Electronic Colloquium on Computational Complexity, Revision 1 of Report No. 125 (2020)

Efficient reconstruction of depth three circuits with top fan-in two

Gaurav Sinha*

Abstract

In this paper we develop efficient randomized algorithms to solve the black-box reconstruction problem for polynomials over finite fields, computable by depth three arithmetic circuits with alternating addition/multiplication gates, such that output gate is an addition gate with in-degree two. Such circuits naturally compute polynomials of the form $G \times (T_1 + T_2)$, where G, T_1, T_2 are product of affine forms computed at the first layer in the circuit, and polynomials T_1, T_2 have no common factors. Rank of such a circuit is defined to be the dimension of vector space spanned by all affine factors of T_1 and T_2. For any polynomial f computable by such a circuit, $rank(f)$ is defined to be the minimum rank of any such circuit computing it. Our work develops randomized reconstruction algorithms which take as input black-box access to a polynomial f (over finite field $\mathbb{F}$), computable by such a circuit. Here are the results.

- [Low rank] : When $5 \leq rank(f) = O(\log^3 d)$, it runs in time $(nd^{\log^3 d} \log |\mathbb{F}|)^{O(1)}$, and, with high probability, outputs a depth three circuit computing f, with top addition gate having in-degree $\leq d^{rank(f)}$.
- [High rank] : When $rank(f) = \Omega(\log^3 d)$, it runs in time $(nd \log |\mathbb{F}|)^{O(1)}$, and, with high probability, outputs a depth three circuit computing f, with top addition gate having in-degree two.

Prior to our work, black-box reconstruction for this circuit class was addressed in [Shp07, KS09, Sin16b]. Reconstruction algorithm in [Shp07] runs in time quasi-polynomial in $n, d, |\mathbb{F}|$ and that in [KS09] is quasi-polynomial in $d, |\mathbb{F}|$. Algorithm in [Sin16b] works only for polynomials over characteristic zero fields. Thus, ours is the first blackbox reconstruction algorithm for this class of circuits, that runs in time polynomial in $\log |\mathbb{F}|$. This problem has been mentioned as an open problem in [GKL12] (STOC 2012). In the high rank case, our algorithm runs in $(nd \log |\mathbb{F}|)^{O(1)}$ time, thereby significantly improving the existing algorithms in [Shp07, KS09].

*Adobe Research Bangalore, India, email: gasinha@adobe.com

ISSN 1433-8092

1 Introduction

Arithmetic circuit reconstruction: Arithmetic circuits (Definition 1.1 in [SY10]) are Directed Acyclic Graphs (DAG), describing succinct ways of computing multivariate polynomials. Analogous to the exact learning problem for Boolean circuits [Ang88], Black-box reconstruction problem (Section 5, [SY10]) has been asked for arithmetic circuits:

Given oracle[1] *access to a multivariate polynomial computable by an arithmetic circuit of size s, construct an explicit circuit (ideally poly(s) sized) that computes the same polynomial.*

Depth three circuit reconstruction: These are layered circuits with three layers of alternating plus(Σ) gates and product(Π) gates. Reconstruction of $\Pi\Sigma\Pi$ circuits amounts to black-box polynomial factorization into sparse factors and efficient randomized algorithms are known [KT90]. However no such algorithm is known for $\Sigma\Pi\Sigma$ circuits[2] (Definition 7). First non-trivial algorithm for this class, which takes exponential time in the fan-in of the multiplication gates, was given in [KS03]. In fact, in a recent work, [KS18] (Section 1.2) discuss that efficient reconstruction algorithms for depth three circuits will imply super-polynomial lower bounds for them which is a long standing open problem in Arithmetic complexity [SW99, Wig06]. Therefore, even for the class of depth three circuits, reconstruction problem appears to be very challenging. Current state of the art reconstruction algorithms for this class either work in the average case [KS18] or restrict the fan-in of the top addition gate (also called top fan-in) [Shp07, KS09, Sin16b].

Bounded top fan-in: These are depth three circuits where fan-in of the top addition gate is assumed to be $k = O(1)$. For $k = 2$, [Shp07] designed a randomized reconstruction algorithm with time complexity quasi-polynomial in $n, d, |\mathbb{F}|$[3]. An important point to note is that when rank (Definition 11) of the input polynomial is $\Omega(\log^2 d)$, their algorithm is proper, i.e. output also has top fan-in 2. This algorithm was generalized in [KS09] to circuits with top fan-in $k = O(1)$, and a deterministic algorithm with time complexity quasi-polynomial in d and $|\mathbb{F}|$ was given. However, unlike [Shp07], their algorithm is improper and output might have much larger top fan-in. In a recent work, [Sin16b] also considered the top fan-in 2 case, but over characteristic zero fields, and rank of input polynomial being $\Omega(1)$. Their algorithm runs in time polynomial in n, d, but their techniques do not work over finite fields. Based on the above, the following questions seem very natural to ask:

- **Q1.** Does there exist a reconstruction algorithm for depth 3 circuits with top fan-in 2 (over a finite field $\mathbb{F}$), whose run-time is polynomial in $\log|\mathbb{F}|$? *This was asked as an open problem in [GKL12] (STOC 2012).*
- **Q2.** Can such an algorithm be fully polynomial time (at-least in high rank case) i.e. runs in time polynomial in n, d and $\log|\mathbb{F}|$? *This will substantially improve the result in [Shp07] (STOC 2007).*

In this paper we resolve both of these questions.

[1] also known as black-box
[2] from here onwards by depth three circuits we mean $\Sigma\Pi\Sigma$ circuits only
[3] n is number of variables in input circuit, d is degree of Π gates and $|\mathbb{F}|$ is size of the underlying field

1.1 Our Results

Let n, d be positive integers and $\mathbb{F}$ be a finite field.

Homogeneity assumption As given in Lemma 3.5 of [DS05], every depth three circuit C of rank r, computing an $n-$variate, degree d polynomial f can be converted into a homogeneous depth three circuit C_{hom} over $\leq n+1$ variables and rank $\leq r+1$, such that it's multiplication gates have in-degree d. Section 1.5 of [Sin16a] implies that black-box access to C_{hom} can be simulated efficiently using black-box access to f and integers n, d. Also there is an efficient algorithm to obtain C from C_{hom}. Hence, from now onwards we only consider homogeneous depth three circuits ($\Sigma\Pi\Sigma(k, n, d, \mathbb{F})$, Definition 8). Also, for any polynomial f, $rank(f)$ (Definition 12) will be the minimum rank of any $\Sigma\Pi\Sigma(2, n, d, \mathbb{F})$ circuit computing it. Here are our results.

Theorem 1 (Low rank reconstruction)**.** *There exists a randomized algorithm which takes as input integers n, d and black-box access to a polynomial f computable by a $\Sigma\Pi\Sigma(2, n, d, \mathbb{F})$ circuit ($5 \leq rank(f) = O(\log^3 d)$), runs in time $(nd^{\log^3 d} \log|\mathbb{F}|)^{O(1)}$ and, with probability $1 - o(1)$, outputs a $\Sigma\Pi\Sigma(k, n, d, \mathbb{F})$ ($k \leq d^{rank(f)}$) circuit computing f.*

Theorem 2 (High rank reconstruction)**.** *There exists a randomized algorithm which takes as input integers n, d and black-box access to a polynomial f computable by a $\Sigma\Pi\Sigma(2, n, d, \mathbb{F})$ circuit ($rank(f) = \Omega(\log^3 d)$), runs in time $(nd \log|\mathbb{F}|)^{O(1)}$ and, with probability $1 - o(1)$, outputs a $\Sigma\Pi\Sigma(2, n, d, \mathbb{F})$ circuit computing f.*

We allow algorithms to query input polynomial at points in a $(nd)^{O(1)}$ sized extension $\mathbb{K}$ of $\mathbb{F}$.

Remarks Here are some remarks on the above results.

- Theorems 1 and 2 completely resolve **Q1**. Therefore we solve an open problem from [GKL12]. Theorem 2 resolves **Q2** in the high rank case ($\Omega(\log^3 d)$) and thus both theorems substantially improve the overall reconstruction time complexity for this circuit class (as compared to [Shp07] and [KS09]).
- When $rank(f) = 1$, the polynomial factors into a product of linear forms and can be reconstructed efficiently using Lemma 4. So only $rank(f) = 2, 3, 4$ are not covered by the algorithms above.
- A crucial component of our proofs is a new structural result, which might be of independent interest. We show that for any polynomial f computable by a $\Sigma\Pi\Sigma(2, n, d, \mathbb{F})$ circuit ($rank(f) \geq 5$), the set of co-dimension 2 subspaces of $\mathbb{F}^n$ on which the "non-linear" part[4] of f vanishes, has size $d^{O(1)}$, and can be computed efficiently. We give a formal statement in Proposition 1.
- In order to prove Theorem 2, we develop an interesting result related to Sylvester Gallai (SG) type configurations (Definition 13) and present it in Proposition 2. We believe it might be of independent interest. Similar results called Quantitative SG theorems are known (Theorem 5.1.2 and Section 5.3 in [Dvi12]). These quantitative versions prove bounds on number of ordinary lines through a point, whereas our theorem considers dimension of the space spanned by the union of ordinary lines through a point.

[4]this is obtained by removing all linear factors of f. See Definition 1

Next, we will state our proposition regarding the number of co-dimension 2 spaces on which the non-linear part of f vanishes. But, first, we formally define what we mean by the non-linear part of a polynomial.

Definition 1 (Linear and Non-linear parts). *Let $f \in \mathbb{F}[\mathbf{x}]$ be a polynomial. We define $Lin(f)$, called the linear part of f, to be the product (with multiplicity) of all affine polynomials dividing f and $NonLin(f)$, called the non-linear part of f as $NonLin(f) = \frac{f}{Lin(f)}$*[5]*.*

Next, we define what we mean by vanishing of $NonLin(f)$ on a co-dimension 2 subspace. In order to do so, we need some more notation. For a set of linear forms $\ell_1, \ldots, \ell_k$, we represent the subspace $\{a \in \mathbb{F}^n : \ell_1(a) = \ldots = \ell_k(a) = 0\}$ by $\mathbb{V}(\ell_1, \ldots, \ell_k)$. It's easy to see that for any co-dimension k subspace $W \subset \mathbb{F}^n$, there exists k linearly independent linear forms $\ell_1, \ldots, \ell_k$ such that $W = \mathbb{V}(\ell_1, \ldots, \ell_k)$. For a subset of variables $x_{i_1}, \ldots, x_{i_k}$, by $f_{|x_{i_1}=\alpha_{i_1}, \ldots, x_{i_k}=\alpha_{i_k}}$, we denote the polynomial obtained on setting $x_{i_1} = \alpha_{i_1}, \ldots, x_{i_k} = \alpha_{i_k}$ in f. Now, we are ready to give the definition.

Definition 2. *Let $f \in \mathbb{F}[x_1, \ldots, x_n]$. For any co-dimension 2 space $W \subset \mathbb{F}^n$, we say that f vanishes on W, if, for linearly independent linear forms ℓ_1, ℓ_2 such that $W = \mathbb{V}(\ell_1, \ell_2)$, and isomorphism $\Phi : \mathbb{F}[x_1, \ldots, x_n] \to \mathbb{F}[x_1, \ldots, x_n]$ that maps $\ell_1 \mapsto x_1, \ell_2 \mapsto x_2$, the polynomial $\Phi(f)_{|x_1=0, x_2=0}$ is identically zero. It's easy to see that this definition is well defined, i.e. if we take some other linear forms ℓ_1', ℓ_2' such that $W = \mathbb{V}(\ell_1', \ell_2')$ and some other isomorphism Φ' which maps $\ell_1' \mapsto x_1, \ell_2' \mapsto x_2$, then,*

$$\Phi(f)_{|x_1=0, x_2=0} = 0 \Leftrightarrow \Phi'(f)_{|x_1=0, x_2=0} = 0.$$

For any polynomial f, we define $\mathcal{S}(f)$ to be the set of all co-dimension 2 sub-spaces $W \subset \mathbb{F}^n$ such that f vanishes on W.

Given the above definition, we now present our proposition about $\mathcal{S}(NonLin(f))$.

Proposition 1. *Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ be a polynomial computable by a $\Sigma\Pi\Sigma(2, n, d, \mathbb{F})$ circuit ($rank(f) \geq 5$). The following are true.*

1. $|\mathcal{S}(NonLin(f))| \leq 3d^7$.
2. *There exists a randomized algorithm that takes as input black-box access to f along with integers n, d, runs in time $(nd \log |\mathbb{F}|)^{O(1)}$ and, outputs a set $\mathcal{S}$ (of size $\leq 3d^7$) containing tuples of independent linear forms in $\mathbb{F}[x_1, \ldots, x_n]$ such that with probability $1 - o(1)$,*

$$\{\mathbb{V}(\ell_1, \ell_2) : (\ell_1, \ell_2) \in \mathcal{S}\} = \mathcal{S}(NonLin(f)).$$

Before moving on to explaining our techniques required in proofs of Theorems 1 and 2, we state the proposition about ordinary lines and the space spanned by them, that was mentioned in remarks following the theorems. To do so, we give a few more definitions.

Definition 3 (Proper set, Section 5.3, [Dvi12]). *We call a set of points $v_1, \ldots, v_m \in \mathbb{F}^n$* proper *if no two points are a constant multiple of each other and the zero point is not in the set (i.e. it is a subset of the projective space).*

[5] $Lin(f), NonLin(f)$ are unique up to scalar factors which are constrained such that $f = Lin(f) \times NonLin(f)$.

Definition 4 (Ordinary line, Section 5.1, [Dvi12]). *Let $\mathcal{S} \subset \mathbb{F}^n$ be a proper set. For any $t \in \mathbb{F}^n$ and $s \in \mathcal{S}$, such that $t \notin sp\{s\}$, the vector space $sp\{t,s\}$ is called an ordinary line from t into $\mathcal{S}$, if and only if $sp\{s,t\} \cap \mathcal{S} \subseteq \{s,t\}$. Define $\mathcal{O}(t,\mathcal{S})$ to be the set of ordinary lines from t into $\mathcal{S}$.*

Here is the main proposition about space spanned by union of all these lines.

Proposition 2. *Let $\mathcal{S} \subset \mathbb{F}^n$ be a proper set (Definition 3) and $\mathcal{T} \subset \mathbb{F}^n$ be any linearly independent set of size $\geq \log|\mathcal{S}| + 2$. Then there exists $t \in \mathcal{T}$, such that union of all elements of $\mathcal{O}(t,\mathcal{S})$ spans a high dimensional space. More precisely,*

$$dim(\sum_{W \in \mathcal{O}(t,\mathcal{S})} W) \geq \frac{dim(sp(\mathcal{S}))}{\log|\mathcal{S}| + 2}.$$

1.2 Ideas and analysis of main algorithms

The algorithms mentioned in Theorems 1, 2 and Proposition 1 are given in Algorithms 2, 3 and 7 respectively. In this section we discuss key technical ideas required for proving these results. Missing details are supplied in the subsequent sections. As described in Definition 11, write $f = G \times (T_1 + T_2)$ where G, T_1, T_2 are product of linear forms and $gcd(T_1, T_2) = 1$. We know that

$$Lin(f) \times NonLin(f) = f = G \times (T_1 + T_2).$$

1.2.1 Theorem 1: Key ideas for Algorithm 2

The algorithm mentioned in Theorem 1 is presented in Algorithm 2 and it's correctness/complexity is discussed in Section 3. We describe the main ideas now. Since $NonLin(f)$ has no linear factors and $Lin(f), G$ are product of linear forms, $NonLin(f)$ divides $T_1 + T_2$ implying that $NonLin(f) = h(y_1, \dots, y_r)$, for some polynomial h over $\mathbb{F}$ and independent linear forms $y_1, \dots, y_r$ spanning the set of linear factors of $T_1 \times T_2$ (here $r = rank(f)$). We also note that $NonLin(f)$ is non-constant, otherwise f would become a product of linear forms and it's rank as a $\Sigma\Pi\Sigma(2, n, d, \mathbb{F})$ circuit would not be >= 5. Using Algorithm 1, with high probability, we get black-box access to $NonLin(f)$ and it's degree t. If we also had access to

(a) the integer, $rank(f)$, and

(b) a $d^{O(1)}$ sized set $\mathcal{L}$ of linear forms which contained such independent forms $y_1, \dots, y_r$, then,

we could just iterate over all r sized subsets $\{y_1, \dots, y_r\}$ of $\mathcal{L}$ and using deterministic multivariate black-box interpolation (Lemma 6) compute polynomial $h(y_1, \dots, y_r)$ as a sum of degree t monomials in $y_1, \dots, y_r$. This interpolation gives a $\Sigma\Pi\Sigma(t^r, n, t, \mathbb{F})$ circuit computing $NonLin(f)$. We can then multiply all linear factors of $Lin(f)$ (also obtained using Algorithm 1) to all multiplication gates of this circuit resulting in a $\Sigma\Pi\Sigma(t^r, n, d, \mathbb{F})$ circuit for f, thereby completing the reconstruction. So we only need to argue about the required access described above.

We do not have direct access to $rank(f)$, but we know that $rank(f) = O(\log^3 d)$. Therefore, we can just try all values of r from 1 to $O(\log^3 d)$. To get access to the set $\mathcal{L}$ described above, we crucially use the results in Proposition 1. It guarantees that the set $\mathcal{S}(NonLin(f))$ of co-dimension 2 subspaces on which $NonLin(f)$ vanishes, has atmost $3d^7$ elements and also provides an efficient algorithm (Algorithm 7) to construct a set $\mathcal{S}$ which contains tuples of linear forms representing each co-dimension 2 space in $\mathcal{S}(NonLin(f))$. The following definition of $\mathcal{L}$ satisfies our requirement described above.

$$\mathcal{L} = \{\ell_1 : \exists \ell_2 \text{ such that } (\ell_1, \ell_2) \in \mathcal{S} \text{ or } (\ell_2, \ell_1) \in \mathcal{S}\}$$

Access to this set can be created using $\mathcal{S}$, by just going over all elements (tuples) in $\mathcal{S}$ and collecting all linear forms that are seen in any of them. Also size of $|\mathcal{L}|$ is seen to be $d^{O(1)}$ since size of $\mathcal{S}$ is $d^{O(1)}$ by Proposition 1. In Lemma 3, we show that it contains an independent set $\{y_1, \ldots, y_r\}$ of linear forms that spans the set of linear factors of $T_1 \times T_2$. Basically, for any of the linear forms ℓ dividing $T_1 \times T_2$, we create two distinct co-dimension 2 sub spaces contained in the hyperplane $\mathbb{V}(\ell)$, such that $NonLin(f)$ vanishes on both of them. Since these subspaces are distinct, there is a unique such hyperplane and it can be constructed using them. Thus we have access to a $d^{O(1)}$ sized set that satisfies our requirements.

At the end of our algorithm, we can perform a randomized polynomial identity test to check whether the reconstructed circuit matches the input polynomial or not. We are able to guarantee that with probability $1 - o(1)$, no incorrect reconstruction is returned while searching for r and $y_1, \ldots, y_r$. At the same time, for the correct value of r and linear forms $y_1, \ldots, y_r$ as described above, with probability $\geq 1 - o(1)$, we will recover the correct circuit. Therefore, overall with probability $\geq 1 - o(1)$ our reconstruction will be correct. The algorithm we described above takes $(nd^{\log^3 d} \log |\mathbb{F}|)^{O(1)}$ time. Complete details can be found in Section 3.

Comparison with algorithm in [Shp07] The broad idea for low rank[6] reconstruction given in Algorithm 3 of [Shp07] is similar to ours. However, their algorithm runs in time quasi-polynomial in n, d and $|\mathbb{F}|$. The main reason for this time complexity is that they search for the required basis $\{y_1, \ldots, y_r\}$ of linear forms (Step 2 of Algorithm 3 in [Shp07]) by essentially iterating over the entire set of linear forms in $O(\log^2 d)$ many variables. This makes their algorithm quasi-polynomial time with respect to field size $|\mathbb{F}|$, since this set of linear forms contains $|\mathbb{F}|^{O(\log^2 d)}$ elements. As described above, our algorithm performs this search more efficiently by searching within the $d^{O(1)}$ sized set $\mathcal{L}$ that can itself be efficiently constructed using Algorithm 7. This leads to only a polynomial time dependence on $\log |\mathbb{F}|$ which is more ideal as $O(\log |\mathbb{F}|)$ bits can represent each scalar in the output circuit.

1.2.2 Theorem 2: Key ideas for Algorithm 3

The algorithm mentioned in Theorem 2 is presented in Algorithm 3. It's correctness and time complexity are discussed in Section 4. We describe the main ideas now. Our algorithm first tries to reconstruct the circuit assuming a corner case where one of the two polynomials T_1, T_2 is a power of some linear form (up to scalar multiplication). If the reconstruction succeeds it is returned, otherwise we go to the general case wherein linear factors of any of the T_is spans at least a two dimensional space. Both these cases utilize the set of "candidate linear forms" which we define next. In order to do so we first need to define what it means for a polynomial to factorize into non-zero linear forms on a co-dimension 1 subspace.

Definition 5. *Let $f \in \mathbb{F}[x_1, \ldots, x_n]$. For any co-dimension 1 space $W \subset \mathbb{F}^n$, we say that f factorizes into non-zero linear forms on W, if, for linear form ℓ_1 such that $W = \mathbb{V}(\ell_1)$, and isomorphism $\Phi : \mathbb{F}[x_1, \ldots, x_n] \to \mathbb{F}[x_1, \ldots, x_n]$ that maps $\ell_1 \mapsto x_1$, the polynomial $\Phi(f)_{|x_1=0}$ is a non-zero product*

[6] their low rank case assumes $rank(f) = O(\log^2 d)$. we assume $rank(f) = O(\log^3 d)$.

of linear forms in $\mathbb{F}[x_2, \ldots, x_n]$. It's easy to see that the this definition is well defined, i.e. if we take some other linear forms ℓ_1' such that $W = \mathbb{V}(\ell_1')$ and some other isomorphism Φ' which maps $\ell_1' \mapsto x_1$, then,

$$\Phi(f)_{|x_1=0} \text{ is a non-zero product of linear forms} \Leftrightarrow \Phi'(f)_{|x_1=0} \text{ is a non-zero product of linear forms}$$

Now we define $\mathcal{L}(f)$, the set of candidate linear forms. This is a set of linear forms which should be thought of as points in the projective space therefore no two elements of $\mathcal{L}(f)$ will be scalar multiples. Here is the definition.

Definition 6 (Candidate linear forms)**.** *Let f be a polynomial in $\mathbb{F}[x_1, \ldots, x_n]$. Let ℓ be a linear form and $W = \mathbb{V}(\ell)$. Suppose the following hold.*

1. *f factorizes into non-zero linear forms on W, and*
2. *There exist linear forms ℓ_1, ℓ_2 with ℓ, ℓ_1, ℓ_2 being linearly independent, such that f vanishes on co-dimension 2 subspaces $\mathbb{V}(\ell, \ell_1), \mathbb{V}(\ell, \ell_2)$.*

Then, ℓ, considered as a point in the projective space, is called a candidate linear form. The set of candidate linear forms, denoted as $\mathcal{L}(f)$, is then the set containing unique multiples of these ℓs and is therefore considered as a subset of the projective space. Using this definition the following observation about the size of $\mathcal{L}(f)$ can be made.

Observation 1. *For any polynomial f, $|\mathcal{L}(f)| \le |\mathcal{S}(f)|^2$.*

Now we are ready to explain the key ideas in our algorithms. We first explain the techniques required by our corner case and then go to the more general case.

Corner case - One of T_1, T_2 is power of a linear form: This is a corner case of our algorithm and is tried before we proceed to reconstructing the general case. Formal statement is provided in Lemma 16 and corner case reconstruction algorithm is given in Algorithm 5. We sketch the idea here. Our polynomial looks like

$$Lin(f) \times NonLin(f) = f = G \times (T_1 + T_2).$$

If one of T_1, T_2 is power of a linear form, then we observe that the structure of this circuit gets heavily constrained. In particular, we can show that $Lin(f) = G$ and $NonLin(f) = T_1 + T_2$. This is proved in Claim 1. The basic idea is that, if $T_1 + T_2$ has a non trivial factor ℓ, span of any any linear factor of T_1 and ℓ will contain some linear factor of T_2. This can be used to show that dimension of $sp\{\text{linear form } \ell : \ell \mid T_1\}$ and $sp\{\text{linear form } \ell : \ell \mid T_2\}$ can at most differ by 1, which is contradictory to what we have assumed in this case. So having observed this, now, using Algorithm 1 we can get black-box access to $T_1 + T_2$, it's degree t and the entire list of linear factors (with multiplicity) of G. Let's assume that for some $i \in [2]$, T_i is power of some linear form. If we also had access to

(a) a linear form ℓ_1 dividing T_i, and

(b) a $d^{O(1)}$ sized set $\mathcal{X}$ of scalars such that $T_i = \delta \ell_1^t$ for some $\delta \in \mathcal{X}$, then we could just go over all scalars $\delta \in \mathcal{X}$ and try to factorize black-box of $NonLin(f) - \delta\ell_1^t$ using Algorithm 1. If factorization gives all linear factors, we would have obtained a $\Sigma\Pi\Sigma(2, n, t, \mathbb{F})$ circuit for $NonLin(f) = T_1 + T_2$. Using this and the linear factors in $Lin(f) = G$, we can obtain a $\Sigma\Pi\Sigma(2, n, d, \mathbb{F})$ circuit for f. So now we only need to argue about the required access above.

In Claim 2, we show that a linear factor ℓ_1 of T_i belongs to the the set of candidate linear forms $\mathcal{L}(NonLin(f))$ that we defined above. To see this, assume $T_i = \delta\ell_1^t$ for some scalar δ and observe that on $\mathbb{V}(\ell_1)$, $NonLin(f) = T_1 + T_2$ will factorize into non-zero linear forms. Therefore ℓ_1 satisfies the first condition of being a "candidate linear form" (Definition 6). The second condition is also satisfied since linear factors of T_{3-i} would span a high dimensional space (as $rank(f) = \Omega(\log^3 d)$ and T_i only has one linear factor), giving us linear factors ℓ_2, ℓ_3 of T_{3-i} such that ℓ_1, ℓ_2, ℓ_3 are linearly independent providing us with two co-dimension 2 spaces $\mathbb{V}(\ell_1, \ell_2), \mathbb{V}(\ell_1, \ell_3)$ on which $NonLin(f) = T_1 + T_2$ vanishes. This set $\mathcal{L}(NonLin(f))$ is $d^{O(1)}$ sized, as implied by Observation 1 and Proposition 1. We also give an efficient algorithm in Algorithm 4 to construct this set. So we can just iterate over it and search for ℓ_1. Now we discuss how to get access to the set $\mathcal{X}$ containing δ. Recall that,

$$NonLin(f) = \delta\ell_1^t + T_{3-i}.$$

We restrict this polynomial to $\mathbb{V}(\ell_1)$, and obtain two linearly independent factors ℓ_2, ℓ_3 of the restriction of T_{3-i} to $\mathbb{V}(\ell_1)$ by factorizing using Algorithm 1 (two independent factors will exists since $rank(f) = \Omega(\log^3 d)$ and T_i only has one linear factor). For simplicity, let's map $\ell_1 \mapsto x_1, \ell_2 \mapsto x_2, \ell_3 \mapsto x_3$. Therefore, our polynomial would have the following form,

$$NonLin(f) = \delta x_1^t + (x_2 - \beta x_1)(x_3 - \gamma\ell_1)T'_{3-i},$$

for some scalars β, γ and product of linear forms T'_{3-i}. The above form implies that the polynomial depends on x_3, but becomes independent on plugging $x_2 = \beta x_1$. This is the crucial idea we use. We first set all other variables i.e. $x_4, \ldots, x_n$ to random values in $\mathbb{F}$ and using multivariate interpolation (Lemma 6), find monomial representation of our polynomial as a degree t polynomial in $\mathbb{F}[x_1, x_2, x_3]$. Then we solve for a fresh variable β such that setting $x_2 = \beta x_1$, makes this polynomial independent of x_3. This can be done by collecting all univariate coefficient polynomials from monomials containing x_3 and jointly setting them to 0. This system of polynomial equations has at most $d^{O(1)}$ many solutions since all polynomials are univariate and of degree $\leq d^{O(1)}$. Therefore, all these solutions can be computed using the algorithm given in Lemma 1. This will give a set of solutions for β and just plugging them back into the coefficient polynomial of x_1^t would give the set $\mathcal{X}$ containing δ. This creates our access for $\mathcal{X}$.

At the end of the reconstruction using polynomial identity testing algorithm given in Lemma 4, we deterministically check whether the reconstruction is correct or not. So for values of ℓ_1 and δ, where the reconstructed circuit was not correct, we don't output anything and for the right values of ℓ_1, δ, by the algorithm described above, we will correctly reconstruct the circuit and output it. The algorithm we described above takes $(nd \log |\mathbb{F}|)^{O(1)}$ time. Complete details can be found in Section 4.2.

General case - Both T_1, T_2 have at least 2 independent linear factors: This is the more general case of our algorithm and is tried after the above mentioned corner case fails to provide a reconstruction. Our algorithm tries to find an $\Omega(\log d)$ sized set of linear forms such that all linear forms in this set divide the same T_i. Once such a set is found we use it to reconstruct all linear forms dividing $G \times T_{3-i}$ and using this the entire circuit. We break down our key ideas below.

- We first explain, how one can complete the reconstruction given access to such a set. Formal statement is given in Lemma 19 and algorithm is provided in Algorithm 6. The basic idea is as follows. For simplicity, without loss of generality, we assume the independent set of linear forms is just the set of variables $x_1, \ldots, x_t$ where $t = \Omega(\log d)$. This can be achieved by making an invertible transformation on the given linear forms. Further assume that all of these divide T_1. Thus our polynomial looks like,

$$Lin(f) \times NonLin(f) = f = G \times (x_1 \ldots x_t T_1' + T_2)$$

where T_1' is also a product of linear forms and $gcd(T_1', T_2) = 1$. Without loss of generality we can also assume that no x_i divides f (i.e. G) since we can divide f by largest power of all the x_i. The idea is to construct all linear factors of $G \times T_2$ by first computing all linear factors of $(G \times T_2)_{|x_i=0}$ for $i \in [t]$ and then gluing these factorizations together. We explain the key steps next.

  - First step is to construct the multi-sets of factors of $(G \times T_2)_{|x_i=0}$. These multi-sets can be constructed by computing black-box for $f_{|x_i=0}$ and factorizing them using Algorithm 1. Since all x_i's divide T_i, all these polynomials will completely factorize into linear factors and our multi-sets containing factors of $(G \times T_2)_{|x_i=0}$ will have the same number (i.e. $deg(f)$) of elements.
  - Now to glue these multisets together, our algorithm exactly matches the one given in Algorithm 5 in [Shp07]. The idea is to find a linear form ℓ_1 dividing $(G \times T_2)_{|x_1=0}$ (with multiplicity say k), and an integer $2 \leq i \leq t$ such that there are exactly k linear factors (could be multiples of each other) ℓ_i of $(G \times T_2)_{|x_i=0}$ such that $\ell_{1|x_i=0}$ and $\ell_{i|x_1=0}$ are scalar multiples. ℓ_1 is then glued with all the ℓ_is by comparing coefficients and k glued linear forms (can be multiples of each other) are constructed. These will divide $G \times T_2$. Then ℓ_1 (with all its multiplicity) and all the ℓ_is are removed from their respective multisets and this process is repeated until the multisets become empty. They show that when the multisets are non-empty, such ℓ_1 and i always exist. If not, then they show that a lower bound (given in Theorem 33 of [Shp07]) on length of linear 2-query locally decodable codes gets violated. At the end, all linear factors (with multiplicity) of $G \times T_2$ would have been constructed. Complete construction is provided inside proof of Theorem 29 in [Shp07]. For cleaner presentation, we do not repeat it here.
  - Now we can factorize the black-box for $f - G \times T_2$ and recover all linear factors of $G \times T_1$ thereby completing[7] the reconstruction. However, we note that the above step only recovers linear factors of $G \times T_2$ up to scalar multiples. So we still need to find a scalar which when multiplied to product of all recovered linear factors would correctly give $G \times T_2$. But this part is rather simple. We can just restrict all linear forms in our computed multi-set to $x_1 = 0$ and then compare with the multi-set of linear forms that divided $(G \times T_2)_{|x_1=0}$ which we had already computed above, thereby giving us the appropriate scalar factor. Once this is found, we can construct a $\Sigma\Pi\Sigma(2, n, d, \mathbb{F})$ circuit as described above. Finally using polynomial identity test in Lemma 4, we can check whether this circuit correctly computes f or not, and only output a correct circuit.

[7] note we also have to add back all powers of x_i that were removed earlier.

- Now, we come back to our process of finding the linearly independent set utilized above. This is where our set of candidate linear forms $\mathcal{L}(NonLin(f))$ comes handy again. As mentioned earlier Algorithm 4 efficiently constructs this set. Now, we will explain how this set can be used to construct the required linearly independent set (actually a small collection of such sets) required by our algorithm.
  - In order to do so we first observe that in this general case, the set $\mathcal{L}(NonLin(f))$ approximates the set of linear factors of $T_1 \times T_2$ really closely. We make this concrete in Claim 3 by showing that linear factors of $T_1 \times T_2$ that belong to $\mathcal{L}(NonLin(f))$ span a space of dimension $\geq rank(f) - 2$, and elements of $\mathcal{L}(NonLin(f))$ which do not divide $T_1 \times T_2$ span at most a $\log d + 2$ dimensional space, thereby establishing closeness of these sets. We exploit this fact and search for the required independent set of factors of $T_1 \times T_2$ inside $\mathcal{L}(NonLin(f))$. However more steps remain to make this search efficient.
  - In Part 1 of Lemma 21, using Proposition 2, we show existence of a linear form $\ell \in \mathcal{L}(NonLin(f))$ such that $\ell \mid T_1 \times T_2$ and the union of ordinary lines from ℓ into set $\mathcal{L}(NonLin(f))$ is $(\Omega(\log^2 d))$ dimensional. Next, in Part 2 of Lemma 21, using the closeness described above, we create a collection[8] of $\Omega(\log d)$ sized linearly independent subsets of $\mathcal{L}(NonLin(f))$ such that one of them $(\mathcal{B})$ satisfies two properties. (a) It only contains linear factors of $T_1 \times T_2$, and (b) for all $\ell' \in \mathcal{B}$, $sp\{\ell, \ell'\}$ does not contain any other linear factor of $T_1 \times T_2$, and any linear factor of $T_1 + T_2$. However, it might still contain factors of both T_1 and T_2 which we separate next.
  - Using properties of $\mathcal{B}$, in Part 3 of Lemma 21, we show that for $\ell' \in \mathcal{B}$, $NonLin(f)$ vanishes on $\mathbb{V}(\ell, \ell')$ if and only if ℓ_1, ℓ_2 divide different T_i. We propose to use this equivalence as a test and split $\mathcal{B}$ into two parts, such that all linear forms in each part divide the same T_i. At least one of these would be $\Omega(\log d)$ in size and would give us the required linearly independent set. Full details can be found in Part 3 of Lemma 21.
  - So all that remains now is to convert the above construction into an efficient algorithm. We search for ℓ in $\mathcal{L}(NonLin(f))$. For each candidate ℓ, we construct the collection of sets that contains $\mathcal{B}$. Then we go through each set in this collection and for each candidate set $\mathcal{B}$, apply our test to divide it into two parts $\mathcal{U}, \mathcal{V}$. Then we pass the larger set to the previous algorithm (details in Algorithm 6) to reconstruct the circuit. In the end, we use deterministic polynomial identity test to reject incorrect constructions. The existence of $\ell, \mathcal{B}$ and the test above, make sure that for the correct choices, we will output the correct circuit.

Comparison with algorithm in [Shp07] As described above, if we have access to an $\Omega(\log d)$ sized set of linear forms such that all of them divide the same T_i, our algorithm exactly matches the one given in Algorithm 5 of [Shp07]. The main difference[9] is in the way such an independent set is created. In Steps 1 and 2 of Algorithm 4 in [Shp07], they iterate over all possible $\Omega(\log d)$ sized sets of linear forms inside an $\Omega(\log^2 d + \log^2 n)$ sized random subspace of $\mathbb{F}^n$. Such a brute force search considers $|\mathbb{F}|^{\Omega(\log d(\log^d + \log^2 n))}$ many sets leading to a quasi polynomial time complexity in n, d and $|\mathbb{F}|$. Using $\mathcal{L}(NonLin(f))$, in Lemma 21 we are able to create a small collection of sets of independent linear forms, such that at least one set in this collection has size $\Omega(\log d)$ and comprises

[8] size of collection is between $\Omega(\log d)$ and $rank(f)$.

[9] also we assume rank to be $\Omega(\log^3 d)$ whereas [Shp07] assumes it to be $\Omega(\log^2 d)$ for high rank reconstruction

of linear forms all of which divide the same T_i. Construction of this collection has required new structural techniques from Proposition 1 and Proposition 2. Searching through this small collection and rejection of incorrect reconstructions by a deterministic polynomial identity test, lead to an overall running time of $(nd\log|\mathbb{F}|)^{O(1)}$. So we get a huge improvement in the time complexity compared to [Shp07].

2 Preliminaries

2.1 Notations and definitions

Throughout the paper $[n]$ will denote the set $\{1,\dots,n\}$, $[m,n]$ will denote the set $\{m,m+1,\dots,n-1,n\}$ and $\mathbb{F}$ will denote a finite field. We use calligraphic letters like $\mathcal{B},\mathcal{P},\mathcal{Q},\mathcal{R},\mathcal{S},\mathcal{T},\mathcal{X}$ to denote sets. Bold small letters $\mathbf{x},\mathbf{y},\mathbf{u}$ are used to represent column vectors or tuples of variables. Unless otherwise specified, $\mathbf{x}$ will denote the tuple $(x_1,\dots,x_n)$. Bold capital letters $\mathbf{A},\mathbf{B}$ are used to represent matrices. $\mathbb{F}[\mathbf{x}]$ denotes the ring of polynomials in variables $\mathbf{x}=(x_1,\dots,x_n)$ with coefficients in field $\mathbb{F}$. Capital letters like $G,H,T_1,T_2,S_1,S_2,U,U_i$ are either used to denote polynomials that are a product of linear forms. Small letters f,g,h,u,ℓ are also used to denote polynomials and linear forms. Let g,f be any two polynomials, then, g divides f is denoted by $g \mid f$ and g does not divide f is denoted by $g \nmid f$.

Definition 7 (Depth 3 circuit, $\Sigma\Pi\Sigma$)**.** *A depth 3 circuit is a layered arithmetic circuit with three layers of nodes labelled by arithmetic operations, defined on a finite number of variables. First and third (Σ) layers have addition nodes and second (Π) layer has multiplication nodes. Top layer has a single addition node.*

Definition 8 (Homogeneous Depth 3 circuit, $\Sigma\Pi\Sigma(k,n,d,\mathbb{F})$)**.** *A $\Sigma\Pi\Sigma(k,n,d,\mathbb{F})$ circuit is a depth three circuit such that the first (Σ) layer computes linear forms on n variables, there are k multiplication nodes at the second (Π) layer all having in-degree d, and the addition node at third(Σ) layer can only have incoming edges from the k multiplication nodes at second layer. Any circuit belonging to this class naturally computes an $n-$variate polynomial $f=M_1+\dots+M_k$, where $M_i, i\in[k]$ are product of linear forms computed at the multiplication gates and $deg(M_1)=\dots=deg(M_k)=d$.*

Definition 9 (Simple $\Sigma\Pi\Sigma(k,n,d,\mathbb{F})$ circuit)**.** *Let C be a $\Sigma\Pi\Sigma(k,n,d,\mathbb{F})$ circuit computing polynomial $f=M_1+\dots+M_k$ as described in Definition 8. We say that C is simple if $gcd(M_1,\dots,M_k)=1$.*

Definition 10 (Minimal $\Sigma\Pi\Sigma(k,n,d,\mathbb{F})$ circuit)**.** *Let C be a $\Sigma\Pi\Sigma(k,n,d,\mathbb{F})$ circuit computing the polynomial $f=M_1+\dots+M_k$ as described in Definition 8. We say that C is minimal if no proper sub collection of polynomials $M_1,\dots,M_k$ sums to zero.*

Definition 11 (Rank of $\Sigma\Pi\Sigma(2,n,d,\mathbb{F})$ circuit, Section 1.3 in [Shp07])**.** *Let C be a $\Sigma\Pi\Sigma(2,n,d,\mathbb{F})$ circuit computing the polynomial $f=M_1+M_2$ as described in Definition 8. If $G=gcd(M_1,M_2)$, then f can be written as $f=G\times T_1+G\times T_2$ where G,T_1,T_2 are product of linear forms with $gcd(T_1,T_2)=1$.* Rank *of C is then defined as*

$$rank(C)=dim(sp\{linear\ form\ \ell\in\mathbb{F}[\mathbf{x}]:\ell\mid T_1\times T_2\})$$

Definition 12 (Rank of polynomial)**.** *For any polynomial $f\in\mathbb{F}[\mathbf{x}]$ computable by a $\Sigma\Pi\Sigma(2,n,d,\mathbb{F})$ circuit, it's rank, called $rank(f)$ is defined as the minimum of $rank(C)$ over all $\Sigma\Pi\Sigma(2,n,d,\mathbb{F})$ circuits computing f.*

Definition 13 (Sylvester Gallai (SG) configuration, Definition 5.3.1, [Dvi12]). *A proper set $\mathcal{S} = \{s_1, \dots, s_m\} \subset \mathbb{F}^n$ is called an SG configuration if for every $i \neq j \in [n]$, $\exists k \in [n] \setminus \{i, j\}$ with s_i, s_j, s_k linearly dependent.*

2.2 Known results

In this subsection, we list a few known results that are used in the paper.

Lemma 1 (Solving polynomial equations, Implied from [Ier89, Laz01]). *There is a randomized algorithm that takes as input n variate polynomials $f_1, \dots, f_m$ each of degree $\leq d$. If the system of equations defined by setting all these polynomials simultaneously to zero, has finitely many solutions in $\bar{\mathbb{F}}$ and all solutions are in $\mathbb{F}^n$, then the algorithm computes all solutions with probability $1 - exp(-mnd \log|\mathbb{F}|)$. Running time of the algorithm is $(md^n \log|\mathbb{F}|)^{O(1)}$.*

Lemma 2 (Schwartz Zippel Lemma, [Sch80, Zip79]). *Let $p(x_1, \dots, x_n)$ be a polynomial of total degree d such that it is not identically zero. Let $S \subset \mathbb{F}$ be any finite set. For $s_1, \dots, s_n$ picked independently and uniformly from S,*

$$Pr[p(s_1, \dots, s_n) = 0] \leq \frac{d}{|S|}.$$

This immediately gives the following randomized polynomial identity test.

Lemma 3 (Randomized polynomial identity test, Section 1, Lemma 1.2 in [Sax09]). *There exists a randomized algorithm that takes as input integer n and black-box access to a degree d, $n-$variate polynomial f with coefficients in $\mathbb{F}_q$, runs in time $(nd\log q)^{O(1)}$ and outputs either 'yes' or 'no' such that,*

$$\begin{array}{ll} \textit{output is 'yes'} & \textit{if } f \equiv 0 \\ Pr[\textit{output is 'no'}] \geq 1 - o(1) & \textit{if } f \not\equiv 0 \end{array}$$

Lemma 4 ($\Sigma\Pi\Sigma(k, n, d, \mathbb{F})$ deterministic polynomial identity test, Theorem 1 in [SS11]). *There exists a deterministic algorithm that takes as input black-box access to a degree d, $n-$variate polynomial f computable by a $\Sigma\Pi\Sigma(k, n, d, \mathbb{F})$ circuit, runs in time $(nd^k \log|\mathbb{F}|)^{O(1)}$ and, outputs 'yes' if $f \equiv 0$ and 'no' if $f \not\equiv 0$.*

Lemma 5 ($\Sigma\Pi\Sigma(k, n, d, \mathbb{F})$ Rank bound, Theorem 1.7 in [SS13]). *Let C be a $\Sigma\Pi\Sigma(k, n, d, \mathbb{F})$ circuit, over an arbitrary field $\mathbb{F}$, that is simple, minimal and zero. Then, $rank(C) < 3k^2 + \frac{k^2}{4} \log d$.*

Lemma 6 (Black-box multivariate polynomial interpolation, Theorem 11 in [KS01]). *Let n, m, d be parameters and $\mathbb{F}$ be a finite field. There exists a deterministic algorithm that runs in time $(nmd \log|\mathbb{F}|)^{O(1)}$, and outputs a set S of points in $\mathbb{F}^n$, such that given black-box access to any polynomial $f \in \mathbb{F}[x_1, \dots, x_n]$ with at most m monomials, the coefficients of all monomials can be recovered in $(nmd \log|\mathbb{F}|)^{O(1)}$ time using evaluations from the set $\{f(s) : s \in S\}$.*

Lemma 7 (Effective Hilbert irreducibility / Quantitative Bertini theorem, Corollary 2 [Kal91], Remarks 11.5.33, 11.5.66 [MP13], Theorem 1.1 [KSS14]). *Let $\mathbb{F}$ be a perfect field and $g(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ be a degree d irreducible polynomial. Pick tuples, $\mathbf{a} = (a_2, \dots, a_n)$, $\mathbf{b} = (b_1, \dots, b_n)$, $\mathbf{c} = (c_1, \dots, c_n)$ such that every a_i, b_j, c_k is chosen uniformly randomly and independently from a set $S \subset \mathbb{F}$. Consider the bi-variate restriction*

$$\hat{g}(X, Y) = g(X + b_1 Y + c_1, a_2 X + b_2 Y + c_2, \dots a_n X + b_n Y + c_n)$$

Then,

$$P[(\mathbf{a},\mathbf{b},\mathbf{c}) \in S^{n-1} \times S^n \times S^n : \hat{g}(X,Y) \textit{ is not irreducible }] \le \frac{2d^4}{|S|}$$

Lemma 8 (Black-box multivariate polynomial factorization, [KT90])**.** *There exists a randomized algorithm that takes as input black-box access to a degree d, $n-$variate polynomial f with coefficients in $\mathbb{F}$, runs in time $(nd\log|\mathbb{F}|)^{O(1)}$ and outputs black-box access to polynomials $f_1,\dots,f_m$ $(m \le d)$ along with integers $e_1,\dots,e_m$ such that,*

$$Pr[f \equiv f_1^{e_1}\dots f_m^{e_m} \bigwedge f_1,\dots,f_m \textit{ are irreducible}] \ge 1-o(1).$$

Corollary 1 (Decomposition into linear and non-linear factors)**.** *There exists a randomized algorithm that takes as input black-box access to a degree d, $n-$variate polynomial f with coefficients in $\mathbb{F}$, runs in time $(nd\log|\mathbb{F}|)^{O(1)}$ and outputs a list $\{\ell_1,\dots,\ell_s\}$ $(s \le d)$ of affine forms along with black-box access to a polynomial $NonLin(f)$ such that,*

$$Pr[f \equiv l_1\dots l_s NonLin(f) \bigwedge NonLin(f) \textit{ has no linear factors}] \ge 1-o(1).$$

Proof. We give the algorithm below. Correctness and time complexity proofs are pretty straightforward using Lemma 8 and Lemma 3.

Algorithm 1 Decomposition into linear and non-linear factors

Input - Black-box access to polynomial f, integers n,d.
Output - List of affine forms L and black-box access to polynomial $NonLin(f)$.

1. Using algorithm in Lemma 8 on black-box computing f, obtain black-box access to polynomials $f_1,\dots,f_m$ along with integers $e_1,\dots,e_m$. Initialize lists $L,B \leftarrow \phi$.

2. For every $i \in [m]$, construct linear form $\ell_i = \sum_{j=1}^{n}(f_i(\mathbf{e_j}) - f_i(\mathbf{0}))x_j + f_i(\mathbf{0})$, where $\mathbf{e_j} \in \mathbb{F}^n$ is the vector with 1 in j^{th} co-ordinate and 0 elsewhere and $\mathbf{0} = (0,\dots,0) \in \mathbb{F}^n$. Using randomized polynomial identity test in Lemma 3, check if $f_i - \ell_i \equiv 0$. If yes, add e_i copies of ℓ_i to L. Otherwise add e_i copies of black-box computing f_i to B.

3. Simulate black-box $\mathcal{B}$ computing polynomial $NonLin(f) = \prod_{h\in B} h$. **Return** $L,\mathcal{B}$.

□

3 Low Rank Reconstruction: Proof of Theorem 1

We first present Algorithm 2 which proves Theorem 1. Then we analyze it's correctness and running-time.

Algorithm 2 Low rank reconstruction

Input - Black-box access to f, integers n, d.
Output - $\Sigma\Pi\Sigma$ circuit C or $\#$.

1. Using Algorithm 1 with inputs as black-box access to f and integers n, d, compute list of linear factors $\ell_1, \dots, \ell_s$ and black-box access to $NonLin(f)$. Compute degree of $NonLin(f)$ as $t = d - s$. Using this black-box and integers n, t as input to Algorithm 7, obtain set $\mathcal{S}(NonLin(f))$ containing tuples of linear forms representing co-dimension 2 subspaces of $\mathbb{F}^n$ on which $NonLin(f)$ vanishes.

2. Construct set $\mathcal{L}$ of linear forms ℓ, such that for some ℓ' either (ℓ, ℓ') or (ℓ', ℓ) is in $\mathcal{S}(NonLin(f))$. For each $r \in [O(\log^3 d)]$, iterate over all r sized linearly independent subsets $\{y_1, \dots, y_r\} \subset \mathcal{L}$. Construct isomorphism Γ mapping $y_i \mapsto x_i, i \in [r]$. Simulate black-box for $\Gamma(NonLin(f))$ and using Lemma 6 interpolate it as a linear combination of degree t monomials in $\mathbb{F}[x_1, \dots, x_r]$, obtaining a polynomial $h(x_1, \dots, x_r)$.

3. By creating appropriate multiplication/addition gates, construct a $\Sigma\Pi\Sigma(t^r, n, d, \mathbb{F})$ circuit C that computes polynomial
$$f' = \ell_1 \times \dots \times \ell_s \times h(y_1, \dots, y_r)$$
Using randomized polynomial identity test from Lemma 3, check if $f - f' = 0$. If yes, **Return** C. If no, try the next r sized subset in Step 2. If all r sized subsets have been tried, $r = r + 1$.

We first prove the correctness of the above algorithm. Using correctness of Algorithm 1 and Algorithm 7, at the end of step 1, with probability $1 - o(1)$, we have obtained a black-box computing $NonLin(f)$, degree t of $NonLin(f)$, and all linear factors $\ell_1, \dots, \ell_s$ (with multiplicity) of f. Next, we show that, for some $r \leq rank(f)$ and linear forms $y_1, \dots, y_r$, Step 2 computes a polynomial $h(x_1, \dots, x_r)$ such that $NonLin(f) = h(y_1, \dots, y_r)$. In order to do so we prove the following lemma.

Lemma 9. *Let $r = rank(f)$. There exists linearly independent subset $\{y_1, \dots, y_r\} \subset \mathcal{L}$ such that it spans the set of linear factors of $T_1 \times T_2$, implying the existence of the polynomial h.*

Proof. Since $rank(f) \geq 5$, we know that $NonLin(f)$ is a non-constant polynomial. Consider any linear form $\ell \mid T_i$ for some $i \in [2]$. We will show that there is some $\ell' \mid T_{3-i}$ such that $NonLin(f)$ vanishes on the co-dimension 2 subspace $\mathbb{V}(\ell, \ell')$. Assuming this is true, we know there is a tuple $(\ell_1, \ell_2) \in \mathcal{S}(NonLin(f))$ such that $\mathbb{V}(\ell, \ell') = \mathbb{V}(\ell_1, \ell_2) \Rightarrow sp(\{\ell, \ell'\}) = sp(\{\ell_1, \ell_2\})$. By construction of set $\mathcal{L}$, $\ell_1, \ell_2 \in \mathcal{L}$. By going over different ℓ dividing $T_1 \times T_2$ this process would give a list of $2m$ ($m = deg(T_1) = deg(T_2)$) linear forms $\{\ell_1, \dots, \ell_{2m}\} \subset \mathcal{L}$ such that
$$sp(\{\text{linear form } \ell : \ell \mid T_1 \times T_2\}) \subset sp(\{\ell_1, \dots, \ell_{2m}\}) \subset sp(\{\text{linear form } \ell : \ell \mid T_1 \times T_2\})$$
Since $sp(\{\text{linear form } \ell : \ell \mid T_1 \times T_2\})$ is $rank(f)$ dimensional we get that there are r linearly independent linear forms $y_1, \dots, y_r \in \{\ell_1, \dots, \ell_{2m}\} \subset \mathcal{L}$ and the proof would be complete. So we only need to show that there exists $\ell' \mid T_{3-i}$ such that $NonLin(f)$ vanishes on the co-dimension 2 subspace $\mathbb{V}(\ell, \ell')$. To see this, let L be the product of all linear factors of $T_1 + T_2$. Let Φ be an isomorphism mapping $\ell \mapsto x_1$. On setting $x_1 = 0$, we get,
$$\Phi(L)_{|x_1=0} \times \Phi(NonLin(f))_{|x_1=0} = \Phi(T_{3-i})_{|x_1=0} \neq 0.$$

The non zeroness comes from the fact that $gcd(T_1, T_2) = 1$. The above equation implies[10] that there is some linear form $\ell' \mid T_{3-i}$ such that $\Phi(\ell')_{|x_1=0}$ divides $\Phi(NonLin(f))_{|x_1=0}$. Now, define the isomorphism Δ mapping $x_1 \mapsto x_1, \Phi(\ell') \mapsto x_2$[11]. Applying Δ to the fact that $\Phi(\ell')_{|x_1=0}$ divides $\Phi(NonLin(f))_{|x_1=0}$, we get that $\Delta(\Phi(\ell')_{|x_1=0}) \mid \Delta(\Phi(NonLin(f))_{|x_1=0})$. Since Δ fixes x_1, we get $\Delta(\Phi(\ell'))_{|x_1=0} \mid \Delta(\Phi(NonLin(f)))_{|x_1=0}$. So there is polynomial g such that

$$\Delta(\Phi(NonLin(f)))_{|x_1=0} = \Delta(\Phi(\ell'))_{|x_1=0} \times g.$$

Now setting $x_2 = 0$ on both sides will send the right hand side to 0 since $\Delta \circ \Phi$ maps $\ell \mapsto x_1, \ell' \mapsto x_2$. Therefore $\Delta(\Phi(NonLin(f)))_{|x_1=0,x_2=0} = 0$, and so using Definition 2 one can see that $NonLin(f)$ vanishes on the co-dimension 2 subspace $\mathbb{V}(\ell, \ell')$. □

$h(y_1, \ldots, y_r)$ naturally exhibits a $\Sigma\Pi\Sigma(t^r, n, t, \mathbb{F})$ circuit. This can be seen as follows. Addition gates at the bottom layer will compute linear forms $y_1, \ldots, y_r$. For each monomial, there will be one multiplication gate. If x_j^k is the largest power of x_j dividing some monomial, then there will be k connections from y_j to the multiplication gate corresponding to this monomial. Finally, the top layer is connected to all the multiplication gates and weight on such an edge is equal to the coefficient of the monomial the multiplication gate corresponded to. Step 3 just multiplies this circuit with all the linear factors and therefore computes a candidate $\Sigma\Pi\Sigma(t^r, n, d, \mathbb{F})$ circuit for f. Randomized polynomial identity test in Step 3 ensures that with high probability we output a correct $\Sigma\Pi\Sigma(d^r, n, t, \mathbb{F})$ circuit for f. If for some r and linear forms $y_1, \ldots, y_r$, an incorrect circuit gets constructed, probability that it will be outputted is $o(1)$. There are at most $(d^{\log^3 d} \log^3 d)^{O(1)}$ many such bad settings of r and $y_1, \ldots, y_r$. Using boosting with independent runs of randomized polynomial identity test, we can make error exponentially small in nd so that overall the probability of error still remains $o(1)$ by union bound $\Rightarrow$ with probability $1 - o(1)$ all these bad settings will be rejected. For $r = rank(f)$ and the correct linearly independent set $\{y_1, \ldots, y_r\}$ (i.e. one spanning all linear factors of $T_1 \times T_2$), we have seen that with probability $1 - o(1)$, a correct circuit will be constructed which will always pass the randomized polynomial identity test and will be returned. So overall with probability $1 - o(1)$, a correct $\Sigma\Pi\Sigma(t^r, n, d, \mathbb{F})$ circuit for f will be returned. Next we discuss the time complexity of the above algorithm.

Lemma 10. *Algorithm 2 takes* $(nd^{\log^3 d} \log |\mathbb{F}|)$ *time.*

Proof. Time complexity of Algorithm 1 and Algorithm 7 imply that Step 1 takes $(nd \log |\mathbb{F}|)^{O(1)}$ time. $\mathcal{L}$ can be constructed in $(nd \log |\mathbb{F}|)^{O(1)}$ time since it involves iterating over the $d^{O(1)}$ sized set $\mathcal{S}(NonLin(f))$. Our search for the correct $r = rank(f)$ and linear forms $y_1, \ldots, y_r$ takes $(nd^{\log^3 d} \log |\mathbb{F}|)^{O(1)}$ time in the worst case and multivariate interpolation (Lemma 6) also takes the same amount of time in the worst case. Step 3 multiplies linear factors to all the gates in the circuit for $NonLin(f)$ and therefore takes $(nd^{\log^3 d} \log |\mathbb{F}|)^{O(1)}$ time and therefore overall time complexity is $(nd^{\log^3 d} \log |\mathbb{F}|)^{O(1)}$. □

4 High Rank Reconstruction: Proof of Theorem 2

The algorithm in Theorem 2 is presented below in Algorithm 3. This algorithm further calls Algorithms 4, 5 and 6. We present and analyze them in Sections 4.1, 4.2 and 4.3 respectively.

[10]by using unique factorization in the ring $\mathbb{F}[x_2, \ldots, x_n]$

[11]x_1 and $\Phi(\ell')$ are linearly independent, otherwise ℓ divides ℓ' violating $gcd(T_1, T_2) = 1$.

Correctness of our algorithm heavily relies on Lemma 21, which we prove in Section 4.4. We first give the full algorithm and then discuss it's correctness and time complexity.

Algorithm 3 High rank reconstruction

Input - Black-box access to f, integers n, d.
Output - $\Sigma\Pi\Sigma(2, n, d, \mathbb{F})$ circuit C or $\#$.

1. Run Algorithm 5 with inputs as black-box access to f along with integers n, d. If output is a circuit C, **Return** C. If output was $\#$, go to the next step.

2. Using Algorithm 1 with input as black-box access to f and integers n, d, compute list of linear factors $\ell_1, \ldots, \ell_s$ and black-box access to $NonLin(f)$. Compute the degree of $NonLin(f)$ as $t = d - s$.

3. Using Algorithm 4 with inputs as black-box access to f and integers n, d, construct the set $\mathcal{L}(NonLin(f))$. For each $\ell \in \mathcal{L}(NonLin(f))$ consider all linear forms $\ell' \in \mathcal{L}(NonLin(f)) \setminus \{\ell\}$ such that $sp\{\ell, \ell'\}$ does not intersect $\mathcal{L}(NonLin(f))$ at any point other than ℓ, ℓ'. Find a maximal independent set $\mathcal{X}$ of such ℓ's and continue if $|\mathcal{X}| = \Omega(\log^2 d)$. If no such ℓ exists, **Return** $\#$. Otherwise, partition $\mathcal{X}$ into equal parts of size $\Omega(\log d)$ each and iterate over all parts $\mathcal{B}$.

   (a) Initialize sets $\mathcal{U}, \mathcal{V} \leftarrow \phi$. Iterate over all linear forms $\ell' \in \mathcal{B}$. Define an isomorphism Φ mapping $\ell \mapsto x_1, \ell' \mapsto x_2$ and using Lemma 3, check if $\Phi(NonLin(f))_{|x_1=0,x_2=0} \equiv 0$. If yes, add ℓ' to $\mathcal{U}$ else add it to $\mathcal{V}$. Select $r = 60\log d + 61$ linear forms $y_1, \ldots, y_r$ from the larger of $\mathcal{U}, \mathcal{V}$.

   (b) Run Algorithm 6 with inputs as black-box access to f, integers n, d and linear forms $y_1, \ldots, y_r$. If it returns a $\Sigma\Pi\Sigma(2, n, d, \mathbb{F})$ circuit C, **Return** C. Else, go to the next partition $\mathcal{B}$ and then to the next linear form ℓ in the search.

4. **Return** $\#$

We first prove the correctness of the above algorithm. Step 1 first tries to solve the corner case where one of T_1, T_2 is power of a linear form. By correctness of Algorithm 6, we know that, if this corner case is satisfied, then with probability $1 - o(1)$, the correct $\Sigma\Pi\Sigma(2, n, d, \mathbb{F})$ circuit is returned. Also Algorithm 6 never returns an incorrect circuit. Therefore with high probability Step 1 will complete the reconstruction if the corner case condition holds. If it does not hold this algorithm will always proceed to Step 2. Also, if it does not return a circuit we can assume that with high probability the corner case does not hold and therefore linear factors of each T_i span at least a two dimensional space. By correctness of Algorithm 1, we know that with probability $1 - o(1)$, Step 2 correctly obtains a black-box computing $NonLin(f)$, it's degree t and correctly identifies all linear factors of f with multiplicity. Correctness of the next step is proved in the following lemma.

Lemma 11. *If outputs of Steps 1 and 2 are correct, then with probability $1 - o(1)$, Step 3 computes a $\Sigma\Pi\Sigma(2, n, d, \mathbb{F})$ circuit computing f.*

Proof. By correctness of Algorithm 4, we know that the set $\mathcal{L}(NonLin(f))$ is correctly computed. Our algorithm goes through all linear forms $\ell \in \mathcal{L}(f)$ and for each such linear form goes through

$\Omega(\log d)$ sized sets which are parts of a partition of the set $\mathcal{X}$ defined using ℓ. In Step $3(b)$, correctness of Algorithm 6 ensures that if a circuit is returned for any choice of $\ell, \mathcal{B}$, it is always correct. So all we need to show is that for some choice of $\ell, \mathcal{B}$, Algorithm 6 will return the correct circuit with high probability. We know from correctness of Algorithm 6 that if the linear forms $y_1, \ldots, y_r$ (that are given as input to it), all divide the same T_i and are independent, then with high probability a correct circuit will be returned. Therefore, now all we need to show is that there is some choice of $\ell, \mathcal{B}$, for which the constructed $y_1, \ldots, y_r$ are independent linear forms dividing the same T_i. Since we have assumed that output of Step 1 is correct, f does not satisfy the corner case implying that linear factors of each T_i span at least a two dimensional space and therefore Lemma 21 can be applied. Parts $1, 2, 3$ of Lemma 21 prove that such $\ell, \mathcal{B}$ exist for which the test in Step $3(a)$ creates a partition $\mathcal{U} \cup \mathcal{V} = \mathcal{B}$ such that linear forms in $\mathcal{U}$ divide T_j and linear forms in $\mathcal{V}$ divide T_{3-j} for some $j \in [2]$. Since $|\mathcal{B}| = \Omega(\log d)$, one of $\mathcal{U}, \mathcal{V}$ has size $\Omega(\log d)$. By construction $\mathcal{B}$ is linearly independent and thus both $\mathcal{U}, \mathcal{V}$ are linearly independent. Therefore $y_1, \ldots, y_r$ with $r = \Omega(\log d)$ are independent linear forms dividing the same T_i. This completes the proof. ☐

Now we discuss the time complexity of the above algorithm.

Lemma 12. *Algorithm 3 takes* $(nd \log |\mathbb{F}|)^{O(1)}$ *time.*

Proof. Time complexity of Algorithm 5 and Algorithm 1 imply that Steps 1 and 2 take $O(nd \log |\mathbb{F}|)^{O(1)}$ time. By Algorithm 4 we know that the set $\mathcal{L}(f)$ has $d^{O(1)}$ size. We iterate over all $\ell \in \mathcal{L}(NonLin(f))$ and for each ℓ' check if $sp\{\ell, \ell'\}$ intersects $\mathcal{L}(NonLin(f))$ at any other point. This can be done in $(nd \log |\mathbb{F}|)^{O(1)}$ time. From these ℓ', we can simply check linear independence of linear forms and create a maximal set in $\mathcal{X}$ in $(nd \log |\mathbb{F}|)^{O(1)}$ time. Creating a partition of $\mathcal{X}$, iterating over all parts $\mathcal{B}$, and isomorphism can be created in $(nd \log |\mathbb{F}|)^{O(1)}$ time. Isomorphism can be efficiently applied to the black-box computing $NonLin(f)$ by taking every input through Φ before applying the black-box. By time complexity of algorithm in Lemma 3, the check in Step $3(a)$ takes $(nd \log |\mathbb{F}|)^{O(1)}$ time. Time complexity of Step Algorithm 6 implies that Step $3(b)$ takes $(nd \log |\mathbb{F}|)^{O(1)}$ time. Therefore overall Algorithm 3 takes $(nd \log |\mathbb{F}|)^{O(1)}$ time. ☐

In the next subsection, we explain construction of the candidate linear forms (Definition 6).

4.1 Computing Candidate Linear forms

Here is a lemma summarizing the construction of set $\mathcal{L}(NonLin(f))$ of candidate linear forms (Definition 6).

Lemma 13. *There exists a randomized algorithm that takes as input integers n, d and black-box access to f, runs in time $(nd \log |\mathbb{F}|)^{O(1)}$, and outputs a set $\mathcal{L}$ of linear forms such that,*

$$Pr[\mathcal{L} =^{12} \mathcal{L}(NonLin(f))] = 1 - o(1).$$

We first give the algorithm and then discuss it's correctness and time complexity.

[12] up to scalar multiplication of linear forms in the sets

Algorithm 4 Candidate linear forms

Input - Black-box access to polynomial f, integers n, d.

Output - A set of linear forms $\mathcal{L}$.

1. Using Algorithm 1 with inputs as black-box access to f and integers n, d, obtain list of linear factors $\ell_1, \dots, \ell_s$ and access to black-box computing $NonLin(f)$. Compute degree of $NonLin(f)$ as $t = d - s$. Using Algorithm 7, compute the set $\mathcal{S}$ of tuples of linear forms representing co-dimension 2 subspaces on which $NonLin(f)$ vanishes.

2. Initialize $\mathcal{L} \leftarrow \phi$. For all pairs of tuples $(p_1, q_1), (p_2, q_2) \in \mathcal{S}$, check if $sp\{p_1, q_1\} \cap sp\{p_2, q_2\}$is one dimensional. For this we construct the $n \times 4$ matrix M with it's columns containing coefficients of p_1, q_1, p_2, q_2 respectively and then check by gaussian elimination whether rank of M is 3 or not. If yes, the same gaussian elimination can be used to obtain the one dimensional space of solutions to $Mv = 0$ for $v \in \mathbb{F}^4$. Fixing one such non-zero solution $u = (\alpha_1, \alpha_2, \alpha_3, \alpha_4)^T$ then gives us a scalar multiple of ℓ as $\alpha_1 p_1 + \alpha_2 q_1$. If no scalar multiple of $\alpha_1 p_1 + \alpha_2 q_1$ is already present in $\mathcal{L}$, then we add it to $\mathcal{L}$.

3. For each $\ell \in \mathcal{L}$, check whether $NonLin(f)$ restricted to $\mathbb{V}(\ell)$ factorizes into a non-zero product of linear forms (See Definition 5). This can be done by defining an isomorphism Φ mapping $\ell \mapsto x_1$, simulating black-box computing $\Phi(NonLin(f))_{|x_1=0}$. Using Lemma 3, check if this black-box computes the 0 polynomial. If 'yes', remove ℓ from $\mathcal{L}$. Otherwise, using Algorithm 1, with inputs as this restricted black-box and integers n, t, compute list of linear factors and check whether there are t of them. If not, then remove ℓ from $\mathcal{L}$. Finally, **Return** $\mathcal{L}$.

Now we prove the correctness of the above algorithm. By correctness of Algorithm 1, we know that Step 1 correctly obtains black-box access to $NonLin(f)$, it's degree t and linear factors (with multiplicity) of f with probability $1 - o(1)$. Similarly by correctness of Algorithm 7, we know that with probability $1 - o(1)$, the set $\mathcal{S}$ representing elements of $\mathcal{S}(NonLin(f))$ is correctly computed. We prove correctness of the next two steps in the following lemma.

Lemma 14. *Assuming Step 1 works correctly, with probability $1 - o(1)$, the output $\mathcal{L}$ of Algorithm 4 is the same*[13] *as $\mathcal{L}(NonLin(f))$.*

Proof. Consider any $\ell \in \mathcal{L}(NonLin(f))$. By definition of the set $\mathcal{L}(NonLin(f))$, we know that there are linear forms ℓ_1, ℓ_2 with ℓ, ℓ_1, ℓ_2 linearly independent, such that the co-dimension 2 subspaces $\mathbb{V}(\ell, \ell_1), \mathbb{V}(\ell, \ell_2) \in \mathcal{S}(NonLin(f))$. So some tuples (p_1, q_1) and (p_2, q_2) corresponding to these two subspaces will be present in $\mathcal{S}$ and will be encountered in Step 2. Note that $\mathbb{V}(p_1, q_1) = \mathbb{V}(\ell, \ell_1)$ and $\mathbb{V}(p_2, q_2) = \mathbb{V}(\ell, \ell_2)$ implies that $sp\{p_1, q_1\} = sp\{\ell, \ell_1\}$ and $sp\{p_2, q_2\} = sp\{\ell, \ell_2\}$ further implying that $sp\{p_1, q_1\} \cap sp\{p_2, q_2\} = sp\{\ell\}$. This implies that there are scalars $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ such that $\alpha_1 p_1 + \alpha_2 q_1 + \alpha_3 p_2 + \alpha_4 q_2 = 0$, giving us the system of equations as described in the algorithm. In order for the intersection to be one dimensional, the matrix M should have rank 3. We check that using gaussian elimination which also gives the one dimensional set of solutions. Any non-zero solution $(\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ will then give a linear form $\alpha_1 p_1 + \alpha_2 q_1$ in the intersection which will be a scalar multiple of ℓ. Thus, Step 2 identifies a scalar multiple of ℓ and adds it to $\mathcal{L}$. Step 3 just checks whether $NonLin(f)$ factorizes as a product of non-zero linear forms on $\mathbb{V}(\ell)$

[13] the linear forms in this output are correct upto scalar multiplication

(see Definition 5). Correctness of Step 3 is implied by correctness of Lemma 3 and Algorithm 1. Since $\ell \in \mathcal{L}(NonLin(f))$, it will pass this test and remain in $\mathcal{L}$. Now consider any $\ell \in \mathcal{L}$ that is returned. In Steps 2 and 3 we have checked whether it satisfies the conditions required for it to be in $\mathcal{L}(NonLin(f))$ or not. Therefore we do not return any extra linear forms and correctly output $\mathcal{L}(NonLin(f))$ with high probability. ☐

Now we discuss the time complexity of the above algorithm.

Lemma 15. *Algorithm 4 takes* $(nd \log |\mathbb{F}|)^{O(1)}$ *time.*

Proof. Time complexity of Algorithm 1 and Algorithm 7 imply that Step 1 takes $(nd \log |\mathbb{F}|)^{O(1)}$ time. By first part of Proposition 1, we know that $|\mathcal{S}| \leq 3d^7$ and therefore going over pairs of elements of $\mathcal{S}$ takes $O(nd \log |\mathbb{F}|)^{O(1)}$ time. Gaussian elimination on matrix M takes $(n \log |\mathbb{F}|)^{O(1)}$ time for each pair of tuples. After Step 2 we will have at most $|\mathcal{S}|^2$ many elements in $\mathcal{L}$ leading to a size of $d^{O(1)}$. In Step 3 for every $\ell \in \mathcal{L}$, the construction of Φ, simulation of black-box for $\Phi(NonLin(f))_{|x_1=0}$ are done in $(n \log |\mathbb{F}|)^{O(1)}$ time. Time complexity of algorithm provided in Lemma 3 tests in time $(nd \log |\mathbb{F}|)^{O(1)}$, whether this new polynomial is identically zero or not. Finally, time complexity of Algorithm 1 implies that in time $(nd \log |\mathbb{F}|)^{O(1)}$ we can check whether it has t linear factors or not. Therefore overall Algorithm 4 takes$(nd \log |\mathbb{F}|)^{O(1)}$ time. ☐

4.2 Reconstruction when T_1 (or T_2) $= \alpha y_1^t$

This is a corner case of our problem and needs slightly different techniques. Here is a lemma summarizing the reconstruction algorithm in this case.

Lemma 16. *If for some* $i \in [2]$, $T_i = \alpha y_1^t$ *for some linear form* y_1 *and* $\alpha \in \mathbb{F}$, *then there exists a randomized algorithm that takes as input integers* n, d *and black-box access to polynomial* f, *runs in time* $(nd \log |\mathbb{F}|)^{O(1)}$, *and with probability* $1 - o(1)$ *outputs a* $\Sigma\Pi\Sigma(2, n, d, \mathbb{F})$ *circuit computing* f.

We first give the algorithm and then discuss it's correctness and time complexity.

Algorithm 5 Corner case

Input - Black-box access to polynomial f, integers n, d.
Output - A $\Sigma\Pi\Sigma(2,n,d,\mathbb{F})$ circuit or $\#$.

1. Using Algorithm 1 with inputs as black-box access to f and integers n, d compute linear factors $\hat{\ell}_1, \ldots, \hat{\ell}_s$ and get access to black-box computing $NonLin(f)$. Compute degree of $NonLin(f)$ as $t = d - s$. Using Algorithm 4, compute set $\mathcal{L}(NonLin(f))$.

2. Iterate over linear forms $\ell_1 \in \mathcal{L}(NonLin(f))$. Construct an isomorphism Φ mapping $\ell_1 \mapsto x_1$.

   (a) Simulate black-box for $\Phi(NonLin(f))_{|\{x_1=0\}}$ and using Algorithm 1 identify two linearly independent factors say ℓ_2, ℓ_3. Construct another isomorphism Δ mapping $x_1 \mapsto x_1, \ell_2 \mapsto x_2, \ell_3 \mapsto x_3$. Pick $\alpha_4, \ldots, \alpha_n$ uniformly randomly from $\mathbb{F}$. Simulate black-box for

   $$g(x_1,x_2,x_3) = \Delta(\Phi(NonLin(f)))_{|\{x_4=\alpha_4,\ldots,x_n=\alpha_n\}}$$

   (b) Using Lemma 6, interpolate g in monomial basis of $\mathbb{F}[x_1,x_2,x_3]$. Substitute $x_2 = yx_1$ in all monomials and rearrange to get a representation in $\mathbb{F}[y][x_1,x_3]$. Equate coefficient polynomials of monomials containing x_3 to 0 and solve the resulting system of equations using Lemma 1. If all ℓ_1's have been tried and no solution was obtained, **Return** $\#$. Otherwise, for each solution, evaluate coefficient polynomial of x_1^t, creating a set of scalars.

   (c) Iterate over all δ's in the set of scalars obtained above. Simulate black-box for $NonLin(f) - \delta\ell_1^t$ and using Algorithm 1 check if it has t linear factors say $\ell_{s+1}, \ldots, \ell_{s+t}$. If not, then go to the next δ. If all δ have been tried, go to next $\ell_1 \in \mathcal{L}(NonLin(f))$. If all ℓ_1's have been tried, **Return** $\#$. Otherwise, simulate black-box for $f - f'$, where

   $$f' = \hat{\ell}_1 \times \ldots \times \hat{\ell}_s \times (\delta\ell_1^t + \ell_{s+1} \times \ldots \times \ell_{s+t})$$

   and using Lemma 4 for $\Sigma\Pi\Sigma(4,n,d,\mathbb{F})$ circuits, check if $f - f' \equiv 0$. If output is 'yes', construct $\Sigma\Pi\Sigma(2,n,d,\mathbb{F})$ circuit C computing f'. **Return** C. If not, then go to next δ. If all δ have been tried, go to next $\ell_1 \in \mathcal{L}(NonLin(f))$. If all ℓ_1's have been tried, **Return** $\#$.

Now we prove the correctness of the above algorithm. By correctness of Algorithm 1, with probability $1 - o(1)$, Step 1 correctly obtains the black-box for $NonLin(f)$, it's degree t and the multi-set of all linear factors of f. If we assume that these are correct, then by correctness of Algorithm 4, with probability $1 - o(1)$, Step 1 also correctly computes the set $\mathcal{L}(NonLin(f))$[14] of linear forms. In order to prove the correctness of Step 2 we give two claims, both of which are proved in Appendix A. The first claim says that in this corner case, $NonLin(f)$ is actually the same as $T_1 + T_2$ (up to scalar multiplication) and the second claim guarantees that some scalar multiple of y_1 actually belongs to the set $\mathcal{L}(NonLin(f))$. Here are the formal statements.

Claim 1. *Assume $T_i = \alpha y_1^t$, for some $i \in [2]$, $\alpha \in \mathbb{F}$ and linear form y_1. Then $Lin(f) = G$ (up to scalar factor). This also means that $NonLin(f)$ and $T_1 + T_2$ are equal up to a scalar factor.*

[14]all linear forms are correct up to scalar multiple.

Claim 2. *Assume $T_i = \alpha y_1^t$, for some $i \in [2]$, $\alpha \in \mathbb{F}$ and linear form y_1, then some scalar multiple of y_1 belongs to $\mathcal{L}(NonLin(f))$.*

We proceed in our correctness proof assuming that these claims are true. Assuming that Step 1 was correct, we show that Step 2 returns the correct circuit with high probability. Note that in Step $2(c)$, using Lemma 4, we check whether the reconstructed circuit is correct or not. This ensures that we only return a correct circuit. Our algorithm in Steps $2(b), 2(c)$ tries all linear forms in $\mathcal{L}(NonLin(f))$ and for each such linear form it constructs a set of scalars. So basically the algorithm iterates over possibilities of ℓ_1, δ with the hope of finding one such that $T_i = \delta\ell_1^t$. If we can show that for some value of ℓ_1, δ with high probability a correct $\Sigma\Pi\Sigma(2, n, d, \mathbb{F})$ circuit is reconstructed, we will be done. We show this in the following lemma. We show this for ℓ_1 being the scalar multiple of y_1 that belongs to $\mathcal{L}(NonLin(f))$ (guaranteed by Claim 2).

Lemma 17. *For ℓ_1, the scalar multiple of y_1 in $\mathcal{L}(NonLin(f))$, the set of scalars constructed in Step $2(b)$ contains a scalar δ such that $T_i = \alpha y_1^t = \delta\ell_1^t$ and with probability $1 - o(1)$ correctly reconstructs a $\Sigma\Pi\Sigma(2, n, d, \mathbb{F})$ circuit computing f.*

Proof. We know that $NonLin(f)$ restricted to the co-dimension 1 subspace $\mathbb{V}(\ell_1)$ factors into a non-zero product of linear forms. By correctness of Algorithm 1, we know that all linear factors of $\Phi(NonLin(f))_{|x_1=0}$ can be computed. By Claim 1, we know that this is the same as $\Phi(T_{3-i})_{|x_1=0}$ up to scalar multiplication. Since $rank(f) = \Omega(\log^3 d)$ and linear factors of T_i span a 1 dimensional space, factors of this polynomial will span an $\Omega(\log^3 d)$ dimensional space and therefore we will be able to find at least two linearly independent factors ℓ_2, ℓ_3 in $\mathbb{F}[x_2, \ldots, x_n]$. This means that the polynomial $\Phi(NonLin(f))$ looks like

$$\Phi(NonLin(f)) = \delta\ell_1^t + (\ell_2 - \beta x_1)(\ell_3 - \gamma x_1)\prod_{i=4}^{t+1} \ell_i,$$

for some scalars β, γ and linear forms $\ell_4, \ldots, \ell_{t+1}$ in $\mathbb{F}[x_1, \ldots, x_n]$. Recall the isomorphism Δ used in the algorithm, mapping $x_1 \mapsto x_1, \ell_2 \mapsto x_2, \ell_3 \mapsto x_3$. Black-box computing the polynomial $\Delta(\Phi(NonLin(f)))$ can be constructed by taking every input of blackbox through the isomorphisms. The new polynomial now looks like

$$\Delta(\Phi(NonLin(f))) = \delta x_1^t + (x_2 - \beta x_1)(x_3 - \gamma x_1)\prod_{i=4}^{t+1} \Delta(\ell_i),$$

Finally, we plug in uniform random values for the variables $x_4, \ldots, x_n$. By Lemma 2 we know that with probability $1 - o(1)$ the polynomial $\prod_{i=4}^{t+1} \Delta(\ell_i)$ will not be identically zero and we will be left with a non-zero polynomial $g(x_1, x_2, x_3)$ computable by a $\Sigma\Pi\Sigma(2, 3, d, \mathbb{F})$ circuit.

$$g(x_1, x_2, x_3) = \delta x_1^t + (x_2 - \beta x_1)(x_3 - \gamma x_1)\prod_{i=4}^{t+1} u_i,$$

where u_i are affine forms in $\mathbb{F}[x_1, x_2, x_3]$. Using the above black-box, we get access to black-box for g and then using deterministic multivariate interpolation (Lemma 6), interpolate it as a degree t polynomial in the monomial basis of $\mathbb{F}[x_1, x_2, x_3]$. g depends on variable x_3. So substituting $x_2 = yx_1$ for a fresh variable y, and solving for common zeros of all coefficient (of monomials involving

x_3) univariate polynomials in $\mathbb{F}[y]$ would give us a set of scalars containing β. Note that, since our system has only univariate polynomials, all of degree $d^{O(1)}$, it can have at most $d^{O(1)}$ solutions. By correctness of algorithm in Lemma 1, with probability $1-o(1)$, this set would be correctly computed. Now substitution of $x_2 = \beta x_1$ would recover δ as coefficient of x_1^t. By correctness of Algorithm 1, with probability $1-o(1)$, we will be able to completely factorize the black-box $NonLin(f) - \delta\ell_1^t$ into a product of t linear factors giving us the correct T_{3-i}. By correctness of Step 1, we know all linear factors of f, were correctly computed and therefore for scalar multiple ℓ_1 of y_1 and the computed scalar δ, with probability $1-o(1)$, we reconstruct a correct $\Sigma\Pi\Sigma(2,n,d,\mathbb{F})$ circuit for f. Hence Proved. □

Now we discuss the time complexity of the above algorithms.

Lemma 18. *Algorithm 5 takes* $(nd\log|\mathbb{F}|)^{O(1)}$ *time.*

Proof. Time complexity of Algorithm 1 and Algorithm 4 imply that Step 1 takes $(nd\log|\mathbb{F}|)^{O(1)}$ time. In Step 2, the outer iteration is over all linear forms in $\mathcal{L}(NonLin(f))$ which has size $d^{O(1)}$ (clear from Definition 6 and explanation given in Algorithm 4). Step $2(a)$ involves simulations of black-boxes post application of isomorphism and setting values for some variables. It also involves using Algorithm 1 to compute all linear factors. All these steps take $(nd\log|\mathbb{F}|)^{O(1)}$ time. Finding linearly independent pair of linear forms out of all linear factors is also done in $(nd\log|\mathbb{F}|)^{O(1)}$ time. Step 3 involves trivariate interpolation (Lemma 6) which takes $(d\log|\mathbb{F}|)^{O(1)}$ time and by time complexity of Lemma 1 solutions of the system of univariate polynomials (all have degree $d^{O(1)}$) are also found in $(nd\log|\mathbb{F}|)^{O(1)}$ time. The set of solutions is $d^{O(1)}$ sized since a univariate polynomial of degree d has at most d roots over a field. Therefore Step $2(b)$ takes $(nd\log|\mathbb{F}|)^{O(1)}$ time and creates a set of scalars of size $d^{O(1)}$. Step $2(c)$ iterates over this $d^{O(1)}$ sized set. Simulation of black-box and factorization using Algorithm 1 take $(nd\log|\mathbb{F}|)^{O(1)}$ time. Blackbox for $f - f'$ is constructed in $(nd\log|\mathbb{F}|)^{O(1)}$ time and by time complexity of algorithm in Lemma 4, it can be checked to be 0 or not in $(nd\log|\mathbb{F}|)^{O(1)}$ time. Therefore overall Algorithm 5 takes $(nd\log|\mathbb{F}|)^{O(1)}$ time. □

4.3 Reconstruction with linearly independent set dividing T_i given

Suppose we are given linearly independent linear forms $u_1,\ldots,u_t$, $t > 60\log d + 61$, such that for some $i \in [2]$, all the u_j's divide T_i. Then there exists an efficient reconstruction algorithm as summarized in lemma below.

Lemma 19. *There exists a randomized algorithm which takes as input integers n,d, black-box access to polynomial f computable by a $\Sigma\Pi\Sigma(2,n,d,\mathbb{F})$ circuit and linearly independent linear forms $u_1,\ldots,u_t$, $t > 60\log d + 61$ (for some $i \in [2]$, all u_j's divide T_i), runs in time $(nd\log|\mathbb{F}|)^{O(1)}$ and with probability $1-o(1)$ outputs a $\Sigma\Pi\Sigma(2,n,d,\mathbb{F})$ circuit computing f.*

We present the algorithm for proving the above lemma in Algorithm 6. We use Algorithm 5 of [Shp07] in Step 2. More details on this merge algorithm can be found in Algorithm 5 and Theorem 29 of [Shp07].

Algorithm 6 Linearly independent linear factors of a multiplication gate are known

Input - Black-box access to polynomial f, integers n, d, linear forms $u_1, \ldots, u_t$, $t > 60 \log d + 61$.

Output - A $\Sigma\Pi\Sigma(2, n, d, \mathbb{F})$ circuit C or #.

1. Construct isomorphism Φ mapping $u_i \mapsto x_i, i \in [t]$ and simulate black-box computing $\Phi(f)$. Using Algorithm 1 with inputs as black-box computing $\Phi(f)$ and integers n, d, obtain all it's linear factors (with multiplicity) along with access to black-box computing $\Phi(NonLin(f))$. By traversing through the factors identify e_i, the largest power of x_i that divides $\Phi(f)$. Using this set of factors and black-box computing $\Phi(f)$, simulate black-box computing $g = \Phi(f)/\prod x_i^{e_i}$.

2. For each $i \in [t]$, simulate black-box computing $g_{|_{\{x_i=0\}}}$ and using Algorithm 1 with inputs as this black-box, compute it's factors. If there are non linear factors, **Return** #. Otherwise, store factors in multi-set $\mathcal{U}_i$. Using Algorithm 5 in [Shp07] merge the multi-sets $\mathcal{U}_i$ together to obtain a multiset $\mathcal{U}$ comprising of all linear factors of one of the product gates in the $\Sigma\Pi\Sigma(2, n, s, \mathbb{F})$ circuit computing g (here s is some integer $\leq d$).

3. Construct the multi-set $\mathcal{U}' = \{\ell_{|_{\{x_1=0\}}} : \ell \in \mathcal{U}\}$. Check if this multi-set $\mathcal{U}'$ and $\mathcal{U}_1$ contain same linear forms (upto multiplicity). If not, **Return** #. Otherwise compute scalar $\alpha = \prod_{\ell \in \mathcal{U}_1} \ell \Big/ \prod_{\ell \in \mathcal{U}'} \ell$ by matching linear forms between $\mathcal{U}', \mathcal{U}_1$.

4. Simulate black-box computing $g - \alpha \prod_{\ell \in \mathcal{U}} \ell$ and factorize this polynomial using Algorithm 1. If all factors are not linear, **Return** #. Otherwise, store factors in multi-set $\mathcal{V}$. Apply Φ^{-1} to all linear forms in $\mathcal{U}, \mathcal{V}$. Simulate black-box for $f - f'$, where
$$f' = \prod_{i=1}^{t} u_i^{e_i} \times (\alpha \prod_{\ell \in \mathcal{U}} \ell + \prod_{\ell \in \mathcal{V}} \ell)$$
Using Lemma 4 for $\Sigma\Pi\Sigma(4, n, d, \mathbb{F})$ circuits, check if $f - f' \equiv 0$. If output is 'yes', construct $\Sigma\Pi\Sigma(2, n, d, \mathbb{F})$ circuit C computing f' and **Return** C. If not, then **Return** #.

Now we prove correctness of the above algorithm. Black-box computing $\Phi(f)$ is simulated by passing every input through Φ first. Correctness of Algorithm 1 imply that with probability $1 - o(1)$, all linear factors of $\Phi(f)$ and black-box access to $\Phi(NonLin(f))$ are correctly computed. From these linear forms, we remove any linear form ℓ that are divisible by some x_i. However we will keep the scalar ℓ/x_i. The black-box obtained by multiplying the black-box of $\Phi(NonLin(f))$ returned by Algorithm 1 with these scalars and black-boxes computing the remaining linear factors simulates black-box access to $g = \Phi(f)/\prod_{i=1}^{t} x_i^{e_i}$. g is a $\Sigma\Pi\Sigma(2, n, s, \mathbb{F})$ circuit for some integer $s \leq d$. Assuming that Step 1 is correct, simulation of black-boxes $g_{|_{x_i=0}}, i \in [t]$ can be done. Correctness of Algorithm 1 implies that with probability $1 - o(1)$ all multi-sets $\mathcal{U}_i$ are correctly computed. By correctness of Algorithm 5 in [Shp07], we know that these multi-sets are glued together to obtain a multi-set $\mathcal{U}$ containing all linear factors of one of the product gates S_2 of g (we are assuming that $g = S_1 + S_2$ where S_1, S_2 are product of linear forms and $x_i \mid S_1$ for $i \in [t]$.). Note that the algorithm only recovers all linear factors of S_2 and therefore it still needs to recover an appropriate scalar α (see algorithm) to completely recover S_2. Note that $g_{|x_1=0} = S_{2|x_1=0} \neq 0$. Therefore we can compare the multi-set of linear forms in $\mathcal{U}_1$ with the multi-set of linear forms $\mathcal{U}' = \{\ell_{|x_1=0} : \ell \in \mathcal{U}\}$. All linear forms will match up to scalar multiplication giving us the scalar α. By correctness of Algorithm 1, we know that with probability $1 - o(1)$, we will be able to correctly factor $g - \alpha \prod_{\ell \in \mathcal{U}} \ell$ and collect them in multi-set $\mathcal{V}$. Finally at the end, we can apply Φ^{-1} and multiply by $\prod_{i=1}^{t} u_i^{t_i}$ and correctly recover the $\Sigma\Pi\Sigma(2, n, d, \mathbb{F})$ circuit with probability $1 - o(1)$. Note that in Step 4, by correctness of Lemma 4, we know that we can deterministically check whether the constructed circuit is correct or not and only return a correct circuit. Now we discuss the time complexity of the above algorithm.

Lemma 20. *Algorithm 6 runs in time $(nd \log|\mathbb{F}|)^{O(1)}$ time.*

Proof. Isomorphism Φ is constructed in $(n \log|\mathbb{F}|)^{O(1)}$ time. Time complexity of Algorithm 1 implies that $(nd \log|\mathbb{F}|)^{O(1)}$ time is spent on factorizing $\Phi(f)$. Removing powers of $x_i, i \in [t]$ again requires scanning through the linear factors and takes $(nd \log|\mathbb{F}|)^{O(1)}$. Black-box for $g = \Phi(f)/\prod_{i=1}^{t} x_i^{e_i}$ is then created by multiplying outputs of all the black-boxes for any input and therefore is also simulated in $(nd \log|\mathbb{F}|)^{O(1)}$ time. Therefore Step 1 takes $(nd \log|\mathbb{F}|)^{O(1)}$ time. Restrictions of black-box g to $x_i = 0, i \in [t]$ can be simulated by passing inputs through the restriction and therefore takes $(nd \log|\mathbb{F}|)^{O(1)}$ time. Time complexity of Algorithm 1 implies that factorization of $g_{|x_i=0}$ can be done in $(nd \log|\mathbb{F}|)^{O(1)}$ time. Running time of Algorithm 5 in [Shp07] is $(nd \log|\mathbb{F}|)^{O(1)}$ and therefore the multi-set $\mathcal{U}$ is created in $(nd \log|\mathbb{F}|)^{O(1)}$ time. Therefore Step 2 takes $(nd \log|\mathbb{F}|)^{O(1)}$ time overall. Step 3 involves iterating through the linear forms in $\mathcal{U}$, restricting them to $x_1 = 0$, giving multi-set $\mathcal{U}'$, and then comparing the $d^{O(1)}$ sized multi-sets $\mathcal{U}'$ and $\mathcal{U}$ to obtain the appropriate scalar α. All these steps can be executed in polynomial time leading to a time complexity of $(nd \log|\mathbb{F}|)^{O(1)}$ for Step 3. Black-box computing polynomial $g - \alpha \prod_{\ell \in \mathcal{U}} \ell$ can be simulated in $(nd \log|\mathbb{F}|)^{O(1)}$ time by going through each of the involved (black-boxes) polynomials and then computing the output after algebraic operations. Time complexity of Algorithm 1 implies that the factorization of this black-box can be done in $(nd \log|\mathbb{F}|)^{O(1)}$ time. Finally computing the black-box for f' and simulating black-box for $f - f'$ can similarly be done in $(nd \log|\mathbb{F}|)^{O(1)}$ time. By time complexity of algorithm in Lemma 4, we know that in time $(nd \log|\mathbb{F}|)^{O(1)}$, we can deterministically test whether $f - f'$ is the zero polynomial or not. Therefore Step 4 also takes time $(nd \log|\mathbb{F}|)^{O(1)}$. So, overall Algorithm 6 runs in time $(nd \log|\mathbb{F}|)^{O(1)}$. □

4.4 Identify Linearly Independent Set Dividing T_i

In this subsection, our goal is to provide proof of Lemma 21. It plays a crucial role in Algorithm 3 as explained in Section 1.2.2, by optimizing the search for a large linearly independent set of linear forms dividing one of T_1, T_2. As we mentioned earlier, [Shp07] compute such an independent set by using a brute force search (Algorithm 4, [Shp07]) on the space of linear forms over many variables, and therefore take quasi-polynomial time even before using this set in Algorithm 5 (of [Shp07]). We significantly improve the search using candidate linear forms $\mathcal{L}(NonLin(f))$ and ordinary lines (see Definition 4) among them. First, in Section 4.4.1 below we give intuition about why set $\mathcal{L}(NonLin(f))$ approximates the set of linear factors of $T_1 \times T_2$ and then in in Lemma 21, Section 4.4.2 use this set to construct the required linearly independent set.

4.4.1 Candidate set approximates set of linear forms dividing T_1, T_2

In order to quantify how close the candidate set $\mathcal{L}(NonLin(f))$ is to the set of linear forms in the input circuit, we define some new sets.

$$\mathcal{L}_{good} = \{\ell \in \mathcal{L}(NonLin(f)) : \ell \mid T_1 \times T_2\}, \qquad \mathcal{L}_{bad} = \mathcal{L}(NonLin(f)) \setminus \mathcal{L}_{good},$$

$$\mathcal{L}_{others} = \{\ell \mid T_1 \times T_2 : sp(\ell) \cap \mathcal{L}(NonLin(f)) = \phi\} \quad \text{and} \quad \mathcal{L}_{factors} = \{\ell : \ell \mid T_1 + T_2\}$$

For all sets, we only keep linear forms upto scalar multiplication and therefore treat them as proper sets (Definition 3). $\mathcal{L}_{good}$ contains all candidate linear forms which also divide one of the two gates T_1, T_2. $\mathcal{L}_{bad}$ are candidates which do not divide T_1 or T_2. $\mathcal{L}_{other}$ are linear forms dividing one of the gates but not captured (even up to scalar multiplication) in the candidate set and $\mathcal{L}_{factors}$ contain linear forms that divide $T_1 + T_2$. In the following claim, we show that $\mathcal{L}_{good}$ is high dimensional and $\mathcal{L}_{bad}, \mathcal{L}_{other}$ are low dimensional quantifying the closeness of $\mathcal{L}(NonLin(f))$ to the set of linear forms dividing $T_1 \times T_2$. We also show that $\mathcal{L}_{factors}$ is low dimensional. For better exposition, proof is provided in Appendix A.

Claim 3. *The following claim is true about these newly constructed sets.*

1. $dim(sp(\mathcal{L}_{factors})) \leq \log d + 2$,
2. $dim(sp(\mathcal{L}_{good})) \geq rank(f) - 2$,
3. $dim(sp(\mathcal{L}_{bad})) \leq \log d + 2$, *and*
4. $dim(sp(\mathcal{L}_{others})) \leq 2$.

4.4.2 Proof of Lemma 21

In this subsection, we prove Lemma 21 which was used by Algorithm 3. Recall that $rank(f) = \Omega(\log^3 d)$. We use definitions of $\mathcal{L}_{good}, \mathcal{L}_{bad}, \mathcal{L}_{other}, \mathcal{L}_{factors}$ given in Section 4.4.1. Recall the definition of the set of ordinary lines from Definition 4.

Lemma 21. *The following are true.*

1. $\exists\ \ell \in \mathcal{L}_{good}$ *such that the set of linear forms* $\ell' \in \mathcal{L}(NonLin(f)) \setminus \{\ell\}$ *for which* $sp\{\ell, \ell'\}$ *intersects* $\mathcal{L}(NonLin(f))$ *only at* $\{\ell, \ell'\}$[15], *spans a space of dimension at least* $\Omega(\log^2 d)$. *Let* $\mathcal{X}$ *be some maximal independent subset* $\Rightarrow |\mathcal{X}| = \Omega(\log^2 d)$.
2. *Every partition of* $\mathcal{X}$ *into* $\Omega(\log d)$ *equal parts of size* $\Omega(\log d)$ *each, contains a part* $\mathcal{B}$ *such that* $\mathcal{B} \subset \mathcal{L}_{good}$ *and for every* $\ell' \in \mathcal{B}$, $sp\{\ell, \ell'\}$ *is an ordinary line into* $\mathcal{L}_{good}, \mathcal{L}_{bad}, \mathcal{L}_{others}, \mathcal{L}_{factors}$.
3. *Let* $\ell' \in \mathcal{B}$ *and assume* $\ell \mid T_i$. *Let* Φ *be an isomorphism mapping* $\ell \mapsto x_1, \ell' \mapsto x_2$, *then,*

$$\Phi(NonLin(f))_{|x_1=0,x_2=0} = 0 \Leftrightarrow \ell' \text{ divides } T_{3-i}.$$

Proof. We prove all parts one by one.

[15] basically $sp\{\ell, \ell'\}$ is an ordinary line into $\mathcal{L}(NonLin(f))$.

1. Let $\mathcal{T} \subset \mathcal{L}_{good}$ be a linearly independent set of size $126 \log d + 2$ (exists by Claim 3). Applying Proposition 1 on $\mathcal{L}(NonLin(f))$ and $\mathcal{T}$ implies that there exists $\ell \in \mathcal{T}$ such that
$$dim(\sum_{W \in \mathcal{O}(\ell, \mathcal{L}(NonLin(f)))} W) \geq \frac{dim(sp(\mathcal{L}(NonLin(f))))}{126 \log d + 2} \geq \frac{dim(sp(\mathcal{L}_{good}))}{126 \log d + 2} = \Omega(\log^2 d)$$
Thus, the set of linear forms $\ell' \in \mathcal{L}(NonLin(f)) \setminus \{\ell\}$ for which $sp\{\ell, \ell'\}$ intersects $\mathcal{L}(NonLin(f))$ only at $\{\ell, \ell'\}$, spans a space of dimension at least $\Omega(\log^2 d)$. Let $\mathcal{X}$ be a maximal independent subset $\Rightarrow |\mathcal{X}| = \Omega(\log^2 d)$.

2. Consider any partition of $\mathcal{X}$ into $\Omega(\log d)$ parts of size $\Omega(\log d)$ each.
   (a) We first claim that $\Omega(\log d)$ parts in this partition are inside $\mathcal{L}_{good}$. If not, then $\Omega(\log d)$ parts intersect $\mathcal{L}_{bad} \Rightarrow dim(sp(\mathcal{L}_{bad})) = \Omega(\log d)$, contradicting Claim 3. Now we will only deal with these $\Omega(\log d)$ parts inside $\mathcal{L}_{good}$. Since $\mathcal{L}_{good}, \mathcal{L}_{bad} \subset \mathcal{L}(NonLin(f))$, we see that for all ℓ' in any of these parts $sp\{\ell, \ell'\}$ is an ordinary line in $\mathcal{L}_{good}, \mathcal{L}_{bad}$ as required.
   (b) Next we show that out of the $\Omega(\log d)$ parts inside $\mathcal{L}_{good}$, there is a part $\mathcal{B}$ such that for all $\ell' \in \mathcal{B}$, $sp\{\ell, \ell'\}$ is an ordinary line in $\mathcal{L}_{others}, \mathcal{L}_{factors}$, thereby completing the proof. If not then there are $\Omega(\log d)$ many ℓ''s, each belonging to a different part among the $\Omega(\log d)$ parts, such that $sp\{\ell, \ell'\}$ intersects $\mathcal{L}_{others} \cup \mathcal{L}_{factors}$ at a linear form outside $sp\{\ell\} \cup sp\{\ell'\}$ say ℓ''. Since all the $\Omega(\log d)$ ℓ's are independent, the ℓ''s span a space of dimension $\Omega(\log d) \Rightarrow dim(sp(\mathcal{L}_{others} \cup \mathcal{L}_{factors})) = \Omega(\log d)$, contradicting Claim 3.

   Therefore, we have shown the existence of a part $\mathcal{B}$ as desired.

3. Since $\ell \mid T_i$, we know that $x_1 \mid \Phi(T_i)$. Therefore, the following equation holds in $\mathbb{F}[x_3, \ldots, x_n]$.
$$\Phi(L)_{|x_1=0,x_2=0} \Phi(NonLin(f))_{|x_1=0,x_2=0} = \Phi(T_i)_{|x_1=0,x_2=0} + \Phi(T_{3-i})_{|x_1=0,x_2=0} = \Phi(T_{3-i})_{|x_1=0,x_2=0}.$$
Here L is the product of all linear factors of $T_1 + T_2$ i.e. $L = Lin(T_1 + T_2)$. First, we assume that $\Phi(NonLin(f))_{|x_1=0,x_2=0} = 0$. This implies using the above equation that $\Phi(T_{3-i})_{|x_1=0,x_2=0} = 0$. Therefore there is a linear form $\ell'' \mid T_{3-i}$ such that $\ell'' \in sp\{\ell, \ell'\}$. If ℓ'' is not a scalar multiple of ℓ or ℓ', by construction of ℓ, ℓ' in parts 1 and 2 of this lemma, we know that no scalar multiple of ℓ'' can belong to $\mathcal{L}_{good}$ or $\mathcal{L}_{others}$ and therefore it cannot divide $T_1 \times T_2$ which is a contradiction since it divides T_{3-i}. Therefore, ℓ'' has to be a scalar multiple of ℓ or ℓ'. It cannot be scalar multiple of ℓ since $\ell \mid T_i$ and $gcd(T_i, T_{3-i}) = 1$. Therefore ℓ'' and ℓ' are scalar multiples implying that ℓ' divides T_{3-i} as needed. Next, for the converse, we assume that $\ell' \mid T_{3-i}$. Again, using the equation we gave at the beginning of this part, we get that,
$$\Phi(L)_{|x_1=0,x_2=0} \Phi(NonLin(f))_{|x_1=0,x_2=0} = \Phi(T_i)_{|x_1=0,x_2=0} + \Phi(T_{3-i})_{|x_1=0,x_2=0} = 0.$$
Therefore, since $\mathbb{F}[x_3, \ldots, x_n]$ is an integral domain, either polynomial $\Phi(L)_{|x_1=0,x_2=0} = 0$ or polynomial $\Phi(NonLin(f))_{|x_1=0,x_2=0} = 0$. Assume that $\Phi(L)_{|x_1=0,x_2=0} = 0$. This implies that there is some linear factor ℓ'' of $T_1 + T_2$ such that $\ell'' \in sp\{\ell, \ell'\}$. Since $gcd(T_1, T_2) = 1$ and $\ell \mid T_i$, $\ell' \mid T_{3-i}$, the linear form ℓ'' cannot be a scalar multiple of ℓ or ℓ'. So we found a linear form on $sp\{\ell, \ell'\}$ different from scalar multiples of ℓ, ℓ', such that some scalar multiple of ℓ'' belongs to $\mathcal{L}_{factors}$. By construction of ℓ, ℓ' in parts 1 and 2 of this lemma, we know that this cannot hold. Therefore our assumption is wrong and polynomial $\Phi(NonLin(f))_{|x_1=0,x_2=0} = 0$ completing the proof.

$\square$

5 Proof of Proposition 1

In this section we prove Proposition 1. Part 1 is proved in Section 5.1. Algorithm proving Part 2 is presented in Algorithm 7 and it's correctness/complexity are analyzed in Section 5.2.

5.1 Proof of Part 1

Let $W = \mathbb{V}(\ell, \ell') \subset \mathbb{F}^n$ be a co-dimension 2 subspace on which $NonLin(f)$ vanishes i.e. $W \in \mathcal{S}(NonLin(f))$. Let Φ be an isomorphism mapping $\ell \mapsto x_1, \ell' \mapsto x_2$. Since $NonLin(f)$ divides $T_1 + T_2$ we get that $\Phi(T_1)_{|x_1=0,x_2=0} + \Phi(T_2)_{|x_1=0,x_2=0} = 0$. This implies that either $\Phi(T_1)_{|x_1=0,x_2=0} = \Phi(T_2)_{|x_1=0,x_2=0} = 0$, or $\Phi(T_1)_{|x_1=0,x_2=0} = -\Phi(T_2)_{|x_1=0,x_2=0} \neq 0$. We prove the following lemma which implies the bound.

Lemma 22. *The following are true.*

1. $\#\{W \in \mathcal{S}(NonLin(f)) : \Phi(T_1)_{|x_1=0,x_2=0} = \Phi(T_2)_{|x_1=0,x_2=0} = 0\} \leq d^2$.
2. $\#\{W \in \mathcal{S}(NonLin(f)) : \Phi(T_1)_{|x_1=0,x_2=0} = -\Phi(T_2)_{|x_1=0,x_2=0} \neq 0\} \leq d^5 + d^7$.

Proof of Part 1: The statement implies that there are linear forms $\ell_1 \mid T_1$ and $\ell_2 \mid T_2$ such that $\Phi(\ell_1)_{|x_1=0,x_2=0} = \Phi(\ell_2)_{|x_1=0,x_2=0} = 0$. Also, ℓ_1, ℓ_2 are linearly independent since $gcd(T_1, T_2) = 1$ implying that $sp\{\Phi(\ell_1), \Phi(\ell_2)\} = sp\{x_1, x_2\}$. On inverting via Φ this implies that $sp\{\ell_1, \ell_2\} = sp\{\ell, \ell'\}$, which further implies that $\mathbb{V}(\ell_1, \ell_2) = \mathbb{V}(\ell, \ell') = W$. There can be at most d^2 such W's completing the proof.

Proof of Part 2: We use the following lemma to prove this part. For clarity of presentation, we move it's proof to Appendix B.

Lemma 23. *There exists a set $\mathcal{A}$ of co-dimension 1 subspaces of $\mathbb{F}^n$ with $|\mathcal{A}| \leq d^4 + d^6$ such that for every $W \in \mathcal{S}(NonLin(f))$ satisfying $\Phi(T_1)_{|x_1=0,x_2=0} = -\Phi(T_2)_{|x_1=0,x_2=0} \neq 0$, $\exists\ V \in \mathcal{A}$ with $W \subset V$.*

Assuming Lemma 23, we complete the proof as follows. For every $W \in \mathcal{S}(NonLin(f))$ satisfying $\Phi(T_1)_{|x_1=0,x_2=0} = -\Phi(T_2)_{|x_1=0,x_2=0} \neq 0$, we consider the co-dimension 1 subspace V given by Lemma 23 such that $W \subset V$. Without loss of generality we assume $V = \mathbb{V}(x_1)$. We can now find a linear form ℓ_3 such that $W = \mathbb{V}(x_1, \ell_3)$ and coeffcient of x_1 in ℓ_3 is 0 i.e. $\ell_3 = \ell_{3|x_1=0}$. Since $NonLin(f)$ vanishes on W we know that $\Psi(NonLin(f))_{|x_1=0,x_2=0}$ for isomorphism Ψ mapping $x_1 \mapsto x_1, \ell_3 \mapsto x_2$. This also implies that x_2 divides $\Psi(NonLin(f))_{|x_1=0}$. Since Ψ keeps x_1 fixed this polynomial is same as $\Psi(NonLin(f)_{|x_1=0})$. Inverting Ψ we get that ℓ_3 divides $NonLin(f)_{|x_1=0}$. There are at most d linear factors (upto scalar multiplication) of any degree d polynomial, thus there are $\leq d$ such possible ℓ_3. By going ever all choices of V we get that there are at most $(d^4 + d^6) \times d$ many such W, completing our proof.

5.2 Analysis of Algorithm 7

We first give the algorithm and then discuss it's correctness and time complexity.

Algorithm 7 Compute co-dimension 2 subspaces on which $NonLin(f)$ vanishes

Input - Black-box access to polynomial f, integers n, d.

Output - A set $\mathcal{S}$ of tuples of independent linear forms in $\mathbb{F}[x_1, \ldots, x_n]$.

1. Create n linear forms $\hat{\ell}_1, \ldots, \hat{\ell}_n$, such that the n^2 scalars used as coefficients in them are sampled uniformly randomly independently from $\mathbb{F}$. If these linear forms are linearly independent, define isomorphism Φ mapping $x_i \mapsto \hat{\ell}_i, i \in [n]$. Simulate black-box for $g = \Phi(f)$. For $i \in [5, n]$, simulate black-box access for the following restricted polynomials in $\mathbb{F}[x_1, x_2, x_3, x_4, x_i]$.

$$g_i = g_{|x_5=0,\ldots,x_{i-1}=0,x_{i+1}=0,\ldots,x_n=0}$$

   Next, for each $i \in [5, n]$ using Algorithm 1 with inputs as black-box access to g_i along with integers $5, d$ obtain black-box access to $NonLin(g_i)$ and integer s denoting the number of linear factors of g_i. Define $t = d - s$. Using multivariate interpolation (Lemma 6), interpolate $NonLin(g_i)$ as a degree t polynomial in the monomial basis of $\mathbb{F}[x_1, x_2, x_3, x_4, x_i]$.

2. Substitute $x_1 = y_3x_3 + y_4x_4 + y_ix_i$, and $x_2 = z_3x_3 + z_4x_4 + z_ix_i$ in $NonLin(g_i)$ to obtain a polynomial in $\mathbb{F}[y_3, y_4, y_i, z_3, z_4, z_i][x_3, x_4]$. Find common solutions to the system of polynomial equations defined by setting all coefficient polynomials ($\in \mathbb{F}[y_3, y_4, y_i, z_3, z_4, z_i]$) to zero. Initialize a set $\mathcal{S}_i \leftarrow \phi$ and for each solution $(y_3, y_4, y_i, z_3, z_4, z_i)$ of the system above add tuple $(x_1 - y_3x_3 - y_4x_4 - y_ix_i, x_2 - z_3x_3 - z_4x_4 - z_ix_i)$ to $\mathcal{S}_i$.

3. Construct isomorphism Δ mapping $x_1 \mapsto x_1, x_2 \mapsto x_2, x_3 \mapsto x_3, x_4 \mapsto x_4$ and for $i \in [5.n]$, $x_i \mapsto x_i + \alpha_{i,3}x_3 + \alpha_{i,4}x_4$. The scalars $\alpha_{i,3}, \alpha_{i,4}, i \in [5, n]$ are sampled uniformly randomly independently from $\mathbb{F}$. Note that Δ can be viewed as an isomorphism on $\mathbb{F}[x_1, \ldots, x_n]$ as well as on each $\mathbb{F}[x_1, x_2, x_3, x_4, x_i]$ for $i \in [5, n]$.

4. Initialize a set $\mathcal{S} \leftarrow \phi$. Iterate over all tuples $(x_1 - \ell_1^5, x_2 - \ell_2^5) \in \mathcal{S}_5$. Initialize $\ell_1 \leftarrow \ell_1^5, \ell_2 \leftarrow \ell_2^5$. Iterate over $i \in [6, n]$. Search for tuple $(x_1 - \ell_1^i, x_2 - \ell_2^i) \in \mathcal{S}_i$ such that tuples

$$(x_1 - \Delta(\ell_1^5)_{|x_5=0}, x_2 - \Delta(\ell_2^5)_{|x_5=0}) = (x_1 - \Delta(\ell_1^i)_{|x_i=0}, x_2 - \Delta(\ell_2^i)_{|x_i=0})$$

   If multiple or none such tuples are found in $\mathcal{S}_i$ then break out of this loop and go to the next tuple in the outer iteration. If only one such tuple is found then update $\ell_1 \leftarrow \ell_1 - \alpha x_i$ and $\ell_2 \leftarrow \ell_2 - \beta x_i$ where α, β are coefficients of x_i in $x_1 - \ell_1^i, x_2 - \ell_2^i$ respectively. At the end of iteration on i, update $\mathcal{S} \leftarrow \mathcal{S} \cup \{(x_1 - \ell_1, x_2 - \ell_2)\}$.

5. For each $(\ell_1, \ell_2) \in \mathcal{S}$, construct isomorphism Ψ mapping $\ell_1 \mapsto x_1, \ell_2 \mapsto x_2$. Simulate black-box access to polynomial

$$\Psi(NonLin(g))_{|x_1=0,x_2=0}.$$

   Using randomized polynomial identity test given in Lemma 3 with input as the above black-box and integer n, check if it is identically the zero polynomial. If 'no', remove the tuple from $\mathcal{S}$, else replace it with $(\Phi^{-1}(\ell_1), \Phi^{-1}(\ell_2))$. **Return** $\mathcal{S}$.

Before going to the correctness of the above algorithm, we state a few useful lemmas. These are repeatedly used in our correctness and time complexity proofs.

Lemma 24. *With probability $1 - o(1)$ over the random choices in Step 1, the following hold.*

1. *The n linear forms constructed in Step 1 with the random coefficients are linearly independent.*
2. *$NonLin(f)$ vanishes on $\mathbb{V}(\ell_1, \ell_2)$ if and only if $NonLin(g)$ vanishes on $\mathbb{V}(\Phi(\ell_1), \Phi(\ell_2))$.*
3. *Polynomial g_i has a $\Sigma\Pi\Sigma(2, 5, d, \mathbb{F})$ circuit and $rank(g_i) = 5$.*
4. *$NonLin(g_i) = NonLin(g)_{|x_5=0,\ldots,x_{i-1}=0,x_{i+1}=0,\ldots,x_n=0}$.*
5. *For all $\mathbb{V}(\ell_1, \ell_2) \in \mathcal{S}(NonLin(g))$, there exist linear forms $\ell_1', \ell_2' \in \mathbb{F}[x_3, \ldots, x_n]$ such that*
$$\mathbb{V}(\ell_1, \ell_2) = \mathbb{V}(x_1 - \ell_1', x_2 - \ell_2')$$
6. *Let $\mathbb{V}(x_1 - \ell_1, x_2 - \ell_2) \in \mathcal{S}(NonLin(g))$ with $\ell_1, \ell_2 \in \mathbb{F}[x_3, \ldots, x_n]$. Then, $NonLin(g_i)$ vanishes on the co-dimension 2 subspace $\mathbb{V}(x_1 - \ell_1^i, x_2 - \ell_2^i)$. Here $\ell_j^i = \ell_{j|x_5=0,\ldots,x_{i-1}=0,x_{i+1}=0,\ldots,x_n=0}$.*

Lemma 25. *With probability $1 - o(1)$ over the random choices in Step 3, the following holds. For all $i \in [5, n]$ and for all pairs of distinct tuples $(x_1 - \ell_1, x_2 - \ell_2)$, $(x_1 - \ell_1', x_2 - \ell_2')$ in $\mathcal{S}_i$,*

$$(x_1 - \Delta(\ell_1)_{|x_i=0}, x_2 - \Delta(\ell_2)_{|x_i=0}) \neq (x_1 - \Delta(\ell_1')_{|x_i=0}, x_2 - \Delta(\ell_2')_{|x_i=0})$$

For better presentation we prove these lemmas in Appendix C. Now, we prove correctness of Algorithm 7. By Part 1 of Lemma 24, the linear forms constructed in Step 1 are linearly independent and therefore isomorphism Φ can be correctly constructed using them. Using this isomorphism, simulation of black-box for g (by passing every input through the isomorphism) is straight forward. Further simulation of black-boxes computing the g_is is also straight forward (by setting $x_5 = 0, \ldots, x_{i=1} = 0, x_{i+1} = 0, \ldots, x_n = 0$ in the input to black-box). From Parts $4, 5$ of Lemma 24, we know that g_i exhibits $\Sigma\Pi\Sigma(2, 5, d, \mathbb{F})$ circuit of rank 5 and $NonLin(g_i) = NonLin(g)_{|x_5=0,\ldots,x_{i=1}=0,x_{i+1}=0,\ldots,x_n=0}$, implying that all g_i and g have the same number of linear factors s and degree of all polynomials $NonLin(g_i)$ are equal ($= t$) which is also the same as degree of $NonLin(g)$. By correctness of Algorithm 1, with probability $1 - o(1)$, Step 1 correctly obtains black-box computing $NonLin(g_i)$ and it's degree t. Since all g_i are $5-$ variate using deterministic multivariate interpolation (Lemma 6), we can interpolate their black-boxes as degree t polynomials in the monomial basis of $\mathbb{F}[x_1, x_2, x_3, x_4, x_i]$. Therefore, at the end of Step 1, we would have correct monomial representations of all the g_i. Next, using Part 5 of Lemma 24, we know that any co-dimension 2 subspace on which $NonLin(g)$ vanishes has the form $\mathbb{V}(x_1 - \ell_1, x_2 - \ell_2)$ with $\ell_1, \ell_2 \in \mathbb{F}[x_3, \ldots, x_n]$. In Part 6 of Lemma 24, we show that $NonLin(g_i)$ vanishes on the co-dimension 2 space $\mathbb{V}(x_1 - \ell_1^i, x_2 - \ell_2^i)$, where for $j \in [2]$ and $i \in [5, n]$, ℓ_j^i are restrictions of ℓ_j to $x_5 = 0, \ldots, x_{i-1} = 0, x_{i+1} = 0, \ldots, x_n = 0$. Since these co-dimension 2 subspaces have the particular form $\mathbb{V}(x_1 - \ell_1^i, x_2 - \ell_2^i)$, substituting $x_1 = \ell_1^i, x_2 = \ell_2^i$ in $NonLin(g_i)$ should give 0. Step 2 uses this observation and computes all possible ℓ_1^i, ℓ_2^i by solving the system of polynomial equations we get on substitution. By correctness of Lemma 1, we can compute all such solutions. Therefore, the set $\mathcal{S}_i$ contain tuples corresponding to all co-dimension 2 spaces of the form $\mathbb{V}(x_1 - u_1, x_2 - u_2)$ (with linear forms $u_1, u_2 \in \mathbb{F}[x_3, x_4, x_i]$) on which $NonLin(g_i)$ vanishes. In the next lemma, we show that these $\mathcal{S}_i$ are then glued in Steps 3 and 4 to create a set $\mathcal{S}$ which contains tuples corresponding to all elements of $\mathcal{S}(NonLin(f))$.

Lemma 26. *Step* 4 *outputs a set* $\mathcal{S}$*, such that with probability* $1-o(1)$*, it contains tuples of linear forms representing all co-dimension* 2 *subspaces on which* $NonLin(g)$ *vanishes.*

Proof. Let $\mathbb{V}(x_1-\ell_1, x_2-\ell_2) \in \mathcal{L}(NonLin(g))$. By Part 6 of Lemma 24 we know that $NonLin(g_i)$ vanishes on the co-dimension 2 subspace $\mathbb{V}(x_1-\ell_1^i, x_2-\ell_2^i)$ where for $j \in [2]$, $\ell_j^i = \ell_{j|_{x_5=0,\dots,x_{i-1}=0,x_{i+1}=0,\dots,x_n=0}}$. Therefore the tuples $(x_1-\ell_1^i, x_2-\ell_2^i)$ belong to $\mathcal{S}_i$ computed at Step 2. Observe that, for $i \in [6,n]$ we glue tuple $(x_1-\ell_1^5, x_2-\ell_2^5)$ with tuple $(x_1-\ell_1^i, x_2-\ell_2^i)$ only if the latter is the only tuple in $\mathcal{S}_i$ satsfying, $(x_1-\Delta(\ell_1^5)_{|_{x_5=0}}, x_2-\Delta(\ell_2^5)_{|_{x_5=0}}) = (x_1-\Delta(\ell_1^i)_{|_{x_i=0}}, x_2-\Delta(\ell_2^i)_{|_{x_i=0}})$. Here Δ is the isomorphism constructed in Step 3. So all we need to show is that, there is no other tuple $(x_1-\ell_1^{i\,\prime}, x_2-\ell_2^{i\,\prime}) \in \mathcal{S}_i$ with $\ell_1^{i\,\prime}, \ell_2^{i\,\prime}$ being linear forms in $\mathbb{F}[x_3,x_4,x_i]$ such that, $(x_1-\Delta(\ell_1^5)_{|_{x_5=0}}, x_2-\Delta(\ell_2^5)_{|_{x_5=0}}) = (x_1-\Delta(\ell_1^{i\,\prime})_{|_{x_i=0}}, x_2-\Delta(\ell_1^{2\prime})_{|_{x_i=0}})$. If there was such a tuple, comparing the two equations we got above gives $(x_1-\Delta(\ell_1^i)_{|_{x_i=0}}, x_2-\Delta(\ell_2^i)_{|_{x_i=0}}) = (x_1-\Delta(\ell_1^{i\,\prime})_{|_{x_i=0}}, x_2-\Delta(\ell_1^{2\prime})_{|_{x_i=0}})$, which contradicts Lemma 25. Therefore tuple $(x_1-\ell_1^5, x_2-\ell_2^5)$ gets correctly glued with each such tuple $(x_1-\ell_1^i, x_2-\ell_2^i)$ for $i \in [6,n]$ leading to the tuple $(x_1-\ell_1, x_2-\ell_2)$ being constructed and added to $\mathcal{S}$. Hence Proved. $\square$

Assuming we have correctly glued the $\mathcal{S}_i$ into set $\mathcal{S}$, Step 5, performs a final pruning by retaining tuples for which $NonLin(g)$ actually vanishes on the co-dimension 2 subspace they represent. By correctness of Lemma 3, this is done correctly and only the right tuples are retained. By Part 1 of Lemma 24, in order to get set $\mathcal{S}(NonLin(f))$ from $\mathcal{S}(NonLin(g))$, we only need to invert all linear forms present in the elements (tuples) of $\mathcal{S}$. Therefore, with probability $1-o(1)$, the set of tuples representing co-dimension 2 subspaces on which $NonLin(f)$ vanishes is correctly computed. Now we discuss the time complexity of the above algorithm.

Lemma 27. *Algorithm 7 runs in* $(nd\log|\mathbb{F}|)^{O(1)}$ *time.*

Proof. Assuming that sampling of a uniformly random scalar from $\mathbb{F}$ takes $O(1)$ time, the n linear forms are created in $(n\log|\mathbb{F}|)^{O(1)}$ time. Checking whether the linear are independent can be done in $(n\log|\mathbb{F}|)^{O(1)}$ time by gaussian elimination on the matrix defined by the n^2 coefficients of these linear forms. Black-boxes for g and g_i are simulated in $(n\log|\mathbb{F}|)^{O(1)}$ time by passing each input through Φ and then restricting to $x_5 = 0, \dots, x_{i-1=0}, x_{i+1} = 0, \dots, x_n = 0$. Time complexity of Algorithm 1 implies that black-box access to all $NonLin(g_i)$ along with their degrees $t = d-s$ can be obtained in $(nd\log|\mathbb{F}|)^{O(1)}$ time. Multivariate interpolation (Lemma 6) on the 5 variate polynomials of degree t each is done in $(nd\log|\mathbb{F}|)^{O(1)}$ time. Therefore Step 1 takes $(nd\log|\mathbb{F}|)^{O(1)}$ time. Each g_i has $d^{O(1)}$ non-zero coefficients in the monomial representation. Substitutions lead to $d^{O(1)}$ many coefficient polynomials in $\mathbb{F}[y_3,y_4,y_i,z_3,z_4,z_i]$ with every polynomial having degree $d^{O(1)}$. By Part 2 of Lemma 24, every g_i has a $\Sigma\Pi\Sigma(2,5,d,\mathbb{F})$ circuit and has rank 5, therefore, by Part 1 of Proposition 1, number of co-dimension 2 subspaces on which they vanish are $d^{O(1)}$. Therefore our system of equations has at most $d^{O(1)}$ solutions since they characterize such co-dimension 2 subspaces of a certain form. By time complexity of Lemma 1, for each g_i all solutions to such a system can be computed in $(d\log|\mathbb{F}|)^{O(1)}$ time leading to $\mathcal{S}_i$. Therefore in time $(nd\log|\mathbb{F}|)^{O(1)}$ time all $\mathcal{S}_i$ are computed in Step 2. Step 3 involves sampling $O(n)$ many uniformly random scalars and construction of the isomorphism Δ can be done in $(n\log|\mathbb{F}|)^{O(1)}$ time. In Step 4, we iterate over all tuples in $\mathcal{S}_5$ and then iterate over $i \in [6,n]$ trying to match our tuple with tuples in the $\mathcal{S}_i$. Since each tuple in $\mathcal{S}_5$ is matched to at most one tuple in each $\mathcal{S}_i$, for each tuple in $\mathcal{S}_5$, we go over all the set $\mathcal{S}_i, i \in [6, n]$ just once. Therefore, overall we take $(nd \log |\mathbb{F}|)^{O(1)}$ time in this step. Also, since each tuple in $\mathcal{S}_5$, creates at most one tuple $(x_1 - \ell_1, x_2 - \ell_2)$ to be added to $\mathcal{S}$, we create at most $d^{O(1)}$ such tuples leading to $|\mathcal{S}| = d^{O(1)}$. In Step 5, for each tuple in $\mathcal{S}$, construction of isomorphism Ψ and black-box access to $\Psi(NonLin(g))_{|x_1=0, x_2=0}$ can be created in $(nd \log |\mathbb{F}|)^{O(1)}$ time. By time complexity of algorithm in Lemma 3, in time $(nd \log |\mathbb{F}|)^{O(1)}$ we can check whether this black-box computes the 0 polynomial or not. Finally application of Φ^{-1} to tuples in $\mathcal{S}$ can be done in $(nd \log |\mathbb{F}|)^{O(1)}$ time. Our final set returned has size $d^{O(1)}$ as it is a subset of the set we created in Step 4. Therefore, overall Algorithm 7 takes $(nd \log |\mathbb{F}|)^{O(1)}$ time. □

6 Proof of Proposition 2

In this section we present our proof of Proposition 2. The proof is immediately implied by Lemma 28 which is itself proved using Lemma 29. Recall definition of set of ordinary lines(Definition 4).

Lemma 28. *Let $\mathcal{S} \subset \mathbb{F}^n$ be a proper set (Definition 3) and $\mathcal{T} \subset \mathbb{F}^n$ be any linearly independent set of size $\log |\mathcal{S}| + 2$. Then, the following holds.*

$$sp(\mathcal{S}) \subseteq \sum_{t \in \mathcal{T}} \sum_{W \in \mathcal{O}(t, \mathcal{S})} W$$

Proof of Proposition 2 using Lemma 28: By simply taking dimension of both sides in the containment, applying union bound on the right hand side and assuming $t \in \mathcal{T}$ maximizes $dim(\sum_{W \in \mathcal{O}(t,\mathcal{S})} W))$, we get

$$dim(\sum_{W \in \mathcal{O}(t,\mathcal{S})} W) \geq \frac{dim(sp(\mathcal{S}))}{\log |\mathcal{S}| + 2}.$$

which proves Proposition 2. So we are left with proving Lemma 28.

Proof of Lemma 28 Let V be the vector space $\sum_{t \in \mathcal{T}} \sum_{W \in \mathcal{O}(t,\mathcal{S})} W$. We define set $\mathcal{S}' = \mathcal{S} \setminus V$. $\mathcal{S}'$ is a proper set. We will show that $\mathcal{S}' = \phi \Rightarrow sp(\mathcal{S}) \subset V$. If not, we show that there cannot be any ordinary line from $\mathcal{T}$ into $\mathcal{S}'$. Suppose there is some such line $sp\{t, s\}$ where $t \in \mathcal{T}$ and $s \in \mathcal{S}'$ are not scalar multiples. Since it is an ordinary line into $\mathcal{S}'$, we get that $sp\{s, t\} \cap \mathcal{S}' \subset sp\{s\} \cup sp\{t\}$. Then one of the following mutually exclusive statements will obviously be true.

1. $sp\{s, t\} \cap V \subset sp\{s\} \cup sp\{t\}$
2. $sp\{s, t\} \cap V \notin sp\{s\} \cup sp\{t\}$

In the first case, since $\mathcal{S} = \mathcal{S}' \cup (\mathcal{S} \cap V) \Rightarrow sp\{s, t\} \cap \mathcal{S} \subset sp\{s\} \cup sp\{t\}$. Therefore it is an ordinary line into $\mathcal{S}$. But all such lines are subsets of $V \Rightarrow s \in V$ which is a contradiction since $s \in \mathcal{S}'$ which is disjoint from V. In the second case, there is some $v \in sp\{s, t\} \cap V$ such that $v \notin sp\{s\} \cup sp\{t\}$. Therefore t, s, v are linearly dependent but t, s and s, v are not $\Rightarrow s \in sp\{t, v\}$. Both t, v are in V by construction and thus $s \in V$ which is again a contradiction since $s \in \mathcal{S}'$ which is disjoint from V. Therefore if $\mathcal{S}'$ is non-empty, there are no ordinary lines from $\mathcal{T}$ into $\mathcal{S}$. Now we use Lemma 29 and complete the proof. We will prove Lemma 29 after the current proof.

Lemma 29. *Let $\mathcal{S}(\neq \phi) \subset \mathbb{F}^n$ be a proper set and $\mathcal{T} \subset \mathbb{F}^n$ be linearly independent such that for every $t \in \mathcal{T}$, there is no ordinary line (Definition 4) from t into $\mathcal{S}$. Then $|\mathcal{T}| \leq \log |\mathcal{S}| + 1$.*

Using Lemma 29 with $\mathcal{S}'$ and $\mathcal{T}$, we get that $\log|\mathcal{S}|+2 = |\mathcal{T}| \leq \log|\mathcal{S}'|+1$ which is a contradiction since $\mathcal{S}' \subset \mathcal{S}$. Therefore, the only conclusion left is $\mathcal{S}' = \phi$, which completes the proof of our lemma as explained earlier.

Proof of Lemma 29: Let $|\mathcal{T}| = d$ and $|\mathcal{S}| = m$. We present a counting argument by building a one-to-one function mapping subsets of $[d-1]$ into $\mathcal{S}$. Such a function implies that $m \geq 2^{d-1}$ and we'll be done. The following describes this one-to-one function. Fix an element $s \in \mathcal{S}$ and let $\mathcal{T} = \{t_1, \ldots, t_d\}$. Without loss of generality we may assume that $s, t_1, \ldots, t_{d-1}$ are linearly independent.

Claim 4. *For any subset $\mathcal{P} \subset [d-1]$, there exists $s_{\mathcal{P}} \in \mathcal{S}$ in the interior*[16] *of $sp\{\{t_i : i \in \mathcal{P}\} \cup \{s\}\}$.*

Proof. We prove by induction on $|\mathcal{P}|$. For $|\mathcal{P}| = 0$, define $s_{\mathcal{P}} = s$ and we are done. Let's assume the claim is true for $|\mathcal{P}| = k-1$. We prove it for $|\mathcal{P}| = k$. Consider any element $p \in \mathcal{P}$ and let $\mathcal{R} = \mathcal{P} \setminus \{p\}$. By induction, we know there exists $s_{\mathcal{R}}$ in the interior of $sp\{\{t_i : i \in \mathcal{R}\} \cup \{s\}\}$. Since there is no ordinary line from any $t \in \mathcal{T}$ into $\mathcal{S}$, the line $sp\{t_p, s_{\mathcal{R}}\}$ contains $s_{\mathcal{P}} \in \mathcal{S}$ such that $s_{\mathcal{P}} \notin sp\{t_p\} \cup sp\{s_{\mathcal{R}}\} \Rightarrow s_{\mathcal{P}} = \alpha t_p + \beta s_{\mathcal{R}}$ with $\alpha, \beta \in \mathbb{F}$ being non-zero scalars $\Rightarrow s_{\mathcal{P}}$ is in the interior of $sp\{\{t_i : i \in \mathcal{P}\} \cup \{s\}\}$ and the proof is complete. $\square$

We can see that the function mapping $\mathcal{P} \subset [d-1]$ to $s_{\mathcal{P}} \in \mathcal{S}$, is one-to-one since for sets $\mathcal{P}, \mathcal{Q} \subset [d-1]$, which differ at some $j \in [d-1]$, exactly one of $s_{\mathcal{P}}, s_{\mathcal{Q}}$ has a non-zero coefficient of t_j, implying they are different. This completes the proof.

7 Acknowledgements

We would like to thank Vineet Nair for helping with organization and presentation of the paper. He also provided multiple insights about the content which led to better presentation. We would also like to thank Neeraj Kayal and Chandan Saha for helpful comments on an early presentation of this work. Neeraj Kayal introduced the author to black-box reconstruction problems for depth three circuits. The simple idea behind proof of Lemma 29, presented in this paper was shared with the author by Neeraj Kayal during a discussion. We would also like to thank Anuja Sharan for proofreading and helping in preparation of this paper.

References

[Ang88] Dana Angluin. Queries and concept learning. *Mach. Learn.*, 2(4):319–342, April 1988.

[DS05] Zeev Dvir and Amir Shpilka. Locally decodable codes with 2 queries and polynomial identity testing for depth 3 circuits. In *Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing*, STOC '05, page 592–601, New York, NY, USA, 2005. Association for Computing Machinery.

[Dvi12] Zeev Dvir. Incidence theorems and their applications. *Foundations and Trends® in Theoretical Computer Science*, 6(4):257–393, 2012.

[16] "interior" means that when $s_{\mathcal{P}}$ is written as a linear combination of $\{\{t_i : i \in \mathcal{P}\} \cup \{s\}\}$, all coefficients are non-zero

[GKL12] Ankit Gupta, Neeraj Kayal, and Satyanarayana V. Lokam. Reconstruction of depth-4 multilinear circuits with top fan-in 2. In *Proceedings of the 44th Symposium on Theory of Computing Conference, STOC 2012, New York, NY, USA, May 19 - 22, 2012*, pages 625–642, 2012.

[Ier89] Douglas John Ierardi. *The Complexity of Quantifier Elimination in the Theory of an Algebraically Closed Field.* PhD thesis, Cornell University, USA, 1989. AAI9001370.

[Kal91] Erich Kaltofen. Effective noether irreducibility forms and applications. In *Proceedings of the Twenty-third Annual ACM Symposium on Theory of Computing*, STOC '91, pages 54–63, New York, NY, USA, 1991. ACM.

[KS01] Adam R. Klivans and Daniel Spielman. Randomness efficient identity testing of multivariate polynomials. In *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing*, STOC '01, page 216–223, New York, NY, USA, 2001. Association for Computing Machinery.

[KS03] Adam R. Klivans and Amir Shpilka. Learning arithmetic circuits via partial derivatives. In Bernhard Schölkopf and Manfred K. Warmuth, editors, *Learning Theory and Kernel Machines*, pages 463–476, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[KS09] Zohar S. Karnin and Amir Shpilka. Reconstruction of generalized depth-3 arithmetic circuits with bounded top fan-in. In *Proceedings of the 2009 24th Annual IEEE Conference on Computational Complexity*, CCC '09, pages 274–285, Washington, DC, USA, 2009. IEEE Computer Society.

[KS18] Neeraj Kayal and Chandan Saha. Reconstruction of non-degenerate homogeneous depth three circuits. In *STOC*, 2018.

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

[KT90] Erich Kaltofen and Barry M. Trager. Computing with polynomials given by black boxes for their evaluations: Greatest common divisors, factorization, separation of numerators and denominators. *J. Symb. Comput.*, 9:301–320, 1990.

[Laz01] Daniel Lazard. Solving systems of algebraic equations. *SIGSAM Bull.*, 35(3):11–37, September 2001.

[MP13] Gary L. Mullen and Daniel Panario. *Handbook of Finite Fields.* Chapman & Hall/CRC, 1st edition, 2013.

[Sax09] Nitin Saxena. Progress on polynomial identity testing. *Bulletin of the EATCS*, 99:49–79, 2009.

[Sch80] J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, October 1980.

[Shp07] Amir Shpilka. Interpolation of depth-3 arithmetic circuits with two multiplication gates. *SIAM J. Comput.*, 38:2130–2161, 2007.

[Sin16a] Gaurav Sinha. *Blackbox Reconstruction of Depth Three Circuits with Top Fan-In Two.* PhD thesis, California Institute of Technology, Pasadena, CA, USA, 2016.

[Sin16b] Gaurav Sinha. Reconstruction of real depth-3 circuits with top fan-in 2. In *Proceedings of the 31st Conference on Computational Complexity*, CCC '16, pages 31:1–31:53, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik.

[SS11] Nitin Saxena and C. Seshadhri. Blackbox identity testing for bounded top fanin depth-3 circuits: The field doesn't matter. In *Proceedings of the Forty-third Annual ACM Symposium on Theory of Computing*, STOC '11, pages 431–440, New York, NY, USA, 2011. ACM.

[SS13] Nitin Saxena and C. Seshadhri. From sylvester-gallai configurations to rank bounds: Improved blackbox identity test for depth-3 circuits. *J. ACM*, 60(5), October 2013.

[SW99] Amir Shpilka and Avi Wigderson. Depth-3 arithmetic circuits over fields of characteristic zero. *Computational Complexity*, 10:1–27, 1999.

[SY10] Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends® in Theoretical Computer Science*, 5(3–4):207–388, 2010.

[Wig06] Avi Wigderson. P, np and mathematics - a computational complexity perspective. *Proceedings oh the International Congress of Mathematicians, Vol. 1, 2006-01-01, ISBN 978-3-03719-022-7, pags. 665-712*, 1, 01 2006.

[Zip79] Richard Zippel. Probabilistic algorithms for sparse polynomials. In *Proceedings of the International Symposiumon on Symbolic and Algebraic Computation*, EUROSAM '79, page 216–226, Berlin, Heidelberg, 1979. Springer-Verlag.

A Proof of Claims 1, 2 and 3

A.1 Proofs of Claims 1 and 2

In these claims we are given that $T_i = \alpha y_1^t$ for some $i \in [2], \alpha \in \mathbb{F}$ and linear form y_1.

1. To see the proof of Claim 1, consider any linear factor ℓ of $T_1 + T_2$. $\ell \nmid T_1, T_2$ since $gcd(T_1, T_2) = 1$. Let Φ be an isomorphism mapping $\ell \mapsto x_1$. Setting $x_1 = 0$, we get that $\Phi(T_1)_{|x_1=0} = -\Phi(T_2)_{|x_1=0} \neq 0$. Both sides are non-zero products of linear forms in $\mathbb{F}[x_2, \dots, x_n]$. Therefore, by unique factorization we can match factors (upto scalar multiplication). This implies that $dim(\{\text{linear form } \ell : \ell \mid T_1\})$ and $dim(\{\text{linear form } \ell : \ell \mid T_2\})$ cannot differ from each other by more than 1. But since $rank(f) = \Omega(\log^3 d)$, this cannot happen since one of the T_i's spans a one dimensional space. Therefore $T_1 + T_2$ has no linear factors and we are done.

2. To see proof of Claim 2, without loss of generality assume $y_1 \mid T_1$. Define isomorphism Φ mapping $y_1 \mapsto x_1$. Using Claim 1 we know that

$$0 \neq \Phi(T_2)_{|x_1=0} = (\Phi(T_1) + \Phi(T_2))_{|x_1=0} = \Phi(NonLin(f))_{|x_1=0}$$

So first condition of Definition 6 is satisfied. As argued in Claim 1, $rank(f) \geq \Omega(\log^3 d) \Rightarrow$ linear forms dividing T_2, span a $\Omega(\log^3 d)$ dimensional space. Since $\Phi(T_2)_{|x_1=0}$ is non-zero, it's factors also span $\Omega(\log^3 d)$ dimensional space and so there exist two linearly independent factors y_2, y_3 of T_2 such that $NonLin(f)$ vanishes on both $\mathbb{V}(y_1, y_2)$ and $\mathbb{V}(y_1, y_3)$. This implies that second condition of Definition 6 is also satisfied. Therefore, some scalar multiple of $y_1 \in \mathcal{L}(NonLin(f))$.

A.2 Proof of Claim 3

Recall definition of sets,

$$\mathcal{L}_{good} = \{\ell \in \mathcal{L}(NonLin(f)) : \ell \mid T_1 \times T_2\}, \qquad \mathcal{L}_{bad} = \mathcal{L}(NonLin(f)) \setminus \mathcal{L}_{good},$$

$$\mathcal{L}_{others} = \{\ell \mid T_1 \times T_2 : sp(\ell) \cap \mathcal{L}(NonLin(f)) = \phi\} \quad \text{and} \quad \mathcal{L}_{factors} = \{\ell : \ell \mid T_1 + T_2\}$$

For all sets, we keep linear forms upto scalar multiplication and therefore treat them as proper sets (Definition 3). Below we prove all parts of Claim 3.

1. $dim(sp(\mathcal{L}_{factors})) \leq \log d + 2$: By definition $\mathcal{L}_{factors}$ is the set of all factors of $T_1 + T_2$. Consider any linearly independent subset $\mathcal{Z} \subset \mathcal{L}_{factors}$ and let $\ell \in \mathcal{Z}$. Define isomorphism Φ mapping $\ell \mapsto x_1$. Setting $x_1 = 0$ in $\Phi(T_1) + \Phi(T_2)$ gives $\Phi(T_1)_{|x_1=0} = -\Phi(T_2)_{|x_1=0} \neq 0$. By unique factorization in ring $\mathbb{F}[x_2, \ldots, x_n]$, for every linear form $\ell_1 \mid T_1$ there exists $\ell_2 \mid T_2$ such that $\ell_2 \in sp\{\ell, \ell_1\}$. Since $\ell_2 \notin sp\{\ell\} \cup sp\{\ell_1\}$, this means that $sp\{\ell, \ell_1\}$ is not an ordinary line from ℓ into the proper set $\mathcal{L}$ containing linear factors of T_1, T_2. This set has size $\leq 2d$. Since ℓ was arbitrary in $\mathcal{Z}$, there are no ordinary lines from $\mathcal{Z}$ into $\mathcal{L}$. So using Lemma 29 we get that $|\mathcal{Z}| \leq \log|\mathcal{L}| + 1 = \log d + 2$, completing the proof.

2. $dim(sp(\mathcal{L}_{good})) \geq rank(f) - 2$ and $\mathcal{L}_{others} \leq 2$: Define $V_i = \{\text{linear form } \ell : \ell \mid T_i\}$. We break the proof into two cases. Note that linear forms dividing T_1, T_2 satisfy first condition of Definition 6. So whenever we are trying to show that they belong to $\mathcal{L}(NonLin(f))$, we only prove that they satisfy second condition of Definition 6.

    (a) First we discuss the case $dim(V_i) \geq \log d + 5$ for all $i \in [2]$. Let H be such that $T_1 + T_2 = H \times NonLin(f)$. Let $\ell_1 \mid T_1$ and Φ be isomorphism mapping $\ell_1 \mapsto x_1$, then, we see that $\Phi(T_2)_{|x_1=0} = \Phi(H)_{|x_1=0} \times \Phi(NonLin(f))_{|x_1=0} \neq 0$. Dimension of span of linear factors of $\Phi(T_2)_{|x_1=0}$ is at least $\log d + 4$ by assumption in this case. By previous part, $dim(sp(\mathcal{L}_{factors})) \leq \log d + 2 \Rightarrow \Phi(NonLin(f))_{|x_1=0}$ has two independent linear factors. Using these we can satisfy second condition of Definition 6 for $\ell_1 \Rightarrow$ some scalar multiple of $\ell_1 \in \mathcal{L}(NonLin(f))$. The same argument can be repeated for a linear factor $\ell_2 \mid T_2$. Thus all linear factors of $T_1 \times T_2$ are in $\mathcal{L}(NonLin(f))$ (upto scalar multiplication) $\Rightarrow$ $dim(\mathcal{L}_{good}) = rank(f)$. This also implies that $dim(\mathcal{L}_{others}) = 0$.

    (b) In the case when $dim(V_i) \leq \log d + 4$ for some $i \in [2]$, we know that $dim(V_{3-i}) = \Omega(\log^3 d)$ and therefore by an argument similar to the one given in proof of Claim 1, $NonLin(f) = T_1 + T_2$. Consider any basis $\{\ell_1, \ldots, \ell_r\}$ of $V_1 + V_2$. If $dim(V_i) \geq 3$ for all $i \in [2]$, then using a similar argument as before, we can show that all ℓ_i satisfy second condition in Definition 6 $\Rightarrow dim(\mathcal{L}_{good}) = rank(f) \Rightarrow dim(\mathcal{L}_{others}) = 0$. In case for some $i \in [2]$, $dim(V_i) = 2$ (recall we have assumed $dim(V_i) \geq 2$ in the statement of Claim 3), then all linear forms dividing T_{3-i} are not contained in V_i and hence satisfy second condition of Definition 6. Thus $dim(\mathcal{L}_{good}) \geq rank(f) - 2$ and $dim(\mathcal{L}_{others}) \leq 2$.

3. $dim(sp(\mathcal{L}_{bad})) \leq \log d + 2$: Assume $dim(\mathcal{L}_{bad}) \geq \log d + 3$. Consider the proper set $\mathcal{L}$ containing all linear factors of $T_1, T_2 \Rightarrow |\mathcal{L}| \leq 2d \Rightarrow |\mathcal{L}_{bad}| \geq \log|\mathcal{L}| + 2$. Let $\mathcal{T} \subset \mathcal{L}_{bad}$ be a linearly independent set of size $\log|\mathcal{L}| + 2$. Then by Proposition 1, there exists $t \in \mathcal{T}$ such that ordinary lines from t into $\mathcal{L}$ span a space of dimension $\geq \frac{dim(sp(\mathcal{L}))}{\log|\mathcal{L}|+2} \geq \frac{rank(f)}{\log d+3} = \Omega(\log^2 d)$. Since $t \in \mathcal{L}_{bad}$, restricting $T_1 + T_2$ to $\mathbb{V}(t)$ (see Definition 5) gives some non-zero product of linear factors, say H. Let Φ be an isomorphism mapping $t \mapsto x_1$. Then,

$$\Phi(T_1)_{|x_1=0} + \Phi(T_2)|_{x_1=0} - H = 0$$

This gives an identically zero $\Sigma\Pi\Sigma(3, n, d, \mathbb{F})$ circuit. Since $t \in \mathcal{L}_{bad}$, it does not divide $T_1, T_2 \Rightarrow$ the above circuit is minimal (Definition 10). After cancelling common linear forms from the three gates $\Phi(T_1)_{|x_1=0}, \Phi(T_2)_{|x_2=0}, H$, we have a simple (Definition 9) and minimal, identically zero $\Sigma\Pi\Sigma(3, n, d, \mathbb{F})$ circuit. The $\Omega(\log^2 d)$ ordinary lines from t into $\mathcal{L}$ imply that after cancelling the common linear forms, the simple minimal circuit has rank $\Omega(\log^2 d)$ which is a contradiction to Lemma 5. Thus we conclude that $dim(sp(\mathcal{L}_{bad})) \leq \log d + 2$.

B Proof of Lemma 23

Let $T_i = \prod_{j=1}^{m} \ell_{i,j}$ where $\ell_{i,j}$ are linear forms. We know that,

$$\prod_{j=1}^{m} \Phi(\ell_{1,j})_{|x_1=0,x_2=0} = -\prod_{j=1}^{m} \Phi(\ell_{2,j})_{|x_1=0,x_2=0} \neq 0.$$

Note that $\Phi(\ell_{i,j})_{|x_1=0,x_2=0}$ can be thought of as linear forms over $\mathbb{F}$ in $n-2$ variables, and by using unique factorization of polynomials over $\mathbb{F}$, without loss of generality we can assume $\Phi(\ell_{1,j})_{|x_1=0,x_2=0} = \beta_j \Phi(\ell_{2,j})_{|x_1=0,x_2=0}$ for some $0 \neq \beta_j \in \mathbb{F}$. This implies[17] $U_j = sp\{\ell_{1,j}, \ell_{2,j}\}$[18] intersects $U = sp\{\ell_1, \ell_2\}$ non-trivially. Since $\Phi(\ell_{i,j})_{|x_1=0,x_2=0} \neq 0$, we know that $U \neq U_j \Rightarrow U \cap U_j$ is 1 dimensional[19]. We split the proof into two cases:

- **There exist two distinct spaces, say U_i, U_j such that $U \cap U_i = U \cap U_j$:** This implies $U \cap U_i \subset U_i \cap U_j$. The space $U_i \cap U_j$ is 1 dimensional since U_i, U_j are distinct, say $U_i \cap U_j = sp\{\ell\}$. Both sides of the containment $U \cap U_i \subset U_i \cap U_j$ are 1 dimensional implying $U_i \cap U_j = U \cap U_i \subset U = sp\{\ell_1, \ell_2\}$. This further implies that $\ell \in U \Rightarrow W \subset \mathbb{V}(\ell) = V$. There are $\leq d^4$ choices for such U_i, U_j and therefore d^4 possibilities for such V.

- **For all distinct U_i, U_j, $U \cap U_i \neq U \cap U_j$:** Vector space $U \cap U_i + U \cap U_j$ is 2 dimensional, since it is a sum of disjoint 1 dimensional spaces. U is also 2 dimensional $\Rightarrow U = U \cap U_i + U \cap U_j \subset U_i + U_j$. Using statement of Proposition 1, we know that

$$5 \leq rank(f) = dim(sp\{\ell_{i,j}\}) = dim(\sum_{j=1}^{m} U_j) \leq \sum_{j=1}^{m} dim(U_j).$$

[17] since Φ is an isomorphism.

[18] $\ell_{1,j}, \ell_{2,j}$ are linearly independent since $gcd(T_1, T_2) = 1$

[19] since both U, U_j are 2 dimensional

$dim(U_i + U_j) \le 4$, thus there exists U_k such that $U_k \notin U_i + U_j$. Note that this would imply that $U_k \cap (U_i + U_j)$ has dimension ≤ 1. Since $U \subset U_i + U_j$, we get that $U_k \cap U \subset U_k \cap (U_i + U_j)$. Both sides are 1 dimensional. Writing $U_k \cap (U_i + U_j) = sp\{\ell\} \Rightarrow \ell \in U \Rightarrow W \subset \mathbb{V}(\ell) = V$. There are $\le d^6$ choices for U_i, U_j, U_k and so $\le d^6$ possibilities for such V.

$\mathcal{A}$ is collection of all V's obtained above. $|\mathcal{A}| \le d^4 + d^6$ and $\mathcal{A}$ satisfies the required conditions.

C Proofs of Lemmas in Algorithm 7

In this appendix, we provide proofs to lemmas that were stated and used in Algorithm 7 but proofs were not provided.

C.1 Proof of Lemma 24

We prove each part one by one below. Let $\hat{\ell}_i = \sum_{j=1}^{n} \alpha_{i,j} x_j, i \in [n]$ be the n linear forms that were constructed using the uniformly randomly independently samples $\alpha_{i,j}, i \in [n], j \in [n]$. Recall that Φ maps $x_i \mapsto \hat{\ell}_i$. Let Γ be a homomorphism from $\mathbb{F}[x_1, \ldots, x_n] \to \mathbb{F}[x_1, x_2, x_3, x_4, x_i]$ that sets $x_5 = 0, \ldots, x_{i-1} = 0, x_{i+1} = 0, \ldots, x_n = 0$.

1. Showing that these linear forms are independent is equivalent to showing that with probability $1 - o(1)$, the matrix $(\alpha_{i,j})_{(i,j) \in [n] \times [n]}$ is invertible. This is equivalent to saying that the determinant polynomial of this matrix is non-zero. Applying Lemma 2 on the determinant polynomial, we get this result.

2. Consider any isomorphism Ψ mapping $\ell_1 \mapsto x_1, \ell_2 \mapsto x_2$, then $\Psi \circ \Phi^{-1}$ is an isomorphism mapping $\Phi(\ell_1) \mapsto x_1, \Phi(\ell_2) \mapsto x_2$. Further, $\Psi(NonLin(f)) = \Psi \circ \Phi^{-1}(\Phi(NonLin(f)))$. Restricting both sides to $x_1 = 0, x_2 = 0$ gives,

$$\Psi(NonLin(f))_{|x_1=0, x_2=0} = \Psi \circ \Phi^{-1}(\Phi(NonLin(f)))_{|x_1=0, x_2=0},$$

   implying that $NonLin(f)$ vanishes on $\mathbb{V}(\ell_1, \ell_2)$ if an only if $\Phi(NonLin(f))$ vanishes on subspace $\mathbb{V}(\Phi(\ell_1), \Phi(\ell_2))$. Since Φ is an isomorphism, irreducible factors of f remain irreducible on applying Φ, thereby implying that $\Phi(NonLin(f)) = NonLin(\Phi(f)) = NonLin(g)$. Hence claim is proved.

3. Recall $f = G \times (T_1 + T_2)$, with G, T_1, T_2 being product of linear forms and $gcd(T_1, T_2) = 1$. Since Φ is an isomorphism, we get that $g = \Phi(G) \times (\Phi(T_1) + \Phi(T_2))$. Since Φ is an isomorphism, $gcd(\Phi(T_1), \Phi(T_2)) = 1$. Therefore we get,

$$g_i = \Gamma(g) = \Gamma(\Phi(G))(\Gamma(\Phi(T_1)) + \Gamma(\Phi(T_2))).$$

   Next, consider linear forms $\ell = \sum_{j=1}^{n} a_j x_j$ and $\ell' = \sum_{j=1}^{n} a'_j x_j$ such that $\ell \mid T_1$ and $\ell' \mid T_2$. Applying Φ to these linear forms we get, $\Phi(\ell) = \sum_{k=1}^{n} \sum_{j=1}^{n} a_j \alpha_{j,k} x_k$. Therefore coefficients of x_1, x_2 in $\Gamma(\Phi(\ell))$ are $\sum_{j=1}^{n} a_j \alpha_{j,1}$, $\sum_{j=1}^{n} a_j \alpha_{j,2}$ respectively and those in $\Gamma(\Phi(\ell'))$ are $\sum_{j=1}^{n} a'_j \alpha_{j,1}$, $\sum_{j=1}^{n} a'_j \alpha_{j,2}$.

We argue that vectors $(\sum_{j=1}^{n} a_j\alpha_{j,1}, \sum_{j=1}^{n} a_j\alpha_{j,2})$ and $(\sum_{j=1}^{n} a'_j\alpha_{j,1}, \sum_{j=1}^{n} a'_j\alpha_{j,2})$ are not scalar multiples with probability $1-o(1)$. This is equivalent to showing that the following determinant is non-zero.

$$\begin{vmatrix} \sum_{j=1}^{n} a_j\alpha_{j,1} & \sum_{j=1}^{n} a_j\alpha_{j,2} \\ \sum_{j=1}^{n} a'_j\alpha_{j,1} & \sum_{j=1}^{n} a'_j\alpha_{j,2} \end{vmatrix}$$

If ℓ, ℓ' are not scalar multiples, this determinant is not an identically zero polynomial in the $\alpha_{j,k}, j \in [n], k \in [2]$ and therefore probability (over the random choices of $\alpha_{j,k}$) that the determinant is non-zero $= 1-o(1)$. Therefore with probability $1-o(1)$, $\Gamma(\Phi(\ell))$ and $\Gamma(\Phi(\ell'))$ are not scalar multiples. Since ℓ, ℓ' are arbitrary linear factors of T_1, T_2 respectively, by union bound with probability $1-o(1)$, $gcd(\Gamma(\Phi(T_1)), \Gamma(\Phi(T_2))) = 1$ implying that all g_i exhibit $\Sigma\Pi\Sigma(2,5,d,\mathbb{F})$ circuit. Since $rank(f) = \Omega(\log^2 d)$, we know that $dim(sp\{\text{linear form } \ell : \ell \mid T_1 \times T_2\}) \geq 5$, therefore, a similar argument (again using Lemma 2), can be used to say that $\{\Gamma(\Phi(\ell)) : \text{linear form } \ell \mid T_1 \times T_2\}$ spans a 5 dimensional space. This set is the same as $\{\text{linear form } \ell : \ell \mid \Gamma(\Phi(T_1)) \times \Gamma(\Phi(T_2))\})\}$, proving that $rank(g_i) = 5$ for all $i \in [5, n]$.

4. By effective Hilbert's irreducibility theorem (Lemma 7), we know that with probability $1-o(1)$ over the $\alpha_{i,j}, i \in [n], j \in [n]$, the irreudicible factors of $\Phi(f)(x_1, \ldots, x_n) = f(\Phi(x_1), \ldots, \Phi(x_n))$ remain irreducible on setting $x_5 = 0, \ldots, x_{i-1} = 0, x_{i+1} = 0, \ldots, x_n = 0$ i.e. on applying Γ. Example if h is an irreducible factor of f, then $\Gamma(\Phi(h))$ is an irreducible factor of $\Gamma(\Phi(f))$. The same will apply to product (with multiplicity) of all non-linear irreducible factors implying that,
$$NonLin(\Gamma(\Phi(f))) = \Gamma(NonLin(\Phi(f))).$$
The left hand side is same as polynomial $NonLin(g_i)$ and right hand side is same as polynomial $NonLin(g)_{|x_5=0,\ldots,x_{i-1}=0,x_{i+1}=0,\ldots,x_n=0}$. Hence Proved.

5. Let $\mathbb{V}(\hat{\ell}_1, \hat{\ell}_2)$ belong to $\mathcal{S}(NonLin(f))$. Assume $\hat{\ell}_1 = \sum_{j=1}^{n} a_j x_j$ and $\hat{\ell}_2 = \sum_{j=1}^{n} b_j x_j$, Then we get that $\Phi(\hat{\ell}_1) = \sum_{k=1}^{n}(\sum_{j=1}^{n} a_j\alpha_{j,k})x_k$ and $\Phi(\hat{\ell}_2) = \sum_{k=1}^{n}(\sum_{j=1}^{n} b_j\alpha_{j,k})x_k$. We define $c_k = \sum_{j=1}^{n} a_j\alpha_{j,k}$ an $d_k = \sum_{j=1}^{n} b_j\alpha_{j,k}$ for $k \in [n]$. Therefore $\Phi(\hat{\ell}_1) = \sum_{k=1}^{n} c_k x_k$ and $\Phi(\hat{\ell}_2) = \sum_{k=1}^{n} d_k x_k$. Now we define new linear forms as follows:
$$\begin{bmatrix} \ell_3 \\ \ell_4 \end{bmatrix} = \begin{bmatrix} d_2 & -c_2 \\ -d1 & c_1 \end{bmatrix} \begin{bmatrix} \Phi(\hat{\ell}_1) \\ \Phi(\hat{\ell}_2) \end{bmatrix} \tag{1}$$
Determinant of the matrix is $d_2c_1 - c_2d_1$. This defines a polynomial in the $\alpha_{j,k}, j \in [n], k \in [2]$. Like in the previous part, unless $\hat{\ell}_1, \hat{\ell}_2$ are linearly dependent this polynomial is not identically 0. Therefore with probability $1-o(1)$ over the uniformly randomly chosen linear forms in Step 1, the determinant is non-zero implying that $d_2c_1 - c_2d_1 \neq 0$. This also means that ℓ_3, ℓ_4 are linearly independent and $\mathbb{V}(\ell_3, \ell_4) = \mathbb{V}(\Phi(\hat{\ell}_1), \Phi(\hat{\ell}_2))$. Analyzing ℓ_3, ℓ_4 we see that,
$$\ell_3 = (d_2c_1 - c_2d_1)x_1 + \sum_{k=3}^{n}(d_2c_k - c_2d_k)x_k, and$$

$$\ell_4 = (d_2c_1 - c_2d_1)x_2 + \sum_{k=3}^{n}(d_kc_1 - c_kd_1)x_k.$$

Define $\ell_1' = -\sum_{k=3}^{n}\frac{d_2c_k - c_2d_k}{d_2c_1 - c_2d_1}x_k$, and $\ell_2' = -\sum_{k=3}^{n}\frac{d_kc_1 - c_kd_1}{d_2c_1 - c_2d_1}x_k$, further implying that $\mathbb{V}(\ell_1,\ell_2) = \mathbb{V}(x_1 - \ell_1', x_2 - \ell_2')$ with $\ell_1',\ell_2' \in \mathbb{F}[x_3,\dots,x_n]$. Now since $\mathcal{S}(NonLin(f))$ has size $d^{O(1)}$, by union bound, with probability $1 - o(1)$, we can prove all of this for every $\mathbb{V}(\hat{\ell}_1,\hat{\ell}_2) \in \mathcal{S}(NonLin(f))$.

Now, given any $\mathbb{V}(\ell_1,\ell_2) \in \mathcal{S}(NonLin(g))$, by Part 2 of this Lemma, we know that $\mathbb{V}(\ell_1,\ell_2) \in \mathcal{S}(NonLin(g))$ if and only if $\mathbb{V}(\Phi^{-1}(\ell_1),\Phi^{-1}(\ell_2)) \in \mathcal{S}(NonLin(g))$. So we can use our argument for $\hat{\ell}_1 = \Phi^{-1}(\ell_1)$, and $\hat{\ell}_2 = \Phi^{-1}(\ell_2)$, thereby completing the proof.

6. Let $\mathbb{V}(\ell_1,\ell_2) \in \mathcal{S}(NonLin(g))$ and $\ell_j^i = \ell_j{}_{|x_5=0,\dots,x_{i-1}=0,x_{i+1}=0,\dots,x_n=0}$. By previous part we know that there exists $\ell_1',\ell_2' \in \mathbb{F}[x_3,\dots,x_n]$ such that $\mathbb{V}(\ell_1,\ell_2) = \mathbb{V}(x_1 - \ell_1', x_2 - \ell_2')$. Let Θ be an isomorphism mapping $x_1 - \ell_1' \mapsto x_1, x_2 - \ell_2' \mapsto x_2$ and for $j \in [3,n]$, $x_j \mapsto x_j$. Similarly let Θ' be isomorphism on $\mathbb{F}[x_1,x_2,x_3,x_4,x_i]$ mapping $x_1 - \ell_1'' \mapsto x_1, x_2 - \ell_2'' \mapsto x_2$ and for $j \in \{3,4,i\}$, $x_j \mapsto x_j$. Finally let Γ be the homomorphism from $\mathbb{F}[x_1,\dots,x_n]$ to $\mathbb{F}[x_1,x_2,x_3,x_4,x_i]$ mapping $x_j \mapsto 0$ for all $j \in [5,i-1] \cup [i+1,n]$. The following diagram commutes.

$$\begin{tikzcd}
\mathbb{F}[x_1,\dots,x_n] \arrow[r, "\Theta"] \arrow[d, "\Gamma"'] & \mathbb{F}[x_1,\dots,x_n] \arrow[d, "\Gamma"] \\
\mathbb{F}[x_1,x_2,x_3,x_4,x_i] \arrow[r, "\Theta'"] & \mathbb{F}[x_1,x_2,x_3,x_4,x_i]
\end{tikzcd}$$

We know that $NonLin(g)$ vanishes on $\mathbb{V}(x_1 - \ell_1', x_2 - \ell_2')$, therefore $\Theta(NonLin(g))_{|x_1=0,x_2=0} = 0$, implying that $\Gamma(\Theta(NonLin(g))_{|x_1=0,x_2=0}) = 0$. We know that Γ fixes x_1, x_2 therefore we can set $x_1 = 0, x_2 = 0$ after applying Γ, thereby giving $\Gamma(\Theta(NonLin(g)))_{|x_1=0,x_2=0} = 0$. Using the above commutative diagram we get, $\Theta'(\Gamma(NonLin(g)))_{|x_1=0,x_2=0} = 0$. Now, Part 4 of this lemma gives $NonLin(g_i) = \Gamma(NonLin(g))$. Using this we get, $\Theta'(NonLin(g_i))_{|x_1=0,x_2=0} = 0$. Therefore $NonLin(g_i)$ vanishes on the co-dimension 2 subspace $\mathbb{V}(x_1 - {\ell_1}'', x_2 - {\ell_2}'')$ of $\mathbb{F}^5$, thereby completing the proof.

C.2 Proof of Lemma 25

Fix $i \in [6,n]$. Consider a pair of distinct tuples $(x_1 - \ell_1, x_2 - \ell_2)$, $(x_1 - \ell_1', x_2 - \ell_2')$ in $\mathcal{S}_i$. By construction, $\ell_1,\ell_2,\ell_1',\ell_2' \in \mathbb{F}[x_3,x_4,x_i]$. So we assume that, $\ell_1 = a_3x_3 + a_4x_4 + a_ix_i$, $\ell_2 = b_3x_3 + b_4x_4 + b_ix_i$, $\ell_1' = a_3'x_3 + a_4'x_4 + a_i'x_i$ and $\ell_2' = b_3'x_3 + b_4'x_4 + b_i'x_i$. Therefore,

$$\begin{aligned}
\Delta(\ell_1) &= (a_3 + \alpha_{i,3}a_i)x_3 + (a_4 + \alpha_{i,4}a_i)x_4 + a_ix_i,\\
\Delta(\ell_2) &= (b_3 + \alpha_{i,3}b_i)x_3 + (b_4 + \alpha_{i,4}b_i)x_i + b_ix_i,\\
\Delta(\ell_1') &= (a_3' + \alpha_{i,3}a_i')x_3 + (a_4' + \alpha_{i,4}a_i')x_4 + a_i'x_i,\\
\Delta(\ell_2') &= (b_3' + \alpha_{i,3}b_i')x_3 + (b_4' + \alpha_{i,4}b_i')x_4 + b_i'x_i
\end{aligned}$$

If $(\Delta(\ell_1)_{|x_i=0}, \Delta(\ell_2)_{|x_i=0}) = (\Delta(\ell_1')_{|x_i=0}, \Delta(\ell_2')_{|x_i=0})$, then we get a system of linear equations in $\alpha_{i,3}, \alpha_{i,4}$ which can be simplified to get

$$\begin{bmatrix} \alpha_{i,3}(a_i - a'_i) \\ \alpha_{i,4}(a_i - a'_i) \\ \alpha_{i,3}(b_i - b'_i) \\ \alpha_{i,4}(b_i - b'_i) \end{bmatrix} = \begin{bmatrix} a'_3 - a_3 \\ a'_4 - a_4 \\ b'_3 - b_3 \\ b'_4 - b_4 \end{bmatrix}$$

Since tuples (ℓ_1, ℓ_2) and (ℓ'_1, ℓ'_2) are distinct, at least one of $(a'_3 - a_3), (a_i - a'_i), (a'_4 - a_4), (b_i - b'_i), (b'_3 - b_3), (b'_4 - b_4)$ is non-zero implying that at least one linear equation is not identically zero. By Lemma 2, we then know that with probability $1 - o(1)$ over the uniformly random choices of $\alpha_{i,3}, \alpha_{i,4}$ the equation cannot be zero. Therefore with probability $1 - o(1)$, $(\Delta(\ell_1)_{|x_i=0}, \Delta(\ell_2)_{|x_i=0}) \neq (\Delta(\ell'_1)_{|x_i=0}, \Delta(\ell'_2)_{|x_i=0})$. Using Part 3 of Lemma 24, we know that $rank(NonLin(g_i)) = 5$ implying that $|\mathcal{S}_i| = d^{O(1)}$. So we can take a union bound over all pairs of tuples in $\mathcal{S}_i$. Finally, we take a union bound over all i and guarantee that with probability $1 - o(1)$, the statement in this lemma holds.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il