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

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

Gaurav Sinha*

August 17, 2020

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 top(output) gate is an addition gate with in-degree 2. 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(addition) 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 algorithms, which take as input a black-box computing polynomial f, with coefficients in a finite field $\mathbb{F}$, exhibiting such a circuit. Here are the results.

- [Low rank] : When $5 \leq r = rank(f) = O(\log^3 d)$, it runs in time $(nd^{\log^3 d} \log |\mathbb{F}|)^{O(1)}$ and outputs a depth three circuit computing f (with high probability), 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 2.

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 1). 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 $n, d, \log|\mathbb{F}|$[3]. An important point to note is that when rank (Definition 3) 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 $\log d$ and $\log|\mathbb{F}|$ was given. However, unlike [Shp07], their algorithm is improper and output might have much larger top fan-in. [Sin16b] also considered the top fan-in 2 case, but over characteristic 0 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 depends polynomially 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 2). Also, for any polynomial f, $rank(f)$ (Definition 4) 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 outputs, with probability $1 - o(1)$, 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 $rank(f) \leq 2$, the polynomial factors into a product of linear forms and can be reconstructed efficiently using Lemma 4. So only $rank(f) = 3, 4$ 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 Theorem 3.
- In order to prove Theorem 2, we develop an interesting result related to Sylvester Gallai (SG) type configurations (Definition 9) and present it in Lemma 1. 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 ordinary lines through a point.

[4] this is obtained by removing all linear factors of f. See Definition 5

Theorem 3. *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 $(rank(f) \geq 5)$. Let $\mathcal{S}(f)$ be the set of co-dimension 2 subspaces on which $NonLin(f)$ (Definition 5) vanishes. The following are true.*

1. $|\mathcal{S}(f)| \leq 3d^7$.
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*

$$Pr[(\ell_1, \ell_2) \in \mathcal{S} \Rightarrow \{\ell_1 = 0, \ell_2 = 0\} \in \mathcal{S}(f)] \geq 1 - o(1),$$

$$Pr[\{\ell_1 = 0, \ell_2 = 0\} \in \mathcal{S}(f) \Rightarrow \exists (\ell_1', \ell_2') \in \mathcal{S} \text{ s.t. } sp\{\ell_1, \ell_2\} = sp\{\ell_1', \ell_2'\}] \geq 1 - o(1).$$

Lemma 1. *Let $\mathcal{S} \subset \mathbb{F}^n$ be a proper set (Definition 7) 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 the set of ordinary lines $\mathcal{O}(t, \mathcal{S})$ (Definition 8) 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 3 are given in Algorithms 3, 1 and 7 respectively. In this section we discuss key technical ideas required for proving these theorems. Missing proofs are supplied in the subsequent sections. Using Definition 2, write $f = G \times (T_1 + T_2)$ where G, T_1, T_2 are product of linear forms and $gcd(T_1, T_2) = 1$. Define $\mathcal{T}_i = \{\text{linear form } \ell : \ell \mid T_i\}$ and $V_i = sp(\mathcal{T}_i)$ for $i \in [2]$. We use the definitions of $NonLin(f)$ (Definition 5), set of ordinary lines (Definition 8) and set of candidate linear forms $\mathcal{L}(f)$ (Definition 6).

1.2.1 Theorem 1: Key ideas for Algorithm 3

The algorithm mentioned in Theorem 1 is presented in Algorithm 3 and it's correctness/complexity is discussed in Section 3. As we discussed earlier, previous work in [Shp07] provides an algorithm[5], that runs in time quasi polynomial in $n, d, \log |\mathbb{F}|$. In Algorithm 3([Shp07]), they perform a brute force search on the space of linear forms to construct a high dimensional linearly independent set. Step 2 in our algorithm makes this search efficient by performing a non-deterministic search over a $d^{O(1)}$ sized set $\mathcal{S}(f)$[6]. It's correctness is justified by the analysis provided in Section 3. Next, we outline the key technical ideas on which our algorithm is built.

- Polynomial $NonLin(f)$ divides $T_1 + T_2$, and therefore it can be written as a polynomial over any basis of the vector space V spanned by linear factors of T_1 and T_2. We try to construct such a basis in a non-deterministic fashion.

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

[6] containing tuples representing co-dimension 2 subspaces on which $NonLin(f)$ vanishes

- Consider any linear form $\ell_1 \mid T_1$ ($\ell_2 \mid T_2$ respectively). Since $NonLin(f)$ is not a constant polynomial(due to $rank(f) \geq 5$), it's easy to see that there exists a linear form $\ell_2 \mid T_2$ ($\ell_1 \mid T_1$ respectively), such that $NonLin(f)$ vanishes on the co-dimension 2 subspace $\{\ell_1 = 0, \ell_2 = 0\}$ of $\mathbb{F}^n$. Any representation of this space, say $\{\ell_1 = 0, \ell_2 = 0\} = \{\ell'_1 = 0, \ell'_2 = 0\}$ then satisfies $sp\{\ell_1, \ell_2\} = sp\{\ell'_1, \ell'_2\}$. Therefore, $rank(f)$ many such co-dimension 2 subspaces can easily give us a basis for V. Details of this argument are provided in Lemma 7.

- By Theorem 3, we know that there are at most $d^{O(1)}$ many co-dimension 2 subspaces on which $NonLin(f)$ vanishes, and that these can be computed in $(nd \log |\mathbb{F}|)^{O(1)}$ time. So we have a $d^{O(1)}$ sized set $\mathcal{S}(f)$ to search for the above kind of subspaces which will help us create a basis for V.

- We do not know $rank(f)$ in advance but know that $rank(f) = O(\log^3 d)$. So we non-deterministically search for $rank(f)$, and the $rank(f)$ sized subset of $\mathcal{S}(f)$ that we described above. For every such subset, we construct a basis of linear forms and try to interpolate $NonLin(f)$ as a homogeneous polynomial[7] in the constructed basis.

- If our guessed value of $rank(f)$, and the $rank(f)$ sized subset are correct, we will be able to interpolate and obtain a $\Sigma\Pi\Sigma(t^{rank(f)}, n, t, \mathbb{F})$ circuit computing $NonLin(f)$. Also, if interpolation is successful for an incorrect guess, we will still have a $\Sigma\Pi\Sigma(t^r, n, t, \mathbb{F})$ circuit for some $r \leq \log^3 d$. Clearly with high probability $r \leq rank(f)$.

- Finally, using Algorithm 2, linear factors of f can be obtained and multiplied to the circuit above. With high probability we would have reconstructed a $\Sigma\Pi\Sigma(t^{rank(f)}, n, d, \mathbb{F})$ circuit for f in $(nd^{\log^3 d} \log |\mathbb{F}|)^{O(1)}$ time.

1.2.2 Theorem 2: Analysis of Algorithm 1

Our algorithm has four stages: *Try corner case → Separate linear/non-linear part → Compute linearly independent set dividing T_i → Reconstruction using independent set.* As mentioned earlier, previous work in [Shp07], provides an algorithm for this case that runs in time quasi polynomial in $n, d, \log |\mathbb{F}|$. In Algorithm 4([Shp07]), they perform a brute force search on linear forms to construct a high dimensional linearly independent set. Step 3 in our algorithm makes this search efficient by performing a non-deterministic search over set $\mathcal{L}(f)$, and therefore results in an efficient algorithm. It's correctness has been justified in the discussion following our algorithm, and can be found in the analysis of stages - "*Compute linearly independent set dividing T_i*" and "*Reconstruction using independent set*". Missing details have been provided in Section 4.2. We first give the algorithm and then discuss correctness and complexity of each stage.

[7] degree of $NonLin(f)$ is easily calculated using Algorithm 2

Algorithm 1 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 6 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. ▷ *Try corner case*

2. Using Algorithm 2 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 $t = d - s$. ▷ *Separate linear/non-linear parts*

3. Using Algorithm 4 with inputs as black-box access to f and integers n, d, construct the set $\mathcal{L}(f)$. Find $\ell \in \mathcal{L}(f)$ such that the set of ordinary lines, $\mathcal{O}(\ell, \mathcal{L}(f))$ (Definition 8) spans a space of dimension $\Omega(\log^2 d)$. If no such ℓ exists, **Return** #. Otherwise, construct an $\Omega(\log^2 d)$ sized linearly independent subset $\mathcal{X} \subset \mathcal{L}(f)$, such that $sp\{\ell, x\}$ is an ordinary line from ℓ into $\mathcal{L}(f)$ (i.e $sp\{\ell, x\} \in \mathcal{O}(\ell, \mathcal{L}(f))$). 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 $\ell' \in \mathcal{B}$, and using Lemma 3, check if $NonLin(f))_{|_{\{\ell=0, \ell'=0\}}} \equiv 0$. If yes, add ℓ' to $\mathcal{U}$ else add it to $\mathcal{V}$. Without loss of generality assume $\mathcal{U}$ is larger and select $r = 60 \log d + 61$ linear forms $u_1, \ldots, u_r \in \mathcal{U}$.
   ▷ *Compute linearly independent set dividing T_i*

   (b) Run Algorithm 5 with inputs as black-box access to f, integers n, d and linear forms $u_1, \ldots, u_r$. If it returns a $\Sigma\Pi\Sigma(2, n, d, \mathbb{F})$ circuit C, **Return** C. Else, go to the next $\mathcal{B}, \ell$ in the search.
   ▷ *Reconstruction using independent set*

4. **Return** #

We outline the correctness and complexity of the algorithm below.

1. *Try corner case(Step 1):* In this stage, we try to reconstruct the circuit using Algorithm 6, assuming that for some $i \in [2]$, $dim(V_i) = 1$. This is equivalent to assuming $T_i = \alpha \ell^t$ for some $\alpha \in \mathbb{F}$ and linear form ℓ. Correctness/complexity are implied by Algorithm 6, which runs in time $(nd \log |\mathbb{F}|)^{O(1)}$. We proceed to next stage only if reconstruction was not possible. Thus, for subsequent stages we can assume $dim(V_i) \geq 2$.

2. *Separate linear/non-linear part(Step 2):* Implied by Algorithm 2.

3. *Compute linearly independent set dividing T_i(Steps 3, 3(a)):* In this stage we try to compute a linearly independent set of size $r = 60 \log d + 61$, such that all linear forms in this set divide T_1 (or all divide T_2). First, using Algorithm 4 with inputs as black-box access to f and integers n, d, we compute $\mathcal{L}(f)$ in $(nd \log |\mathbb{F}|)^{O(1)}$ time. We search for the linearly independent set inside $\mathcal{L}(f)$ as follows.

   (a) By Part 1 of Lemma 11, there exists $\ell \in \mathcal{L}(f)$ dividing T_i (for some $i \in [2]$) such that given ℓ, we can compute a linearly independent set $\mathcal{X} \subset \mathcal{L}(f)$ of size $\Omega(\log^2 d)$, such that for all $\ell' \in \mathcal{X}$, $sp\{\ell, \ell'\}$ is an ordinary line from ℓ into $\mathcal{L}(f)$.

(b) By Part 2 of Lemma 11, any partition of $\mathcal{X}$ into $\Omega(\log d)$ equal parts, has a part $\mathcal{B}$, such that for all $\ell' \in \mathcal{B}$, $\ell' \mid T_1 \times T_2$, $sp\{\ell, \ell'\}$ is an ordinary line from ℓ into the set of linear factors of $T_1 \times T_2$ and it is also an ordinary line into the set of linear factors of $(T_1 + T_2)$.

(c) By Part 3 of Lemma 11, for any $\ell' \in \mathcal{B}$, $NonLin(f)|_{\{\ell=0,\ell'=0\}} = 0 \Leftrightarrow \ell'$ divides T_{3-i}. This splits $\mathcal{B}$ into disjoint sets $\mathcal{U}, \mathcal{V}$ as:

$$\mathcal{U} = \{\ell' \in \mathcal{B} : NonLin(f)|_{\{\ell=0,\ell'=0\}} = 0\}, \text{ and } \mathcal{V} = \mathcal{B} \setminus \mathcal{U}.$$

Therefore all linear forms $\in \mathcal{U}$ divide T_{3-i}, and those $\in \mathcal{V}$ divide T_i. Clearly, larger of $\mathcal{U}, \mathcal{V}$ has size $\Omega(\log d)$[8].

We non deterministically search for this $\ell \in \mathcal{L}(f)$, use ℓ to compute $\mathcal{X}$, and partition $\mathcal{X}$ into $\Omega(\log d)$ many equal parts. Then we non-deterministically search for the part $\mathcal{B}$. After this, for all $\ell' \in \mathcal{B}$, we restrict $NonLin(f)$ to $\{\ell = 0, \ell' = 0\}$ and check (using Lemma 3) if $NonLin(f)|_{\{\ell=0,\ell'=0\}} \equiv 0$. This creates the two sets $\mathcal{U}, \mathcal{V}$ described earlier. We select larger of the two. If our guesses of ℓ and $\mathcal{B}$ are correct, using arguments above, the larger of $\mathcal{U}, \mathcal{V}$ has size $\Omega(\log d)$, and all linear forms in it divide T_1 or T_2. These linear forms are then used in Algorithm 5 as described in next stage. We note that if an incorrect $\ell, \mathcal{B}$ was guessed, leading to an incorrect reconstruction in Algorithm 5, it gets rejected inside Algorithm 5, during the validation stage. Clearly all these steps work in $(nd \log |\mathbb{F}|)^{O(1)}$ time.

4. *Reconstruction using independent set(Step 3(b)):* If the guesses of ℓ and $\mathcal{B}$ in the previous stage are correct, Algorithm 5 with inputs as black-box access to f, integers n, d, and the $\Omega(\log d)$ many linearly independent linear forms computed above, reconstructs the circuit. As mentioned before, Algorithm 5 validates the constructed circuit and rejects if incorrect. Therefore if we output a circuit, it is always correct. Correctness and complexity are implied by Algorithm 5.

1.2.3 Theorem 3: Key ideas for proof of Part 1

We outline the key technical ideas that form the proof. Details are provided in Section 5.1.

1. For any co-dimension 2 space on which $NonLin(f)$ vanishes, we know that $T_1 + T_2$ vanishes as well (since $NonLin(f)$ divides $T_1 + T_2$). Thus $T_1|_W = -T_2|_W$. We split into two cases from here. Either both $T_1|_W, T_2|_W$ are zero or both are non-zero. When both are zero, we immediately get that some linear forms $\ell_1 \mid T_1$ and $\ell_2 \mid T_2$ vanish on W. Since $gcd(T_1, T_2) = 1$, we get that $W = \{\ell_1 = 0, \ell_2 = 0\}$, and therefore there are $\leq d^2$ such W's.

2. In Lemma 13, we show that there is a fixed set $\mathcal{A}$ of $d^{O(1)}$ many co-dimension 1 subspaces, such that for all W satisfying $NonLin(f)|_W = 0$ and $T_1|_W = -T_2|_W \neq 0$, there is some $V \in \mathcal{A}$ such that $W \subset V$. Using V and the fact that $NonLin(f)|_W = 0$, we can easily identify $\leq d$ possibilities for W. Here each possibility corresponds to some linear factor of $NonLin(f)|_V$ as shown in Section 5.1. Thus there are $d^{O(1)}$ such W's. So only Lemma 13 is left to be explained.

[8] we chose constants so that the larger has size $60 \log d + 61$

3. Here is the main idea for proving Lemma 13. Let $W = \{\ell_1 = 0, \ell_2 = 0\}$, be such that $T_{1|_W} = -T_{2|_W} \neq 0$ and $T_i = \prod_{j=1}^{d} \ell_{i,j}$ for $i \in [2]$. By unique factorization and without loss of generality we conclude that there are $\leq d$ distinct spaces $U_j = sp\{\ell_{1,j}, \ell_{2,j}\}$ for $j \in [d]$ that intersect $U = sp\{\ell_1, \ell_2\}$ non-trivially, i.e. $U \cap U_j$ is 1 dimensional. Then we have two subcases and in each we identify a co-dimension 1 subspace containing W. We sketch the idea of these subcases now and urge the reader to read the full proof in Appendix B

   (a) *There exist distinct U_i, U_j such that $U \cap U_i = U \cap U_j$:* In this case we show that $U_i \cap U_j$ is a one-dimensional($= sp\{\ell\}$) subset of $U \Rightarrow W \subset V = \{\ell = 0\}$. There are clearly $\leq d^4$ possibilities for U_i, U_j.

   (b) *For all distinct U_i, U_j, $U \cap U_i \neq U \cap U_j$:* Recall, we are given that $5 \leq rank(f) = dim(\sum_{i \in [d]} U_i)$. Using this we can conclude that, there are three spaces U_i, U_j, U_k such that $U_k \nsubseteq U_i + U_j$. In this case we show that $(U_i + U_j) \cap U_k$ is a one-dimensional($= sp\{\ell\}$) subset of $U \Rightarrow W \subset V = \{\ell = 0\}$. There are clearly $\leq d^6$ possibilities for U_i, U_j, U_k.

   (c) Using the above, we have a fixed set $\mathcal{A}$ of $d^{O(1)}$ size, as required in Part 2 above.

1.2.4 Theorem 3: Key ideas for Algorithm 7

The algorithm mentioned in Theorem 3 is presented in Algorithm 7 and it's correctness/complexity is discussed in Section 5.2. Here, we outline the key ideas involved in the different steps of our algorithm. We urge the reader to go through the missing details in Section 5.2.

1. First, we apply a random transformation on our input black-box, giving us black-box computing another polynomial g. An important aspect of this transformation is that we can freely assume(using Fact 1) that (with high probability) co-dimension 2 subspaces on which $NonLin(g)$ vanishes have the form $W = \{x_1 - \ell_1(x_3, \ldots, x_n) = 0, x_2 - \ell_2(x_3, \ldots, x_n) = 0\}$. So we only need to construct such spaces.

2. Next, we restrict $NonLin(g)$ to different combinations of 5 variables x_1, x_2, x_3, x_4, x_i at a time, and interpolate to get monomial representation. For each such restriction, we compute all co-dimension 2 subspaces of the type $\{x_1 = y_3x_3 + y_4x_4 + y_ix_i, x_2 = z_3x_3 + z_4x_4 + z_ix_i\}$, on which the restriction vanishes. This is done by substituting for x_1, x_2 in the restricted polynomials and then solving for $y_3, y_4, y_i, z_3, z_4, z_i$, by equating coefficient polynomials to zero. Application of Theorem 3 on the restricted polynomials(along with some simple facts presented in Fact 1) guarantees that with high probability there are $d^{O(1)}$ many solutions, and so we compute them by using Lemma 2.

3. Then, we merge the co-dimension 2 subspaces that were obtained for different restrictions. By applying another random transformation on variables, we ensure efficient merging of the spaces. Correctness is easily guaranteed with high probability using Facts 1 and 2.

4. After merging, as discussed above, we have a $(nd)^{O(1)}$ sized set of co-dimension 2 subspaces. We prune this set and remove the spaces on which $NonLin(g)$ does not vanish(using Lemma 3). Finally, we apply the inverse transformation, thereby giving us co-dimension 2 subspaces on which $NonLin(f)$ vanishes.

2 Preliminaries

2.1 Notations and definitions

Throughout the paper $[n]$ will denote the set $\{1,\ldots,n\}$ 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 of variables. Bold capital letters $\mathbf{A},\mathbf{B}$ are used to represent matrices. Unless otherwise mentioned, 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 or are used to represent vector spaces of linear forms. Small letters f,g,h,u,ℓ are also used to denote polynomials and linear forms. Next, we give some definitions that are used in the paper.

Definition 1 (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 set of n variables. First and third* (Σ) *layers have addition nodes and second* (Π) *layer has multiplication nodes. Top layer has a single addition node.*

Definition 2 (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*[9] *on n variables, there are k multiplication nodes at the second* (Π) *layer all having in-degree equal to 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 *of the following form.*

$$f = G\times T_1 + G\times T_2$$

where G,T_1,T_2 *are product of linear forms with* $gcd(T_1,T_2)=1$ *and* $deg(T_1)=deg(T_2)$*, such that polynomials* $G\times T_1, G\times T_2$ *are computed at the multiplication gates.*

Definition 3 (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 with multiplication gates computing products of linear forms* $G\times T_1$ *and* $G\times T_2$ *with* $gcd(T_1,T_2)=1$ *as described above, then the* rank *of* C *is defined as*

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

Definition 4 (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 5 (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* $NonLin(f)=\frac{f}{Lin(f)}$*.*

Definition 6 (Candidate linear form)**.** *Let* f *be a polynomial computable by a homogeneous* $\Sigma\Pi\Sigma(2,n,d,\mathbb{F})$ *circuit. Consider any non-zero* $n-$*variate linear form satisfying*

1. $NonLin(f)_{|_{\{l=0\}}}$ *is a non-zero product of linear forms.*
2. *There exist linear forms* ℓ_1,ℓ_2 *with* ℓ,ℓ_1,ℓ_2 *linearly independent such that for* $i\in[2]$*,*

$$NonLin(f)_{|_{\{l=0,l_i=0\}}} = 0.$$

Define set $\mathcal{L}(f)$ *to be a collection of such linear forms (modulo scalar multiplication)*[10]*. Elements*

[9] no constant term

[10] basically, the set is defined in the projective space

of $\mathcal{L}(f)$ are called "candidate linear forms".

Definition 7 (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).*

Definition 8 (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 $s \notin sp\{t\}$, the vector space $sp\{s,t\}$ is called an Ordinary line from s into $\mathcal{S}$, if and only if $sp\{s,t\} \cap \mathcal{S} \subseteq \{s,t\}$. Define $\mathcal{O}(s,\mathcal{S})$ to be the set of ordinary lines from s into $\mathcal{S}$.*

Definition 9 (Sylvester Gallai (SG) configuration, Definition 5.3.1, [Dvi12]). *A proper set $\mathcal{S} = \{s_1, \ldots, s_m\} \subset \mathbb{F}^n$ 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 2 (Solving polynomial equations, Implied from [Ier89, Laz01]). *There is a randomized algorithm that takes as input n variate polynomials $f_1, \ldots, f_m$ each of degree $\leq d$. If the system of equations defined by setting these all 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 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,*

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

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 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 $f_1, \ldots, f_m$ ($m \leq d$) along with integers $e_1, \ldots, e_m$ such that,*

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

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 $\{\ell_1, \ldots, \ell_s\}$ ($s \leq d$) of affine forms along with black-box access to a polynomial $NonLin(f)$ such that,*

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

Proof. We give the algorithm below. Correctness and time complexity proofs are pretty straightforward using Lemma 6 and Lemma 3.

Algorithm 2 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 6 on black-box computing f, obtain black-box access to polynomials $f_1, \ldots, f_m$ along with integers $e_1, \ldots, e_m$. Initialize lists $L, B \leftarrow \phi$.
2. For every $i \in [s]$, 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 $\mathbf{0} = (0, \ldots, 0) \in \mathbb{F}^n$. 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** $L, \mathcal{B}$.

□

2.3 Known facts

We list a few useful facts below without proof since they can be easily derived from popularly known results. These are used later in the paper. Let $\mathbb{F}$ be a finite field and $f(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$. Parts 1, 2, 3 of Fact 1 and Fact 2 can be easily proved by applying the Schwartz-Zippel lemma [Sch80, Zip79]. Part 4 of Fact 1 can be proved using Effective Hilbert irreducibility [Kal91].

Fact 1. *Let $V_i = \{x_5 = \ldots = x_{i-1} = x_{i+1} = \ldots = x_n\} \subset \mathbb{F}^n$ be a co-dimension 5 subspace. Let $\mathbf{A} \in \mathbb{F}^{n \times n}$ be a matrix with entries chosen independently and uniformly at random from $\mathbb{F}$, then the following are true with probability $1 - o(1)$.*

1. *$\mathbf{A}$ is invertible.*
2. *ℓ_1, ℓ_2 be linearly independent linear forms in $\mathbb{F}[x_1, \ldots, x_n]$, then the vector space $sp\{\ell_1(\mathbf{Ax}), \ell_2(\mathbf{Ax})\}$ is also spanned by some linear forms of the type $x_1 - \ell_1'(x_3, \ldots, x_n)$ and $x_2 - \ell_2'(x_3, \ldots, x_n)$ where ℓ_1', ℓ_2' are linear forms in $\mathbb{F}[x_3, \ldots, x_n]$.*
3. *Let f be computable by a $\Sigma\Pi\Sigma(2, n, d, \mathbb{F})$ circuit such that $rank(f) \geq 5$ and $g(\mathbf{x}) = f(\mathbf{Ax})$, then $rank(g_{|_{V_i}}) = 5$.*
4. *Let $NonLin(f)$ be the "non-linear" part of f as defined in Definition 5. and $g(\mathbf{x}) = f(\mathbf{Ax})$. Then $NonLin(g_{|_{V_i}}) = NonLin(g)_{|_{V_i}}$.*

Fact 2. *Let $\mathbf{B} = (B_{i,j})$ be a random matrix defined as:*

$$B_{i,j} = \begin{cases} 1 & i = j \\ uniform(\mathbb{F}) & (i,j) \in [5,n] \times [3,4] \\ 0 & otherwise \end{cases}$$

where $uniform(\mathbb{F})$ is chosen independently and uniformly at random from $\mathbb{F}$. Let $\mathbf{x}$ be a tuple of variables which is zero everywhere except x_3, x_4 and x_i. For any two distinct tuples of linear forms $(x_1 - \ell_1(\mathbf{x}), x_2 - \ell_2(\mathbf{x}))$ and $(x_1 - \ell'_1(\mathbf{x}), x_2 - \ell'_2(\mathbf{x}))$, the tuples $(x_1 - \ell_1(\mathbf{Bx})_{|_{\{x_i=0\}}}, x_2 - \ell_2(\mathbf{Bx})_{|_{\{x_i=0\}}}$ and $(x_1 - \ell'_1(\mathbf{Bx})_{|_{\{x_i=0\}}}, x_2 - \ell'_2(\mathbf{Bx})_{|_{\{x_i=0\}}})$ are distinct.

3 Low Rank Reconstruction: Proof of Theorem 1

We first present Algorithm 3 which proves Theorem 1. Then we analyze it's correctness and running-time. Our algorithm has three main stages: *Separate linear/non-linear parts → Interpolate non-linear part → Build Circuit.*

Algorithm 3 Low rank reconstruction

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

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

   ▷ *Separate linear/non-linear parts*

2. For each $r \in [\log^3 d]$, iterate over all r sized sets $\mathcal{T} \subset \mathcal{S}(f)$. If the vector space spanned by all linear forms from all tuples in $\mathcal{T}$ is r dimensional, find a basis of linear forms $\{y_1, \ldots, y_r\}$ spanning this space. Interpolate $NonLin(f)$ in the monomial basis $\{\mathbf{y^a} = y_1^{a_1} \times \ldots \times y_r^{a_r} : \mathbf{a} = (a_1, \ldots, a_r), \sum a_i = t\}$ to get coefficients $c_{\mathbf{a}}$ such that $NonLin(f) = \sum_{\mathbf{a}} c_{\mathbf{a}} \mathbf{y^a}$. If interpolation was successful go to next step. If it wasn't successful for any r, and any r sized subset then **Return** $\#$. ▷ *Interpolate non-linear part*

3. By creating appropriate multiplication/addition gates, construct a $\Sigma\Pi\Sigma(t^r, n, d, \mathbb{F})$ circuit C that computes polynomial

$$f' = \ell_1 \times \ldots \times \ell_s \times (\sum_{\mathbf{a}} c_{\mathbf{a}} \times y_1^{a_1} \times \ldots \times y_r^{a_r})$$

   Finally, **Return** C. ▷ *Build Circuit*

We outline the correctness and complexity of the algorithm below. Using Definition 2, we write $f = G \times (T_1 + T_2)$ where G, T_1, T_2 are product of linear forms and $gcd(T_1, T_2) = 1$.

1. *Separate linear/non-linear parts(Step 1):* Obvious using Algorithms 2, 7.

2. *Interpolate non-linear part(Step 2):* In this stage, we try to interpolate the degree t polynomial $NonLin(f)$(Definition 5) as a linear combination of degree t monomials, defined over some linearly independent set of linear forms. Note that $NonLin(f)$ divides $T_1 + T_2$ and therefore the interpolation can be done using any basis of the space V spanned by linear forms dividing

$T_1 \times T_2$. The following lemma describes a way to construct a basis of V using set $\mathcal{S}(f)$, containing tuples of linear forms representing co-dimension 2 subspaces on which $NonLin(f)$ vanishes. For the sake of presentation, we move the proof to Section 3.1.

Lemma 7. *There exist $rank(f)$ many tuples in $\mathcal{S}(f)$, such that the space spanned by all linear forms in these tuples is same as V.*

We only know that $rank(f) \leq \log^3 d$, so we non-deterministically search for it using $r \in [\log^3 d]$. Then, we non-deterministically search for an r sized subset $\mathcal{T} \subset \mathcal{S}(f)$, such that linear forms from tuples in $\mathcal{T}$ span an r dimensional space. If r is correctly guessed (i.e. $r = rank(f)$), Lemma 7 implies the existence of such a set which spans V. If our guess for r and $\mathcal{T}$ are correct, then using the r tuples in $\mathcal{T}$, we can create a basis $\{y_1, \ldots, y_r\}$ of V and interpolate $NonLin(f)$ in the monomial basis of y_i's, using black-box access to it. So for the correct guesses, we will always be able to interpolate. An important point to note is that even for incorrect guesses, if the interpolation is successful, it is guaranteed to be correct. With high probability, our search would have reconstructed the circuit for $r \leq rank(f)$. Hence at the end of the algorithm we have a $\Sigma\Pi\Sigma(k, n, t, \log|\mathbb{F}|)$ circuit computing $NonLin(f)$ with $k \leq t^{rank(f)}$. By Part 1 of Theorem 3, $\mathcal{S}(f)$ has size $d^{O(1)}$ and therefore the non deterministic search has $d^{O(\log^3 d)}$ iterations. For every such iteration of the search, basis computation can be done in $(nd \log|\mathbb{F}|)^{O(1)}$ time, and interpolation of $NonLin(f)$ takes $(nt^{\log^3 d} \log|\mathbb{F}|)^{O(1)}$ time. Therefore over all we take $(nd^{\log^3 d} \log|\mathbb{F}|)^{O(1)}$ time.

3. *Build Circuit(Step 3):* The correctness and complexity of this stage are pretty straightforward. It's easy to see we output a $\Sigma\Pi\Sigma(k, n, d, \log|\mathbb{F}|)$ circuit computing f, with $k \leq t^{rank(f)}$, and take $(nd^{\log^3 d} \log|\mathbb{F}|)^{O(1)}$ time.

3.1 Proof of Lemma 7

Since $rank(f) \geq 5$, we know that $NonLin(f)$ is a non-constant polynomial. Consider any linear form ℓ_1 dividing T_1. Since $gcd(T_1, T_2) = 1$, it's easy to see that $NonLin(f)_{|_{\{\ell_1=0\}}}$ divides $T_{2|_{\{\ell_1=0\}}} \neq 0$. Hence, there exists an ℓ_2 dividing T_2 such that $\ell_{2|_{\{\ell_1=0\}}}$ is a factor of $NonLin(f)_{|_{\{\ell_1=0\}}}$, further implying that $NonLin(f)$ vanishes on the co-dimension 2 space $\{\ell_1 = 0, \ell_2 = 0\}$. Therefore $\mathcal{S}(f)$ contains a tuple (p, q) of linear forms representing this co-dimension 2 space. It's easy to see that $sp\{\ell_1, \ell_2\} = sp\{p, q\}$. Similarly for every linear form ℓ_2 dividing T_2 there exists some linear form ℓ_1 dividing T_1, such that there is tuple $(p, q) \in \mathcal{S}(f)$ satisfying $sp\{\ell_1, \ell_2\} = sp\{p, q\}$. Now if we just use this argument with $rank(f)$ many independent linear forms dividing T_1 or T_2, we will end up with $rank(f)$ many tuples in $\mathcal{S}(f)$ spanning the space V spanned by linear factors of T_1 and T_2.

4 High Rank Reconstruction: Proof of Theorem 2

The algorithm in Theorem 2 is presented in Algorithm 1 and analysis is presented in Section 1.2.2. Algorithm 1 uses Algorithms 4, 5 and 6. We present and analyze them in Sections 4.1, 4.2 and 4.3 respectively. Also, to complete our analysis given in Section 1.2.2 we provide a proof to Lemma 11 in Section 4.4. This lemma was used in stage *"Compute linearly independent set dividing T_i"* of Algorithm 1. Since $f \in \Sigma\Pi\Sigma(2, n, d, \mathbb{F})$ (Definition 2), throughout this section we will assume a

representation of the form

$$f = G \times (T_1 + T_2)$$

where G, T_1, T_2 are product of linear forms such that $gcd(T_1, T_2) = 1$. We define $V_i = sp(\{\text{linear form } \ell : \ell \mid T_i\})$ for $i \in [2]$. In the next subsection, we explain construction of the set of candidate linear forms (Definition 6).

4.1 Computing Candidate Linear forms

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

Lemma 8. *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,*

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

Our algorithm for the above lemma has three main stages: *Separate linear/non-linear parts → Join co-dimension 2 subspaces into candidates → Restrict non-linear parts and check factorization.* We first give the algorithm and then discuss correctness and complexity of each stage.

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 2 with inputs as black-box access to f and integers n, d, obtain list of linear factors $\ell_1, \ldots, \ell_s$ and access to black-box computing $NonLin(f)$. Compute degree of $NonLin(f)$ as $t = d - s$. ▷ *Separate linear/non-linear parts*

2. Using Algorithm 7, compute the set $\mathcal{S}(f)$ of tuples of linear forms representing co-dimension 2 subspaces on which $NonLin(f)$ vanishes. Initialize $\mathcal{L} \leftarrow \phi$. For all pairs of tuples $(p_1, q_1), (p_2, q_2) \in \mathcal{S}(f)$, check if $sp\{p_1, q_1\} \cap sp\{p_2, q_2\} = sp\{\ell\}$ (i.e. is one dimensional), for some linear form ℓ. If yes and no scalar multiple of ℓ is already present in $\mathcal{L}$, then add ℓ to $\mathcal{L}$. ▷ *Join co-dimension 2 subspaces into candidates*

3. For each $\ell \in \mathcal{L}$, simulate black-box computing $NonLin(f)_{|_{\{\ell=0\}}}$. Using Lemma 3, check if this black-box computes the 0 polynomial. If 'yes', remove ℓ from $\mathcal{L}$. Otherwise, using Algorithm 2, 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** $\mathcal{L}$.

▷ *Restrict non-linear parts and check factorization*

We outline the correctness and complexity of all stages below.

1. *Separate linear/non-linear parts(Step 1):* Obvious using Algorithm 2.

2. *Join co-dimension 2 subspaces into candidates(Step 2):* This stage tries to obtain all linear forms which satisfy second condition of Definition 6. Correctness is clear from Definition 6 and Algorithm 7. Algorithm 7 runs in $(nd \log |\mathbb{F}|)^{O(1)}$ time. Part 1 of Theorem 3 implies $|\mathcal{S}(f)| \leq 3d^7$, therefore iterating over all pairs of tuples in $\mathcal{S}(f)$, and taking intersection of the

2 dimensional space they form takes $(nd \log |\mathbb{F}|)^{O(1)}$ time. Size bound on $\mathcal{S}(f)$ also implies that at the end of this stage $\mathcal{L}$ has $\leq 9d^{14}$ elements.

3. *Restrict non-linear parts and check factorization(Step 3):* At the end of the previous stage, we have all linear forms ($d^{O(1)}$ many) that satisfied second condition of Definition 6. In this stage we select the linear forms which also satisfy the first condition of Definition 6. Lemma 3 efficiently checks for non-zeroness, and then Algorithm 2 computes all linear factors of $NonLin(f)_{|_{\{\ell=0\}}}$, and checks whether there are $t = deg(NonLin(f))$ such factors. Correctness/Complexity of Lemma 3 and Algorithm 2 clearly imply that this stage takes $(nd \log |\mathbb{F}|)^{O(1)}$ time.

4.2 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 9. *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 5. The algorithm has two stages: *Compute and merge restrictions* → *Reconstruct and validate*. We use Algorithm 5 of [Shp07] in the "Compute and merge restrictions" stage to merge a collection of multi-sets corresponding to restrictions of linear forms. More details on this merge algorithm can be found in Algorithm 5 and Theorem 29 of [Shp07].

Algorithm 5 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. Extend $\{u_1, \ldots, u_t\}$ to a basis $\{u_1, \ldots, u_n\}$ and define invertible transformation $u_i \mapsto x_i$ using matrix $\mathbf{A}$ such that $\mathbf{x} = \mathbf{Au}$ ($\mathbf{u} = (u_1, \ldots, u_n)$). Simulate blackbox computing $h(\mathbf{x}) = f(\mathbf{Ax})$. Using Algorithm 2 with inputs as black-box computing h and integers n, d, obtain all linear factors $\ell_1, \ldots, \ell_s$ of h and black-box access to $NonLin(h)$. Compute degree of $NonLin(h)$ as $t = d - s$. Remove all multiples of $x_1, \ldots, x_t$ from $\ell_1, \ldots, \ell_s$ and without loss of generality assume that $\ell_1, \ldots, \ell_q$ are left after removal. Simulate black-box computing $g = \ell_1 \times \ldots \ell_q \times NonLin(h)$.

2. For each $i \in [t]$, simulate black-box computing $g_{|_{\{x_i=0\}}}$ and using Algorithm 2 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 $\mathcal{U}$. ▷ *Compute and merge Restrictions*

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 the two sets. Simulate black-box computing $g - \alpha \prod_{\ell \in \mathcal{U}} \ell$ and factorize this polynomial using Algorithm 2. If all factors are not linear, **Return** $\#$. Otherwise, store factors in multi-set $\mathcal{V}$. Apply inverse transformation $\mathbf{x} \mapsto \mathbf{A^{-1}x}$ to all linear forms in $\mathcal{U}, \mathcal{V}$. Simulate black-box for $f - f'$, where

$$f' = \prod_{i=1}^{t} u_i^{e_i} \times (\alpha \prod_{\ell \in \mathcal{U}} \ell + \prod_{\ell \in \mathcal{V}} \ell)$$

where e_i is number of multiples of x_i in the linear factors $\ell_1, \ldots, \ell_s$. 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** $\#$.

▷ *Reconstruct and validate*

We outline the correctness and complexity of all stages below.

1. *Compute and merge restrictions(Steps 1,2):* In this stage, we make some preparations to use Algorithm 5 from [Shp07]. We first map our linearly independent set $\{u_1, \ldots, u_t\}$ onto the variables $\{x_1, \ldots, x_t\}$ using an invertible transformation, apply it to the input black-box and simulate black-box computing $h(\mathbf{x}) = f(\mathbf{Ax})$. All these steps clearly run in $(n \log |\mathbb{F}|)^{O(1)}$ time. Using Algorithm 2, for all x_j we remove its multiples from h. Since u_j divided T_i, the polynomial g that remains still exhibits a $\Sigma\Pi\Sigma(2, n, d, \mathbb{F})$ circuit, say $g = H \times (S_1 + S_2)$ with H, S_1, S_2 being product of linear forms and $gcd(S_1, S_2) = 1$. It's easy to see that reconstructing

this circuit is enough. Without loss of generality all x_j divide S_1 implying,

$$g_{|_{\{x_j=0\}}} = (H \times S_2)_{|_{\{x_j=0\}}} \neq 0.$$

We factorize the above using Algorithm 2, thereby computing multi-sets $\mathcal{U}_j$ containing all linear factors of $(H \times S_2)_{|_{\{x_j=0\}}}$ correctly in $(nd \log |\mathbb{F}|)^{O(1)}$ time. Since $|\mathcal{U}_j| \leq d$ and $t > 60 \log d + 61$, all conditions of Algorithm 5([Shp07]) are met, and we use it to merge the $\mathcal{U}_j$'s and obtain $\mathcal{U}$ such that $H \times S_2 = \alpha \prod_{\ell \in \mathcal{U}} \ell$ for some $\alpha \in \mathbb{F}$, as guaranteed by Theorem 29 of [Shp07]. Running time of Algorithm 5([Shp07]) can be easily seen to be $(nd \log |\mathbb{F}|)^{O(1)}$ time.

2. *Reconstruct and validate(Step 3):* In the previous stage, we computed a multi-set $\mathcal{U}$ containing all linear factors of $H \times S_2$ (upto scalar multiplication). To reconstruct, we first compute the scalar α described above. We know that $H \times S_2 = \alpha \prod\limits_{\ell \in \mathcal{U}} \ell$. On restricting to $\{x_1 = 0\}$, we see that

$$\alpha = (H \times S_2)_{|_{\{x_1=0\}}} \Big/ \prod_{\ell \in \mathcal{U}} \ell_{|_{\{x_1=0\}}}.$$

Thus, α is easily computed by first computing set $\mathcal{U}' = \{\ell_{|_{\{x_1=0\}}} : \ell \in \mathcal{U}\}$ and then matching sets $\mathcal{U}_1$ and $\mathcal{U}'$. Once α is computed, we can simulate black-box computing $g - \alpha \prod_{\ell \in \mathcal{U}} \ell$ and factorize it using Algorithm 2 to get all linear factors of $H \times S_1$. Then we apply inverse transformation $\mathbf{x} \mapsto \mathbf{A^{-1}x}$ on all linear forms in $\mathcal{U}, \mathcal{V}$. Using $\ell_1, \ldots, \ell_s$ from Step 1, we can easily find the largest power e_i of x_i that divides h and then construct a $\Sigma\Pi\Sigma(2, n, d, \mathbb{F})$ circuit C computing,

$$f' = \prod_{i=1}^{t} u_i^{e_i} \times (\alpha \prod_{\ell \in \mathcal{U}} \ell + \prod_{\ell \in \mathcal{V}} \ell)$$

Finally we validate our solution by checking if $f - f' \equiv 0$. For this we simulate black-box computing $f - f'$ using C. Note that $f - f'$ can be computed by a $\Sigma\Pi\Sigma(4, n, d, \mathbb{F})$ circuit. Therefore using Lemma 4, we can check whether $f - f' \equiv 0$ in $(nd \log |\mathbb{F}|)^{O(1)}$ time and output C if the test returns 'yes' and # otherwise. Due to this, our output will always be correct.

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

Suppose we know that one of the multiplication gates is power of a linear form (upto scalar multiplication) y_1 over $\mathbb{F}$. In this case we need slightly different techniques. Here is a lemma summarizing the reconstruction algorithm in this case.

Lemma 10. *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.*

Next, we give the algorithm mentioned in the above lemma. Our algorithm has three stages: *Obtain G and $T_1 + T_2$ → Guess y_1 and compute α s.t. $T_i = \alpha y_1^t$ → Reconstruct other gate and validate.*

Algorithm 6 A 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 2 with inputs as black-box access to f and integers n, d compute linear factors $\ell_1, \ldots, \ell_s$ and get access to black-box computing $NonLin(f)$. Compute degree of $NonLin(f)$ as $t = d - s$. ▷ *Obtain G and $T_1 + T_2$*

2. Using Algorithm 4, compute set $\mathcal{L}(f)$. Iterate over linear forms $y_1 \in \mathcal{L}(f)$.

   (a) Simulate black-box for $NonLin(f)|_{\{y_1=0\}}$ and using Algorithm 2 identify two linearly independent factors say y_2, y_3. Extend $\{y_1, y_2, y_3\}$ to a linearly independent set $\{y_1, \ldots, y_n\}$. Compute invertible matrix $\mathbf{A}$ such that $\mathbf{x} = \mathbf{A}\mathbf{y}$. Simulate black-box for

   $$g(x_1, x_2, x_3) = NonLin(f)(\mathbf{A}\mathbf{x})|_{\{x_4=\ldots=x_n=0\}}$$

   (b) Interpolate g in monomial basis of $\mathbb{F}[x_1, x_2, x_3]$. Substitute $x_2 = \beta x_1$ in all monomials and rearrange to get a representation in $\mathbb{F}[\beta][x_1, x_3]$. Equate coefficient polynomials of monomials containing x_3 to 0 and solve the resulting system of equations using Lemma 2. If all y_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.
   ▷ *Guess y_1 and compute α s.t. $T_i = \alpha y_1^t$*

   (c) Iterate over all α's in the set of scalars obtained above. Simulate black-box for $NonLin(f) - \alpha y_1^t$ and using Algorithm 2 check if it has t linear factors say $\ell_{s+1}, \ldots, \ell_{s+t}$. If not, then go to the next α. If all α have been tried, go to next $y_1 \in \mathcal{L}(f)$. If all y_1's have been tried, **Return** #. Otherwise, simulate black-box for $f - f'$, where

   $$f' = \ell_1 \times \ldots \times \ell_s \times (\alpha y_1^t + \ell_{s+1} \times \ldots \times \ell_{s+t})$$

   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 α. If all α have been tried, go to next $y_1 \in \mathcal{L}(f)$. If all y_1's have been tried, **Return** #.
   ▷ *Reconstruct other gate and validate*

We outline the correctness and complexity of all the stages below.

1. *Obtain G and $T_1 + T_2$ (Step 1):* We claim the following. Proof of this claim is provided in Appendix A.

   Claim 1. *Assume $T_i = \alpha y_1^t$, for some $i \in [2]$, $\alpha \in \mathbb{F}$ and linear form y_1. Then $Lin(f) = G$ ($NonLin(f) = T_1 + T_2$).*

   This claim implies that in Step 1, using Algorithm 2, we obtain all linear factors of G and black-box access to $T_1 + T_2$ in $(nd \log |\mathbb{F}|)^{O(1)}$ time.

2. *Guess y_1 and compute α s.t. $T_i = \alpha y_1^t$ (Steps 2(a),2(b)):* We claim that y_1 is already in $\mathcal{L}(f)$.

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 $\mathcal{L}(f)$.*

Using the above claim, in this stage, we non-deterministically search for y_1 inside $\mathcal{L}(f)$ and use it to compute a set of size $\leq t$ which contains α satisfying $T_i = \alpha y_1^t$. In last stage, the reconstructed circuit is validated, and rejected if incorrect. So we show that the set of scalars obtained at end of this stage contains the correct α when y_1 is chosen correctly.

(a) This implies that $(T_1 + T_2)_{|_{\{y_1=0\}}} = T_{3-i}|_{\{y_1=0\}}$ is a non-zero product of linear forms. Since $rank(f) \geq 5$, using Algorithm 2 on black-box computing $T_1 + T_2$ (that was obtained in previous stage), we can easily find independent linear factors y_2, y_3 of $T_{3-i|_{\{y_1=0\}}}$ in $(nt \log |\mathbb{F}|)^{O(1)}$ time. Clearly, there is some $\beta \in \mathbb{F}$ such that $(y_2 - \beta y_1)$ divides T_{3-i}. Computing these β's will lead to us to computing a small set containing α.

(b) Extending $\{y_1, y_2, y_3\}$ to basis $\{y_1, \ldots, y_n\}$, application of invertible transformation $y_i \mapsto x_i$ and restriction to $\{x_4 = \ldots = x_n = 0\}$ on black-box computing $(T_1 + T_2)$ are easily done in $(n \log |\mathbb{F}|)^{O(1)}$ time. Clearly $g(x_1, x_2, x_3) = \alpha x_1^t + (x_2 - \beta x_1)h(x_1, x_2, x_3)$ for some polynomial h. In $(t \log |\mathbb{F}|)^{O(1)}$ time, we interpolate g in monomial basis of $\mathbb{F}[x_1, x_2, x_3]$. It's easy to see that g depends on x_3 (y_3 divides $T_{3-i|_{\{y_1=0\}}}$), but $g(x_1, \beta x_1, x_3)$ is independent of x_3. We use this observation to compute set containing β.

(c) Substituting $x_2 = \beta x_1$ and computing coefficient polynomials in $\mathbb{F}[\beta]$ is easily done in $(t \log |\mathbb{F}|)^{O(1)}$ time. There are $\leq t^{O(1)}$ many univariate coefficient polynomials(each of degree $\leq t$) corresponding to monomials containing x_3. The system of equations defined by these polynomials have $\leq t$ solutions since they are univariates. All solutions are found in $(t \log |\mathbb{F}|)^{O(1)}$ time using Lemma 2. Substituting $x_2 = \beta x_1$ with the correct β, reduces g to αx_1^t and therefore gives α. So we plug all solutions for β and create a set of size $\leq t$ which contains α.

3. *Reconstruct other gate and validate(Step 2(c)):* We non-deterministically search for α in set created during previous stage. As discussed earlier, we validate later for checking correctness. Thus we only need to show that for correct choice of y_1 and α, circuit is reconstructed correctly. Note, when y_1, α are correct, $(T_1 + T_2) - \alpha y_1^t$ is a product of linear forms, say $\ell_{s+1}, \ldots, \ell_{s+t}$, which are found using Algorithm 2 in $(nd \log |\mathbb{F}|)^{O(1)}$ time. Finally we validate our solution by checking if $f - f' \equiv 0$, where $f' = \ell_1 \times \ldots \times \ell_s \times (\alpha y_1^t + \ell_{s+1} \times \ldots \times \ell_{s+t})$. Polynomial $f - f'$ is clearly computable by a $\Sigma\Pi\Sigma(4, n, d, \mathbb{F})$ circuit and a black-box computing it can be simulated in $(nd \log |\mathbb{F}|)^{O(1)}$ time using definition of f'. Lemma 4, checks whether $f - f' \equiv 0$ in $(nd \log |\mathbb{F}|)^{O(1)}$ time. If this validation fails we try the next guesses for α, y_1. It's easy to see that the search over α, y_1 happens at most $d^{O(1)}$ times. The validation ensures that we only output the correct circuit.

4.4 Identify Linearly Independent Set Dividing T_i

In this subsection, our goal is to provide proof of Lemma 11. It plays a crucial role in Algorithm 1 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 and ordinary lines among them. First, in Section 4.4.1 below we give intuition about why set of candidate linear forms approximates set of linear factors of $T_1 \times T_2$ and then in Section 4.4.2 use this set of candidate linear forms in Lemma 11 to construct the required linearly independent set. Define $V_i = sp(\{\text{linear form } \ell : \ell \mid T_i\})$ for $i \in [2]$. Results in this subsection are true under the assumption $dim(V_i) \geq 2$. for $i = 1, 2$.

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

In order to quantify how close the candidate set is to the set of linear forms in the input circuit, we define some new sets.

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

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

For all sets, we only keep linear forms upto scalar multiplication and therefore treat them as proper sets (Definition 7). $\mathcal{L}_{good}$ 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 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 $\mathcal{L}(f)$ 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 11

In this subsection, we prove Lemma 11 which was used by Algorithm 1. 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 8.

Lemma 11. *The following are true.*

1. $\exists\ \ell \in \mathcal{L}_{good}$ *such that ordinary lines from* ℓ *into* $\mathcal{L}(f)$ *span an* $\Omega(\log^2 d)$ *dimensional space* $\Rightarrow \exists$ *a linearly independent subset* $\mathcal{X} \subset \mathcal{L}(f)$ *of size* $\Omega(\log^2 d)$ *such that for all* $\ell' \in \mathcal{X}$, $sp\{\ell, \ell'\}$ *is an ordinary line from* ℓ *into* $\mathcal{L}(f)$.
2. *Let* $\mathcal{X}$ *be as described in the previous part. Then, 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. *Consider any linear form* $\ell' \in \mathcal{B}$. *Assume* ℓ *divides* T_i, *then we can show that*

$$NonLin(f)|_{\{\ell=0, \ell'=0\}} = 0 \Leftrightarrow \ell' \text{ divides } T_i.$$

Proof. We prove all parts one by one.

1. Let $\mathcal{T} \subset \mathcal{L}_{good}$ be a linearly independent set of size $126 \log d + 2$ (exists by Claim 3). Applying Lemma 1 on $\mathcal{L}(f)$ and $\mathcal{T}$ implies that there exists $\ell \in \mathcal{T}$ such that

$$dim(\sum_{W \in \mathcal{O}(\ell, \mathcal{L}(f))} W) \geq \frac{dim(sp(\mathcal{L}(f)))}{126 \log d + 2} \geq \frac{dim(sp(\mathcal{L}_{good}))}{126 \log d + 2} = \Omega(\log^2 d)$$

   Thus ordinary lines from ℓ into $\mathcal{L}(f)$ span a space of dimension $\Omega(\log^2 d)$ and therefore we can compute the linearly independent set $\mathcal{X}$ of size $4(\log d + 4)^2$ as required.

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 $\mathcal{L}_{good}, \mathcal{L}_{bad} \subset \mathcal{L}(f)$, 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 spance of $\Omega(\log d) \Rightarrow dim(sp(\mathcal{L}_{others} \cup \mathcal{L}_{factors})) = \Omega(\log d)$, contradicting Claim 3.

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

3. Without loss of generality assume ℓ divides T_1. For all $\ell' \in \mathcal{B}$, we show

$$NonLin(f)|_{\{\ell=0,\ell'=0\}} = 0 \Leftrightarrow T_2|_{\{\ell=0,\ell'=0\}} = 0 \Leftrightarrow \ell' \mid T_2,$$

   which completes the proof. Let H be product of all linear factors of $T_1 + T_2$. Therefore $(T_1 + T_2) = H \times NonLin(f)$. On restricting to the space $\{\ell = 0, \ell' = 0\}$, we get

$$T_2|_{\{\ell=0,\ell'=0\}} = (T_1 + T_2)|_{\{\ell=0,\ell'=0\}} = H|_{\{\ell=0,\ell'=0\}} \times NonLin(f)|_{\{\ell=0,\ell'=0\}}$$

   The first equality holds since $\ell \mid T_1$.

   - (First $\Leftrightarrow$) $NonLin(f)|_{\{\ell=0,\ell'=0\}} = 0 \Rightarrow T_2|_{\{\ell=0,\ell'=0\}} = 0$ is obvious from the above equation. The other direction is also true. If not then by the equation above, $H|_{\{\ell=0,\ell'=0\}} = 0 \Rightarrow sp\{\ell, \ell'\}$ intersects $\mathcal{L}_{factors}$. The intersection cannot happen inside $sp\{\ell\} \cup sp\{\ell'\}$ since both ℓ, ℓ' divide T_1 or T_2 and can't divide $T_1 + T_2$(as $gcd(T_1, T_2) = 1$).
   - (Second $\Leftrightarrow$) $\ell' \mid T_2 \Rightarrow T_2|_{\{\ell=0,\ell'=0\}} = 0$ is obvious. For the other direction, observe that, $T_2|_{\{\ell=0,\ell'=0\}} = 0 \Rightarrow$ there is some ℓ'' dividing T_2 such that $\ell'' \in sp\{\ell, \ell'\}$. Clearly $\ell'' \in \mathcal{L}_{good} \cup \mathcal{L}_{others}$ (since they together contain all linear factors of T_1, T_2). We know from the previous part that $sp\{\ell, \ell'\}$ is an ordinary line in $\mathcal{L}_{good} \cup \mathcal{L}_{others} \Rightarrow \ell'' \in sp\{\ell\} \cup sp\{\ell'\}$. $gcd(T_1, T_2) = 1$ implies that ℓ, ℓ'' are not scalar multiples, therefore $\ell'' \in sp\{\ell'\} \Rightarrow \ell' \mid T_2$, completing the proof.

□

5 Co-dim 2 subspaces where f vanishes: Proof of Theorem 3

In this section we prove Theorem 3. 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. Since our input polynomial f is computable by a $\Sigma\Pi\Sigma(2, n, d, \mathbb{F})$ circuit we write $f = G \times (T_1 + T_2)$, where G, T_1, T_2 are product of linear forms and $gcd(T_1, T_2) = 1$. Recall the definition of $NonLin(f)$ (Definition 5), which denotes the "non-linear" part of f. The set of co-dimension 2 subspaces on which $NonLin(f)$ vanishes is denoted by $\mathcal{S}(f)$[11].

5.1 Proof of Part 1

Let $W \subset \mathbb{F}^n$ be a co-dimension 2 subspace on which $NonLin(f)$ vanishes. Since $NonLin(f)$ divides $T_1 + T_2$ we get that $T_{1|_W} + T_{2|_W} = 0$. This implies that either $T_{1|_W} = T_{2|_W} = 0$, or $T_{1|_W} = -T_{2|_W} \neq 0$. We prove the following lemma which implies the bound.

Lemma 12. *The following are true.*

1. $\#\{W \subset \mathbb{F}^n : codim(W) = 2, T_{1|_W} = T_{2|_W} = 0\} \leq d^2$.
2. $\#\{W \subset \mathbb{F}^n : codim(W) = 2, T_{1|_W} = -T_{2|_W} \neq 0\} \leq d^5 + d^7$.

Proof of Part 1: We are given that $T_{1|_W} = T_{2|_W} = 0 \Rightarrow$ there are linear forms $\ell_1 \mid T_1$ and $\ell_2 \mid T_2$ such that $\ell_{1|_W} = \ell_{2|_W} = 0$. Also, ℓ_1, ℓ_2 are linearly independent since $gcd(T_1, T_2) = 1$. Since W is a co-dimension 2 subspace of $\mathbb{F}^n$, we get that $W = \{\ell_1 = 0, \ell_2 = 0\}$. There are $\leq d^2$ 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 13. *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 co-dimension 2 subspace $W \subset \mathbb{F}^n$ satisfying $T_{1|_W} = -T_{2|_W} \neq 0$, $\exists\ V \in \mathcal{A}$ with $W \subset V$.*

Assuming Lemma 13, we complete the proof as follows. For every W on which $NonLin(f)$ vanishes and $T_{1|_W} = -T_{2|_W} \neq 0$, we consider any $V = \{\ell = 0\} \in \mathcal{A}$ given by Lemma 13 such that $W \subset V$. We can now get a representation of W as $W = \{\ell = 0, \ell' = 0\}$ for some linear form ℓ'. From here we claim that $\ell'' = \ell'_{|_V}$ is then a linear form (in one less variable) that divides polynomial $NonLin(f)_{|_V}$, implying that there are $\leq d$ such ℓ'' upto scalar multiplication. Note that $W = \{\ell = 0, \ell' = 0\} = \{\ell = 0, \ell'' = 0\}$ and therefore there can be at most $(d^4 + d^6) \times d$ many such W's. Our claim is implied by the following easy to verify statement.

$$NonLin(f)_{|_W} = 0 \Rightarrow (NonLin(f)_{|_V})_{|_{\{\ell''=0\}}} = 0$$

5.2 Analysis of Algorithm 7

Our algorithm can be divided into five main stages: *Transform* → *Restrict* → *Compute* → *Merge* → *Validate*. We first give the algorithm and then discuss correctness and complexity of each stage.

[11] we compute this as a set of tuples of linearly independent linear forms

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. Pick a random matrix $\mathbf{A} \in \mathbb{F}^{n \times n}$. If $\mathbf{A}$ is not invertible, **Return** ϕ. Else, simulate black-box for $g(\mathbf{x}) = f(\mathbf{Ax})$. ▷ *Transform*

2. Using Algorithm 2 with inputs as black-box access to g along with integers n, d obtain black-box access to $NonLin(g)$ and integer s denoting the number of linear factors of g. Define $t = d - s$. For every $i \in [5, n]$, simulate black-box for the restricted polynomial

$$h_i = NonLin(g)(x_1, x_2, x_3, x_4, 0, \ldots, 0, x_i, 0, \ldots, 0).$$

   Using multivariate polynomial interpolation, with inputs as this black-box computing h_i and integer t, interpolate h_i as a homogeneous polynomial of degree t and get it's coefficients.

   ▷ *Restrict*

3. 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 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 $\mathcal{S}_i$. ▷ *Compute*

4. Construct a matrix $\mathbf{B} = (B_{i,j}) \in \mathbb{F}^{n \times n}$ as follows.

$$B_{i,j} = \begin{cases} 1 & i = j \\ uniform(\mathbb{F}) & (i,j) \in [5,n] \times [3,4] \\ 0 & otherwise \end{cases}$$

   Where $uniform(\mathbb{F})$ is sampled from the uniform distribution on $\mathbb{F}$. Replace each tuple $(p(\mathbf{x}), q(\mathbf{x})) \in \bigcup_{i \in [5,n]} \mathcal{S}_i$, with $(p(\mathbf{Bx}), q(\mathbf{Bx}))$. Initialize a set $\mathcal{S} \leftarrow \phi$.

5. For each tuple $(x_1 - \alpha x_3 - \beta x_4 - \gamma x_5, x_2 - \delta x_3 - \theta x_4 - \psi x_5) \in \mathcal{S}_5$, initialize linear forms $\ell_1 \leftarrow x_1 - \alpha x_3 - \beta x_4 - \gamma x_5$ and $\ell_2 \leftarrow x_2 - \delta x_3 - \theta x_4 - \psi x_5$.

   (a) For each $i \in [6, n]$, try to find tuple $(x_1 - \alpha x_3 - \beta x_4 - \kappa x_i, x_2 - \delta x_3 - \theta x_4 - \omega x_i) \in \mathcal{S}_i$ for some $\kappa, \omega \in \mathbb{F}$. If only one such tuple is found in $\mathcal{S}_i$ then update $\ell_1 \leftarrow \ell_1 - \kappa x_i$ and $\ell_2 \leftarrow \ell_2 - \omega x_i$. If multiple or no tuples are found in $\mathcal{S}_i$ then break out of this loop and go to the next tuple in Step 5.

   (b) Update $\mathcal{S} \leftarrow \mathcal{S} \cup \{(\ell_1(\mathbf{B^{-1}x}), \ell_2(\mathbf{B^{-1}x}))\}$ ▷ *Merge*

6. For each $(\ell_1, \ell_2) \in \mathcal{S}$, simulate black-box access to polynomial

$$NonLin(g)_{|_{\{\ell_1 = 0, \ell_2 = 0\}}}.$$

   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 $(\ell_1(\mathbf{A^{-1}x}), \ell_2(\mathbf{A^{-1}x}))$. **Return** $\mathcal{S}$. ▷ *Validate*

We outline the correctness and complexity of all the stages below.

1. *Transform(Step 1):* In this stage, we apply a random transformation on our variables to help us deal with some degeneracy later. By Part 1 of Fact 1, $\mathbf{A}$ is invertible and we proceed. Black-box for g can be easily simulated in $(n \log |\mathbb{F}|)^{O(1)}$ time. Since $rank(g) \geq 5$, Part 1 of Theorem 3 implies that $|\mathcal{S}(g)| = d^{O(1)}$ and we try to compute this set first. It's easy to see that $NonLin(f)$ vanishes on $\{\ell_1 = 0, \ell_2 = 0\} \Leftrightarrow NonLin(g)$ vanishes on $\{\ell_1(\mathbf{Ax}) = 0, \ell_2(\mathbf{Ax}) = 0\}$. Part 2 of Fact 1 implies that every co-dimension 2 subspace on which $NonLin(g)$ vanishes has the form $W = \{x_1 - \ell_1(x_3, \ldots, x_n) = 0, x_2 - \ell_2(x_3, \ldots, x_n) = 0\}$. So we only need to construct such spaces.

2. *Restrict(Step 2):* In this stage, for each $i \in [5, n]$ we compute restriction of $NonLin(g)$ to subspaces $V_i = \{x_5 = \ldots = x_{i-1} = x_{i+1} = \ldots x_n = 0\}$ and then in $(t \log |\mathbb{F}|)^{O(1)}$ time interpolate to get their monomial representation in $\mathbb{F}[x_1, x_2, x_3, x_4, x_i]$. Algorithm 2 implies that $NonLin(g)$ is correctly computed from black-box computing g in $(nd \log |\mathbb{F}|)^{O(1)}$ time. Parts 4 and 3 of Fact 1 imply that $NonLin(g_{|_{V_i}}) = NonLin(g)_{|_{V_i}}$ (upto scalar multiplication) and that $rank(g_{|_{V_i}}) \geq 5 \Rightarrow |\mathcal{S}(g_{|_{V_i}})| = d^{O(1)}$ Part 1 of Theorem 3. These further imply that for all co-dimension 2 subspaces $\{\ell_1 = 0, \ell_2 = 0\}$ on which $NonLin(g)$ vanishes, the restriction $NonLin(g)_{|_{V_i}}$ vanishes on $\{\ell_{1|_{V_i}} = 0, \ell_{2|_{V_i}} = 0\}$, and,$NonLin(g)_{|_{V_i}}$ vanishes on at most $d^{O(1)}$ co-dimension 2 subspaces. So we first try to compute co-dimension 2 subspaces on which $NonLin(g)_{|_{V_i}}$ vanishes.

3. *Compute(Step 3):* To compute co-dimension 2 subspaces on which $NonLin(g)_{|_{V_i}}$ vanishes, we simply 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 every monomial and set the resulting polynomial to 0. This gives us a system of $t^{O(1)}$ many polynomial equations of degree $\leq t$ in 6 variables $y_3, y_4, y_i, z_3, z_4, z_i$. It's easy to see that the coefficient polynomials can be computed in $(t \log |\mathbb{F}|)^{O(1)}$ time. Each solution to this system uniquely defines a co-dimension 2 subspace on which $NonLin(g)_{|_{V_i}}$ vanishes. By discussion above, each such system has at most $d^{O(1)}$ solutions. Therefore, using Lemma 2, for each i we can compute all solutions in $(d \log |\mathbb{F}|)^{O(1)}$ time, giving us sets $\mathcal{S}_i$ that contain tuples $(x_1 - y_3x_3 - y_4x_4 - y_ix_i, x_2 - z_3x_3 - z_4x_4 - z_ix_i)$ representing co-dimension 2 subspaces $\{x_1 - y_3x_3 - y_4x_4 - y_ix_i = 0, x_2 - z_3x_3 - z_4x_4 - z_ix_i = 0\}$. We show in the merge stage that it is enough to only compute subspaces of this kind.

4. *Merge(Steps 4-5):* In this stage, we merge the elements from different $\mathcal{S}_i$ computed in the previous stage. As explained in the *Transform* stage above, all co-dimension 2 spaces where $NonLin(g)$ vanishes have the form:

$$W = \{x_1 - \ell_1(x_3, \ldots, x_n) = 0, x_2 - \ell_2(x_3, \ldots, x_n) = 0\}.$$

   Also from discussion in the *Restrict* stage above, we know that, for any such W, $NonLin(g)_{|_{V_i}}$ vanishes on $\{x_1 - \ell_1(x_3, \ldots, x_n)_{|_{V_i}} = 0, x_2 - \ell_2(x_3, \ldots, x_n)_{|_{V_i}} = 0\}$ and so tuple corresponding to it has been computed in the previous stage. Hence the plan is to merge these tuples for $i \in [5, n]$ to get tuple of linear forms representing W. However, simply merging them is not possible due to some degenerate cases and so we apply another random transformation. We

use a random matrix $\mathbf{B} = (B_{i,j})$ defined as:

$$B_{i,j} = \begin{cases} 1 & i = j \\ uniform(\mathbb{F}) & (i,j) \in [5,n] \times [3,4] \\ 0 & otherwise \end{cases}$$

where $uniform(\mathbb{F})$ is sampled uniformly randomly from $\mathbb{F}$. It's easy to see that $\mathbf{B}$ is invertible and so is the transformation $\mathbf{x} \mapsto \mathbf{Bx}$. For each set $\mathcal{S}_i$, this transformation is applied to all tuples by applying it to linear forms in the tuple. Note that applying this transformation does not affect the coefficients of x_1, x_2 and therefore the form we mention is maintained. Since size of $\mathcal{S}_i$ is bounded by $3d^7$ as argued in the previous stage, for each $\mathcal{S}_i$ the transformation takes $(d \log |\mathbb{F}|)^{O(1)}$ time. Now we can show that the merge is possible for any co-dimension 2 subspace W that we considered. Consider a tuple in $\mathcal{S}_5$ that looks like $(x_1 - \alpha x_3 - \beta x_4 - \gamma x_5, x_2 - \delta x_3 - \theta x_4 - \psi x_5)$. We iterate through $\mathcal{S}_i$[12], $i \in [6,n]$ and look for tuples of linear forms which match with the above tuple in coefficients of x_1, x_2, x_3, x_4 and are therefore of the kind $(x_1 - \alpha x_3 - \beta x_4 - \kappa x_5, x_2 - \delta x_3 - \theta x_4 - \omega x_5)$. Using our discussion above, there is at least one such tuple in every $\mathcal{S}_i$[13]. By Fact 2, there can be at most one such tuple. Therefore we will be able to merge these tuples and create a tuple of linear forms, which after applying the inverse transformation $\mathbf{x} \mapsto \mathbf{B^{-1}x}$ represents W. All $\mathcal{S}_i$ had size $\leq 3d^7$ and therefore the merged set can have at most $3nd^7$ tuples. Due to the efficient merge process, we take $(nd \log |\mathbb{F}|)^{O(1)}$ time in this stage.

5. *Validate(Step 6):* As discussed above, the previous stage gives us a set (of tuples of linear forms) of size $\leq 3nd^7$. Also by the discussion above, tuples representing all co-dimension 2 subspaces where $NonLin(g)$ vanishes will be in this set. However we might have added more tuples and so we complete the algorithm by pruning this set and throwing away tuples for which $NonLin(g)$ does not vanish on the subspace they represent. This is easily done by iterating through all the tuples (ℓ_1, ℓ_2) and simulating black-box for $NonLin(g)_{|_{\{\ell_1=0,\ell_2=0\}}}$ and then using efficient randomized black-box polynomial identity test in Lemma 3 to check if this restricted polynomial is identically 0. The black-box can be easily simulated in $(n \log |\mathbb{F}|)^{O(1)}$ time. We have computed $\mathcal{S}(g)$ and by applying inverse transformation $\mathbf{x} \mapsto \mathbf{A^{-1}x}$ to each tuple in $\mathcal{S}(g)$, we get $\mathcal{S}(f)$. Clearly, these operations take $(nd \log |\mathbb{F}|)^{O(1)}$ time.

6 Ordinary lines span large space: Proof of Lemma 1

In this section we present our proof of Lemma 1. The proof is immediately implied by Lemma 14 which is itself proved using Lemma 15. Recall definition of set of ordinary lines(Definition 8).

Lemma 14. *Let $\mathcal{S} \subset \mathbb{F}^n$ be a proper set (Definition 7) and $\mathcal{T} \subset \mathbb{F}^n$ be any linearly independent set of size $\log |\mathcal{S}| + 2$. Then, the following holds.*

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

[12] after the transformation

[13] corresponding to restrictions of linear forms representing W after transformation

Proof of Theorem 1 using Lemma 14: 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 Theorem 1. So we are left with proving Lemma 14.

Proof of Lemma 14 Let V be the vector space $\sum_{t \in \mathcal{T}} \sum_{W \in \mathcal{O}(t,\mathcal{S})} W$. We define set $\mathcal{S}' = \mathcal{S} \setminus V$. Clearly $\mathcal{S}'$ 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. $sp\{s,t\} \cap V \not\subset sp\{s\} \cup sp\{t\}$

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 15 and complete the proof. We will prove Lemma 15 after the current proof.

Lemma 15. *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 8) from t into $\mathcal{S}$. Then $|\mathcal{T}| \leq \log|\mathcal{S}| + 1$.*

Using Lemma 15 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 15: 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 clearly 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[14] 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. □

[14] "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

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 15, 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.

[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, page 54–63, New York, NY, USA, 1991. 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.

[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.

[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 $T_1 + T_2$. Clearly $\ell \nmid T_1, T_2$ since $gcd(T_1, T_2) = 1$. So on restriction to $\{\ell = 0\}$, we get that $T_1|_{\{\ell=0\}} = -T_2|_{\{\ell=0\}} \neq 0$. Both sides are non-zero products of linear forms(in one less variable) and so by unique factorization we can match factors(upto scalar multiplication). This clearly 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 $y_1 \mid T_1$. Using Claim 1 we know that
$$0 \neq T_2|_{\{y_1=0\}} = (T_1 + T_2)|_{\{y_1=0\}} = NonLin(f)|_{\{y_1=0\}}$$
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 $T_2|_{\{y_1=0\}}$ 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 $NonLin(f)_{\{y_1=0,y_2=0\}}$ and $NonLin(f)_{\{y_1=0,y_3=0\}}$ are both zero, implying that second condition of Definition 6 is also satisfied. Therefore, some scalar multiple of $y_1 \in \mathcal{L}(f)$.

A.2 Proof of Claim 3

Recall definition of sets,

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

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

For all sets, we keep linear forms upto scalar multiplication and therefore treat them as proper sets (Definition 7). 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 $\mathcal{B} \subset \mathcal{L}_{factors}$ and let $\ell \in \mathcal{B}$. Restricting $T_1 + T_2$ to $\{\ell = 0\}$ gives $T_1|_{\{\ell=0\}} = -T_2|_{\{\ell=0\}} \neq 0$. So 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 $\mathcal{B}$, there are no ordinary lines from $\mathcal{B}$ into $\mathcal{L}$. So using Lemma 15 we get that $|\mathcal{B}| \leq \log |\mathcal{L}| + 1 = \log d + 2$, completing the proof.

2. $dim(sp(\mathcal{L}_{good})) \geq rank(f) - 2$ and $\mathcal{L}_{others} \leq 2$: Recall definition $V_i = \{\text{linear form } \ell : \ell \mid T_i\}$. We break the proof into two cases. Note that linear forms dividing T_1, T_2 clearly satisfy first condition of Definition 6. So whenever we are trying to show that they belong to $\mathcal{L}(f)$, 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 $T_1+T_2 = H \times NonLin(f)$. On restricting to $\{\ell = 0\}$, we see that $T_{2|_{\{\ell_1=0\}}} = H_{|_{\{\ell_1=0\}}} \times NonLin(f)_{|_{\{\ell_1=0\}}} \neq 0$. Dimension of span of linear factors of $T_{2|_{\{\ell_1=0\}}}$ is at least $\log d + 4$ by assumption in this case. By previous part, $dim(sp(\mathcal{L}_{factors})) \leq \log d + 2 \Rightarrow NonLin(f)_{|_{\{\ell_1=0\}}}$ has two independent linear factors. Using these we can satisfy second condition of Definition 6 for $\ell_1 \Rightarrow$ some scalar multiple of $\ell_1 \in \mathcal{L}(f)$. 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 $\mathcal{L}(f)$ (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 Lemma 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 $\{t = 0\}$ gives some non-zero product of linear factors, say H.

$$T_{1|_{\{t=0\}}} + T_2|_{\{t=0\}} - H = 0$$

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 1.2 in [SS13]). After cancelling common linear forms from the three gates $T_{1|_{\{t=0\}}}, T_{2|_{\{t=0\}}}, H$, we have a simple and minimal (Definition 1.2 in [SS13]), identically zero $\Sigma\Pi\Sigma(3, n, d, \mathbb{F})$ circuit. It's easy to see that 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 13

Let $T_i = \prod_{j=1}^{m} \ell_{i,j}$ where $\ell_{i,j}$ are linear forms. Consider $W = \{\ell_1 = 0, \ell_2 = 0\}$[15] such that

$$\prod_{j=1}^{m} \ell_{1,j|_W} = -\prod_{j=1}^{m} \ell_{2,j|_W} \neq 0.$$

Note that $\ell_{i,j|_W}$ 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 $\ell_{1,j|_W} = \beta_j \ell_{2,j|_W}$ for

[15] ℓ_1, ℓ_2 are linear forms in $\mathbb{F}[\mathbf{x}]$

some $0 \neq \beta_j \in \mathbb{F}$. This implies $U_j = sp\{\ell_{1,j}, \ell_{2,j}\}$[16] intersects $U = sp\{\ell_1, \ell_2\}$ non-trivially. Since $\ell_{i,j}|_W \neq 0$, we know that $U \neq U_j \Rightarrow U \cap U_j$ is 1 dimensional[17]. 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 $\ell \in U \Rightarrow W \subset \{\ell = 0\} = V$. 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, $U \cap U_i \neq U \cap U_j$:** It's easy to see that $U \cap U_i + U \cap U_j$ 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 Theorem 3, we know that

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

  $dim(U_i + U_j) \leq 4$, thus there exists U_k such that $U_k \nsubseteq U_i + U_j$. 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 $U_k \cap (U_i + U_j) = sp\{\ell\} \Rightarrow \ell \in U \Rightarrow W \subset \{\ell = 0\} = V$. There are $\leq d^6$ choices for U_i, U_j, U_k and so $\leq d^6$ possibilities for such V.

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

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

[17] since both U, U_j are 2 dimensional

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il