# Reconstruction of Real Depth-3 Circuits with Top Fan-In 2

## Gaurav Sinha

**Department of Mathematics, California Institute of Technology, Pasadena, USA**
`gsinha@caltech.edu`

──── **Abstract** ────

Reconstruction of arithmetic circuits has been heavily studied in the past few years and has connections to proving lower bounds and deterministic identity testing. In this paper we present a polynomial time randomized algorithm for reconstructing $\Sigma\Pi\Sigma(2)$ circuits over $\mathbb{F}$ ($char(\mathbb{F}) = 0$), i.e. depth$-3$ circuits with fan-in 2 at the top addition gate and having coefficients from a field of characteristic 0.

The algorithm needs only a blackbox query access to the polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ of degree $d$, computable by a $\Sigma\Pi\Sigma(2)$ circuit $C$. In addition, we assume that the "*simple rank*" of this polynomial (essential number of variables after removing the gcd of the two multiplication gates) is bigger than a fixed constant. Our algorithm runs in time $poly(n, d)$ and returns an equivalent $\Sigma\Pi\Sigma(2)$ circuit (with high probability).

The problem of reconstructing $\Sigma\Pi\Sigma(2)$ circuits over finite fields was first proposed by Shpilka [24]. The generalization to $\Sigma\Pi\Sigma(k)$ circuits, $k = O(1)$ (over finite fields) was addressed by Karnin and Shpilka in [15]. The techniques in these previous involve iterating over all objects of certain kinds over the ambient field and thus the running time depends on the size of the field $\mathbb{F}$. Their reconstruction algorithm uses lower bounds on the lengths of Linear Locally Decodable Codes with 2 queries. In our settings, such ideas immediately pose a problem and we need new ideas to handle the case of the characteristic 0 field $\mathbb{F}$.

Our main techniques are based on the use of Quantitative Sylvester Gallai Theorems from the work of Barak et al. [3] to find a small collection of "*nice*" subspaces to project onto. The heart of our paper lies in subtle applications of the Quantitative Sylvester Gallai theorems to prove why projections w.r.t. the "*nice*" subspaces can be "glued". We also use Brill's Equations from [8] to construct a small set of candidate linear forms (containing linear forms from both gates). Another important technique which comes very handy is the polynomial time randomized algorithm for factoring multivariate polynomials given by Kaltofen [14].

## 1 Introduction

The last few years have seen significant progress towards interesting problems dealing with arithmetic circuits. Some of these problems include Deterministic Polynomial Identity Testing, Reconstruction of Circuits and recently Lower Bounds for Arithmetic Circuits. There has also been work connecting these three different aspects. In this paper we will primarily be concerned with the reconstruction problem. Even though it's connections to Identity Testing and Lower Bounds are very exciting, the problem in itself has drawn a lot of attention because of elegant techniques and connections to learning. The strongest version of the problem requires that for any $f \in \mathbb{F}[x_1, \ldots, x_n]$ with blackbox access given one wants to

construct (roughly) most succinct representation i.e. the smallest possible arithmetic circuit computing the polynomial. This general problem appears to be very hard. Most of the work done has dealt with some special type of polynomials i.e. the ones which exhibit constant depth circuits with alternating addition and multiplication gates. Our result adds to this by looking at polynomials computed by circuits of this type (alternating addition/multiplication gates but of depth 3). Our circuits will have variables at the leaves, operations $(+, \times)$ at the gates and scalars at the edges. We also assume that the top gate has only two children and the "*simple rank*" of this polynomial (essential number of variables after removing the gcd of the two multiplication gates) is bigger than a constant. The bottom most layer has addition gates and so computes linear forms, the middle layer then multiplies these linear forms together and the top layer adds two such products. Later in Remark 1.1 we discuss that we may assume the linear forms computed at bottom level to be homogeneous and the in-degree of all gates at middle level to be the same (= degree of $f$). Therefore these circuits compute polynomials with the following form:

$$f(x_1, \ldots, x_n) = G(x_1, \ldots, x_n)(T_0(x_1, \ldots, x_n) + T_1(x_1, \ldots, x_n))$$

where $T_i(x_1, \ldots, x_n) = \prod_{j=1}^{M} l_{ij}$ and $G(x_1, \ldots, x_n) = \prod_{j=1}^{d-M} G_j$ with the $l_{ij}$'s and $G_j$'s being linear forms for $i \in \{0, 1\}$. Also assume $gcd(T_0, T_1) = 1$. Our condition about the essential number of variables (after removing gcd from the multiplication gates) is called "*simple rank*" of the polynomial and is defined as dimension of the space

$$sp\{l_{ij} : i \in \{0, 1\}, j \in \{1, \ldots, M\}\}$$

When the underlying field $\mathbb{F}$ is of characteristic 0 ($\mathbb{Q}, \mathbb{R}$ or $\mathbb{C}$ for simplicity), we give an efficient randomized algorithm for reconstructing the circuit representation of such polynomials. Formally our main theorem reads:

▶ **Theorem 1.1** ($\Sigma\Pi\Sigma_{\mathbb{F}}(2)$ Reconstruction Theorem). *Let $f = G(T_0 + T_1) \in \mathbb{F}[x_1, \ldots, x_n]$ be any degree $d$, $n-$ variate polynomial (to which we have blackbox access) which can be computed by a depth 3 circuit with top fan-in 2 (i.e. a $\Sigma\Pi\Sigma(2)$ circuit) i.e. $G, T_i$ being products of affine forms. Assume $gcd(T_0, T_1) = 1$ and $span\{l : l \mid T_0 T_1\}$ is bigger than $s + 1$ (a fixed constant defined below). We give a randomized algorithm which runs in time $poly(n, d)$ and computes the cicuit for $f$ with high probability.*

▶ **Definition 1.2.** We fix $s$ to be any constant $> \max(C_{2k-1} + k, c_{\mathbb{F}}(4))$ where:
1. $c_{\mathbb{F}}(l) = 3l^2$ is the rank lower bound (see Theorem 1.7) that guarantees non-zeroness of any simple, minimal, $\Sigma\Pi\Sigma(l)$ circuit with rank $> c_{\mathbb{F}}(l)$.
2. $k = c_{\mathbb{F}}(3) + 2$.
3. $\delta$ is some fixed number in $(0, \frac{7-\sqrt{37}}{6})$.
4. $C_k = \frac{C^k}{\delta}$ the constant that appears in Theorem B.4.

From our discussion before the theorem about Remark 1.1, we can assume in the above theorem that the polynomial and all linear forms involved are homogeneous.

As per our knowledge this is the first algorithm that efficiently reconstructs such circuits (over the char 0 fields). Over finite fields, the same problem has been considered by [24] and our method takes inspiration from their work. They also generalized this finite field version to circuits with arbitrary (but constant) top fan-in in [15]. However we need many new tools and techniques as their methods don't generalize at a lot of crucial steps. For example:

- They iterate through linear forms in a finite field which we unfortunately cannot do.
- They use lower bounds for Locally Decodable Codes given in [7] which again does not work in our setup.

We resolve these issues by

- Constructing candidate linear forms by solving simultaneous polynomial equations obtained from Brill's Equations (Chapter 4, [8]).
- Using quantitative versions of the Sylvester Gallai Theorems given in [3] and [6]. This new method enables us to construct *nice* subspaces, take projections onto them and glue the projections back to recover the cicuit representation.

## 1.1 Previous Work and Connections

Efficient Reconstruction algorithms are known for some concrete class of circuits. We list some here:

- Depth-2 $\Sigma\Pi$ circuits (sparse polynomials) in [20]
- Read-once arithmetic formulas in [25]
- Non-commutative ABP's [2]
- $\Sigma\Pi\Sigma(2)$ circuits over finite fields in [24], extended to $\Sigma\Pi\Sigma(k)$ circuits (over finite fields) with $k = O(1)$ in [15].
- Random Multilinear Formular in [11]
- Depth 4 ($\Sigma\Pi\Sigma\Pi$) multilinear circuits with top fan-in 2 in [10]
- Random Arithmetic Formulas in [12]

All of the above work introduced new ideas and techniques and have been greatly appreciated.

It's straightforward to observe that a polynomial time deterministic reconstruction algorithm for a circuit class $C$ also implies a polynomial time Deterministic Identity Testing algorithm for the same class. From the works [1] and [13] it has been established that blackbox Identity Testing for certain circuit classes imply superpolynomial circuit lower bounds for an explicit polynomial. Hence the general problem of deterministic reconstruction cannot be easier than proving superpolynomial lower bounds. So one might first try and relax the requirements and demand a randomized algorithm. Another motivation to consider the probabilistic version comes from Learning Theory. A fundamental question called the *exact learning problem using membership queries* asks the following: *Given oracle access to a Boolean function, compute a small description for it.* This problem has attracted a lot of attention in the last few decades. For e.g. in [18][9] and [17] a negative result stating that a class of boolean circuits containing the trapdoor functions or pseudo-random functions has no efficient learning algorithms. Among positive works [23], [4], [19] show that when $f$ has a small circuit (inside some restricted class) exact learning from membership queries is possible. Our problem is a close cousin as we are looking for exact learning algorithms for algebraic functions. Because of this connection with learning theory it makes sense to also allow randomized algorithms for reconstruction.

## 1.2 Depth-3 Arithmetic Circuits

We will use the definitions from [16]. Let $C$ be an arithmetic circuit with coefficients in the field $\mathbb{F}$. We say $C$ is a $\Sigma\Pi\Sigma(k)$ circuit if it computes an expression of the form:

$$C(\bar{x}) = \sum_{i \in [k]} \prod_{j \in [d]} l_{i,j}(\bar{x}).$$

$l_{i,j}(\bar{x})$ are linear forms of the type $l_{i,j}(\bar{x}) = \sum\limits_{s \in [n]} a_s x_s$ where $(a_1, \dots, a_n) \in \mathbb{F}^n$ and $(x_1, \dots, x_n)$ is an $n-$ tuple of indeterminates. For convenience we denote the multiplication gates in $C$ as

$$T_i = \prod_{j \in [d]} l_{i,j}(\bar{x}) \,.$$

$k$ is the top fanin of our circuit $C$ and $d$ is the fanin of each multiplication gate $T_i$. With these definitions we will say that our circuit is of type $\Sigma\Pi\Sigma_{\mathbb{F}}(k, d, n)$. When most parameters are understood we will just call it a $\Sigma\Pi\Sigma(k)$ circuit.

▶ **Remark 1.1.** *Note that we are cosidering homogeneous circuits. There are two basic assumptions:*

**1.** *$l_{i,j}$'s have no constant term i.e. they are linear forms.*

**2.** *Fanin of each $T_i$ is equal to $d$.*

*If these are not satisfied we can homogenize our circuit by considering $Z^d(C(\frac{X_1}{Z}, \dots, \frac{X_n}{Z}))$. Now both the conditions will be taken care of by reconstructing this new homogenized circuit. We need a rank condition on our polynomial which remains essentially unchanged even after this substitution.*

▶ **Definition 1.3** (Minimal Circuit). We say that the circuit $C$ is minimal if no strict non empty subsets of the $\Pi\Sigma$ polynomials $\{T_1, \dots, T_k\}$ sums to zero.

▶ **Definition 1.4** (Simple Circuit and Simplification). A circuit $C$ is called Simple if the gcd of the $\Pi\Sigma$ polynomials $gcd(T_1, \dots, T_k)$ is equal to 1 (i.e. is a unit). The simplification of a $\Sigma\Pi\Sigma(k)$ circuit $C$ denoted as $Sim(C)$ is the $\Sigma\Pi\Sigma(k)$ circuit obtained by dividing each term by the gcd of all terms i.e.

$$Sim(C) \stackrel{def}{=} \sum_{i \in [k]} \frac{T_i}{gcd(T_1, \dots, T_k)} \,.$$

▶ **Definition 1.5** (Rank of a Circuit). Identifying each linear form $l(\bar{x}) = \sum\limits_{s \in [n]} a_s x_s$ with the vector $(a_1, \dots, a_n) \in \mathbb{F}^n$, we define the rank of $C$ to be the dimension of the vector space spanned by the set $\{l_{i,j} | i \in [k], j \in [d]\}$.

▶ **Definition 1.6** (Simple Rank of a Circuit). For a $\Sigma\Pi\Sigma(k)$ circuit $C$ we define the *Simple Rank* of $C$ as the rank of the circuit $Sim(C)$.

Before we go further into the paper and explain our algorithm we state some results about uniqueness of these circuits. In a nutshell for a $\Sigma\Pi\Sigma_{\mathbb{F}}(2, d, n)$ circuit $C$, if one assumes that the *Simple rank* of $C$ is bigger than a constant ($c_{\mathbb{F}}(4)$ : defined later) then the circuit is essentially unique.

## 1.3   Uniqueness of Representation

Shpilka et al. showed the uniqueness of circuit representation in [24] using rank bounds for Polynomial Identity Testing. The bound they used were from the work of Dvir et al. in [7]. It essentialy states that the rank of a simple, minimal $\Sigma\Pi\Sigma(k)$ circuit ($d \geq 2, k \geq 3$) which computes the identically zero polynomial is $\leq 2^{O(k^2)} \log^{k-2} d$. For circuits over char 0 fields improved rank bounds were given by Kayal et al. in [16].

In a series of following work the rank bounds for identically zero $\Sigma\Pi\Sigma(k)$ circuits got further improved. The best known bounds over char 0 fields were given by Saxena et al. in [22]. We rewrite Theorem 1.5 in [22] here for completion.

▶ **Theorem 1.7** (Theorem 1.5 in [22])**.** *Let* $C$ *be a* $\Sigma\Pi\Sigma(k, d, n)$ *circuit over field* $\mathbb{F}$ *that is simple, minimal and zero. Then,* $rk(C) < 3k^2$.

Let $c_{\mathbb{F}}(k) = 3k^2$. This gives us the following version of Corollary 7, Section 2.1 in [24].

▶ **Theorem 1.8** ([24])**.** *Let* $f(\bar{x}) \in \mathbb{F}[x]$ *be a polynomial which exhibits a* $\Sigma\Pi\Sigma(2)$ *circuit*

$$C = G(A + B).$$

$A = \prod_{j \in [M]} A_j, B = \prod_{j \in [M]} B_j, G = \prod_{i \in [d-M]} G_i,$ *where* $A_i, B_j, G_k \in Lin_{\mathbb{F}}[\bar{x}]$. $gcd(A, B) = 1,$ *and* $Sim(C) = A + B$ *has rank* $\geq c_{\mathbb{F}}(4) + 1$ *then the representation is unique. That is if:*

$$f = G(A + B) = \tilde{G}(\tilde{A} + \tilde{B})$$

*where* $A, B, \tilde{A}, \tilde{B}$ *are* $\Pi\Sigma$ *polynomials over* $\mathbb{F}$ *and* $gcd(\tilde{A}, \tilde{B}) = 1$ *then we have* $G = \tilde{G}$ *and* $(A, B) = (\tilde{A}, \tilde{B})$ *or* $(\tilde{B}, \tilde{A})$ *(upto scalar multiplication).*

**Proof.** Let $g = gcd(G, \tilde{G})$ and let $G = gG_1, \tilde{G} = g\tilde{G}_1$. Then $gcd(G_1, \tilde{G}_1) = 1$ and we get

$$G_1 A + G_1 B - \tilde{G}_1 \tilde{A} - \tilde{G}_1 \tilde{B} = 0$$

This is a simple $\Sigma\Pi\Sigma(4)$ circuit with *rank* bigger than $c_{\mathbb{F}}(4) + 1$ and is identically 0 so it must be not minimal. Considering the various cases one can easily prove the required equality. ◄

## 1.4 Notation

$[n]$ denotes the set $\{1, 2, \ldots, n\}$. Throughout the paper we will work over the field $\mathbb{F}$. Let $V$ be a finite dimensional $\mathbb{F}$ vector space and $S \subset V$, $sp(S)$ will denote the linear span of elements of $S$. $dim(S)$ is the dimension of the subspace $sp(S)$. If $S = \{s_1, \ldots, s_k\} \subset V$ is a set of linearly independent vectors then $fl(S)$ denotes the affine subspace generated by points in $S$ (also called a $(k-1) - flat$ or just $flat$ when dimension is understood). In particular:

$$fl(S) = \{\sum_{i=1}^{k} \lambda_i s_i : \lambda_i \in \mathbb{F}, \sum_{i=1}^{k} \lambda_i = 1\}.$$

Let $W \subset V$ be a subspace, then we can extend basis and get another subspace $W'$ (called the complement of $W$) such that $W \oplus W' = V$. Note that the complement need not be unique. Corresponding to each such decomposition of $V$ we may define orthogonal projections $\pi_W, \pi_{W'}$ onto $W, W'$ respectively. Let $v = w + w' \in V, w \in W, w' \in W'$:

$$\pi_W(v) = w, \pi_{W'}(v) = w'.$$

$(\bar{x})$ will be used for the tuple $(x_1, \ldots, x_n)$.

$$Lin_{\mathbb{F}}[\bar{x}] = \{a_1 x_1 + \ldots + a_n x_n : a_i \in \mathbb{F}\} \subset \mathbb{F}[\bar{x}]$$

is the vector space of all linear forms over the variables $(x_1, \ldots, x_n)$. For a linear form $l \in Lin_{\mathbb{F}}[\bar{x}]$ and a polynomial $f \in \mathbb{F}[x]$ we write $l \mid f$ if $l$ divides $f$ and $l \nmid f$ if it does not. We say $l^d \mid\mid f$ if $l^d \mid f$ but $l^{d+1} \nmid f$.

$$\Pi\Sigma_{\mathbb{F}}^d[\bar{x}] = \{l_1(\bar{x}) \ldots l_d(\bar{x}) : l_i \in Lin_{\mathbb{F}}[\bar{x}]\} \subset \mathbb{F}[\bar{x}]$$

is the set of degree $d$ homogeneous polynomials which can be written as product of linear forms. This collection for all possible $d$ is called the set

$$\Pi\Sigma_{\mathbb{F}}[\bar{x}] = \bigcup_{d \in \mathbb{N}} \Pi\Sigma_{\mathbb{F}}^d[\bar{x}]$$

also called $\Pi\Sigma$ polynomials for convenience. Let $f(\bar{x}) \in \mathbb{F}[x]$ then $Lin(f) \in \Pi\Sigma_{\mathbb{F}}[\bar{x}]$ denotes the product of all linear factors of $f(\bar{x})$. Let $\mathcal{L}(f)$ denote the set of all linear factors of $f$. For any set of polynomials $S \subset \mathbb{C}[\bar{x}]$, we denote by $\mathbb{V}(S)$, the set of all complex simultaneous solutions of polynomials in $S$ (this set is called the variety of $S$), i.e.

$$\mathbb{V}(S) = \{a \in \mathbb{C} : \text{ for all } f \in S, f(a) = 0\}.$$

Let $\mathcal{B} = \{b_1, \ldots, b_n\}$ be an ordered basis for $V = Lin_{\mathbb{F}}[\bar{x}]$. We define maps $\phi_{\mathcal{B}} : V \setminus \{0\} \to V$ as

$$\phi_{\mathcal{B}}(a_1 b_1 + \ldots + a_n b_n) = \frac{1}{a_k}(a_1 b_1 + \ldots + a_n b_n)$$

where $k$ is such that $a_i = 0$ for all $i < k$ and $a_k \neq 0$.

A non-zero linear form $l$ is called <u>normal</u> with respect to $\mathcal{B}$ if $l \in \Phi_{\mathcal{B}}(V)$ i.e. the first non-zero coefficient is 1. A polynomial $P \in \Pi\Sigma_{\mathbb{F}}[\bar{x}]$ is normal w.r.t. $\mathcal{B}$ if it is a product of normal linear forms. For two polynomials $P_1, P_2 \in \Pi\Sigma_{\mathbb{F}}[\bar{x}]$ we define:

$$gcd_{\mathcal{B}}(P_1, P_2) = P \in \Pi\Sigma_{\mathbb{F}}[\bar{x}], P \text{ normal w.r.t. } \mathcal{B} \text{ such that } P \mid P_1, P \mid P_2$$

When a basis is not mentioned we assume that the above definitions are with respect to the standard basis.

We can represent any linear form in $Lin_{\mathbb{F}}[\bar{x}]$ as a point in the vector space $\mathbb{F}^n$ and vice versa. To be precise we define the cannonical map $\Gamma : Lin_{\mathbb{F}}[\bar{x}] \to \mathbb{F}^n$ as

$$\Gamma(a_1 x_1 + \ldots + a_n x_n) = (a_1, \ldots, a_n).$$

$\Gamma$ is a linear isomorphism of vector spaces $Lin_{\mathbb{F}}[\bar{x}]$ and $\mathbb{F}^n$. Because of this isomorphism we will interchange between points and linear forms whenever we can. We choose to represent the linear form $a(\bar{x}) = a_1 x_1 + \ldots + a_n x_n$ as the point $a = (a_1, \ldots, a_n)$.

**LI** will be the abbreviation for Linearly Independent and **LD** will be the abbreviation for Linearly Dependent.

▶ **Definition 1.9** (Standard Linear Form). A non zero vector $v$ is called *standard* with respect to basis $\mathcal{B} = \{b_1, \ldots, b_n\}$ if the coefficient of $b_1$ in $v$ is 1. When a basis is not mentioned we assume we're talking about the standard basis. (Equivalently for linear forms the coefficient of $x_1$ is 1). A $\Pi\Sigma$ polynomial will be called *standard* if it is a product of standard linear forms.

We close this section with a lemma telling us when can we replace the span of some vectors with the affine span or flat. We've used this several times in the paper.

▶ **Lemma 1.10.** *Let* $l, l_1, \ldots, l_t \in Lin_{\mathbb{F}}[\bar{x}]$ *be* standard *linear forms w.r.t. some basis* $\mathcal{B} = \{b_1, \ldots, b_n\}$ *such that* $l \in sp(\{l_1, \ldots, l_t\})$ *then*

$$l \in fl(\{l_1, \ldots, l_t\}).$$

**Proof.** Since $l \in sp(\{l_1, \ldots, l_t\})$, we know that $l = \sum\limits_{i \in [t]} \alpha_i l_i$ for some scalars $\alpha_i \in \mathbb{F}$. All linear forms are *standard* w.r.t. $\mathcal{B} \Rightarrow$ comparing the coefficients of $b_1$ we get that $\sum\limits_{i \in [t]} \alpha_i = 1$ and therefore $l \in fl(\{l_1, \ldots, l_t\})$.                                                                  ◀

Let $T \subset \mathbb{F}^n$, By a scaling of $T$ we mean a set where all vectors get scaled (possibly by different scalars).

## 1.5   Summary of Technical Ideas

This Subsection includes the very broad technical ideas we used. First we explain a technique to reconstruct points from their projections. Then we give an overview of the Project-Reconstruct-Lift algorithm and how we plan to execute it. After that we illustrate the algorithm in quite generality. In this illustration we keep a lot of technicalities aside and try to motivate and picturize the algorithm through geometric intiuition.

### 1.5.1   A Simple Reconstruction Technique

We describe a method to recover points from their projections. A more rigorous treatment is in Appendix C. It also contains details and proofs of the Algorithm that is used in this paper. Suppose we have two disjoint sets of points $A = \{a_1, a_2\}, B = \{b_1, \ldots, b_d\}$ in the projective space $\mathbb{P}^{n+1}$ such that:

- We know the set $A$.
- We know the projections of points in $B$ w.r.t. $a_1$ and $a_2$ i.e we know lines joining $L_{i,j} = \overrightarrow{a_i, b_j}$ for $i \in [2]$ and $j \in [d]$.

We want to use lines $L_{i,j} = \overrightarrow{a_i b_j}$ to find the set $\{b_1, \ldots, b_t\}$ in $O(poly(d))$ time. Note that there are $\leq d$ lines through $a_1$ and $\leq d$ lines through $a_2$. The $b_j$'s lie at the intersection of these lines and so we have $\leq d^2$ intersections. These intersections form a set of candidate points for $B$ but it is very hard to cutdown this set to $B$ in $poly(d)$ time. There is a trivial $O(\binom{d^2}{d})$ algorithm - Go through all $d$ points in these intersection points, make the lines and check if you get the same set of lines. This will give all sets of size $d$ which could generate this configuration. Here is how the entire point configuration looks like. The green points $c_j$'s are intersections of our lines which do not belong to $B$.
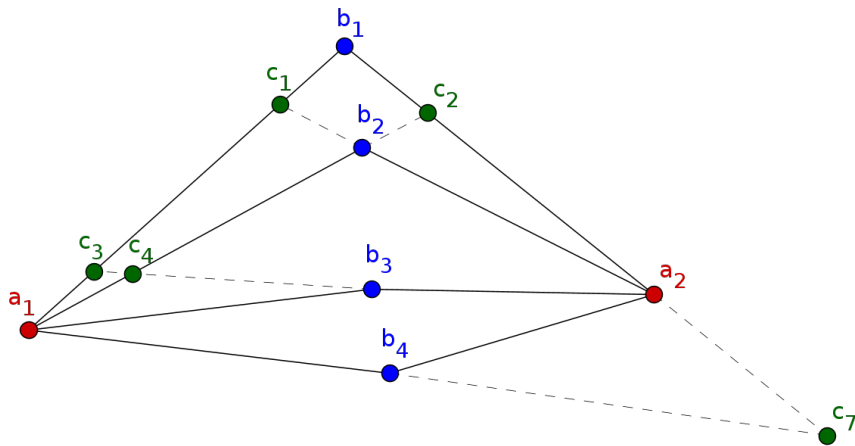
However if one assumes some restrictions then a subset of $B$ might be found in $poly(d)$ time. Assume that for some $t \in [d]$:

- $\{a_1, a_2, b_1\}$ are affinely independent.
- $fl\{a_2, b_1\} \cap B = \{b_1, \ldots, b_t\}$.
- $fl\{a_1, a_2, b_1\} \cap B = \{b_1, \ldots, b_t\}$.

That is we have a sub configuration that looks like:
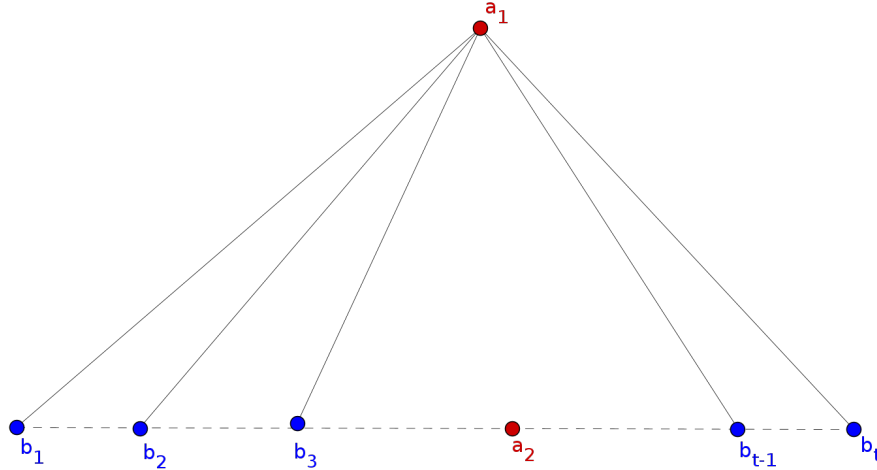


Here is an algorithm to recover all $\{b_1, \ldots, b_t\} \subset B$ such that the above conditions are satisfied.

- We iterate through all lines passing through $a_2$.
- For each such line $L$, find the set of lines $S_L$ through $a_1$ which intersects $L$. Clearly all lines in $S_L$ and $L$ are co-planar.
- If this plane does not contain any other line through $a_2$, output the intersections of lines in $S_L$ with $L$.

It is more or less straightforward that this algorithm works. The line $L$ we choose has to have some $b_j$ on it. Now all lines $\tilde{L} \in S_L$ that intersect $L$ have to intersect it in some $b_i$ otherwise $\tilde{L}$ has some other $b_s$ on it but then the plane of $S_L, L$ will have another line $\overrightarrow{a_2 b_s}$ passing through $a_2$ on it which is a contradiction. The algorithm actually finds all such configurations $\{b_1, \ldots, b_t\} \subset B$.

## 1.5.2   General Overview of the Algorithm

The broad structure of our algorithm is similar to that of Shpilka in [24] however our techniques are different. We first restrict the blackbox inputs to a low ($O(1)$) dimensional random subspace of $\mathbb{F}^n$ and interpolate this restricted polynomial. Next we try to recover the $\Sigma\Pi\Sigma(2)$ structure of this restricted polynomial and finally lift it back to $\mathbb{F}^n$. The random subspace and unique $\Sigma\Pi\Sigma(2)$ structure will ensure that the lifting is unique. Similar to [24] we try to answer the following questions. However our answers (algorithms) are different from theirs:

1. For a $\Sigma\Pi\Sigma(2)$ polynomial $f$ over $r = O(1)$ variables, can one compute a small set of linear forms which contains all factors from both gates?

**2.** Let $V_0$ be a co-dimension $k$ subspace($k = O(1)$) and $V_1, \dots, V_t$ be co-dimension 1 subspaces of a linear space $V$. Given circuits $C_i$ ($i \in \{0, \dots, t\}$) computing $f|_{V_i}$(restriction of $f$ to $V_i$) can we reconstruct from them a single circuit $C$ for $f|_V$?

**3.** Given co-dimension 1 subspaces $V \subset U$ and circuits $f|_V$ when is the $\Sigma\Pi\Sigma(2)$ circuit representations of lifts of $f|_V$ to $f|_U$ unique?

Our first question is easily solved using Brill's equations (See Chapter 4 [8]). These provide a set of polynomials whose simultaneous solutions completely characterize coefficients of complex $\Pi\Sigma$ polynomials. A linear form $l = x_1 - a_2 x_2 - \dots - a_r x_r$ divides one of the gates of $f(x_1, \dots, x_r) \Rightarrow f(a_2 x_2 + \dots + a_r x_r, x_2, \dots, x_r)$ is a $\Pi\Sigma$ polynomial modulo $l$. When this is applied into Brill's equation (see Corollary A.2) we recover possible $l$'s which obviously include linear factors of gates. We can show that (see Claim E.2) the extra linear forms we get are not too many ($poly(d)$) and also have some special structure. We call this set $\mathcal{C}$ of linear forms as Candidate linear forms and non-deterministically guess from this set. It should be noted that we do all this when our polynomial is over $O(1)$ variables.

We deal with the second question while trying to reconstruct the $\Sigma\Pi\Sigma(2)$ representation of the interpolated polynomial $f|_V$, where $V$ is the random low dimensional subspace. We divide the algorithm into Easy Case, Medium Case and a Hard Case.

- For the Easy Case our algorithm tries to reconstruct one of the multiplication gates of $f|_V$ by first looking at it's restriction to a special co-dimension 1 subspace $V_1$. If $f = A + B$ with $A, B$ being $\Pi\Sigma$ polynomials, the projection of one of the gates (say $A$) with respect to $V_1$ will be 0 and the other (say $B$) will remain unchanged giving us $B$ and therefore both gates by factoring $f|_V - B$.

- In the Medium Case we have alteast two extra dimensions in one of the gates. This can be used to show that the only linear factors of $f_{|V}$ are those coming from $G$. Now we can recover $G$ by factoring $f$ and then use Easy Case for the remaining polynomial. An important consequence of this case is that in the Hard Case we may now assume that both gates are high dimensional which is very crucial.

- In the Hard Case we will first need $V_0$, a co-dimension $k$ (where $k = O(1)$) subspace and then iteratively select co-dimension 1 subspaces $V_1, \dots, V_t$. For some gate (say $B$), all pairs $(V_0, V_i)$ ($i \in [t]$) will reconstruct some linear factors of $B$. This process will either completely reconstruct $B$ or we will fall into the Easy Case. Once $B$ is known we can factor $f|_V - B$ to get $A$.

The restrictions that we compute always factor into product of linear forms and can be easily computed since we know $f|_V$ explicitly. They can then be factorized into product of linear forms using the factorization algorithms from [14]. It is the choice of the subspaces $V_0, V_1, \dots, V_t$ where our algorithm differs from that in [24] significantly. Our algorithm selects $V_0$ and iteratively selects the $V_i$'s ($i \in [t]$) such that $(V_0, V_i)$ have certain "*nice*" properties which help us recover the gates in $f|_V$. The existence of subspaces with "*nice*" properties is guaranteed by Quantitative Sylvester Gallai Theorems given in [3]. To use the theorems we had to develop more machinery that has been explained later.

The third question comes up when we want to lift our solution from the random subspace $V$ to the original space. This is done in steps. We first consider random spaces $U$ such that $V$ has co-dimension 1 inside them. Now we reconstruct the circuits for $f|_V$ and $f|_U$. The $\Sigma\Pi\Sigma(2)$ circuits for $f|_V$ and $f|_U$ are unique since the simple ranks are high enough (because $U, V$ are random subspaces of high enough dimension) implying that the circuit for $f|_V$ lifts to a unique circuit for $f|_U$. When this is done for multiple $U$'s we can find the gates exactly.

#### 1.5.2.1  Project-Reconstruct-Lift Algorithm

Here is a broad outline of the three aspects. This technique is quite common. Details of Project and Lift are in Section 4 and that of Reconstruct is in Section 3.

#### 1.5.2.2  Project

- Input:$f \in \mathbb{F}[x_1, \ldots, x_n]$ as blackbox
- Choose random basis $\{y_1, \ldots, y_n\}$ of $\mathbb{F}^n$, $V = sp(\{y_1, \ldots, y_s\}), V_i = sp(\{v_1, \ldots, v_s, v_i\})$ for $i \in \{s+1, \ldots, n\}$.
- Define $f_0(y_1, \ldots, y_s) = f_{|V}, f_i(y_1, \ldots, y_s, y_i) = f_{|V_i}$.
- Consider sets $H \subset V, H_i \subset V_i$ with $|H| \geq d^s, |H_i| \geq d^{s+1}$ and interpolate to find $f_0, f_i$.

#### 1.5.2.3  Reconstruct

- Reconstruct to get $f_0 = M_0 + M_1$ and $f_i = M_0^i + M_1^i$ with $M_0, M_1 \in \Pi\Sigma[y_1, \ldots, y_s], M_0^i, M_1^i \in \Pi\Sigma[y_1, \ldots, y_s, y_i]$.

#### 1.5.2.4  Lift

- Use $M_0, M_1, M_0^i, M_1^i$ to compute gates $N_0, N_1$ such that $f = N_0 + N_1$.
- If the reconstruction was successful return it, else return failed.

## 2   An Illustrative Example

Let $\bar{x}$ denote the variables $(x_1, \ldots, x_r)$ where $r$ is a constant (we will fix this constant later). Consider the following polynomial $f(\bar{x}) \in \mathbb{F}[x_1, \ldots, x_r]$

$$f(\bar{x}) = T_0(\bar{x}) + T_1(\bar{x}).$$

Such that:

1. $T_0(\bar{x}) = A_1 \ldots A_d, T_1(\bar{x}) = B_1 \ldots B_d$ with $A_i, B_j$ linear forms
2. $gcd(A_i, B_j) = 1$ for all $1 \leq i, j \leq d$.
3. $dim(\{A_i, B_j : i, j \in [d]\}) = r$ i.e. there are no redundant variables.

Define the sets $A = \{A_1, \ldots, A_d\}$, $B = \{B_1, \ldots, B_d\}$. We are going to view the points in $A$ and $B$ as points in the space $\mathbb{F}^r$. We also identify (keep only one copy) linear forms which are scalar multiples of each other.

▶ **Theorem 2.1.** *Consider $f(\bar{x})$ from above and assume $f(\bar{x}) = \sum_{\lambda \in \Lambda} \mathbf{c}_\lambda \mathbf{x}^\lambda$ where $\lambda = (\lambda_1, \ldots, \lambda_r)$ and $\mathbf{x}^\lambda = x_1^{\lambda_1} \ldots x_r^{\lambda_r}$. Suppose we know all the coefficients $\mathbf{c}_\lambda$ then in time poly(d) we can reconstruct $T_0(\bar{x}), T_1(\bar{x})$ with high probability.*

We will describe an algorithm which proves the above theorem. At many points during the algorithm we will need results that are mentioned later in the paper. For better understanding we encourage the reader to first go through this algorithm assuming all the claims mentioned.

## 2.1 Candidate Linear Forms

Our job in this algorithm is to reconstruct $T_0(\bar{x}), T_1(\bar{x})$ i.e. $A_i$'s and $B_j$'s. Let us first observe a property these linear forms satisfy. One can see that for $l \in \{A_i, B_j : i, j \in [d]\}$ the following holds:

$f_{|_{l=0}}$ is a non-zero product of linear forms .

Can we use this to reconstruct $A_i, B_j$? The two questions that pop up are:

1. Are there linear forms other than $A_i, B_j$ that satisfy the above condition?
2. If yes, can we find out some structure of the bad $l$'s ( which are not $A_i, B_j$)?
3. Can we bound the total number of such $l$'s by a polynomial in $d$?
4. Can we construct this set efficiently?

The answer to all the above questions is a YES!

▶ **Example 2.2.** Consider $f(x_1, \ldots, x_r) = (x_1 + x_2)(x_1 + x_3) \ldots (x_1 + x_r) + x_2 \ldots x_r$. We can see that $f_{|_{x_1=0}} = x_2 \ldots x_r$ but $x_1$ is not a factor of any of the gates.

The next claim contains the information structure of the bad $l$'s and their number. Proof will be given later in the paper in Appendix E.

▶ **Claim 2.3.** *Consider the set $\mathcal{C} = \{l : f_{|_{l=0}}$ is a non zero product of linear forms $\}$ and let $\{l_1, \ldots, l_k\} \subset T_i$ be a set of LI linear forms where $k = c_{\mathbb{F}}(3) + 2$ (rank bound for $\Sigma\Pi\Sigma(3)$ circuits) then*

1. $\{A_i, B_j : i, j \in [d]\} \subseteq \mathcal{C}$
2. $|\mathcal{C}| \leq O(d^4)$
3. *If $l \in \mathcal{C} \setminus \{A_i, B_j, i, j \in [d]\}$, then there exists $i \in [k]$ and $j \in [d]$ such that $\{l, A_i, B_j\}$ are linearly dependent i.e. for every LI set $\{A_1, \ldots, A_k\}$, a bad $l$ will match one of these $A_i$ $(i \in [k])$ to some $B_j$.*

Moreover the above set $\mathcal{C}$ can be constructed in time $poly(d)$. This is done by solving a set of multivariate polynomial equations of $poly(d)$ degree in $O(1)$ variables. Please see Appendix E for details.

## 2.2 Reconstruction Algorithm

Before going to the core of the algorithm let's explain an easy case. Recall $A = \{A_1, \ldots, A_d\}$ and $B = \{B_1, \ldots, B_d\}$. Also color the points in $A$ red and the points in $B$ blue.

### 2.2.1 Easy Case

For this case we assume

$$sp(A) \subsetneq sp(B) .$$

So let's say $A_1 \notin sp(B)$. The main advantage of such an $A_1$ is that on setting $A_1$ to 0 no linearly independent $\{B_i, B_j\}$ become dependent. Geometrically we have the following picture:
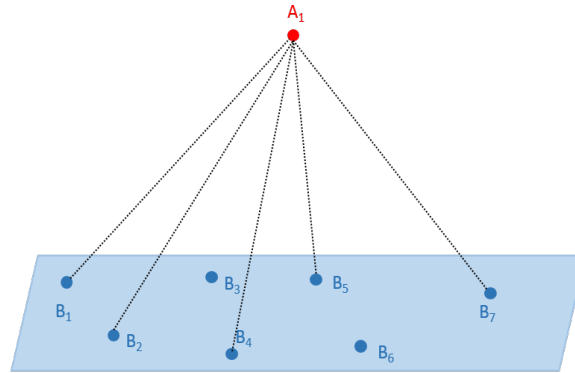
We guess a basis $\{l_1, \ldots, l_r\}$ of linear forms from the set $\mathcal{C}$. While doing this we assume:

- $l_1 = A_1$
- $l_2, \ldots, l_t$ is a basis for $B$
- $l_{t+1}, \ldots, \ldots, l_r$ are the rest of the basis vectors

If our guess was actually a basis we define an invertible linear transformation $T$ sending $l_i$ to $x_i$. We apply $T$ to $f(\bar{x})$ by applying it to each variable in the most natural way. If our guess was correct we get

$$f'(\bar{x}) = f(T(\bar{x})) = x_1 A'_2 \ldots A'_d + B'_1 \ldots B'_d.$$

Note that if our assumption for the basis is correct then none of the $B'_i$'s contain $x_1$. So we can compute $f'_{|_{x_1=0}} = B'_1 \ldots B'_d$. Then we can apply $T^{-1}$ and get back $T_1(\bar{x}) = B_1, \ldots, B_d$. We remind the reader that everything is recovered upto a scalar multiple but that is not a problem since that can be merged into one scalar for the gate $B(\bar{x})$ which can be easily recovered. We then factorize $f - T_1(\bar{x})$ and check if it factors into a product of linear forms and recover $T_0(\bar{x})$. Note that during the process we will guess the basis correctly atleast once. Also the last step checks if we actually get a $\Sigma\Pi\Sigma(2)$ circuit and therefore the reconstruction will be complete. The case where $sp(B) \subsetneq sp(A)$ is symmetrical and is handled in the same way. Next we deal with the hard case.

### 2.2.2 Hard Case

The other case i.e. $sp(A) = sp(B)$ is much harder but high dimensionality enables us to apply the Quantitative version of Sylvester Gallai Theorems from [3]. Let's first just give some consequences of the Quantitative Sylvester Gallai theorem (from [3]) which will be useful for us. A slightly more general version with proof can be found in Appendix B.

▶ **Corollary 2.4.** *Let $S = \{s_1, \ldots, s_n\} \subseteq \mathbb{C}^d$ be a set of points. Assume $dim(S) > \Omega(C^k)$ for some constant $C$, then there exists a set of linearly independent points $\{s_1, \ldots, s_k\}$ and a set $T \subset S$ with $|T| \geq 0.99n$, such that $fl(\{s_1, \ldots, s_k, t\})$ is an elementary $k - flat$ for every $t \in T$. That is:*

- $t \notin fl(\{s_1, \ldots, s_k\})$
- $fl(\{s_1, \ldots, s_k, t\}) \cap S = \{s_1, \ldots, s_k, t\}.$

▶ **Lemma 2.5** (Bichromatic semi-ordinary line). *Let $X$ and $Y$ be disjoint finite sets in $\mathbb{C}^d$ satisfying the following conditions.*

1. $dim(Y) > \Omega(C^4)$ *where $C$ is the constant in the above corollary.*
2. $|Y| \leq 99|X|$

*Then there exists a line $l$ such that $|l \cap Y| = 1$ and $|l \cap X| \geq 1$.*

At this point we would like to mention that the constants $99, 0.99$ and the one hidden in $\Omega(C^k)$ have more general values given by a parameter $\delta$. For the time being we've fixed them for better exposition. Please see Appendix B for more details.

Using high dimensionality of $A, B$ and the above mentioned corollaries we are able to prove the following theorem which forms the backbone of our algorithm.

▶ **Theorem 2.6.** *For some product gate (say $A$), there exists $k = O(1)$ points $S = \{A_1, \ldots, A_k\}$ and a large set $D \subset A$ such that on projecting $D, B$ to the subspace $W$ defined by $\{A_1 = 0, \ldots, A_k = 0\}$ (and throwing away zeros):*

- *There exists a lines $L = \overrightarrow{B_1' D_1'}$ where $B_1'$ and $D_1'$ are projections of $B_1, D_1$ onto $W$. Also if $B'$ is the projection of $B$ onto $W$ then $L \cap B' = \{B_1'\}$, so the line is a bichromatic semi-ordinary which were discussed in the lemma above.*
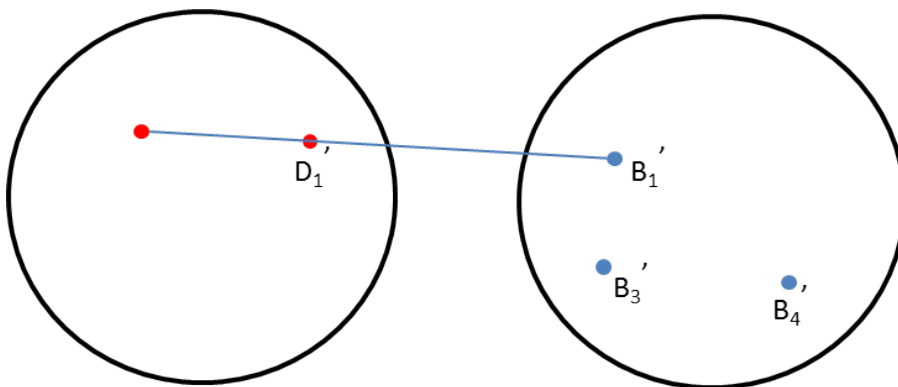


Let's pick one of these lines and see what would have happened in $\mathbb{F}^r$ which led us to this line in $W$.
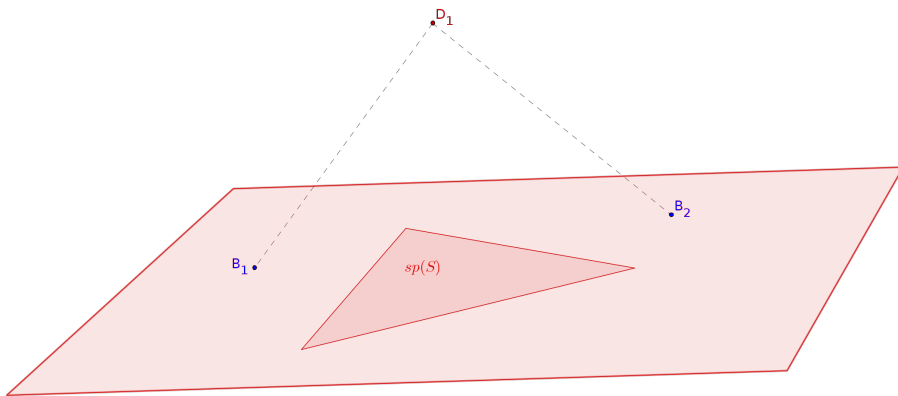


In the picture above the inner triangle denotes $sp(S)$ and the outer parallelogram denotes $sp(S \cup \{B_1\})$. The line in the previous picture i.e. projecting the points onto $W$ has only one blue point implying:

- $sp(S \cup \{B_1\}) \cap B = sp(S \cup \{B_1\})$
- $sp(S \cup \{B_1\} \cup \{D_1\}) \cap B = sp(S \cup \{B_1\})$

Note that this looks very similar to what we had in Subsection 1.5.1. We used this kind of a configuration to recover points using their projections. A similar method is implemented here. Given that such a configuration exists we can come up with the following algorithm.

1. From the set $\mathcal{C}$ guess the set $S = \{A_1, \ldots, A_k\}$ mentioned in the theorem above.
2. Using condition (3) in Claim 2.3 obtain a set $X$ such that $D \subset X \subset A$. This can be done as explained in Algorithm 4. The reader should just assume this at the moment. We need to make sure that $D_1$ comes from $A$ because the algorithm is iterative and we don't want a spurious linear form in $\mathcal{C}$ give any reconstruction. We always want to set some $A_i$'s to 0 so that we only recover $B_j$'s.
3. Iterate over this set $X$ and guess $D_1$.
4. By projecting $f$ to the subspaces $\{A_1 = 0, \ldots, A_k = 0\}$ and $\{D_1 = 0\}$ we get $B'_1$ and $(B_1)_{|_{D_1=0}}$. Because of the diagram above these two projections can be matched and used to reconstruct $B_1$.
5. If no $D_1 \in X$ worked then go to Easy Case since dimension should have fallen.

Basically the algorithm just exploits the existence of the line mentioned in the previous theorem and reconstructs the corresponding $B_1$ (whose projection lies on the line). This reconstruction was possible because this line had only one blue point. After finding $B_1$ we declare it known so that in the next iteration we can remove it's projection when required. We will continue to get such bichromatic semi-ordinary lines till the unkown linear forms in the $B$ set have high dimension. If at any stage this reconstruction is not possible then this dimension would have fallen and we can use the Easy Case.

### 2.2.2.1 Return Type

In all our algorithms we wish to return the reconstructed form of $f$. Since $f$ and the two gates $T_0, T_1$ are to be returned we define an object for it. We call this object Decomposition. We assume having a data type polynomial for general polynomials and pi_sigma for polynomials which are product of linear forms. We use C++ syntax to define our structure.

```
struct decomposition {
  bool iscorrect; // iscorrect will be true if f = M_0 + M_1
  polynomial f;
  pi_sigma M_0;
  pi_sigma M_1;

  // Constructor when a reconstruction is found
  decomposition(polynomial g, pi_sigma A, pi_sigma B){
    iscorrect =true;
    f=g;
    M_0=A;
    M_1=B;
  }

  // Constructor when no reconstruction is found
  decomposition(){
    iscorrect=false;
  }
};
```

## 3 Reconstruction for low rank

Let's recall Definition 1.2 following Theorem 1.1 in Section 1.

▶ **Definition 3.1.** We fix $s$ to be any constant $> \max(C_{2k-1} + k, c_{\mathbb{F}}(4))$ where:
1. $c_{\mathbb{F}}(l) = 3l^2$ is the rank lower bound (see Theorem 1.7) that guarantees non-zeroness of any simple, minimal, $\Sigma\Pi\Sigma(l)$ circuit with rank $> c_{\mathbb{F}}(l)$.

2. $k = c_{\mathbb{F}}(3) + 2$.
3. $\delta$ is some fixed number in $(0, \frac{7-\sqrt{37}}{6})$.
4. $C_k = \frac{C^k}{\delta}$ the constant that appears in Theorem B.4.

Let $r$ be any constant $\geq s$ (In our application we need $s$ and $s+1$). Our main theorem for this section therefore is:

▶ **Theorem 3.2.** *Let $r$ be as defined above. Consider $f(\bar{x}) \in \mathbb{F}[\bar{x}]$, a multivariate homogeneous polynomial of degree $d$ over the variables $\bar{x} = (x_1, \ldots, x_r)$ which can be computed by a $\Sigma\Pi\Sigma(2)$ circuit $C$ over $\mathbb{F}$. Assume that rank of the simplification of $C$ i.e. $Sim(C) = r$. We give a poly($d$) time randomized algorithm which computes $C$ given blackbox access to $f(\bar{x})$.*

We assume $f$ has the following $\Sigma\Pi\Sigma(2)$ representation:

$$f = \tilde{G}(\tilde{\alpha}_0 \tilde{T}_0 + \tilde{\alpha}_1 \tilde{T}_1)$$

where $\tilde{G}, \tilde{T}_i \in \Pi\Sigma_{\mathbb{F}}[\bar{x}]$ are *normal* (i.e. leading non-zero coefficient is 1 in every linear factor) and $\tilde{\alpha}_0, \tilde{\alpha}_1 \in \mathbb{F}$ with $gcd(\tilde{T}_0, \tilde{T}_1) = 1$. The $rank(Sim(C)) = r$ condition then becomes

$$sp(\mathcal{L}(\tilde{T}_0) \cup \mathcal{L}(\tilde{T}_1)) = Lin_{\mathbb{F}}[\bar{x}].$$

Consider the set $T = \mathcal{L}(\tilde{G}) \cup \mathcal{L}(\tilde{T}_0) \cup \mathcal{L}(\tilde{T}_1)$. By abuse of notation we will treat these linear forms also as points in $\mathbb{F}^r$. Since linear factors of $\tilde{G}, \tilde{T}_i$ are normal, two linear factors of $\tilde{G}, \tilde{T}_i$ are LD iff they are same.

**Random Transformation and Assumptions**

Let $\Omega, \Lambda$ be two $r \times r$ matrices such that their entries $\Omega_{i,j}$ and $\Lambda_{i,j}$ are picked independently from the uniform distribution on $[N]$. Here $N = 2^d$. We begin our algorithm by making a few assumptions. All of these assumptions are true with very high probability and we assume them in our algorithm. These assumptions make our work easy by removing redundancy in the co-ordinates. *The idea is to move vectors randomly thereby introducing non-zero coefficients in them.* Consider the standard basis of $\mathbb{F}^r$ given as $\mathcal{S} = \{e_1, \ldots, e_r\}$. Let $E_j = sp(\{e_1, \ldots, e_j\})$ and $E'_j = sp(\{e_{j+1}, \ldots, e_r\})$, clearly $\mathbb{F}^r = E_j \oplus E'_j$. Let $\pi_{W_{E_j}}$ be the orthogonal projection onto $E_j$ w.r.t. this decomposition. Note that $T$ is a finite set of vectors in $\mathbb{F}^r$.

- **Assumption 0:** $\Omega$ is invertible. This is just the complement of event $\mathcal{E}_0$ in Section D and so occurs with high probability.
- **Assumption 1:** For all $t \in T$, $\pi_{W_{E_1}}(\Omega(t)) \neq 0$ i.e. $[\Omega(t)]^1_{\mathcal{S}} \neq 0$ (coefficient of $e_1$ is non-zero). This is the complement of event $\mathcal{E}_1$ in Section D. and so occurs with high probability.
- **Assumption 2:** For all LI sets $\{t_1, \ldots, t_r\} \subset T$, $\{\Omega(t_1), \ldots, \Omega(t_r)\}$ is LI. This essentially means that $\Omega$ is invertible. This is the complement of $\mathcal{E}_2$ in Section D and so occurs with high probability.
- **Assumption 3:** Fix a $k < r$. For all LI sets $\{t_1, \ldots, t_r\} \subset T$, $\{\Omega(t_1), \ldots, \Omega(t_k), \Lambda\Omega(t_{k+1}), \ldots, \Lambda\Omega(t_d)\}$ is LI i.e. is a basis. This is the complement of event $\mathcal{E}_3$ in Section D and so occurs with high probability. It'll be used later in this paper.
- **Assumption 4:** Fix a $k < r$. For all LI sets $\tilde{T} = \{t_1, \ldots, t_r\} \subset T$, define the set $\mathcal{B} = \{\Omega(t_1), \ldots, \Omega(t_k), \Lambda\Omega(t_{k+1}), \ldots, \Lambda\Omega(t_r)\}$. By Assumption 3 this is a basis. Consider any $t \in T$ such that $\Omega(t) \notin sp(\{\Omega(t_1), \ldots, \Omega(t_k)\})$. Then $[\Omega(t)]^{k+1}_{\mathcal{B}} \neq 0$. This event is the complement of $\mathcal{E}_5$ and so it occurs with high probability. We want non-zeroness of co-ordinates even after projecting to a codimension-$k$ subspace. That is where this will be useful.

From now onwards we will assume that all the above assumptions are true. Since all of them occur with very high probability, their complements occur with very low probability and by union bound the union of their complements is a low probability event. So intersection of the above assumptions occurs with high probability and we assume all of them are true. *Note that the assumptions will continue to be true if we scale all linear forms (possibly different scaling for different vectors, but non-zero scalars) in $T$ i.e. if the assumptions were true for $T$ then they would have been true had we started with a scaling of $T$.*

The first step of our algorithm is to apply $\Omega$ to $f$. We have a natural identification between linear forms and points in $\mathbb{F}^r$. This identification converts $\Omega$ into a linear map on $Lin_{\mathbb{F}}[\bar{x}]$ which can be further converted to a ring homomorphism on polynomials by assuming that it preserves the products and sums of polynomials. So $\Omega$ gets applied to all linear forms in the $\Sigma\Pi\Sigma(2)$ representation of $f$. Since $f$ is a degree $d$ polynomial in $r$ variables it has atmost $poly(d^r)$ coefficients. Applying $\Omega$ to each monomial and expanding it takes $poly(d^r)$ time and gives $poly(d^r)$ terms. So computing $\Omega(f)$ takes $poly(d^r)$ time and has $poly(d^r)$ monomials.

Now we try and reconstruct the circuit for $\Omega(f)$. If this reconstruction can be done correctly, we can apply $\Omega^{-1}$ and get back $f$. Note that Assumption 1 tells us that the coefficient of $x_1$ in $\Omega(l)$ is non-zero for all $l$ in $T$. Let $X = \{x_1, \ldots, x_r\}$ and $\bar{x}$ is used for the tuple $(x_1, \ldots, x_r)$. From this discussion we know that:

$$\Omega(f) = \Omega(\tilde{G})(\tilde{\alpha}_0 \Omega(\tilde{T}_0) + \tilde{\alpha}_1 \Omega(\tilde{T}_1)) = G(\alpha_0 T_0 + \alpha_1 T_1)$$

where $\alpha_i$ are chosen such that linear factors of $G, T_i$ have their first coefficient( that of $x_1$) equal to 1. So they are *standard* $\Pi\Sigma$ polynomials. Note that we've used Assumption 1 here. Since we've moved constants to make linear forms standard we can assume $G = \lambda\Omega(\tilde{G}), T_i = \lambda_i\Omega(\tilde{T}_i)$ with $\lambda, \lambda_i \in \mathbb{F}$. Consider some scaling $T_{sc}$ of $T$ such that $\mathcal{X} = \mathcal{L}(G) \cup \mathcal{L}(T_0) \cup \mathcal{L}(T_1)$ is $= \Omega(T_{sc})$. All above assumptions are true for $T_{sc}$ and so we may use the conclusions about $\Omega(T_{sc})$ i.e. $\mathcal{X}$. Also since $\Omega$ is invertible $gcd(T_0, T_1) = 1$.

Let

$$T_i = \prod_{j \in [M]} l_{ij}, i = 0, 1 \text{ and } G = \prod_{k \in [d-M]} G_k$$

with $l_{ij}, G_k$ linear forms (so $d = deg(f)$ ).

For simplicity from now onwards we call $\Omega(f)$ by $f$ and try to reconstruct it's circuit. Once this is done we may apply $\Omega^{-1}$ to all the linear forms in the gates and get the circuit for $f$. This step clearly takes $poly(d^r)$ time in the same way as applying $\Omega$ took. Since $r$ is a constant, the steps described above take $poly(d)$ time overall.

**Known and Unknown Parts**

We also define some other $\Pi\Sigma$ polynomials $K_i, U_i, i = 0, 1$ which satisfy

$$K_i \mid \alpha_i G T_i, U_i = \frac{\alpha_i G T_i}{K_i}.$$

with the extra condition

$$gcd(K_i, U_i) = 1.$$

$K_i$ are the known factors of $\alpha_i G T_i$ and $U_i$ the unknown factors. The *gcd* condition just means that that known and unknown parts of $\alpha_i G T_i$ don't have common factors. In other words linear forms in $\alpha_i G T_i$ are known with full multiplicity. We initialize $K_i = 1$ and during the course of the algorithm update them as and when we recover more linear forms. At the end $K_i = \alpha_i G T_i$ and so we know both gates.

## 3.1 Outline of the algorithm

1. **Set $\mathcal{C}$ of Candidate Linear Forms:** We compute a *poly(d)* size set $\mathcal{C}$ of linear forms which contains $\mathcal{L}(T_i), i = 0, 1$. We will non-deterministically guess from this set $\mathcal{C}$ making only a constant number of guesses everytime(thus polynomial work overall). It is important to note that the uniqueness of our circuit guarantees that our answer if computed can always be tested to be right. For more details on this please see Appendix E. We also give an efficient algorithm to construct this set. See Algorithm 8.

2. **Easy Case:** $\boxed{\mathcal{L}(T_{1-i}) \subsetneq sp(U_i), \text{ for some } i \in \{0, 1\}}$
   So $T_{1-i}$ has a linear factor $l_{(1-i)1}$ such that

   $$sp(\{l_{(1-i)1}\}) \cap sp(U_i) = \{0\} \tag{1}$$

   Let $W = sp(\{l_{(1-i)1}\})$ and extend to a basis of $V$ and in the process obtain another subspace $W' \subset V$ such that $W \oplus W' = V$. We can see from Equation 1 that LI linear forms in $U_i$ remain LI when we project to $W'$. We use this to compute $U_i$ and then since $K_i U_i = \alpha_i G T_i$ we know one of the gates. To find the other gate simply factorize $f - \alpha_i G T_i$. If it factors into a product of linear forms we have the reconstruction.

3. **Medium Case:** $\boxed{dim(sp(T_{1-i}) + sp(T_i)/sp(T_i)) \geq 2 \text{ for some } i \in \{0, 1\}}$
   This case is just to facilitate the Hard Case. We know that $T_{1-i}$ has two linear factors $l_{(1-i)1}, l_{(1-i)2}$ such that $sp(\{l_{(1-i)1}, l_{(1-i)2}\}) \cap sp(T_i) = \{0\}$. We show that the only linear factors of $f$ are those which appear in $G$. So we can first factorize $f$ using Kaltofen's factoring ([14]) and obtain $G$. Update $K_j = G, j = 0, 1$. So $U_j = \alpha_j T_j$ for $j = 0, 1$. Clearly we also have $\mathcal{L}(T_{1-i}) \subsetneq sp(T_i) = sp(U_i)$ and we can go to Easy Case above with $K_i = G$.

4. **Hard Case:** $\boxed{\mathcal{L}(T_{1-i}) \subseteq sp(U_i), \text{ for } i = 0 \text{ and } 1}$
   We know that we are not in Medium Case and so $dim(sp(T_0) + sp(T_1)) - sp(T_i) \leq 1$ for $i = 0, 1$. Also $dim(sp(T_0) + sp(T_1)) = r$ by assumption on the simple rank of our polynomial. So this guarantees that $dim(sp(T_{1-i})) \geq r - 1 \Rightarrow$ (by the condition of this hard case) $dim(sp(U_i)) \geq r - 1$ for $i = 0, 1$. This enables us to use the Quantitative Sylvester Gallai theorems on both sets $\mathcal{L}(T_i), \mathcal{L}(U_i)$.

   ▪ Our first step is to identify a certain "*bad*" $\Pi\Sigma$ factor $I$ of $G$ and get rid of it to get $G = \frac{G}{I}$ and thus $f = \frac{f}{I}$. The factors of $I$ don't satisfy certain properties we need later and so we remove them. Thankfully we have an efficient algorithm to recover $I$. Our algorithm uses something we call a Detector Pair (See 3.4) whose existence is shown using the Quantitative Sylvester Galai Theorems mentioned above.

   ▪ So now our job is to reconstruct $f$ with known (and unknown resp.) parts as $K_0^\star, K_1^\star$ ($U_0^\star, U_1^\star$ resp.).

   ▪ If $sp(U_{1-i})$ becomes low dimensional we may fall in Easy Case and recover the circuit for $f$ directly. Otherwise the same detector pairs then provide certain "*nice*" subspaces corresponding to linear forms in $T_i$. Projection of $U_{1-i}$ onto these subspaces can be easily glued together to recover some linear factors(with multiplicities) of $U_{1-i}$, which will then be multiplied to $K_{1-i}^\star$.

- The process continues as long as $sp(U_{1-i})$ remains high dimensional. As soon as this condition fails we end up in Easy Case and the gates are recovered.

We give algorithms for Easy and Medium cases. Hard Case will require more prepration and will be done after these subsections. From now onwards we assume that we have constructed a $poly(d)$ sized set of linear forms $\mathcal{C}$ which contains $\mathcal{L}(T_i)$ for $i = 0, 1$. We have other structural results about linear forms in this set. See Appendix E for more details and algorithms. Algorithm 8 constructs this set in $poly(d)$ time.

## 3.2 Easy Case

$$\boxed{\mathcal{L}(T_{1-i}) \subsetneq sp(U_i), \text{ for some } i \in \{0,1\}}$$

▶ **Claim 3.3.** *Suppose for some $i \in \{0,1\}$, $\mathcal{L}(T_{1-i}) \subsetneq sp(U_i)$ then we can reconstruct $f$.*

---

**FunctionName :** EasyCase
**input**         : $f \in \Sigma\Pi\Sigma_{\mathbb{F}}(2)[\bar{x}], K_0 \in \Pi\Sigma_{\mathbb{F}}[\bar{x}], K_1 \in \Pi\Sigma_{\mathbb{F}}[\bar{x}], \mathcal{C} \subset Lin_{\mathbb{F}}[\bar{x}])$
**output**        : An object of type *decomposition*

1 **for** $i \leftarrow 0$ **to** 1 **do**
2     **for** *each LI set* $\{l_1, l_2, \ldots, l_r\} \subset \mathcal{C}$ **do**
3        Define $K'_i \leftarrow K_i$;
4        Find $t$ such that $l_1^t \,||\, f$ ;
5        // i.e. $l_1^t \,|\, f$ && $l_1^{t+1} \nmid f$
6        $W \leftarrow sp(\{l_1\}), W' \leftarrow sp(\{l_2, \ldots, l_r\})$;
7        **if** $l_1^t \,||\, K'_i$ **then**
8           $\tilde{f} = \frac{f}{l_1^t}; \tilde{K}_i = \frac{K'_i}{l_1^t}$;
9           **if** $U_i = \frac{\pi_{W'}(\tilde{f})}{\pi_{W'}(\tilde{K}_i)} \in \Pi\Sigma_{\mathbb{F}}[\bar{x}]$ && $f - K_i U_i \in \Pi\Sigma_{\mathbb{F}}[\bar{x}]$ **then** $K_i = K_i U_i$, $K_{1-i} = f - K_i U_i$;
10           **return** *decomposition*$(f, K_0, K_1)$;
11        **end**
12     **end**
13 **end**
14 **return** *decomposition*();

**Algorithm 1:** Easy Case Reconstruction.

---

### Explanation and Correctness Analysis

- The first for loop just guesses the gate with extra dimensions i.e. it's not contained in span of the unknown part of the other gate.
- If for some basis $\{l_1, \ldots, l_r\} \subset \mathcal{C}$ the algorithm actually computes a $\Sigma\Pi\Sigma(2)$ representation in the end then it ought to be correct since the last 'if' also checks if it is correct.
- If our guess for $i$ is correct, we show that there exists a basis $\{l_1, \ldots, l_r\} \subset \mathcal{C}$ for which all conditions will be satisfied and we actually arrive at a $\Sigma\Pi\Sigma(2)$ representation in the end. Since $\mathcal{L}(T_{1-i}) \subsetneq sp(U_i)$ and $\mathcal{L}(T_{1-i}), \mathcal{L}(U_i) \subset \mathcal{C}$ there exists $l_1 \in \mathcal{L}(T_{1-i}) \setminus sp(U_i) \subset \mathcal{C}$. Choose a basis $\{l_2, \ldots, l_s\}$ of $sp(U_i)$, then $\{l_1, \ldots, l_s\}$ is an LI set. Now extend this to a basis $\{l_1, \ldots, l_s, l_{s+1}, \ldots, l_r\} \subset \mathcal{C}$ of $V$. We go over all choices of basis in $\mathcal{C}$ and will arrive at the right one.

- We initialize a dummy polynomial $K_i'$ to represent $K_i$ since we do not want to update $K_i$ till we actually have a solution. Let's assume $l_1^t \,||\, f$ i.e. $l_1^t \mid f$ and $l_1^{t+1} \nmid f$. We know $l_1 \mid T_{1-i} \Rightarrow l_1 \nmid T_i \Rightarrow l_1 \nmid \alpha_i T_i + \alpha_{1-i} T_{1-i}$. Therefore $l_1^t \,||\, G \Rightarrow l_1^t \,||\, \alpha_i G T_i = K_i U_i$. Also $l_1 \notin sp(U_i) \Rightarrow l_1 \nmid U_i$ thus $l_1^t \,||\, K_i \Rightarrow l_1^t \,||\, K_i'$. We remove $l_1^t$ from both $f, K_i'$ to get $\tilde{f}, \tilde{K}_i$. Let $W = sp(\{l_1\})$ and $W' = sp(\{l_2, \dots, l_r\})$, therefore $V = W \oplus W'$. Note that since $l_1 \in \mathcal{L}(T_{1-i})$

$$\pi_{W'}(\tilde{f}) = \pi_{W'}(U_i)\pi_{W'}(\tilde{K}_i)$$

  Since $\pi_{W'}(\tilde{K}_i) \neq 0$, we get $\pi_{W'}(U_i) = \frac{\pi_{W'}(\tilde{f})}{\pi_{W'}(\tilde{K}_i)}$. If $U_i = u_1 \dots u_s$ with $u_j \in W'$, we see that $\pi_{W'}(U_i) = \pi_{W'}(u_1) \dots \pi_{W'}(u_s) = u_1 \dots u_s = U_i$. So we get $U_i$ and hence $\alpha_i G T_i = K_i U_i$. Once $\alpha_i G T_i$ is known we factorize $f - \alpha_i G T_i$ to get $\alpha_{1-i} G T_{1-i}$. For the correct choice of our basis this will factorize completely into a $\Pi\Sigma$ polynomial. Now we update $K_i = K_i U_i$ and $K_{1-i} = f - K_i U_i$ and an object $decomposition(f, K_0, K_1)$. Throughout the algorithm we use Kaltofen's factoring [14] wherever necessary.

- If we were not able to find the $\Sigma\Pi\Sigma(2)$ representation then we return an object $decomposition()$.

**Time Complexity**

We can see above all loops run only $poly(d)$ many times. The most expensive step is choosing $r$ vectors from $\mathcal{C}$. But recall that $r$ is a constant and so this also takes only polynomial time in $d$. Other steps like factoring polynomials (using Kaltofen's factoring algorithm from [14]), taking projection onto known subspaces, divding by polynomials require $poly(d)$ time ($r$ is a constant) as has been explained multiple times before.

## 3.3 Medium Case

$$\boxed{dim\left({}^{sp(T_{1-i}) + sp(T_i)}/{}_{sp(T_i)}\right) \geq 2 \text{ for some } i \in \{0, 1\}}$$

▶ **Claim 3.4.** *If $dim\left({}^{sp(T_{1-i}) + sp(T_i)}/{}_{sp(T_i)}\right) \geq 2$ then $\mathcal{L}(\alpha_i T_i + \alpha_{1-i} T_{1-i}) = \phi$.*

**Proof.** $dim\left({}^{sp(T_{1-i}) + sp(T_i)}/{}_{sp(T_i)}\right) \geq 2 \Rightarrow$, there exists $l_1', l_2' \in \mathcal{L}(T_{1-i}) \setminus sp(T_i)$ be such that $dim(\{l_1', l_2'\} \cup \mathcal{L}(T_i)) = dim(\mathcal{L}(T_i)) + 2$. Assume there exist $l \in \mathcal{L}(\alpha_i T_i + \alpha_{1-i} T_{1-i})$.

$l \mid \alpha_i T_i + \alpha_{1-i} T_{1-i} \Rightarrow l \nmid T_i$ and $l \nmid T_{1-i}$ (since they are coprime)

$0 \neq \alpha_i \prod_{j \in [M]} l_{ij} = -\alpha_{1-i} \prod_{j \in [M]} l_{(1-i)j} \pmod{\{l\}}$.

Thus there exist $l_1, l_2 \in \mathcal{L}(T_i)$ and scalars $\gamma_j, \delta_j, j \in [2]$ such that $l = \gamma_j l_j + \delta_j l_j'$. Since $l \nmid T_0, l \nmid T_1$ we get $\gamma_j, \delta_j$ are non zero.
$\delta_1, \delta_2 \neq 0 \Rightarrow$,

$l_1', l_2' \in sp(\{l\} \cup \mathcal{L}(T_i)) \Rightarrow dim(\{l_1', l_2'\} \cup \mathcal{L}(T_i)) \leq dim(\mathcal{L}(T_i)) + 1$

which is a contradiction. So $\mathcal{L}(\alpha_i T_i + \alpha_{1-i} T_{1-i}) = \phi$.

Therefore the only linear factors of $f$ are present in $G$, which can now be correctly found by using Kaltofen's algorithm [14] and identifying the linear factors. Update $K_j = G$ for $j = 0, 1$, therefore $U_j = T_j$. Also this case implies that $\mathcal{L}(T_{1-i}) \subsetneq sp(T_i) = sp(U_i)$, and so we can use Easy Case.  ◀

So we have the following claim:

▶ **Claim 3.5.** *If the condition in Medium Case is true, the following algorithm reconstructs $f$, if there is a reconstruction.*

---

**FunctionName :** MediumCase
**input**          : $f \in \Sigma\Pi\Sigma_{\mathbb{F}}(2)[\bar{x}], \mathcal{C} \subset Lin_{\mathbb{F}}[\bar{x}])$
**output**         : An object of type *decomposition*

**1** $L \leftarrow Lin(f)$;

**2** // Use Kaltofen's factoring from [14] to compute $Lin(f) \overset{def}{=}$ product of all linear factors of $f$

**3** **if** $EasyCase(f, L, L, \mathcal{C}) \rightarrow iscorrect$ **then**

**4**  |  **return** $EasyCase\ (f, L, L, \mathcal{C})$;

**5** **end**

**6** **return** $decomposition()$;

**Algorithm 2:** Medium Case Reconstruction

---

The above algorithm does exactly what has been explained in the preceeding paragraph. It works in $poly(d)$ time if EasyCase$(f, K_0, K_1, \mathcal{C})$ works in $poly(d)$ time. Kaltofen's factoring and all other steps are $poly(d)$ time.

Now we need to handle the Hard Case. This is quite technical and so we do some more preparation. We devise a technique to get rid of some factors of $f$ to get a new polynomial $f$ without destroying the $\Sigma\Pi\Sigma(2)$ structure. If Easy Case holds for $f$ we stop there itself. Otherwise we will use combination of different subspaces of $V$, project $f$ onto them and glue projections to get gates for $f$.

## 3.4   Detector Pair, Reducing Factors, Hard Case Preparation

Let's recall:

$$g = \frac{f}{G} = \alpha_0 T_0 + \alpha_1 T_1$$

We outline an approach to identify some factors of $f$. These factors will divide $G$ but won't divide $g$. This is going to be useful in the Hard Case. The linear factors left after removing these identified factors will have very strong structural properties and so will be instrumental in reconstruction. The main tool in this identification is a pair $(S, D)$ (defined below) inside one of the $\mathcal{L}(T_i)$'s. This pair will be called a *"Detector Pair"*. It will also decide the subspaces on which we take projections of $f$ and glue back to get the gates.

### Detector Pairs $(S, D)$

Fix $k = c_{\mathbb{F}}(3) + 2$ (See Theorem 1.7 for definition of $c_{\mathbb{F}}(m)$). Let $S = \{l_1, \ldots, l_k\} \subset \mathcal{L}(T_i)$ be an LI set of linear forms. Let $D(\neq \phi) \subseteq \mathcal{L}(T_i)$. We say that $(S, D)$ is a *"Detector Pair"* in $\mathcal{L}(T_i)$ if the following are satisfied for all $l_{k+1} \in D$:

- $\{l_1, \ldots, l_k, l_{k+1}\}$ is an LI set. Let $\mathcal{F} = fl(\{l_1, \ldots, l_k, l_{k+1}\})$. $\mathcal{F}$ is elementary in $\mathcal{L}(T_i)$ i.e. $\mathcal{F} \cap \mathcal{L}(T_i) = \{l_1, \ldots, l_k, l_{k+1}\}$. See Definition B.1.
- $\mathcal{F} \cap \mathcal{L}(T_{1-i}) \subseteq fl(\{l_1, \ldots, l_k\})$ i.e. $\mathcal{F}$ contains only those points from $\mathcal{L}(T_{1-i})$ which lie inside $fl(\{l_1, \ldots, l_k\})$.

### 3.4.1 Identifying Some Factors Which Don't Divide $g$

The two claims below give results about structure of linear forms which divide $g$. The proofs are easy but technical and so we move them to the appendix.

▶ **Claim 3.6.** *Let $(S = \{l_1 \ldots, l_k\}, D)$ be a Detector set in $\mathcal{L}(T_i)$. Let $l_{k+1} \in D$. For a standard linear form $l \in V$, if $l \mid g$ then $l \notin sp(\{l_1, \ldots, l_k\})$.*

**Proof.** See F.1 in appendix. ◀

▶ **Claim 3.7.** *Let $l \in Lin_{\mathbb{F}}[\bar{x}]$ be standard such that $l \mid g$ and $\mathcal{C}$ be the candidate set. Assume $(S = \{l_1, \ldots, l_k\}, D(\neq \phi))$ is a Detector pair in $\mathcal{L}(T_i)$. Then $|\mathcal{L}(T_{1-i}) \cap (fl(S \cup \{l\}) \setminus fl(S))| \geq 2$. That is the flat $fl(\{l_1, \ldots, l_k, l\})$ contains atleast two distinct points from $\mathcal{L}(T_{1-i})(\subseteq \mathcal{C})$ outside $fl(\{l_1, \ldots, l_k\})$.*

**Proof.** See F.2 in appendix. ◀

▶ **Claim 3.8.** *Suppose $(S = \{l_1, \ldots, l_k\}, D(\neq \phi))$ is a Detector Pair in $\mathcal{L}(T_i)$. The following algorithm identifies some factors in $\mathcal{L}(G) \setminus \mathcal{L}(g)$. It returns the product of all linear forms identified.*

---

    **FunctionName** : IdentifyFactors
    **input**            : $f \in \Sigma\Pi\Sigma_{\mathbb{F}}(2)[\bar{x}], \mathcal{C} \subset Lin_{\mathbb{F}}[\bar{x}], S = \{l_1, \ldots, l_k\} \subset Lin_{\mathbb{F}}[\bar{x}])$
    **output**         : a $\Pi\Sigma_{\mathbb{F}}[\bar{x}]$ polynomial

**1**  $\mathbf{I} = 1$, *bool flag*;
**2**  **for** *each factor $l$ of $f$* **do**
**3**     $flag = false$;
**4**     **if** $l, l_1, \ldots, l_k$ *are LI* **then**
**5**         **for** $l'_1 \neq l'_2 \in \mathcal{C} \setminus fl(\{l_1, \ldots, l_k\})$ **do**
**6**             **if** $l'_1, l'_2 \in sp(\{l, l_1, \ldots, l_k\})$ **then** $flag = true$;
**7**             *break*;
**8**         **end**
**9**     **end**
**10**    **if** *!flag* **then**
**11**        $\mathbf{I} = \mathbf{I} \times l$;
**12**    **end**
**13** **end**
**14** **return I**;

**Algorithm 3:** Identify Factors.

---

**Proof.** The proof of the claim is a part of Lemma 3.9 below. ◀

#### 3.4.1.1 Time Complexity

Since $\mathcal{C}$ has size $poly(d)$ and $deg(f) = d$, the nested loops run $poly(d)$ times. $k, r$ are constants so checking linear independence of $k + 1$ linear forms in $r$ variables takes constant time. Checking if some vectors belong to a $k + 1$ dimensional space also takes constant time. Multiplying linear forms to $\mathbf{I}$ takes $poly(d)$ time. So overall the algorithm runs in $poly(d)$ time.

So the above algorithm identified a factor **I** of $G$ for us. Let us define new polynomials

$$G = \frac{G}{\mathbf{I}} = \prod_{t \in [N_1]} G_t$$

and

$$f = \frac{f}{\mathbf{I}} = G(\alpha_0 T_0 + \alpha_1 T_1)$$

▶ **Lemma 3.9.** *The following are true:*
1. *If $l \mid I$ (i.e. $l$ was identified) then $l \in \mathcal{L}(G) \setminus \mathcal{L}(g)$.*
2. *If $l \mid G$ (i.e. $l$ was retained) then $(fl(\{l_1, \ldots, l_k, l\}) \setminus fl(\{l_1, \ldots, l_k\})) \cap (\mathcal{L}(T_{1-i}) \cup (\mathcal{L}(T_i) \setminus D)) \neq \phi$ that is:*
   *$(fl(\{l_1, \ldots, l_k, l\}) \setminus fl(\{l_1, \ldots, l_k\}))$ contains a point from $\mathcal{L}(T_i) \setminus D$ or $\mathcal{L}(T_{1-i})$.*
3. *If $l \mid G$ and $l_{k+1} \in D$ then $l \notin sp(\{l_1, \ldots, l_k, l_{k+1}\})$.*

**Proof.** See F.3 in appendix.                                         ◀

## 3.4.2   Overestimating the set $D$ of the detector pair $(S, D)$

Lemma 3.9 is going to help us actually find an overestimate of $D$ corresponding to $S = \{l_1, \ldots, l_k\}$ in the detector pair $(S, D)$ as described in the lemma below. This will be important since we need $D$ during our algorithm for the Hard Case.

▶ **Lemma 3.10.** *Let $(S = \{l_1, \ldots, l_k\}, D)$ be a detector in $\mathcal{L}(T_i)$. For each $(l, l_j) \in \mathcal{C} \times S$ define the space $U_{\{l,l_j\}} = sp(\{l, l_j\})$. Extend $\{l, l_j\}$ to a basis and in the process obtain $U'_{\{l,l_j\}}$ such that $V = U_{\{l,l_j\}} \oplus U'_{\{l,l_j\}}$. Define the set:*

$$X = \{l \in \mathcal{C} : \pi_{U'_{\{l,l_j\}}}(f) \neq 0, \text{ for all } l_j \in S\}$$

*Then $D \subset X \subset \mathcal{L}(T_i)$.*

**Proof.** See F.4 in appendix.                                         ◀

This set $X$ is an overestimate of $D$ inside $\mathcal{L}(T_i)$ and also easy to compute. Given $S$ we may easily construct $X$ in time $poly(d)$ because of it's simple description. Let's give an algorithm to compute $X$ given $f, S, \mathcal{C}$.

▶ **Claim 3.11.** *Algorithm 4 computes the overestimate $X$ of $D$ as discussed above.*

### 3.4.2.1   Time Complexity

Inside the inner for loop we look for $(r - 2)$ linear forms from $\mathcal{C}$. $|\mathcal{C}| = poly(d)$ and $r$ is a constant and so this step only needs $poly(d)$ time. The nested loops run polynomially many times. Checking linear independece of $r$ linear forms and projecting to known constant dimensional subspaces also take $poly(d)$ time as has been discussed before. So the algorithm runs in $poly(d)$ time.

```
FunctionName : OverestimateDetector
input          : f ∈ ΣΠΣ_F(2)[x̄], S = {l_1, ..., l_k} ⊂ Lin_F[x̄], C ⊂ Lin_F[x̄])
output         : Set of linear forms
1  bool flag;
2  Define X ← φ;
3  for  each l ∈ C do
4  │   flag = true;
5  │   for  each l_j ∈ S with {l, l_j} LI do
6  │   │   Find {l'_1, ..., l'_{r-2}} ⊂ C such that {l, l_j, l'_1, ..., l'_{r-2}} is LI;
7  │   │   U ← Fl ⊕ Fl_j; U' ← Fl'_1 ⊕ ... ⊕ Fl'_{r-2};
8  │   │   if π_{U'}(f) == 0 then
9  │   │   │   flag = false;
10 │   │   │   break;
11 │   │   end
12 │   end
13 │   if flag then
14 │   │   X ← X ∪ {l};
15 │   end
16 end
17 return X;
```

**Algorithm 4:** Overestimate Detector.

## 3.5 Hard Case

$$\boxed{\mathcal{L}(T_{1-i}) \subseteq sp(U_i), \text{ for } i = 0 \text{ and } 1}$$

This subsection will involve the most non trivial ideas. We handled $dim\left(sp(T_{1-i}) + sp(T_i)/sp(T_i)\right) \geq 2$ in the Medium Case (see Subsection 3.3) completely, so let's assume $dim\left(sp(T_{1-i}) + sp(T_i)/sp(T_i)\right) \leq 1 \Rightarrow dim(\mathcal{L}(T_{1-i}) \cup \mathcal{L}(T_i)) \leq dim(\mathcal{L}(T_i)) + 1$ for both $i = 0, 1$. We already know that $rank(f) = r$, implying $dim(\mathcal{L}(T_i) \cup \mathcal{L}(T_{1-i})) = r$. Thus for $i = 0, 1$; $dim(\mathcal{L}(T_i)) \geq r - 1$. This works in our favour for applying the quantitative version of the Sylvester Gallai theorems given in [3]. To be precise we will use Lemma B.6 from Appendix B in this paper.

1. Our first application (see Lemma 3.13) of Quantitative Sylvester Gallai will help us prove the existence of a Detector pair $(S = \{l_1, ..., l_k\}, D)$ in $\mathcal{L}(T_i)$ with $k = c_F(3) + 2$ (See definition of $c_F(.)$ in Theorem 1.7) and large size of $D$. For this we will only need $dim(\mathcal{L}(T_i)) \geq C_{2k-1}$ for $i = 0, 1$ (see Appendix B for definition of $C_{2k-1}$). From Definition 1.2 we know that this is true with $k = c_F(3) + 2$.

2. The above point shows the existence of a detector pair $(S, D)$ in $\mathcal{L}(T_i)$ with large $|D|$. So now we go back to Subsection 3.4 and remove some factors of $f$ to get $f = G(\alpha_0 T_0 + \alpha_1 T_1)$ such that linear factors of $G$ satisfy properties given in Lemma 3.9. We also compute the overestimate $X$ of $D$ using Algorithm 4. Let the known and unknown parts of $f$ be $K_0^\star, K_1^\star$ and $U_0^\star, U_1^\star$ respectively. If for some $i \in \{0, 1\}$, $\mathcal{L}(T_i) \subsetneq sp(U_{1-i})$ then we are in Easy Case for $f$ and can recover the gates for $f$. Otherwise for both $i = 0, 1$; $\mathcal{L}(T_i) \subseteq sp(U_{1-i}) \Rightarrow dim(\mathcal{L}(U_{1-i})) \geq r - 1$ and we continue with reconstruction below.

3. Next to actually reconstruct linear forms in $U_{1-i}$, we will use it's high-dimensionality $(\geq r - 1 \geq C_{2k-1})$ discussed above. Lemma B.6 from Appendix B will enable us to

prove the existence of a $d_1 \in D$ which together with the set $S$ found above will give the existence of a "*Reconstructor*" (see Claim C.4 and Algorithm 7) which recovers some linear factors of $U_{1-i}$ with multiplicity (see Theorem 3.14).

### 3.5.1    Large Size of Detector Sets

W.l.o.g. we assume $|\mathcal{L}(T_0)| \leq |\mathcal{L}(T_1)|$. First we point out a simple calculation that will be needed later. For $\delta \in (0, \frac{7-\sqrt{37}}{6})$ and $\theta \in (\frac{3\delta}{1-\delta}, 1-3\delta)$, let $v(\delta, \theta)$ be defined as follows:

$$v(\delta, \theta) = \begin{cases} 1 - \delta - \theta & \text{if } |\mathcal{L}(T_0)| \leq \theta|\mathcal{L}(T_1)| \\ (1-\delta)(1+\theta) - 1 & \text{if } \theta|\mathcal{L}(T_1)| < |\mathcal{L}(T_0)| \leq |\mathcal{L}(T_1)| \end{cases}$$

▶ **Claim 3.12.** *The following is true*

$$\frac{(2 - v(\delta, \theta))}{v(\delta, \theta)} \leq \frac{1-\delta}{\delta} \ .$$

**Proof.** See G.1 in appendix.                                                      ◀

▶ **Lemma 3.13.** *Let $k = c_\mathbb{F}(3)+2$ (see defn of $c_\mathbb{F}(m)$ in Theorem 1.7). Fix $\delta, \theta$ in range given in Claim 3.12 above. Then for some $i \in \{0, 1\}$ there exists a Detector $(S = \{l_1, \ldots, l_k\}, D)$ in $\mathcal{L}(T_i)$ with $|D| \geq v(\delta, \theta) \max(|\mathcal{L}(T_0)|, |\mathcal{L}(T_1)|)$.*

**Proof.** See G.2 in appendix.                                                      ◀

### 3.5.2    Assuming $\mathcal{L}(T_i) \subseteq sp(\mathcal{L}(U_{1-i}))$ and reconstructing factors of $U_{1-i}$

Let's begin by stating our main reconstruction theorem for this Subsubsection. We will go through several steps to prove it:

▶ **Theorem 3.14.** *There exist pairwise disjoint LI sets $S_0, S_1, S_2$ with $S_0 \cup S_1 \cup S_2$ being a basis of $V = Lin_\mathbb{F}[x_1, \ldots, x_r] \simeq \mathbb{F}^r$, and non constant polynomials $P, Q$ dividing $U_{1-i}$ such that $P \mid Q$ and $(Q, P, S_0, S_1, S_2)$ is a Reconstructor.*

Once we know this result we actually recover $P$ by computing $\pi_{W'_0}(Q)$ and $\pi_{W'_1}(Q)$ and then using Algorithm 7. We state this in the following corollary. Proof is given as Algorithm 5

▶ **Corollary 3.15.** *Using $f, K_{1-i}, S_0, S_1, S_2$ from above we can compute $\pi_{W'_0}(Q), \pi_{W'_1}(Q)$ for $Q$ defined in the proof above.*

Before going to the proof let's do some more more preparation.

Consider the set of linear forms $\mathcal{X} = \mathcal{L}(G) \cup \mathcal{L}(T_0) \cup \mathcal{L}(T_1)$. We know that $sp(\mathcal{X}) = V = Lin_\mathbb{F}[\bar{x}] \simeq \mathbb{F}^r$ (By abuse of notation we will use linear forms as points in $\mathbb{F}^r$ wherever required). Let $(S_0 = \{l_1, \ldots, l_k\}, D)$ be a detector in $\mathcal{L}(T_i)$ with $|D| \geq v(\delta, \theta) \max(|\mathcal{L}(T_0)|, |\mathcal{L}(T_1)|)$ as obtained in the preceeding discussion.

Define $W_0 = sp(S_0)$ and extend $S_0$ to a basis $\{l_1, \ldots, l_k, l'_{k+1}, \ldots, l'_r\}$. Now it's time to use the other random matrix $\Lambda$. Since we had applied $\Omega$ in the beginning, $\{\Omega^{-1}(l_1), \ldots, \Omega^{-1}(l_k)\}$ are linear forms in our input polynomial for this section. By Assumption 3 we know that the set

$$\{\Omega(\Omega^{-1}l_1), \ldots, \Omega(\Omega^{-1}l_k), \Lambda\Omega(\Omega^{-1}l'_{k+1}), \ldots, \Lambda\Omega(\Omega^{-1}l'_r)\}$$

is LI. Let $l_j = \Lambda l'_j, j \in \{k+1, \ldots, r\}$. So $\mathcal{B} = \{l_1, \ldots, l_r\}$ is a basis. and define $W_0^{\perp} = sp(\{l_{k+1}, \ldots, l_r\})$. Clearly $V = W_0 \oplus W_0^{\perp}$.

By Assumption 4 for any $l \in \mathcal{X} \setminus W_0$, $[l]_{\mathcal{B}}^{k+1} \neq 0$. We re-normalize all linear forms in $\mathcal{X} \setminus W_0$ making sure that the coefficient of $l_{k+1}$ is 1 in them. From now onwards this will be assumed.

With this notation we proceed towards detecting linear factors of the unknown parts. But first let's show that even after projecting onto $W_0^{\perp}$, the detector is larger in size (upto a function of $\delta$) compared to one of the unknown parts.

▶ **Lemma 3.16.** *The following are true:*
1. $dim(\pi_{W_0^{\perp}}(\mathcal{L}(U_{1-i}))) > C_4$
2. $\pi_{W_0^{\perp}}(\mathcal{L}(U_{1-i})) \cap \pi_{W_0^{\perp}}(D) = \phi$
3. $|\pi_{W_0^{\perp}}(\mathcal{L}(U_{1-i})) \setminus \{0\}| \leq \frac{1-\delta}{\delta}|\pi_{W_0^{\perp}}(D)|$

**Proof.** See G.3 in appendix. ◀

This lemma enables us to apply Lemma B.6 from Appendix B. Consider the sets $Y = \pi_{W_0^{\perp}}(\mathcal{L}(U_{1-i})) \setminus \{0\}$ and $X = \pi_{W_0^{\perp}}(D)$. We've shown all conditions in Lemma 2.5, so there exists a line $\vec{L}$ (called a "*semiordinary bichromatic*" line) in $W_0^{\perp}$ such that $|\vec{L} \cap Y| = 1$ and $|\vec{L} \cap X| \geq 1$.

Let's prove another short lemma which is useful for technical reasons.

▶ **Lemma 3.17.** *For any subspace $W_0'$ such that $V = W_0 \oplus W_0' = W_0 \oplus W_0^{\perp}$ there is a line $\vec{L} \subset W_0'$ such that*
1. $|\vec{L} \cap \pi_{W_0'}(D)| \geq 1$
2. $|\vec{L} \cap (\pi_{W_0'}(\mathcal{L}(U_{1-i})) \setminus \{0\})| = 1$

**Proof.** We have the following commutative diagram:

$$
\begin{array}{ccc}
 & V & \\
\pi_{W_0'} \downarrow & & \searrow \pi_{W_0^{\perp}} \\
W_0' & \xleftarrow{\pi_{W_0'}} & W_0^{\perp}
\end{array}
$$

Let $v = w + w^{\perp} \in V$ where $w \in W_0, w^{\perp} \in W_0^{\perp}$, then

$$\pi_{W_0'}(\pi_{W_0^{\perp}}(v)) = \pi_{W_0'}(w^{\perp}) = \pi_{W_0'}(w^{\perp}) + \pi_{W_0'}(w) = \pi_{W_0'}(v)$$

So $\pi_{W_0'} = \pi_{W_0'} \circ \pi_{W_0^{\perp}}$

Next let $T : V \to V$ be any bijection then $T(A \cap B) = T(A) \cap T(B)$ and therefore $|A \cap B| = |T(A) \cap T(B)|$. Since the maps above are projections one can easily see that $\pi_{W_0'} : W_0^{\perp} \to W_0'$ is an isomorphism where the inverse of any $w' \in W_0'$ is given as $\pi_{W_0^{\perp}}(w')$. Call this map $T$. Now any linear isomorphism between vector spaces also preserves affine dependence since:

$$T(\lambda u + (1-\lambda)v) = \lambda T(u) + (1-\lambda)T(v).$$

So image of a line is a line. Let $\vec{L}$ be the line obtained in Lemma 3.16.

- $T(\vec{L})$ is a line in $W_0'$.

- $|T(\vec{L}) \cap \pi_{W_0'}(D)| = |T(\vec{L}) \cap T(\pi_{W_0^\perp}(D))| = |\vec{L} \cap \pi_{W_0^\perp}(D)| \geq 1$
- $|T(\vec{L}) \cap \pi_{W_0'}(\mathcal{L}(U_{1-i}))| = |T(\vec{L}) \cap T(\pi_{W_0^\perp}(\mathcal{L}(U_{1-i})))| = |\vec{L} \cap \pi_{W_0^\perp}(\mathcal{L}(U_{1-i}))|$

Since $T$ is a linear isomorphism $0 \in \pi_{W_0^\perp}(\mathcal{L}(U_{1-i})) \Leftrightarrow 0 \in T(\pi_{W_0^\perp}(\mathcal{L}(U_{1-i}))) = \pi_{W_0'}(\mathcal{L}(U_{1-i}))$ and $0 \in \vec{L} \Leftrightarrow 0 \in T(\vec{L})$, therefore the third condition above is same as

$$|T(\vec{L}) \cap (\pi_{W_0'}(\mathcal{L}(U_{1-i})) \setminus \{0\})| = |\vec{L} \cap (\pi_{W_0^\perp}(\mathcal{L}(U_{1-i})) \setminus \{0\})| = 1$$

So the lemma is true with $\vec{L}$ being the line $T(\vec{L})$ obtained in this proof.     ◄

Finally it's time to give the proof of Theorem 3.14. Let $d_1 \in D$ such that $\pi_{W_0^\perp}(d_1) \in \vec{L}$ where $\vec{L}$ was the line obtained rightafter Lemma 3.16. Since coefficient of $l_{k+1}$ is non-zero in $d_1$, $\{l_1, \ldots, l_k, d_1, l_{k+2}, \ldots, l_r\}$ is also a basis. Define $S_0 = \{l_1, \ldots, l_k\}, S_1 = \{d_1\}, S_2 = \{l_{k+2}, \ldots, l_r\}, W_i = sp(S_i), W_i' = \bigoplus_{j \neq i} W_j$. Note this implies $V = W_0 \oplus W_0'$ and so Lemma 3.17 above can be used. Let $\vec{L}$ be the line the Lemma 3.17 gives. *By re-normalization we also assume that all linear forms in $\mathcal{X} \setminus W_0'$ have coefficient of $d_1$ equal to 1.*

### Proof of Theorem 3.14

We show this in steps:
- Let $S_0, S_1, S_2$ be as defined in the discussion above.
- Let $Q$ be the largest factor of $U_{1-i}$ such that for all linear forms $q \mid Q$, $\pi_{W_2}(q) \neq 0$. So $\pi_{W_2}(Q) \neq 0$ and if $u^\star \mid \frac{U_{1-i}}{Q}$ is a linear form then $\pi_{W_2}(u^\star) = 0$. Let $P$ be the $\Pi\Sigma$ polynomial with the largest possible degree such that for all linear factors $p$ of $P$, $\pi_{W_0'}(p) \in \vec{L} \cap (\pi_{W_0'}(\mathcal{L}(U_{1-i})) \setminus \{0\})$. Clearly $P$ is non constant since $|\vec{L} \cap (\pi_{W_0'}(\mathcal{L}(U_{1-i})) \setminus \{0\})| = 1$. Clearly $\pi_{W_0'}(P) \neq 0 \Rightarrow P \mid Q$. Then $(Q, P, S_0, S_1, S_2)$ is a *Reconstructor* (See Subsection C for definition) for $P$. Let's check this is true:
  - $\pi_{W_2}(Q) \neq 0$ - By definition of $Q$ we know this for all it's factors and therefore for $Q$ itself.
  - $\pi_{W_0'}(P) = \pi_{W_0'}(p)^t$, for some linear form $p \mid P$ (since $|\vec{L} \cap (\pi_{W_0'}(\mathcal{L}(U_{1-i})) \setminus \{0\})| = 1$).
  - Let $q \mid \frac{Q}{P}$ such that $gcd(\pi_{W_2}(P), \pi_{W_2}(q)) \neq 1 \Rightarrow$ there exists some linear factor $p \mid P$ such that $\pi_{W_2}(p), \pi_{W_2}(q)$ are LD. $\{\pi_{W_2}(p), \pi_{W_2}(q)\}$ are LD and non-zero implies $q \in sp(\{l_1, \ldots, l_k, d_1, p\}) \Rightarrow \pi_{W_0'}(q) \in sp(\{\pi_{W_0'}(d_1), \pi_{W_0'}(p)\}) = sp(\{d_1, \pi_{W_0'}(p)\})$. So clearly $\pi_{W_0'}(q) \in sp(\{d_1, \pi_{W_0'}(p)\})$. Since coefficient of $d_1$ in $\pi_{W_0'}(q), d_1$, and $\pi_{W_0'}(p)$ is 1, and therefore using Lemma 1.10 it's easy to see that $\pi_{W_0'}(q) \in fl(\{d_1, \pi_{W_0'}(p)\}) = \vec{L}$. Since $Q \mid U_{1-i}$ we have $\pi_{W_0'}(q) \in \pi_{W_0'}(\mathcal{L}(U_{1-i})) \setminus \{0\} \Rightarrow \pi_{W_0'}(q) \in \vec{L} \cap (\pi_{W_0'}(\mathcal{L}(U_{1-i})) \setminus \{0\}) = \{\pi_{W_0'}(p)\}$ which can't be true since $P$ is the largest polynomial dividing $Q$ where linear factors have this property and $q \nmid P$. So such a $q$ does not exist.

Now we give the algorithm for reconstruction in this case, see Algorithm 5.

### 3.5.2.1   Correctness

Let's assume we returned an object *obj* of type decomposition.
1. **If** $obj \rightarrow iscorrect == true$: then we ought to be right since we check if $obj \rightarrow f = obj \rightarrow M_0 + obj \rightarrow M_1$. Since the representation is unique this will be the correct answer.
2. **If** $obj \rightarrow iscorrect == false$: Let's assume $f$ actually has a $\Sigma\Pi\Sigma(2)$ representation. If we were in Easy Case or Medium Case we would have already found the circuit using their algorithms. So we are in the Hard Case. So by Lemma 3.13 there exists $i$ such that

**FunctionName :** HardCase
**Fix** $: k = c_{\mathbb{F}}(3) + 2$
**input** $: f \in \Sigma\Pi\Sigma_{\mathbb{F}}(2)[\bar{x}], \mathcal{C} \subset Lin_{\mathbb{F}}[\bar{x}], \Lambda \in \mathbb{F}^{r \times r}$
**output** : An object of type *decomposition*

```
 1  for i ← 0 to 1 do
 2      for each LI B' = {l₁,...,l_k, l'_{k+1},...,l'_r} ⊂ C do
 3          S₀ = {l₁,...,l_k};
 4          for j ← k + 1 to r do
 5              l_j ← Λ(l'_j);
 6          end
 7          if B = {l₁,...,l_r} is LI then
 8              I ← IdentifyFactors(f, C, S₀);
 9              if I | f then
10                  f ← f/I, K₀⋆ = 1, K₁⋆ = 1, X ← OverestDetector(f⋆, C, S₀);
11                  while deg(K⋆_{1−i}) < deg(f) do
12                      if EasyCase(f, K₀⋆, K₁⋆, C) → iscorrect then
13                          return object decomposition(f, IK₀⋆, IK₁⋆);
14                      end
15                      else
16                          for each d₁ ∈ X do
17                              if B₂ = {l₁,...,l_k, d₁, l_{k+2},...,l_r} is LI then
18                                  V_j = 𝔽l_j, j ∈ [r] \ {k+1}, V_{k+1} = 𝔽d₁, V'_j = ⊕_{t∈[r]\{j}} V_t;
19                                  S₀ = {l₁,...,l_k}, S₁ = {u_{k+1}}, S₂ = {l_{k+2},...,l_r};
20                                  W_j = sp(S_j), W'_j = ⊕_{j₁≠j} W_{j₁} for j ∈ {0,1,2};
21                                  Q₀ = π_{V'₁}(f)/π_{V'₁}(K⋆_{1−i}), Q₁ = π_{W'₁}(f)/π_{W'₁}(K⋆_{1−i});
22                                  if Q₀, Q₁ ∈ ΠΣ[x̄] and non-zero then
23                                      for q₀ | Q₀ && q₀ ∈ W'₂, q₁ | Q₁ && q₁ ∈ W'₂ do
24                                          Q₀ = Q₀/q₀, Q₁ = Q₁/q₁;
25                                      end
26                                      Q₀ = π_{W'₀}(Q₀);
27                                      if deg(Reconstructor(Q₀, Q₁, S₀, S₁, S₂)) ≥ 1 then
28                                          K⋆_{1−i} ← K⋆_{1−i} × Reconstructor(Q₀, Q₁, S₀, S₁, S₂);
29                                      end
30                                  end
31                              end
32                          end
33                      end
34                  end
35                  if f − IK⋆_{1−i} ∈ ΠΣ[x̄] then
36                      M₀ = IK⋆_{1−i}, M₁ = f − M₀, return new object
                          decomposition(f, M₀, M₁);
37                  end
38              end
39          end
40      end
41  end
42  return decomposition();
```

**Algorithm 5:** Hard Case Reconstruction.

$\mathcal{L}(T_i)$ has a detector pair $(S_0, D)$ with $|D|$ large. For this $i$ there exists such an $S_0$, so sometime during the algorithm we would have guessed the correct $i$ and the correct $S_0$. *Now let's analyze what happens inside the while and the third for loop when the first two guesses are correct.* Note that this also implies that the $I$ we have identified is correct and now we need to solve for

$$f = G(\alpha_0 T_0 + \alpha_1 T_1)$$

Let $K_0^\star, K_1^\star$ (initialized to 1) be the known parts of the gates for this polynomial $f$ and $U_0^\star, U_1^\star$ be the unknown parts. Note that $T_0, T_1$ are same for both polynomials so $rank(f) = rank(f)$ and for $j = 0, 1$; $K_j \mid GT_j$.

*Assume until the $m^{th}$ iteration of the while loop $K_t^\star \mid GT_t$ for $t \in \{0, 1\}$, we show that after the $(m+1)^{th}$ iteration, this property continues to hold and $deg(K_{1-i}^\star)$ increases.*

- If after the $m^{th}$ iteration of the while loop for some $j \in \{0, 1\}$, $\mathcal{L}(T_j) \subsetneq sp(\mathcal{L}(U_{1-j}^\star))$ we are in Easy Case for $f$. The first step in while loop is to call EasyCase$(f, \mathcal{C}, K_0^\star, K_1^\star)$. This will clearly recover the circuit for $f$ and return true since $K_t^\star \mid GT_t$ for $t \in \{0, 1\}$. However this does not happen so for both $j = 0, 1$, we have $\mathcal{L}(T_i) \subsetneq \mathcal{L}(U_{1-i})$. This means that we can use the ideas in Subsection 3.5.2, specifically Theorem 3.14.
- The first two guesses are correct imply that $D \subseteq X \subseteq \mathcal{L}(T_i)$.
- If $d$ gets rejected then $K_t, t \in \{0, 1\}$ remain unchanged. If some $d_1$ does not get rejected then since $d_1 \in \mathcal{L}(T_i)$, $Q_0 = \pi_{V_1'}(U_{1-i})$ is a non zero $\Pi\Sigma$ polynomial. Then some factors (the ones $\in W_2'$) are removed from $Q_0$. Also on projecting to $W_0'$ this still remains non-zero (as $d_1$ was not rejected).
- We know that $d_1 \in \mathcal{L}(T_i)$ and $d_1$ not getting rejected implies that $Q_1 = \pi_{W_1'}(U_{1-i})$ is a non-zero $\Pi\Sigma$ polynomial. We again remove some factors (i.e. the ones in $W_2'$) from $Q_1$. The non-zeroness of $Q_0, Q_1$ imply that $Q_0 = \pi_{W_1'}(Q), Q_1 = \pi_{W_1'}(Q)$ i.e. they are projections of the same polynomial $Q$ which is the largest factor of $U_{1-i}$ with the property that any linear form $q \mid Q$ is not in $W_2' = W_0 \oplus W_1$.
- $d_1$ was not rejected implies that $Reconstructor(Q_0, Q_1, S_0, S_1, S_2)$ returned a non-trivial polynomial $P$. This has to be a factor of $Q$ by Claim C.6 following Algorithm 7 and therefore a factor of $U_{1-i}$.
- Proof of Theorem 3.14 implies that in every iteration atleast some $d_1$ will not be rejected.
- So clearly the new $K_{1-i}^\star = K_{1-i}^\star \times P$ divides $GT_{1-i}$. $K_i$ remains unchanged. Therefore even after the $(m+1)^{th}$ iteration $K_t \mid GT_t$ for both $j = 0, 1$ but degree of $K_{1-i}^\star$ increases.

So the while loop cannot run more than $deg(f)$ times and in the end $GT_{1-i}$ will be reconstructed completely and correctly and we should have returned $obj$ with $obj \to iscorrect = true$. Therefore we have a contradiction and so $f$ did not have a $\Sigma\Pi\Sigma(2)$ circuit and we correctly returned false.

## Running Time

- First for loop runs twice.
- Inside it chossing $r$ linear forms from $\mathcal{C}$ ($|\mathcal{C}| = poly(d)$) takes $poly(d)$ time.
- Applying $\Lambda$ to $r - k$ vectors takes $poly(r) = O(1)$ time.

- Checking if a set of size $r$ inside $\mathbb{F}^r$ is LI takes $poly(r) = O(1)$ time since it is equivalent to computing determinant.
- $IdentifyFactors()$ takes $poly(d)$ time and computing $f$ also takes $poly(d)$ time.
- $OverestDetector()$ runs in $poly(d)$ time.
- while loop runs atmost $d$ times
- $EasyCase$ runs in $poly(d)$ time and so does polynomial multiplication.
- $X \subseteq \mathcal{L}(T_i)$ and $|\mathcal{L}(T_i)| \leq deg(f)$ and so for loop runs $d$ times.
- Change of bases in $\mathbb{F}^r$ and application to a polynomial of degree $d$ takes $poly(d)$ time.
- Therefore projecting to subspaces also takes $poly(d)$ time.
- $Reconstructor()$ runs in $poly(d)$ time (since $r$ is a constant) and so does polynomial multiplication and factoring by [14].

Since all of the above steps run in $poly(d)$ time, nesting them a constant number of times also takes $poly(d)$ time. Therefore the running time of our algorithm is $poly(d)$.

## 3.6 Algorithm including all cases

The algorithm we give here will be the final algorithm for rank $r$ $\Sigma\Pi\Sigma$ polynomials. It will use the previous three cases. Our input will be a $\Sigma\Pi\Sigma(2)$ polynomial $f(x_1, \ldots, x_r)$ and output will be a circuit computing the same.

---

**FunctionName :** RECONSTRUCT
**input**             : $f \in \Sigma\Pi\Sigma_{\mathbb{F}}(2)[\bar{x}]$
**output**            : An object of type *decomposition*

**1** *decomposition* obj;
**2** $(\Omega_{i,j}), (\Lambda_{i,j})$, $r \times r$ matrices with entries chosen uniformly randomly from $[N]$;
**3** $L_i(\bar{x}) \leftarrow \sum\limits_{k=1}^{r} \Omega_{i,k} x_k$;
**4** $f(x_1, \ldots, x_r) \leftarrow f(L_1(\bar{x}), \ldots, L_r(\bar{x}))$;
**5** $\mathcal{C} \leftarrow Candidates(f(x_1, \ldots, x_r))$;
**6 if** $MediumCase(f, \mathcal{C})) \rightarrow iscorrect$ **then**
**7** | obj $\leftarrow$ MediumCase($f, \mathcal{C}$);
**8 end**
**9 else if** $EasyCase(f, K_0, K_1, \mathcal{C}) \rightarrow iscorrect$ **then**
**10** | obj $\leftarrow$ EasyCase($f, K_0, K_1, \mathcal{C}$);
**11 end**
**12 else**
**13** | obj $\leftarrow$ HardCase($f, \mathcal{C}, \Lambda$);
**14 end**
**15** Apply $\Omega^{-1}$ to obj$\rightarrow f$, obj$\rightarrow M_0$, obj$\rightarrow M_1$;
**16 return** *obj;*

**Algorithm 6:** Reconstruction in low rank.

---

**Explanation**

Here we explain every step of the given algorithm:
- The function RECONSTRUCT($f$) takes as input a polynomial $f \in \Sigma\Pi\Sigma_{\mathbb{F}}(2)[\bar{x}]$ of $rank = r$ and outputs two polynomials $K_0, K_1 \in \Pi\Sigma_{\mathbb{F}}[\bar{x}]$ which are the two gates in it's circuit representation.

- Steps $2, 3$ picks a random matrix $\Omega$ and transforms each variable using the linear transformation this matrix defines. With high probability this will be an invertible transformation. We do the reconstruction for this new polynomial since the linear factors of it's gates satisfy some non-degenerate conditions(because they have been randomly transformed) our algorithm needs. We apply $\Omega^{-1}$ after the reconstruction and get back our original $f$.
- The next step constructs the set of candidate linear forms $\mathcal{C}$. We've talked about the size, construction and structure of this set in Section E.
- We first assume Medium Case. It that was not the case we check for Easy Case. If both did not occur we can be sure we are in the Hard case.
- We apply $\Omega^{-1}$ to polynomials in obj and return it.

## 4 Reconstruction for arbitrary $rank$

This section reduces the problem from $\Sigma\Pi\Sigma(2)$ Circuits with arbitrary rank $n$ $(> s)$ to one with constant rank $r$ (still $> s$). Also once the problem has been solved efficiently in the low rank case we use multiple instances of such solutions to lift to the general $\Sigma\Pi\Sigma(2)$ circuit. The idea is to project the polynomial to a small (polynomial) number of random subspaces of dimension $r$, reconstruct these low rank polynomials and then lift back to the original polynomial. The uniqueness of our circuit's representation plays a major role in both the projection and lifting steps. Let

$$f = G(\alpha_0 T_0 + \alpha_1 T_1)$$

$G, T_i$ are normal $\Pi\Sigma$ polynomials. All notations are borrowed from the previous section. It is almost identical to the restriction done in [24] except that the dimension of random subspaces is different. For more details see Section 4.2.1 and 4.2.3. in [24]. Since all proofs have been done in detail in [24] we do not spend much time here. A clear sketch with some proofs is however given.

### 4.1 Projection to a Random Low Dimensional Subspace

We explain the procedure of projecting to the random subspace below. In this low dimensional setup we can get coefficient representation of $\pi_V(f)$, also some important properties of $f$ are retained by $\pi_V(f)$. Proofs are simple and standard so we discuss them in the appendix at end.

Pick $n$ vectors $v_i, i \in [n]$ with each co-ordinate chosen independently from the uniform distribution on $[N]$. Let $V = sp(\{v_i : i \in [r]\})$ and $V' = sp\{v_i : i \in \{r+1, \ldots, n\}\}$. Then $V \oplus V' = \mathbb{F}^n$ Let $\pi_V$ denote the orthogonal projection onto $V$. With high probability the following hold:

1. $\{v_i : i \in [n]\}$ is linearly independent (see Appendix H.1 for proof).
2. Let $\{l_1, \ldots, l_r\}$ be a set of $r$ linearly independent linear forms in $\mathcal{L}(T_0) \cup \mathcal{L}(T_1)$. Then $\pi_V(\{l_1, \ldots, l_r\})$ is linearly independent with high probability. So $rank(\pi_V(f)) = r$ (see Appendix H.2 for Proof).
3. Let $l_{01} \in \mathcal{L}(T_0), l_{11} \in \mathcal{L}(T_1)$, then $\pi_V(l_{01}), \pi_V(l_{11})$ are linearly independent with high probability and so $gcd(\pi_V(T_0), \pi_V(T_1)) = 1$.

Pick large number of $(\geq d^r)$ random points $p_i, i = 1, \ldots, d^r$ in the space $V$. Use the values $\{f(p_i)\}$ and get a coefficient representation for $\pi_V(f)$. With high probability over the choice of points interpolation will work (See Appendix H.3 for Proof). We will effectively be solving a linear system. Note that the number of coefficients in $f|_V = O(d^r)$. Now this coefficient

representation of $\pi_V(f)$ is reconstructed using the algorithm in Section 3. A number of such reconstructions are then glued to reconstruct the original polynomial.

## 4.2 Lifting from the Random Low Dimensional Subspace

1. Consider spaces $V_i = V \oplus \mathbb{F}v_i$ for $i = r + 1, \ldots, n$.
2. Reconstruct $\pi_{V_i}(f)$ and $\pi_V(f)$ for each $i \in \{r + 1, \ldots, n\}$.
3. Let $l = \sum\limits_{i=1}^{n} a_i v_i$ be a linear form dividing one of the gates of $f$ say $T_0$. $\pi_V(l) = \sum\limits_{i=1}^{r} a_i v_i$
   and $\pi_{V_i}(l) = \sum\limits_{j=1}^{r} a_j v_j + a_i v_i$. Using our algorithm discussed in Section 3 we would have
   reconstructed $\pi_V(f)$ and $\pi_{V_i}(f)$. So we know the triples $(\pi_V(G), \pi_V(T_0), \pi_V(T_1))$ and
   $(\pi_{V_i}(G), \pi_{V_i}(T_0), \pi_{V_i}(T_1))$
   On restricting $V_i$ to $V$:
   a. *Only Factors become factors* with high probability so we can easily find the correspondence between $\pi_V(G)$ and $\pi_{V_i}(G)$.
   b. $\pi_V(\pi_{V_i}(T_0)) = \pi_V(T_0)$ and $\neq \pi_V(T_1)$ because of uniqueness of representation and therefore we get the correspondence between gates.
   c. Now to get correspondence between linear forms. Let $\pi_V(l)$ have multiplicity $k$ in $\pi_V(T_0)$. Then with high probability $l$ has multiplicity $k$ in $T_0$ Since two LI vectors remain LI on projecting to a random subspace of dimension $\geq 2$ (again see Appendix H.2 for proof). Therefore $\pi_{V_i}(l)$ has multiplicity $k$ and is the unique lift of $\pi_V(l)$ for all $i$. Let $\pi_{V_i}(l) = \pi_V(l) + a_i v_i$. Then $l = \pi_V(l) + \sum_{i=r+1}^{n} a_i v_j$. This finds $G, T_0, T_1$ for us

## 4.3 Time Complexity

- Interpolation to find coefficient representation $\pi_V(f)$ which is a degree $d$ polynomial over $r$ variables clearly takes $poly(d^r)$ time (accounts to solving a linear system of size $poly(d^r)$).
- Solving $n - r$ instances of the low rank problem (simple ranks $r$ and $r + 1$) takes $npoly(d^r)$ time.
- The above mentioned approach to glue the linear forms in the gates clearly takes $poly(n, d)$ time.
- Overall the algorithm takes $poly(n, d)$ time since $r$ is a constant.

## 5 Conclusion and Future Work

We described an efficient randomized algorithm to reconstruct circuit representation of multivariate polynomials which exhibit a $\Sigma\Pi\Sigma(2)$ representation. Our algorithm works for all polynomials with rank(number of independent variables greater than a constant $r$). In future we would like to address the following:

- **Reconstruction for Lower Ranks:** As can be seen in the paper, rank of the polynomial for uniqueness (i.e. $c_\mathbb{F}(4)$) and the rank we've assumed in the low rank reconstruction (i.e. $r$) are both $O(1)$ but $c_\mathbb{F}(4)$ is smaller than $r$. Since one would expect a reconstruction algorithm whenever the circuit is unique we would like to close this gap.
- $\Sigma\Pi\Sigma(k)$ **circuits:** The obvious next step would be to consider more general top fan-in. In particular we could consider $\Sigma\Pi\Sigma(k)$ circuits with $k = O(1)$.
- **Derandomization:** We would like to derandomize the algorithm as it was done in the finite field case in [15].

## References

1   Manindra Agrawal. Proving lower bounds via pseudo-random generators. In *FSTTCS 2005: Foundations of Software Technology and Theoretical Computer Science, 25th International Conference, Hyderabad, India, December 15-18, 2005, Proceedings, volume 3821 of Lecture*, pages 92–105. Springer, 2005.

2   V. Arvind, Partha Mukhopadhyay, and Srikanth Srinivasan. New results on noncommutative and commutative polynomial identity testing. *2012 IEEE 27th Conference on Computational Complexity*, 0:268–279, 2008. `doi:10.1109/CCC.2008.22`.

3   B. Barak, Z. Dvir, A. Wigderson, and A. Yehudayoff. Rank bounds for design matrices with applications to combinatorial geometry and locally correctable codes. In *Proceedings of the 43rd annual ACM symposium on Theory of computing*, STOC'11, pages 519–528, New York, NY, USA, 2011. ACM.

4   Amos Beimel, Francesco Bergadano, Nader H. Bshouty, Eyal Kushilevitz, and Stefano Varricchio. Learning functions represented as multiplicity automata. *J. ACM*, 47(3):506–530, May 2000. `doi:10.1145/337244.337257`.

5   B. Buchberger. A theoretical basis for the reduction of polynomials to canonical forms. *SIGSAM Bull.*, 10(3):19–29, August 1976. `doi:10.1145/1088216.1088219`.

6   Z. Dvir, S. Saraf, and A. Wigderson. Improved rank bounds for design matrices and a new proof of Kelly's theorem. Forum of mathematics – Sigma (to appear), 2012.

7   Zeev Dvir and Amir Shpilka. Locally decodable codes with 2 queries and polynomial identity testing for depth 3 circuits. *SIAM J. COMPUT*, 36(5):1404–1434, 2007.

8   Izrail Moiseevitch Gelfand, Mikhail M. Kapranov, and Andrei V. Zelevinsky. *Discriminants, resultants, and multidimensional determinants*. Mathematics: theory & applications. Birkhäuser, Boston, Basel, Berlin, 1994. Autre tirage de l'édition Birkhäuser chez Springer Science+ Business Media.

9   Oded Goldreich, Shafi Goldwasser, and Silvio Micali. How to construct random functions. *J. ACM*, 33(4):792–807, August 1986. `doi:10.1145/6490.6503`.

10  Ankit Gupta, Neeraj Kayal, and Satya Lokam. Reconstruction of depth-4 multilinear circuits with top fan-in 2. In *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*, STOC'12, pages 625–642, New York, NY, USA, 2012. ACM. `doi:10.1145/2213977.2214035`.

11  Ankit Gupta, Neeraj Kayal, and Satyanarayana V. Lokam. Efficient reconstruction of random multilinear formulas. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS 2011, Palm Springs, CA, USA, October 22-25, 2011*, pages 778–787, 2011. `doi:10.1109/FOCS.2011.70`.

**12** Ankit Gupta, Neeraj Kayal, and Youming Qiao. Random arithmetic formulas can be reconstructed efficiently. *computational complexity*, 23(2):207–303, 2014. `doi:10.1007/s00037-014-0085-0`.

**13** J. Heintz and C. P. Schnorr. Testing polynomials which are easy to compute (extended abstract). In *Proceedings of the Twelfth Annual ACM Symposium on Theory of Computing*, STOC'80, pages 262–272, New York, NY, USA, 1980. ACM. `doi:10.1145/800141.804674`.

**14** Erich Kaltofen and Barry M. Trager. Computing with polynomials given byblack boxes for their evaluations: Greatest common divisors, factorization, separation of numerators and denominators. *J. Symb. Comput.*, 9(3):301–320, March 1990. `doi:10.1016/S0747-7171(08)80015-6`.

**15** Zohar S. Karnin and Amir Shpilka. Reconstruction of generalized depth-3 arithmetic circuits with bounded top fan-in. In *Proceedings of the 24rd Annual CCC*, pages 274–285, 2009.

**16** Neeraj Kayal and Shubhangi Saraf. Blackbox polynomial identity testing for depth 3 circuits. In *Proceedings of the 2009 50th Annual IEEE Symposium on Foundations of Computer Science*, FOCS'09, pages 198–207, Washington, DC, USA, 2009. IEEE Computer Society. `doi:10.1109/FOCS.2009.67`.

**17** Michael Kearns and Leslie Valiant. Cryptographic limitations on learning boolean formulae and finite automata. *J. ACM*, 41(1):67–95, January 1994. `doi:10.1145/174644.174647`.

**18** Michael Kharitonov. Cryptographic lower bounds for learnability of boolean functions on the uniform distribution. In *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*, COLT'92, pages 29–36, New York, NY, USA, 1992. ACM. `doi:10.1145/130385.130388`.

**19** Adam Klivans and Amir Shpilka. Learning restricted models of arithmetic circuits. *Theory of computing*, 2(10):185–206, 2006.

**20** Adam R. Klivans and Daniel Spielman. Randomness efficient identity testing of multivariate polynomials. In *Proceedings of the Thirty-third Annual ACM Symposium on Theory of Computing*, STOC'01, pages 216–223, New York, NY, USA, 2001. ACM. `doi:10.1145/380752.380801`.

**21** Nitin Saxena. Progress on polynomial identity testing. *Bulletin of the EATCS*, 2009(99):49–79, 2009.

**22** Nitin Saxena and C. Seshadhri. From sylvester-gallai configurations to rank bounds: Improved black-box identity test for depth-3 circuits. *CoRR*, abs/1002.0145, 2010. URL: `http://arxiv.org/abs/1002.0145`.

**23** Robert E. Schapire and Linda M. Sellie. Learning sparse multivariate polynomials over a field with queries and counterexamples. In *Proceedings of the Sixth Annual ACM Workshop on Computational Learning Theory*, pages 17–26, 1996.

**24** Amir Shpilka. Interpolation of depth-3 arithmetic circuits with two multiplication gates. In *STOC'07: Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 284–293. ACM Press, 2007.

**25** Amir Shpilka and Ilya Volkovich. Improved polynomial identity testing for read-once formulas. In *APPROX-RANDOM*, pages 700–713, 2009.

## A Characterizing $\Pi\Sigma$ polynomials (Brill's Equations)

In this section we will explicitly compute a set of polynomials whose common solutions characterize the coefficients of all homogeneous $\Pi\Sigma_{\mathbb{C}}[x_1, \ldots, x_r]$ polynomials of degree $d$. A clean mathematical construction is given by Brill's Equations given in Chapter 4, [8]. However we still need to calculate the time complexity. But before that we define some operations on polynomials and calculate the time taken by the operation along with the size

of the output. Note that all polynomials are over the field of complex numbers $\mathbb{C}$ and all computations are also done for the complex polynomial rings.

Let $\bar{x} = (x_1, \ldots, x_r)$ and $\bar{y} = (y_1, \ldots, y_r)$ be variables. For any homogeneous polynomial $f(\bar{x})$ of degree $d$, define

$$f_{\bar{x}^k}(\bar{x}, \bar{y}) = \frac{(d-k)!}{d!} (\sum_i x_i \frac{\partial}{\partial y_i})^k f(\bar{y})$$

Expanding $(\sum_i x_i \frac{\partial}{\partial y_i})^k$ as a polynomial of differentials takes $O((r+k)^r)$ time and has the same order of terms in it. $f(\bar{y})$ has $O((r+k)^r)$ terms. Taking partial derivatives of each term takes constant time and therefore overall computing $(\sum_i x_i \frac{\partial}{\partial y_i})^k f(\bar{y})$ takes $O((r+k)^{2r})$ time. Also the expression obtained will have atmost $O((r+k)^{2r})$ terms. Computing the external factor takes $poly(d)$ time and so for an arbitrary $f(\bar{x})$ computing all $f_{\bar{x}^k}(\bar{x}, \bar{y})$ for $0 \leq k \leq d$ takes $poly((r+d)^r)$ time and has $poly((r+d)^r)$ terms in it. From Section E., Chapter 4 in [8] we also know that $f_{\bar{x}^k}(\bar{x}, \bar{y})$ is a bihomogeneous form of degree $k$ in $\bar{x}$ and degree $d-k$ in $\bar{y}$. It is called the $k^{th}$ polar of $f$.

Next we define an $\odot$ opeartion between homogeneous forms. Let $f(\bar{x})$ and $g(\bar{x})$ be homogeneous polynomials of degrees $d$, define

$$(f \odot g)(\bar{x}, \bar{y}) = \frac{1}{d+1} \sum_{k=0}^{d} (-1)^k \binom{d}{k} f_{\bar{y}^k}(\bar{y}, \bar{x}) g_{\bar{x}^k}(\bar{x}, \bar{y})$$

From the discussion above we know that computing $f_{\bar{y}^k}(\bar{y}, \bar{x}) g_{\bar{x}^k}(\bar{x}, \bar{y})$ takes $poly((r+d)^r)$ time and it is obvious that this product has $poly((r+d)^r)$ terms. Rest of the operations take $poly(d)$ time and therefore computing $(f \odot g)(\bar{x}, \bar{y})$ takes $poly((r+d)^r)$ time and has $poly((r+d)^r)$ terms. From the discussion before we may also easily conclude that the degrees of $\bar{x}, \bar{y}$ in $(f \odot g)(\bar{x}, \bar{y})$ are $poly(d)$. The form $(f \odot g)$ is called the vertical(Young) product of $f$ and $g$. See Section G., Chapter 4 in [8].

Next for $k \in \{0, \ldots, d\}$ and $\bar{z} = (z_1, \ldots, z_r)$ consider homogeneous forms:

$$e_k = \binom{d}{k} f_{\bar{x}^k}(\bar{x}, \bar{z}) f(\bar{z})^{k-1}$$

Following arguments from above, it's straightforward to see that computing $e_k$ takes $poly((r+d)^r)$ time and has $poly((r+d)^r)$ terms. Each $e_k$ is a homogeneous form in $\bar{x}, \bar{z}$ and $f$. It has degree $k$ in $\bar{x}$, degree $k(d-1)$ in $z$, and $k$ in coefficients of $f$. See Section H. of Chapter 4 in [8]. Let's define the following function of $\bar{x}$ with parameters $f, z$

$$P_{f,z}(\bar{x}) = (-1)^d d \sum_{i_1 + 2i_2 + \ldots + ri_r = d} (-1)^{(i_1 + \ldots + i_r)} \frac{(i_1 + \ldots + i_r - 1)!}{i_1! \ldots i_r!} e_1^{i_1} \ldots e_r^{i_r}$$

Note that $\{(i_1, \ldots, i_r) : i_1 + 2i_2 + \ldots + ri_r = d\} \subseteq \{(i_1, \ldots, i_r) : i_1 + i_2 + \ldots + i_r \leq d\}$ and therefore the number of additions in the above summand is $O(poly(r+d)^r)$. For every fixed $(i_1, \ldots, i_r)$ computing the coefficient $\frac{(i_1 + \ldots + i_r - 1)!}{i_1! \ldots i_r!}$ takes $O(poly((r+d)^r))$ time using multinomial coefficients. Each $e_k$ takes $poly((r+d)^r)$ time to compute. There are $r$ of them in each summand and so overall we take $O(poly((r+d)^r))$ time. A similar argument shows that number of terms in this polynomial is $O(poly((r+d)^r))$. Some more analysis shows

that $P_{f,z}(\bar{x})$ is a form of degree $d$ in $\bar{x}$ whose coefficients are homogeneous polynomials of dedgree $d$ in $f$ and degree $d(d-1)$ in $\bar{z}$. Let

$$B_f(\bar{x}, \bar{y}, \bar{z}) = (f \odot P_{f,z})(\bar{x}, \bar{y})$$

By the arguments given above calculating this form also takes time $poly((r+d)^r)$ and it has $poly((r+d)^r)$ terms. This is a homogeneous form in $(\bar{x}, \bar{y}, \bar{z})$ of multidegree $(d, d, d(d-1))$ and it's coefficients are forms of degree $(d+1)$ in the coefficients of $f$. See Section H., Chapter 4 in [8]. So in time $poly((r+d)^r)$ we can compute $B_f(\bar{x}, \bar{y}, \bar{z})$ explicitly.

Now we arrive at the main theorem

▶ **Theorem A.1** (Brill's Equation – see 4.H, [8]). *A form $f(\bar{x})$ is a product of linear forms if and only if the polynomial $B_f(\bar{x}, \bar{y}, \bar{z})$ is identically 0.*

We argued above that computing $B_f(\bar{x}, \bar{y}, \bar{z})$ takes $O(poly((r+d)^r))$ time. It's degrees in $\bar{x}, \bar{y}, \bar{z}$ are all $poly(d)$ and so the number of coefficients when written as a polynomial over the $3r$ variables

$(x_1, \ldots, x_r, y_1, \ldots, y_r, z, \ldots, z_r)$ is $poly((r+d)^r)$. We mentioned that each coefficient is a polynomial of degree $(d+1)$ in the coefficients of $f$. Therefore we have the following corollary.

▶ **Corollary A.2.** *Let*

$$I \stackrel{def}{=} \{(\alpha_1, \ldots, \alpha_n) : \forall i : \alpha_i \geq 0, \sum_{i \in [r]} \alpha_i = d\}$$

*be the set capturing the indices of all possible monomials of degree exactly $d$ in $r$ variables $(x_1, \ldots, x_r)$. Let $f_{\mathbf{a}}(y_1, \ldots, y_r) = \sum_{\alpha \in I} a_\alpha \mathbf{y}^\alpha$ denote an arbitrary homogeneous polynomial. The coefficient vector then becomes $\mathbf{a} = (a_\alpha)_{\alpha \in I}$. Then there exists an explicit set of polynomials $F_1(\mathbf{a}), \ldots, F_m(\mathbf{a})$ on $poly((r+d)^r)$ variables $(\mathbf{a} = (a_\alpha)_{\alpha \in I})$, with $m = poly((r+d)^r)$, $deg(F_i) \leq poly(d)$ such that for any particular value of $\mathbf{a}$, the corresponding polynomial $f_{\mathbf{a}}(\mathbf{y}) \in \Pi\Sigma_{\mathbb{F}}^d[\bar{y}]$ if and only if $F_1(\mathbf{a}) = \ldots = F_m(\mathbf{a}) = 0$. Also this set $\{F_i, i \in [m]\}$ can be computed in time $poly((r+d)^r)$ time.*

**Proof.** Clear from the theorem and discussion above. ◀

Note that in our application $r = O(1)$ and so $poly((d+r)^r) = poly(d)$.

## B    Tools from Incidence Geometry

Later in the paper we will use the quantitative version of Sylvester-Gallai Theorem from [3]. In this subsection we do preparation for the same. Our main application will also involve a corollary we prove towards the end of this subsection.

▶ **Definition B.1** ([3]). *Let $S$ be a set of $n$ distinct points in complex space $\mathbb{C}^r$. A $k - flat$ is elementary if its intersection with $S$ has exactly $k+1$ points.*

▶ **Definition B.2** ([3]). *Let $S$ be a set of $n$ distinct points in $\mathbb{C}^r$. $S$ is called a $\delta - SG_k$ configuration if for every independent $s_1, \ldots, s_k \in S$ there are atleast $\delta n$ points $t \in S$ such that either $t \in fl(\{s_1, \ldots, s_k\})$ or the $k-$flat $fl(\{s_1, \ldots, s_k, t\})$ contains a point in $S \setminus \{s_1, \ldots, s_k, t\}$.*

▶ **Theorem B.3** ([3]). *Let $S$ be a $\delta - SG_k$ configuration then $dim(S) \leq \frac{2^{C^k}}{\delta^2}$. Where $C > 1$ is a universal constant.*

This bound on the dimension of $S$ was further improved by Dvir et al. in [6]. The latest version now states

▶ **Theorem B.4** ([6]). *Let $S$ be a $\delta - SG_k$ configuration then $dim(S) \leq C_k = \frac{C^k}{\delta}$. Where $C > 1$ is a universal constant.*

▶ **Corollary B.5.** *Let $dim(S) > C_k$ then $S$ is not a $\delta - SG_k$ configuration i.e. there exists a set of independent points $\{s_1, \ldots, s_k\}$ and $\geq (1 - \delta)n$ points $t$ such that $fl(\{s_1, \ldots, s_k, t\})$ is an elementary $k - flat$. That is:*
- $t \notin fl(\{s_1, \ldots, s_k\})$
- $fl(\{s_1, \ldots, s_k, t\}) \cap S = \{s_1, \ldots, s_k, t\}$.

Right now we set $\delta$ to be a constant $< 0.5, C_k = \frac{C^k}{\delta}$. Note that $C_i < C_{i+1}$. Using the above theorem we prove the following lemma which will be useful to us later

▶ **Lemma B.6** (Bichromatic semi-ordinary line). *Let $X$ and $Y$ be disjoint finite sets in $\mathbb{C}^r$ satisfying the following conditions.*
1. $dim(Y) > C_4$.
2. $|Y| \leq c|X|$ with $c < \frac{1-\delta}{\delta}$.
*Then there exists a line $l$ such that $|l \cap Y| = 1$ and $|l \cap X| \geq 1$*

**Proof.** We consider two cases:

**Case 1:** $c|X| \geq |Y| \geq |X|$ Since $dim(Y) > C_1$, using the corollary above for $S = X \cup Y, k = 1$ we can get a point $s_1 \in X \cup Y$ for which there exist $(1 - \delta)(|X| + |Y|)$ points $t$ in $X \cup Y$ such that $t \notin fl\{s_1\}$ and $fl\{s_1, t\}$ is elementary. If $s_1 \in X$ then $(1 - \delta)(|X| + |Y|) - |X| \geq (1 - 2\delta)|X| > 0$ of these flats intersect $Y$ and thus we get such a line $l$. If $s_1 \in Y$ then $(1 - \delta)(|X| + |Y|) - |Y| \geq ((1 - \delta)(\frac{1}{c} + 1) - 1)|Y| > 0$ of these flats intersect $X$ giving us the required line $l$ with $|l \cap X| = 1$ and $|l \cap Y| = 1$.

**Case 2:** $|Y| \leq |X|$ Now choose a subset $X_1 \subseteq X$ such that $|X_1| = |Y|$. Now using the same argument as above for $S = X_1 \cup Y$ there is a point $s_1 \in X_1 \cup Y$ such that $(1 - \delta)(|X_1| + |Y|) = 2(1 - \delta)|Y| = 2(1 - \delta)|X_1|$ flats through it are elementary in $X_1 \cup Y$. If $s_1 \in Y$ $(1 - 2\delta)|Y| > 0$ of these flats intersect $X_1$. If $s_1 \in X_1$, $(1 - 2\delta)|X_1| > 0$ of these flats intersect $Y$. In both these above possibilities the flat intersects $Y$ and $X_1$ in exactly one point each. But it may contain more points from $X \setminus X_1$ so we can find a line $l$ such that $|l \cap Y| = 1$ and $|l \cap X| \geq 1$. ◀

## C A Method of Reconstructing Linear Forms

In a lot of circumstances one might reconstruct a linear form (upto scalar multiplication) inside $V = Lin_{\mathbb{F}}[\bar{x}]$ from it's projections (upto scalar multiplication) onto some subspaces of $V$. For example consider a linear form $L = a_1 x_1 + a_2 x_2 + a_3 x_3 (\in Lin_{\mathbb{F}}[x_1, x_2, x_3])$ with $a_3 \neq 0$, and assume we know scalar multiples of projections of $L$ onto the spaces $\mathbb{F}x_1$ and $\mathbb{F}x_2$ i.e. we know $L_1 = \alpha(a_2 x_2 + a_3 x_3)$ and $L_2 = \beta(a_1 x_1 + a_3 x_3)$ for some $\alpha, \beta \in \mathbb{F}$. Scale these projections to $\tilde{L}_1 = x_3 + \frac{a_2}{a_3} x_3$ and $\tilde{L}_2 = x_3 + \frac{a_1}{a_3} x_3$. Using these two define a linear form $x_3 + \frac{a_1}{a_3} x_1 + \frac{a_2}{a_3} x_2$. This is a scalar multiple of our original linear form $L$. We generalize this a little more below.

Let $\bar{x} \equiv (x_1, \ldots, x_r)$, $\mathcal{B} = \{l_1, \ldots, l_r\}$ be a basis for $V = Lin_{\mathbb{F}}[x_1, \ldots, x_r]$. For $i \in \{0, 1, 2\}$, let $S_i$ be pairwise disjoint non empty subsets of $\mathcal{B}$ such that $S_0 \cup S_1 \cup S_2 = \mathcal{B}$. Let $W_i = sp(S_i)$ and $W_i' = \bigoplus_{j \neq i} W_j$. Clearly $V = W_0 \oplus W_1 \oplus W_2 = W_i \oplus W_i', i \in \{0, 1, 2\}$.

▶ **Lemma C.1.** *Assume $L \in V$ is a linear form such that*
- $\pi_{W_2}(L) \neq 0$
- *For $i \in \{0,1\}, L_i = \beta_i \pi_{W_i'}(L)$ are known for some non-zero scalars $\beta_i$.*

*Then $L$ is unique upto scalar multiplication and we can construct a scalar multiple $\tilde{L}$ of $L$.*

**Proof.** Let $L = a_1 l_1 + \ldots + a_r l_r, a_i \in \mathbb{F}$. Since $\pi_{W_2}(L) \neq 0$, there exists $l_j \in S_2$ such that $a_j \neq 0$. Let $\tilde{L} = \frac{1}{a_j} L$. For $i \in \{0,1\}$, re-scale $L_i$ to get $\tilde{L}_i$ making sure that coefficient of $l_j$ is 1 in them. Thus for $i = 0, 1$

$$\pi_{W_i'}(\tilde{L}) = \tilde{L}_i \,.$$

Since $W_0' = W_1 \oplus W_2$ and $W_1' = W_0 \oplus W_2$ by comparing coefficients we can get $\tilde{L}$. ◀

**Algorithm** Assume we know $S_0, S_1, S_2$ and therefore the basis change matrix to convert vector representations from $\mathcal{S}$ to $\mathcal{B}$. It takes $poly(r)$ time to convert $[v]_{\mathcal{S}}$ to $[v]_{\mathcal{B}}$. Given $L_i$ in the basis $\mathcal{B}$ it takes $poly(r)$ time(by a linear scan) to find $l_j \in S_2$ with $a_j \neq 0$. This $l_j$ has a non zero coefficient in both $L_0, L_1$. After this we just rescale $L_i$ to get $\tilde{L}_i$ such that coefficient of $l_j$ is 1. Then since $\tilde{L}_i = \pi_{W_i'}(\tilde{L})$ the coefficient of $l_t$ in $\tilde{L}$ is as follows:

$$= \begin{cases} \text{coefficient of } l_t \text{ in } \tilde{L}_1 & : l_t \in S_0 \\ \text{coefficient of } l_t \text{ in } \tilde{L}_0 & : l_t \in S_1 \\ \text{coefficient of } l_t \text{ in } \tilde{L}_0 = \text{ coefficient of } l_t \text{ in } \tilde{L}_1 & : l_t \in S_2 \end{cases}$$

Finding the right coefficients using this also takes $poly(r)$ time.

Next we try and use this to reconstruct $\Pi\Sigma$ polynomials. This case is slightly more complicated and so we demand that the projections have some special form. In particular the projections onto one subspace preserves pairwise linear independence of linear factors and onto the other makes all linear factors scalar multiples of each other.

▶ **Corollary C.2.** *Let $S_i, W_i, i \in \{0,1,2\}$ be as above and $P \in \Pi\Sigma_{\mathbb{F}}[x_1, \ldots, x_r]$ such that*
1. $\pi_{W_2}(P) \neq 0$
2. *For $i \in \{0,1\}$ there exists $\beta_i(\neq 0) \in \mathbb{F}$ such that $P_0 = \beta_0 \pi_{W_0'}(P) = p^t$ and $P_1 = \beta_1 \pi_{W_1'}(P) = d_1 \ldots d_t$. are known i.e. $p, d_j \ (j \in [t])$ and $t$ are known.*

*Then $P$ is unique upto scalar multiplication and we can construct a scalar multiple $\tilde{P}$ of $P$.*

**Proof.** Let $P = L_1 \ldots L_t$ with $L_i \in Lin_{\mathbb{F}}[\bar{x}]$. There exists $\beta_i^j, i \in \{0,1\}, j \in [t]$, such that $\beta_0^j \pi_{W_0'}(L_j) = p$ and $\beta_1^j \pi_{W_1'}(L_j) = d_j$. Since $p, d_j$ are known by above Lemma C.1 we find a scalar multiple $\tilde{L}_j = \beta^j L_j$ of $L_j$ and therefore find a scalar multiple $\tilde{P} = \tilde{L}_1 \ldots \tilde{L}_t$ of $P$. Note that this method also tells us that such a $P$ is unique upto scalar multiplication. Since we've used the above Algorithm C at most $t$ times with $t \leq deg(P)$, it takes $poly(deg(P), r)$ time to find $\tilde{P}$. ◀

This corollary is the backbone for reconstructing $\Pi\Sigma$ polynomials from their projections. But first we formally define a "*Reconstructor*":

▶ **Definition C.3** (Reconstructor). Let $S_i, W_i, i \in \{0,1,2\}$ be as above. Let $Q$ be a standard $\Pi\Sigma$ polynomial and $P$ be a standard $\Pi\Sigma$ polynomial dividing $Q$ with $Q = PR$. Then $(Q, P, S_0, S_1, S_2)$ is called a *Reconstructor* if:
- $\pi_{W_2}(P) \neq 0$.
- $\pi_{W_0'}(P) = \alpha p^t$, for some linear form $p$.
- Let $l \mid R$ be a linear form and $\pi_{W_2}(l) \neq 0$ then $gcd(\pi_{W_2}(P), \pi_{W_2}(l)) = 1$.

**Note:** Let $L_1, L_2$ be two LI linear forms dividing $P$, then one can show

$$L_1, L_2 \text{ are LI} \iff \pi_{W_1'}(L_1), \pi_{W_1'}(L_2) \text{ are LI}$$

To see this first observe that the second bullet implies for $i \in [2]$, $L_i \in W_0 + p \Rightarrow sp(\{L_1, L_2\}) \subseteq W_0 + p$.

If $\pi_{W_1'}(L_1), \pi_{W_1'}(L_2)$ are LD then

$$sp(\{L_1, L_2\}) \cap W_1 \neq \{0\}$$

$\Rightarrow (W_0 + p) \cap W_1 \neq \{0\}$. Since $W_0 \cap W_1 = \{0\}$ we get that $p \in W_0 \oplus W_1 = W_2' \Rightarrow \pi_{W_2}(p) = 0 \Rightarrow \pi_{W_2}(P) = 0$ contradicting the first bullet.

Geometrically the conditions just mean that all linear forms dividing $P$ have LD projection ($= \gamma p$ for some non zero $\gamma \in \mathbb{F}$) w.r.t. the subspace $W_0'$ and LI linear forms $p_1, p_2$ dividing $P$ have LI projections (w.r.t. subspace $W_1'$). Also no linear form $l$ dividing $R$ belongs to $fl(S_0 \cup S_1 \cup \{p\})$.

We are now ready to give an algorithm to reconstruct $P$ using $\pi_{W_0'}(Q)$ and $\pi_{W_1'}(Q)$ by gluing appropriate projections corresponding to $P$. To be precise:

▶ **Claim C.4.** *Let $Q, P$ be standard $\Pi\Sigma$ polynomials and $P \mid Q$. Assume $(Q, P, S_0, S_1, S_2)$ is a* Reconstructor. *If we know both $\pi_{W_0'}(Q)$ and $\pi_{W_1'}(Q)$. Then we can reconstruct $P$.*

**Proof.** See Algorithm 7. ◀


## C.1   Explanation

- The algorithm takes as input projections $\pi_{W_0'}(Q)$ and $\pi_{W_1'}(Q)$ along with the sets $S_i, i = 0, 1, 2$ which form a partition of a basis $\mathcal{B}$. We know that there exists a polynomial $P \mid Q$ such that $(Q, P, S_0, S_1, S_2)$ is a reconstructor and so we try to compute the projections $\pi_{W_0'}(P), \pi_{W_1'}(P)$.
- If one assumes that $\pi_{W_0'}(Q) = \gamma \prod_{i \in [s]} c_i^{m_i}$ with the $c_i$'s co-prime, then by the properties of a reconstructor the projection (of a scalar multiple of $P$) onto $W_0'$ say $P_0 = \beta_0 \pi_{W_0'}(P)$ (for some $\beta_0$) has to be equal to $c_i^{m_i}$ for some $i$. We do this assignment inside the first for loop.
- The third property of a reconstructor implies that when we project further to $W_2$, it should not get any more factors and so we check this inside the second for loop by going over all other factors $c_j$ of $\pi_{W_0'}(Q)$ and checking if $c_i, c_j$ become LD on projecting to $W_2$ (i.e. by further projecting to $W_1'$).
- Now to find (scalar multiple of) the other projections i.e. $P_1 = \beta_1 \pi_{W_1'}(P)$ (for some $\beta_1$), we go through $\pi_{W_1'}(Q)$ and find $d_k$ such that $\{\pi_{W_1'}(c_i), \pi_{W_0'}(d_k)\}$ are LD (i.e. they are projections of the same linear form). We collect the product of all such $d_k$'s. If the choice of $c_i$ were correct then all $d_k$'s would be obtained correctly.
- The last "*if*" statement just checks that the number of $d_k$'s found above is the same as $m_i$ since $P_0 = c_i^{m_i}$ tells us that the degree of $P$ was $m_i$. We recover a scalar multiple of $P$ using the algorithm explained in Corollary C.2 and then make it standard to get $P$.


## C.2   Correctness

The corectness of our algorithm is shown by the lemma below.

---

**input** : $\pi_{W_0'}(Q) \in \Pi\Sigma[\bar{x}], \pi_{W_1'}(Q) \in \Pi\Sigma[\bar{x}], S_0, S_1, S_2$
**output:** a $\Pi\Sigma$ polynomial $P \mid Q$

**1** $bool\,flag$, $\Pi\Sigma$ polynomial $P_0, P_1$;;
**2** Factor $\pi_{W_0'}(Q) = \gamma \prod\limits_{i \in [s]} c_i^{m_i}$, $c_i$'s pairwise LI and normal, $\gamma \in \mathbb{F}$;
**3** Factor $\pi_{W_1'}(Q) = \delta d_1 \dots d_m$, $\delta \in \mathbb{F}$ and $d_j$ normal;
**4 for** $i \in [s]$ *&& $\pi_{W_1'}(c_i) \neq 0$* **do**
**5** $\quad$ $flag = true, P_0 = c_i^{m_i}$;
**6** $\quad$ // Assuming projection w.r.t. $W_0'$ to be $c_i^{m_i}$.
**7** $\quad$ **for** $j \in [s]$ *&& $j \neq i$ && $\pi_{W_1'}(c_j) \neq 0$* **do**
**8** $\quad\quad$ **if** $gcd(\pi_{W_1'}(c_i), \pi_{W_1'}(c_j)) \neq 1$ **then**
**9** $\quad\quad\quad$ $flag = false$;
**10** $\quad\quad$ **end**
**11** $\quad$ **end**
**12** $\quad$ **if** $flag == true$ **then**
**13** $\quad\quad$ $P_1 = 1$;
**14** $\quad$ **end**
**15** $\quad$ **for** $j \in [m]$ **do**
**16** $\quad\quad$ **if** $\pi_{W_0'}(d_j) \neq 0$ *& & $\{\pi_{W_0'}(d_j), \pi_{W_1'}(c_i)\}$ are LD* **then**
**17** $\quad\quad\quad$ $P_1 = P_1 d_j$ ;
**18** $\quad\quad\quad$ // This steps collects projection w.r.t. $W_1'$ in $P_1$.
**19** $\quad\quad$ **end**
**20** $\quad$ **end**
**21** $\quad$ **if** $(deg(P_1) = m_i)$ *&& $(P_0, P_1)$ give $\tilde{P} = \beta P$ using Corollary C.2* **then**
**22** $\quad\quad$ Make $\tilde{P}$ *standard* w.r.t. the standard basis $\mathcal{S}$ to get $P$;
**23** $\quad\quad$ Return $P$;
**24** $\quad$ **end**
**25 end**
**26** Return 1;

**Algorithm 7:** Reconstructing Linear Factors.

▶ **Claim C.5.** *If $(Q, P, S_0, S_1, S_2)$ is a reconstructor for non-constant $P$, then Algorithm refreconalgo returns $P$.*

**Proof.** $(Q, P, S_0, S_1, S_2)$ is a reconstructor therefore

- $\pi_{W_2}(P) \neq 0$
- $\pi_{W_0'}(P) = \delta p^t$
- $q \mid \frac{Q}{P} \Rightarrow gcd(\pi_{W_2}(q), \pi_{W_2}(P)) = 1$

1. It is clear that for one and only one value of $i$, $c_i$ divides $p$. Fix this $i$. Let $Q = PR$, if $c_i^{m_i} \nmid \pi_{W_0'}(P)$ then $c_i \mid l$ for some linear form $l \mid \pi_{W_0'}(R)$. Condition 3 in definition of Reconstructor implies that $gcd(\pi_{W_2}(P), \pi_{W_2}(l)) = 1$ but $\pi_{W_2}(c_i)$ divides both of them giving us a contradiction. Since $\pi_{W_0'}(P)$ has just one linear factor $\Rightarrow \pi_{W_0'}(P)$ is a scalar multiple of $c_i^{m_i}$ for some $i$.

2. Assume the correct $c_i^{m_i}$ has been found. Now let $d_j \mid \pi_{W_1'}(Q)$ such that $\{\pi_{W_2}(c_i), \pi_{W_2}(d_j)\}$ are LD. then we can show that $d_j \mid \pi_{W_1'}(P)$. Assume not, then for some linear form $l \mid R = \frac{Q}{P}$, $d_j \mid \pi_{W_1'}(l)$. $\pi_{W_0'}(d_j) \neq 0$ (which we checked) $\Rightarrow \pi_{W_2}(l) \neq 0$. So we get

$\pi_{W_2}(c_i) \mid \pi_{W_2}(l)(\neq 0)$ and so $\pi_{W_2}(c_i) \mid gcd(\pi_{W_2}(P), \pi_{W_2}(l))$ which is therefore $\neq 1$ and condition 3 of Definiton C.3 is violated. So whatever $d_j$ we collect will be a factor of $\pi_{W_1'}(P)$ and we will collect all of them since they are all present in $\pi_{W_1'}(Q)$.

3. We know from proof of Corollary C.2 that if we know $c_i, m_i$ and $d_j$'s correctly then we can recover a scalar multiple of $P$ correctly. But $Q, P$ are standard so we return $P$ correctly.                                                                                                     ◄

In fact we can show that if we return something it has to be a factor of $Q$.

▶ **Claim C.6.** *If Algorithm 7 returns a* $\Pi\Sigma$ *polynomial $P$, then $P \mid Q$.*

- If the algorithm returns 1 from the last return statement, we are done. So let's assume it returns something from the previous return statement.
- So *flag* has to be true at end $\Rightarrow$ there is an $i \in [s]$ such that $P_0 = c_i^{m_i}$ with the conditions that $\pi_{W_1'}(c_i) \neq 0$ and $gcd(c_i, c_j) = 1$ for $j \neq i$. It also means that for exactly $m_i$ of the $d_j$'s (say $d_1, \ldots, d_{m_i}$) $\{\pi_{W_1'}(c_i), \pi_{W_0'}(d_j)\}$ are LD and $P_1 = d_1 \ldots d_{m_i}$.
- Since $c_i^{m_i} \mid \pi_{W_0'}(Q)$, there exists a factor $\tilde{P} \mid Q$ of degree $m_i$ such that $\pi_{W_0'}(\tilde{P}) = c_i^{m_i}$ and $\pi_{W_1'}(c_i) \neq 0$. This $\Rightarrow \pi_{W_2}(\tilde{P}) \neq 0$. Clearly $\pi_{W_1'}(\tilde{P}) \mid \pi_{W_1'}(Q) = d_1 \ldots d_m$, hence for all linear factors $\tilde{p}$ of $\tilde{P}$, $\pi_{W_1'}(\tilde{p})$ should be some $d_j$ with the condition that $\{\pi_{W_0'}((\pi_{W_1}')(\tilde{p})), \pi_{W_1'}(c_i)\}$ should be LD. The only choice we have are $d_1, \ldots, d_{m_i}$. So $\pi_{W_0'}(\tilde{P}) = d_1 \ldots d_{m_i}$. All conditions of Corollary C.2 are true and so $\tilde{P}$ is uniquely defined (upto scalar multiplication) by the reconstruction method given in Corollary C.2. So what we returned was actually a factor of $Q$.

## C.3   Time Complexity

Factoring $\pi_{W_0'}(Q), \pi_{W_1'}(Q)$ takes $poly(d)$ time (using Kaltofen's Factoring from [14]). The nested for loops run $\leq d^3$ times. Computing projections with respect to the known decomposition $W_0 \oplus W_1 \oplus W_2 = \mathbb{F}^r$ of linear forms over $r$ variables takes $poly(r)$ time. Computing $gcd$ and linear independence of linear forms takes $poly(r)$ time. The final reconstruction of $P$ using $(P_0, P_1)$ takes $poly(d, r)$ time as has been explained in Corollary C.2. Multiplying linear forms to $\Pi\Sigma$ polynomial takes $poly(d^r)$ time. Therefore overall the algorithm takes $poly(d^r)$ time. In our application $r = O(1)$ and therefore the algorithm takes $poly(d)$ time.

## D   Random Linear Transformations

This section will prove some results about linear independence and non-degeneracy under random transformations on $\mathbb{F}^r$. This will be required to make our input non-degenerate. From here onwards we fix a natural number $N \in \mathbb{N}$ and assume $0 < k < r$. Let $T \subset \mathbb{F}^r$ be a finite set with $dim(T) = r$. Next we consider two $r \times r$ matrices $\Omega, \Lambda$. Entries $\Omega_{i,j}, \Lambda_{i,j}$ are picked independently from the uniform distribution on $[N]$. For any basis $\mathcal{B}$ of $\mathbb{F}^r$ and vector $v \in \mathbb{F}^r$, let $[v]_{\mathcal{B}}$ denote the co-ordinate vector of $v$ in the basis $\mathcal{B}$. If $\mathcal{B} = \{b_1, \ldots, b_r\}$ then $[v]_{\mathcal{B}}^i$ denotes the $i$-th co-ordinate in $[v]_{\mathcal{B}}$. Let $\mathcal{S} = \{e_1, \ldots, e_r\}$ be the standard basis of $\mathbb{F}^r$. Let $E_j = sp(\{e_1, \ldots, e_j\})$ and $E_j' = sp(\{e_{j+1}, \ldots, e_r\})$, then $\mathbb{F}^r = E_j \oplus E_j'$. Let $\pi_{W_{E_j}}$ be the orthogonal projection onto $E_j$. For any matrix $M$, we denote the matrix of it's co-factors by $co(M)$. We consider the following events:

- $\mathcal{E}_0 = \{\Omega$ is not invertible $\}$
- $\mathcal{E}_1 = \{\exists t(\neq 0) \in T : \pi_{W_{E_1}}(\Omega(t)) = 0\}$
- $\mathcal{E}_2 = \{\exists \{t_1, \ldots, t_r\}$ LI vectors in $T : \{\Omega(t_1), \ldots, \Omega(t_r)\}$ is LD $\}$
- $\mathcal{E}_3 = \{\exists \{t_1, \ldots, t_r\}$ LI vectors in $T : \{\Omega(t_1), \ldots, \Omega(t_k), \Lambda\Omega(t_{k+1}), \ldots, \Lambda\Omega(t_r)\}$ is LD $\}$

- When $t_i, \Lambda, \Omega$ are clear we define the matrix $M = [M_1 \ldots M_r]$ with columns $M_i$ given as:

$$M_i = \begin{cases} [\Omega(t_i)]_{\mathcal{S}} : i \le k \\ [\Lambda\Omega(t_i)]_{\mathcal{S}} : i > k \end{cases}$$

  $M$ corresponds to the linear map

  $$e_i \mapsto \Omega(t_i) \text{ for } i \le k \text{ and } e_i \mapsto \Lambda\Omega(t_i) \text{ for } i > k$$

- $\mathcal{E}_4 = \{\{\exists \{t_1, \ldots, t_r\} \text{ LI vectors in } T \text{ and } t \in T \setminus sp(\{t_1, \ldots, t_k\}) : [co(M)[\Omega(t)]_{\mathcal{S}}]_{\mathcal{S}}^{k+1} = 0\}$
- $\mathcal{E}_5 = \mathcal{E}_4 \mid \mathcal{E}_3^c$

Next we show that the probability of all of the above events is small. Before doing that let's explain the events. This will give an intuition to why the events have low probabilities.

- $\mathcal{E}_0$ is the event where $\Omega$ is not-invertible. Random Transformations should be invertible.
- $\mathcal{E}_1$ is the event where there is a non-zero $t \in T$ such that the projection to the first co-ordinate (w.r.t. $\mathcal{S}$) of $\Omega$ applied on $t$ is 0. We don't expect this for a random linear transformation. Random Transformation on a non-zero vector should give a non-zero coefficient of $e_1$.
- $\mathcal{E}_2$ is the event such that $\Omega$ takes a basis to a LD set i.e. $\Omega$ is not invertible (random linear operators are invertible).
- $\mathcal{E}_3$ is the event such that for some basis applying $\Omega$ to the first $k$ vectors and $\Lambda\Omega$ to the last $n - k$ vectors gives a LD set. So this operation is not-invertible. For ranrom maps this should not be the case.
- $\mathcal{E}_4$ is the event that there is some basis $\{t_1, \ldots, t_r\}$ and $t$ outside $sp(t_1, \ldots, t_k)$ such that the $(k+1)^{th}$ co-ordinate of $co(M)[\Omega(t)]_{\mathcal{S}}$ w.r.t the standard basis is 0. If $M$ were invertible, clearly the set $\mathcal{B} = \{\Omega(t_1), \ldots, \Omega(t_k), \Lambda\Omega(t_{k+1}), \ldots, \Lambda\Omega(t_r)\}$ would be a basis and $co(M)$ will be a scalar multiple of $M^{-1}$. So we are asking if the $(k+1)^{th}$ co-ordinate of $\Omega(t)$ in the basis $\mathcal{B}$ is 0. For random $\Omega, \Lambda$ we would expect $M$ to be invertible and this co-ordinate to be non-zero.

Now let's formally prove everything. We will repeatedly use the popular Schawrtz-Zippel Lemma which the reader can find in [21].

▶ **Claim D.1.** $Pr[\mathcal{E}_1] \le \frac{|T|}{N}$.

**Proof.** Fix a non-zero $t = \begin{pmatrix} a_1 \\ . \\ . \\ a_r \end{pmatrix}$ with $a_i \in \mathbb{F}$ and let $\Omega = (\Omega_{i,j}), 1 \le i, j \le r$. Then

the first co-ordinate of $\Omega(t)$ is $\Omega_{1,1}a_1 + \Omega_{1,2}a_2 + \ldots + \Omega_{1,r}a_r$. Since $t \ne 0$, not all $a_i$ are 0 and this is therefore not an identically zero polynomial in $(\Omega_{1,1}, \ldots, \Omega_{1,r})$. Therefore by Schwartz-Zippel lemma $Pr[[\Omega(t)]_{\mathcal{S}}^1 = 0] \le \frac{1}{N}$. Using a union bound inside $T$ we get $Pr[\exists t(\ne 0) \in T : [\Omega(t)]_{\mathcal{S}}^1 = 0] \le \frac{|T|}{N}$. ◀

▶ **Claim D.2.** $Pr[\mathcal{E}_2] \le \frac{r}{N}$.

**Proof.** Clearly $\mathcal{E}_2 \subseteq \mathcal{E}_0$ and so $Pr[\mathcal{E}_2] \le Pr[\mathcal{E}_0]$. $\mathcal{E}_0$ corresponds to the polynomial equation $det(\Omega) = 0$. $det(\Omega)$ is a degree $r$ polynomial in $r^2$ variables and is also not identically zero, so using Schwartz-Zippel lemma we get $Pr[\mathcal{E}_2] \le Pr[\mathcal{E}_0] \le \frac{r}{N}$. ◀

▶ **Claim D.3.** $Pr[\mathcal{E}_3] \le \binom{|T|}{r}\frac{2r}{N}$.

**Proof.** Fix an LI set $t_1, \ldots, t_r$. The set $\{\Omega(t_1), \ldots, \Omega(t_k), \Lambda\Omega(t_{k+1}), \ldots \Lambda\Omega(t_r)\}$ is LD iff the $r \times r$ matrix $M$ formed by writing these vectors (in basis $\mathcal{S}$) as columns (described in part D above) has determinant 0. $M$ has entries polynomial (of degree $\leq 2$) in $\Omega_{i,j}$ and $\Lambda_{i,j}$ and so $det(M)$ is a polynomial in $\Omega_{i,j}, \Lambda_{i,j}$ of degree $\leq 2r$. For $\Omega = \Lambda = I$ (identity matrix) this matrix just becomes the matrix formed by the basis $\{t_1, \ldots, t_r\}$ which has non-zero determinant and so $det(M)$ is not the identically zero polynomial. By Schwartz-Zippel lemma $Pr[det(M) = 0] \leq = \frac{2r}{N}$. Now we vary the LI set $\{t_1, \ldots, t_r\}$, there are $\leq \binom{|T|}{r}$ such sets and so by a union bound $Pr[\mathcal{E}_3] \leq \binom{|T|}{r}\frac{2r}{N}$. ◀

▶ **Claim D.4.** $Pr[\mathcal{E}_4] \leq \binom{|T|}{r+1}\frac{2r-1}{N}$.

**Proof.** Fix an LI set $t_1, \ldots, t_r$ and a vector $t \notin sp(\{t_1, \ldots, t_k\})$. Let $t = \sum_{i=1}^{r} a_i t_i$. Since $t \notin sp(\{t_1 \ldots, t_k\})$, $a_s \neq 0$ for some $s \in \{k+1, \ldots, r\}$. Let $\mathcal{B} = \{\Omega(t_1), \ldots, \Omega(t_k), \Lambda\Omega(t_{k+1}), \ldots \Lambda\Omega(t_r)\}$. Let $M$ be the matrix whose columns are from $\mathcal{B}$ (Construction has been explained in part D above). We know that the co-factors of a matrix are polynomials of degree $\leq r - 1$ in the matrix elements. In our matrix $M$ all entries are polynomials of degree $\leq 2$ in $\Omega_{i,j}, \Lambda_{i,j}$, so all entries of $co(M)$ are polynomials of degree $\leq 2r - 2$ in $\Omega_{i,j}, \Lambda_{i,j}$. Thus $[co(M)[\Omega(t)]_\mathcal{S}]_\mathcal{S}^{k+1} = \sum_{i=1}^{r} co(M)_{k+1,i}[\Omega(t)]_\mathcal{S}^i$ is a polynomial of degree $\leq 2r - 1$. This polynomial is not identically zero. Define $\Omega$ to be the matrix (w.r.t. basis $\mathcal{S}$) of the linear map $\Omega(t_i) = e_i$ and $\Lambda$ to be the matrix (w.r.t. basis $\mathcal{S}$) of the map

$$\Lambda = \begin{cases} \Lambda(e_i) = e_i : i \notin \{s, k+1\} \\ \Lambda(e_s) = e_{k+1} \\ \Lambda(e_{k+1}) = e_s \end{cases}$$

With these values the set $\mathcal{B}$ becomes $\{e_1, \ldots, e_k, e_s, e_{k+2}, \ldots, e_{s-1}, e_{k+1}, e_{s+1}, \ldots, e_r\}$. If one now looks at $M$ i.e. the matrix formed using entries of $\mathcal{B}$ as columns it's just the permutation matrix that flips $e_s$ and $e_{k+1}$. This matrix is the inverse of itself and so has determinant $= \pm 1$,

thus $co(M) = \pm M^{-1} = \pm M$. Therefore $co(M)[\Omega(t)]_\mathcal{S} = \pm M \begin{pmatrix} a_1 \\ . \\ . \\ a_r \end{pmatrix} = \pm \begin{pmatrix} a_1 \\ . \\ a_k \\ a_s \\ a_{k+2} \\ . \\ a_{s-1} \\ a_{k+1} \\ .a_{s+1} \\ . \\ a_r \end{pmatrix}.$

Since $a_s \neq 0$, we get $[co(M)[\Omega(t)]_\mathcal{S}]_\mathcal{S}^{k+1} \neq 0$. So the polynomial is not identically zero and we can use Schwartz-Zippel Lemma to say that $Pr[[co(M)[\Omega(t)]_\mathcal{S}]_\mathcal{S}^{k+1} = 0] \leq \frac{2r-1}{N}$. Now we vary $\{t_1, \ldots, t_r, t\}$ inside $T$ and use union bound to show $Pr[\mathcal{E}_4] \leq \binom{|T|}{r+1}\frac{2r-1}{N}$. ◀

Even though this is just basic probability we include the following:

▶ **Claim D.5.** $Pr[\mathcal{E}_5] \leq \binom{|T|}{r}\frac{2r-1}{N-\binom{|T|}{r}2r}$.

**Proof.** $Pr[\mathcal{E}_5] = Pr[\mathcal{E}_4 \mid \mathcal{E}_3^c] = \frac{Pr[\mathcal{E}_4 \cap \mathcal{E}_3^c]}{Pr[\mathcal{E}_3^c]} \leq \frac{Pr[\mathcal{E}_4]}{Pr[\mathcal{E}_3^c]} \leq \binom{|T|}{r+1}\frac{\frac{2r-1}{N}}{1-\binom{|T|}{r}\frac{2r}{N}} = \binom{|T|}{r+1}\frac{2r-1}{N-\binom{|T|}{r}2r}$. ◀

In our application of the above $r = O(1), |T| = poly(d), N = 2^d$ and so all probabilities are very small as $d$ grows. So we will assume that none of the above events occur. By union bound that too will have small probability and so with very high probability $\mathcal{E}_0, \mathcal{E}_1, \mathcal{E}_2, \mathcal{E}_3, \mathcal{E}_4, \mathcal{E}_5$ do not occur.

## E    Set $\mathcal{C}$ of Candidate Linear Forms

This section deals with constructing a $poly(d)$ size set $\mathcal{C}$ which contains each $l_{ij}, (i,j) \in \{0,1\} \times [M]$. First we define the set and prove a bound on it's size.

### E.1    Structure and Size of $\mathcal{C}$

Let's recall $f = G(\alpha_0 T_0 + \alpha_1 T_1)$ and define two other polynomials:

$$g = \frac{f}{G} = \alpha_0 T_0 + \alpha_1 T_1$$

$$h = \frac{f}{Lin(f)} = \frac{g}{Lin(g)}$$

Assume $deg(h) = d_h$:

▶ **Definition E.1.** Our candidate set is defined as:

$$\mathcal{C} \overset{def}{=} \{l = x_1 - a_2 x_2 - \ldots - a_r x_r \in Lin_{\mathbb{F}}[\bar{x}] : h(a_2 x_2 + \ldots + a_r x_r, x_2, \ldots, x_r) \in \Pi\Sigma_{\mathbb{F}}^{d_h}[x_2, \ldots, x_r]\}$$

(for definition of $\Pi\Sigma_{\mathbb{F}}^{d_h}[x_2, \ldots, x_r]$ see Section 1.4).

In the claim below we show that linear forms dividing polynomials $T_i, i = 0, 1$ are actually inside $\mathcal{C}$ (first part of claim). The remaining linear forms in $\mathcal{C}$ (which we call *"spurious"*) have a nice structure (second part of claim). In the third part of our claim we arrive at a bound on the size of $\mathcal{C}$. Recall the definition of $c_{\mathbb{F}}(k)$ from Theorem 1.7.

▶ **Claim E.2.** *The following are true about our candidate set $\mathcal{C}$.*
1. $\mathcal{L}(T_i) \subseteq \mathcal{C}, i = 0, 1$.
2. *Let $k = c_{\mathbb{F}}(3) + 2$ and suppose $\{l_j; j \in [k]\} \subset \mathcal{L}(T_i)$ are LI. Then for any $l \in \mathcal{C} \setminus (\mathcal{L}(T_0) \cup \mathcal{L}(T_1))$, there exists $j \in [k]$ such that $fl(\{l, l_j\}) \cap \mathcal{L}(T_{1-i}) \neq \phi$ i.e. the line joining $l$ and $l_j$ does not intersect the set $\mathcal{L}(T_{1-i})$.*
3. $|\mathcal{C}| \leq M^4 + 2M \leq d^4 + 2d$.

**Proof.** Let's first recall the definition of our candidate set

$$\mathcal{C} \overset{def}{=} \{l = x_1 - a_2 x_2 - \ldots - a_r x_r \in Lin_{\mathbb{F}}[\bar{x}] : h(a_2 x_2 + \ldots + a_r x_r, x_2, \ldots, x_r) \in \Pi\Sigma_{\mathbb{F}}^{d_h}[x_2, \ldots, x_r]\}$$

Also recall that

$$h = \frac{g}{Lin(g)} = \frac{f}{Lin(f)} \ .$$

1. Let $l = x_1 - a_2 x_2 - \ldots - a_r x_r \in \mathcal{L}(T_{1-i})$. Let's denote the tuple $v \equiv (a_2 x_2 + \ldots + a_r x_r, x_2, \ldots, x_r)$. Since $gcd(T_0, T_1) = 1$ and $l \mid T_{1-i}$ we know that $l \nmid T_i$ and therefore $Lin(g)(v) \neq 0$. We can then compute

$$h(v) = \frac{\alpha_i T_i(v)}{Lin(g)(v)} = \alpha_i H_1(v) \ldots H_{d_h}(v) \in \Pi\Sigma_{\mathbb{F}}^{d_h}[x_2, \ldots, x_r]$$

where $H_j \in Lin_{\mathbb{F}}[x_2, \ldots, x_r]$. So $\mathcal{L}(T_i) \subseteq \mathcal{C}$ for $i = 0, 1$.

2. Consider $l = x_1 - a_2 x_2 - \ldots - a_r x_r \in \mathcal{C} \setminus (\mathcal{L}(T_0) \cup \mathcal{L}(T_1))$ and assume that $sp(\{l, l_j\}) \cap \mathcal{L}(T_{1-i}) = \phi$ for all $j \in [k]$. We know that

$$g(v) = Lin(g)(v) H_1(v) \ldots H_{d_h}(v) = \alpha_0 T_0(v) + \alpha_1 T_1(v).$$

Let $g'$ be the following identically zero $\Sigma\Pi\Sigma(3)[x_2, \ldots, x_r]$ polynomial (with circuit $\mathcal{C}'$)

$$g' = Lin(g)(v) H_1(v) \ldots H_{d_h}(v) - \alpha_0 T_0(v) - \alpha_1 T_1(v).$$

We know

$$\mathcal{C}' = gcd(\mathcal{C}') Sim(\mathcal{C}') \Rightarrow Sim(\mathcal{C}') \equiv 0.$$

Recall that $l_j(v) \mid T_i(v)$, therefore the $l_j(v)$ cannot be factors of $gcd(\mathcal{C}')$ because if they did then there exist pair $l_j, l_{(1-i)t}$ such that $\{l_j(v), l_{(1-i)t}(v)\}$ is LD or in other words $sp(\{l, l_j\}) \cap \mathcal{L}(T_{1-i}) \neq \phi$ and we have a contradiction. Also the set $\{l_j(v) : j \in [k]\}$ has dimension $\geq k - 1$ since the dimension could fall only by 1 when we go modulo a linear form (project to hyperplane). This means that $rank(Sim(\mathcal{C}')) \geq k - 1 \geq c_{\mathbb{F}}(3) + 1$.

If $Sim(\mathcal{C}')$ *were not minimal* $\Rightarrow \mathcal{C}'$ is not minimal $\Rightarrow$ one of it's gates would be 0. Since $l \notin \mathcal{L}(T_0) \cup \mathcal{L}(T_1) \Rightarrow \alpha_0 T_0(v) + \alpha_1 T_1(v) \equiv 0 \Rightarrow$ for every $j \in [k]$ there exist $l_{(1-i)j} \mid T_{1-i}$ such that $l_{(1-i)j}(v), l_j(v)$ are LD. $\Rightarrow sp(\{l, l_j\}) \cap \mathcal{L}(T_{1-i}) \neq \phi$ for $j \in [k]$, a contradiction to our assumption.

If $Sim(\mathcal{C}')$ *were minimal*, we have an identically zero simple minimal circuit $Sim(\mathcal{C}')$ with $rank(Sim(\mathcal{C}')) \geq c_{\mathbb{F}}(3) + 1$ contradicting Theorem 1.7.

So our assumption is wrong and $sp(\{l, l_j\}) \cap \mathcal{L}(T_{1-i}) \neq \phi$ for some $j \in [k]$.

3. Let $l \in \mathcal{C} \setminus (\mathcal{L}(T_0) \cup \mathcal{L}(T_1))$. Consider a set $\{l_1, \ldots, l_{k+2}\} \subset \mathcal{L}(T_i)$ of $k + 2$ LI linear forms. By the above argument there exist three distinct elements in this set say $l_1, l_2, l_3$ such that $sp(\{l_j, l\}) \cap \mathcal{L}(T_{1-i}) \neq \phi$ for $j \in [3]$. Let $\{l_1', l_2', l_3'\} \subset \mathcal{L}(T_{1-i})$ such that $l_j' \in sp(\{l_j, l\})$ for $j \in [3]$. Then $gcd(l_j, l_j') = 1$ implies that $l \in sp(\{l_j, l_j'\})$ for $j \in [3]$. Since $l, l_j, l_j'$ are all standard (coefficient of $x_1$ is 1), Lemma 1.10 tells us

$$l \in fl(\{l_j, l_j'\})$$

for $j \in [3]$. So $l$ lies on the lines $\vec{L}_j = fl(\{l_j, l_j'\})$ for $j \in [3]$. Atleast two of these lines should be distinct otherwise $dim(\{l_1, l_2, l_3\}) \leq 2$ which is a contradiction. So $l$ is the intersection of these two lines. There are $M^2$ such lines and so $M^4$ such intersections. If $l \in \mathcal{L}(T_0) \cup \mathcal{L}(T_1)$ we have $\leq 2M$ other possibilities. So $|\mathcal{C}| \leq M^4 + 2M = O(d^4)$. ◄

Let's now give an algorithm to construct this set.

## E.2 Constructing the set $\mathcal{C}$

Here is an algorithm, Algorithm 8, to construct the set $\mathcal{C}$. An explanation is given in the lemma below.

▶ **Lemma E.3.** *Given a polynomial $f \in \mathbb{F}[x_1, \ldots, x_r]$ of degree $d$ in $r$ independent variables which admits a $\Sigma\Pi\Sigma_{\mathbb{F}}(2)[x_1, \ldots, x_r]$-representation : $f = \prod\limits_{i \in [d-M]} G_i(\alpha_0 \prod\limits_{j \in [M]} l_{0j} + \alpha_1 \prod\limits_{k \in [M]} l_{1k})$ such that $G_t, l_{ij}(t \in [d - M], i \in \{0, 1\}, j \in [M])$ are standard w.r.t. the standard basis $\{x_1, \ldots, x_n\}$ then we can find in deterministic time poly($d$), the corresponding candidate set $\mathcal{C}$ (see Definition E.1) described above.*

---

**FunctionName :** Candidates
**input** : $f \in \Sigma\Pi\Sigma_{\mathbb{F}}(2)[\bar{x}]$
**output** : Set $\mathcal{C}$ of Linear Forms
**1** Define $\mathcal{C} = \phi$;;
**2** Use polynomial factorization from [14] to find $Lin(f)$;
**3** Consider polynomial $h = \frac{f}{Lin(f)}$;
**4** Let $a_2, \dots, a_r$ be variables.;
**5** Compute coefficient vector $\mathbf{b}$ of $h(a_2x_2 + \dots + a_rx_r, x_2, \dots, x_r)$.;
**6** Consider the polynomials $\{F_i, i \in [m]\}$ constructed in Corollary A.2.;
**7** Using your favorite algorithm (e.g. Buchberger's [5]) to solve polynomial equations, find all complex solutions to the system $\{F_i(\mathbf{b}) = 0, i \in [m]\}$.;
**8** For each solution $(a_2, \dots, a_r) \in \mathbb{F}^r$ do : $\mathcal{C} = \mathcal{C} \cup \{(1, a_2, \dots, a_r)\}$;
**9** **return** $\mathcal{C}$;

**Algorithm 8:** Set $\mathcal{C}$ of candidate linear forms

---

**Proof.** The proof also contains an explanation of the algorithm above

- Let $l = x_1 - a_2x_2 - \dots - a_rx_r \in \mathcal{C}$ be a candidate linear form. We know that $h(a_2x_2 + \dots + a_rx_r, x_2, \dots, x_r) \in \Pi\Sigma_{\mathbb{F}}^{d_h}[x_2, \dots, x_r] \subset \Pi\Sigma_{\mathbb{C}}^{d_h}[x_1, \dots, x_r]$.

- Using Theorem A.2 we know that $h(a_2x_2 + \dots + a_rx_r, x_2, \dots, x_r) \in \Pi\Sigma_{\mathbb{C}}^{d_h}[x_2, \dots, x_r] \Leftrightarrow$ for the coefficient vector $\mathbf{b}$ of $h(a_2x_2 + \dots + a_rx_r, x_2, \dots, x_r)$ inside $\mathbb{C}[x_2, \dots, x_r]$ satisifes $F_1(\mathbf{b}) = \dots = F_m(\mathbf{b}) = 0$ for the polynomials $\{F_i : i \in [m]\}$ obtained in Corollary A.2.

- For any $t \le d_h$, computing $(a_2x_2 + \dots + a_rx_r)^t$ requires $poly(t^r)$ time and it also has $poly(t^r)$ terms and degree $t$. Multiplying such powers to other variables and adding $poly(d_h^r)$ many such expressions also requires $poly(d_h^r)$ time. Hence computing the coefficient vector $\mathbf{b}$ takes polynomial time since $r$ is a constant. Each co-ordinate of this coefficient vector is a polynomial in $r - 1$ variables $(a_2, \dots, a_r)$ of degree $poly(d_h^r)$.

- Now we think of the $a_i$'s as our unknowns and obtain them by solving the polynomial system $\{F_i(\mathbf{b}) = 0, i \in [m]\}$. The number of polynomials is $m = poly(d^r)$ and degrees are $poly(d)$. $F_i$'s are polynomials in $poly(d^r)$ variables. Expanding $F_i(\mathbf{b})$ will clearly take $poly(d^r)$ time and now we will have $poly(d^r)$ polynomials in $r$ variables of degrees $poly(d^r)$. Note that $r = O(1)$ and so we need to solve $poly(d)$ polynomials of degree $poly(d)$ in constant many variables. Also Claim E.2 implies that the number of solutions $\le M^4 + 2M = O(poly(d))$. So using Buchberger's algorithm [5] we can solve the system for $(a_2, \dots, a_r)$ in $poly(d)$ time. Once we have the solutions we consider only those linear forms which are in $\mathbb{F}[x_1, \dots, x_r]$ and add them to $\mathcal{C}$. ◀

## F Proofs from Subsection 3.4

▶ **Claim F.1.** *Let $(S = \{l_1 \dots, l_k\}, D)$ be a Detector pair in $\mathcal{L}(T_i)$. Let $l_{k+1} \in D$. For a standard linear form $l \in V$, if $l \mid g$ then $l \notin sp(\{l_1, \dots, l_k\})$.*

**Proof.** Assume $l \mid g$ and $l \in sp(\{l_1, \dots, l_k\})$. Let $W = sp(\{l\})$, extend it to a basis and in the process obtain $W'$ such that $W \oplus W' = V$. We get

$$\pi_{W'}(\alpha_0 T_0 + \alpha_1 T_1) = 0.$$

$\pi_{W'}(\alpha_i T_i) \neq 0$ (i.e. $l \nmid T_0 T_1$), otherwise $l$ divides both $T_0, T_1$ and $gcd(T_0, T_1)$ won't be 1. So

we have an equality of non zero $\Pi\Sigma$ polynomials

$$\alpha_0 \prod_{j=1}^{M} \pi_{W'}(l_{0j}) = -\alpha_1 \prod_{j=1}^{M} \pi_{W'}(l_{1j}) \,.$$

Therefore there exists a permutation $\theta : [M] \to [M]$ such that $\{\pi_{W'}(l_{(1-i)j}), \pi_{W'}(l_{i\theta(j)})\}$ are LD $\Rightarrow l \in sp(\{l_{(1-i)j}, l_{i\theta(j)}\})$. Since $l \nmid T_0 T_1$ this also means that $l_{(1-i)j} \in sp(\{l, l_{i\theta(j)}\})$ and $l_{i\theta(j)} \in sp(\{l, l_{(1-i)j}\})$.

In particular there is an $l'_{k+1} \in \mathcal{L}(T_{1-i})$ such that $l'_{k+1} \in sp(\{l, l_{k+1}\})$ and $l_{k+1} \in sp(\{l, l'_{k+1}\})$.

Since $l \in sp(\{l_1, \ldots, l_k\}) \Rightarrow l'_{k+1} \in sp(\{l_1, \ldots, l_k, l_{k+1}\})$. All linear forms here are standard(i.e. coefficient of $x_1$ is 1) and so by Lemma 1.10, $l'_{k+1} \in fl(\{l_1, \ldots, l_k, l_{k+1}\})$. Below we use the definition of detector pair and get

$$l'_{k+1} \in fl(\{l_1, \ldots, l_k, l_{k+1}\}) \cap \mathcal{L}(T_{1-i}) \subseteq fl(\{l_1, \ldots, l_k\}) \,.$$

And $l_{k+1} \in sp(\{l, l'_{k+1}\}) \Rightarrow l_{k+1} \in sp(\{l_1, \ldots, l_k\})$ which is a contradiction to $(S, D)$ being a detector pair.                                                                                                  ◀

▶ **Claim F.2.** *Let $l \in Lin_{\mathbb{F}}[\bar{x}]$ be standard such that $l \mid g$ and $\mathcal{C}$ be the candidate set. Assume $(S = \{l_1, \ldots, l_k\}, D(\neq \phi))$ is a Detector pair in $\mathcal{L}(T_i)$. Then $|\mathcal{L}(T_{1-i}) \cap (fl(S \cup \{l\}) \setminus fl(S))| \geq 2$. That is the flat $fl(\{l_1, \ldots, l_k, l\})$ contains atleast two distinct points from $\mathcal{L}(T_{1-i})(\subseteq \mathcal{C})$ outside $fl(\{l_1, \ldots, l_k\})$.*

**Proof.** From the previous claim we know that $\{l_1, \ldots, l_k, l\}$ is an LI set. Also like above we know there exists $l'_j \in \mathcal{L}(T_{1-i}), j \in [3]$ such that $l_j \in sp(\{l, l'_j\}), l'_j \in sp(\{l, l_j\})$. Since $\{l_1, l_2, l_3\}$ are LI, atleast two of the $l'_j$'s, $j \in [3]$ must be distinct, otherwise $sp(\{l_1, l_2, l_3\}) \subset sp(\{l, l'_1\})$ which is not possible as LHS has dimension 3 and RHS has dimension 2. Thus there exist two distinct $l'_1, l'_2 \in sp(\{l_1, l_2, l_3, l\}) \subset sp(\{l_1, \ldots, l_k, l\})$. Note that $l_1, \ldots, l_k, l, l'_1, l'_2$ are all standard (i.e. coefficient of $x_1$ is 1) and so by Lemma 1.10

$$l'_j \in fl(\{l_1, \ldots, l_k, l\})$$

for $j \in [2]$.

If for any $j \in [2]$, $l'_j \in sp(\{l_1, \ldots, l_k\})$ then $l \in sp(\{l_j, l'_j\}) \Rightarrow l \in sp(\{l_1, \ldots, l_k\})$ which is a contradiction. This also shows that $l'_j \notin fl(\{l_1, \ldots, l_k\})$ for $j \in [2]$.

From what we showed above we may conclude:

$$l'_j \in fl(\{l_1, \ldots, l_k, l\}) \setminus fl(\{l_1, \ldots, l_k\})$$

for $j \in [2]$. Hence proved.                                                                                                                       ◀

▶ **Lemma F.3.** *The following are true:*
1. *If $l \mid I$ (i.e. $l$ was identified) then $l \in \mathcal{L}(G) \setminus \mathcal{L}(g)$.*
2. *If $l \mid G$ (i.e. $l$ was retained) then $(fl(\{l_1, \ldots, l_k, l\}) \setminus fl(\{l_1, \ldots, l_k\})) \cap (\mathcal{L}(T_{1-i}) \cup (\mathcal{L}(T_i) \setminus D)) \neq \phi$ that is*
   *$(fl(\{l_1, \ldots, l_k, l\}) \setminus fl(\{l_1, \ldots, l_k\}))$ contains a point from $\mathcal{L}(T_i) \setminus D$ or $\mathcal{L}(T_{1-i})$.*
3. *If $l \mid G$ and $l_{k+1} \in D$ then $l \notin sp(\{l_1, \ldots, l_k, l_{k+1}\})$.*

**Proof.**

1. Assume $l \mid I$ (i.e. $l$ was identified) and $l \mid g$. Then by Claim 3.6 we know that $\{l_1, \ldots, l_k, l\}$ are LI and so the first "*if*" condition is true. By Claim 3.7 we know that there are two other points $\{l'_1, l'_2\} \subset \mathcal{C} \cap (fl(\{l_1, \ldots, l_k, l\}) \setminus fl(\{l_1, \ldots, l_k\}))$, so the second "*if*" condition will also be true and thus $l$ will not be identified which is a contradiction. Therefore $l \in \mathcal{L}(G) \setminus \mathcal{L}(g)$.

2. Assume $l \mid G$ (i.e. $l$ was not identified). This means both "*if*" statements were true for $l$. Thus $\{l_1, \ldots, l_k, l\}$ is LI. Also there exist distinct $\{l'_1, l'_2\} \in \mathcal{C} \cap (fl(\{l_1, \ldots, l_k, l\}) \setminus fl(\{l_1, \ldots, l_k\}))$. If

   $$l'_1 \in (\mathcal{L}(T_{1-i}) \cup (\mathcal{L}(T_i) \setminus D)) \text{ or } l'_2 \in (\mathcal{L}(T_{1-i}) \cup (\mathcal{L}(T_i) \setminus D))$$

   we are done so assume both are in

   $$\mathcal{C} \setminus ((\mathcal{L}(T_{1-i}) \cup (\mathcal{L}(T_i) \setminus D)))) = (\mathcal{C} \setminus (\mathcal{L}(T_i) \cup \mathcal{L}(T_{1-i}))) \cup D$$

   If one of them say $l'_1 \in \mathcal{C} \setminus (\mathcal{L}(T_i) \cup \mathcal{L}(T_{1-i}))$, then by Part 2 of Claim E.2, for some $j \in [k]$, $sp(\{l'_1, l_j\}) \cap \mathcal{L}(T_{1-i}) \neq \phi$. Let $\tilde{l}_j \in sp(\{l'_1, l_j\}) \cap \mathcal{L}(T_{1-i}) \Rightarrow$

   $$\tilde{l}_j \in sp(\{l'_1, l_j\}) \subseteq sp(\{l_1, \ldots, l_k, l\})$$

   Since all linear forms $\tilde{l}_j, l_1, \ldots, l_k, l$ are standard (coefficient of $x_1$ is 1) by Lemma 1.10

   $$\tilde{l}_j \in fl(\{l_1, \ldots, l_k, l\})$$

   Also $\tilde{l}_j, l_j$ are LI and $\tilde{l}_j \in sp(\{l'_1, l_j\})$ together imply $l'_1 \in sp(\{l_j, \tilde{l}_j\})$. Note that $l'_1 \notin fl(\{l_1, \ldots, l_k\}) \Rightarrow l'_1 \notin sp(\{l_1, \ldots, l_k\})$ which along with $l'_1 \in sp(\{l_j, \tilde{l}_j\})$ will then give

   $$\tilde{l}_j \notin sp(\{l_1, \ldots, l_k\})$$

   So we found $\tilde{l}_j \in \mathcal{L}(T_{1-i}) \cap (fl(\{l_1, \ldots, l_k, l\}) \setminus fl(\{l_1, \ldots, l_k\}))$ and we are done.

   So the only case that remains now is that $l'_1, l'_2 \in D$. Let's complete the proof in the following steps
   - $l'_1 \in fl(\{l_1, \ldots, l_k, l\}) \setminus fl(\{l_1, \ldots, l_k\}) \Rightarrow l \in sp(\{l_1, \ldots, l_k, l'_1\})$
   - Using the above bullet, $l'_2 \in fl(\{l_1, \ldots, l_k, l\}) \Rightarrow l'_2 \in sp(\{l_1, \ldots, l_k, l'_1\})$. Linear forms $l'_2, l_1, \ldots, l_k, l$ are standard (coefficient of $x_1$ is 1) so using Lemma 1.10, $l'_2 \in fl(\{l_1, \ldots, l_k, l'_1\})$
   - $l'_2 \in D \Rightarrow l'_2 \notin fl(\{l_1, \ldots, l_k\})$
   - The above two bullets and $\{l'_1, l'_2\} \subset \mathcal{L}(T_i)$ tell us that $fl(\{l_1, \ldots, l_k, l'_1\})$ is not elementary which is a contradiction.

   So atleast one of $l'_1, l'_2$ is inside $\mathcal{L}(T_{1-i}) \cup (\mathcal{L}(T_i) \setminus D)$

3. Let $l_{k+1} \in D$ and $l \in sp(\{l_1, \ldots, l_k, l_{k+1}\})$. Since $l, l_1, \ldots, l_k, l_{k+1}$ are standard, by Lemma 1.10, $l \in fl(\{l_1, \ldots, l_k, l_{k+1}\})$. Clearly $l \notin fl(\{l_1, \ldots, l_k\})$ otherwise it would get identified at the first "*if*". Therefore $l \in fl(\{l_1, \ldots, l_k, l_{k+1}\}) \setminus fl(\{l_1, \ldots, l_k\})$ By Part 2 above let $l'_1 \in (fl(\{l_1, \ldots, l_k, l\}) \setminus fl(\{l_1 \ldots, l_k\})) \cap (\mathcal{L}(T_{1-i}) \cup (\mathcal{L}(T_i) \setminus D))$. So $l'_1 \in \mathcal{L}(T_{1-i})$ or $l'_1 \in \mathcal{L}(T_i) \setminus D$.

   This tells us that $l'_1 \in sp(\{l_1, \ldots, l_k, l_{k+1}\}) \setminus fl(\{l_1, \ldots, l_k\})$. All linear forms $l'_1, l_1, \ldots, l_k, l_{k+1}$ are standard (i.e. coefficients of $x_1$ is 1) so by Lemma 1.10 we get that $l'_1 \in fl(\{l_1, \ldots, l_k, l_{k+1}\}) \setminus fl(\{l_1, \ldots, l_k\})$. Now using the definition of detector pair $l'_1 \notin \mathcal{L}(T_{1-i})$ since $fl(\{l_1, \ldots, l_k, l_{k+1}\}) \cap \mathcal{L}(T_{1-i}) \subseteq fl(\{l_1, \ldots, l_k\})$. The flat $fl(\{l_1, \ldots, l_k, l_{k+1}\})$ is elementary in $\mathcal{L}(T_i)$, so $l'_1$ can belong here only if $l'_1 = l_{k+1}$ which is not possible since $l'_1 \notin D$. So we have a contradiction. Hence proved. ◀

▶ **Lemma F.4.** *Let $(S = \{l_1, \ldots, l_k\}, D)$ be a detector in $\mathcal{L}(T_i)$. For each $(l, l_j) \in \mathcal{C} \times S$ define the space $U_{\{l,l_j\}} = sp(\{l, l_j\})$. Extend $\{l, l_j\}$ to a basis and in the process obtain $U'_{\{l,l_j\}}$ such that $V = U_{\{l,l_j\}} \oplus U'_{\{l,l_j\}}$. Define the set:*

$$X = \{l \in \mathcal{C} : \pi_{U'_{\{l,l_j\}}}(f) \neq 0, \ for \ all \ l_j \in S\}$$

*Then $D \subset X \subset \mathcal{L}(T_i)$.*

**Proof.** $\underline{(D \subset X):}$ Consider $l_{k+1} \in D$. Since $D \subset \mathcal{L}(T_i) \Rightarrow l_{k+1} \in \mathcal{C}$. Assume $l_{k+1} \notin X$, so there exists a $j \in [k]$ such that $\pi_{U'_{\{l_{k+1},l_j\}}}(f) = 0$. That is:

$$\pi_{U'_{\{l_{k+1},l_j\}}}(G(\alpha_0 T_0 + \alpha_1 T_1)) = 0.$$

So

$$\prod_{t \in [N_1]} \pi_{U'_{\{l_{k+1},l_j\}}}(G_t)(\alpha_0 \prod_{s \in [M]} \pi_{U'_{\{l_{k+1},l_j\}}}(l_{0s}) + \alpha_1 \prod_{s \in [M]} \pi_{U'_{\{l_{k+1},l_j\}}}(l_{1s})) = 0$$

Now

$$l_j \in \mathcal{L}(T_i) \Rightarrow \pi_{U'_{\{l_{k+1},l_j\}}}(T_i) = 0 \Rightarrow \prod_{t \in [N_1]} \pi_{U'_{\{l_{k+1},l_j\}}}(G_t) \prod_{s \in [M]} \pi_{U'_{\{l_{k+1},l_j\}}}(l_{(1-i)s}) = 0.$$

Since $G_t \mid G$, by Part (3) of Lemma 3.9 $\pi_{U'_{\{l_{k+1},l_j\}}}(G_t) \neq 0$ for all $t \in [N_1]$. If for some $s \in [M]$, $\pi_{U'_{\{l_{k+1},l_j\}}}(l_{(1-i)s}) = 0$ then $l_{(1-i)s} \in sp(\{l_j, l_{k+1}\}) \Rightarrow l_{(1-i)s} \in sp(\{l_1, \ldots, l_k, l_{k+1}\}) \Rightarrow l_{(1-i)s} \in sp(\{l_1, \ldots, l_k\})$ (by definition of Detector Pair in 3.4).

$$l_{(1-i)s} \in sp(\{l_j, l_{k+1}\}) \text{ and } \{l_{(1-i)s}, l_j\} \text{ LI } \Rightarrow l_{k+1} \in sp(\{l_{(1-i)s}, l_j\})$$

This means $l_{k+1} \in sp(\{l_1, \ldots, l_k, l_{(1-i)s}\}) \subset sp(\{l_1, \ldots, l_k\})$ which is a contradiction to $l_{k+1} \in D$. So $\pi_{U'_{\{l_{k+1},l_j\}}}(f) \neq 0$ for all $j \in [k] \Rightarrow l_{k+1} \in X$. Therefore $D \subset X$.

$\underline{(X \subset \mathcal{L}(T_i)):}$ Consider $l \in X$. We need to show $l \in \mathcal{L}(T_i)$. We already know $l \in \mathcal{C}$.

- If $l \in \mathcal{L}(T_{1-i})$, then $\pi_{U'_{\{l,l_j\}}}(f) = 0$ for all $j \in [k]$ since $l \mid T_{1-i}$ and $l_j \mid T_i$. Contradiction to $l \in X$.

- If $l \in \mathcal{C} \setminus (\mathcal{L}(T_i) \cup \mathcal{L}(T_{1-i}))$ by Part 2 of Claim E.2 we know that there exists $j \in [k]$ such that $sp(\{l_j, l\}) \cap \mathcal{L}(T_{1-i}) \neq \phi$. Let $l'_j \in sp(\{l_j, l\}) \cap \mathcal{L}(T_{1-i})$. We show that $sp(\{l'_j, l_j\}) = sp(\{l_j, l\}) = U_{\{l_j, l\}}$.
  - $l'_j \in sp(\{l_j, l\}) \Rightarrow sp(\{l'_j, l_j\}) \subset sp(\{l_j, l\})$.
  - Let $l'_j = \alpha l_j + \beta l$. We know that $\{l_j, l'_j\}$ are LI since $l_j \in \mathcal{L}(T_i)$ and $l'_j \in \mathcal{L}(T_{1-i})$. So $\beta \neq 0 \Rightarrow l \in sp(\{l'_j, l_j\}) \Rightarrow sp(\{l, l_j\}) \subset sp(\{l'_j, l_j\}) \Rightarrow sp(\{l, l_j\}) = sp(\{l'_j, l_j\})$.
  Use the same extension for $sp(\{l, l_j\}) = sp(\{l'_j, l_j\}) = U_{\{l_j, l\}}$ to get $\pi_{U'_{\{l,l_j\}}}(f) = \pi_{U'_{\{l'_j,l_j\}}}(f) = 0$ (since $l'_j \mid T_{1-i}$ and $l_j \mid T_i$). Contradiction to $l \in X$.

Therefore $l \in \mathcal{L}(T_i) \Rightarrow X \subset \mathcal{L}(T_i)$. ◀

## G    Proofs from Subsection 3.5

▶ **Claim G.1.** *The following is true*

$$\frac{(2 - v(\delta, \theta))}{v(\delta, \theta)} \leq \frac{1 - \delta}{\delta}.$$

**Proof.** Note that

$$\frac{(2 - v(\delta, \theta))}{v(\delta, \theta)} = \begin{cases} \frac{1+\delta+\theta}{1-\delta-\theta} & \text{if } |\mathcal{L}(T_0)| \leq \theta|\mathcal{L}(T_1)| \\ \frac{3-(1-\delta)(1+\theta)}{(1-\delta)(1+\theta)-1} & \text{if } \theta|\mathcal{L}(T_1)| < |\mathcal{L}(T_0)| \leq |\mathcal{L}(T_1)| \end{cases}$$

By simple computation $\delta \in (0, \frac{7-\sqrt{37}}{6})$ gives

$$3\delta^2 - 7\delta + 1 > 0 \Rightarrow 0 < \frac{3\delta}{1-\delta} < 1 - 3\delta < 1 \Rightarrow \frac{1+\delta+\theta}{1-\delta-\theta} < \frac{1-\delta}{\delta}$$

Also

$$\theta > \frac{3\delta}{1-\delta} \Rightarrow \frac{3-(1-\delta)(1+\theta)}{(1-\delta)(1+\theta)-1} < \frac{1-\delta}{\delta} \qquad \blacktriangleleft$$

▶ **Lemma G.2.** *Let $k = c_{\mathbb{F}}(3) + 2$ (see definition of $c_{\mathbb{F}}(k)$ in Theorem 1.7). Fix $\delta, \theta$ in range given in Claim 3.12 above. Then for some $i \in \{0, 1\}$ there exists a Detector Pair $(S = \{l_1, \ldots, l_k\}, D)$ in $\mathcal{L}(T_i)$ with $|D| \geq v(\delta, \theta) \max(|\mathcal{L}(T_0)|, |\mathcal{L}(T_1)|)$.*

**Proof.** We assume $|\mathcal{L}(T_0)| \leq \mathcal{L}(T_1)$. The other case gives the same result for(maybe) a different value of $i$. We will consider linear forms as points in the space $\mathbb{F}^r$. Let's consider the two cases used in the definition of $v(\delta, \theta)$.

**Case 1: $|\mathcal{L}(T_0)| \leq \theta|\mathcal{L}(T_1)|$ (i.e. $\mathcal{L}(T_0)$ is much smaller ) $\Rightarrow v(\delta, \theta) = 1 - \delta - \theta$** Since $dim(\mathcal{L}(T_1)) \geq r - 1 \geq C_{2k-1} > C_k$ (see Appendix B for definition of $C_k$) by Corollary B.5 there exists a set $S$ of $k$ LI points say $S = \{l_1, \ldots, l_k\} \subseteq \mathcal{L}(T_1)$ and a set $Z \subseteq \mathcal{L}(T_1)$ of size $\geq (1 - \delta)|\mathcal{L}(T_1)|$ such that for any $l_{k+1} \in Z$

- $l_{k+1} \notin fl(\{l_1, \ldots, l_k\})$.
- $fl(\{l_1, \ldots, l_k, l_{k+1}\})$ is elementary in $\mathcal{L}(T_1)$.

Next we define our set $D$ according to the condition we needed in the definition of detector (See Subsection 3.4).

$$D \stackrel{def}{=} \{l_{k+1} \in Z : fl(\{l_1, \ldots, l_k, l_{k+1}\}) \cap \mathcal{L}(T_0) \subset fl(\{l_1, \ldots, l_k\})\}$$

In the following lines we will show that this set $D$ has large size, to be precise:

$$|D| \geq (1 - \delta - \theta)|\mathcal{L}(T_1)|$$

We do this in steps:

1. First we define a special subset of $Z$

   $$\tilde{Z} = \{l_{k+1} \in Z : (fl(\{l_1, \ldots, l_{k+1}\}) \setminus fl(\{l_1, \ldots, l_k\})) \cap \mathcal{L}(T_0) \neq \phi\}$$

   We claim that $Z \setminus \tilde{Z} \subset D$. Let $l_{k+1} \in Z \setminus \tilde{Z} \Rightarrow (fl(\{l_1, \ldots, l_{k+1}\}) \setminus fl(\{l_1, \ldots, l_k\})) \cap \mathcal{L}(T_0) = \phi \Rightarrow fl(\{l_1, \ldots, l_{k+1}\}) \cap \mathcal{L}(T_0) \subset fl(\{l_1, \ldots, l_k\})$ and so $l_{k+1} \in D$.

2. Next we show that for distinct $l_{k+1}, \tilde{l}_{k+1} \in Z(\subseteq \mathcal{L}(T_1))$

   $$(fl(\{l_1, \ldots, l_k, l_{k+1}\}) \setminus fl(\{l_1, \ldots, l_k\})) \cap (fl(\{l_1, \ldots, l_k, \tilde{l}_{k+1}\}) \setminus fl(\{l_1, \ldots, l_k\})) = \phi$$

   If not then there exist scalars $\mu_j, \nu_j, j \in [k+1]$ such that

   $$\nu_1 l_1 + \ldots \nu_k l_k + \nu_{k+1} l_{k+1} = \mu_1 l_1 + \ldots \mu_k l_k + \mu_{k+1} \tilde{l}_{k+1}$$

   with $\nu_{k+1} \neq 0$ implying that $l_{k+1} \in sp(\{l_1, \ldots, l_k, \tilde{l}_{k+1}\})$. Since all linear forms are *standard* this implies $l_{k+1} \in fl(\{l_1, \ldots, l_k, \tilde{l}_{k+1}\})$ (see Lemma 1.10). Also $l_{k+1} \in Z \Rightarrow l_{k+1} \notin fl(\{l_1, \ldots, l_k\})$. Together this means that $l_{k+1} \in fl(\{l_1, \ldots, l_k, \tilde{l}_{k+1}\}) \setminus fl(l_1, \ldots, l_k)$ and we arrive at a contradiction to $fl(\{l_1, \ldots, l_k, \tilde{l}_{k+1}\})$ being elementary.

3. From what we showed above every $l \in \mathcal{L}(T_0)$ can belong to atmost one of the sets $fl(\{l_1, \ldots, l_{k+1}\}) \setminus fl(\{l_1, \ldots, l_k\})$ with $l_{k+1} \in Z$ (since intersection between two such sets is $\phi$) and therefore there can be atmost $|\mathcal{L}(T_0)|$ such $l_{k+1}$'s in $\tilde{Z} \Rightarrow |\tilde{Z}| \leq |\mathcal{L}(T_0)|$.

   So we get:

   $$|D| \geq |Z| - |\mathcal{L}(T_0)| \geq (1 - \delta - \theta)|\mathcal{L}(T_1)|$$

   $(S, D)$ is a detector pair in $\mathcal{L}(T_1)$ by the choice of $Z$ and $D$.

**Case 2: $\theta|\mathcal{L}(T_1)| < |\mathcal{L}(T_0)| \leq |\mathcal{L}(T_1)|$ (i.e. sizes are comparable ) $\Rightarrow v(\delta, \theta) = (1 - \delta)(1 + \theta) - 1$** Since $dim(\mathcal{L}(T_0) \cup \mathcal{L}(T_1)) = r > C_{2k-1}$, by Corollary B.5 we know that there exist $2k-1$ independent points $l_1, \ldots, l_{2k-1} \in \mathcal{L}(T_0) \cup \mathcal{L}(T_1)$ and a set $Z \subseteq \mathcal{L}(T_0) \cup \mathcal{L}(T_1)$ of size $\geq (1 - \delta)(|\mathcal{L}(T_0)| + |\mathcal{L}(T_1)|)$ such that for all $l \in Z$

- $l \notin fl(\{l_1, \ldots, l_{2k-1}\})$.
- $fl(\{l_1, \ldots, l_{2k-1}, l\})$ is elementary in $\mathcal{L}(T_0) \cup \mathcal{L}(T_1)$.

By pigeonhole principle, $k$ of the $\{l_j\}_{j=1}^{2k-1}$ points must belong to either $\mathcal{L}(T_0)$ or $\mathcal{L}(T_1)$. Let's assume they belong to $\mathcal{L}(T_i)$ (for some $i \in \{0, 1\}$) (say the points are $l_1, \ldots, l_k$), then consider $D = Z \cap \mathcal{L}(T_i)$. Clearly for every $l \in D$, $l \notin fl(\{l_1, \ldots, l_k\})$ and $fl(\{l_1, \ldots, l_k, l\})$ is elementary in $\mathcal{L}(T_0) \cup \mathcal{L}(T_1)$. This immediately tells us that $(S = \{l_1, \ldots, l_k\}, D)$ satisfies all properties of being a detector pair in $\mathcal{L}(T_i)$. We defined $D = Z \cap \mathcal{L}(T_i)$. Since $Z \subseteq \mathcal{L}(T_i) \cup \mathcal{L}(T_{1-i})$ we have $Z = (Z \cap \mathcal{L}(T_i)) \cup (Z \cap \mathcal{L}(T_{1-i})) \subset D \cup \mathcal{L}(T_{1-i})$ giving

$$|D| + |\mathcal{L}(T_{1-i})| \geq |Z| \Rightarrow |D| \geq |Z| - |\mathcal{L}(T_{1-i})| \geq (1 - \delta)(|\mathcal{L}(T_0)| + |\mathcal{L}(T_1)|) - |\mathcal{L}(T_{1-i})|$$

$$\geq ((1 - \delta)(1 + \theta) - 1) \max(|\mathcal{L}(T_0)|, |\mathcal{L}(T_1)|)$$

Combining the two cases we see that for some $i \in \{0, 1\}$ there exists a Detector set $(S = \{l_1, \ldots, l_k\}, D)$ in $\mathcal{L}(T_i)$ with $|D| \geq v(\delta, \theta) \max(|\mathcal{L}(T_0)|, |\mathcal{L}(T_1)|)$. ◀

▶ **Lemma G.3.** *The following are true:*
1. $dim(\pi_{W_0^\perp}(\mathcal{L}(U_{1-i}))) > C_4$
2. $\pi_{W_0^\perp}(\mathcal{L}(U_{1-i})) \cap \pi_{W_0^\perp}(D) = \phi$
3. $|\pi_{W_0^\perp}(\mathcal{L}(U_{1-i}))| \leq \frac{1-\delta}{\delta}|\pi_{W_0^\perp}(D)|$

**Proof.**
1. Since $dim(\mathcal{L}(U_{1-i})) \geq r - 1$ we get $dim(\pi_{W_0^\perp}(\mathcal{L}(U_{1-i}))) \geq r - 1 - k > C_4$.
2. Assume $\exists\, d_1 \in D, u \in \mathcal{L}(U_{1-i})$ such that $\pi_{W_0^\perp}(d) = \pi_{W_0^\perp}(u) \Rightarrow \exists \lambda, \nu \in \mathbb{F}$ such that $\nu d_1 + \lambda u \in W_0^\perp$. Since $\pi_{\tilde{W}_0}(d_1) \neq 0$ both $\nu, \lambda \neq 0$. Thus $u \in sp(\{l_1, \ldots, l_k, d_1\}) \Rightarrow u \in fl(\{l_1, \ldots, l_k, d_1\})$ (using Lemma 1.10 since all linear forms involved are *standard* i.e. have coefficient of $x_1$ equal to 1). Also $u \in \mathcal{L}(GT_{1-i}) \Rightarrow u \in fl(\{l_1, \ldots, l_k, d_1\}) \cap (\mathcal{L}(G) \cup \mathcal{L}(T_{1-i}))$. We know from Part (2) of Lemma 3.9 that $fl(\{l_1, \ldots, l_k, d_1\}) \cap \mathcal{L}(G) = \phi \Rightarrow u \in fl(\{l_1, \ldots, l_k, d_1\}) \cap \mathcal{L}(T_{1-i}) \subseteq fl\{l_1, \ldots, l_k\}$ because $(S, D)$ was a detector pair. But $u \in fl(\{l_1, \ldots, l_k\}) \Rightarrow d_1 \in sp(\{l_1, \ldots, l_k\})$ which is a contradiction because $d_1 \in D$ and $(S, D)$ is a detector pair.
3. We first plan to show $\pi_{W_0^\perp}(\mathcal{L}(U_{1-i})) \subset \pi_{W_0^\perp}(\mathcal{L}(T_{1-i})) \cup \pi_{W_0^\perp}(\mathcal{L}(T_i) \setminus D)$. Clearly $U_{1-i} \mid GT_{1-i} \Rightarrow \mathcal{L}(U_{1-i}) \subset \mathcal{L}(GT_{1-i}) \Rightarrow \pi_{W_0^\perp}(\mathcal{L}(U_{1-i})) \subset \pi_{W_0^\perp}(\mathcal{L}(GT_{1-i})) \subset \pi_{W_0^\perp}(\mathcal{L}(G)) \cup \pi_{W_0^\perp}(\mathcal{L}(T_{1-i}))$. Now consider any $l \in \mathcal{L}(G)$. We know that $(S_0 = \{l_1, \ldots, l_k\}, D)$ is a detector pair, so by Part (2) of Lemma 3.9 we get

   $$(fl(\{l_1, \ldots, l_k, l\}) \setminus fl(\{l_1, \ldots, l_k\})) \cap (\mathcal{L}(T_{1-i}) \cup (\mathcal{L}(T_i) \setminus D)) \neq \phi$$

So there exists $l' \in \mathcal{L}(T_{1-i}) \cup (\mathcal{L}(T_i) \setminus D)$ such that $\pi_{W_0^\perp}(l), \pi_{W_0^\perp}(l')$ are both non-zero and are LD $\Rightarrow \pi_{W_0^\perp}(l) = \pi_{W_0^\perp}(l')$ implying that $\pi_{W_0^\perp}(\mathcal{L}(G)) \subset \pi_{W_0^\perp}(\mathcal{L}(T_{1-i}) \cup (\mathcal{L}(T_i) \setminus D))$ giving us $\pi_{W_0^\perp}(\mathcal{L}(U_{1-i})) \subset \pi_{W_0^\perp}(\mathcal{L}(T_{1-i})) \cup \pi_{W_0^\perp}(\mathcal{L}(T_i) \setminus D)$ and therefore

$$|\pi_{W_0^\perp}(\mathcal{L}(U_{1-i}))| \leq |\pi_{W_0^\perp}(\mathcal{L}(T_{1-i}))| + |\pi_{W_0^\perp}(\mathcal{L}(T_i) \setminus D)|$$

Now we try to show $|\pi_{W_0^\perp}(\mathcal{L}(T_i) \setminus D)| = |\pi_{W_0^\perp}(\mathcal{L}(T_i))| - |D|$

**a.** It's straightforward to see $\pi_{W_0^\perp}(\mathcal{L}(T_i)) = \pi_{W_0^\perp}(D) \cup \pi_{W_0^\perp}(\mathcal{L}(T_i) \setminus D)$. Also $\pi_{W_0^\perp}(\mathcal{L}(T_i) \setminus D) \cap \pi_{W_0^\perp}(D) = \phi$. If not then there exists $l' \in \mathcal{L}(T_i) \setminus D, l'' \in D$ such that $0 \neq \pi_{W_0^\perp}(l'') = \pi_{W_0^\perp}(l') \Rightarrow \pi_{W_0^\perp}(l''), \pi_{W_0^\perp}(l')$ are LD $\Rightarrow l' \in sp\{l_1, \ldots, l_k, l''\} \setminus sp\{l_1, \ldots, l_k\} \Rightarrow$ (by Lemma 1.10), $l' \in fl\{l_1, \ldots, l_k, l''\} \setminus fl\{l_1, \ldots, l_k\}$ which is a contradiction to the flat being elementary inside $\mathcal{L}(T_i)$. So $|\pi_{W_0^\perp}(\mathcal{L}(T_i))| = |\pi_{W_0^\perp}(D)| + |\pi_{W_0^\perp}(\mathcal{L}(T_i) \setminus D)|$.

**b.** $\pi_{W_0^\perp}$ is injective on $D$. Let $\pi_{W_0^\perp}(l') = \pi_{W_0^\perp}(l'')$ for LI forms $\{l', l''\} \subset D$, then $l' \in sp(\{l_1, \ldots, l_k, l''\}) \Rightarrow$ (by Lemma 1.10), $l' \in fl(\{l_1, \ldots, l_k, l''\})$ and clearly $l' \notin fl\{l_1, \ldots, l_k\}$ (since it's in $D$), which is again a contradiction to the flat being elementary , thus $|\pi_{W_0^\perp}(D)| = |D| = |D|$ (since $D$ is a set of *normal* linear forms ).

Combining these with Claim 3.12 and Lemma 3.13 we get

$$|\pi_{W_0^\perp}(\mathcal{L}(U_{1-i}))| \leq 2 \max(|\mathcal{L}(T_0)|, |\mathcal{L}(T_1)|) - |D| \leq (2 - v(\delta, \theta)) \max(|\mathcal{L}(T_0)|, |\mathcal{L}(T_1)|)$$

$\Rightarrow$

$$\frac{|\pi_{W_0^\perp}(\mathcal{L}(U_{1-i}))|}{|\pi_{W_0^\perp}(D)|} \leq \frac{(2 - v(\delta, \theta))}{v(\delta, \theta)} \leq \frac{1 - \delta}{\delta} \qquad \blacktriangleleft$$

## H Proofs from Section 4

Our field $\mathbb{F}$ has characteristic zero. For simplicity let's assume it is an extension of $\mathbb{Q}$ and therefore contains $\mathbb{Z}$. All random selections are done from the set $[N] = \{1, \ldots, N\}$.

▶ **Lemma H.1.** *Let $\mathbb{F}^n$ be the $n$ dimensional vector space over $\mathbb{F}$. Suppose $v_i : i \in [n]$ are vectors in $\mathbb{F}^n$ with each co-ordinate chosen independently from the uniform distribution on $[N]$. Consider the event*

$$\mathcal{E} = \{\{v_1, \ldots, v_n\} \text{ are LI }\}.$$

*Then $Pr[\mathcal{E}] \geq 1 - \frac{n}{N}$.*

**Proof.** Each $v_i \in \mathbb{F}^n$ is chosen such that each co-ordinate is chosen uniformly randomly from the set $[N]$. Let $v_i$ be the vector $(V_{i,1}, \ldots, V_{i,n})$. Consider the matrix $\tilde{V} = (V_{i,j})$. The $v_i$'s will be linearly independent if and only if $\tilde{V}$ is invertible i.e. $det(V_{i,j}) \neq 0$. Note that $det(V_{i,j})$ is not the zero polynomial since the monomial $V_{1,1} V_{2,2} \ldots V_{n,n}$ has coefficient 1. Now we can use Schwartz-Zippel Lemma [21] on this polynomial to yield:

$$Pr[det(\tilde{V}) = 0]. \leq \frac{n}{N}$$

Therefore $Pr[\mathcal{E}] = Pr[det(\tilde{V}) \neq 0] \geq 1 - \frac{n}{N}$. ◀

▶ **Lemma H.2.** *Assume conditions in the previous lemma. For a fixed $r$, consider the subspaces $V = sp\{v_1, \ldots, v_r\}$ and $V' = sp\{v_{r+1}, \ldots, v_n\}$. Let's assume that that $\mathcal{E}$ occurs i.e. $\{v_1, \ldots, v_n\}$ are LI. So $dim(V) = r$. We know $\mathbb{F}^n = V \oplus V'$. Let $\pi_V : \mathbb{F}^n \to V$ be the*

*orthogonal projection onto $V$ under this decomposition. Let $T \subset \mathbb{F}^n$ be finite. Consider the event*

$$\mathcal{F} = \{\exists \text{ an LI set } \{l_1, \ldots, l_r\} \subset T \text{ such that } \{\pi_V(l_1), \ldots, \pi_V(l_r)\} \text{ is LD }\}.$$

*Then $Pr[\mathcal{F}] \leq \binom{|T|}{r}\{\frac{n}{N} + \frac{r(n-1)}{N}\}$.*

**Proof.** Fix $\{l_1, \ldots, l_r\} \subset T$ an LI set. Extend it to get a basis $\{l_1, \ldots, l_n\}$ of $\mathbb{F}^n$. Let $l_i = \sum_{j \in [n]} L_{i,j} e_j$ and $L$ be the matrix $(L_{i,j})$. From the discussion above we have $\tilde{V} = (V_{i,j})$. Now let $P_r$ be the $n \times n$ matrix

$$P_r = \begin{bmatrix} I_r & 0_{r,n-r} \\ 0_{n-r,r} & 0_{n-r,n-r} \end{bmatrix}$$

where $I_r$ is the $r \times r$ identity matrix and $0_{p,q}$ is the $p \times q$ matrix with all 0 entries. Also for any $n \times n$ matrix $A$, define $M_r(A)$ to be the principal $r \times r$ minor of $A$. Consider the equation given by

$$det(M_r(P_r Lco(\tilde{V}))) = 0$$

where $co(\tilde{V})$ is the co-factor matrix of $\tilde{V}$. Since entries of $co(\tilde{V})$ are polynomials in the $V_{i,j}$'s and $L$ is a fixed matrix, the entries of $P_r Lco(\tilde{V})$ are polynomials in $V_{i,j}$'s. So $det(M_r(P_r Lco(\tilde{V})))$ is a polynomial in $V_{i,j}$'s. This polynomial can't be identically 0. Choose $V_{i,j} = L_{i,j}$, then since $\tilde{V}$ is invertible, $Lco(\tilde{V}) = det(L)I$ giving $P_r Lco(\tilde{V}) = det(L)P_r \Rightarrow det(M_r(P_r Lco(\tilde{V}))) = det(L) \neq 0$. Degree of the polynomial $det(M_r(P_r Lco(\tilde{V})))$ is clearly $\leq r(n-1)$. Therefore by Schwartz Zippel Lemma

$$Pr[det(M_r(P_r Lco(\tilde{V}))) = 0] \leq \frac{r(n-1)}{N}.$$

Consider the set

$$S(\{l_1, \ldots, l_r\}) = \{(V_{i,j}) : det(\tilde{V}) \neq 0, det(M_r(P_r Lco(\tilde{V})) \neq 0\}.$$

On this set $S(\{l_1, \ldots, l_r\})$, $\{v_1, \ldots, v_n\}$ is a basis and we have the following matrix equations:

$$\begin{bmatrix} v_1 \\ \cdot \\ \cdot \\ v_n \end{bmatrix} = \tilde{V} \begin{bmatrix} e_1 \\ \cdot \\ \cdot \\ e_n \end{bmatrix} \text{ and } \begin{bmatrix} l_1 \\ \cdot \\ \cdot \\ l_n \end{bmatrix} = L \begin{bmatrix} e_1 \\ \cdot \\ \cdot \\ e_n \end{bmatrix} \Rightarrow \begin{bmatrix} l_1 \\ \cdot \\ \cdot \\ l_n \end{bmatrix} = L\tilde{V}^{-1} \begin{bmatrix} v_1 \\ \cdot \\ \cdot \\ v_n \end{bmatrix}$$

and so

$$\begin{bmatrix} \pi_V(l_1) \\ \cdot \\ \pi_V(l_r) \end{bmatrix} = \frac{1}{det(\tilde{V})} M_r(P_r Lco(\tilde{V})) \begin{bmatrix} v_1 \\ \cdot \\ v_r \end{bmatrix}$$

Therefore $\{\pi_V(l_1), \ldots, \pi_V(l_r)\}$ is an LI set. Now $S(\{l_1, \ldots, l_r\})^c = \{(V_{i,j}) : det(\tilde{V}) = 0$ or $det(M_r Lco(M)) = 0\} \Rightarrow Pr[S(\{l_1, \ldots, l_r\})^c] \leq \frac{n}{N} + \frac{r(n-1)}{N}$. Next we vary $\{l_1, \ldots, l_r\}$ and apply union bound to get

$$Pr[\mathcal{F}] \leq \sum_{\{l_1, \ldots, l_r\} \subset T} S(\{l_1, \ldots, l_r\})^c \leq \binom{|T|}{r}\left\{\frac{n}{N} + \frac{r(n-1)}{N}\right\}.$$

In our application $|T| = poly(d)$ and $r$ is a constant, so we choose $N = 2^{d+n}$ and make this probability very small. ◄

▶ **Lemma H.3.** *Let $f|_V(\bar{X}) = \sum\limits_{\{\bar{\alpha}:|\bar{\alpha}|=d\}} a_{\bar{\alpha}}\bar{X}^{\bar{\alpha}}$ be a homogeneous multivariate polynomial of degree $d$ in $r$ variables $X_1, \ldots, X_r$. Let $p_i : 1 \leq i \leq \binom{d+r-1}{r-1}$ be randomly chosen points in $V$ ( dimension $r$ random subspace of $\mathbb{F}^n$ chosen in the above lemmas). Then with high probability one can find all the $a_{\bar{\alpha}}$.*

**Proof.** We evaluate the polynomial at each of the $p_i$'s. So we have $\binom{d+r-1}{r-1}$ evaluations. The number of coefficients is also $\binom{d+r-1}{r-1}$ so we get a linear system in the coefficients where the matrix $(X)$ entries are just monomials evaluated at the $p_i$'s. Since $f$ is not identically zero clearly there exist values for the points $p_i$'s such that the determinant of this matrix is non zero polynomial so it cannot be identically zero. Now the degree of the determinant polynomial is bounded by $d\binom{d+r-1}{r-1} \leq poly((d+r)^r)$. So by Schwarz Zippel lemma

$$Pr[a_{\bar{\alpha}} \text{ is recovered correctly }] = Pr[det(X) \neq 0] \geq 1 - \frac{poly(d^r)}{N} \qquad ◄$$