspaces in S_i.

3. Every subspace $J \in S_i$ contains exactly $\binom{i}{j} \cdot d^j$ subspaces in S_{i-j}.

Now by counting the number of subspaces in S_{i-j} contained in a given subspacein S_i one sees by induction that the p_{ij}'s must be $2n$-wise (linearly) independent. For example if p_{11} and p_{21} were linearly dependent then f would vanish identically on the codimension $(n-1)$ subspace H defined by $p_{11}, p_{21}, \ldots, p_{n1}$ (this would yield a contradiction as H would then contain infinitely many distinct subspaces of codimension n on which f vanishes). Similarly if p_{11} and p_{12} were linearly dependent then the codimension $(n-1)$ subspace H in S_1 defined by $p_{21}, p_{31}, \ldots, p_{n1}$ woud contain at most $(d-1)$ distinct subspaces in S_0. Once the $2n$-wise independence of the p_{ij}'s is established then one sees that every subspace in S_{n-1} is defined by a unique p_{ij}. Every subspace in S_{n-1} is also defined by some ℓ_{ij} from which we deduce that there exists a permutation

$$\pi : ([n] \times [d]) \mapsto ([n] \times [d])$$

p_{ij} is a scalar multiple of $\ell_{\pi(i,j)}$. Finally for any pair of linear forms $\ell_{i_1 j_1}, \ell_{i_2 j_2}$, the subspace of codimension 2 defined by them is in S_{n-2} if and only if i_1 is distinct from i_2. From this we deduce that the permutation has the second property as well, i.e. $\pi(i_1, j_1)$ and $\pi(i_2, j_2)$ agree on their first coordinate if and only if $i_1 = i_2$.

□

8 Summary

Our works shows that while PolyProj subsumes many problems of intense interest in arithmetic complexity, it is an intractable problem in general. There are two common themes that emerge as one goes about tackling the special cases of practical interest. The first is that for some of the most widely encountered families of polynomials, the affine equivalence of a given polynomial to a member of some such family can be determined efficiently. The second common theme is that many lower bound proofs have been through the discovery of affinely invariant properties which can in turn lead to efficient algorithms for the relevant cases of PolyProj . Going forward, it seems reasonable to expect affine projections to be an important area of investigation with techniques and ideas from one special case getting mirrored in the other special cases.

References

[Aar08] Scott Aaronson. Arithmetic natural proofs theory is sought. available at `http://scottaaronson.com/blog/?p=336`, 2008.

[AFT11] Boris Alexeev, Michael Forbes, and Jacob Tsimerman. Tensor rank: Some lower and upper bounds. In *CCC*, 2011.

[Agr06] Manindra Agrawal. Determinant versus permanent. In *Proceedings of the International Congress of Mathematicians (ICM)*, pages 1409–1421, 2006.

[AS06] M. Agrawal and N. Saxena. Equivalence of F-algebras and cubic forms. In *Proceedings of the 23rd Annual Symposium on Theoretical Aspects of Computer Science*, pages 115–126, 2006.

[AvM10] Scott Aaronson and Dieter van Melkebeek. A note on circuit lower bounds from derandomization. *Electronic Colloquium on Computational Complexity (ECCC)*, 17:105, 2010.

[BGI09] A Bernardi, A Gimigliano, and M Id. Computing symmetric rank for symmetric tensors. *Journal of Symbolic Computation*, 46:34–53, 2009.

[BI11] Peter Burgisser and Christian Ikenmeyer. Geometric complexity theory and tensor rank. In *STOC*, 2011.

[Bot67] Peter Botta. Linear transformations that preserve the permanent. *Proceedings of the American Mathematical Society*, 18:566–569, 1967.

[Cai90] J.-Y. Cai. A note on the determinant and permanent problem. *Information and Computation*, 84(1):119–127, 1990.

[CCL08] J.-Y. Cai, X. Chen, and D. Li. A quadratic lower bound for the permanent and determinant problem over any characteristic $\neq 2$. In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, pages 491–498, 2008.

[CGLM08] Pierre Comon, Gene Golub, Lek-Heng Lim, and Bernard Mourrain. Symmetric tensors and symmetric tensor rank. *SIAM J. Matrix Anal. Appl.*, 30:1254–1279, September 2008.

[CKW11] Xi Chen, Neeraj Kayal, and Avi Wigderson. Partial derivatives in arithmetic complexity. *Foundations and Trends in Theoretical Computer Science*, 2011.

[dG97] Willem de Graaf. *Algorithms for Finite-Dimensional Lie Algebras.* PhD thesis, Technical University of Eindhoven, 1997.

[Die48] J. Dieudonné. Sur une geńeŕalisation du groupe orthogonal aq́uatre variables. *Archiv Der Math.*, 1:282–287, 1948.

[Ell69] W. J. Ellison. A 'waring's problem' for homogeneous forms. *Proceedings of the Cambridge Philosophical Society*, 65:663–672, 1969.

[ER93] Richard Ehrenborg and Gian-Carlo Rota. Apolarity and canonical forms for homogeneous polynomials. *Eur. J. Comb.*, 14(3):157–181, 1993.

[Fro97] Georg Frobenius. Ueber die darstellung der endlichen gruppen durch linearc substitutionen. *Sitzungber. der Berliner Akademie*, 7:994–1015, 1897.

[Hal07] Brian Hall. *Lie Groups, Lie Algebras, and Representations*. Springer-Verlag, NewYork, 2007.

[Hås90] Johan Håstad. Tensor rank is np-complete. *J. Algorithms*, 11:644–654, December 1990.

[Kal89] E. Kaltofen. Factorization of polynomials given by straight-line programs. *Randomness and Computation*, 5:375–412, 1989.

[Kan97] S. Kantor. Théorie der aquivalenz von linearen $\infty^{\lambda}-$ scharen bilinearer formen. *Sitzungber. der Münchener Akademie*, 2:367–381, 1897.

[Kar72] Richard Karp. Reducibility among combinaorial problems. In *Complexity of Computer Computations*, pages 85–103, 1972.

[Kay11] Neeraj Kayal. Efficient algorithms for some special cases of the polynomial equivalence problem. In *Proceedings of ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1409–1421, 2011.

[KI04] V. Kabanets and R. Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Computational Complexity*, 13(1/2):1–46, 2004.

[Kir08] Alexander Kirillov. *An introduction to Lie Groups and Lie Algebras.* Cambridge University Press, Cambridge, 2008.

[Kle99] Johannes Kleppe. Representing a homogeneous polynomial as a sum of powers of linear forms. Master's thesis, University Of Oslo, 1999.

[KN97] Eyal Kushilevitz and Noam Nisan. *Communication Complexity.* Cambridge University Press, Cambridge, 1997.

[KS09] Zohar Shay Karnin and Amir Shpilka. Reconstruction of generalized depth-3 arithmetic circuits with bounded top fan-in. In *IEEE Conference on Computational Complexity*, pages 274–285, 2009.

[Mea] D. G. Mead. Newton's identities. *The American Mathematical Monthly*, 99(8):749–751.

[Mes89] R. Meshulam. On two extremal matrix problems. *Linear Algebra and its Applications*, 114/115:261–271, 1989.

[MM59] Marvin Marcus and B. N. Moyls. Linear transformations on algebras of matrices. *Canadian Journal of Math.*, 11:61–66, 1959.

[MM62] Marvin Marcus and Francis May. The permanent function. *Canadian Journal of Math.*, 14:177–189, 1962.

[Mor44] K. Morita. Schwarz's lemma in a homogeneous space of higher dimensions. *Japanese Journal of Math*, 19:45–46, 1944.

[MP59] Marvin Marcus and Roger Purves. Linear transformations on algebras of matrices: the invariance of the elementary symmetric functions. *Canadian Journal of Math.*, 11:383–396, 1959.

[MR04] T. Mignon and N. Ressayre. A quadratic bound for the determinant and permanent problem. *International Mathematics Research Notices*, pages 4241–4253, 2004.

[NW97] N. Nisan and A. Wigderson. Lower bounds on arithmetic circuits via partial derivatives. *Computational Complexity*, 6(3):217–234, 1997.

[Pat96] J. Patarin. Hidden field equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms. In *Advances in Cryptology* — EUROCRYPT 1996, pages 33–48, 1996.

[Pól71] G. Pólya. Aufgabe 424. *Arch. Math. Phys.*, 20:1913, 271.

[Raz10] Ran Raz. Tensor-rank and lower bounds for arithmetic formulas. In *STOC*, pages 659–666, 2010.

[RR94] A.A. Razborov and S. Rudich. Natural proofs. *Journal of Computer and System Sciences*, 55:204–213, 1994.

[Sch25] I. Schur. Einige bemerkungen zur determinantentheorie. *der Preussischen Akademie der Wissenschaften zu Berlin*, 25:454–463, 1925.

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

[Shp07] Amir Shpilka. Interpolation of depth-3 arithmetic circuits with two multiplication gates. In *STOC*, pages 284–293, 2007.

[Str69] V. Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13:354–356, 1969.

[SW01] A. Shpilka and A. Wigderson. Depth-3 arithmetic circuits over fields of characteristic zero. *Computational Complexity*, 10(1):1–27, 2001.

[SY10] Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science*, 5:207–388, March 2010.

[Sze92] G. Szegö. Zu aufgabe 424. *Arch. Math. Phys.*, 21:1913, 291–292.

[vzG87] J. von zur Gathen. Permanent and determinant. *Linear Algebra and its Applications*, 96:87–100, 1987.

[Wig02] Avi Wigderson. Arithmetic complexity - a survey. Technical report, Institute for Advanced Studies, 2002. available at `http://www.math.ias.edu/~avi/TALKS/arithmetic_complexity.pdf`.

ECCC ISSN 1433-8092
http://eccc.hpi-web.de