Reconstruction of Depth 3 Arithmetic Circuits with Top Fan-In 3

Shubhangi Saraf
Department of Mathematics & Department of Computer Science, University of Toronto, Canada

Devansh Shringi
Department of Computer Science, University of Toronto, Canada

Abstract

In this paper, we give the first subexponential (and in fact quasi-polynomial time) reconstruction algorithm for depth 3 circuits of top fan-in 3 ($\Sigma\Pi\Sigma(3)$ circuits) over the fields $\mathbb{R}$ and $\mathbb{C}$. Concretely, we show that given blackbox access to an n-variate polynomial f computed by a $\Sigma\Pi\Sigma(3)$ circuit of size s, there is a randomized algorithm that runs in time quasi-poly(n, s) and outputs a generalized $\Sigma\Pi\Sigma(3)$ circuit computing f. The size s includes the bit complexity of coefficients appearing in the circuit.

Depth 3 circuits of constant fan-in ($\Sigma\Pi\Sigma(k)$ circuits) and closely related models have been extensively studied in the context of polynomial identity testing (PIT). The study of PIT for these models led to an understanding of the structure of identically zero $\Sigma\Pi\Sigma(3)$ circuits and $\Sigma\Pi\Sigma(k)$ circuits using some very elegant connections to discrete geometry, specifically the Sylvester-Gallai Theorem, and colorful and high dimensional variants of them. Despite a lot of progress on PIT for $\Sigma\Pi\Sigma(k)$ circuits and more recently on PIT for depth 4 circuits of bounded top and bottom fan-in, reconstruction algorithms for $\Sigma\Pi\Sigma(k)$ circuits has proven to be extremely challenging.

In this paper, we build upon the structural results for identically zero $\Sigma\Pi\Sigma(3)$ circuits that bound their rank, and prove stronger structural properties of $\Sigma\Pi\Sigma(3)$ circuits (again using connections to discrete geometry). One such result is a bound on the number of codimension 3 subspaces on which a polynomial computed by an $\Sigma\Pi\Sigma(3)$ circuit can vanish on. Armed with the new structural results, we provide the first reconstruction algorithms for $\Sigma\Pi\Sigma(3)$ circuits over $\mathbb{R}$ and $\mathbb{C}$.

Our work extends the work of [Sinha, CCC 2016] who provided a reconstruction algorithm for $\Sigma\Pi\Sigma(2)$ circuits over $\mathbb{R}$ and $\mathbb{C}$ as well as the works of [Shpilka, STOC 2007] who provided a reconstruction algorithms for $\Sigma\Pi\Sigma(2)$ circuits in the setting of small finite fields, and [Karnin-Shpilka, CCC 2009] who provided reconstruction algorithms for $\Sigma\Pi\Sigma(k)$ circuits in the setting of small finite fields.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory; Theory of computation → Circuit complexity

Keywords and phrases arithmetic circuits, learning, reconstruction

Digital Object Identifier 10.4230/LIPIcs.CCC.2025.21

Related Version *Full Version*: `https://eccc.weizmann.ac.il/report/2025/008/` [41]

Funding *Shubhangi Saraf*: Research partially supported by an NSERC Discovery Grant and a McLean Award.

Acknowledgements The authors want to thank Nitin Saxena for helpful discussions.

1 Introduction

Arithmetic circuits are directed acyclic graphs (DAGs) that compute multivariate polynomials in a compact form, constructing these polynomials from variables using addition ($+$) and multiplication ($\times$) operations.

Reconstruction of arithmetic circuits is the following problem: given black-box or oracle access to a polynomial computed by a circuit C of size s from a certain class of circuits $\mathcal{C}$, design an efficient algorithm (either deterministic or randomized) to recover a circuit that computes the same polynomial as C. This question is the algebraic equivalent of exact learning in Boolean circuit complexity [2]. If it is additionally required that the output circuit belongs to the same class $\mathcal{C}$ as the input circuit, this process is referred to as *proper learning*.

Reconstruction of arithmetic circuits is a fundamental and challenging problem. In recent years, there has been a flurry of works on developing reconstruction algorithms for various interesting subclasses of arithmetic circuits [4, 32, 31, 15].

Thanks to the depth reduction results from [1, 33, 49, 19], we now know that even depth-3 and depth-4 circuits are quite expressive. Thus, efficient reconstruction algorithms even for depth three circuits would have significant implications for more general circuit models. Perhaps not surprisingly, we are quite far from achieving efficient reconstruction algorithms for general depth-3 circuits. However, in recent years there have been several works developing reconstruction algorithms for restricted yet interesting subclasses of depth 3 ($\Sigma\Pi\Sigma$) and depth 4 ($\Sigma\Pi\Sigma\Pi$) circuits [45, 25, 47, 48, 8, 40, 9, 21, 7].

A closely related problem is that of blackbox polynomial identity testing (PIT) which asks for the following. Given blackbox access to a polynomial f computed by some circuit C of size s from some class $\mathcal{C}$, the goal is to decide if f is identically zero. In other words, the goal is to compute an explicit hitting set for the class $\mathcal{C}$[1].

It is easy to see that obtaining deterministic reconstruction algorithms for a class of circuits $\mathcal{C}$ is at least as hard is derandomizing black-box PIT for $\mathcal{C}$. Even randomized reconstruction almost always requires some deep understanding of the structure of the underlying circuit class and in almost every case we know, it seems harder than derandomizing PIT for that class. Indeed for most circuits classes that have been studied, efficient PIT algorithms have been a precursor to understanding reconstruction algorithms for that class. Since reconstruction for depth 3 circuits of constant top fan-in ($\Sigma\Pi\Sigma(k)$ circuits) is the main focus of this paper, we first give some context by describing what is know about PIT for this class.

PIT for $\Sigma\Pi\Sigma(k)$ circuits

The recent breakthrough work of [34] gives the first subexponential deterministic blackbox PIT for $\Sigma\Pi\Sigma$ (and in fact any constant depth) circuits. If we want truly polynomial time blackbox derandomization, then we only know how do this for restricted classes of depth 3 circuits. When the the top fan-in of the output sum gate is a constant k, then this class is referred to as the class of $\Sigma\Pi\Sigma(k)$ circuits. $\Sigma\Pi\Sigma(k)$ circuits have received a great deal of attention in the context of blackbox PIT, and there has been a large body of beautiful works eventually showing polynomial time blackbox PIT algorithms for $\Sigma\Pi\Sigma(k)$ circuits [14, 24, 27, 43, 42]. A running theme through several of these works is to show that identically zero $\Sigma\Pi\Sigma(k)$ circuits have some very interesting structure; they must be *low rank*. Along the way some very elegant connections to discrete geometry, specifically the Sylvester-Gallai Theorem, and colorful and high dimensional variants of them were developed and used. In the last few years, there have been several exciting works trying obtaining similar results for interesting subclasses of depth-4 circuits, in particular for $\Sigma^k\Pi\Sigma\Pi^r$ which are depth 4 circuits with bounded top and bottom fan-in. A sequence of results [46, 38, 37, 39, 17] developed a beautiful theory of Sylvester-Gallai type configurations for quadratic polynomials

[1] With randomness, this problem is easy using the Schwartz-Zippel Lemma [44, 50]

and was able to obtain a polynomial time deterministic PIT result for $\Sigma^3\Pi\Sigma\Pi^2$ (think of $\Sigma\Pi\Sigma(3)$ circuits but with product of quadratics computed at the second layer of gates instead of a product of linear forms). Extending this to larger k and r is a very interesting direction and partial results in this direction have been obtained in [35, 16, 36]. Using completely different techniques, a remarkable work by [12] gives quasipolynomial blackbox PIT for $\Sigma^k\Pi\Sigma\Pi^r$ circuits for any constants k and r.

Reconstruction for $\Sigma\Pi\Sigma(k)$ circuits

Despite all this progress for PIT, far less is known for reconstruction of $\Sigma\Pi\Sigma(k)$ circuits even when the algorithms are allowed to be randomized.

For now let us assume the underlying field has characteristic 0. In particular let us assume the coefficients lie in $\mathbb{R}$ or $\mathbb{C}$. Without additional restrictions like multilinearity and set-multilinearity, we essentially only how to do efficient reconstruction of $\Sigma\Pi\Sigma(k)$ circuits over infinite fields like $\mathbb{R}$ and $\mathbb{C}$ when $k = 2$ [47]! The result in [47] gives a randomized $\text{poly}(n, d)$ time reconstruction algorithm for n variate, degree d polynomials represented by a $\Sigma\Pi\Sigma(2)$ circuit over $\mathbb{R}$ or $\mathbb{C}$.

Note that when $k = 2$, then derandomizing PIT is easy, and it only starts becoming challenging once $k \geq 3$. However reconstruction is much more challenging and despite all the progress on the PIT front, it took a really long time to get efficient reconstruction for $\Sigma\Pi\Sigma(2)$ circuits. The proof in [47] is quite sophisticated and uses some really beautiful connections to discrete geometry, in particular to the robust Sylvester-Gallai theorems (inspired by the theory developed for PIT of $\Sigma\Pi\Sigma(k)$ circuits). Thus even for the seemingly simple case of $k = 2$, reconstruction can be fairly complex. Already when $k = 3$, the techniques of the above work break down and nothing nontrivial was known for reconstructing $\Sigma\Pi\Sigma(3)$ circuits over $\mathbb{R}$ or $\mathbb{C}$.

In the setting of finite fields, there are some additional very interesting results for reconstruction of $\Sigma\Pi\Sigma(k)$ circuits. Over small (only polynomially large) finite fields, the first (and very nontrivial) reconstruction algorithm for $\Sigma\Pi\Sigma(2)$ circuits was given in [45]. This algorithm run time has a quasipolynomial dependence on $|\mathbb{F}|$ (it crucially needs to iterate over all field constants) and is therefore only efficient for small fields. This work was extended to $\Sigma\Pi\Sigma(k)$ circuits for any constant k in [25], but again it is only efficient for small finite fields due to the quasipolynomial dependence on $|\mathbb{F}|$. When the input is an n-variate, degree d polynomial computed by a size s circuit, both the above algorithms run in $\text{quasi-poly}(n, d, |\mathbb{F}|, s)$ time. In the setting of $k = 2$, only very recently it was shown in [48] that there is a polynomial time reconstruction algorithms with a run time that has a polynomial dependence on $\log |\mathbb{F}|$. In the further restricted setting where we have additional constraints of multilinearity or set-multilinearity[2], there has been a large body of work on reconstruction algorithms for $\Sigma\Pi\Sigma(k)$ circuits [4, 45, 25, 8, 40, 9].

Thus to summarize, over large fields, or infinite fields such as $\mathbb{R}$ or $\mathbb{C}$, we knew no nontrivial reconstruction algorithms for general $\Sigma\Pi\Sigma(k)$ circuits even for $k = 3$. The main result of this paper is to give the first efficient reconstruction algorithm for $\Sigma\Pi\Sigma(3)$ circuits over infinite fields such as $\mathbb{R}$ and $\mathbb{C}$.

Our result (informal). *Given blackbox access to an n-variate degree d polynomial f over $\mathbb{R}$ or $\mathbb{C}$, computed by a $\Sigma\Pi\Sigma(3)$ circuit, there is a randomized quasi-poly(n, d, s) time reconstruction algorithm for f, where s is the maximum bit complexity of any constant appearing in the circuit.*

[2] This setting captures tensor reconstruction for constant rank tensors

Before we state our results more formally, we first introduce some definitions and notions related to $\Sigma\Pi\Sigma(k)$ circuits.

Some definitions related to $\Sigma\Pi\Sigma(k)$ circuits

The model of depth-3 arithmetic circuits with top fan-in k, which we refer as $\Sigma\Pi\Sigma(k)$ circuits, has three layers of alternating Σ and Π gates and computes a polynomial of the form

$$C(\bar{x}) = \sum_{i=1}^{k} T_i(\bar{x}) = \sum_{i=1}^{k} \prod_{j=1}^{d_i} l_{ij}(\bar{x})$$

where the $l_{ij}(\bar{x})$-s are linear polynomials.

We will in fact assume that the circuits are homogeneous and all the d_i's are actually the same. This is because for the purpose of reconstruction and PIT, one can easily reduce to the homogeneous setting.

We say that the circuit is simple if $\gcd\{T_i | i \in [k]\} = 1$ and minimal if for all proper subsets $S \subset [k]$, $\sum_{i \in S} T_i \neq 0$. We define $\gcd(C) = \gcd(T_1, \ldots, T_k)$. The simplification of C, denoted by $\mathrm{sim}(C)$, is defined as $C/\gcd(C)$. We define the rank of a circuit ($\mathrm{rank}(C)$) as the dimension of the space spanned by all the linear forms in the circuit $\dim(\mathrm{span}(\{l_{i,j} : i \in [k], j \in [d_i]\}))$. We will often be concerned with $\mathrm{rank}(\mathrm{sim}(C))$.

A generalized depth 3 circuit $\Sigma\Pi\Sigma(k, d, r)$ is of the form

$$C = \sum_{i=1}^{k} \left(\prod_{j=1}^{d_i} l_{ij} \cdot h_i(\bar{l}_{i1}, \ldots, \bar{l}_{ir}) \right)$$

where l_{ij},$\bar{l}_{ik}$ are linear forms in $\mathbb{F}[x_1, \ldots, x_n]$ and $d = max_i(d_i + deg(h_i))$. Notice that when r is small (say constant or $O(\log d)$, the representation looks like a $\Sigma\Pi\Sigma(k)$ circuit where every product gate is further multiplied by a polynomial in few (r) linear forms.

Our techniques: Rank bounds and connections to discrete geometry

As mentioned previously, several of the blackbox PIT results for $\Sigma\Pi\Sigma(k)$ circuits and related models follows from some insight into the structure of identically zero $\Sigma\Pi\Sigma(k)$ circuits. One such remarkable structural result which is also a central ingredient in our proof is that identically zero simple and minimal $\Sigma\Pi\Sigma(k)$ circuits over $\mathbb{R}$ or $\mathbb{C}$ must be of only constant rank [27, 42, 43]. In particular, if a simple and minimal $\Sigma\Pi\Sigma(3)$ circuit computes the identically zero polynomial, then the set of all linear forms appearing at any gates in the circuit can only span a constant dimensional space.

In this paper, we develop a deeper understanding of some structural properties of $\Sigma\Pi\Sigma(3)$ circuits. Since we need to learn the linear forms in a given underlying $\Sigma\Pi\Sigma(3)$ circuit (not just determine whether the polynomial is zero or nonzero), thus we need also to understand and develop structural properties of *non-zero* $\Sigma\Pi\Sigma(3)$ circuits. One example of such a structural result we show is that if a polynomial f is computed by an $\Sigma\Pi\Sigma(3)$ circuit (which has some mild non-degeneracy property) then the number of codimension 3 subspaces on which it can vanish is polynomially bounded (see Lemma 11).

1.1 Our Results

In this paper, we give the first subexponential time (and in fact quasipolynomial time) algorithm for reconstructing $\Sigma\Pi\Sigma(3)$ circuits over $\mathbb{R}$ and $\mathbb{C}$. When the three multiplication gates in our circuit are sufficiently distant, i.e. when $\forall i, j \in [3], i \neq j, \mathrm{rank}(\mathrm{sim}(T_i + T_j)) \geq$

$c \log d$ for some absolute constant c, then our algorithm does "proper learning", i.e. its output is the unique $\Sigma\Pi\Sigma(3)$ circuit computing f. If this distance property does not hold, then our algorithm outputs a generalized depth 3 circuit of top fan-in at most 2 with parameters $\Sigma\Pi\Sigma(2, d, c \log d)$. We state our main theorem below. The running time in the statement is suppressing a $\text{poly}(s)$ dependence on the max bit complexity s of any constant appearing in the circuit C.

▶ **Theorem 1.** *Let $\mathbb{F}$ be a field that is $\mathbb{R}$ or $\mathbb{C}$. Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ be a degree d polynomial computed by $\Sigma\Pi\Sigma(3)$ circuit of the form $C = T_1 + T_2 + T_3$. There exist an absolute constant $c > 0$ such that the following holds. There is a randomized algorithm that runs in $(nd)^{O(\log d)}$ time, makes blackbox queries to f, and with probability $1 - o(1)$ does the following:*

1. *If $\forall i, j \in [3], i \neq j, \text{rank}(\text{sim}(T_i + T_j)) \geq c \log d$ then it outputs a $\Sigma\Pi\Sigma(3)$ circuit computing f.*
2. *If $\exists i, j \in [3], i \neq j$, such that $\text{rank}(\text{sim}(T_i + T_j)) < c \log d$ then it outputs a $\Sigma\Pi\Sigma(2, d, c \log d)$ generalized depth 3 circuit computing f.*

▶ Remark 2 (Dependence on bit complexity). If s is the maximum bit complexity of any coefficient appearing in C, then our algorithm run time also depends polynomially on s. In the statement of the above theorem and later in the paper, we have suppressed the $\text{poly}(s)$ dependence in the running time for clarity of exposition.

▶ Remark 3 (Proper vs improper learning). Note that our algorithm is a proper learning algorithm only when every pair of multiplication gates had enough "distance". Otherwise, the output came from the model of generalized depth 3 circuits. All prior works on reconstruction of $\Sigma\Pi\Sigma(2)$ circuits and $\Sigma\Pi\Sigma(k)$ circuits ([45, 25, 47, 48]) also had a similar kind of output.

1.2 Other related works

As the case of general $\Sigma\Pi\Sigma(k)$ is considered hard, several restrictions of the model have been considered for reconstruction. Some well-studied models are powering depth-3 circuits $\Sigma \wedge \Sigma(k)$, multilinear $\Sigma\Pi\Sigma(k)$, and set-multilinear $\Sigma\Pi\Sigma(k)$ circuits. The restrictions of powering circuits $\Sigma \wedge \Sigma(k)$ and set-multilinear $\Sigma\Pi\Sigma(k)$ have received special attention due to their connections with finding symmetric tensor decomposition and tensor decomposition problems [4, 45, 25, 8, 40, 9].

For the model of depth 2 circuits, $\Sigma\Pi$, the problem of reconstruction is equivalent to sparse multivariate interpolation for which we have a polynomial time algorithm in [5]. The work of [7] studied multilinear depth-4 circuits with bounded top fan-in ($\Sigma\Pi\Sigma\Pi(k)$ circuits), and gave deterministic reconstruction algorithms which ran in $\text{poly}(n, d, |\mathbb{F}|)$ time. The running time is however still at least $\text{poly}(|\mathbb{F}|)$, and hence it does not work over large/infinite fields. Note that when the top fan-in is 2, i.e. for $\Sigma\Pi\Sigma\Pi(2)$ circuits, we do know such efficient polynomial-time reconstruction algorithms by the work of [21]. Read-once oblivious branching programs (ROABPs) are another model that have been well studied in the context of reconstruction. In [31], the authors presented a randomized reconstruction (proper learning) algorithm for ran in time $\text{poly}(n, d, s)$. This was derandomized in [15], giving a deterministic quasi-$\text{poly}(n, d, s)$ reconstruction algorithm.

Recently, there have also been several works studying *average case* learning algorithms for arithmetic circuits. In [30], the authors give a $\text{poly}(n, d, k)$ time reconstruction algorithm for non-degenerate homogeneous depth three circuits $\Sigma\Pi\Sigma(k)$ circuits. A $\text{poly}(n, d, s)$ learning algorithm for generalized depth three circuits in the non-degenerate case is presented in [6]. Reconstruction algorithms for other constant depth circuits in the non-degenerate case have also been obtained in [20, 22, 26, 29, 18].

Outline of the paper

In the rest of the paper, we will present an overview of the proof in Section 2 and the detailed proof of a structure theorem about the vanishing codimension 2 and 3 spaces of a polynomial computed by a $\Sigma\Pi\Sigma(3)$ circuit, which we crucially use in our reconstruction algorithm in Section 4. The rest of the details and proofs can be found in the full version [41] of the paper.

2 Proof Overview

Let f be a polynomial that has a $\Sigma\Pi\Sigma(3)$ representation and let

$$C = T_1 + T_2 + T_3$$

be a $\Sigma\Pi\Sigma(3)$ circuit computing f. Thus each T_i is a product of linear forms, and as we describe in the preliminaries, with some simple preprocessing, we can assume that the circuit and all gates within it are homogeneous. In general, the gates T_i might have nontrivial gcd, which has to be dealt with, but for the purpose of the proof overview, let us assume that $\gcd(T_1, T_2, T_3) = 1$. Note that we cannot easily reduce to the case of $\gcd(T_1, T_2, T_3) = 1$ by factoring and dividing out the linear factors since there might be linear factors which do not divide the gcd, and division by those factors might not preserve the property of the polynomial being representable by a $\Sigma\Pi\Sigma(3)$ circuit.

In order to reconstruct the circuit C, we need to somehow try and learn the linear forms appearing in C. What we have is (randomized) access to a blackbox computing f.

Notice that if l_1 is a linear form dividing T_1, l_2 is a linear form dividing T_2 and l_3 is a linear form dividing T_3 then if we go modulo l_1, l_2 and l_3, then the polynomial f vanishes identically. In other words, for any input where l_1, l_2 and l_3 evaluate to 0, f evaluates to 0 as well.

Let $\mathcal{S}_3(f)$ to be the set of all codimension 3 subspaces of $\mathbb{F}^n$ over which f vanishes. Then if we could somehow "learn" all the spaces in this set, then one of them would correspond to $\mathbb{V}(l_1, l_2, l_3)$ i.e. the codimension 3 space where l_1, l_2, l_3 vanish (or evaluate to 0). Thus, a starting challenge for us is to show that the set $\mathcal{S}_3(f)$ can be learned. It turns out that the set can be infinite. Suppose $\mathbb{V}(l, l')$ is some codimension 2 space on which f vanishes. Then any codimension 3 space contained within $\mathbb{V}(l, l')$ will also be a space on which f vanishes and hence be contained in $\mathcal{S}_3(f)$. This makes the set unwieldy to deal with and hence we modify the definition.

Let $\mathcal{S}_3(f)$ to be the set of all codimension 3 subspaces of $\mathbb{F}^n$ over which f vanishes, such that it is not contained within any codimension 1 or 2 space on which f vanishes. One of our significant structural results is to show that other than in certain degenerate settings, $\mathcal{S}_3(f)$ is finite and in fact has at most $\text{poly}(d)$ distinct elements. This is indeed the first major ingredient of our proof and we prove this in Section 4. We also show that $\mathcal{S}_2(f)$, which is defined similarly with codimension 2 spaces, is finite, and has at most $\text{poly}(d)$ distinct elements. This is the starting point of our analysis.

Once we prove that $\mathcal{S}_3(f)$ and $\mathcal{S}_2(f)$ are finite and polynomially bounded (other than in degenerate settings), the next thing we show is how to actually compute $\mathcal{S}_3(f)$ and $\mathcal{S}_2(f)$. Once we have these sets, we then use them to learn the linear forms appearing in C.

Our reconstruction algorithm for $\Sigma\Pi\Sigma(3)$ circuits over $\mathbb{R}$ or $\mathbb{C}$ (and proof of correctness of the algorithm) follows from the following broad outline

1. Obtain an upper bound on the number of a codimension 2 subspaces ($\mathcal{S}_2(f)$) and codimension 3 subspaces ($\mathcal{S}_3(f)$) on which f vanishes (other than in some degenerate settings). This is shown in Section 4.

2. Algorithmically compute $\mathcal{S}_2(f)$ and $\mathcal{S}_3(f)$. This is shown in Section 5 of the full version [41]. At a high level, we consider projections of the circuit to constantly many variables, compute $\mathcal{S}_2$ and $\mathcal{S}_3$ for the constant variate polynomials by solving a suitable system of polynomial equations for each projection, and then "glue" or "lift" the solutions to a global solution over the entire original space. The ideas are inspired by the algorithms in [48]. However there are some additional nontrivial challenges that arise in our setting.
3. Use $\mathcal{S}_2(f)$ and $\mathcal{S}_3(f)$ to form a list of linear forms (which we call $\mathcal{L}_{cand}$) such that several of these linear forms actually divide one of the gates of the circuit. We will do this in settings where $\mathcal{S}_2(f)$ and $\mathcal{S}_3(f)$ can be computed as well as in the degenerate cases where they cannot, and for this we will use the structure of the degeneracy. This is shown in Section 6 of the full version [41] and is the most technically complex part of the proof.
4. Reconstruct the entirety of the circuit using the few linear forms learned in the previous part. This is achieved by going modulo the linear forms and learning the projected circuit of top fan-in at most 2, and then gluing the projections to recover the original circuit. This part uses several ideas from the reconstruction algorithms of Shpilka [45] and Karnin-Shpilka [25].

We will discuss each of the four parts in some more detail in the following subsections below.

2.1 Overview: Upper bounding the size of $\mathcal{S}_2(f)$ and $\mathcal{S}_3(f)$

For $\mathcal{S}_1(f)$, which is defined to be the number of codimension 1 spaces over which f vanishes, bounding its size is easy since, each member of $\mathcal{S}_1(f)$ corresponds to a linear factor of f, and there can be at most d of those.

We now give a flavor of what goes into bounding $\mathcal{S}_2(f)$. We will only be able to bound $\mathcal{S}_2(f)$ when the circuit $C = T_1 + T_2 + T_3$ is such that $\text{rank}(\text{sim}(C)) > c_2$ for some absolutely constant c_2 which depends on the rank bound for identically zero $\Sigma\Pi\Sigma(3)$ circuits.

By assumption, let $\text{rank}(\text{sim}(C)) > c_2$. For any $\mathbb{V}(l, l')$ which is an $\mathcal{S}_2(f)$ space, let $C' = C \mod \langle l_1, l_2\rangle$. The C' computes a polynomial that is identically zero. Thus, by rank bounds for identically zero $\Sigma\Pi\Sigma(3)$ circuits (see Theorem 6), it holds that $\text{rank}(\text{sim}(C')) < \mathcal{R}(3)$ which is an absolute constant much smaller than c_2. Suppose we are only trying to bound the number of those $\mathbb{V}(l, l') \in \mathcal{S}_2(f)$ over which none of the individual T_i vanish (the other case is not so hard to bound) then the fact that $\text{rank}(\text{sim}(C \mod \langle l_1, l_2\rangle))$ is much smaller than $\text{rank}(\text{sim}(C))$ means that many triples of linear forms coming from distinct gates must "collapse" and become identical when we go mod $\langle l, l'\rangle$. Thus they will no longer contribute to the rank of $\text{sim}(C')$ as they will be in the gcd. (If there is no movement to the gcd, then the overall rank can reduce by at most two). Now if l_1 which divides T_1, l_2 which divides T_2 and l_3 which divides T_3 are three linearly independent linear forms that become identical mod $\langle l, l'\rangle$ then it must hold that $\text{span}(l, l') \subseteq \text{span}(l_1, l_2, l_3)$. Suppose this happens for another triple l'_1, l'_2, l'_3 which is linearly independent and such that $\text{span}(l_1, l_2, l_3)$ is distinct from $\text{span}(l'_1, l'_2, l'_3)$. Then, since $\text{span}(l, l')$ must belong to the spans of both triples, it must hold that $\text{span}(l, l')$ is in fact equal to $\text{span}(l_1, l_2, l_3) \cap \text{span}(l'_1, l'_2, l'_3)$. Thus $l_1, l_2, l_3, l'_1, l'_2, l'_3$ jointly determine $\text{span}(l, l')$ and hence, given the circuit C, there are only $O(d^6)$ choices for $\text{span}(l, l')$.

Note that we haven't covered all cases. It could be that the triples which collapse and move to the gcd are not linearly independent and they only span 2-dimensional spaces. This case needs to be handled separately. Also the case where one of the gates (say T_1) identically vanishes over $\mathbb{V}(l, l')$ has to be dealt with. In each of these cases we are able to bound the number of such spaces in $\mathcal{S}_2(f)$ and thus we get a polynomial bound on $|\mathcal{S}_2(f)|$.

Bounding $|\mathcal{S}_3(f)|$ is significantly more challenging, and the analysis breaks down into a larger number of cases. Also, we are not able to bound the size of $\mathcal{S}_3(f)$ whenever $\text{rank}(\text{sim}(C))$ is large, unlike the bounding of $|\mathcal{S}_2(f)|$. Notice that even if $\text{rank}(\text{sim}(C))$ is large, there could exists a linear form l such that $\text{rank}(\text{sim}(C \mod l))$ is quite small. In this case perhaps $f \mod l$ which is computed by $C \mod l$ can vanish on a large (maybe infinite) set of codimension 2 spaces. Then along with l, this will give us a large (possibly unbounded) set of codimension 3 spaces that f vanishes on in $\mathcal{S}_3(f)$. Thus we are only able to bound $\mathcal{S}_3(f)$ when we have the added condition that there is no linear form l such that $\text{rank}(\text{sim}(C \mod l))$ is small. This is the non-degeneracy condition that we alluded to earlier.

For circuits C that start off with $\text{rank}(\text{sim}(C))$ being large but such that there exists a linear form l such that $\text{rank}(\text{sim}(C \mod l))$ is small and $C \mod l \neq 0$, we call these circuits "special form" circuits. We are unable to bound the number of $\mathcal{S}_3$ spaces for polynomials computed by such circuits, but nevertheless, we show that such circuits have other additional nice properties that we will eventually exploit to learn them.

Comparison to [48]

In [48], the authors develop a similar structural result where they bound the number of codimension 2 vanishing spaces for the non-linear part for a $\Sigma\Pi\Sigma(2)$ circuit. The structure of $\Sigma\Pi\Sigma(2)$ circuits is simpler. Note for instance that derandomizing PIT for $\Sigma\Pi\Sigma(2)$ circuits is trivial whereas derandomizing PIT for $\Sigma\Pi\Sigma(3)$ circuits is more challenging and is closely related to the rank bound. Indeed (unlike in [48]) we need to crucially use the rank bound to bound vanishing spaces, and the analysis is more intricate.

2.2 Overview: Algorithmically computing $\mathcal{S}_2(f)$ and $\mathcal{S}_3(f)$

We will only show how to compute $\mathcal{S}_2(f)$ and $\mathcal{S}_3(f)$ in the settings where we have proved that their size is polynomially bounded.

Our high level strategy is inspired by the algorithms in [48]. However there are some additional nontrivial challenges that arise in our setting.

For the purpose of the proof overview, we focus on the case of computing $\mathcal{S}_3(f)$, and assume that we know how to compute $\mathcal{S}_2(f)$.

Constant variate case

We first show that if the underlying polynomial is constant variate, then this can be done. We do this by setting up a suitable system of polynomial equations. Setting up equations to find all l_1, l_2, l_3 (the variables are the coefficients of the monomials in l_1, l_2, l_3) such that f vanishes over the codimension 3 space $\mathbb{V}(l_1, l_2, l_3)$ is fairly straightforward. However this might have infinitely many solutions unless we ensure that $\mathbb{V}(l_1, l_2, l_3)$ does not lie within any codimension 2 space on which f vanishes. For this we first compute $\mathcal{S}_1(f)$ and $\mathcal{S}_2(f)$ and for each element in these sets we will require that some certain matrix has large rank. By introducing additional variables (just constantly many) we show how to capture these "large rank" constraints as well using polynomial equations. Thus overall we have polynomially many equations of polynomial degree, but in only constantly many variables. This can be solved (over $\mathbb{R}$ or $\mathbb{C}$) to recover all solutions.

Large variate case

In case f is over a large number of variables, we consider several distinct projections of f to the constant variate case. We learn the $\mathcal{S}_3$ spaces for each of the projections and then glue them together to recover the $\mathcal{S}_3$ spaces for f. Before performing the projections, we first apply a random linear invertible transformation to the variables of f to ensure that the projection has nice properties (such as being able to apply Hilbert irreducibility). For our proof to go through, we will require that the projections of f do not contain new linear factors (which would correspond to new $\mathcal{S}_1$ spaces not arising from projections of original $\mathcal{S}_1$ spaces) and this is easy to obtain using Hilbert irreducibility. However, crucially we will also require that the projections don't generate new $\mathcal{S}_2$ spaces. This is important since if new $\mathcal{S}_2$ spaces were generated, then potentially one could lose out on $\mathcal{S}_3$ spaces when we take a projection (since we are not able to learn codimension 3 spaces contained within an $\mathcal{S}_2$ space). Proving that new $\mathcal{S}_2$ spaces (i.e. not just the ones that are projections of $\mathcal{S}_2$ spaces of f) are not generated does not follow immediately from Hilbert irreducibility, and indeed we are not able to prove this fact for general polynomials (though perhaps it might be true in general as well). Our proof crucially use the fact that f can be represented as a high rank $\Sigma\Pi\Sigma(3)$ circuit. Indeed a crucial ingredient in our proof (of the fact that no new $\mathcal{S}_2$ spaces are generated in the projections) is the upper bound on the number of $\mathcal{S}_3$ spaces of f.

Once we can prove that no new $\mathcal{S}_2$ spaces are generated in the projections, then it is not hard to show that every space in $\mathcal{S}_3(f)$ gets projected to a distinct space in $\mathcal{S}_3(g_i)$ for each projected polynomial g_i. Now g_i for each i is a constant variate polynomial and we can show that $\mathcal{S}_3(g_i)$ can be computed. Given the structure of how g_i are chosen, it is then not hard to see that the spaces in $\mathcal{S}_3(g_i)$ can be stitched across the different choices of g_i to recover $\mathcal{S}_3(f)$.

Comparison to [48]

A similar algorithm (Algorithm 7) appeared in [48] for computing the set of codimension 2 vanishing spaces for $\Sigma\Pi\Sigma(2)$ circuits. Our algorithm for learning the set of codimension 3 vanishing spaces is inspired by this work, but there is one crucial difficulty/difference. In [48], when $\mathcal{S}_2(f)$ is being learnt, one needs to discard those vanishing spaces that are contained in an $\mathcal{S}_1$ space. This can be easily achieved by just dividing out the linear factors. This process needs to be modified when learning $\mathcal{S}_3(f)$ since there is no way of just "factoring out" the spaces in $\mathcal{S}_2(f)$. Instead, we have to set up a modified system of polynomial equations that handles this. We also need to prove a structural result that ensures that after projecting the large variate case to the constant variate case, the bounds on the number of $\mathcal{S}_3$ spaces still holds (no linear form exists modulo which the rank of the simple part drops below c_2), and no new $\mathcal{S}_2$ spaces are generated. These issues did not arise in [48].

2.3 Overview: Using $\mathcal{S}_2(f)$ and $\mathcal{S}_3(f)$ to learn some linear forms appearing in C

Recall, f is a polynomial that has a $\Sigma\Pi\Sigma(3)$ representation and let

$$C = T_1 + T_2 + T_3$$

be the $\Sigma\Pi\Sigma(3)$ circuit computing f. Each T_i is product of linear forms and for now we are assuming that $\gcd(T_1, T_2, T_3) = 1$. We will show that we are able to learn $\Omega(\log d)$ independent linear forms from one of the gates in C.

If l_1 is a linear form dividing T_1, l_2 is a linear form dividing T_2 and l_3 is a linear form dividing T_3 such that $\dim(\text{span}(l_1, l_2, l_3)) = 3$, then $\mathbb{V}(l_1, l_2, l_3)$ will belong to $\mathcal{S}_3(f)$ unless it is contained within some space in $\mathcal{S}_2(f)$ or is some space in $\mathcal{S}_1(f)$. In such a case, we will say $\mathbb{V}(l_1, l_2, l_3)$ got "blocked" by an $\mathcal{S}_1$ or $\mathcal{S}_2$ space and hence did not get learned.

Now suppose that $\mathbb{V}(l_1, l_2, l_3)$ did not get blocked and hence lies in $\mathcal{S}_3(f)$. Thus we can use $\mathcal{S}_3(f)$ to learn $\text{span}(l_1, l_2, l_3)$. Suppose there exists l_3' dividing T_3 such that $\mathbb{V}(l_1, l_2, l_3')$ is some other distinct space in $\mathcal{S}_3(f)$. Thus we can also learn $\text{span}(l_1, l_2, l_3')$. The intersection of these two spaces allows us to learn $\text{span}(l_1, l_2)$. Just like we managed to learn $\text{span}(l_1, l_2)$, if we could also somehow learn $\text{span}(l_1, l_2')$ for some other linear form l_2' dividing T_2, then we could take the intersection of $\text{span}(l_1, l_2)$ and $\text{span}(l_1, l_2')$ to learn l_1 and hence learn one of the linear forms appearing in C !

Thus intersections of kernels of spaces in $\mathcal{S}_3(f)$ can be useful in learning linear forms in C. Can this strategy always be carried out to learn linear forms? Or could it be that we cannot learn any linear forms because $\mathcal{S}_3(f)$ is empty, since every codimension 3 space on which f vanishes got blocked by a space in $\mathcal{S}_1(f)$ or $\mathcal{S}_2(f)$?

It turns out that this part proved to be surprisingly challenging to show and is the technically most difficult and intricate part of the paper. Indeed we are not able to show that intersections of spaces in $\mathcal{S}_3(f)$ will suffice in learning linear forms. However, we do show that in most interesting cases, either intersections of kernels of $\mathcal{S}_3$ spaces, or intersections of kernels of $\mathcal{S}_2$ spaces will suffice. When these do not suffice, then we show that the underlying circuit has some other nice structure which can be exploited to learn the linear forms. We first prove some useful structural properties of $\mathcal{S}_1(f)$ and $\mathcal{S}_2(f)$.

In the rest of the proof overview, we will just try to give some high level ideas of the kinds of structural results we need to prove, and some of the algorithmic ideas that go into the reconstruction. It will be a considerable simplification of all the actual cases we need to consider and their analyses.

$\mathcal{S}_1(f)$ is essentially low dimensional

$\mathcal{S}_1(f)$ is in correspondence with the linear factors of f. The linear factors could either divide $\gcd(T_1 + T_2 + T_3)$ (which we assume is 1 for the proof overview) or it could divide $\text{sim}(T_1 + T_2 + T_3)$. We show that the set of linear factors dividing $\text{sim}(T_1 + T_2 + T_3)$ can span dimension at most $O(\log d)$. This uses lower bounds for 2-query locally decodable codes (similar such bounds also appeared in [14, 45, 25]).

Understanding the structure of $\mathcal{S}_2(f)$

If we could show that linear forms defining the kernels of spaces in $\mathcal{S}_2(f)$ are also low dimensional then that would be very convenient, since then if the gates in the circuit start off having many high rank linear forms, it would show that many of the spaces in $\mathcal{S}_3(f)$ remain unblocked, and then we can use them to learn linear forms.

However, this turns out to be not true and this causes the proof to become quite a bit more involved. Though we are not able to bound the dimension of $\mathcal{S}_2(f)$ as a whole, we still manage to prove some structural results that suffice for our purpose.

Consider any $\mathbb{V}(l, l') \in \mathcal{S}_2(f)$. Thus when we consider f modulo l and l', it is identically zero. There are two ways this can happen. Either each of the T_i's vanishes modulo l and l' (in other words, each T_i has a linear form dividing it that is in the span of l and l') or the T_i's don't *all* individually vanish, but still their sum vanishes.

We would like to partition the set $\mathcal{S}_2(f)$ based on the above two possibilities. We say $\mathbb{V}(l,l') \in \mathcal{S}_2(f)$ is in $\mathcal{S}_2^{sp}(f)$ if each T_i has a linear factor lying in $\text{span}(l,l')$ and we say it is in $\mathcal{S}_2^{reg}(f)$ otherwise. (We are cheating a bit here - our actual definitions of these two sets is a bit more subtle, but for intuition, this is good enough. In reality the set $\mathcal{S}_2^{reg}(f)$ is a bit larger. It could be that each T_i has a linear form in $\text{span}(l,l')$ but when we remove those linear forms (taking into account multiplicities) then the resulting circuit still vanishes when we go mod l and l'. In this case we add $\mathbb{V}(l,l')$ to $\mathcal{S}_2^{reg}(f)$).

The set $\mathcal{S}_2^{sp}(f)$ is actually a nice helpful set. It can be quite useful in learning linear forms that appear in the circuit. For l_1 dividing T_1, l_2 dividing T_2 and l_3 dividing T_3 suppose that $\mathbb{V}(l_1,l_2,l_3)$ did not belong to $\mathcal{S}_3(f)$ since it got blocked by a space in $\mathcal{S}_2(f)$. Then we show that if that was a space in $\mathcal{S}_2^{sp}(f)$, that is usually not a big problem, since the space in $\mathcal{S}_2^{sp}(f)$ can actually be used to learn the space $\mathbb{V}(l_1,l_2,l_3)$ unless one of l_1,l_2, or l_3 is in the kernel of the $\mathcal{S}_2^{sp}(f)$ space. If a large fraction $\mathbb{V}(l_1,l_2,l_3)$ spaces are blocked by $\mathcal{S}_2^{sp}(f)$ spaces whose kernel contains one of l_1,l_2, or l_3, then we learn these linear forms from the intersection of kernels of $\mathcal{S}_2^{sp}(f)$ spaces, so this case turns out not be be a problem either.

The bigger issue is when $\mathbb{V}(l_1,l_2,l_3)$ gets blocked by a space in $\mathcal{S}_2^{reg}(f)$. We show that this cannot happen *too often*. Though we cannot say that the union of kernels of spaces in $\mathcal{S}_2^{reg}(f)$ is low dimensional, we can say something close. Consider the maximum number of spaces, k, in $\mathcal{S}_2^{reg}(f)$ such that their kernels are completely linearly independent. In other words, the k kernels (that are each 2 dimensional) in total span a $2k$ dimensional space. We show that k is at most $O(\log d)$. This structure ends up being sufficient to show that $\mathcal{S}_2^{reg}(f)$ cannot just block all spaces of the form $\mathbb{V}(l_1,l_2,l_3)$ that we wanted to learn, assuming that each of the T_i's started off with enough linearly independent linear forms present in all of them.

Though our target is to learn $\Omega(\log d)$ independent linear forms appearing in one gate, in most cases it is sufficient to learn two independent linear forms (not in kernels of $\mathcal{S}_1$ spaces) from one gate. We can then reconstruct the circuit mod these linear forms as $\Sigma\Pi\Sigma(2)$ circuits using the reconstruction algorithm in [47] and use it to get projections of a high-rank gate, which we can then glue to obtain $\Omega(\log d)$ independent linear forms from one gate.

We now discuss a few additional algorithmic tools that go into the analysis of one specific case that cannot be learned using the above mentioned ideas. Suppose that there is one gate such that all linear forms appearing in it only span a constant dimensional space.

Learning linear forms when some of the gates have low rank

Assume the linear forms appearing in T_3 span a c dimensional space for some constant c. In this case, it can be that $\mathcal{S}_3(f)$ and $\mathcal{S}_2^{sp}(f)$ are both empty, and spaces in $\mathcal{S}_1(f)$ and $\mathcal{S}_2^{reg}(f)$ block any codimension 3 space from appearing in $\mathcal{S}_3(f)$. We mention a few ingredients that go into the analysis.

We need a new algorithmic insight, since $\mathcal{S}_1(f)$, $\mathcal{S}_2(f)$ and $\mathcal{S}_3(f)$ might be useless. This method also shows up in the learning of a few other other instances (of the structural partitioning) of $\Sigma\Pi\Sigma(3)$ circuits.

When T_3 has only low rank linear forms, it turns out we can then (essentially) compute $\mathcal{S}_2(T_1+T_2)$. Note that we do not have blackbox access to T_1+T_2. We would still like to compute $\mathcal{S}_2(T_1+T_2)$. The key observation is that if $\mathbb{V}(l,l') \in \mathcal{S}_2(T_1+T_2)$ then even though we don't know that $C \mod \langle l,l' \rangle$ is zero, we can still conclude that $C \mod \langle l,l' \rangle$ has *few essential variables* (see Definition 7) , i.e. it can be written as a polynomial depending on constantly many linear forms. In order to compute $\mathcal{S}_2(T_1+T_2)$ we will attempt to find all $\mathbb{V}(l,l')$ such that $C \mod \langle l,l' \rangle$ has few essential variables. We show that this can be

done using our algorithms for finding $\mathcal{S}_2$ spaces of a polynomial, combined with a suitable modification of an algorithm by Carlini [11] which can compute the number of essential variables in a polynomial. Once we can compute $\mathcal{S}_2(T_1 + T_2)$, we can use intersections of the kernels of these spaces to compute linear forms appearing in one of T_1 or T_2[3]

Learning linear forms when C is of special form

We say a circuit C is of special form if there is a linear form l such that when we go modulo l, the simple part of the circuit has low rank. Note, this is the case when we don't know how to bound and hence compute $\mathcal{S}_3(f)$. Thus again new ideas are needed. In this case we inspect the structure of the circuit and show that it must be one of three types In each of these types, though $\mathcal{S}_3(f)$ cannot be learned, we show how to bound and learn an interesting subset of $\mathcal{S}_3(f)$ (which we call $\mathcal{S}_3^*(f)$) that still contains enough useful codimension 3 spaces that allow us to learn some of the linear forms appearing in C. We then again have to consider cases based on whether all gates are high rank or not, and construct learning algorithms similar to the non-special form cases, but with $\mathcal{S}_3^*(f)$ playing the role of $\mathcal{S}_3(f)$.

2.4 Overview: From a few linear forms to reconstructing the entire circuit

Once we learn a few linear independent linear factors (we will actually ensure we learn $\Omega(\log d)$ linear factors) appearing in one of the gates (say T_1) then the high-level plan is to consider the circuit modulo each of the factors to obtain a projected circuit of top fan-in at most two. This can be learnt using a reconstruction algorithm for $\Sigma\Pi\Sigma(2)$ circuits from [47] as a unique $\Sigma\Pi\Sigma(2)$ circuit when the distance between the projected gates is high or as a $\Sigma\Pi\Sigma(1, d, r)$ circuit when the distance is low. Moreover by a result by Shpilka [45], given enough linearly independent projections of $T_2 + T_3$ suffices in recovering $T_2 + T_3$. An extension of this result by Karnin and Shpilka [25], gives a way of recovering $T_2 + T_3$ from enough $\Sigma\Pi\Sigma(1, d, r)$ projections of low distance $T_2 + T_3$.

Comparison to [25]

As described in our overview, once we have a few linear forms from a gate, the process of learning the entire circuit follows closely the outline from [25]. The main difference stems from how the few linear forms are obtained. The main technical contribution of this paper is to show how to efficiently compute these linear forms over large fields. In the works of [25, 45], the authors obtained the linear forms using a brute-force search approach, by searching over all possible linear forms with $O(\log d)$ variables, which took quasi-poly($|\mathbb{F}|$) time.

3 Preliminaries

A detailed discussion on the notations and preliminaries can be found in the full version of the paper [41].

[3] This is again a bit of a simplification of our algorithm. It could be that the rank of T_3 is super constant, or it could be that most of $\mathcal{S}_2(T_1 + T_2)$ is blocked by the $\mathcal{S}_1$ spaces.

3.1 Polynomial Identity Testing and Rank Bounds

A finite set of points S with the property that every line through two points of S passes through a third point in S is called a Sylvester-Gallai configuration. The famous Sylvester-Gallai theorem states that the only Sylvester-Gallai configurations in $\mathbb{R}^n$ are those formed by collinear points. This basic theorem about point-line incidences was extended to higher dimensional flats in [23, 10] over the Real numbers and in [3, 13] over $\mathbb{C}$. We define the *rank* of a set of vectors to be the dimension of the linear space they span.

▶ **Definition 4** ($\mathrm{SG}_k(\mathbb{F}, m)$)**.** *Let S be a set of non-zero vectors in $\mathbb{F}^{n+1}$ without multiples: ie no two vectors in S are scalar multiples of each other. Suppose that for every set $V \subseteq S$ of k linearly independent vectors, the linear span of V contains at least $k+1$ vectors of S. Then, the set S is said to be SG_k-closed. The largest possible rank of an SG_k-closed set of at most m vectors in $\mathbb{F}^n$ (for any n) is denoted by $\mathrm{SG}_k(\mathbb{F}, m)$.*

Over the field of Real numbers, it is known that $\mathrm{SG}_k(\mathbb{R}, m) = 2(k-1)$ [23, 10]. The rank of high dimensional Sylvester-Gallai configurations over $\mathbb{C}$ was bounded by 2^{c^k} for a fixed constant c in [3]. This bound was further improved to $\mathrm{SG}_k(\mathbb{C}, m) = c^k$(for a fixed constant c) in [13].

The polynomial time blackbox PIT algorithms for $\Sigma\Pi\Sigma(k)$ circuits over $\mathbb{R}$ and $\mathbb{C}$ follow from some strong structural properties of identically zero $\Sigma\Pi\Sigma(k)$ circuits. In [27] it was shown that the rank of any identically zero, simple and minimal $\Sigma\Pi\Sigma(k)$ circuit is at most some constant depending on k. This bound was improved in [42, 43], and the theorem below gives the best bound we know.

▶ **Theorem 5** ([43])**.** *Let C be a $\Sigma\Pi\Sigma(k)$ circuit, over field $\mathbb{F}$, that is simple, minimal and zero. Then, we have* $\mathrm{rank}(C) \leq 2k^2 + k \cdot \mathrm{SG}_k(\mathbb{F}, d)$.

Combining the above theorem with the best bounds we know for $\mathrm{SG}_k(\mathbb{R}, m)$ and $\mathrm{SG}_k(\mathbb{C}, m)$ we obtain the following,

▶ **Theorem 6.** *Let C be a simple, minimal and identically zero $\Sigma\Pi\Sigma(k)$ circuit over $\mathbb{R}$ or $\mathbb{C}$. Then there is an absolute constant $\mathcal{R}(k)$ depending only on k such that* $\mathrm{rank}(C) < \mathcal{R}(k)$*. If C is over $\mathbb{R}$ then we can bound* $\mathrm{rank}(C)$ *by $3k^2$. If C is over $\mathbb{C}$ then we can bound* $\mathrm{rank}(C)$ *by $2k^2 + k \cdot c^k$ for some absolute constant c.*

3.2 Essential Variables of a Polynomial

This notion will be useful in reconstruction when the input circuit is low rank, as well as when one of more gates is low rank.

▶ **Definition 7** ([28], Essential Variables)**.** *The number of essential variables in $f(x_1, \ldots, x_n)$ is the smallest t such that there exists an invertible linear transformation $A \in \mathbb{F}^{(n \times n)}$ on the variables such that $f(A \cdot \overline{x})$ depends on only t variables.*

4 Upper bounding the size of $\mathcal{S}_2(f)$ and $\mathcal{S}_3(f)$

In this section, we will show that if a degree d polynomial $f \in \mathbb{F}[\overline{x}]$ is computed by a $\Sigma\Pi\Sigma(3)$ circuit C then other then in certain degenerate cases, it will "vanish" on only finitely many codimension 3 subspaces. In the next sections we will show how to compute these subspaces and then how to extract the linear forms of C from these subspaces.

Given k linearly independent linear forms $l_1, l_2, \cdots l_k$, let $\mathbb{V}(l_1, l_2, \ldots, l_k) \subseteq \mathbb{F}^n$ denote the codimension k subspace of $\mathbb{F}^n$ corresponding to those vectors where $l_1, l_2, \cdots l_k$ evaluate to 0. We say that $\mathbb{V}(l_1, l_2, \ldots, l_k)$ is a vanishing space for a polynomial f if f vanishes on all points of $\mathbb{V}(l_1, l_2, \ldots, l_k)$.

Equivalently, consider any invertible linear transformation $\phi \in \mathbb{F}^{n \times n}$ such that $\phi(l_i) = x_i$ for all $i \in [k]$. Let $\phi \cdot f = f(\phi(\bar{x}))$. Then setting $x_1, x_2, \ldots x_k$ to 0 in $\phi \cdot f$ results in the identically 0 polynomial.

For a polynomial f defined over $\mathbb{F}^n$, we will define $\mathcal{S}_1(f)$ to be the set of codimension 1 subspaces over which f vanishes. $\mathcal{S}_1(f) = \{\mathbb{V}(l) \mid \mathbb{V}(l)$ is a vanishing space for$f\}$

We would like to define $\mathcal{S}_2(f)$ to be the set of codimension 2 subspaces over which f vanishes and try to show that this is finite. However note that if f has even one codimension 1 subspace on which it vanishes, then there will be infinitely many codimension 2 subspaces on which it vanishes, since for any $W \in \mathcal{S}_1(f)$, f vanishes on every single codimension 2 subspace of W. Thus when we define $S_2(f)$, we will not consider such subspaces.

Let $\mathcal{S}_2(f) = \{W \mid W$ is a codimension 2 subspace of $\mathbb{F}^n, f$ vanishes over W and $W \not\subset W'$ for any
$W' \in \mathcal{S}_1(f)\}$.

Note that any $W \in \mathcal{S}_2(f)$ is of the form $\mathbb{V}(l_1, l_2)$ for some two linear forms l_1, l_2. Moreover any two independent linear forms in the span of l_1 and l_2 will result in the same space W.

Similarly we define $\mathcal{S}_3(f)$ to be the set of codimension 3 subspaces W over which f vanishes such that W is not contained in any subspace from $\mathcal{S}_2(f)$ or $\mathcal{S}_1(f)$.

$\mathcal{S}_3(f) = \{W \mid W$ is a codimension 3 subspace of $\mathbb{F}^n, f$ vanishes over W and $W \not\subset W'$ for any $W' \in \mathcal{S}_1(f) \cup \mathcal{S}_2(f)\}$.

Note again that any $W \in \mathcal{S}_3(f)$ is of the form $\mathbb{V}(l_1, l_2, l_3)$ for some three linearly independent linear forms l_1, l_2, l_3. Moreover any three independent linear forms in the span of l_1, l_2 and l_3 will result in the same space W.

▶ **Lemma 8.** *Let f be a degree d polynomial. Then, $|\mathcal{S}_1(f)| \leq d$.*

Proof. The proof is quite simple. Any linear form l such that f vanishes on $\mathbb{V}(l)$ must be a linear factor of f. Since f has degree d, it can have at most d distinct factors. ◀

We will also show how to bound the size of $\mathcal{S}_2(f)$ and $\mathcal{S}_3(f)$ under additional structural assumptions of f. Note that if f is computed by an $\Sigma\Pi\Sigma(3, d)$ circuit of the form $T_1 + T_2 + T_3$, then by picking one linear form from each of the three multiplication gates, we can obtain codimension 3 spaces on which f vanishes. Learning these spaces will eventually allow us to learn the linear forms, and it will be an important ingredient of our final algorithm. Note however that f might have other codimension 3 vanishing subspaces that are not of this form. The main goal of this section is to show that nevertheless under some structural assumptions, we are able to bound the number of these spaces.

Before we state and prove our upper bounds for $\mathcal{S}_2$ and $\mathcal{S}_3$ spaces we state and prove a slightly modified version of a result from [48] that we will be useful for us.

▷ Claim 9. Let P_1, P_2, P_3 be distinct 2-dim subspaces of some vector space V such that $\dim(\text{span}(P_1, P_2, P_3)) \geq 4$ and P_3 is not contained in $\text{span}(P_1, P_2)$. Suppose that P is a 2-dim subspace such that for each $i \in [3]$, $\dim(P_i \cap P) = 1$. Then either

- $\dim(P_1 \cap P_2) = 1$ and $P_1 \cap P_2 \subset P$ or
- $\dim(P_3 \cap \text{span}(P_1, P_2)) = 1$ and $P_3 \cap \text{span}(P_1, P_2) \subset P$.

Proof. Case (1): $P_1 \cap P = P_2 \cap P$. Then since P_1 and P_2 are distinct, clearly $\dim(P_1 \cap P_2) = 1$ and $P_1 \cap P_2 \subset P$.

Case(2): $P_1 \cap P \neq P_2 \cap P$.

Here, we have $\dim(\text{span}(P \cap P_1, P \cap P_2)) = \dim(P \cap \text{span}(P_1, P_2))) = 2$, but as $\dim(P) = 2$, we have $P \subset \text{span}(P_1, P_2)$. As the P_1, P_2 and P_3 together span a space of dimension ≥ 4 and $P_3 \not\subset \text{span}(P_1, P_2)$, thus $\dim(P_3 \cap \text{span}(P_1, P_2)) \leq 1$. Now as $P \subset \text{span}(P_1, P_2)$, thus $(P_3 \cap P) \subset (P_3 \cap \text{span}(P_1, P_2))$. But we also know $\dim(P_3 \cap P) = 1$, and therefore $\dim(P_3 \cap \text{span}(P_1, P_2)) = 1$ and $P_3 \cap P = P_3 \cap \text{span}(P_1, P_2)$, which means $P_3 \cap \text{span}(P_1, P_2) \subset P$. ◁

4.1 Bounding the number of vanishing codimension 2 subspaces

▶ **Lemma 10.** *Let $\mathbb{F}$ be a field that is $\mathbb{R}$ or $\mathbb{C}$. Let f be a degree d, n variate polynomial in $\mathbb{F}[x_1, \ldots, x_n]$. Then there is an absolute constant c_2 such that if f is computed by a $\Sigma\Pi\Sigma(3)$ circuit $C = T_1 + T_2 + T_3$ with $\text{rank}(\text{sim}(C)) \geq c_2$, then*

$$\mathcal{S}_2(f) \leq O(d^7).$$

Proof. We will be dividing our analysis into the following cases.

1. **Case 1:** We bound the number of $W = \mathbb{V}(l_1, l_2) \in \mathcal{S}_2(f)$ such that for some $i \in [3]$, T_i vanishes on $\mathbb{V}(l_1, l_2)$.
   Wlog $T_i = T_1$. Note that if T_1 vanishes on $\mathbb{V}(l_1, l_2)$, then there must be some $l \in \text{span}(l_1, l_2)$ such that l divides T_1. Note that since $\mathbb{V}(l_1, l_2) \notin \mathbb{V}(l)$ for any $\mathbb{V}(l) \in \mathcal{S}_1(f)$, thus $C' = (T_2 + T_3 \mod l)$ is nonzero, and moreover there is some linear form l' with $\text{span}(l, l') = \text{span}(l_1, l_2)$ such that $C' \mod l' \equiv 0$. There are at most d choices for l' from the factors of $(T_2 + T_3 \mod l)$, and there were at most d choices for l (once we fix i). Therefore, there are at most $O(d^2)$ possibilities for $W \in \mathcal{S}_2(f)$.
2. **Case 2:** We bound the number of $W = \mathbb{V}(l_1, l_2) \in \mathcal{S}_2(f)$ such that no T_i vanishes on $\mathbb{V}(l_1, l_2)$.
   Consider $C \mod \langle l_1, l_2 \rangle$ which is of the form $G \times (T_1' + T_2' + T_3')$ where $T_1' + T_2' + T_3'$ is a simple, minimal $\Sigma\Pi\Sigma(3)$ circuit computing the identically zero polynomial and G is a product of linear forms. By the rank bound given in Theorem 6, $\text{rank}(T_1' + T_2' + T_3') < \mathcal{R}(3)$ where $\mathcal{R}(3)$ is some absolute constant. Over the real numbers $\mathcal{R}(3) = 5$.
   Let c_2 be any constant greater than $\mathcal{R}(3) + 10$. Therefore, $\text{rank}(\text{sim}(C)) \geq \mathcal{R}(3) + 10$. When we consider $C \mod \langle l_1, l_2 \rangle$, the linear forms appearing in the gates of $\text{sim}(C)$ get mapped to linear forms in G or in $(T_1' + T_2' + T_3')$. The rank of those linear forms that get mapped to $(T_1' + T_2' + T_3')$ can be at most $\mathcal{R}(3) + 2$. Thus the rank of the set of linear forms that gets mapped to G is at least 8.
   Consider 3 linear forms (not all the same) l_{1j}, l_{2j}, l_{3j} from distinct product gates of $\text{sim}(C)$ such that when we consider them $\mod \langle l_1, l_2 \rangle$, they map to the same linear form l (we don't distinguish between a linear form and its multiple) and hence all get mapped to the linear forms of G. Call such triple of linear forms a "collapsing" triple when we go $\mod \langle l_1, l_2 \rangle$. These linear forms must be of the form

   $$l_{1j} = l + \alpha_1 l_1 + \alpha_2 l_2$$

   $$l_{2j} = l + \beta_1 l_1 + \beta_2 l_2$$

   $$l_{3j} = l + \gamma_1 l_1 + \gamma_2 l_2$$

   where the α, β, γ denote field constants.
   A collapsing triple can either span a 2 or 3 dimensional space. Now, since the rank of linear forms mapping to G is at least 8, one of the following two scenarios must occur.

- **Case 2(a):** There are two collapsing triples (l_{1j}, l_{2j}, l_{3j}) and (l_{1k}, l_{2k}, l_{3k}) such that each spans a 3-dim space and jointly the two triples span at least a 4-dim space.
  Let $V_j = \text{span}(l_{1j}, l_{2j}, l_{3j})$ and $V_k = \text{span}(l_{1k}, l_{2k}, l_{3k})$. Let $U = span(l_1, l_2)$. Since V_j and V_k both become 1-dimensional when $U = \text{span}(l_1, l_2)$ is projected to 0, it must hold that $U \subset V_j$ and $U \subset V_k$. Moreover as V_j and V_k are distinct 3-dim spaces, thus it must hold that $V_j \cap V_i = U$. Thus V_j and V_k determine U. In particular, the two triples of linear forms determine U.
  Since, there can be d possibilities for each of $l_{1j}, l_{2j}, l_{3j}, l_{1k}, l_{2k}, l_{3k}$, we have $O(d^6)$ possibilities for U, and hence for $\mathbb{V}(l_1, l_2) \in \mathcal{S}_2(f)$.
- **Case 2(b):** The collapsing triples that span 2-dim spaces have combined rank at least 5.
  Clearly there must be at least 3 collapsing triples. Let V_i, V_j and V_k be the span of of 3 of the triples such that $\text{span}(V_i \cup V_j \cup V_k) \geq 5$. Since V_i, V_j and V_k all become 1-dimensional when $U = \text{span}(l_1, l_2)$ is projected to 0, it follows that each of V_i, V_j, V_k intersects U in a 1-dim space. By Claim 9, it follows that knowing V_i, V_j and V_k is enough to determine a vector $l \in U$. Once l is determined, then the rest of the argument is similar to case 1. $C' = (T_2 + T_3 \mod l)$ is nonzero, and moreover there is some linear form l' with $\text{span}(l, l') = \text{span}(l_1, l_2)$ such that $C' \mod l' \equiv 0$. There are at most d choices for l' from the factors of $(T_2 + T_3 \mod l)$. Since there were at most $O(d^6)$ possibilities for V_i, V_j and V_k, and hence at most $O(d^6)$ possibilities for the choice of l, thus overall there are at most $O(d^7)$ possibilities for U and hence for for $W \in \mathcal{S}_2(f)$. ◀

4.2 Bounding the number of vanishing codimension 3 subspaces

▶ **Lemma 11.** *Let $\mathbb{F}$ be a field that is $\mathbb{R}$ or $\mathbb{C}$. Let f be a degree d, n variate polynomial in $\mathbb{F}[x_1, \ldots, x_n]$. Then there is an absolute constant c_3 such that if f is computed by a $\Sigma\Pi\Sigma(3)$ circuit $C = T_1 + T_2 + T_3$ with the following properties*

1. $\text{rank}(\text{sim}(C)) \geq c_3$,
2. *There is no linear form l such that $(C \mod l)$ is nonzero and* $\text{rank}(\text{sim}(C \mod l)) < c_2$.

then

$$\mathcal{S}_3(f) \leq O(d^{15}).$$

Proof. We will be dividing our analysis into the following cases.

1. **Case 1:** We bound the number of $W = \mathbb{V}(l_1, l_2, l_3) \in \mathcal{S}_3(f)$ such that for some $i \in [3]$, T_i vanishes on $\mathbb{V}(l_1, l_2, l_3)$.
   Note that if T_i vanishes on $\mathbb{V}(l_1, l_2, l_3)$, then there must be some $l \in \text{span}(l_1, l_2, l_3)$ such that l divides T_i. Note that since $\mathbb{V}(l_1, l_2, l_3) \notin \mathbb{V}(l)$ for any $\mathbb{V}(l) \in \mathcal{S}_1(f)$ thus $C' = (T_2 + T_3 \mod l)$ is nonzero. Moreover by assumption, $\text{rank}(\text{sim}(C \mod l)) = \text{rank}(\text{sim}(T_2 + T_3 \mod l)) \geq c_2$.
   Observe that $(T_2 + T_3 \mod l)$ must vanish when we consider it $\mod \langle l_a, l_b \rangle$ for any l_a, l_b such that $\text{span}(l, l_a, l_b) = \text{span}(l_1, l_2, l_3)$. Thus once l is fixed, any such $W' = \text{span}(l_a, l_b)$ is a vanishing codimension 2 space for $(T_2 + T_3 \mod l)$. By Lemma 10, there are at most $O(d^7)$ choices for W'. Given that there are at most $O(d)$ choices for l, thus there are totally $O(d^8)$ possibilities for $W \in \mathcal{S}_3(f)$ in this case.
2. **Case 2:** We bound the number of $W = \mathbb{V}(l_1, l_2, l_3) \in \mathcal{S}_3(f)$ such that no T_i vanishes on $\mathbb{V}(l_1, l_2, l_3)$.
   Consider $C \mod \langle l_1, l_2, l_3 \rangle$ which is of the form $G \times (T_1' + T_2' + T_3')$ where $T_1' + T_2' + T_3'$ is a simple, minimal $\Sigma\Pi\Sigma(3)$ circuit computing the identically zero polynomial and G is a product of linear forms. By the rank bound given in Theorem 6, $\text{rank}(T_1' + T_2' + T_3') < \mathcal{R}(3)$ where $\mathcal{R}(3)$ is some absolute constant. Over the real numbers $\mathcal{R}(3) = 5$.

Let c_3 be any constant greater than $\mathcal{R}(3)+17$. Therefore, $\text{rank}(\text{sim}(C)) \geq \mathcal{R}(3)+17$. When we consider $C \mod \langle l_1, l_2, l_3\rangle$, the linear forms appearing in the gates of $\text{sim}(C)$ get mapped to linear forms in G or in $(T_1'+T_2'+T_3')$. The rank of those linear forms that get mapped to $(T_1'+T_2'+T_3')$ can be at most $\mathcal{R}(3)+3$. Thus the rank of the set of linear forms that gets mapped to G is at least 14.

Note that by assumption, there is no linear form l such that $C \mod l$ is nonzero and $\text{rank}(\text{sim}(C \mod l)) \leq c_2$. However there could exists two independent linear forms $l, l' \in \text{span}(l_1, l_2, l_3)$, such that $\text{rank}(\text{sim}(C \mod \langle l, l'\rangle)) < \mathcal{R}(3)$. We consider two further sub-cases based on whether this happens or not.

(a) Case 2(a): We bound the number of $W = \mathbb{V}(l_1, l_2, l_3) \in \mathcal{S}_3(f)$ such that no T_i vanishes on $\mathbb{V}(l_1, l_2, l_3)$ but there exists two independent linear forms $l, l' \in \text{span}(l_1, l_2, l_3)$, such that $\text{rank}(\text{sim}(C \mod \langle l, l'\rangle)) < \mathcal{R}(3)$.

The analysis for this case is almost identical to that of Case 2 in Lemma 10. By the analysis of Case 2 in Lemma 10, the number of spaces $W' = \mathbb{V}(l, l')$ such that $\text{rank}(\text{sim}(C \mod \langle l, l'\rangle)) < \mathcal{R}(3)$ is at most $O(d^7)$. For each fixing of $W' = \mathbb{V}(l, l')$ we now count the number of $W = \mathbb{V}(l_1, l_2, l_3) \in \mathcal{S}_3(f)$ such that $l, l' \in \text{span}(l_1, l_2, l_3)$. Thus any such W is of the form $\mathbb{V}(l, l', l'')$, Note that since for any space in $\mathcal{S}_3(f)$, it is not contained in any space in $\mathcal{S}_2(f)$, we only need to consider those $W' = \mathbb{V}(l, l')$ such that $C \mod \langle l, l'\rangle \neq 0$. Since $C' = (C \mod \langle l, l'\rangle) \neq 0$ but $(C \mod \langle l, l', l''\rangle) = 0$ thus l'' must be a linear factor of C' and hence there can be only d choices for l''. Thus in this case there are at most $O(d^8)$ possibilities for $W \in \mathcal{S}_3(f)$.

(b) Case 2(b): We bound the number of $W = \mathbb{V}(l_1, l_2, l_3) \in \mathcal{S}_3(f)$ such that no T_i vanishes on $\mathbb{V}(l_1, l_2, l_3)$ and there do not exist two independent linear forms $l, l' \in \text{span}(l_1, l_2, l_3)$, such that $\text{rank}(\text{sim}(C \mod \langle l, l'\rangle)) < \mathcal{R}(3)$.

Consider 3 linear forms (not all the same) l_{1j}, l_{2j}, l_{3j} from distinct product gates of $\text{sim}(C)$ such that when we consider them $\mod \langle l_1, l_2, l_3\rangle$, they map to the same linear form l (we don't distinguish between a linear form and its multiple) and hence all get mapped to the linear forms of G. Call such triple of linear forms a "collapsing" triple when we go $\mod \langle l_1, l_2, l_3\rangle$. These linear forms must be of the form

$$l_{1j} = l + \alpha_1 l_1 + \alpha_2 l_2 + \alpha_3 l_3$$

$$l_{2j} = l + \beta_1 l_1 + \beta_2 l_2 + \beta_3 l_3$$

$$l_{3j} = l + \gamma_1 l_1 + \gamma_2 l_2 + \gamma_3 l_3$$

where the α, β, γ denote field constants.

A collapsing triple can either span a 2 or 3 dimensional space. Now, since the rank of linear forms mapping to G is at least 14, one of the following two scenarios must occur.

Either (i) The set of collapsing triples such that each spans a 3-dim space has combined rank (over all the triples) of at least 7 or (ii) The set of collapsing triples such that each spans a 2-dim space has combined rank at least 8.

We will separately analyze both these subcases.

- **Case 2(b)(i):** The set of collapsing triples such that each spans a 3-dim space has combined rank (over all the triples) of at least 7.

  Let V_1, V_2, V_3 be the vector spaces spanned by three of the triples such that the V_1, V_2 and V_3 are distinct and $\text{span}(V_1 \cup V_2 \cup V_3) \geq 5$. Three such vector spaces must exist.

Now since going $\bmod \langle l_1, l_2, l_3\rangle$ maps these triples to a line l, letting $U = \text{span}(l_1, l_2, l_3)$, it must hold that for all $i \in [3]$, $\dim(V_i \cap U) = 2$. It follows that for all $i, j \in [3]$, $\dim(V_i \cap V_j) \geq 1$. Since the spaces are distinct, thus $\dim(V_i \cap V_j)$ can equal 1 or 2.

Now if for any $i, j \in [3]$, $\dim(V_i \cap V_j) = 1$ then thus intersection must be contained in U. Thus knowing V_i and V_j determines a 1-dim subspace of U. Let l' be the linear form representing this space. Since l' is determined by two of the triples, thus there are at most $O(d^6)$ possibilities for l'. Once l' is determined, we consider $C' = C \mod l'$. This is nonzero and by assumption, the rank of its simple part is at least c_2. By Lemma 10 there are only $O(d^7)$ choices of co-dim 2 subspaces modulo which C' vanishes (we disregard those co-dim 2 spaces which contain a co-dim 1 space on which C' vanishes. Thus in total, in this case there are at most $O(d^{13})$ choices of $W = \mathbb{V}(l_1, l_2, l_3) \in \mathcal{S}_3(f)$.

We need to still consider the case when for all distinct $i, j \in [3]$, $\dim(V_i \cap V_j) = 2$. We first prove the following simple claim.

▷ Claim 12. Let V_1, V_2, V_3 be distinct 3-dim subspaces of some vector space V such that $\dim(\text{span}(V_1, V_2, V_3)) \geq 5$. Suppose that for any distinct $i, j \in [3]$, $\dim(V_i \cap V_j) = 2$. Then there exists a subspace $U' \subseteq V$ such that $\dim(U') = 2$ such that for any distinct $i, j \in [3]$, $(V_i \cap V_j) = U'$.

Proof. Since V_1, V_2 are distinct 3-dimensional spaces with $\dim(V_1 \cap V_2) = 2$, we have $\dim(V_1 \cup V_2) = 4$. As $\dim(\text{span}(V_1, V_2, V_3)) \geq 5$, we have $V_3 \not\subset V_1 \cup V_2$. If V_3 intersects V_1 and V_2 in distinct 2-dimensional spaces, then $\dim((V_3 \cap V_1) \cup (V_3 \cap V_2)) \geq 3$. But $(V_3 \cap V_1) \cup (V_3 \cap V_2) = V_3 \cap (V_1 \cup V_2) \subseteq V_3$ and as $\dim(V_3) = 3$, $V_3 = V_3 \cap (V_1 \cup V_2) \subseteq (V_1 \cup V_2)$, which is a contradiction. Therefore, V_3 intersects V_1 and V_2 in the same 2-dimensional plane, let it be $U' = V_1 \cap V_3 = V_2 \cap V_3$. Moreover, this means $U' \subset V_1$ and $U' \subset V_2$, and therefore $U' \subseteq (V_1 \cap V_2)$. As $\dim(V_1 \cap V_2) = 2$, we have $U' = (V_1 \cap V_2)$. ◁

By the above claim, it follows that $V_1 \cap V_2 \cap V_3 = U'$ for some 2-dim space U'. Moreover we know that U is a 3-dim space intersecting each V_i in a 2-dim space. It must hold that $U' \subseteq U$. If not, then $\dim(U' \cap U) \leq 1$. Thus U needs to still intersect each V_i in a vector outside of U'. These three additional vectors will be distinct and also linearly independent since $\text{span}(V_1 \cup V_2 \cup V_3) \geq 5$. This is not possible since $\dim(U) = 3$.

Thus $U' \subseteq U$. Hence V_1 and V_2 determine U' and hence determine a 2-dim subspace of U. Since U' is determined by two of the triples, thus there are at most $O(d^6)$ possibilities for U'. Once we fix $U' \subseteq U$, it remains to find the number of $W = \mathbb{V}(l_1, l_2, l_3) \in \mathcal{S}_3(f)$ such that $U' \in \text{span}(l_1, l_2, l_3)$. When we go mod U', we can assume that C does not vanish (since we are only counting those W that are not contained in an $\mathcal{S}_2$ space), and hence for each fixing of U', since $C \mod U'$ has only at most d linear factors, thus there are at most $O(d^7)$ possibilities for W.

- **Case 2(b)(ii):** The set of collapsing triples such that each spans a 2-dim space, has combined rank at least 8.

Let P_1, P_2, P_3, P_4 be the vector spaces spanned by four of the triples such that the P_1, P_2, P_3 and P_4 are distinct and $\text{span}(P_1, P_2, P_3, P_4) \geq 5$. Moreover $P_3 \not\subseteq \text{span}(P_1, P_2)$ and $P_4 \not\subseteq \text{span}(P_1, P_2, P_3)$. Four such vector spaces must exist.

Now since going $\bmod \langle l_1, l_2, l_3\rangle$ maps these triples to a single line l, letting $U = \text{span}(l_1, l_2, l_3)$, it must hold that for all $i \in [4]$, $\dim(P_i \cap U) = 1$.

There are three subcases that we will consider. In each case we show that the knowledge of the four subspaces P_i allows us to determine a single linear form $l' \in U$. There are at most hence d^8 choices for l'. Once we find and fix $l' \in U$ we consider $C' = C \mod l'$. This is nonzero and by assumption, the rank of its simple part is at least c_2. By Lemma 10 there are only $O(d^7)$ choices of co-dim 2 subspaces modulo which C' vanishes (we disregard those co-dim 2 spaces which contain a co-dim 1 space on which C' vanishes. Thus in total, the number of choices of $W = \mathbb{V}(l_1, l_2, l_3) \in \mathcal{S}_3(f)$ is $O(d^{15})$.

- **subcase 1:** P_1, P_2 are such that $P_1 \cap U = P_2 \cap U$.
  In particular $P_1 \cap P_2 \subset U$. Thus knowing P_1 and P_2 determines a 1-dim subspace of U.
- **subcase 2:** The 2-dim space defined by $P' = \text{span}(U \cap P_1, U \cap P_2) \subset U$ contains the line $U \cap P_3$. Since $P_3 \not\subseteq \text{span}(P_1, P_2)$, thus $\dim(\text{span}(P_1, P_2, P_3)) \geq 4$ and $\dim(P' \cap P_1) = \dim(P' \cap P_2) = \dim(P' \cap P_3) = 1$.
  Using Claim 9, it follows that P' and hence U contains the line $P_3 \cap \text{span}(P_1, P_2)$.
- **subcase 3:** $\dim(\text{span}(U \cap P_1, U \cap P_2, U \cap P_3)) = 3$ i.e. the three 1-dim intersections are independent. But as $\dim(U) = 3$ and $\text{span}(U \cap P_1, U \cap P_2, U \cap P_3) \subseteq U$, we have $U = U \cap \text{span}(P_1, P_2, P_3)$, which means $U \subset \text{span}(P_1, P_2, P_3)$. As $\dim(P_4 \cap U) = 1$, we have $\dim(P_4 \cap \text{span}(P_1, P_2, P_3)) \geq 1$. By assumption, $P_4 \not\subseteq \text{span}(P_1, P_2, P_3)$ and therefore $\dim(P_4 \cap \text{span}(P_1, P_2, P_3)) \leq 1$. So, $P_4 \cap \text{span}(P_1, P_2, P_3) = U \cap P_4$ is a 1-dim space that is contained in U, and is determined by knowing P_1, P_2, P_3 and P_4 ◀

References

1 M. Agrawal and V. Vinay. Arithmetic circuits: A chasm at depth four. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 67–75, 2008.

2 D. Angluin. Queries and concept learning. *Machine Learning*, 2:319–342, 1988.

3 Boaz Barak, Zeev Dvir, Avi Wigderson, and Amir Yehudayoff. Fractional sylvester–gallai theorems. *Proceedings of the National Academy of Sciences*, 110(48):19213–19219, 2013.

4 A. Beimel, F. Bergadano, N. H. Bshouty, E. Kushilevitz, and S. Varricchio. Learning functions represented as multiplicity automata. *J. ACM*, 47(3):506–530, 2000. doi:10.1145/337244.337257.

5 M. Ben-Or and P. Tiwari. A deterministic algorithm for sparse multivariate polynominal interpolation. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing (STOC)*, pages 301–309, 1988.

6 Vishwas Bhargava, Ankit Garg, Neeraj Kayal, and Chandan Saha. Learning generalized depth three arithmetic circuits in the non-degenerate case. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2022)*, volume 245 of *LIPIcs*, pages 21:1–21:22. Schloss-Dagstuhl-Leibniz Zentrum für Informatik, 2022. doi:10.4230/LIPICS.APPROX/RANDOM.2022.21.

7 Vishwas Bhargava, Shubhangi Saraf, and Ilya Volkovich. Reconstruction of depth-4 multilinear circuits. *SODA 2020*, 2020.

8 Vishwas Bhargava, Shubhangi Saraf, and Ilya Volkovich. Reconstruction algorithms for low-rank tensors and depth-3 multilinear circuits. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 809–822, 2021. doi:10.1145/3406325.3451096.

9 Vishwas Bhargava and Devansh Shringi. Faster & deterministic FPT algorithm for worst-case tensor decomposition. *Electron. Colloquium Comput. Complex.*, TR24-123, 2024. URL: https://eccc.weizmann.ac.il/report/2024/123.

10 William E. Bonnice and Michael Edelstein. Flats associated with finite sets in P^d. *Niew. Arch. Wisk*, 15:11–14, 1967.

11 Enrico Carlini. Reducing the number of variables of a polynomial. In *Algebraic geometry and geometric modeling*, pages 237–247. Springer, 2006. doi:10.1007/978-3-540-33275-6_15.

12 Pranjal Dutta, Prateek Dwivedi, and Nitin Saxena. Deterministic Identity Testing Paradigms for Bounded Top-Fanin Depth-4 Circuits. In Valentine Kabanets, editor, *36th Computational Complexity Conference (CCC 2021)*, volume 200 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 11:1–11:27, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CCC.2021.11.

13 Zeev Dvir, Shubhangi Saraf, and Avi Wigderson. Improved rank bounds for design matrices and a new proof of kelly's theorem. In *Forum of Mathematics, Sigma*, volume 2, page e4. Cambridge University Press, 2014.

14 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, pages 592–601, New York, NY, USA, 2005. Association for Computing Machinery. doi:10.1145/1060590.1060678.

15 M. A. Forbes and A. Shpilka. Quasipolynomial-time identity testing of non-commutative and read-once oblivious algebraic branching programs. *Electronic Colloquium on Computational Complexity (ECCC)*, 19:115, 2012.

16 Abhibhav Garg, Rafael Oliveira, Shir Peleg, and Akash Kumar Sengupta. Radical sylvester-gallai theorem for tuples of quadratics. In *38th Computational Complexity Conference (CCC 2023)*, pages 20:1–20:30. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2023. doi:10.4230/LIPICS.CCC.2023.20.

17 Abhibhav Garg, Rafael Oliveira, and Akash Kumar Sengupta. Robust Radical Sylvester-Gallai Theorem for Quadratics. In Xavier Goaoc and Michael Kerber, editors, *38th International Symposium on Computational Geometry (SoCG 2022)*, volume 224 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 42:1–42:13, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.SoCG.2022.42.

18 Ankit Garg, Neeraj Kayal, and Chandan Saha. Learning sums of powers of low-degree polynomials in the non-degenerate case. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 889–899. IEEE, 2020. doi:10.1109/FOCS46700.2020.00087.

19 A. Gupta, P. Kamath, N. Kayal, and R. Saptharishi. Arithmetic circuits: A chasm at depth three. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 578–587, 2013. doi:10.1109/FOCS.2013.68.

20 A. Gupta, N. Kayal, and S. V. Lokam. Efficient reconstruction of random multilinear formulas. In *IEEE 52nd Annual Symposium on Foundations of Computer Science, FOCS*, pages 778–787, 2011. doi:10.1109/FOCS.2011.70.

21 A. Gupta, N. Kayal, and S. V. Lokam. Reconstruction of depth-4 multilinear circuits with top fanin 2. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC)*, pages 625–642, 2012. Full version at https://eccc.weizmann.ac.il/report/2011/153.

22 A. Gupta, N. Kayal, and Y. Qiao. Random arithmetic formulas can be reconstructed efficiently. *Computational Complexity*, 23(2):207–303, 2014. doi:10.1007/s00037-014-0085-0.

23 Sten Hansen. A generalization of a theorem of sylvester on the lines determined by a finite point set. *Mathematica Scandinavica*, 16(2):175–180, 1965.

24 Zohar S Karnin and Amir Shpilka. Black box polynomial identity testing of generalized depth-3 arithmetic circuits with bounded top fan-in. In *2008 23rd Annual IEEE Conference on Computational Complexity*, pages 280–291. IEEE, 2008.

25 Zohar S Karnin and Amir Shpilka. Reconstruction of generalized depth-3 arithmetic circuits with bounded top fan-in. In *2009 24th Annual IEEE Conference on Computational Complexity*, pages 274–285. IEEE, 2009.

26 N. Kayal, V. Nair, C. Saha, and S. Tavenas. Reconstruction of full rank algebraic branching programs. In *32nd Computational Complexity Conference, CCC 2017.*, pages 21:1–21:61, 2017. doi:10.4230/LIPIcs.CCC.2017.21.

27 N. Kayal and S. Saraf. Blackbox polynomial identity testing for depth 3 circuits. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 198–207, 2009. Full version at https://eccc.weizmann.ac.il/report/2009/032.

28 Neeraj Kayal. Efficient algorithms for some special cases of the polynomial equivalence problem. In *Proceedings of the twenty-second annual ACM-SIAM symposium on Discrete algorithms*, pages 1409–1421. SIAM, 2011. doi:10.1137/1.9781611973082.108.

29 Neeraj Kayal, Vineet Nair, and Chandan Saha. Average-case linear matrix factorization and reconstruction of low width algebraic branching programs. *computational complexity*, 28:749–828, 2019. doi:10.1007/S00037-019-00189-0.

30 Neeraj Kayal and Chandan Saha. Reconstruction of non-degenerate homogeneous depth three circuits. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 413–424, 2019. doi:10.1145/3313276.3316360.

31 A. Klivans and A. Shpilka. Learning restricted models of arithmetic circuits. *Theory of computing*, 2(10):185–206, 2006. doi:10.4086/TOC.2006.V002A010.

32 A. Klivans and D. Spielman. Randomness efficient identity testing of multivariate polynomials. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC)*, pages 216–223, 2001.

33 P. Koiran. Arithmetic circuits: the chasm at depth four gets wider. *CoRR*, abs/1006.4700, 2010. arXiv:1006.4700.

34 Nutan Limaye, Srikanth Srinivasan, and Sébastien Tavenas. Superpolynomial lower bounds against low-depth algebraic circuits. In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 804–814, 2022. doi:10.1109/FOCS52979.2021.00083.

35 Rafael Oliveira and Akash Kumar Sengupta. Radical sylvester-gallai theorem for cubics. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 212–220. IEEE, 2022. doi:10.1109/FOCS54457.2022.00027.

36 Rafael Oliveira and Akash Kumar Sengupta. Strong algebras and radical sylvester-gallai configurations. In *Proceedings of the 56th Annual ACM Symposium on Theory of Computing*, pages 95–105, 2024. doi:10.1145/3618260.3649617.

37 Shir Peleg and Amir Shpilka. Polynomial time deterministic identity testing algorithm for $\Sigma^{[3]}\Pi\Sigma\Pi^{[2]}$ circuits via Edelstein–Kelly type theorem for quadratic polynomials. In *Proceedings of the 53rd Annual ACM SIGACT Symposium on Theory of Computing*, pages 259–271, 2021. doi:10.1145/3406325.3451013.

38 Shir Peleg and Amir Shpilka. A generalized sylvester–gallai-type theorem for quadratic polynomials. In *Forum of Mathematics, Sigma*, volume 10, page e112. Cambridge University Press, 2022.

39 Shir Peleg and Amir Shpilka. Robust Sylvester-Gallai Type Theorem for Quadratic Polynomials. In Xavier Goaoc and Michael Kerber, editors, *38th International Symposium on Computational Geometry (SoCG 2022)*, volume 224 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 43:1–43:15, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.SoCG.2022.43.

40 Shir Peleg, Amir Shpilka, and Ben Lee Volk. Tensor Reconstruction Beyond Constant Rank. In Venkatesan Guruswami, editor, *15th Innovations in Theoretical Computer Science Conference (ITCS 2024)*, volume 287 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 87:1–87:20, Dagstuhl, Germany, 2024. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.ITCS.2024.87.

41 Shubhangi Saraf and Devansh Shringi. Reconstruction of depth 3 arithmetic circuits with top fan-in 3. *Electron. Colloquium Comput. Complex.*, TR25-008, 2025. URL: https://eccc.weizmann.ac.il/report/2025/008.

42 Nitin Saxena and Comandur 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*, pages 431–440, 2011.

43 Nitin Saxena and Comandur Seshadhri. From sylvester-gallai configurations to rank bounds: Improved blackbox identity test for depth-3 circuits. *Journal of the ACM (JACM)*, 60(5):1–33, 2013. doi:10.1145/2528403.

44 Jacob T Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM (JACM)*, 27(4):701–717, 1980. doi:10.1145/322217.322225.

45 Amir Shpilka. Interpolation of depth-3 arithmetic circuits with two multiplication gates. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 284–293, 2007. doi:10.1145/1250790.1250833.

46 Amir Shpilka. Sylvester-gallai type theorems for quadratic polynomials. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 1203–1214, 2019. doi:10.1145/3313276.3316341.

47 Gaurav Sinha. Reconstruction of Real Depth-3 Circuits with Top Fan-In 2. In Ran Raz, editor, *31st Conference on Computational Complexity (CCC 2016)*, volume 50 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 31:1–31:53, Dagstuhl, Germany, 2016. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CCC.2016.31.

48 Gaurav Sinha. Efficient reconstruction of depth three arithmetic circuits with top fan-in two. In *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*, pages 118:1–118:33. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.ITCS.2022.118.

49 S. Tavenas. Improved bounds for reduction to depth 4 and depth 3. In *MFCS*, pages 813–824, 2013. doi:10.1007/978-3-642-40313-2_71.

50 Richard Zippel. Probabilistic algorithms for sparse polynomials. In *International symposium on symbolic and algebraic manipulation*, pages 216–226. Springer, 1979. doi:10.1007/3-540-09519-5_73.