Electronic Colloquium on Computational Complexity, Report No. 222 (2025)

Reconstruction of Depth-3 Arithmetic Circuits with Constant Top Fan-in

Shubhangi Saraf* Devansh Shringi† Narmada Varadarajan ‡

December 28, 2025

Abstract

In this paper, we give the first subexponential (in fact, quasi-polynomial time) reconstruction algorithm for depth-3 circuits of any constant top fan-in ($\Sigma\Pi\Sigma(k)$ circuits) over $\mathbb{R}$, $\mathbb{C}$, or any large characteristic finite field $\mathbb{F}$. More explicitly, we show that for any constant k, given black-box access to an n-variate polynomial f computed by a $\Sigma\Pi\Sigma(k)$ circuit of size s, there is a randomized algorithm that runs in time quasi-poly(n, s) and outputs a generalized $\Sigma\Pi\Sigma(k)$ circuit computing f. The size s includes the bit complexity of coefficients appearing in the circuit: this is the max bit complexity if the field is $\mathbb{R}$ or $\mathbb{C}$, and $\log |\mathbb{F}|$ if the field is finite.

Depth-3 circuits of constant fan-in ($\Sigma\Pi\Sigma(k)$ circuits) and closely related models have been very well studied in the context of polynomial identity testing (PIT). In this paper, we build upon the structural results for identically zero $\Sigma\Pi\Sigma(k)$ circuits that were studied in the context of PIT. Using connections to discrete geometry, we prove new structural properties of *vanishing spaces* of polynomials computed by such circuits.

Prior to our work, the only subexponential reconstruction algorithm for $\Sigma\Pi\Sigma(k)$ circuits is by [Karnin–Shpilka, CCC 2009]. However, the run time is quasipolynomial in $|\mathbb{F}|$, and hence this is only efficient over small finite fields. Over general (potentially exponentially large size) finite fields, efficient reconstruction algorithms were only known for $k = 2$ ([Sinha, ITCS 2022]); and over $\mathbb{R}$ and $\mathbb{C}$, they were only known for $k = 2$ ([Sinha, CCC 2016]) and $k = 3$ ([Saraf–Shringi, CCC 2025]).

1 Introduction

Arithmetic circuits are directed acyclic graphs (DAGs) used to represent multivariate polynomials succinctly. They generate polynomials from input variables by repeatedly applying addition ($+$) and multiplication ($\times$) operations.

The *reconstruction* problem for arithmetic circuits asks the following: given only black-box (or oracle) access to a polynomial computed by an unknown circuit C of size s from some circuit class $\mathcal{C}$, design an efficient (deterministic or randomized) algorithm that outputs another circuit computing the same polynomial. This can be viewed as the algebraic counterpart of exact learning in Boolean

*Department of Mathematics & Department of Computer Science, University of Toronto, Toronto, Canada. Research partially supported by an NSERC Discovery Grant and a McLean Award. Email: shubhangi.saraf@utoronto.ca

†Department of Computer Science, University of Toronto, Toronto, Canada. Research partially supported by an NSERC Discovery Grant and a McLean Award. Email: devansh@cs.toronto.edu

‡Department of Mathematics, University of Toronto, Toronto, Canada. Research partially supported by an NSERC Discovery Grant and a McLean Award. Email: narmada.varadarajan@mail.utoronto.ca

ISSN 1433-8092

circuit complexity [Ang88]. When we additionally require that the reconstructed circuit must lie in the same class $\mathcal{C}$ as the original, the task is referred to as *proper learning*.

Reconstructing arithmetic circuits is a central yet notoriously challenging problem. Recent years have seen substantial progress, with numerous works developing reconstruction algorithms for several restricted but natural subclasses of arithmetic circuits [BBB+00, KS01, KS06, FS12, GKQ14, KNST17, KNS19, GKS20, CGK+24].

Depth-reduction results [AV08, Koi10, Tav13, GKKS13] have shown that even low-depth circuits—notably depth-3 and depth-4 models—are remarkably powerful. Consequently, efficient reconstruction even for depth-3 circuits would have far-reaching implications. However, despite sustained efforts, we remain far from a complete understanding of general depth-3 reconstruction. Still, significant progress has been made for restricted subclasses of depth-3 ($\Sigma\Pi\Sigma$) and depth-4 ($\Sigma\Pi\Sigma\Pi$) circuits [Shp07, KS09a, BSV21, PSV24, BS25, GKL12, BSV20, Sin16b, Sin22, SS25, KS19, BGKS22].

A closely related problem is *black-box polynomial identity testing (PIT)*. Here, given black-box access to a polynomial f computed by a circuit C of size s from a class $\mathcal{C}$, the goal is to determine whether f is identically zero. Equivalently, the task is to construct an explicit hitting set for $\mathcal{C}$.[1]

It is straightforward to observe that deterministic reconstruction for a circuit class $\mathcal{C}$ is at least as hard as derandomizing black-box PIT for $\mathcal{C}$. Even randomized reconstruction typically demands a deep understanding of the structure of $\mathcal{C}$, and in most known cases, appears to be strictly harder than derandomizing PIT. In fact, for nearly every circuit class studied so far, efficient PIT algorithms have preceded progress on reconstruction algorithms. Since this work focuses on reconstruction for depth-3 circuits with constant top fan-in (i.e., $\Sigma\Pi\Sigma(k)$ circuits), we begin by surveying what is known about PIT for this model.

PIT for $\Sigma\Pi\Sigma(k)$ circuits A recent breakthrough by [LST22] established the first subexponential-time deterministic black-box PIT for $\Sigma\Pi\Sigma$ circuits (and more generally, constant-depth circuits). However, truly polynomial-time derandomization is only known for restricted subclasses of depth-3 circuits. When the top fan-in of the output gate is bounded by a constant k, the model is referred to as $\Sigma\Pi\Sigma(k)$ *circuits*. This class has been extensively studied in the context of black-box PIT, with to a sequence of works culminating in polynomial-time algorithms [DS05, KS08, KS09b, SS13, SS11].

A unifying idea in many of these results is that $\Sigma\Pi\Sigma(k)$ circuits that compute the zero polynomial must exhibit strong algebraic structure—specifically, they must be *low-rank*. This insight led to deep connections with discrete geometry, particularly variants of the Sylvester–Gallai theorem. Inspired by this, several recent works [Shp19, PS22a, PS21, PS22b, GOS22, OS22, GOPS23, OS24, DDS21] have recently attempted to lift these techniques to restricted depth-4 circuits, such as $\Sigma^k\Pi\Sigma\Pi^r$ circuits (i.e., depth-4 circuits with bounded top and bottom fan-in).

Reconstruction for $\Sigma\Pi\Sigma(k)$ circuits Despite all this progress on our understanding of the structure of identically zero $\Sigma\Pi\Sigma(k)$ circuits and $\Sigma^k\Pi\Sigma\Pi^r$ circuits, reconstruction algorithms for these models have still proved to be nearly intractable. Here is an overview of what was known in the case of infinite or exponentially large fields. Until recently, the only known subexponential reconstruction algorithms $\mathbb{R}$ and $\mathbb{C}$ held for $k \leq 3$. Even for $k = 2$, the problem is still very challenging and, despite attracting a lot of interest, was only resolved in the last few years by the works of Sinha [Sin16b] (over $\mathbb{R}$ and $\mathbb{C}$), and [Sin22](over all finite fields, including exponentially large finite fields). This was extended to $k = 3$ in the work of Saraf and Shringi [SS25] (over $\mathbb{R}$ and $\mathbb{C}$) and needed several new ideas and techniques.

[1] With randomness, this can be solved easily using the Schwartz–Zippel lemma [Sch80, Zip79]

Since reconstruction for general values of k seemed more challenging, much attention focused on interesting restricted submodels such as powering, multilinear and set-multilinear $\Sigma\Pi\Sigma(k)$ circuits; the last few years have seen some exciting progress on reconstruction algorithms in these settings [BBB+00, Shp07, KS09a, BSV21, PSV24, BS25]. Among these, powering circuits $\Sigma \wedge \Sigma(k)$, and set-multilinear $\Sigma\Pi\Sigma(k)$ have attracted particular interest due to their close connections with symmetric tensor decomposition and tensor decomposition problems. The focus of our paper will be on *general* $\Sigma\Pi\Sigma(k)$ circuits with no restrictions on structure.

The minimum k for a homogeneous polynomial such that it can be computed by a homogeneous $\Sigma\Pi\Sigma(k)$ circuit is known as the *Chow Rank* of the polynomial. It is a generalization of complexity measures like Tensor rank and Waring rank and has been studied in various settings [Lan15, BCC+18, DGI+24].

In the setting of small finite fields, one can exhaustively search over all field elements. This makes algorithm design easier in several settings, and indeed this allowed [Shp07, KS09a] to obtain an efficient reconstruction algorithm (which depended quasi-polynomially on field size) for $\Sigma\Pi\Sigma(k)$ circuits. Even though exhaustive search is possible, these results are highly non-trivial, with several beautiful ideas and algorithms developed, which enabled and influenced the design of several of the known reconstruction algorithms.

Our result (informal): *Given black-box access to an n-variate degree-d polynomial f over $\mathbb{R}$,$\mathbb{C}$, or a finite field $\mathbb{F}$ with $\mathrm{char}(\mathbb{F})$ greater than d, computed by a $\Sigma\Pi\Sigma(k)$ 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 if the field is $\mathbb{R}$ or $\mathbb{C}$, or it is $\log|\mathbb{F}|$ if the field is finite.*

Some definitions $\Sigma\Pi\Sigma(k)$ circuits are arithmetic circuits of depth 3 with top fan-in k. These circuits have three layers of alternating Σ and Π gates, and compute a polynomial of the form

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

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

For the purpose of reconstruction and PIT, one can easily reduce to the homogeneous setting where all the d_i's are the same (see discussion in Lemma 3.2). Henceforth we will assume that our circuits are homogeneous.

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* or the *simple part* 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 linear forms.

1.1 Our Results

In this paper, we give the first subexponential time (in fact, quasipolynomial time) algorithm for reconstructing $\Sigma\Pi\Sigma(k)$ circuits over $\mathbb{R}$, $\mathbb{C}$, or any finite field with $\mathrm{char}(\mathbb{F})$ greater than d. When the k multiplication gates in our circuit are sufficiently *distant*—$\forall i, j \in [k], i \neq j, \mathrm{rank}(\mathrm{sim}(T_i + T_j)) \geq c_1 \log^{c_2} d$ for some absolute constant c_1, c_2—then our algorithm does *proper learning*, i.e. its output is the unique $\Sigma\Pi\Sigma(k)$ circuit computing f. If this distance property does not hold, then our algorithm still outputs a circuit computing f, but from a slightly more general class—it computes a generalized depth-3 circuit of top fan-in at most $k-1$. The running time in the statement supresses a $\mathrm{poly}(s)$ dependence on the max bit complexity s of any constant appearing in the circuit C if the underlying field is $\mathbb{R}$ and $\mathbb{C}$, and a $\mathrm{poly}\log(|\mathbb{F}|)$ factor if $\mathbb{F}$ is a finite field.

Here is the formal statement of our main result.

Theorem 1.1. *Let $\mathbb{F}$ be a field that is a finite field with $\mathrm{char}(\mathbb{F})$ greater than d, or $\mathbb{R}$ or $\mathbb{C}$. Let $k \in \mathbb{N}^+$. Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ be a degree-d polynomial computed by $\Sigma\Pi\Sigma(k)$ circuit of the form $C = T_1 + T_2 + \cdots + T_k$. There exist constants $c_1, c_2 > 0$ (depending on k) such that the following holds: There is a randomized algorithm that runs in $(nd)^{O(\log d)^{O(1)}}$ time, makes black-box queries to f, and with probability $1 - o(1)$ does the following:*

1. *If $\forall i, j \in [k], i \neq j, \mathrm{rank}(\mathrm{sim}(T_i + T_j)) \geq c_1(\log d)^{c_2}$, then it outputs a $\Sigma\Pi\Sigma(k)$ circuit computing f.*
2. *If $\exists i, j \in [k], i \neq j$, such that $\mathrm{rank}(\mathrm{sim}(T_i + T_j)) < c_1(\log d)^{c_2}$ then it outputs a $\Sigma\Pi\Sigma(k-1, d, c_1(\log d)^{c_2})$ generalized depth-3 circuit computing f.*

Remark 1.2 (Dependence on bit complexity and field size)**.** *Over fields $\mathbb{R}$ or $\mathbb{C}$, if s is the maximum bit complexity of any coefficient appearing in C, then our algorithm's run time also depends quasipolynomially on s. In the statement of the above theorem and later in the paper, we have suppressed the $\mathrm{quasipoly}(s)$ dependence in the running time for clarity of exposition. Over finite fields, our algorithm needs query access over a $\mathrm{poly}(d)$ degree extension of $\mathbb{F}$, and has a run time that also depends quasipolynomially on $\log |\mathbb{F}|$.*

Remark 1.3 (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 [Shp07, KS09a, Sin16b, Sin22, SS25] also had a similar kind of output.*

Remark 1.4 (Field Characteristic)**.** *The requirement of $\mathrm{char}(\mathbb{F}) > d$ for finite fields comes from using Carlini's algorithm ([Car06]). We use it only when there are gates in the circuit that have low rank. In cases where the rank of each gate in the $\Sigma\Pi\Sigma(k)$ circuit is at least $c_1(\log d)^{c_2}$, we can do reconstruction even if the characteristic is small. For more details, check Remark 3.13.*

Outline of the paper In Section 2, we give a high-level overview of our proof. Section 3 comprises preliminaries, such as rank bounds for identically zero circuits and solving systems of polynomial equations. In Section 4, we bound the number of 'nice' spaces V such that the circuit vanishes on V (*vanishing spaces*). Section 5 consists of several technical structural results about the circuit and its vanishing spaces, which aid us in algorithmically computing a large class of vanishing spaces in Section 6. Finally, in Section 7, we show how to use these computed vanishing spaces to learn some of the linear forms in the circuit C, and in Section 8 we complete the proof of our theorem by reconstructing the circuit from these linear forms.

2 Proof Overview

Let f be an n-variate degree-d polynomial that has a $\Sigma\Pi\Sigma(k)$ representation and let

$$C = T_1 + T_2 + \cdots + T_k$$

be a $\Sigma\Pi\Sigma(k)$ circuit computing f. As discussed in the preliminaries, after a simple preprocessing step, we may assume that the circuit—and every gate within it—is homogeneous (see Lemma 3.2), so that each T_i is a product of exactly d linear forms. In general, the gates T_i may share a nontrivial greatest common divisor which will affect our proof methods. One might hope to reduce to the case $\gcd(T_1, T_2, \ldots, T_k) = 1$ by factoring out the common linear factors, but this is not possible: there may be linear factors that divide some T_i but are not part of the gcd, and dividing by such factors might destroy the property that the polynomial is computable by a $\Sigma\Pi\Sigma(k)$ circuit. However, for the sake of this proof overview, let us assume that $\gcd(T_1, T_2, \ldots, T_k) = 1$.

Our goal is to learn all of the linear forms in the circuit C, so we will first attempt to learn *some* of the linear forms that appear in one of the gates of C. In order to do this, we will try and analyze the linear spaces of low codimension on which the polynomial f vanishes. A space V is a *vanishing space* for f if f vanishes identically on V. If we write $V = \mathbb{V}(l_1, \ldots, l_r) = \{l_1, \ldots, l_r\}^{\perp}$, we say $\mathrm{span}(l_1, \ldots, l_r)$ is a *vanishing kernel* for f since $f \mod \langle l_1, \ldots, l_r \rangle \equiv 0$. It turns out that these spaces encode valuable information about the polynomial f. We will show that if we can somehow "learn" these spaces—or even enough information about them—then we can learn some linear forms appearing in C. Once we can learn enough (some large polynomial in $\log d$) linear forms in a single gate, then by combining results developed in past works [Shp07, KS09a] with ideas from the theory of locally-decodable codes, we can recover the entire circuit.

Our high-level approach, at this generality, is not so different from that of [Sin16b, Sin22, SS25]. The works of [Sin16b, Sin22] attempted to learn codimension 1 and 2 vanishing spaces for circuits of fan-in 2, and the work of [SS25] attempted to learn codimension 1, 2 and 3 vanishing spaces for circuits of fan-in 3. To learn these spaces, these prior works had to prove several highly nontrivial results about their structure.

In this work, we will need to understand and learn vanishing spaces up to codimension k for circuits of top fan-in k (where k might be an arbitrarily large constant). For this, we prove much more general structural theorems about vanishing spaces. Interestingly, given our improved understanding, our structural results and algorithms for learning the spaces are a lot less ad-hoc, and do not involve an elaborate case-by-case analysis as in the work of [SS25]. Nevertheless, the final algorithms and their analysis are still fairly intricate.

As a first step to learning vanishing spaces, we need to be able to show that their number is bounded. Unfortunately, this is not true. However, we will show how to identify a large and important subset of them, and show that that subset has bounded size. We will then attempt to learn a large class of these vanishing spaces, and use the spaces we have learned to determine some of the linear forms appearing in the circuit. Once we are able to learn a few of the linear forms, we have already made serious progress, and learning the rest can be done recursively.

Here is a brief outline of our reconstruction algorithm and its proof of correctness. After the brief outline, we will elaborate on some of the steps in an attempt to give a clearer high-level picture of what is going on.

1. For each $r \leq k$, we attempt to bound the number of maximal vanishing spaces $\mathcal{S}_r(f)$ of codimension r. In general, this may not be finite. We instead define a large important subclass of vanishing spaces—for each $r \leq k$, $\mathcal{S}'_r(f)$ is the set of those vanishing spaces whose kernels do not contain a subspace such that going modulo the subspace will crash the rank

of the circuit by too much. We formally define this class and show how to bound its size in Section 4.

2. We would like to algorithmically compute $\mathcal{S}'_r(f)$. We show how to do this (or something close to this) in Section 6. This is the most intricate and challenging part of our analysis. Instead of computing all of $\mathcal{S}'_r(f)$ (which we do not know how to do), we show how to learn (for many values of r) a large subset of $\mathcal{S}'_r(f)$ that we call $\mathcal{S}^*_r(f)$. We elaborate more on this step in the next part of the proof overview. Additionally, we look at the spaces in $\mathcal{S}^*_r(f)$ and show how to algorithmically extend them to get additional vanishing spaces (which may not be maximal anymore) but which have interesting properties such that some individual gates vanish on those spaces. We call this richer set of vanishing spaces $\overline{\mathcal{S}}_r(f)$.

3. We consider intersections of spaces in $\mathcal{S}^*_r(f)$ and $\overline{\mathcal{S}}_r(f)$ and show that these have enough information to recover many individual linear forms in C. The details of this step appear in Section 7. This works as long as the linear forms in each gate span a high rank subspace. If they do not, then we use Carlini's algorithm [Car06] to compute suitable vanishing spaces of the subset of gates that have high rank. We then can again learn enough linear forms from one of the high rank gates.

4. Reconstruct the entirety of the circuit using the few linear forms learned in the previous part. This part of the proof is very similar to what was done in prior works. These ideas first appeared in the works of Shpilka [Shp07] and Karnin–Shpilka [KS09a] and then were also used in Saraf–Shringi [SS25]. The details appear in Section 8.

2.1 Vanishing spaces and bounds on the number of them

Note that if for each $i \in [k]$, l_i is a linear form dividing T_i, then the polynomial f vanishes identically modulo $\mathrm{span}(l_1, \ldots, l_k)$. This gives us the linear subspace $\mathbb{V}(l_1, l_2, \ldots, l_k)$ of codimension at most k on which f vanishes, and the *kernel* of this space $\mathrm{span}(l_1, \ldots, l_k)$.

For any $r \in [k]$, we will let $\mathcal{V}_r$ be the set of codimension r spaces on which f vanishes. Note that all vanishing spaces need not be of the above form (determined by linear forms in the circuit), and this is one reason why understanding vanishing spaces is challenging. Now, if we could learn *all* spaces of codimension up to k on which f vanishes, then we would also learn $\mathbb{V}(l_1, l_2, \ldots, l_k)$ (and hence also $\mathrm{span}(l_1, \ldots, l_k)$) for any tuple $(l_1, \ldots, l_k)$ with l_i dividing T_i. And if we learn these spaces for each such k-tuple, then using suitable intersections, we could start recovering linear forms in the circuit.

The situation turns out to be far from so simple. One significant obstacle is that, in general, we cannot even bound the number of spaces in $\mathcal{V}_r$. There are two reasons for this. The first is that if there is any codimension $r-1$ space on which f vanishes, then any codimension r space contained in the codimension $r-1$ space will be a vanishing space (and thus there can be infinitely many of them over infinite fields). This is not a serious issue: let us just define $\mathcal{S}_r$ to be the *maximal* codimension r spaces on which f vanishes. The other more serious issue is a subtle one. We are only able to bound even the number of $\mathcal{S}_2$ spaces when the circuit C is *high-rank*—the linear forms appearing in the circuit span sufficiently high dimension. Now, suppose that there is some $(r-2)$-dimensional space $\mathrm{span}(l_1, \ldots, l_{r-2})$ such that the rank of $C \mod \langle l_1, \ldots, l_{r-2} \rangle$ *crashes*. That is, the original circuit $C = \sum T_i$ has gcd 1 and its linear forms span a high-dimensional space, but the circuit $C \mod \langle l_1, \ldots, l_{r-2} \rangle$ suddenly has a high gcd and the *simple part* $\mathrm{sim}(C \mod \langle l_1, \ldots, l_{r-2} \rangle))$ has low rank. We call any such space a *crashing space*, and call the set of all crashing spaces of dimension $r-2$ to be $\mathscr{C}_{\leq r-2}$. Since the circuit $C \mod \langle l_1, \ldots, l_{r-2} \rangle$ is low rank, it could have infinitely many

codimension-2 vanishing spaces, resulting in infinitely many codimension-r vanishing spaces for the original circuit C.

To deal with this issue, we define the set $\mathcal{S}'_r(f)$ to be those spaces $V \in \mathcal{S}_r(f)$ such that the kernel of V does not contain any $(r-2)$-dimensional crashing subspace V' (i.e. V' does not lie in $\mathscr{C}_{\leq r-2}$).

Bounding the size of $\mathcal{S}'_r(f)$ The starting point in our proof is to use ideas from discrete geometry and rank bounds for identically zero $\Sigma\Pi\Sigma(k)$ circuits to show that, for each $r \leq k$, the size of $\mathcal{S}'_r(f)$ is polynomially bounded. A beautiful sequence of works [DS05, KS08, KS09b, SS11, SS13] uses techniques from incidence geometry to prove that the rank of the simple part of any identically zero $\Sigma\Pi\Sigma(k)$ must be bounded (see Theorem 3.5 for a precise statement).

Observe that if $\mathbb{V}(l_1, l_2, \ldots, l_r) \in \mathcal{S}'_r(f)$, then $C \mod \langle l_1, l_2, \ldots, l_r \rangle$ is identically zero, and hence we can invoke Theorem 3.5. Let $V = \mathrm{span}(l_1, l_2, \ldots, l_r)$ with $\dim(V) = r$. If the original circuit $\mathrm{sim}(C)$ had high rank while $\mathrm{sim}(C \mod V)$ has low rank, then several linear forms in $\mathrm{sim}(C)$ must move into $\gcd(C \mod V)$. In other words, there must be several k-tuples $(l'_1, \ldots, l'_k)$, with each l'_i dividing T_i such that $\dim(\mathrm{span}(l'_1, \ldots, l'_k)) \geq 2$, but $\dim(\mathrm{span}(l'_1, \ldots, l'_k) \mod V) = 1$. This can only happen if V nontrivially intersects $\mathrm{span}(l'_1, \ldots, l'_k)$. Now, if this happens for several choices of $(l'_1, \ldots, l'_k)$, we show that these k-tuples determine a subspace $\overline{V}$ of V of some dimension r'. Since $\overline{V}$ is determined by a few k-tuples of linear forms from C, it follows that the number of possible choices of $\overline{V}$ is polynomially bounded. Now for any such $\overline{V}$, consider the circuit $C' = C \mod \overline{V}$. The ciruit C' vanishes modulo $(V \mod \overline{V})$. So, either $\dim \overline{V} = r-1$ and the number of extensions of $\overline{V}$ to V is bounded by d, the number of linear factors of C', or, since V does not contain crashing spaces of dimension $r-2$, C' is high-rank and we can apply induction to bound the number of possible spaces $(V \mod \overline{V})$. There are many details omitted in this overview; for example, we need to show that $(V \mod \overline{V})$ is actually in $\mathcal{S}_{r-r'}(C')$ to apply our induction hypothesis. The full details appear in Section 4.

Once we bound the size of $\mathcal{S}'_r(f)$, the next steps are to compute $\mathcal{S}'_r(f)$, and to show that learning $\mathcal{S}'_r(f)$ for various choices of r can help us learn linear forms appearing in C. Both these steps have their challenges, and we will try to elaborate on these challenges and how we overcome them in the discussion below.

2.2 Learning many of the vanishing spaces

As in previous works [SS25, Sin16b, Sin22], one way to learn vanishing spaces (assuming one can bound them) is to project the function f to only constantly many (or polylogarithmically many) variables, compute the vanishing spaces (here, $\mathcal{S}'_r$) for the low-variate polynomials by solving a suitable system of polynomial equations (now in few variables) for each projection, and then 'glue' or 'lift' the solutions to a global solution over the entire original space.

However, given the nature of the definition of $\mathcal{S}'_r(f)$ (vanishing spaces whose kernels do not contain any crashing spaces) there seems to be no way to encode its computation as a system of polynomial equations that is efficiently solvable. Indeed, the property of being a vanishing space is easy to encode, but that of being a vanishing space whose kernel does not contain a crashing space seems much harder. The way we get around this issue is to prove a structural result about crashing spaces that shows that they cannot be arbitrarily positioned.

Structure of crashing spaces We prove that there must exist (for each relevant r) a low-dimensional subspace $\mathcal{W}_r$ that intersects every crashing space of dimension up to $r-2$. We use lower bounds from locally-decodable codes (Similar arguments have been used in the past in related

settings for reconstruction.) to prove this fact. Essentially, the existence of any crashing space implies several linear dependencies among the linear forms appearing in the gates of C. We show that having too many independent crashing spaces would imply the existence of a 2-query locally-decodable code with parameters that we know cannot exist.

Learning a low-dimensional space that hits all crashing spaces We will show how to algorithmically learn such a space $\mathcal{W}_r$, which is perhaps one of the most intricate parts of our argument. To learn $\mathcal{W}_r$, the hope is that we first learn all crashing spaces of dimension up to $r-2$ and then pick a maximal independent subset of them whose span $\mathcal{W}_r$. However, learning crashing spaces of dimension up to $r-2$ (i.e. spaces in $\mathscr{C}_{\leq r-2}$) turns out to be difficult and something we are unable to do. Instead, here is our idealistic proof strategy. The actual details are much more involved, so we will later point out what goes wrong with our strategy and how we fix it.

Note that for $W \in \mathscr{C}_{\leq r-2}$, when we consider the circuit $C \mod W$, many of the linear forms from the gates of C must move into the gcd of $C \mod W$, resulting in a high-rank $\gcd(C \mod W)$, for this is the only way $\mathrm{sim}(C \mod W)$ can be so low rank. Now, for any linear form l dividing the gcd of $C \mod W$, the space $\mathrm{span}(W, l)$ is a *vanishing kernel* of dimension at most $r-1$ for C. If we can recursively show that for most choices of l the space $\mathrm{span}(W, l)$ can be learned (since we can recursively show that most vanishing spaces of codimension $r-1$ can be learned), then we would be well-positioned to learn the crashing space W. Perhaps the intersection of $\mathrm{span}(W, l_1)$ and $\mathrm{span}(W, l_2)$ for some choice of l_1 and l_2 (as long as the linear forms dividing the gcd span high enough dimension) will allow us to recover W.

The main thing that goes wrong with this proof strategy is that though $\mathrm{span}(W, l)$ is a vanishing kernel, it may not be a *minimal* vanishing kernel. Thus, though we may not be able to learn $\mathrm{span}(W, l)$, we will still be able to recursively learn a vanishing kernel that is a subspace of $\mathrm{span}(W, l)$. In order to learn vanishing kernels of dimension r, we assume we can learn most vanishing kernels of dimension up to $r-1$. We will also not be able to do this for all linear forms dividing the gcd of $C \mod W$ but for most linear forms, but let us not worry about this for now.

Once we learn minimal vanishing kernels contained within $\mathrm{span}(W, l)$ for most choices of l dividing gcd of $C \mod W$, we can still learn a large subspace $\overline{W}$ of W. While $\overline{W}$ may not 'crash' the rank as much as W does, we will show that $\overline{W}$ is still *crashing-like*.

Thus we will algorithmically be able to construct *crashing-like* subspaces of almost all crashing spaces in $\mathscr{C}_{\leq r-2}$. Our structure theorem for crashing spaces also extends to *crashing-like* spaces, and a maximal independent subset of them suffices in obtaining the space $\mathcal{W}_r$ that we seek. The space $\mathcal{W}_r$ will have: every $V \in \mathscr{C}_{\leq r-2}$ has nontrivial intersection with $\mathcal{W}_r$.

Learning $\mathcal{S}_r^*(f)$—a large subset of $\mathcal{S}_r'(f)$ Instead of learning all of $\mathcal{S}_r'$, our algorithm will only learn those spaces in $\mathcal{S}_r'$ whose kernels do not intersect $\mathcal{W}_r$; we call this set of spaces $\mathcal{S}_r^*(f)$. By the bound on the size of $\mathcal{S}_r'$, $\mathcal{S}_r^*$ is also bounded. Assuming that we have already computed $\mathcal{W}_r$, we can now encode the computation of spaces in $\mathcal{S}_r^*$ by a system of polynomial equations (albeit in n variables). In order to efficiently solve the system, we take several projections of the function f to only constantly many (or polylogarithmically many) variables, and compute the $\mathcal{S}_r^*(f)$ spaces for the low-variate polynomials by solving a suitable system of polynomial equations (now in few variables) for each projection. There are many details involved in successfully executing this process—for instance, we need to show that all properties that we need of $\mathcal{W}_r$ are consistent with projections, and this is somewhat subtle. We then show how to 'lift' the solutions for the projected spaces by gluing them together to form a global solution over the entire original space using ideas similar to those in past works such as [Sin22, SS25, Shp07, KS09a]. We finally compute the set

$\mathcal{S}_r^*(f)$.

Learning more general vanishing spaces—$\overline{\mathcal{S}}_r(f)$ The spaces in $\mathcal{S}_r^*$ have a lot of information, but to effectively use this information, we use $\mathcal{S}_r^*$ to first learn a richer collection of possibly non-maximal vanishing spaces, $\overline{\mathcal{S}}_r(f)$. The spaces in $\overline{\mathcal{S}}_r(f)$ are harder to define concisely, but they have some very useful and interesting properties that allow us to extract information about the circuit. One key property is the following. Consider any k-tuples $(l_1, \ldots, l_k)$ with l_i appearing in T_i such that $\text{span}(l_1, \ldots, l_k)$ is r-dimensional and does not have nontrivial intersection with $\mathcal{W}_r$. The space $\text{span}(l_1, \ldots, l_k)$ is the kernel of a vanishing space for C, but we cannot algorithmically learn it if the corresponding vanishing space is not maximal. However, we will still learn a subspace of $\text{span}(l_1, \ldots, l_k)$ as a kernel of some space in $\mathcal{S}_{r'}^*$ for some $r' \leq r$, but this subspace could be somewhat arbitrary. We will algorithmically and recursively show how to "grow" this subspace until we obtain a subspace that contains one of the linear forms $l_1, \ldots, l_k$. These "grown" subspaces will be the kernels of the vanishing spaces in the set $\overline{\mathcal{S}}_r$.

2.3 From vanishing spaces to linear forms

Armed with $\mathcal{W}_r$, $\mathcal{S}_r^*$, and $\overline{\mathcal{S}}_r$ for various choices of r, we then show that intersections of the kernels of spaces in $\overline{\mathcal{S}}_r$ (for various choices of r) suffice to learn several linear forms from at least one multiplication gate of C. This step uses the assumption that all gates of C have high rank.

Recall that by the properties of $\overline{\mathcal{S}}_r$, for any k-tuple $(l_1, \ldots, l_k)$ with l_i appearing in T_i such that $\text{span}(l_1, \ldots, l_k)$ is r-dimensional and does not have nontrivial intersection with $\mathcal{W}_r$, we can learn a subspace of this span (as a kernel of a space in $\overline{\mathcal{S}}_r$) which contains one of the l_i. Now, if every T_i is high rank that it follows that are a large number of such k-tuples such that we learn a subspace with one of the linear forms of the k-tuple; moreover, we can find many such k-tuples such that the subspaces learned correspond to the *same* linear form. In such a situation (if the k-tuples are independent enough), the intersections of these subspaces will let us algorithmically *learn* that linear form. Thus, the proof involves some clever counting along with basic linear algebra.

Handling the case when some gates have low rank The above procedure only works as long as the linear forms in each gate span a high-rank subspace. If they do not, then we use Carlini's algorithm [Car06] to (in some sense) reduce to the case of all gates having high rank.

Suppose that C is of the form $\sum_{i \in A} T_i + \sum_{i \in B} T_i$ where the indices in A correspond to the high-rank gates and those in B correspond to the low-rank gates. We then use Carlini's algorithm [Car06] to show how to set up a system of polynomial equations whose solutions correspond to the vanishing spaces of $\sum_{i \in A} T_i$, i.e. the vanishing spaces of just the high-rank gates. Let $C_A = \sum_{i \in A} T_i$ compute the polynomial f_A. Though we do not have black-box access to f_A, we will still be able to learn $\mathcal{W}_r$, $\mathcal{S}_r^*$ and $\overline{\mathcal{S}}_r$ spaces corresponding to f_A. (This is not entirely accurate—there are some complications, but this is the spirit of the argument.) Once we can do this, then using the previous analysis, we can again learn enough linear forms from one of the high-rank gates.

2.4 Few linear forms to the entire circuit

To go from a few ($\text{poly}(\log d)$) linear forms to the entire circuit, we use ideas appearing in previous works [Shp07, KS09a]. Once we have enough linear forms from one of the gates, we essentially go modulo these linear forms and recursively reconstruct the projections of $\Sigma\Pi\Sigma(k')$ circuits with $k' \leq k - 1$.

In [KS09a], the authors gave a way to convert a $\Sigma\Pi\Sigma(k)$ circuit into a $\Sigma\Pi\Sigma(s,d,r)$ generalized circuit ($s \leq k$), such that if the rank of the simple part of the sum of any two gates in the generalized circuit is high, it is a unique representation (Theorem 8.2).

We do proper learning if the initial gates of the circuit satisfy the distance properties of the clustering, i.e. after the clustering, each cluster is just one gate.

We go then mod the linear forms we learned earlier, and reconstruct the generalized circuit with smaller top fan-in recursively. With access to enough linear forms, we can learn $\text{poly}(\log d)$ many independent projections of one gate in the generalized circuit.

Then, using the technique of [Shp07](Theorem 3.16), one can combine the $O(\log d)$ independent projections of the gcd part of the gate in the generalized circuit to learn the gcd part exactly. Similarly, using $\text{poly}(\log d)$ projections and techniques of [KS09a] (Lemma 3.17), we can learn the sim part of the gate. Thus, we can learn one gate of the generalized circuit representation exactly.

We subtract it from the given circuit and learn the rest of the circuit recursively. At the end, we output the circuit after a randomized PIT check from Lemma 3.3 to ensure the output circuit computes the correct polynomial.

3 Preliminaries

Notation Let $\mathbb{N} := \{0, 1, 2, \dots\}$ and $\mathbb{N}^+ := \{1, 2, \dots\}$. Denote $\{1, 2, \dots, n\}$ by $[n]$. The cardinality of a set S is denoted by $|S|$. $\mathbb{F}$ is usually used to denote the underlying field, with $\text{char}(\mathbb{F})$ denoting its characteristic. $\mathbb{R}$ refers to the field of real numbers, and $\mathbb{C}$ refers to the field of complex numbers. Denote by $\log a$ the logarithm of a with base two.

Throughout the paper, we use uppercase letters X, Y to denote sets of variables, lowercase x_i to denote variables, $\mathbf{x}, \mathbf{y}$ or $\bar{x}, \bar{y}$ to denote vectors/tuples of variables, and $\mathbf{v}$ to denote a vector/tuple of field constants.

Whenever we say linear forms divide a multiplication gate, we mean up to scalar multiples. For a polynomial f, $\text{Lin}(f)$ denotes the multiset of linear factors of f (including multiplicities), and $\text{NonLin}(f)$ refers to $\frac{f}{\prod_{l \in \text{Lin}(f)} l}$. We use $\text{span}(l_1, \dots, l_r)$ to refer to the vector space that is the span of the linear forms $\{l_1, \dots, l_r\}$. In other words, it is the set of all vectors of the form $\sum_{i=1}^{r} \alpha_i l_i$ for $\alpha_i \in \mathbb{F}$. For a vector space $\mathbf{V}$, $\dim(\mathbf{V})$ denotes the dimension of $\mathbf{V}$.

Given k linearly independent linear forms $l_1, l_2, \cdots l_k$, let $\mathbb{V}(l_1, l_2, \dots, l_k) \subseteq \mathbb{F}^n$ 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 $\mathbb{V}(l_1, l_2, \dots, l_k)$ is a vanishing space for a polynomial f if f vanishes on all points of $\mathbb{V}(l_1, l_2, \dots, l_k)$. For a space V, $\mathcal{K}V$ or $Ker(V)$ are used to denote the kernel of V. For a set of spaces $\mathcal{S}$, $\mathcal{K}\mathcal{S}$ denotes the set of kernels of spaces in $\mathcal{S}$.

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 $x_1, x_2, \dots x_k$ to 0 in $\phi \cdot f$ results in the identically 0 polynomial. The polynomial $f \mod \langle l_1, \dots, l_k \rangle$ is equivalent (up to an invertible linear map) to $\phi \cdot f$ after setting $x_1, x_2, \dots x_k$ to 0. Often it is easier to think in terms of $\phi \cdot f$, and once we learn $\phi \cdot f$, one can recover f after applying the inverse linear map. We simplify notation and use $f \mod l$ (and $C \mod l$ where C is a circuit computing f) to denote $f \mod \langle l \rangle$ (or $C \mod \langle l \rangle$) for a linear form l. let V be a r-dimensional vector space spanned by linear forms $l_1, \dots, l_r$(any basis), then we might also use $f \mod V$ (and $C \mod V$) to denote $f \mod \langle l_1, \dots, l_r \rangle$ (and $C \mod \langle l_1, \dots, l_r \rangle$).

Remark 3.1. *For most of the paper, we will assume* $\text{char}(\mathbb{F})$ *is either 0 or greater than d. We will also assume the field* $\mathbb{F}$ *to be of large size, i.e.* $|\mathbb{F}| \geq d^n$*. If the field is small, we consider the circuit to be over an extension of size* d^n*. Due to uniqueness, results up to scaling of linear factors*

in Theorem 3.5 and Theorem 8.2, the output circuit will have constants in $\mathbb{F}$. This only changes the bit complexity by $n \log d$.

3.1 Depth-3 Circuits

In this section, we formally introduce the general model of depth-3 circuits which is the focus of our paper.

Definition 1. *A depth-3 $\Sigma\Pi\Sigma(k)$ circuit C computes a polynomial of the form*

$$C(X) = \sum_{i=1}^{k} T_i(X) = \sum_{i=1}^{k} \prod_{j=1}^{d_i} l_{i,j}(X),$$

where the $l_{i,j}$'s are linear functions; $l_{i,j}(X) = \sum_{t=1}^{n} a_{i,j}^t x_t + a_{i,j}^0$ with $a_{i,j}^t \in \mathbb{F}$.

We say that C is minimal *if no strict subset of the multiplication gates sums to zero. We define* $\gcd(C)$ *as the linear product of all the non-constant linear functions that belong to all the T_i's. I.e.* $\gcd(C) = \gcd(T_1, \ldots, T_k)$. *We say that C is* simple *if* $\gcd(C) = 1$. *The* simplification *or the* simple part *of C, denoted by* $\mathrm{sim}(C)$, *is defined as* $C/\gcd(C)$. *In other words,* $\mathrm{sim}(C)$ *is the circuit resulting upon the removal of all the linear functions that appears in* $\gcd(C)$.

Definition 2 (Homogeneous Depth-3 circuit)**.** *A depth 3 circuit $\Sigma\Pi\Sigma(k)$ computing a polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ is a* homogeneous depth-3 circuit $\Sigma\Pi\Sigma(k)$ *if f is homogeneous and the polynomial computed in every gate of the circuit is homogeneous as well. It will have the following form*

$$C(X) = \sum_{i=1}^{k} T_i(X) = \sum_{i=1}^{k} \prod_{j=1}^{d} l_{i,j}(X),$$

where the $l_{i,j}$'s are linear functions; $l_{i,j}(X) = \sum_{t=1}^{n} a_{i,j}^t x_t + a_{i,j}^0$ with $a_{i,j}^t \in \mathbb{F}$ and $a_{i,j}^0 = 0$.

Definition 3 (Rank of a circuit)**.** *The* rank *of a circuit $C(X) = \sum_{i=1}^{k} T_i(X) = \sum_{i=1}^{k} \prod_{j=1}^{d_i} l_{i,j}(X)$ is defined 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 denote it by* $\mathrm{rank}(C)$.

Definition 4 (Rank of the simple part of a circuit)**.** *The* rank *of the simple part of the circuit $C(X) = \sum_{i=1}^{k} T_i(X) = \sum_{i=1}^{k} \prod_{j=1}^{d_i} l_{i,j}(X)$ is defined as the rank of the simple part (obtained after removing the gcd of T_i's). We will denote the simple rank of C using $\Delta(C) = \mathrm{rank}(\mathrm{sim}(C))$. This also defines a distance measure between 2 circuits C_1, C_2 as $\Delta(C_1, C_2) = rank(sim(C_1 + C_2))$.*

In the following lemma from [SS25, Sin16a], it was shown that reconstruction of any $\Sigma\Pi\Sigma(k)$ circuit could be reduced to reconstruction of homogenous $\Sigma\Pi\Sigma(k)$ circuits with $\mathrm{poly}(n, d)$ overhead. Therefore, from now on, we are only concerned with the reconstruction of $\Sigma\Pi\Sigma(k)$ circuits in this paper and all $\Sigma\Pi\Sigma(k)$ circuits we consider will be assume to be homogeneous.

Lemma 3.2 (Section 1.5, [Sin16a];Lemma 3.1,[SS25])**.** *Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ be a degree-d polynomial computed by a $\Sigma\Pi\Sigma(k)$ circuit C. Then in time* $\mathrm{poly}(n, d)$ *(per query), one can simulate a black-box access to a homogeneous $\Sigma\Pi\Sigma(k)$ circuit computing a homogeneous $f^{hom} \in \mathbb{F}[x_1, \ldots, x_n, z]$, such that any reconstruction algorithm for f^{hom} immediately implies a reconstruction algorithm for f, with only a* $\mathrm{poly}(n, d)$ *overhead in time complexity.*

3.2 Generalized Depth-3 circuits

Definition 5. *A* generalized depth-3 circuit $\Sigma\Pi\Sigma(k,d,r)$ *is of the form*

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

where l_{ij},$\bar{l}_{ik}$ *are linear forms in* $\mathbb{F}[x_1, \ldots, x_n]$ *and* $d = max_i(d_i + deg(h_i))$.

In particular, in the setting where 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 a few (i.e. r) linear forms. Without loss of generality we can also assume that the h_i's have no linear factors.

3.3 Polynomial Identity Testing and Rank Bounds

Lemma 3.3 (Schwartz-Zippel Lemma,[Sch80, Zip79])**.** *Let* $f \in \mathbb{F}[x_1, \ldots, x_n]$ *be a polynomial of total degree* d *that is not identically zero. Let* $S \subset \mathbb{F}$ *be any finite set. For* $s_1, \ldots, s_n$ *chosen independently and uniformly at random from* S*,*

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

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 [Han65, BE67] over the real numbers and in [BDWY13, DSW14] over $\mathbb{C}$. We define the *rank* of a set of vectors to be the dimension of the linear space they span.

Definition 6 ($\mathrm{SG}_k(\mathbb{F}, m)$)**.** *Let* S *be a set of non-zero vectors in* $\mathbb{F}^{n+1}$ *such that 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* $\mathrm{SG}_k(\mathbb{F}, m)$*.*

It is known that $\mathrm{SG}_k(\mathbb{R}, m) = 2(k-1)$ [Han65, BE67]. The rank of high-dimensional Sylvester-Gallai configurations over $\mathbb{C}$ was bounded by 2^{c^k} for a fixed constant c in [BDWY13]. This bound was further improved to $\mathrm{SG}_k(\mathbb{C}, m) = c^k$ (for a fixed constant c) in [DSW14]. In Theorem 7 of [SS13], it was shown that for any field $\mathbb{F}$, $\mathrm{SG}_k(\mathbb{F}, m) \leq 9k \log m$, while for $\mathbb{F} = \mathbb{F}_p$, we know constructions such that $\mathrm{SG}_k(\mathbb{F}_p, m) = \Omega(k \log_p m)$.

The polynomial time black-box PIT algorithms for $\Sigma\Pi\Sigma(k)$ circuits follow from some strong structural properties of identically zero $\Sigma\Pi\Sigma(k)$ circuits. In [KS09b] 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 [SS11, SS13], and the theorem below gives the best bound we know.

Theorem 3.4 ([SS13])**.** *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 $\mathrm{SG}_k(\mathbb{R}, m)$, $\mathrm{SG}_k(\mathbb{C}, m)$ and $\mathrm{SG}_k(\mathbb{F}_q, m)$ we obtain the following:

Theorem 3.5. *Let C be a simple, minimal and identically zero $\Sigma\Pi\Sigma(k)$ circuit over $\mathbb{R}$, $\mathbb{C}$ or $\mathbb{F}_q$. Then there is an absolute constant $\mathcal{R}_{\mathbb{F}}(k,d)$ depending only on k,d such that $\text{rank}(C) < \mathcal{R}_{\mathbb{F}}(k,d)$. If C is over $\mathbb{R}$ then we can bound $\text{rank}(C)$ by $3k^2$. If C is over $\mathbb{C}$ then we can bound $\text{rank}(C)$ by $2k^2 + k \cdot c^k$ for some absolute constant c. If C is over $\mathbb{F}_q$, then we can bound $\text{rank}(C)$ by $9k \log d$.*

We represent the rank bound for simple, minimal generalized $\Sigma\Pi\Sigma(k,d,\rho)$ circuit by $\mathcal{R}_{\mathbb{F}}(k,d,\rho)$.

Lemma 3.6 (Lemma 4.2,[KS08])**.** *Let C be a simple and minimal $\Sigma\Pi\Sigma(k,d,\rho)$ circuit in n indeterminates computing the zero polynomial. Then $\text{rank}(C) < \mathcal{R}_{\mathbb{F}}(k,d,\rho) = \mathcal{R}_{\mathbb{F}}(k,d) + k \cdot \rho$.*

3.4 Other Known Results

Theorem 3.7 (Effective Hilbert irreducibility, Theorem 1.1[KSS14])**.** *Let $S \subseteq \mathbb{F}$ be a finite set and $g(X, A_1, \ldots, A_n)$ a monic polynomial in X of total degree at most d. If g is irreducible then it holds that*

$$\mathbb{P}_{\boldsymbol{\alpha},\boldsymbol{\beta}}[g(X, \alpha_1 T + \beta_1, \ldots, \alpha_n T + \beta_n) \text{ is not irreducible}] < O(d^5/|S|),$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ are chosen uniformly and independently from S^n.

Lemma 3.8 (Black-box multivariate polynomial interpolation, [BOT88, KS01])**.** *Let n, m, d be parameters and $\mathbb{F}$ be a field that is $\mathbb{R}$, $\mathbb{C}$, or a large enough finite field. There exists a deterministic algorithm that runs in time $(nmd)^{O(1)}$, and outputs a set S of points in $\mathbb{F}^n$, such that given black-box access to any degree-d polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ with at most m monomials, the coefficients of all monomials can be recovered in $(nmd)^{O(1)}$ time using evaluations from the set $\{f(s) : s \in S\}$.*

Lemma 3.9 (Black-box Factoring, [KT90])**.** *There exists a randomized algorithm that takes as input black-box access to a degree-d, n-variate polynomial f with coefficients in some field $\mathbb{F}$, runs in time $\text{poly}(nd)$ 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,*

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

Using the above, we can also decompose any circuit into its linear factors (which we can interpolate) and $\text{NonLin}(f)$ in randomized $\text{poly}(n,d)$ time.

3.5 Solving a System of Polynomial Equations

We obtain the vanishing spaces of our circuit by solving a system of polynomial equations. The problem of solving a system of polynomial equations is generally considered to be difficult, even undecidable for certain fields. A longer discussion on the complexity of finding a single solution to a system of polynomial equations for various fields can be found in [BSV21].

The system of polynomial equations we will solve in this paper will have an extra condition that the number of solutions is finite. In this case, the problem can be solved in time that is exponential in number of variables for various fields as described below. We will also need to find all possible solutions of the system that we set up (instead of a single solution), and in order to do this, we show that the number of solutions is finite, and in particular polynomially bounded. Note that once we can find a single solution, then by iteratively adding additional equations, we can find all solutions.

In this work, the polynomial systems we solve have a small ($O(1)$) number of variables, and hence once can find solutions efficiently. The theorem we state below is a variant of an analogous one that appears in [BSV21], and it describes the current known upper bounds for solving a system of polynomial equations for various fields.

Let $\overline{\mathbb{F}}$ denote the algebraic closure of $\mathbb{F}$.

Theorem 3.10. *Let $f_1, f_2, \ldots f_m \in \mathbb{F}[x_1, \ldots, x_n]$ be n-variate polynomials of degree at most d. Suppose that the system of equations $f_1(x) = 0, \ldots, f_m(x) = 0$ has N solutions in the algebraic closure of $\mathbb{F}$, where N is finite. Then, the complexity of finding all the solutions in an appropriate extension is as follows:*

1. *[GVJ88] For $\mathbb{F} = \mathbb{R}$, there is a deterministic $\text{poly}((mdN)^{n^2})$ time algorithm. Here the authors assumed that the constants appearing in the system are integers (or rationals). Note that for all computational applications we can WLOG assume this by simply approximating/truncating a given real number at some number of bits.*
2. *[Ier89] For $\mathbb{F} = \mathbb{C}$ (or any algebraically closed field), there is a deterministic $(mn)^{O(n)} \cdot (dN)^{O(n^2)}$ time algorithm.*
3. *[Laz01] For $\mathbb{F} = \mathbb{F}_q$, there is a randomized $(mdnN)^{\text{poly}(n)}$ time algorithm that computes solutions in a degree N extension of $\mathbb{F}$.*

Thus, in randomized time $(mdnN)^{\text{poly}(n)}$, we can find all the solutions of $f_1(x) = 0, \ldots, f_m(x) = 0$ if it has N solutions in the algebraic closure of $\mathbb{F}$.

Remark 3.11. *In the results used above, we have suppressed a $\text{poly}(s)$ multiplicative dependence in the running time, where s is the maximum bit complexity of any coefficient appearing in the input circuit ($\log |\mathbb{F}|$ for finite fields). We use the above algorithm only in cases where n is $\text{poly}(\log d)$ and the number of solutions $N = \text{poly}(d)$, and hence the solutions are only in $\text{poly}(d)$ degree extensions and there is an additional $\text{quasipoly}(s, d)$ running time factor, which we suppress throughout the paper.*

3.6 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. We start by defining essential variables in a polynomial.

Definition 7 (Essential variables, [Kay11])**.** *The* number of essential variables *in $f(x_1, \ldots, x_n)$ is the smallest t for which there exists an invertible linear transformation $A \in \mathbb{F}^{(n \times n)}$ such that $f(A \cdot \overline{x})$ depends on only t variables.*

The number of *redundant variables* is the number of essential variables subtracted from n. We will use the following result from [Car06] that allows us to compute t, the number of essential variables, and the linear transformation A.

Theorem 3.12 ([Car06],[Kay11])**.** *Let n, d be positive integers and $\mathbb{F}$ be a field with $\text{char}(\mathbb{F}) > d$ or $= 0$. There is a randomized algorithm that takes as input black-box access to an n-variate degree-d polynomial $f(x) \in \mathbb{F}[\bar{x}]$ with t essential variables, runs in time $(nd)^{O(1)}$, and outputs an invertible matrix $A \in \mathbb{F}^{(n \times n)}$ such that $f(A \cdot \bar{x})$ depends only on the first t variables.*

Remark 3.13. *Theorem 3.12 is the only tool we use that requires the characteristic of the underlying field to be at either 0 or greater than d. This is used in Lemma 7.2 and Lemma 8.1, both of which contribute to Theorem 1.1 requiring $\text{char}(\mathbb{F})$ to be 0 or greater than d.*

The partial derivative $\partial_i f$ is used to represent $\frac{\partial f}{\partial x_i}$. We use ∂f to denote $(\partial_1 f, \ldots, \partial_n f)$. We define the partial derivative matrix of a polynomial f, $M(f)$, as the matrix with columns indexed by monomials over n variables and degree $d-1$, while the rows are indexed by $[n]$, and $M_{i,j} = \text{coeff}_j(\partial_i f)$ where $\text{coeff}_j(g)$ is the coefficient of monomial j (represented as vector) in g.

We denote $\partial f^{\perp}$ as the set of vectors $\mathbf{a} \in \mathbb{F}^n$ such that $\mathbf{a} \cdot \partial f = 0$. The proof of the above theorem relies on the following lemma which describes the relation between the partial derivative matrix and the number of essential variables.

Lemma 3.14 ([Car06], Lemma B.1[Kay11]). *The number of redundant variables in a polynomial $f(x_1, \ldots, x_n)$ equals the dimension of $\partial f^{\perp}$. In particular, the number of essential variables of f is the rank of the partial derivative matrix $M(f)$.*

The following lemma from [Shp07] will also be useful to us.

Lemma 3.15 (Lemma 23, [Shp07]). *Let $f(x_1, \ldots, x_n) \in \mathbb{F}[x_1, \ldots, x_n]$ be a polynomial with k essential variables and two different representations: $f = g(l_1, \ldots, l_k) = h(l'_1, \ldots, l'_k)$ for polynomials $g, h \in \mathbb{F}[y_1, \ldots, y_k]$, and linear forms $l_1, \ldots, l_k, l'_1, \ldots, l'_k \in \mathbb{F}[x_1, \ldots, x_n]$. Then,* $\mathrm{span}(\{l_i\}_{i\in[k]}) = \mathrm{span}(\{l'_i\}_{i\in[k]})$.

3.7 Gluing Projections

Using lower bounds for locally-decodable codes in [Shp07], the authors gave an algorithm that could learn a product of linear forms exactly with multiplicities if given access to $\Omega(\log d)$ independent non-zero projections of the product. This is summarized in the theorem below.

Theorem 3.16 (Implicit in [Shp07]). *Let L be a multiset containing d linear functions in n variables. Let $\{\varphi_1, \ldots, \varphi_m\}$ be a set of linearly independent linear functions such that $m \geq 100 \log(d)$. For each $j \in [m]$ define the multiset*

$$L_j \triangleq \{l \mod \varphi_j : l \in L\}.$$

Then there exists a deterministic algorithm that, given $\{L_j\}_{j=1}^m$, outputs L in $\mathrm{poly}(n, d)$ *time.*

In [KS09a], the authors gave a way to similarly glue projections of gates in a generalized circuit. The two key ingredients were the theorem above, and the following lemma below that allows one to glue the projections of low-rank polynomials.

Lemma 3.17 (Special case of Lemma 4.20 in [KS09a]). *Let h be a non-zero n-variate polynomial of degree d with r essential variables. Let l_1, l_2 be two independent linear forms in $\mathbb{F}[x_1, \ldots, x_n]$ such that $h \mod \langle l_1, l_2 \rangle$ also has r essential variables. Then, there exists a deterministic algorithm which when given as the input the two polynomials $\{h \mod l_1, h \mod l_2\}$, outputs a representation of h as a polynomial of r linear functions in $O(n \cdot d^r)$ time.*

4 Upper bounding the size of $\mathcal{S}_r(f)$

Let f be an n-variate degree-d polynomial in $\mathbb{F}[x_1, \ldots, x_n]$ computed by a $\Sigma\Pi\Sigma(k)$ circuit $C = T_1 + T_2 + \cdots + T_k$.

Definition 8 (Vanishing Spaces). *A codimension-r space $\mathbb{V}(l_1, \ldots, l_r)$ is a* vanishing space *for a polynomial f if $f \mod \langle l_1, \ldots, l_r \rangle = 0$. The vanishing space is* maximal *if it is not properly contained in any vanishing spaces, i.e. for any $q < r$ linear forms $l'_1, \ldots, l'_q \in \langle l_1, \ldots, l_r \rangle$, $f \mod \langle l'_1, \ldots, l'_q \rangle \neq 0$.*

We define $\mathcal{V}_r(f)$ to be the set of all codimension-r vanishing spaces of f, and $\mathcal{S}_r(f)$ to be the set of all maximal codimension-r vanishing spaces of f.

$$\mathcal{S}_r(f) = \{V : V \in \mathcal{V}_r(f) \text{ is maximal } \}.$$

For most of our proofs, we will need the actual linear forms $l_1, \ldots, l_r$ instead of the codimension-r subspace, so we define the *r-kernel* of the polynomial f as

$$\mathcal{KV}_r(f) = \{\text{span}(l_1, \ldots, l_r) : \mathbb{V}(l_1, \ldots, l_r) \in \mathcal{V}_r(f)\},$$

and the *minimal r-kernel*

$$\mathcal{KS}_r(f) = \{\text{span}(l_1, \ldots, l_r) : \mathbb{V}(l_1, \ldots, l_r) \in \mathcal{S}_r(f)\}.$$

We will refer to a space $V \in \mathcal{KS}_r(f)$ as a *minimal vanishing kernel.*

Definition 9. *Let f be a polynomial computed by a $\Sigma\Pi\Sigma(k)$ circuit C, and $c_q = \mathcal{R}_{\mathbb{F}}(k, d) + 2q^2$. A q-dimensional space* $\text{span}(l_1, \ldots, l_q)$ *is a* crashing space *of f if*

(i) $\text{rank}(\text{sim}(C \mod \langle l_1, \ldots, l_q\rangle)) \leq c_{q+1}$*;*

(ii) $C \mod \langle l_1, \ldots, l_q\rangle \neq 0$*;*

(iii) and $C \mod \langle l_1, \ldots, l_q\rangle$ *is not a product of linear forms.*

A crashing space is minimal *if it does not contain any other proper crashing spaces.*

Define

$$\mathscr{C}_q = \{\textit{minimal } q\textit{-dimensional crashing spaces of } f\}.$$

We can only bound and compute those kernels that do not contain any crashing spaces, so we define

$$\mathcal{S}'_r(f) = \{V \in \mathcal{S}_r(f) : \ker(V) \text{ does not contain any space in } \mathscr{C}_q(f) \text{ for all } q \leq r-2\},$$

and

$$\mathcal{KS}'_r(f) = \{V \in \mathcal{KS}_r(f) : V \text{ does not contain any space in } \mathscr{C}_q(f) \text{ for all } q \leq r-2\}.$$

Theorem 4.1. *Let f be a polynomial computed by a degree-d $\Sigma\Pi\Sigma(k)$ circuit C. For any $r \leq k$, let $c_r = \mathcal{R}_{\mathbb{F}}(k, d) + 2r^2$. If* $\text{rank}(\text{sim}(C)) \geq c_r$*, then*

$$|\mathcal{S}'_r(f)| = O(d^{2kr^2})$$

Proof. We will show this using induction on r and k. First, for any k, $|\mathcal{KS}_1(f)| \leq d = O(d^{2k})$ since each linear form generating a $\mathcal{KS}_1$ space must be a factor of f. Assume for our induction hypothesis that for some k and some $r \geq 2$, we know that for all $h \leq k$ and $q \leq r-1$, $|\mathcal{KS}'_q| = O(d^{2kq^2})$. Our proof is divided in two cases.

Case 1. *We bound the number of $V = \text{span}(l_1, \ldots, l_r) \in \mathcal{KS}'_r(f)$ such that some gate T_i vanishes modulo $\langle V\rangle$.*

Suppose without loss of generality that $T_i = T_1$. There must be some linear form $l \in V$ that divides T_1; extend this to a basis $\{l, l'_2, \ldots, l'_r\}$ for V. Define $V' = \text{span}\{l'_2, \ldots, l'_r\}$. Then, the circuit $C' = T_2 + \cdots + T_k \mod \langle l\rangle$ is nonzero by the minimality of V, and $C' \mod V' = C \mod V = 0$. This shows that V' is a vanishing kernel for C'. We want to apply our induction hypothesis to C' and bound the number of possible extensions of V' to V.

First, we need to show that $V' \in \mathcal{KS}'_{r-1}(C')$. If V' contains a proper vanishing kernel $U' \subset V'$ such that $C' \mod U' = 0$, then $\text{span}\{U', l\}$ is a proper subspace of V such that $C \mod \text{span}\{U', l\} = 0$, contradicting the minimality of V. So, V' is a minimal vanishing $(r-1)$-kernel for C'. Similarly, suppose V' contains a q-dimensional crashing space $W \subseteq V'$ such that $\text{rank}(\text{sim}(C' \mod W)) \leq c_{q+1}$, for some $q \leq r-3$. Then, $\text{span}\{W, l\} \subseteq V$ is a $(q+1)$-dimensional crashing space for C, since $\text{rank}(\text{sim}(C \mod \text{span}\{W, l\})) = \text{rank}(\text{sim}(C' \mod W)) \leq c_{q+1} \leq c_{q+2}$, contradicting that V does not contain any crashing spaces of dimension at most $r-2$. So, $V' \in \mathcal{KS}_{r-1}(C')$.

If $r = 2$, then V' is a $\mathcal{KS}_1$ space, i.e. it is generated by a linear factor of the circuit C', so there are at most d choices of V'. And there are at most d choices of the linear form l as a factor of the gate T_1, so there are at most $d^2 = O(d^{2kr^2})$ choices for the space $V = \text{span}\{V', l\}$.

If $r = \dim V \geq 3$, then the circuit C' satisfies our high-rank hypothesis since

$$\text{rank}(C') = \text{rank}(\text{sim}(T_2 + \cdots + T_k) \mod \langle l \rangle) = \text{rank}(\text{sim}(C \mod \langle l \rangle) \geq c_{r-1},$$

where the last inequality holds since $\text{span}\{l\} \subseteq V$ cannot be a crashing space.

From the induction hypothesis, there are at most $O(d^{2k(r-1)^2})$ choices for $\text{span}\{l'_2, \ldots, l'_r\}$ from $\mathcal{KS}'_{r-1}(f \mod l)$, and there are at most d choices for a divisor l of T_1. Therefore, in this case there are at most $O(d^{2kr^2})$ possibilities for $V \in \mathcal{KS}_r(f)$.

Case 2. *We bound the number of $V = \text{span}\{l_1, \ldots, l_r\} \in \mathcal{KS}'_r(f)$ such that no gate T_i vanishes modulo V.*

First, suppose the circuit has the form $C = G \times (T_1 + \cdots + T_k)$ where $G = \gcd(C)$ and $T_1 + \cdots + T_k = \text{sim}(C)$. Define

$$\mathcal{L}(T_i) = \{l : l \text{ is a linear form dividing } T_i\};$$
$$\mathcal{L}(\text{sim}(C)) = \bigcup_{i=1}^{k} \mathcal{L}(T_i).$$

Since C vanishes modulo V, the rank bounds in Theorem 3.5 imply that $\text{rank}(C \mod V) < \mathcal{R}_{\mathbb{F}}(k, d)$. For each $i = 1, \ldots, k$, define

$$A_i = \{l \in \mathcal{L}(T_i) : l \text{ divides } \gcd(C \mod V)\},$$
$$B_i = \mathcal{L}(T_i) \setminus A_i.$$

Since V is r-dimensional, $\text{rank}\{l \mod V \; l \in B_1 \cup \cdots \cup B_k\} \geq \text{rank}\{l : l \in B_1 \cup \cdots \cup B_k\} - r$, so $\text{rank}(A_1 \cup \cdots \cup A_r) = \text{rank}\{l \in \mathcal{L}(\text{sim}(C)) : l \text{ divides } \gcd(C \mod V)\} \geq \text{rank}(\text{sim}(C)) - r - \mathcal{R}_{\mathbb{F}}(k, d) \geq 3r + 1$. We will actually prove something stronger: we will show by induction that the number of r-dimensional spaces V such that $\text{rank}(C \mod V) < \mathcal{R}_{\mathbb{F}}(k, d)$ and no gate T_i vanishes mod V is at most $O(d^{2kr})$..

For the base case of $r = 1$, we want to bound the number of linear forms l such that $\text{rank}(C \mod l) < \mathcal{R}_{\mathbb{F}}(k, d)$ and l does not divide any gate of C. Since $\text{rank}(\text{sim}(C)) \geq \mathcal{R}_{\mathbb{F}}(k, d) + 2$, there are at least 2 independent linear forms $l_1, l_2 \in \mathcal{L}(C)$ that divide $\gcd(C \mod l)$. That is, for each $j = 1, 2$, there are k linear forms $l_{1,j} \in \mathcal{L}(T_1), \ldots, l_{k,j} \in \mathcal{L}(T_k)$ and scalars $\lambda_{1,j}, \ldots, \lambda_{k,j}, \alpha_{1,j}, \ldots, \alpha_{k,j}$ such that

$$l_{i,j} = \alpha_{i,j} l + \lambda_{i,j} l_j.$$

So, for $j = 1, 2$, the vector space

$$V_j = \text{span}\{l_{1,j}, \ldots, l_{k,j}\} = \text{span}\{l, l_j\}$$

has dimension 2. The intersection $V_1 \cap V_2 = \text{span}\{l\}$, so l is determined by at most 2 linear forms from each of $\mathcal{L}(T_1), \ldots, \mathcal{L}(T_k)$, so there are at most d^{2k} possibilities for l.

For the inductive step, suppose we know that for all $q < r$, the number of q-dimensional spaces $U \in \mathcal{KS}'_q$ such that $\text{rank}(C \mod U) < \mathcal{R}_{\mathbb{F}}(k, d)$ and no gate T_i vanishes modulo U is $O(d^{2kq^2})$. To bound the number of r-dimensional spaces $V \in \mathcal{KS}'_r$ such that $\text{rank}(C \mod V) < \mathcal{R}_{\mathbb{F}}(k, d)$ and no gate vanishes mod V, we consider two cases: the case when V contains a $(r-1)$-dimensional subspace W such that $\text{rank}(C \mod W) < \mathcal{R}_{\mathbb{F}}(k, d)$, and the case where it does not. Recall by definition of $\mathcal{KS}'_r$, V cannot contain any crashing spaces of dimension $\leq r-2$.

Case 2a. *We bound the number of spaces V that contain some $(r-1)$-dimensional subspace W such that* $\text{rank}(C \mod W) < \mathcal{R}_{\mathbb{F}}(k, d)$.

This proof is similar to the proof of case 1. Let $V = \text{span}\{W, l\}$ for some linear form l. Then, the circuit $C' = C \mod W$ is nonzero by the minimality of V, but $C' \mod l = 0$, so l is a factor of C'. No gate of the original circuit C vanishes modulo W, so the number of choices for W is $O(d^{2k(r-1)^2})$ by the induction hypothesis applied to C and W. Then, we have at most d choices for the factor l of C', which implies we have $O(d^{2kr^2})$ choices for the space V.

Case 2b. *We bound the number of spaces V such that for any $(r-1)$-dimensional subspace $W \subseteq V$,* $\text{rank}(C \mod W) \geq \mathcal{R}_{\mathbb{F}}(k, d)$.

This proof uses a linear algebra trick similar to the base case. We know that least $2r+1$ independent linear forms collapse into $\gcd(C \mod V)$, so let $l_1, \ldots, l_c$, for some $c \geq 2r+1$, be independent linear forms that divide $\gcd(C \mod V)$. Again, for each $j \in [c]$, we have linear forms $l_{i,j} \in \mathcal{L}(T_i), i = 1, \ldots, k$ such that

$$l_{i,j} \mod V = \lambda_{i,j} l_j$$

for some scalars $\lambda_{i,j}$. Define

$$V_j = \text{span}\{l_{1,j}, \ldots, l_{k,j}\}$$

so $\dim(V_j \cap V) = \dim(V_j) - 1$.

Note that $\dim V_j \geq 2$, otherwise the linear forms $l_{1,j}, \ldots, l_{k,j}$ would all belong to the gcd of C. Our goal is to use a subset of the spaces $\{V_1, \ldots, V_c\}$ to learn a subspace U' of V.

We say a set of spaces $V_1, \ldots, V_a$ is *independent* if for every $i \in [a]$, $V_i \notin \text{span}\{V_1, \ldots, V_{i-1}, V_{i+1}, \ldots, V_a\}$. Suppose (relabelling the spaces if necessary) $V_1, \ldots, V_a$ is a maximal independent set of spaces in $\{V_j : j \in [c]\}$. This implies that $\text{span}\{V_1, \ldots, V_a\} = \text{span}\{V_j : j \in [c]\}$. Since $\dim(V_i \mod V) = 1$ for al $i \in [a]$, we know that

$$\dim \text{span}\{V_1, \ldots, V_a\} = \dim\left(\text{span}\{V_1 \mod V, \ldots, V_a \mod V\} \cup V\right) \leq a + r.$$

On the other hand,

$$\dim \text{span}\{V_1, \ldots, V_a\} = \dim \text{span}\{V_j : j \in [c]\} \geq 3r + 1,$$

so $a \geq 2r+1$. Thus, we can find r independent spaces $V_1, \ldots, V_r$.

Now, by Lemma 4.2, for some $m \leq r$, the space $U' = \text{span}\{V_1, \ldots, V_{m-1}\} \cap V_m$ is nonzero and contained in V. There are d^k choices for each of the spaces $V_1, \ldots, V_m$, so there are at most d^{kr} choices for the space U'. If $U' = V$, this shows there are at most $O(d^{2kr^2})$ choices for V.

Suppose U' is a proper subspace of V. Consider the circuit $C' = C \mod U'$ and two cases, one where $\dim(U') = r-1$, and the other where $\dim(U') \leq r-2$. In the first case, if U' is $(r-1)$-dimensional then the linear form $V \mod U'$ divides the circuit C', so we have at most d possible extensions of U' to V, and therefore at most $d \cdot d^{kr} = O(d^{2kr})$ choices for C.

In the second case, suppose $1 \le \dim U' = q \le r-2$. We know that V does not contain any crashing spaces of dimension $\le r-2$, so $\text{rank}(\text{sim}(C')) \ge c_{r-1}$. We will now argue that $V \mod U'$ is in fact in $\mathcal{KS}_{r-q}(C')$. First, $V \mod U'$ is a vanishing kernel for $C' = C \mod U'$ since V is a vanishing kernel for C. Next, if $V \mod U'$ contains a proper subspace $V' \mod U'$ that is vanishing for C', then V' is a proper subspace of V that is vanishing for C, contradicting the minimality of V. Finally, if $V \mod U'$ contains a subspace $V' \mod U'$ that is crashing for C', then V' is a subspace of V that is crashing for C, which is again a contradiction. This all shows that $V \mod U' \in \mathcal{KS}_{r-q}(C')$. Recall that $r-q \le r-1$, so by the induction hypothesis, there are at most $O(d^{2k(r-1)^2})$ choices for $V \mod U'$. So, there are at most $O(d^{2k(r-1)^2}) \cdot d^{kr} = O(d^{2kr^2})$ choices for the space V. □

Definition 10. *We say a collection of spaces $V_1, \dots, V_a$ is* independent *if for every $i \in [a]$, $V_i \notin \text{span}\{V_1, \dots, V_{i-1}, V_{i+1}, \dots, V_a\}$.*

Lemma 4.2. *Let $V_1, \dots, V_a$ be a set of independent spaces[2], and let U be an r-dimensional space, for some $1 < r < a$, such that $\dim(U \cap V_j) = \dim(V_j) - 1$ for all $j = 1, \dots, a$. Then, for some $i \le r$, the space $\text{span}(V_1, \dots, V_i) \cap V_{i+1} \subseteq U$ and $\dim(\text{span}(V_1, \dots, V_i) \cap V_{i+1}) \ge 1$.*

Proof. Define $U_i = U \cap V_i$ for $i = 1, \dots, a$. Choose the maximum i such that the spaces $U_1, \dots, U_i$ are independent. Note that since U is r-dimensional, the maximum such i is at most r, which is strictly less than a. We will show that $\text{span}\{V_1, \dots, V_i\} \cap V_{i+1} \subseteq U$ and $\dim(\text{span}\{V_1, \dots, V_i\} \cap V_{i+1}) \ge 1$.

For each j, since $V_j \mod U$ is 1-dimensional, choose a nonzero vector w_j so that $V_j = \text{span}(w_j, U_j)$. Suppose for contradiction that $\text{span}\{V_1, \dots, V_i\} \cap V_{i+1} \not\subseteq U$. Equivalently, $\text{span}(w_1, \dots, w_i) \cap V_{i+1} \neq \{0\}$. So, we can choose a nonzero vector $z_{i+1} \in (\text{span}(w_1, \dots, w_i) \cap V_{i+1}) \setminus U$.

By the maximality of i, $U_{i+1} \subseteq \text{span}\{U_1, \dots, U_{i-1}\}$. Since $V_{i+1} \not\subseteq \text{span}\{V_1, \dots, V_i\}$ by independence, it follows that $w_{i+1} \notin \text{span}(V_1, \dots, V_i)$, which implies that $w_{i+1} \notin \text{span}(U_{i+1}, z_{i+1})$.

Finally, this tells us that $\text{span}(U_{i+1}, w_{i+1}, z_{i+1}) \subseteq V_{i+1}$ and $\dim(V_{i+1}) \ge \dim(U_{i+1}) + 2 = \dim(V_{i+1}) + 1$, a contradiction. □

5 Structural results about vanishing spaces

In this section we prove a variety of useful structural results on vanishing spaces. Many of these statements are generalizations of similar statements that appeared in [SS25] in the setting of $k = 3$. The proofs are a simple variation of those when $k = 3$, but we include them for completeness.

The first structural result that will be useful in our reconstruction is the following lemma, which bounds the dimension of the set of *rank-reducing* linear forms. The proof of Claim 4.8 in [DS05] implies this result. It can also be inferred as a special case of Lemma B.1 (restated with better notation in section B.1.2) in [KS09a], with $A = [k]$, $\hat{r} = \text{rank}(sim(T_1 + T_2 + \cdots + T_k))$, $r_t = r$, and $\chi = \left\lfloor \frac{\hat{r}}{2r' \log d} \right\rfloor$. We omit the proofs when they can be found in the referenced papers.

Lemma 5.1 (Implied from Claim 4.8,[DS05])**.** *Let C be a $\Sigma\Pi\Sigma(k)$ circuit of the form $T_1 + T_2 + \cdots + T_k$ such that $\gcd(T_1, \dots, T_k) = 1$. Fix $r' > 0$ to be any constant such that $\text{rank}(\text{sim}(T_1 + T_2 + \cdots + T_k)) > 2r' \log d + 2k$. We define a linear form l to be rank-reducing if $\forall i \in [k]$ $l \nmid T_i$ and $rank(sim(C \mod l)) \le r'$. If we define a set of rank-reducing linear forms for C as*

$$\mathcal{L} := \{l : l \text{ is rank-reducing for } C\}$$

then $\dim(\text{span}(\mathcal{L})) \le max(r' \log d, 2k \log dk + 2k)$.

[2] As defined in the proof of case 2b of Theorem 4.1, a set of spaces $V_1, \dots, V_a$ is *independent* if for every $i \in [a]$, $V_i \notin \text{span}\{V_1, \dots, V_{i-1}, V_{i+1}, \dots, V_a\}$.

Using the above lemma along with rank bounds for $\Sigma\Pi\Sigma(k)$ circuits, we show how to bound the dimension of linear forms that divide f but do not divide the gcd G. This is a generalization of Lemma 6.4 from [SS25] which showed it for $k = 3$.

Lemma 5.2. *Let $\mathcal{R}_{\mathbb{F}}(k,d)$ be as defined in Theorem 3.5. Let $f \in \mathbb{F}[x_1,\ldots,x_n]$ be a degree-d polynomial computed by $\Sigma\Pi\Sigma(k)$ ($k \geq 2$) circuit of the form $C = G \times (T_1 + \cdots + T_k)$ such that $\gcd(T_1,\ldots,T_k) = 1$. Define*

$$\mathcal{L}_{s1} := \{l : \mathbb{V}(l) \in \mathcal{S}_1(f)\} = \{l : l|f\}.$$

Then, we have $\dim(\mathrm{span}(\mathcal{L}_{s1} \setminus \mathrm{Lin}(G))) \leq 2^{k^2}\mathcal{R}_{\mathbb{F}}(k,d)\log d$.

Proof. The only interesting case is when $\mathrm{rank}(T_1+\cdots+T_k) \geq 2^{k^2}\mathcal{R}_{\mathbb{F}}(k,d)\log d$, as the contribution of linear forms dividing $T_1 + \cdots + T_k$ has to be less than $\mathrm{rank}(T_1 + \cdots + T_k)$. So, assuming that $\mathrm{rank}(T_1 + \cdots + T_k) \geq 2^{k^2}\mathcal{R}_{\mathbb{F}}(k,d)\log d$, we will prove the statement by induction on k, where the base case is simply [SS25, Lemma 6.4].

For the induction hypothesis, suppose that for any $k' < k$ and f' computed by a $\Sigma\Pi\Sigma(k')$ circuit, we have $\dim(\mathrm{span}(\mathcal{L}_{s1}(f') \setminus \mathrm{Lin}(G))) \leq 2^{(k-1)^2}\mathcal{R}_{\mathbb{F}}(k-1,d)\log d$. Let l be any linear form that divides $T_1 + \cdots + T_k$. We divide our proof into two cases.

Case 1. $\forall i \in [k]$, $l \nmid T_i$.

As $f \equiv 0 \mod l$, the circuit $(C \mod l)$ computes the zero polynomial. Let $C \mod l$ be of the form $G' \times (T_1' + \cdots + T_k')$ with $\gcd(T_1',\ldots,T_k') = 1$. Note that $(T_1' + \cdots + T_k')$ is a simple circuit computing 0, and hence by Theorem 3.5, $\mathrm{rank}(T_1' + \cdots + T_k') < \mathcal{R}_{\mathbb{F}}(k,d)$. Let $\mathcal{L}'_{s1}$ be the set of linear forms in $\mathcal{L}_{s1}$ which do not divide any T_i. We can use Lemma 5.1 with $r' = \mathcal{R}_{\mathbb{F}}(k,d)$ and $\mathrm{rank}(\mathrm{sim}(T_1 + \cdots + T_k)) > 2\mathcal{R}_{\mathbb{F}}(k,d)\log d + 2k$, to obtain

$$\dim(\mathrm{span}(\mathcal{L}'_{s1})) \leq \max(\mathcal{R}_{\mathbb{F}}(k,d)\log d, 2k\log(kd) + 2k) \leq 2\mathcal{R}_{\mathbb{F}}(k,d)\log d + 2k.$$

Case 2. $\exists i \in [k]$ *such that* $l|T_i$.

Since we have $l \in \mathcal{L}_{s1} \setminus Lin(G)$, there must be a set of gates $K \subset [k]$ such that $l \nmid T_j, j \in K$. We also know $|K| \geq 2$. Now, since $l|(T_1 + \cdots + T_k)$ and $l|T_i, i \in [k] \setminus K$, we have $l|(\sum_{j\in K} T_j)$. By the induction hypothesis, l lies in a set of dimensions at most $2^{|K|^2}\mathcal{R}_{\mathbb{F}}(k,d)\log d$. The number of such subsets would be $\binom{k}{|K|}$. Therefore, all such l will lie in a space of dimension at most $\sum_{i=2}^{k-1}\binom{k}{i}2^{i^2}\mathcal{R}_{\mathbb{F}}(i,d)\log d \leq 2^{k^2}\mathcal{R}_{\mathbb{F}}(k,d)\log d$.

Combining the two cases we get that $\dim(\mathrm{span}(\mathcal{L}_{s1}) \setminus \mathrm{Lin}(G)) \leq 2^{k^2}\mathcal{R}_{\mathbb{F}}(k,d)\log d + 2k$.

□

A very interesting corollary of the above result is that for our reconstruction algorithm, we can essentially reduce to the case where the gcd of the gates in the input circuit is low-dimensional.

Corollary 5.3. *Let f be a n-variate degree-d polynomial computed by a $\Sigma\Pi\Sigma(k)$ circuit $C = G \times (T_1 + \cdots + T_k)$ and c_{cand} be any integer. Given black-box access to C, there exists an algorithm that runs in time $d^{O(\mathcal{R}_{\mathbb{F}}(k,d)\log d)}$ outputs a size $d^{O(\mathcal{R}_{\mathbb{F}}(k,d)\log d)}$-list $\mathcal{L}$ of pairs of polynomials (L, f') such that*

- *for each $(L, f') \in \mathcal{L}$, L is a product of linear forms and $L \cdot f' = f$, and*
- *for at least one of the $(L, f') \in \mathcal{L}$, f' can be computed by a $\Sigma\Pi\Sigma(k)$ circuit $C' = G' \times (T_1 + \cdots + T_k)$ with $\dim(\mathrm{span}(\mathcal{L}_{s1}(C'))) \leq 2^{k^2}\mathcal{R}_{\mathbb{F}}(k,d)\log d + 2k$.*

Note that the above corollary implies that we can assume for our reconstruction algorithm that $\dim(\text{span}(\mathcal{L}_{s1}(C))) \leq 2^{k^2}\mathcal{R}_{\mathbb{F}}(k,d)\log d + 2k$ for input $\Sigma\Pi\Sigma(k)$ circuits C. This is because we can run our reconstruction for each f' such that $(L, f') \in \mathcal{L}$, and once the reconstruction succeeds, we can verify the correct one using a PIT algorithm.

Proof. Using Lemma 3.9, we can get access to the set of linear factors $\mathcal{L}_{s1}$ of f in $\text{poly}(n,d)$ time with high probability. This set will contain linear factors of G and $\text{sim}(C)$. From Lemma 5.2, we know the linear factors of $\text{sim}(C)$ lie in a $\left(2^{k^2}\mathcal{R}_{\mathbb{F}}(k,d)\log d + 2k\right)$ -dimensional space. Therefore, we guess a set of up to $2^{k^2}\mathcal{R}_{\mathbb{F}}(k,d)\log d + 2k$ independent linear forms from $\mathcal{L}_{s1}$ and divide f by all linear forms in $\mathcal{L}_{s1}$ not in the span of the linear forms we guessed. For the correct guess, we will have guessed the space corresponding to linear factors of $\text{sim}(C)$ and therefore, all divisions will correspond to linear forms in $\text{Lin}(G)$.

The number of possible guesses is at most $d^{O(\mathcal{R}_{\mathbb{F}}(k,d)\log d)}$ and hence the running time and size of the list. □

5.1 Special and regular vanishing spaces

The vanishing spaces of the $\Sigma\Pi\Sigma(k)$ circuit, as we saw in Theorem 4.1, can be of two types: First, where some gate vanishes, and second, where the simple part of the circuit becomes identically 0 mod them without any gates vanishing. The first type of spaces (*Special spaces*) is useful to us as they give information about linear forms in the circuit, while the second (*Regular spaces*) interfere with us learning the first type of spaces. The actual definition of the spaces below in Definition 11 is a little more technical, which allows us to bound the structure of regular spaces.

Definition 11 (Regular and Special Spaces)**.** *We classify vanishing spaces in $\mathcal{V}_r(f)$ into two kinds, the* regular spaces

$$\mathcal{S}_r^{reg}(f) := \{\mathbb{V}(l_1,\ldots,l_r) \in \mathcal{V}_r(f) : \text{ For a random } W \in \text{span}(l_1,\ldots,l_r) \text{ with } \dim(W) = r-1 \text{ and } \text{span}(l') := (\text{span}(l_1,\ldots,l_r) \mod), l' | \text{sim}(C \mod W) \text{ with probability } 1 - o_{\mathbb{F}}(1)\},$$

and the special spaces

$$\mathcal{S}_r^{sp}(f) := \mathcal{V}_r \setminus \mathcal{S}_r^{reg}(f)$$

Observe that for $r = 1$, $\mathcal{S}_1^{reg} = \{\mathbb{V}(l) : l \in \mathcal{L}_{s1} \setminus Lin(G)\}$ and $\mathcal{S}_1^{sp} = \{\mathbb{V}(l) : l \in Lin(G)\}$.

Notice that Lemma 5.2 above shows that the kernels of spaces in $\mathcal{S}_1^{reg}$ span a low-dimensional space. We will prove a generalization of this result to higher-dimensional kernels of regular spaces. We first introduce a definition to make the statement formal.

Definition 12 (Totally independent regular set)**.** *We define a subset $V_r \subseteq \mathcal{S}_r^{reg}(f)$ to be a* totally independent *regular set if*

$$\dim\left(\text{span}\left(\{l_1,\ldots,l_r : \mathbb{V}(l_1,\ldots,l_r) \in V_r\}\right)\right) = r|V_r|,$$

i.e. the kernels of the spaces in W_r are totally independent.

We will argue in the following lemma that the size of any totally independent regular set W_r is small.

Lemma 5.4. *Let f be a n-variate degree-d polynomial over any field $\mathbb{F}$ such that $|\mathbb{F}| \geq O(d^5)$, computed by a circuit $C = G \times (T_1 + \cdots + T_k)$ such that $\gcd(T_1, \ldots, T_k) = 1$. Let $\mathcal{R}_{\mathbb{F}}(k, d)$ be as defined in Theorem 3.5. Then, for any totally independent regular set $W_r \subseteq \mathcal{S}_r^{reg}(f)$, where $r < k$, we have*

$$|W_r| \leq 2^{k^2} \mathcal{R}_{\mathbb{F}}(k, d) \log d$$

We will prove this using induction on r. The base case is for $r = 1$ which is in Lemma 5.2. We first state and prove a simple claim that will be useful for the proof. Note that this claim only requires a weaker assumption that the spaces are *independent*, that is, no space is in the span of the others.

Claim 5.5. *Let $S = \{U_1, U_2, \ldots U_b\}$ be any collection of independent r-dimensional subspaces of $\mathbb{F}^n$ (where $\mathbb{F}$ is an infinite or large enough field). For $i \in [b]$, let l_i be a linear form corresponding to a uniformly random vector sampled from U_i. Let $f \in \mathbb{F}[x_1 \ldots x_n]$ be a nonzero degree-d polynomial. Then, the probability that $f = 0 \mod \langle l_1, \ldots, l_b \rangle \leq bd/|\mathbb{F}|$.*

Proof. First observe that it suffices to prove the above result for $b = 1$ and then use induction to iterate this on the polynomials $f \mod \langle l_1 \rangle, f \mod \langle l_1, l_2 \rangle, \ldots, f \mod \langle l_1, \ldots, l_b \rangle$ (since $U_1, U_2, \ldots U_b$ are all independent) and apply the union bound. For $b = 1$, note that for any linear form l such that $f = 0 \mod \langle l \rangle$, l must be a linear factor of f. Since f can have at most d distinct linear factors (up to scaling), the result follows. □

We now prove Lemma 5.4

Proof of Lemma 5.4. Let $t = |W_r|$. Let $W_r = \{W_1, W_2, \ldots W_t\}$. Now each $W_i \in W_r$ is of the form $\mathbb{V}(l_{i1}, \ldots, l_{ir})$. Let $V_i = \text{span}\{v_{i1}, \ldots, v_{ir}\}$ where $v_{ij} \in \mathbb{F}^n$ is the vector corresponding to l_{ij}.

For each i, let l_i be the linear form corresponding to a uniformly random vector sampled from V_i. Let V_i' be a space of dimension $r - 1$ such that $\text{span}\{l_i, V_i'\} = V_i$.

Let $A = \{l_i : i \in [t]\}$.

Now, we consider the circuit $C' = C \mod \langle A \rangle$. Let $f' = f \mod \langle A \rangle$.

Let C' be of the form $C' = G' \times (T_1' + \cdots + T_k')$, where $\gcd(T_1', \ldots, T_k') = 1$. Observe that by Claim 5.5, $C' \neq 0$. Moreover each of $G', T_1', \ldots, T_k'$ compute nonzero polynomials. We will show that $V_1', V_2', \ldots, V_t'$ are all in $\mathcal{S}_{r-1}^{reg}(f')$ (in the space $\mathbb{V}(l_1, l_2, \ldots, l_k)$) i.e. are regular vanishing spaces of codimension $r - 1$. Then, by the induction hypothesis, the bound on t follows.

We first show that V_i' is a $\mathcal{S}_{r-1}^{reg}$ of $C \mod l_i$. As C vanishes mod $V_i = \text{span}(V_i', l_i)$, $C \mod l_i$ must vanish mod V_i'. Consider any random $r-2$ space V_i'' of V_i'. As we first chose l_i randomly from V_i, $\text{span}(V_i'', l_i)$ is a random codimension $r - 1$ space of V_i, and therefore for l_i' such that $l_i' = V_i' \mod V_i'' = V_i \mod \langle V_i'', l_i \rangle$, we have l_i' divides $\text{sim}(C \mod \langle l_i, V_i'' \rangle) = \text{sim}((C \mod l_i) \mod V_i'')$.

We would like to show that V_i' continues to be a regular codimension $r-1$ vanishing space of $C \mod l_i$, i.e. l_i' divides $\text{sim}(C \mod \langle A, V_i'' \rangle)$. Let $C \mod \langle l_i, V_i'' \rangle$ be of the form $G_i \times (T_{1i} + \cdots + T_{k'i})$ with $\gcd(T_{1i}, \ldots, T_{k'i}) = 1$ and $2 \leq k' \leq k$.

Let T_{1i} be a gate such that $l_i' \nmid T_{1i}$. Such a gate exists as $\gcd(T_{1i}, \ldots, T_{k'i}) = 1$. It suffices to show that no linear form dividing T_{1i} becomes equal to l_i' when we consider it over $\mathbb{V}(V_i'', l_1, l_2, \ldots, l_t)$. Consider any linear form l dividing T_{1i}. By assumption, $\text{span}\{l_i'\}$ cannot contain any factor of T_{1i}. Thus $l \notin \text{span}\{l_i'\}$ and hence it is of the form $\beta l_i' + l'$ where $l' \notin \text{span}\{l_i'\}$. By Claim 5.5 (applied to $\beta l_i' + l'$ but in the space $\mathbb{V}(l_i')$), l' remains nonzero with high probability when we go mod $l_1, l_2, \ldots, l_{i-1}, l_{i+1}, \ldots l_t$, and moreover is still not in $\text{span}\{l_i'\}$. □

5.2 Essential variables and rank

We now state a lemma which will eventually be useful in analyzing circuits where some gates have their linear forms only spanning a low dimensional space. This is a variant of Lemma 6.9 of [SS25], where a very similar result was shown for $k = 3$; the proof is essentially the same as well.

Lemma 5.6. *Let f be a homogeneous polynomial computed by a $\Sigma\Pi\Sigma(k)$ circuit $C = G \times (T_1 + \cdots + T_k)$ and $\gcd(T_1, \ldots, T_k) = 1$. Let $t > 0$ be any nonnegative integer such that $\forall i \in [k], \dim(\mathrm{span}(\mathrm{Lin}(G \times T_i))) \geq t$ and for any $r < k$ let $l_1, \ldots, l_r \in \mathbb{F}[x_1, \ldots, x_n]$ be arbitrary linear forms. $\mathcal{R}_{\mathbb{F}}(k, d)$ as defined in Theorem 3.5. Then,*

- *The number of essential variables (Definition 7) in $G \times (T_1 + \cdots + T_k)$ is at least $\frac{t - \mathcal{R}_{\mathbb{F}}(k+1,d)}{3}$.*
- *If $f \mod \langle l_1, \ldots, l_r \rangle$ has less than $\frac{t - \mathcal{R}_{\mathbb{F}}(k+1,d) - r}{4}$ essential variables, then $f \mod \langle l_1, \ldots, l_r \rangle = 0$*

Proof. We will first prove the first item. In the case where $\dim(\mathrm{span}(\mathrm{Lin}(G))) \geq \frac{t - \mathcal{R}_{\mathbb{F}}(k+1,d)}{3}$, then clearly we are done as the number of essential variables in $G \times (T_1 + \cdots + T_k)$ is at least $(t - \mathcal{R}_{\mathbb{F}}(k+1, d))/3$. Therefore, we assume $\dim(\mathrm{span}(\mathrm{Lin}(G))) < \frac{t - \mathcal{R}_{\mathbb{F}}(k+1,d)}{3}$ and hence $\forall i \in [k], \dim(\mathrm{span}(\mathrm{Lin}(T_i))) \geq \frac{2t + \mathcal{R}_{\mathbb{F}}(k+1,d)}{3}$. Let the number of essential variables in $T_1 + \cdots + T_k$ be c. Thus by definition $(T_1 + \cdots + T_k) = g(l_1, \ldots, l_c)$ for some homogeneous polynomial g in $\mathbb{F}[x_1, \ldots, x_c]$ and linear forms $l_1, \ldots, l_c$ in $\mathbb{F}[x_1, \ldots, x_n]$. Let z be a new variable and consider a random linear isomorphism Φ which for each $i \in [c]$ maps $l_i \to \alpha_i z$ for a random $\alpha_i \in \mathbb{F}$. Then with high probability g is nonzero and is of the form αz^d for some constant $\alpha \in \mathbb{F}$. Therefore, we have $\Phi(T_1 + \cdots + T_k) - \alpha z^d = 0$. Now we have a $\Sigma\Pi\Sigma(k)$ circuit equalling 0 and hence we can use rank bounds! By Theorem 3.5, $\mathrm{rank}(\mathrm{sim}(\Phi(T_1 + \cdots + T_k) - \alpha z^d)) \leq \mathcal{R}_{\mathbb{F}}(k, d)$.

We will first show that the linear forms contributing to the GCD have rank at most c.

Consider any two linear forms $l_a \in T_a, l_b \in T_b$ such that $\mathrm{span}(l_a) \neq \mathrm{span}(l_b)$ which we can find as $\gcd(T_1, \ldots, T_k) = 1$, and hence these linear forms do not contribute to the gcd already. Suppose after applying Φ, two distinct linear forms got "collapsed" to the same and moved into the gcd. In other words $\mathrm{span}(\Phi(l_a)) = \mathrm{span}(\Phi(l_b))$. We will now show that the only way this can happen is if $l_a, l_b \in \mathrm{span}(l_1, \ldots, l_c)$. This will then imply that the linear forms in the gcd have rank at most c.

Let $l_a = l'_a + \sum_{i=1}^{c} \beta_{a,i} l_i$ where $\beta_{a,i} \in \mathbb{F}$, and $l'_a = 0$ or $l'_a \notin \mathrm{span}(l_1, \ldots, l_c)$. Similarly $l_b = l'_b + \sum_{i=1}^{c} \beta_{b,i} l_i$ where $\beta_{b,i} \in \mathbb{F}$, and $l'_b = 0$ or $l'_b \notin \mathrm{span}(l_1, \ldots, l_c)$. We have $\Phi(l_a) = l'_a + \sum_{i=1}^{c} \beta_{a,i} \alpha_i z$ and $\Phi(l_b) = l'_b + \sum_{i=1}^{c} \beta_{b,i} \alpha_i z$.

Case 1: $\mathrm{span}(l'_a) \neq \mathrm{span}(l'_b)$. In this case $\Phi(l_a)$ and $\Phi(l_b)$ clearly remain independent.

Case 2: $\mathrm{span}(l'_a) = \mathrm{span}(l'_b)$. In case these spans are actually 0, then we are done. So let us assume the span is nonzero. In this case, without loss of generality assume the linear forms are scaled such that $l'_a = l'_b$. Then, since $\mathrm{span}(l_a) \neq \mathrm{span}(l_b)$, for some i, $\beta_{a,i} \neq \beta_{b,i}$. Hence with high probability $\sum_{i=1}^{c} \beta_{a,i} \alpha_i \neq \sum_{i=1}^{c} \beta_{b,i} \alpha_i$. Therefore $\Phi(l_a)$ and $\Phi(l_b)$ remain independent with high probability.

So, the only linear forms that move to the gcd of $\Phi(T_1), \ldots, \Phi(T_k)$ are the ones that lie in $\mathrm{span}(l_1, \ldots, l_c)$. Therefore, the linear forms that move into the gcd lie in a space of dimension at most c. Moreover, after applying Φ, the span of the linear forms from T_a that do not move to the gcd can get shrunk by at most c. Therefore, $\mathrm{rank}(\mathrm{sim}(\Phi(T_1 + \cdots + T_k) - \alpha z^d)) \geq 2t/3 - 2c$. By the rank bound, $(2t + \mathcal{R}_{\mathbb{F}}(k+1, d))/3 - 2c \leq \mathcal{R}_{\mathbb{F}}(k+1, d)$, which gives us $c \geq \frac{t - \mathcal{R}_{\mathbb{F}}(k+1,d)}{3}$. Thus, we finish the first part of the lemma.

We now show that if $f \mod \langle l_1, \ldots, l_r \rangle$ is non-zero then it must have a large number of essential variables, and we will show how to deduce this either from the gcd or from the simple part of the circuit.

$C \mod \langle l_1, \ldots, l_r \rangle$ is of the form $G' \times (T'_1 + \cdots + T'_k)$ with $\gcd(T'_1, \ldots, T'_k) = 1$ with at least some T'_i that are non-zero. Now, if $\dim(\text{span}(\text{Lin}(G'))) \geq \frac{t - \mathcal{R}_{\mathbb{F}}(k+1,d) - r}{4}$, then clearly the number of essential variables of $C \mod \langle l_1, \ldots, l_r \rangle$ is greater than or equal to $\frac{t - \mathcal{R}_{\mathbb{F}}(k+1,d) - r}{4}$ (except $C \mod \langle l_1, \ldots, l_r \rangle = 0$) since G' is a product of linear forms which will continue to have a high rank under any linear isomorphism. In the other case, when $\dim(\text{span}(\text{Lin}(G'))) < \frac{t - \mathcal{R}_{\mathbb{F}}(k+1,d) - r}{4}$, then for all nonzero T'_i, $\dim(\text{span}(\text{Lin}(T'_i))) \geq \frac{3t + \mathcal{R}_{\mathbb{F}}(k+1,d) - 3r}{4}$. Therefore, by part 1 of the current lemma, we see that $C \mod \langle l_1, \ldots, l_r \rangle$ has at least $\frac{t - \mathcal{R}_{\mathbb{F}}(k+1,d) - r}{4}$ essential variables. Therefore, if $C \mod \langle l_1, \ldots, l_r \rangle$ has less than $\frac{t - \mathcal{R}_{\mathbb{F}}(k+1,d) - r}{4}$ essential variables, then $C \mod \langle l_1, \ldots, l_r \rangle = 0$. □

5.3 μ-Cluster Representation

For a $\Sigma\Pi\Sigma(k)$ circuit C, recall that $\Delta(C) = \text{rank}(\text{sim}(C))$, and for two $\Sigma\Pi\Sigma(k)$ circuits C_1, C_2, $\Delta(C_1, C_2) = \text{rank}(\text{sim}(C_1 + C_2))$. Based on this notion of distance, we can cluster the gates close to each other together into a representation given by the following lemma.

Lemma 5.7. *For any μ and $\Sigma\Pi\Sigma(k)$ circuit $C = G \times (T_1 + \cdots + T_k)$, there is a partition of $[k]$ into nonempty sets $A_1, \ldots, A_s$ (where s depends on the circuit and μ) such that for $C_i = \sum_{j \in A_i} T_j$ and $C = G \times (C_1 + \cdots + C_s)$, we have $\forall i \neq j \in [s], \Delta(C_i, C_j) > \mu$ and $\forall i \in [s], \Delta(C_i) < 2^k \cdot \mu$.*

We call such a representation a *μ-cluster representation*.

Proof. We do this greedily. Start with k clusters, each containing a single gate and merge any two clusters into one if they do not satisfy $\Delta(C_i, C_j) > \mu$. Therefore, the greedy merge stops either if there is only 1 cluster remaining or the distance condition between clusters is satisfied for all clusters. At each merge for any cluster C_i, $\Delta(C_i)$ at most doubles. Also, after the first merge for any cluster C_i, $\Delta(C_i, C_j) \leq \mu$ and there are at most $k - 1$ merges. Therefore, we have $\forall i \in [s] \Delta(C_i) < 2^k \cdot \mu$. □

6 Algorithmically Computing Vanishing Spaces

In this section we will show how to compute a large and interesting subset of all vanishing spaces, which we will later use to compute linear forms from the circuit. Though we cannot even bound the size of $\mathcal{S}_r(f)$, we are able to bound the size of the set $\mathcal{S}'_r(f)$ (maximal vanishing spaces of codimension r which don't have crashing spaces of dimension $r-2$ in their kernels) in Theorem 4.1 assuming the circuit is high rank. The high-level goal will be to try and compute these spaces, and then use them to compute linear forms from the circuit. The way we learn these spaces is by encoding the property of being a vanishing space as the solution of a system of polynomial equations. However, in order to set up a polynomial system for computing $\mathcal{S}'_r(f)$ spaces, one has to first get a handle on how to compute *crashing spaces*, which is quite challenging.

The way we get around this challenge is to show that there exist low-dimensional spaces $\mathcal{W}_r$ (LDICR spaces, to be formally defined later) which intersect all crashing spaces of dimension at most $r-2$. Moreover, we can even algorithmically compute these spaces (this step is quite challenging). Thus, though we aren't even able to compute $\mathcal{S}'_r(f)$ for various r, we will show how to compute a large enough subset of them (those that don't intersect $\mathcal{W}_r$), and then show that this essentially suffices.

To make all this more precise we introduce some definitions and notation. We will deal with a more general notion of crashing spaces, which we define below as rank-reducing spaces. These

capture reduction of rank to any prespecified threshold, unlike crashing spaces for which we fix a certain threshold.

Definition 13 (Rank-Reducing Spaces)**.** *Let f be a polynomial computed by a $\Sigma\Pi\Sigma(k)$ circuit C, and $t > 0$. A q-dimensional space* $\mathrm{span}(l_1, \ldots, l_q)$ *is a* t-rank reducing space *of f if*

1. $\mathrm{rank}(\mathrm{sim}(C \mod \langle l_1, \ldots, l_q\rangle)) \leq t$,
2. $C \mod \langle l_1, \ldots, l_q\rangle \neq 0$,
3. *and* $C \mod \langle l_1, \ldots, l_q\rangle$ *is not a product of linear forms.*

Define

$$\mathscr{C}_{q,t}(C) = \{\textit{minimal } q\textit{-dimensional } t\textit{-rank reducing spaces of } C\},$$

and

$$\mathscr{C}_{\leq q,t}(C) = \bigcup_{i=1}^{q} \mathscr{C}_{i,t}(C).$$

Note that the set of all minimal crashing spaces of dimension r (as defined in Section 4) is $\mathscr{C}_{r,c_{r-1}}$.

Computing and avoiding all the crashing spaces seems very complicated so we instead deal with LDICR (Low-Dimensional Intersecting Crashing and Regular) spaces as defined below. These will be low-dimensional spaces intersecting all the crashing spaces of dimension up to $r-2$, (moreover we will also ensure that $\mathcal{W}_r$ intersects all regular spaces - we will see why this is useful only later on) and thus avoiding intersection any such space will allow us to avoid containing any crashing space.

Definition 14 (LDICR (Low-dimensional Intersecting Crashing and Regular) Space)**.** *Let f be a n-variate, degree-d polynomial in $\mathbb{F}[x_1, \ldots, x_n]$ that is computed by a $\Sigma\Pi\Sigma(k)$ circuit. We will call a space $\mathcal{W}_r$ an LDICR space with parameter r (for $r \geq 3$) if it has all of the following three properties:*

- $\dim(\mathcal{W}_r) \leq 2^{2k^2}\mathcal{R}_{\mathbb{F}}(k,d)(k^4 \log d)^{r-2}$
- *every space $V \in \mathscr{C}_{\leq r-2,c_k}$ non-trivially intersects $\mathcal{W}_r$*
- *every space $V \in \mathcal{S}^{reg}_{\leq r-1}$ non-trivially intersects $\mathcal{W}_r$.*

We define $\mathcal{W}_1 = \{0\}$ and $\mathcal{W}_2 = \mathrm{Lin}(f)$.

We will show that with iterative computation, we can algorithmically compute a $d^{\mathrm{poly}(\log d)}$ set of spaces, at least one of which will be an LDICR space, but we may not be able to identify which one it is. Thus we will branch on all possible choices of these spaces (this will blow up our run time by a quasi-polynomial factor), and for the correct choice of space, our reconstructing will succeed. Thus one can essentially assume that we are able to pick a true and canonical choice of LDICR space and we fix it to be $\mathcal{W}_r$ for given r. We also have $\mathcal{W}_1 = 0$ and $\mathcal{W}_2 = \mathrm{span}(\{l : l|C\})$. From Corollary 5.3, we can assume $\dim(\mathcal{W}_2) \leq 2^{k^2}\mathcal{R}_{\mathbb{F}}(k,d)\log d + 2k$.

Therefore, instead of computing $\mathcal{S}_r$ or $\mathcal{S}'_r$, we will compute $\mathcal{S}_r$ up to a LDICR space $\mathcal{W}_r$, i.e., compute all spaces in $\mathcal{S}_r$ whose kernels do not intersect $\mathcal{W}_r$. We will define this set to be $\mathcal{S}^*_r$ as follows

$$\mathcal{S}^*_r := \{V \in \mathcal{S}_r(f) : \dim(Ker(V) \cap \mathcal{W}_r) = 0\}.$$

Claim 6.1. $\mathcal{S}_r^* \subseteq \mathcal{S}'_r$

Proof. Since, every space in $\mathscr{C}_{\leq r-2,c_k}$ intersects $\mathcal{W}_r$, any space in $\mathcal{S}_r(f)$ whose kernel contained a crashing space of dimension up to $r-2$ will intersect $\mathcal{W}_r$. Therefore $\mathcal{S}_r^* \subseteq \mathcal{S}'_r$. □

Thus, our size bounds on $\mathcal{S}'_r$ in Theorem 4.1 also hold for $\mathcal{S}_r^*$.

In Section 6.1 we will show how to compute $\mathcal{W}_r$ and in Section 6.2 we will show how to compute $\mathcal{S}_r^*$.

The spaces in $\mathcal{S}_r^*$ have a lot of information, but to effectively use the information, we use $\mathcal{S}_r^*$ to first learn a richer collection of possibly non-maximal vanishing spaces (We call this set $\overline{\mathcal{S}}_r(f)$) which will have some additional properties that will allow us to extract information about the circuit.

The spaces in $\overline{\mathcal{S}}_r(f)$ are harder to define concisely, but they have the interesting property of "saturation" as described below.

Definition 15 ($\overline{\mathcal{S}}_r$ spaces)**.** *Let f be computed by a $\Sigma\Pi\Sigma(k)$ circuit of the form $G \times (T_1 + \cdots + T_k)$. Let $\mathcal{W}_r$ be an LDICR space for f with parameter r. We say that a collection of spaces of codimension at most r is an $\overline{\mathcal{S}}_r$ family for f if it has the following properties.*

1. *Consider any space $\mathbb{V}(l_1, \ldots, l_k)$ of codimension r where $\forall i \in [k], l_i \in \mathrm{Lin}(T_i)$, and such that $\mathrm{span}(l_1, \ldots, l_k) \cap \mathcal{W}_r = \{0\}$. Then, there is a space V in a $\overline{\mathcal{S}}_r$ such that $Ker(V) \subseteq \mathrm{span}(l_1, \ldots, l_k)$ and contains at least one of the l_i's.*

2. *Consider any crashing space $W \in \mathscr{C}_{r-1,c_k}(f)$ such that it doesn't intersect $\mathcal{W}_r$. Let l be a linear form and e be a positive integer such that $l^e | \gcd(C \mod W)$, $\mathrm{span}(W, l) \cap \mathcal{W}_r = \{0\}$ and $l \nmid \mathrm{sim}(C \mod W)$. We say that such a linear form l is* saturated *by the set of spaces $\overline{\mathcal{S}}_r$, if $\overline{\mathcal{S}}_r$ contains a space V with the following properties.*

   (a) *$Ker(V) \subset \mathrm{span}(W, l)$. Let $W' = Ker(V) \cap W$.*

   (b) *There exist e linear forms $l'_1, \ldots, l'_e$, each of the form $l + l'$, where $l' \in W$ such that $\gcd(C \mod W')$ is divisible by $\prod_{i \in [e]} (l'_i \mod W')$.*

   Then, the dimension of the span of linear forms l in $\gcd(C \mod W)$, such that (1) $\mathrm{span}(W, l) \cap \mathcal{W}_r = \{0\}$, (2) $l \nmid \mathrm{sim}(C \mod W)$ and (3) l is not saturated by $\overline{\mathcal{S}}_r$, is at most k.

We will learn the spaces in $\mathcal{W}_r$, $\mathcal{S}_r^*$ and $\overline{\mathcal{S}}_r$ in an iterative manner, with spaces corresponding to the lower values of r being used in the construction of spaces with larger r.

The space $\mathcal{W}_r$ will be constructed from $\mathcal{S}_1^*, \ldots, \mathcal{S}_{r-1}^*, \overline{\mathcal{S}}_1, \ldots, \overline{\mathcal{S}}_{r-1}$ as described in Lemma 6.6. We will then use $\mathcal{W}_r$ to compute $\mathcal{S}_r^*$ and $\overline{\mathcal{S}}_r$ which we will describe in Lemma 6.12 and Lemma 6.14.

We will only be able to compute $\mathcal{W}_r$, and hence $\mathcal{S}_r^*, \overline{\mathcal{S}}_r$, in cases when the circuit doesn't have a certain property which we call the SCS-property (Single Cluster Survives Property) for a given r, as defined below. We will later see that in cases where the circuit does have the property, it only makes our life easier and we are even more easily able to learn linear forms without even learning additional vanishing spaces (see Lemma 6.19).

Definition 16 (SCS (Single Cluster Survives) Property)**.** *Let $t = 2^{2k^2} \mathcal{R}_{\mathbb{F}}(k, d)(k^4 \log d)^{k-3}$ and $\mu = 2t \log d + 2k$. Let f be a n-variate, degree-d polynomial in $\mathbb{F}[x_1, \ldots, x_n]$ that is computed by a $\Sigma\Pi\Sigma(k)$ circuit $C = G \times (T_1 + \cdots + T_k)$ with $gcd(T_1, \ldots, T_k) = 1$ such that it has a μ-cluster representation $C = G \times (C_1 + \cdots + C_s)$ as defined in Lemma 5.7.*

We say a circuit has the $SCS(r)$-Property if r is the smallest positive integer less than s such that for some $j \in [s]$, there exists two totally independent spaces[3] W_a, W_b such that

[3] $\dim(\mathrm{span}(W_a, W_b)) = \dim(W_a) + \dim(W_b)$

1. $W_a, W_b \in \mathscr{C}_{\leq r-2,t_r}$ *for* $t_r := 2^{2k^2}\mathcal{R}_{\mathbb{F}}(k,d)(k^4 \log d)^{r-3}$
2. *There exists space* $W_1, W_2 \in \mathscr{C}_{\leq r-2,c_k}$ *such that* $W_a \subseteq W_1$, $W_b \subseteq W_2$.
3. W_a, W_b *do not non-trivially intersect* $\mathcal{W}_{r-1}$
4. $C \mod \langle W_a \rangle = C_j \mod \langle W_a \rangle$ *and* $C \mod \langle W_b \rangle = C_j \mod \langle W_b \rangle$.

In cases where the $SCS(r)$-Property is satisfied, we give up on computing $\mathcal{S}_r^*$ and $\overline{\mathcal{S}}_r$, but instead directly compute linear forms in the gcd of a cluster using Lemma 6.19.

Else, for any r such that the circuit doesn't have this property, we will show that given $\mathcal{W}_{r-1}$, $\mathcal{S}_1^*, \ldots, \mathcal{S}_{r-1}^*, \overline{\mathcal{S}}_1, \ldots, \overline{\mathcal{S}}_{r-1}$, we can learn $\mathcal{W}_r$, $\mathcal{S}_r^*$, $\overline{\mathcal{S}}_r$. This will be the main goal of this section. The goal of Section 7 will then be to use these $\mathcal{W}_r$, $\mathcal{S}_r^*$, $\overline{\mathcal{S}}_r$ (for all $r \leq s$) to compute enough linear forms from one of the gates.

Theorem 6.2. *Let* $t = 2^{2k^2}\mathcal{R}_{\mathbb{F}}(k,d)(k^4 \log d)^{k-3}$. *Let* f *be a* n*-variate, degree-*d *polynomial in* $\mathbb{F}[x_1, \ldots, x_n]$ *that is computed by a* $\Sigma\Pi\Sigma(k)$ *circuit* $C = G \times (T_1 + \cdots + T_k)$ *with* $gcd(T_1, \ldots, T_k) = 1$ *such that it has a* $2t \log d + 3k$*-cluster representation* $C = G \times (C_1 + \cdots + C_s)$ *as defined in Lemma 5.7. Let* r *be in* $[s]$ *and* $t_r := 2^{2k^2}\mathcal{R}_{\mathbb{F}}(k,d)(k^4 \log d)^{r-3}$. *Then, given access to any LDICR space* $\mathcal{W}_{r-1}$ *of parameter* $r-1$ *and the corresponding sets* $\mathcal{KS}_1^*, \ldots, \mathcal{KS}_{r-1}^*, \mathcal{K}\overline{\mathcal{S}}_1, \ldots, \mathcal{K}\overline{\mathcal{S}}_{r-1}$, *and given black-box access to* f *either*

1. $SCS(r)$*-Property is not satisfied. In this case, we show that there are efficient randomized algorithms for the following tasks.*
   - *A There is an algorithm that runs in time* $(nd)^{\text{poly}(\log d)}$ *and outputs a* $d^{\text{poly}(\log d)}$*-sized list of spaces such that at least one of the spaces* $\mathcal{W}_r$ *that is output is an LDICR space with parameter* r *as defined in Definition 14. (proved in Lemma 6.6)*
   - *B Given* $\mathcal{W}_r$*, there is an algorithm that runs in randomized time* $(nd)^{\text{poly}(\log d)}$ *and computes* $\mathcal{S}_r^*(f)$ *with* $1 - o(1)$ *probability. (proved in Lemma 6.12)*
   - *C Given* $\mathcal{W}_r$ *and* $\mathcal{S}_r^*$*, there is an algorithm that runs in randomized time* $(nd)^{\text{poly}(\log d)}$ *and computes* $\overline{\mathcal{S}}_r$ *with* $1 - o(1)$ *probability. (proved in Lemma 6.14)*
2. $SCS(r)$*-Property is satisfied. Let* $\lambda = min_{i \in [k]}(\dim(\text{span}(\text{Lin}(T_i))))$. *Then there exists an algorithm that computes in* $\text{poly}(n,d)$ *time, a set of linear forms* $\mathcal{L}_{cand}$ *such that* $|\mathcal{L}_{cand}| = d^{O(1)}$ *and* $\exists j \in [k]$ *such that* $\dim(\text{span}(\text{Lin}(T_j) \cap \mathcal{L}_{cand})) \geq \lambda - 2^k \cdot (2t_r \log d + 2k)$. *(Lemma 6.19)*

6.1 Computing $\mathcal{W}_r$

In this section we will show how to compute an LDICR space of paramater r, $\mathcal{W}_r$, given access to any LDICR space $\mathcal{W}_{r-1}$ of parameter $r-1$ and the corresponding sets $\mathcal{KS}_1^*, \ldots, \mathcal{KS}_{r-1}^*, \mathcal{K}\overline{\mathcal{S}}_1, \ldots, \mathcal{K}\overline{\mathcal{S}}_{r-1}$. We will also assume that the $SCS(r)$ property is not satisfied. We will first prove some preliminary lemmas which don't assume anything about the $SCS(r)$ property. This property will only be used in Lemma 6.6. Lemma 6.3 and Lemma 6.5 will also eventually be useful to prove Lemma 6.19 when the $SCS(r)$ property is satisfied.

Lemma 6.3. *Let* $f \in \mathbb{F}[x_1, \ldots, x_n]$ *be a degree-*d *polynomial computed by a* $\Sigma\Pi\Sigma(k)$ *circuit* C *such that* $\text{rank}(\text{sim}(C)) \geq c_k + \dim \mathcal{W}_{r+1} + 2r$. *Let* $1 \leq r \leq k-2$, *and* $t' := c_k + \dim(\mathcal{W}_{r+1}) + 2^{k^2}\mathcal{R}_{\mathbb{F}}(k,d) \log d + 6k$. *There is an algorithm that takes as input* $\mathcal{W}_{r+1}, \mathcal{KS}_1^*, \ldots, \mathcal{KS}_{r+1}^*, \mathcal{K}\overline{\mathcal{S}}_1, \ldots, \mathcal{K}\overline{\mathcal{S}}_{r+1}$, *runs in time* $d^{O_k(1)}$, *and outputs a set* $\mathcal{P}_r$ *with the following properties:*

(i) The number of subspaces in $\mathcal{P}_r$ is $|\mathcal{P}_r| = d^{O_k(1)}$, and

(ii) for every $W \in \mathscr{C}_{\le r,c_k}(C)$ with $W \cap \mathcal{W}_{r+1} = \{0\}$, there is a subspace $\overline{W} \subseteq W$ such that $\overline{W} \in \mathcal{P}_r$ and $\overline{W} \in \mathscr{C}_{\le r,t'}(C)$.

Proof. Let us fix some notation first. Define $\mathcal{K}_r = \mathcal{KS}_1^* \cup \cdots \cup \mathcal{KS}_{r+1}^* \cup \mathcal{K}\overline{\mathcal{S}}_1 \cup \cdots \cup \mathcal{K}\overline{\mathcal{S}}_{r+1}$. Given a crashing space $W \in \mathscr{C}_{\le r,c_k}(C)$ with $W \cap \mathcal{W}_{r+1} = \{0\}$, define

$$\mathcal{L}(W) = \{l : l \text{ is a linear factor of } \gcd(C \mod W)\}$$
$$\overline{\mathcal{L}}(W) = \{l \in \mathcal{L}(W) : \operatorname{span}(W, l) \cap \mathcal{W}_{r+1} = \{0\}\}.$$

Since W is crashing, $m = \operatorname{rank}(\overline{\mathcal{L}}(W)) \ge \operatorname{rank}(\operatorname{sim}(C)) - c_k - \dim \mathcal{W}_{r+1} \ge 2r$. We say a subspace $\overline{W} \subseteq W$ *saturates* a linear form $l \in \overline{\mathcal{L}}(W)$ if any space $V \in \mathcal{K}_r$ with $V \subseteq \operatorname{span}(W, l)$ satisfies $V \cap W \subseteq \overline{W}$. We say $\overline{W}$ is *saturating* if $\operatorname{rank}\{l \in \overline{\mathcal{L}}(W) : \overline{W} \text{ does not saturate } l\} \le 2r$. We will first construct a set $\mathcal{P}_r$ such that for any space $W \in \mathscr{C}_{\le r,c_k}$ with $W \cap \mathcal{W}_{r+1} = \{0\}$, there is a saturating subspace $\overline{W} \subseteq W$ such that $\overline{W} \in \mathcal{P}_r$. Then we will show that this saturating subspace $\overline{W}$ is also in $\mathscr{C}_{\le r,t'}$ for $t' = c_k + \dim(\mathcal{W}_{r+1}) + 2^{k^2}\mathcal{R}_{\mathbb{F}}(k, d) \log d + 6k$, as in the statement of the lemma.

We will construct $\mathcal{P}_r$ iteratively. First, define

$$\mathcal{P}_r^{(1)} = \{\operatorname{span}(V_1, \ldots, V_r) \cap \operatorname{span}(V_{r+1}, \ldots, V_{2r}) : V_1, \ldots, V_{2r} \in \mathcal{K}_r\},$$
$$\mathcal{P}_r^{(i)} = \mathcal{P}_r^{(i-1)} + \mathcal{P}_r^{(1)}.$$

Set $\mathcal{P}_r = \cup_{i=1}^r \mathcal{P}_r^{(i)}$. Fix $W \in \mathscr{C}_{\le r,c_k}$ with $W \cap \mathcal{W}_{r+1} = \{0\}$, and define $\mathcal{P}_r^{(i)}(W) = \{V \in \mathcal{P}_r^{(i)} : V \subseteq W\}$, and $\mathcal{P}_r(W) = \cup_{i=1}^r \mathcal{P}_r^{(i)}(W)$ analogously. We will show by induction on i that any maximum-dimensional subspace in $\mathcal{P}_r^{(i)}(W)$ is either saturating or has dimension at least i. Then, since W itself is an r-dimensional saturating subspace of itself, this shows that there must be a saturating space $\overline{W} \in \mathcal{P}_r(W)$.

For the base case, we only need to show that $\mathcal{P}_r^{(1)}(W)$ contains a nonzero subspace. For each linear form $l \in \overline{\mathcal{L}}(W)$, the space $\operatorname{span}(W, l)$ is a vanishing kernel for the circuit C. Choose a subspace $V_l \in \mathcal{K}_r$ with $V_l \subseteq \operatorname{span}\{W, l\}$, and define $\tilde{W} = \operatorname{span}(V_l : l \in \overline{\mathcal{L}}(W)) \cap W$. Since $\dim \tilde{W} \le r$, we can choose r linear forms $l_1, \ldots, l_r \in \overline{\mathcal{L}}(W)$ such that $\tilde{W} = \operatorname{span}(V_{l_1}, \ldots, V_{l_r}) \cap W$. Since $\operatorname{rank}\overline{\mathcal{L}}(W) \ge 2r$, we can choose linear forms $l_{r+1}, \ldots, l_{2r} \in \overline{\mathcal{L}}(W)$ that are independent from $\{l_1, \ldots, l_r\}$ and define $W' = \operatorname{span}(l_{r+1}, \ldots, l_{2r}) \cap W$. By construction, the spaces $V_{l_{r+1}} \mod W, \ldots, V_{l_{2r}} \mod W$ are all independent from $\{V_{l_1} \mod W, \ldots, V_{l_r} \mod W\}$, so the intersection $\operatorname{span}(V_{l_1}, \ldots, V_{l_r}) \cap \operatorname{span}(V_{l_{r+1}}, \ldots, V_{l_{2r}})$ must in fact be contained in W. In particular, $\operatorname{span}(V_{l_1}, \ldots, V_{l_r}) \cap \operatorname{span}(V_{l_{r+1}}, \ldots, V_{l_{2r}}) = W' \cap \tilde{W}$. Note that by our choice of $l_1, \ldots, l_r$, we actually have $W' \subseteq \tilde{W}$, so $W' \cap \tilde{W} = W'$. Finally, recall that for each $l \in \overline{\mathcal{L}}(W)$, $\operatorname{span}(W, l) \cap \mathcal{W}_{r+1} = \{0\}$, so $\operatorname{span}(W, l)$ does not contain any linear factors of C. This implies that each of the spaces $V_{l_1}, \ldots, V_{l_{2r}}$ is at least 2-dimensional and thus has nonzero intersection with W, so the space W' is a nonzero space in $\mathcal{P}_r^{(1)}(W)$.

For the inductive step, suppose we know that any maximum-dimensional subspace in $\mathcal{P}_r^{(i-1)}(W)$ is either saturating or has dimension $\ge i - 1$. Let $V \in \mathcal{P}_r^{(i-1)}(W)$ be a maximum-dimensional subspace. We will show that if V is not saturating, then we can find a space in $\mathcal{P}_r^{(i)}(W)$ that strictly contains V. For each unsaturated linear form $l \in \overline{\mathcal{L}}(W)$, choose a space $V_l \in \mathcal{K}_r$ with $V_l \subseteq \operatorname{span}(W, l)$ such that $V_l \cap W \not\subseteq V$. As in the base case, define $\tilde{W} = \operatorname{span}(V_l : l \text{ is not saturated by } V) \cap W$. Once again, since $\dim \tilde{W} \le r$, we can find linear forms $l_1, \ldots, l_r$ that are unsaturated by V such that $\tilde{W} = \operatorname{span}(V_{l_1}, \ldots, V_{l_r}) \cap W$. Since V is not saturating, we can also find unsaturated linear forms $l_{r+1}, \ldots, l_{2r}$ independent from $\{l_1, \ldots, l_r\}$, and set $W' = \operatorname{span}(V_{l_{r+1}}, \ldots, V_{l_{2r}}) \cap W$. Then,

following the same reasoning as the base case, $W' \subseteq \tilde{W}$ by definition, $W' \not\subseteq V$ since V is not saturating, and $\text{span}(V_{l_{r+1}}, \ldots, V_{l_{2r}}) \cap \text{span}(V_{l_1}, \ldots, V_{l_r}) = W'$ since the linear forms $l_{r+1}, \ldots, l_{2r}$ are independent from $\{l_1, \ldots, l_r\}$. The fact that $W' \not\subseteq V$ and the expression $\text{span}(V_{l_{r+1}}, \ldots, V_{l_{2r}}) \cap \text{span}(V_{l_1}, \ldots, V_{l_r}) = W'$ tell us that W' is a nonzero subspace in $\mathcal{P}_r^{(1)}(W)$. This shows that $V + W' \in \mathcal{P}_r^{(i)}(W)$ is a strictly larger space than V, so the maximum-dimensional subspace of $\mathcal{P}_r^{(i)}$ has dimension at least i.

Claim 6.4. *Let $W \in \mathscr{C}_{r,c_k}$ be a rank-reducing space such that $\dim(W \cap \mathcal{W}_{r+1}) = 0$. Let $\overline{W} \subset W$ be a subspace of W of dimension r' such that $\overline{W}$ is a saturating space.*

Let $t = c_k + \dim(\mathcal{W}_{r+1}) + 2^{k^2}\mathcal{R}_{\mathbb{F}}(k,d)\log d + 6k$. Then $\overline{W} \in \mathscr{C}_{r',t}$.

Proof. As $W \in \mathscr{C}_{r,c_k}$, we know $\text{rank}(\text{sim}(C \mod W)) \leq c_k$. Our goal is to show that $\text{rank}(\text{sim}(C \mod \overline{W})) \leq c_k + \dim(\mathcal{W}_{r+1}) + 2^{k^2}\mathcal{R}_{\mathbb{F}}(k,d)\log d + 6k$. Let $C \mod W = C' := G' \times (T_1' + \cdots + T_{k'}')$. Therefore, mod W, each gate T_i either vanished or the following holds: some of the linear forms in $\text{Lin}(T_i)$ get mapped to linear forms in T_i', and some get mapped to the gcd. Thus, we can write $\text{Lin}(T_i)$ as $A_i \cup B_i$ where A_i gets mapped to the linear forms in G' and B_i gets mapped to the linear forms in T_i'. Let us assume that k' of the gates remain nonzero, and WLOG these are $T_1, T_2, \ldots, T_{k'}$. Thus $\text{rank}(B_1, \ldots, B_{k'}) \leq c_k + r$ (since W has dimension r). When we go mod $\overline{W}$, the linear forms in B_i's continue to be a part of $\text{sim}(C \mod \overline{W})$. The rank they contribute is at most $c_k + \dim(W) - \dim(\overline{W}) \leq c_k + r$.

The major part of the rank drop when we go mod W comes from the linear forms in the A_i's as they move into G'. These linear forms can be considered in tuples $(l_1, \ldots, l_{k'})$ with a linear form l such that $\forall i \in [k'], l_i \mod W = \alpha_i l$ for some $\alpha_i \in \mathbb{F}$. We will show that except for those tuples whose corresponding l lies in a small-dimensional space, all other tuples will have their linear forms move into the gcd of $C \mod \overline{W}$. We will give up on all those l such that $\text{span}(W, l)$ intersects $\mathcal{W}_{r+1}$, as well as on all those l that are not saturated by $\overline{W}$. We also give up on l such that l divides the $\text{sim}(C \mod W)$. All such l must lie in a $\left(\dim(\mathcal{W}_{r+1}) + 2r + 2^{k^2} \cdot \mathcal{R}_{\mathbb{F}}(k,d)\log d + 2k\right)$-dimensional space as $\overline{W}$ is a saturating space and using Lemma 5.2.

Notice that the space $\text{span}(W, l)$ is such that C vanishes modulo it. Also, $\text{span}(l_1, \ldots, l_{k'}) \subseteq \text{span}(W, l)$. From the saturation property of $\overline{\mathcal{S}}_{r+1}$, described in property 2 of Definition 15, we have that unless l lies in a k-dimensional space, $\mathcal{K}\overline{\mathcal{S}}_r$ will have a subspace V of $\text{span}(W, l)$ such that l has same multiplicity in $C \mod (V \cap W)$ as in $C \mod W$. Since $\overline{W}$ is a saturating space, for any such V, we have $V \cap W \subseteq \overline{W}$. Therefore, when we go mod $\overline{W}$, all the linear forms in the previously mentioned tuples in A_i's, except those whose span intersect a k-dimensional space (from property of $\overline{\mathcal{S}}_r$ spaces), do move into the gcd, just as they did mod W.

Therefore, every $l \in \gcd(C \mod W)$ has the same multiplicity in $\gcd(C \mod \overline{W})$ as in $\gcd(C \mod W)$ unless it lies in a $\left(\dim(\mathcal{W}_{r+1}) + 2^{k^2}\mathcal{R}_{\mathbb{F}}(k,d)\log d + 3k + 2r\right)$-dimensional space. As discussed, the rank increase in linear form from B_i's as we go mod $\overline{W}$ compared to going mod W is at most $c_k + r$. Thus, $\text{rank}(\text{sim}(C \mod \overline{W}) \leq c_k + 3r + \dim(\mathcal{W}_{r+1}) + 2^{k^2}\mathcal{R}_{\mathbb{F}}(k,d)\log d + 3k$. □

□

Definition 17 (Totally Independent Rank-Reducing Sets)**.** *We define a subset $\mathcal{U}_r$ of $\mathscr{C}_{\leq r,t}(C)$ to be a Totally independent rank-reducing set if*

$$\dim(\text{span}(\{W \in \mathcal{U}_r\})) = \sum_{W \in \mathcal{U}_r} \dim(W)$$

i.e. the spaces in $\mathcal{U}_r$ are totally independent.

Similar to Lemma 5.4 for spaces in $\mathcal{S}_r^{reg}$, we will prove a similar structural result for rank-reducing spaces.

Lemma 6.5. *Let f be an n-variate, degree-d polynomial in $\mathbb{F}[x_1, \ldots, x_n]$ that is computed by a $\Sigma\Pi\Sigma(k)$ circuit $C = G \times (T_1 + \cdots + T_k)$ with $gcd(T_1, \ldots, T_k) = 1$ such that it has a $2t \log d + 2k$-cluster representation $C = G \times (C_1 + \cdots + C_s)$ as defined in Lemma 5.7. Let r be any integer such that $1 \leq r \leq s - 2$, and let $\mathcal{U}_r \subseteq \mathscr{C}_{\leq r,t}$ be a totally independent rank-reducing set for f. Then one of the following two scenarios must hold.*

1. *There is a positive integer $j \leq s$, and at least two totally independent spaces[4] $W_a, W_b \in \mathscr{C}_{\leq r,t}(C)$ such that $C \mod \langle W_a \rangle = C_j \mod \langle W_a \rangle$ and $C \mod \langle W_b \rangle = C_j \mod \langle W_b \rangle$.*
2. *$|\mathcal{U}_r| \leq r \cdot (k^2 t \log d + 2k \cdot t)$*

Proof. We will show the lemma for totally independent sets of $\mathscr{C}_{r,t}(C)$ and the results for $\mathscr{C}_{\leq r,t}(C)$ follows with a factor of r. The proof will follow the same outline as Lemma 5.4. We will prove this using induction over r. Consider any space $W \in \mathscr{C}_{r,t}(C)$.

For the base case of $r = 1$, we need to bound the number of independent linear forms modulo which the rank crashes. In case there are two independent linear forms, such that mod them, a single cluster survives, we are done. Therefore, there are at most s independent linear forms for which only one cluster survives, and the rest of all linear forms in $\mathscr{C}_1(t)$ are such that at least two clusters survive mod them. Since, we have at least 2 cluster that survive, which means that the rank of the simple part of the surviving clusters $(\Delta(C_i, C_j))$ drops from $2t \log d + 2k$ (because of distance condition from clustering) to below t. From Lemma 5.1, we know that the number of such independent linear forms is at most $t \log d$. Since, there is are k^2 choices for which clusters survive, we have $|W_1| \leq k^2 t \log d + 2s \cdot t$.

Now for the induction hypothesis, assume the size bound to be true for all $W_1, \ldots, W_{r-1}$. Let $a = |W_r|$. Let $W_r = \{V_1, V_2, \ldots, V_a\}$.

For each i, let l_i be the linear form corresponding to a uniformly random vector sampled from V_i. Let V_i' be a space of dimension $r - 1$ such that $\text{span}\{l_i, V_i'\} = V_i$.

Let $A = \{l_i : i \in [t]\}$.

Now, we consider the circuit $C' = C \mod \langle A \rangle$. Let $f' = f \mod \langle A \rangle$.

Let C' be of the form $C' = G' \times (C_1' + \cdots + C_s')$, where $C_1', \ldots, C_s'$ are all projections of $C_1, \ldots, C_s$ modulo $\langle A \rangle$. Observe that by the definition of rank-reducing spaces, $C' \neq 0$. Moreover each of $G', C_1', \ldots, C_s'$ compute nonzero polynomials because of Claim 5.5. We will show that $V_1', V_2', \ldots, V_a'$ are all in $\mathscr{C}_{r-1,t}(C')$ i.e. are rank-reducing of codimension $r - 1$ for C'. Then, by the induction hypothesis, the bound on a follows.

From Claim 5.5, we see that no two distinct linear forms in the circuit C become the same, and hence there is no new gcd. Consider two clusters C_1, C_2 that survive when we go mod V_i. We know $\Delta(C_1, C_2) \geq 2t \log d + 2k$ and $\Delta(C_1 \mod V_i, C_2 \mod V_i) < t$. As there is no new gcd mod $\langle A \rangle$, we have $\Delta(C_1 \mod \langle A \rangle, C_2 \mod \langle A \rangle) \geq 2t \log d + 2k - a$. Let the drop in $\Delta(C_1, C_2)$ when we go mod $\langle A \rangle$ be δ, which we know is at most a. The independence of spaces $V_1, \ldots, V_a$ means that $\Delta(C_1 \mod \langle A, V_i' \rangle, C_2 \mod \langle A, V_i' \rangle) < t - \delta + 1$. Therefore, the rank crashes from $2t \log d + 2k - \delta$ to at most $t - \delta$, and therefore V_i' is in $\mathscr{C}_{r-1,t}(C')$. □

We now show how to learn $\mathcal{W}_r$ using access to $\mathcal{W}_{r-1}, \mathcal{S}^*_{\leq r-1}, \overline{\mathcal{S}}_{\leq r-1}$.

Lemma 6.6. *Let $t = 2^{2k^2} \mathcal{R}_{\mathbb{F}}(k, d)(k^4 \log d)^{k-3}$. Let f be an n-variate, degree-d polynomial in $\mathbb{F}[x_1, \ldots, x_n]$ that is computed by a $\Sigma\Pi\Sigma(k)$ circuit $C = G \times (T_1 + \cdots + T_k)$ with $gcd(T_1, \ldots, T_k) = 1$*

[4] $\dim(W_a + W_b) = \dim W_a + \dim W_b$

such that it has a $2t\log d + 2k$-cluster representation $C = G \times (C_1 + \cdots + C_s)$ as defined in Lemma 5.7 with $s \geq 3$. Let r be in $[3, s]$ and $t_r := 2^{2k^2}\mathcal{R}_{\mathbb{F}}(k,d)(k^4\log d)^{r-3}$. Assume that the circuit C doesn't satisfy the $SCS(r)$-property as defined in Definition 16. Given access to $\mathcal{W}_{r-1}$ and the sets $\mathcal{KS}^*_1, \ldots, \mathcal{KS}^*_{r-1}, \mathcal{K\overline{S}}_1, \ldots, \mathcal{K\overline{S}}_{r-1}$, there exists an algorithm that outputs a list of size $d^{\text{poly}(\log d)}$ in time $(nd)^{\text{poly}(\log d)}$ which contains spaces of dimension $2^{2k^2}\mathcal{R}_{\mathbb{F}}(k,d)(k^4\log d)^{r-2} = t_{r+1}$ and at least one space satisfies the properties of LDICR spaces as defined in Definition 14.

Proof. We initialize $\mathcal{W}_r := \mathcal{W}_{r-1}$. We then consider all the spaces in $\mathcal{KS}^*_{r-1}$, and find all completely set of independent spaces that span a space of dimension at most $r \cdot 2^{k^2}\mathcal{R}_{\mathbb{F}}(k,d)\log d$, and append them to $\mathcal{W}_r$. From Lemma 5.4, we know that the maximal set of independent regular spaces of dimension r has size at most $2^{k^2}\mathcal{R}_{\mathbb{F}}(k,d)\log d$. Therefore, for the right guess of spaces, every regular space of dimension r must from intersect the space spanned by these independent spaces. Note that any regular space of dimension r, not in $\mathcal{S}^*_r$ must intersect $\mathcal{W}_{r-1}$. The number of such guesses is at most $d^{O(\log d)}$, and we can find them in time $d^{O(\log d)}$ as well.

We then use Lemma 6.3 to obtain the set $\mathcal{P}_{r-2}$ such that for every space $V \in \mathscr{C}_{\leq r-2,c_k}$, there is a subspace W of it in $\mathcal{P}_{r-2}$ such that $W \in \mathscr{C}_{\leq r-2,t'}$ where $t' = t + \dim(\mathcal{W}_{r+1}) + 2^{k^2}\mathcal{R}_{\mathbb{F}}(k,d)\log d + 6k \leq t_r$. Note the existence of at least two clusters having a large distance is enough to meet the rank requirements of Lemma 6.3. Therefore, there exists a subset of $\mathcal{P}_{r-2} \cap \mathscr{C}_{\leq r-2,t_r}$ which contains a subspace of every space in $\mathscr{C}_{\leq r-2,c_k}$. From Lemma 6.5, we know that this subset will have a maximal independent rank-reducing set of size at most $r \cdot (k^2 t_r \log d + 2k \cdot t_r) < 2^{2k^2}\mathcal{R}_{\mathbb{F}}(k,d)(k^4\log d)^{r-2}$. Therefore, an LDICR space as defined in Definition 14 exists and is the span of all the spaces in the maximal independent rank-reducing set described above, union with the independent set of $\mathcal{S}^{reg}_r$ and $\mathcal{W}_{r-1}$.

To find such a $\mathcal{W}_r$, we construct completely independent sets of size $2^{2k^2}\mathcal{R}_{\mathbb{F}}(k,d)(k^4\log d)^{r-2}$ from $\mathcal{P}_{r-2}$. The number of such sets is $d^{r\cdot(k^2 t_r \log d + 2k\cdot t_r)} = d^{\text{poly}(\log d)}$, the span of the spaces in these sets forms the list of spaces we required. □

6.2 Computing $\mathcal{S}^*_r$

In this section we will show how to compute $\mathcal{S}^*_r(f)$ given that we have computed $\mathcal{S}^*_{\leq r-1}(f)$ and $\mathcal{W}_r$. We will first show how to obtain a quasipolynomial time algorithm to compute $\mathcal{S}^*_r(f)$ when the number of variables in the circuit is about $\text{poly}(\log d)$. This algorithm will crucially use algorithms for solving systems of polynomial equations in few variables, which can be efficiently done over $\mathbb{R}, \mathbb{C}$ and finite fields, see Theorem 3.10.

6.2.1 Small variate case

Lemma 6.7. *Let $t = 2^{2k^2}\mathcal{R}_{\mathbb{F}}(k,d)(k^4\log d)^{k-3}$. Let f be a m-variate, degree-d polynomial in $\mathbb{F}[x_1, \ldots, x_m]$ that is computed by a $\Sigma\Pi\Sigma(k)$ circuit $C = G \times (T_1 + \cdots + T_k)$ with $gcd(T_1, \ldots, T_k) = 1$ such that it has a $2t\log d + 2k$-cluster representation $C = G \times (C_1 + \cdots + C_s)$ with $s \geq 2$ as defined in Lemma 5.7. Let r be in $[s]$. Then, given access to $\mathcal{W}_r$ and the sets $\mathcal{KS}^*_1, \ldots, \mathcal{KS}^*_{r-1}$, there exists a randomized algorithm (Algorithm 1) that computes $\mathcal{S}^*_r(f)$ in time $\text{poly}(d^{\text{poly}(m)})$.*

Proof. Let Φ be a random linear isomorphism on $\mathbb{F}[x_1, \ldots, x_m]$ such that $\forall i \in [m], \Phi(x_i) = \sum_{j=1}^m \alpha_{ij}x_j$ where α_{ij} are sampled randomly from $[d^m]$. We first observe that if f vanishes over a codimension r space $\mathbb{V}(l_1, \ldots, l_r)$, then after a random linear isomorphism Φ on the variables, $g = \Phi(f) = f(\Phi(x))$ will vanish over a space $\mathbb{V}(\Phi(l_1), \ldots, \Phi(l_r))$ and moreover this space can be represented in the form $\mathbb{V}(x_1 - l_{a1}, x_2 - l_{a2}, \ldots, x_r - l_{ar})$ for linear forms $\forall j \in [r], l_{aj} \in \mathbb{F}[x_{r+1}, \ldots, x_n]$. Using Lemma 3.8, we can get monomial access to g in time $\text{poly}(d^m)$.

Let $l_{ai} = \sum_{j=r+1}^{m} a_{ij}x_j$ for variables $a_{ij}, i \in [r], j \in [r+1, m]$. We will substitute $x_i = l_{ai}$ into the monomial representation of the polynomial, and obtain the polynomial $g \mod \langle x_1 - l_{a1}, \ldots, x_r - l_{ar}\rangle$. We equate the coefficients in $x_{r+1}, \ldots, x_m$ to 0 to obtain a system of polynomial equations in $r(m-r)$ variables of degree d and at most d^m equations. This system of polynomial equations might have infinitely many solutions unless we discard all those codimension r spaces that are contained in $\mathcal{S}^*_{<r}$ spaces and those which intersect $\mathcal{W}_r$. We also know from Theorem 4.1 and the fact that $\mathcal{W}_r$ is an LDICR space (Definition 14) that this suffices as $m \geq c_r$. The challenge remains to remove these spaces. To discard all those codimension r spaces that are contained in $\mathcal{S}^*_{<r}$ spaces, we add additional polynomial equations to the system of polynomial equations, that ensure for any $r' < r$ and $\mathbb{V}(l'_1, \ldots, l'_{r'}) \in \mathcal{S}^*_{r'}(g)$, $\dim(\text{span}(l'_1, \ldots, l'_{r'}, x_1 - l_{a1}, \ldots, x_r - l_{ar})) \geq r + 1$. We assume we are given a basis of $\mathcal{W}_r$, computed in Lemma 6.6. Then, we add another equation to ensure that $\text{span}(x_1 - l_{a1}, \ldots, x_r - l_{ar})$ doesn't intersect $\mathcal{W}_r$, which ensures the spaces computed do not contain any crashing spaces of dimension at most $r-2$. Finally, having computed the spaces $\mathbb{V}(x_1 - l_{a1}, \ldots, x_r - l_{ar})$ on which g vanishes by solving the system of equations, we simply apply Φ^{-1} to get $\mathbb{V}(l_1, \ldots, l_r)$.

We now give a more detailed analysis.

We first observe that in Step 1 of Algorithm 1, the random linear forms $\tilde{l}_1, \ldots, \tilde{l}_m$ will be independent with high probability (as otherwise it will correspond to a certain determinant evaluating to 0, which happens with probability at most $d^{-(m-1)}$ due to Lemma 3.3).

Thus, with high probability Φ is a random isomorphism, and we obtain the polynomial $g = \Phi(f)$ which is also computable by a $\Sigma\Pi\Sigma(k)$ circuit over m variables, and the simple part of the circuit has rank m. From now onwards, let us assume that Φ is an isomorphism.

g vanishes on spaces of the form $\mathbb{V}(x_1 - l_{a1}, \ldots, x_r - l_{ar})$ As Φ is an isomorphism, f vanishes on $\mathbb{V}(l_1, \ldots, l_r)$ if and only if g vanishes on $\mathbb{V}(\Phi(l_1), \ldots, \Phi(l_r))$. We will first observe that with high probability, for any space $\mathbb{V}(\Phi(l_1), \ldots, \Phi(l_r)) \in \mathcal{S}_r(g)$, there are linear forms $l_{a1}, \ldots, l_{ar} \in \mathbb{F}[x_{r+1}, \ldots, x_m]$ such that $\mathbb{V}(x_1 - l_{a1}, \ldots, x_r - l_{ar}) = \mathbb{V}(\Phi(l_1), \ldots, \Phi(l_r))$. The reason is the following: For $i \in [r]$, let $l_i = \sum_{j \in [m]} u_{ij}x_j$. As $\mathbb{V}(l_1, \ldots, l_r)$ is a codimension r space, hence $\dim(\text{span}(l_1, \ldots, l_r)) = r$. After applying the isomorphism Φ, they remain independent with high probability and coefficients of x_a in $\Phi(l_i)$ as $\sum_{j=1}^{n} \alpha_{a,j}u_{ij}x_j$. As they were independent, the determinant of the $r \times r$ matrix formed by the coefficients of $x_1, \ldots, x_r$ from $\Phi(l_1), \ldots, \Phi(l_r)$ will be a non-zero polynomial in $\alpha_{1,1}, \ldots, \alpha_{r,m}$ and will vanish with low probability due to Lemma 3.3. This means that for space $\mathbb{V}(l_1, \ldots, l_r) \in \mathcal{S}_r(g)$ there is a space $\mathbb{V}(x_1 - l_{a1}, \ldots, x_r - l_{ar}) \in \mathcal{S}_r(g)$ where $l_{a1}, \ldots, l_{ar} \in \mathbb{F}[x_{r+1}, \ldots, x_m]$.

Setting up a system of equations Observe that we can use interpolation to get monomial access to g in time $\text{poly}(d^m)$ using Lemma 3.8.

For all $i \in [r]$, we set $l_{ai} = \sum_{j=r+1}^{m} a_{ij}x_j$ for variables $\{a_{ij} : i \in [r], j \in ([m] \setminus [r])\}$. Substituting $x_i = l_{ai}$ into the monomial form, we obtain a system of $d^{O(m)}$ equations of degree at most d in $r(m-r)$ variables by equating the coefficients of monomials in the variables $x_{r+1}, \ldots, x_m$ to 0. Solutions to this would correspond to codimension r spaces that g vanishes on.

To remove the codimension r spaces that are contained in $\mathcal{S}^*_{r'}(f)$ for $r' < r$, we apply Φ on the linear forms that are the basis of the spaces in $\mathcal{S}^*_1(f), \ldots, \mathcal{S}^*_{r-1}(f)$, to compute $\mathcal{S}^*_1(f), \ldots, \mathcal{S}^*_{r-1}(f)$.

Then, we need to ensure that the solution to our system of equations $x_1 - l_{a1}, \ldots, x_r - l_{ar}$ is such that $\forall(l'_1, \ldots, l'_{r'})$ such that $\mathbb{V}(l'_1, \ldots, l'_{r'}) \in \mathcal{S}^*_{r'}(g)$, we have that $\dim(\text{span}(x_1 - l_{a1}, \ldots, x_r - l_{ar}, l'_1, \ldots, l'_{r'})) = r + 1$. This is the same as saying that the $m \times (r' + r)$ matrix $A_{(l'_1, \ldots, l'_{r'})}$ with $x_1 - l_{a1}, \ldots, x_r - l_{ar}, l'_1, \ldots, l'_{r'}$ as rows have rank at least $r + 1$. This means at least one of the

$(r+1)\times(r+1)$ minors of $A_{(l'_1,\ldots,l'_{r'})}$ is full rank, with a non-zero determinant. Let the number of such minors be $k_{r'} := \binom{m}{r+1}\cdot\binom{r+r'}{r+1}$. To handle these, we introduce new variables $y_{r'1},\ldots,y_{r'k_{r'}}$, and for each relevant $(l'_1,\ldots,l'_{r'})$ we consider the inequalities $sum_{(l'_1,\ldots,l'_{r'})} = \sum_{j=1}^{k_{r'}} y_j M_j \neq 0$, where M_j are the determinants of the $(r+1)\times(r+1)$ minors of $A_{(l'_1,\ldots,l'_{r'})}$. The inequality has solutions if and only if there exists a solution for which at least one of the M_j is non-zero. We note that we can use the same new variables $y_{r'1},\ldots,y_{r'k_{r'}}$ in all of the inequalities. Observe that the set of inequalities $\forall\ \mathbb{V}(l'_1,\ldots,l'_{r'})\in\mathcal{S}_{r'}(g), sum_{(l'_1,\ldots,l'_{r'})}\neq 0$ is the same as having a single inequality $Prod_{r'} = (\prod_{\mathbb{V}(l'_1,\ldots,l'_{r'})\in\mathcal{S}_{r'}(g)} sum_{(l'_1,\ldots,l'_{r'})})\neq 0$. We can further combine these inequalities further for all $r'\in[r-1]$, by using the inequality $\prod_{i\in[r-1]} Prod_i\neq 0$, which is same as requiring that $\prod_{i\in[r-1]} Prod_i\cdot z = 1$ has a solution for a new variable z.

Thus, we can handle the condition of the solution not lying in any $\mathcal{S}^*_{r'}(g)$ space for $r'<r$, by simply adding one extra equation of degree $\sum_{i=1}^{r-1}(r+2)\cdot|\mathcal{S}_{r'}(g)| = d^{O(1)}$ and $(\sum_{i=1}^{r-1}k_i)+1 = \text{poly}(m)$ variables to the system of equations we had earlier.

Removing Spaces intersecting $\mathcal{W}_r$ In this algorithm, we assume we have already computed the LDICR space $\mathcal{W}_r$ we are going to avoid. Let $\mathcal{W}_r = \text{span}(l'_1,\ldots,l'_{\dim(\mathcal{W}_r)})$ where $l'_1,\ldots,l'_{\dim(\mathcal{W}_r)}$ format basis of $\mathcal{W}_r$ which we are given. We first compute $\mathcal{W}_r$ for g by applying Φ to $l'_1,\ldots,l'_{\dim(\mathcal{W}_r)}$. To ensure that the new spaces we find don't intersect $\mathcal{W}_r$, we will ensure that $\dim(\text{span}(\mathcal{W}_r, x_1-l_{a1},\ldots,x_r-l_{ar})) = \dim(\mathcal{W}_r)+r$. We do this by ensuring the matrix(M) with the basis of $\mathcal{W}_r$ and $x_1-l_{a1},\ldots,x_r-l_{ar}$ as rows is of full rank, i.e., it has a non-zero determinant. This can be ensured by introducing a new variable z' and adding the equation $z'\cdot det(M)=1$ to our system of equations. The equation adds one extra variable and has degree $\dim(\mathcal{W}_r)+r\leq t+r\leq m$.

Running Time Analysis The sampling of the random $\alpha_{i,j}$ can be done in randomized poly$(m,\log d)$ time. From Lemma 3.9, we can get black-box access to the factors in time randomized poly(m,d). We can do interpolation and get monomial access to g in time $d^{O(m)}$ using Lemma 3.8. As $|\mathcal{S}'_{r'}(g)|\leq d^{O(1)}$ from Theorem 4.1, combined with the loop on line 11 runs $d^{O(1)}$ times, doing poly(m) computation. Finally, the system of equations has $d^{O(m)}$ equations with degree $(md)^{O(1)}$ in poly(m) variables. The system has $d^{O(1)}$ solution by combining Theorem 4.1 and Claim 6.1. Therefore, we can find $l_{a1},\ldots,l_{ar}$ by solving the system of equations in time poly$(d^{\text{poly}(m)})$ using Theorem 3.10. Thus, the entire algorithm works in poly$(d^{\text{poly}(m)})$ time. □

Algorithm 1 Computing Vanishing $\mathcal{S}_r(f)$ for constant variate polynomials

Input: Black-box access to circuit C of form $\Sigma\Pi\Sigma(k)$ computing polynomial $f \in \mathbb{F}[x_1, \ldots, x_m]$ with properties as described in Lemma 6.7, $\mathcal{W}_r$, $\mathcal{S}_1^*, \ldots \mathcal{S}_{r-1}^*$

1: **function** $\mathcal{S}_r^*(f)$

2: Sample m^2 random values $\alpha_{ij}; i, j \in [m]$ uniformly from $\{1, \ldots, d^m\}$, and use them to define m linear forms $\tilde{l}_i = \sum_{j=1}^m \alpha_{ij} x_j$. Check if they are independent, otherwise repeat. Define isomorphism Φ such that for all $i \in [m]$, $\Phi(x_i) := \tilde{l}_i'$. Let $g = \Phi(f) = f(\Phi(x))$.

3: Using randomized black-box factoring from Lemma 3.9 and get access to the linear factors of g

4: Interpolate g to get monomial access to it

5: **for** $i \in [r]$ **do**

6: Substitute $x_i = \sum_{j=r+1}^m a_{ij} x_j$ for linear form $x_i - l_{ai}$ in g.

7: Obtain equations in $\{a_{ij} : i \in [r], j \in ([m] \setminus [r])\}$ by equating the coefficients of monomials in the variables $x_{r+1}, \ldots, x_{m-1}, x_m$ to 0.

8: **for** $i \in [r-1]$ **do**

9: Let $k_i = \binom{m}{r+1} \cdot \binom{i+r}{r+1}$. Introduce new variables $y_{i,1}, \ldots, y_{i,k_i}$. Set $Prod_i := 1$.

10: **for** $\mathbb{V}(l_1', \ldots, l_i') \leftarrow \mathcal{S}_i^*(g)$ **do**

11: Consider the $(i+r) \times m$ matrix $A_{(l_1', \ldots, l_i')}$ formed by $l_1', \ldots, l_i'$ and $x_1 - l_{a1}, \ldots, x_r - l_{ar}$.

12: Compute the determinant of each $(r+1) \times (r+1)$ minor M_j of $A_{(l_1', \ldots, l_i')}$. Compute $sum_{(l_1', \ldots, l_i')} := \sum_{j=1}^{k_1} y_j M_j$.

13: $Prod_i := Prod_i \times sum_{(l_1', \ldots, l_i')}$

14: Introduce a new variable z. Add Equation $\prod_{i \in [r-1]} Prod_i \cdot z = 1$ to the system of equations in Step 7.

15: Construct M with basis of $\Phi(\mathcal{W}_r)$ and $x_1 - l_{a1}, \ldots, x_r - l_{ar}$. Introduce a new variable z'. Add equation $z' \cdot \det(M) = 1$ to the system of equations.

16: Solve the system of equations in $\{a_{ij} : i \in [r], j \in [m] \setminus [r]\}$ and $\{y_{i,j} : i \in [r-1], j \in [k_i]\}, z, z'$ using Theorem 3.10 to obtain a set of $(x_1 - l_{a1}, \ldots, x_r - l_{ar})$.

17: Verify for each $(x_1 - l_{a1}, \ldots, x_r - l_{ar})$ if f vanishes on $\mathbb{V}(\Phi^{-1}(x_1 - l_{a1}), \ldots, \Phi^{-1}(x_r - l_{ar})) = \mathbb{V}(l_1, \ldots, l_r)$ using Lemma 3.3, then add $\mathbb{V}(l_1, \ldots, l_r)$ to $\mathcal{S}_r^*(f)$

18: **Output** $\mathcal{S}_r^*(f)$.

6.2.2 General case

We will now discuss how we can use the solution for the low-variate case to compute $\mathcal{S}_r^*(f)$ in the general case.

Let $t = 2^{2k^2} \mathcal{R}_{\mathbb{F}}(k, d)(k^4 \log d)^{k-3}$. We will start by using a random linear isomorphism Φ on f such that $\Phi(x_i) = \sum_{j=1}^n \alpha_{ij} x_j$, where α_{ij} are chosen randomly from $[d^n]$, and define $g = \Phi(f) = f(\Phi(x))$. Let $m = 2t \log d + 2k + 2$. We will then consider the m variate polynomials g_i (for $i \geq m$) which are obtained from g by setting all variables x_j for $j > m-1$ to zero, except x_i.

Thus

$$g_i = g|_{x_m = \cdots = x_{i-1} = x_{i+1} = \cdots = x_n = 0}$$

We will then find $\mathcal{S}_r^*(g_i)$ spaces using the low-variate algorithm and then show how to glue the learned spaces to get $\mathcal{S}_r^*(g)$ and then $\mathcal{S}_r^*(f)$. The $\mathcal{W}_r$ we use for computation of g_i, will be the $\mathcal{W}_r$ we obtain from Lemma 6.6, with Φ and projected down same as g_i.

We will need the following collection of simpler properties about g_i's summed up in the following lemmas to prove the correctness of the algorithm computing $\mathcal{S}_r^*(f)$.

Lemma 6.8. *Let $t = 2^{2k^2}\mathcal{R}_{\mathbb{F}}(k,d)(k^4\log d)^{k-3}$. Let f be a n-variate, degree-d polynomial in $\mathbb{F}[x_1,\ldots,x_n]$ that is computed by a $\Sigma\Pi\Sigma(k)$ circuit $C = G\times(T_1+\cdots+T_k)$ with $\gcd(T_1,\ldots,T_k)=1$ such that it has a $2t\log d+2k$-cluster representation $C = G\times(C_1+\cdots+C_s)$ as defined in Lemma 5.7.*

With probability $1-o(1)$, For each $i\in\{m,\ldots,n\}$, the polynomials g_i can be computed by $\Sigma\Pi\Sigma(k)$ circuits C_i such that they have $2t\log d+2k$-cluster presentation $C_i = G_i\times(C_{1i}+\cdots+C_{si})$ as defined in Lemma 5.7.

Proof. The circuit C after applying the isomorphism Φ will be of the form

$$\Phi(C) = \Phi(G)\times(\Phi(C_1)+\cdots+\Phi(C_s)),$$

We denote Γ_i as the homomorphism from $\mathbb{F}[x_1\ldots,x_n]$ to $\mathbb{F}[x_1,\ldots,x_{m-1},x_i]$ mapping $x_j\to 0$ for $j\in\{m,\ldots,i-1\}\cup\{i+1,\ldots,n\}$. Then for each i, g_i is computable by the following circuit

$$g_i = \Gamma_i(g) = \Gamma_i(\Phi(G))\times(\Gamma_i(\Phi(C_1))+\cdots+\Gamma_i(\Phi(C_s)))$$

We will first argue that with high probability $\Delta(\Gamma_i(\Phi(C_a)),\Gamma_i(\Phi(C_b)))\geq 2t\log d+2k$ for any $a\neq b\in[s]$. Consider any two linear forms $l=\sum_{i=1}^n c_ix_i$ and $l'=\sum_{i=1}^n c'_ix_i$ such that $\text{span}(l)\neq\text{span}(l')$ and $l\in\gcd(C_a), l'\in\gcd(C_b)$ respectively. After applying Φ, the coefficients of x_1,x_2 in $\Phi(l),\Phi(l')$ will be $\sum_{i=1}^n\alpha_{1,i}c_ix_i$ and $\sum_{i=1}^n\alpha_{2,i}c_ix_i$ for $\Phi(l)$ and $\sum_{i=1}^n\alpha_{1,i}c'_ix_i$ and $\sum_{i=1}^n\alpha_{2,i}c'_ix_i$ for $\Phi(l')$. As $\text{span}(l)\neq\text{span}(l')$, if these two linear forms were distinct, they would become equal and move to the gcd of C_a+C_b only if the determinant of the 2×2 matrix with these coefficients as entries becomes identically 0. This happens with vanishingly small probability (by Lemma 3.3) as the determinant is a non-zero polynomial in $\alpha_{1,1},\ldots,\alpha_{2,n}$ and we choose $\alpha_{i,j}$ from a large set of size d^n.

Also, as $\Delta(C_a,C_b)\geq 2t\log d+2k$, we have $\text{rank}(\text{sim}(C_i+C_j))\geq 2t\log d+2k$. From above we have that the linear forms that were in $\text{sim}(C_i+C_j)$, stay in $\text{sim}(\Gamma_i(\Phi(C_A))+\Gamma_i(\Phi(C_B)))$. These linear forms spanned a space with dimension at least $2t\log d+2k$, and after the random projection with high probability still span a space of dimension $2t\log d+2k$. Therefore, for each i, and $a,b\in[s]$ we conclude that $\text{rank}(\text{sim}(C_a+C_b))\geq 2t\log d+2k$. Thus, all g_i have $2t\log d+2k$ cluster representation with the same top fan-in as C with $1-o(1)$ probability.

□

We show that after the random projection, crashing spaces of g_i continue to intersect $\mathcal{W}_r$ that was computed for f.

Lemma 6.9. *Let $t = 2^{2k^2}\mathcal{R}_{\mathbb{F}}(k,d)(k^4\log d)^{k-3}$. Let f be a n-variate, degree-d polynomial in $\mathbb{F}[x_1,\ldots,x_n]$ that is computed by a $\Sigma\Pi\Sigma(k)$ circuit $C = G\times(T_1+\cdots+T_k)$ with $gcd(T_1,\ldots,T_k)=1$ such that it has a $2t\log d+2k$-cluster representation $C = G\times(C_1+\cdots+C_s)$ as defined in Lemma 5.7. Let $m = 2t\log d+2k+2$.*

Then with probability $1-o(1)$, for all $r\in[k]$ and every $V\in\mathscr{C}_{\leq r-2,c_k}(g_i)$

$$\dim(V\cap\Phi(\mathcal{W}_r)|_{x_m=\cdots=x_{i-1}=x_{i+1}=\cdots=x_n=0})\geq 1$$

Proof. Note that g_i is obtained by first applying a random linear isomorphism and then setting several variables to zero. To prove the lemma, it is enough to handle the simpler case where we apply a random linear isomorphism and then set just one variable to zero. If we can show that, with very high probability, the resulting circuit still retains the desired property in this case, then we can iterate the process: alternately applying random linear isomorphisms and setting one variable to zero at a time. This allows us to reduce the circuit to m variables while maintaining, with high

probability, any crashing space which $r-2$ dimensional continues to intersect $\mathcal{W}_r$ projected to the m variables. Note that by Lemma 6.8, the intermediate polynomials till g_i continue to have a $2t\log d + 2k$-cluster representation.

Note that, repeatedly applying random isomorphisms and zeroing variables one by one is equivalent to applying a single random isomorphism and then zeroing several variables simultaneously. We will thus work with the iterated process. Adapting the argument to the original distribution (or redefining g_i to be drawn from the recursive distribution) is straightforward.

Therefore, what we need to show is that for a random linear form l (coefficients are chosen randomly from $[d^n]$), all $(r-2)$-dimensional crashing spaces of $C \mod l$ intersect $\mathcal{W}_r$. To show this, we will first show the following claim:

Claim 6.10. *For $C = G \times (T_1 + \cdots + T_k)$ with $\gcd(T_1, \ldots, T_k) = 1$ and $\operatorname{rank}(\operatorname{sim}(C)) > t$, any minimal rank-reducing space $V \in \mathscr{C}_{r,t}$ of C is in the span of $t + r \cdot (k+1)$ linear forms in $\operatorname{Lin}(T_1), \ldots, \operatorname{Lin}(T_k)$.*

Proof. For the purpose of this proof, note that we can without loss of generality assume that $\gcd(C) = 1$, so that $C = \operatorname{sim}(C) = T_1 + \cdots + T_k$. Let the circuit $C \mod V$ be of the form $C \mod V = G' \times (T_1' + \cdots + T_k')$, where $G' = \gcd(C \mod V)$ and $T_1' + \cdots + T_k' = \operatorname{sim}(C \mod V)$. For each $i = 1, \ldots, k$, we can write each T_i as $T_i = A_i B_i$, where $A_i \mod V = G'$ and $B_i \mod V = T_i'$.

First, for each linear factor l of G' (with multiplicity), we can find linear forms l_1 dividing A_1,..., l_k dividing A_k such that $l_1 \mod V = \cdots = l_k \mod V = l$. Define $V_l = \operatorname{span}(l_1, \ldots, l_k)$, and let $W_A = \operatorname{span}\left(\bigcup_{l \mid G'} V_l\right) \cap V$. By construction, $\gcd(C \mod W_A) = \gcd(C \mod V)$. Since $\dim W_A \leq r$, W_A is contained in the span of at most r of the spaces V_l, where l divides G', so it is contained in the span of at most rk linear forms in $T_1, \ldots, T_k$.

Next, we turn our attention to $\operatorname{sim}(C \mod V)$. We know that $t \geq \operatorname{rank}(\operatorname{sim}(C \mod V)) = \operatorname{rank}\left(\sum_{i=1}^k B_i \mod V\right) \geq \operatorname{rank}\left(\sum_{i=1}^k B_i\right) - r$. So, we can choose $t+r$ linear forms $l_1, \ldots, l_{t+r}$ from $\operatorname{Lin}(B_1), \ldots, \operatorname{Lin}(B_k)$ that span $\operatorname{span}(\operatorname{Lin}(B_1) \cup \cdots \cup \operatorname{Lin}(B_k))$. Define $W_B = \operatorname{span}(l_1, \ldots, l_{t+r}) \cap V$. Then, $\operatorname{rank}\left(\sum_{i=1}^k B_i \mod W\right) = \operatorname{rank}\left(\sum_{i=1}^k B_i \mod V\right)$ by construction.

Finally, set $W = W_A + W_B$. The space W is contained in the span of $t + r(k+1)$ linear forms in $T_1, \ldots, T_k$. We know that $\gcd(C \mod W) = \gcd(C \mod V)$ by construction of W_A. And by the construction of W_B, this implies that $\operatorname{rank}(\operatorname{sim}(C \mod W)) = \operatorname{rank}(\operatorname{sim}(C \mod V)) \leq t$, so W is a crashing space. By the minimality of V, W must be equal to V which concludes our proof. □

For the sake of contradiction, let l be a random linear form and let $V \in \mathscr{C}_{\leq r-2, c_k}(C \mod l)$ be a crashing space that doesn't intersect $\mathcal{W}_r \mod l$. First observe, with $1 - o(1)$ probability, l doesn't lie in $\mathcal{W}_r$. This follows from the fact that if l lied in $\mathcal{W}_r$, the matrix formed by linear forms in basis of $\mathcal{W}_r$ and l will have rank $\dim(\mathcal{W}_r)$, which means the determinant of the sub-matrices of size $(\dim(\mathcal{W}_r)+1) \times (\dim(\mathcal{W}_r)+1)$ is zero, which happens with $o(1)$ probability from Lemma 3.3. Then, we consider the space $\operatorname{span}(V, l)$, which is a crashing space of dimension $r-1$ for C. We will first argue that $\operatorname{span}(V, l)$ is a minimal crashing space and therefore is in $\mathscr{C}_{r-1,c_k}(C)$. For contradiction, assume there is a strict subspace V' that is the the minimal crashing space in $\operatorname{span}(V, l)$, i.e. $V' \subset V$ such that $\operatorname{rank}((C \mod V')) \leq c_k$. As it is a proper subspace, $\dim(V') \leq r-2$. Therefore, V' must intersect $\mathcal{W}_r$ non-trivially from definition of $\mathcal{W}_r$. But we know $V \mod l$ doesn't intersect $\mathcal{W}_r \mod l$, which means $\operatorname{span}(V, l)$ doesn't intersect $\mathcal{W}_r$ and therefore V' doesn't intersect $\mathcal{W}_r$ which is a contradiction. Therefore, we have $\operatorname{span}(V, l) \in \mathscr{C}_{r-1,c_k}$.

From Claim 6.10, we have that $\operatorname{span}(V, l)$ must lie in a space spanned by $c_k + k(r+1)$ linear forms in $\operatorname{sim}(C)$, which means the random linear form l lies in the span of these linear forms.

For any fixing of $c_k + k(r+1)$ linear forms, we will show that the probability that l lies in their span is very small, and then we will take a union bound over all possible choices of $c_k + k(r+1)$ linear forms in the circuit. Let S be any set of $c_k + k(r+1)$ linear forms in $\text{sim}(C)$ and let their span have dimension a where $a \leq c_k + k(r+1)$. This means the matrix formed by the vectors corresponding to the linear forms in S and our random linear form l has rank a and therefore the sub-matrices of dimensions $(a+1) \times (a+1)$ have zero determinant, which happens with probability $\frac{a+1}{d^n} \leq \frac{c_k+k(r+1)+1}{d^n}$ by Lemma 3.3. The number of possibilities for the choice of S is $(kd)^{c_k+k(r+1)}$. Taking union bound over all such choices, we get that the probability that l lies in the span of at most $c_k + k(r+1)$ linear forms in $\text{sim}(C)$ is at most $\frac{(c_k+k(r+1)+1)\cdot(kd)^{c_k+k(r+1)}}{d^n} = o(1)$. □

We will also need to show that there are no new minimal vanishing spaces of r' dimensions for $r' < r$ that will affect the computation of $\mathcal{S}^*_r$. We don't show this, but rather show that any such new space must intersect $\mathcal{W}_{r'+1}$ (and hence $\mathcal{W}_r$), and therefore still doesn't affect the computation of $\mathcal{S}^*_r$.

Lemma 6.11. *If f is computed by a $\Sigma\Pi\Sigma(k)$ circuit C as described in Theorem 6.2, then with probability $1-o(1)$, for all $r \in [k-1]$, any $V \in \mathcal{S}^*_r(g_i)$ such that $V \notin \{\Phi(\mathcal{S}^*_r(f))\}|_{x_m=\cdots=x_{i-1}=x_{i+1}=\cdots=x_n=0}$, we have $\dim(Ker(V) \cap \Phi(\mathcal{W}_{r+1})|_{x_m=\cdots=x_{i-1}=x_{i+1}=\cdots=x_n=0}) \geq 1$.*

Proof. We already argued in Lemma 6.9, that we can use the same $\mathcal{W}_r$ projected down for the calculations of $\mathcal{S}^*_r$, and therefore only need to argue that there is no new codimension r space over which g_i vanishes, which doesn't correspond to a space in $\mathcal{S}^*_r(g)$ for $r < k-1$ or it intersects $\mathcal{W}_{r+1}$.

Similar to Lemma 6.9, we argue this recursively and therefore only need to show that no new $\mathcal{S}^*_r$ spaces are added when we consider $f \mod l$ for a random linear form l.

Consider a new space $\mathbb{V}(l_1, \ldots, l_r) \in \mathcal{S}^*_r(f \mod l)$ such that it doesn't correspond to a space in $\mathcal{S}^*_r(f)$, i.e. $\mathbb{V}(l_1, \ldots, l_r) \notin \{\mathcal{S}^*_r(f)\}|_{l=0}$. This means that the space $\mathbb{V}(l_1, \ldots, l_r, l)$ is a codimension $r+1$ space on which f vanishes. If $\text{span}(l_1, \ldots, l_r)$ intersects $\mathcal{W}_{r+1}$, we are done. We will now argue that if $\text{span}(l_1, \ldots, l_r)$ doesn't intersect $\mathcal{W}_{r+1}$, then $\text{span}(l_1, \ldots, l_r, l)$ doesn't intersect $\mathcal{W}_{r+1}$ either with high probability. For that to happen, the matrix with basis of $\mathcal{W}_{r+1}$ and $l_1, \ldots, l_r, l$ will need to have rank at most $\dim(\mathcal{W}_{r+1}) + r$, which using Lemma 3.3 happens with $o(1)$ probability.

Therefore, we know now that $V = \text{span}(l_1, \ldots, l_r, l)$ doesn't intersect $\mathcal{W}_{r+1}$ with high probability. Thus, it must have a subspace in $\mathcal{KS}^*_{\leq r+1}$. Let this subspace be V'. If V' doesn't contain l, then either V' is just $\text{span}(l_1, \ldots, l_r)$ or its a subspace of $\text{span}(l_1, \ldots, l_r)$ is not minimal and not in $\mathcal{S}^*_r$. Therefore, V' must contain l. Also, V' must be in $\mathcal{KS}^*_{r'}$ for some $r' \leq r+1$. For a given V', the probability that l lies inside it is equal to probability that the rank of matrix formed by l and basis of V' has rank equal to dimension of V which happens with probability at most $r + 1/d^n$. From Claim 6.1 and Theorem 4.1, we know the number of V' spaces will be at most $d^{O(1)}$, and taking union bound over it, we see that it happens with probability $o(1)$.

□

We will now combine the above lemmas and use a gluing procedure to show that we can use the computation of $\mathcal{S}^*_r(g_i)$'s to obtain $\mathcal{S}^*_r(f)$.

Lemma 6.12. *Let $t = 2^{2k^2}\mathcal{R}_{\mathbb{F}}(k,d)(k^4 \log d)^{k-3}$. Let f be a n-variate, degree-d polynomial in $\mathbb{F}[x_1, \ldots, x_n]$ that is computed by a $\Sigma\Pi\Sigma(k)$ circuit $C = G \times (T_1 + \cdots + T_k)$ with $gcd(T_1, \ldots, T_k) = 1$ such that it has a $2t \log d + 2k$-cluster representation $C = G \times (C_1 + \cdots + C_s)$ as defined in Lemma 5.7. Let r be in $[s]$. Then, given access to $\mathcal{W}_r$ and the sets $\mathcal{KS}^*_1, \ldots, \mathcal{KS}^*_{r-1}$, there exists a randomized algorithm (Algorithm 2) that outputs $\mathcal{S}^*_r$, in $\text{poly}(n, d^{\text{poly}(t)})$-time with probability $1 - o(1)$.*

Proof. We first apply a random linear isomorphism and obtain $g = \Phi(f)$. For $m = 2t \log d + 2k + 2$, we obtain $n - m$ polynomials $g_i \in \mathbb{F}[x_1, \ldots, x_{m-1}, x_i]$ by setting all except $x_1, \ldots, x_{m-1}, x_i$ to 0. Then we solve the low variate cases, using Lemma 6.7 discussed above to recover $\mathcal{S}^*_r(g_i)$. We will then recover $\mathcal{S}^*_r(g)$ by gluing together spaces from $\mathcal{S}^*_r(g_i)$ (across different choices of i) when the spaces are consistent when restricted to $x_1, \ldots, x_{m-1}$. Once, we have $\mathcal{S}^*_r(g)$, we can immediately obtain $\mathcal{S}^*_r(f)$ by using Φ^{-1}.

By Lemma 6.8, the random invertible linear isomorphism ensures that when we set some of the variables to 0, that the distance between the clusters remains high (equal to m) with high probability. Given access to $\mathcal{W}_{r-1}, \mathcal{KS}^*_1, \ldots, \mathcal{KS}^*_{r-1}, \mathcal{K\overline{S}}_1, \ldots, \mathcal{K\overline{S}}_{r-1}$, we can compute $\mathcal{W}_r$ from Lemma 6.6. Also, from Lemma 6.9, we know all crashing spaces of g_i must intersect $\Phi(\mathcal{W}_r)|_{x_m=\cdots=x_{i-1}=x_{i+1}=\cdots=x_n=0}$. Therefore, we use it as $\mathcal{W}_r$ in the computation of $\mathcal{S}^*_r(g_i)$.

As discussed in the proof of Lemma 6.7, we see that for a vanishing codimension r space $\mathbb{V}(l_1, \ldots, l_r)$ of f, there will be a $\mathbb{V}(x_1 - l_{1a}, \ldots, x_r - l_{ra})$ vanishing space of codimension r of g, where $\forall j \in [r], l_{ja} \in \mathbb{F}[x_{r+1}, \ldots, x_n]$. It is fairly straightforward to see that if g vanishes on $\mathbb{V}(x_1 - l_1, \ldots, x_r - l_r)$ then g_i vanishes on $\mathbb{V}(x_1 - l_{1i}, \ldots, x_r - l_{ri})$ where $l_{ji} = l_j|_{x_m=0,\ldots,x_{i-1}=0,x_{i+1}=0,\ldots,x_n=0}$.

We will now show that if $\mathbb{V}(l_1, \ldots, l_r) \in \mathcal{S}^*_r(f)$, then we have $\mathbb{V}(x_1 - l_{1i}, \ldots, x_r - l_{ri}) \in \mathcal{S}^*_r(g_i)$. As we use the same $\mathcal{W}_r$ projected down for calculations of $\mathcal{S}^*_r(g_i)$, if $\text{span}(l_1, \ldots, l_r)$ doesn't intersect $\mathcal{W}_r$ (as $\mathbb{V}(l_1, \ldots, l_r) \in \mathcal{S}^*_r(f)$), we have $\text{span}(x_1 - l_{1i}, \ldots, x_r - l_{ri})$ doesn't intersect $\Gamma_i(\Phi(\mathcal{W}_r))$. From Lemma 6.11, we know any vanishing space of $\mathcal{S}^*_{r'}(g_i)$ for $r' \in [r-1]$ that doesn't correspond to a space in $\mathcal{S}^*_{r'}(f)$, must intersect $\Gamma_i(\Phi(\mathcal{W}_r))$. Since, $\text{span}(x_1 - l_{1i}, \ldots, x_r - l_{ri})$ doesn't intersect $\Gamma_i(\Phi(\mathcal{W}_r))$, it cannot be blocked by a new minimal vanishing space of g_i. Hence, we have $\mathbb{V}(x_1 - l_{1i}, \ldots, x_r - l_{ri}) \in \mathcal{S}^*_r(g_i)$. We can find $\mathcal{S}^*_r(g_i)$ in $d^{\text{poly}(m)}$ time as described in Lemma 6.7.

To obtain $\mathcal{S}^*_r(g)$ (wlog of the form $\mathbb{V}(x_1 - l_1, \ldots, x_r - l_r)$), we will show how to glue these spaces learned in the low-variate case. To do this, we will look at spaces in $\mathcal{S}^*_r(g_i)$ and $\mathcal{S}^*_r(g_j)$ (for $i \neq j$) and "glue" them if they are consistent in the first $m - 1$ variables. For this to be efficient, it will be very useful to have the property that distinct spaces in $\mathcal{S}^*_r(g_i)$ are distinct when restricted to the first $m - 1$ coordinates. However to make the argument simpler to analyze, we consider and apply another random linear isomorphism Ψ (we will apply these to the spaces in $\mathcal{S}^*_r(g_i)$ for each i) defined as follows: $\forall i < m$, $\Psi(x_i) = x_i$ and $\forall i \in [m, n], \Psi(x_i) = x_i + \beta_{i,r+1}x_{r+1} + \cdots + \beta_{i,m-1}x_{m-1}$ where $\beta_{i,j}$ are sampled independently and uniformly from $[d^n]$. The goal of introducing the map Ψ is to ensure that distinct spaces in $\mathcal{S}^*_r(g_i)$ are distinct when restricted to the first $m-1$ coordinates, and we prove this formally in the claim below.

Claim 6.13. *For all $i \in \{m, \ldots, n\}$, let $\mathbb{V}(l_1, \ldots, l_r)$ and $\mathbb{V}(l'_1, \ldots, l'_r)$ be distinct spaces in $\mathcal{S}^*_r(g_i)$ such that $l_1, \ldots, l_r, l'_1, \ldots, l'_r$ only depend on $x_{r+1}, \ldots, x_{m-1}, x_i$. Then $\mathbb{V}(\Psi(x_1 - l_1)_{x_i=0}, \ldots, \Psi(x_r - l_r)_{x_i=0}) \neq \mathbb{V}(\Psi(x_1 - l'_1)_{x_i=0}, \ldots, \Psi(x_r - l'_r)_{x_i=0})$. In particular*

$$\langle \Psi(l_1)|_{x_i=0}, \ldots, \Psi(l_r)|_{x_i=0} \rangle \neq \langle \Psi(l'_1)|_{x_i=0}, \ldots, \Psi(l'_r)|_{x_i=0} \rangle$$

Proof. Consider 2 distinct elements $\mathbb{V}(x_1 - l_1, \ldots, x_r - l_r)$ and $\mathbb{V}(x_1 - l'_1, \ldots, x_r - l'_r)$ in $\mathcal{S}^*_r(g_i)$, with $l_1, l'_1, \ldots, l_r, l'_r \in \mathbb{F}[x_{r+1}, \ldots, x_{m-1}, x_i]$. Let $l_1 = a_{1,r+1}x_{r+1} + \cdots + a_{1,m-1}x_{m-1} + a_{1,i}x_i$, $\ldots$, $l_r = a_{r,r+1}x_{r+1} + \cdots + a_{r,m-1}x_{m-1} + a_{r,i}x_i$, $l'_1 = a'_{1,r+1}x_{r+1} + \cdots + a'_{1,m-1}x_{m-1} + a'_{1,i}x_i$, $\ldots$, $l'_r = a'_{r,r+1}x_{r+1} + \cdots + a'_{r,m-1}x_{m-1} + a'_{r,i}x_i$. Therefore, we have

$$
\begin{aligned}
l_1 &= (a_{1,r+1} + \beta_{i,r+1}a_{1,i})x_{r+1} + \cdots + (a_{1,m-1} + \beta_{i,m-1}a_{1,i})x_{m-1} + a_{1,i}x_i \\
&\vdots \\
l_r &= (a_{r,r+1} + \beta_{i,r+1}a_{r,i})x_{r+1} + \cdots + (a_{r,m-1} + \beta_{i,m-1}a_{r,i})x_{m-1} + a_{r,i}x_i \\
l'_1 &= (a'_{1,r+1} + \beta_{i,r+1}a'_{1,i})x_{r+1} + \cdots + (a'_{1,m-1} + \beta_{i,m-1}a'_{1,i})x_{m-1} + a'_{1,i}x_i \\
&\vdots \\
l'_r &= (a'_{r,r+1} + \beta_{i,r+1}a'_{r,i})x_{r+1} + \cdots + (a'_{r,m-1} + \beta_{i,m-1}a'_{r,i})x_{m-1} + a'_{r,i}x_i
\end{aligned}
$$

Now, if $\langle \Psi(x_1 - l_1), \ldots, \Psi(x_r - l_r)\rangle \mid_{x_i=0} = \langle \Psi(x_1 - l'_1), \ldots, \Psi(x_r - l'_r)\rangle \mid_{x_i=0}$, we have a system of linear equations in $\beta_{i,r+1}, \ldots, \beta_{i,m-1}$ given by $\forall j \in \{r+1, \ldots, m-1\}$ $\beta_{i,j}(a_{1,i} - a'_{1,i}) = (a_{1,j} - a'_{1,j}), \ldots, \beta_{i,j}(a_{r,i} - a'_{r,i}) = (a_{r,j} - a'_{r,j})$. Since $\langle x_1 - l_1, \ldots, x_r - l_r\rangle$ and $\langle x_1 - l'_1, \ldots, x_r - l'_r\rangle$ are distinct, there is some choice of p, q for which $a_{p,q} \neq a'_{p,q}$ must hold. This gives a nonzero linear equation in $\beta_{i,j}$ (when viewed as formal variables) which must become zero for the specific choice of sampled values. By using Lemma 3.3,the probability this can happen is $\frac{1}{d^n}$ as we choose the $\beta_{i,j}$ from $[d^n]$. From Theorem 4.1, Claim 6.1, and Lemma 6.8 we know the $\mathcal{S}^*_r(g_i) = d^{O(1)}$, and taking the union of all pairs from $\mathcal{S}^*_r(g_i)$ gives us that with probability $1 - o(1)$ no two spaces in any of the $\mathcal{S}^*_r(g_i)$ are equal after applying Ψ and setting x_i to zero. □

As described in Algorithm 2, for each $\mathbb{V}(x_1 - l_{m1}, \ldots, x_r - l_{mr}) \in \mathcal{S}^*_r(g_m)$ we "glue" or combine it with a corresponding $\mathbb{V}(x_1 - l_{i1}, \ldots, x_r - l_{ir}) \in \mathcal{S}^*_r(g_i)$ if they are consistent in first $m-1$ variables, i.e. $\langle \Psi(l_{i1})|_{x_i=0}, \ldots, \Psi(l_{ir})|_{x_i=0}\rangle = \langle \Psi(l_{m1})|_{x_m=0}, \ldots, \Psi(l_{mr})|_{x_m=0}\rangle$. We each fixed space in $\mathcal{S}^*_r(g_m)$ with high probability there is a unique space in $\mathcal{S}^*_r(g_i)$ where this happens by the above claim. Note that every space in $\mathcal{S}^*_r(g)$ corresponds to some unique space in $\mathcal{S}^*_r(g_m)$ (since the spaces in $\mathcal{S}^*_r(g_m)$ are distinct restricted to first $m-1$ coordinates). To recover the spaces of $\mathcal{S}^*_r(g)$ with information for all coordinates, for each i, we use the information present in the glued space in $\mathcal{S}^*_r(g_i)$ to recover the information for the coordinate corresponding to x_i. Thus we obtain all spaces $\mathbb{V}(l_1, \ldots, l_r)$ on which g vanishes and use Φ^{-1} to obtain $\mathcal{S}_r(f)$. □

Algorithm 2 Computing Vanishing Codimension r Subspaces

Input: Black-box access to circuit C of form $\Sigma\Pi\Sigma(k)$ computing polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ with properties as described in Theorem 6.2, $\mathcal{W}_r$, $\mathcal{S}^*_1, \ldots, \mathcal{S}^*_{r-1}$

function $\mathcal{S}^*_r(f)$

Sample n^2 random values $\alpha_{ij}; i, j \in [n]$ uniformly from $\{1, \ldots, d^n\}$, and use them to define n linear forms $l'_i = \sum_{j=1}^{n} \alpha_{ij} x_j$. Check if they are independent, otherwise repeat. Define isomorphism Φ such that for all $i \in [n]$, $\Phi(x_i) := l'_i$. Let $g = \Phi(f) = f(\Phi(x))$.

Set $t = c_k + 2^{k^2}\mathcal{R}_{\mathbb{F}}(k, d) \log d$ and $m = 2t \log d + 2k + 2$. For $i \in [m, n]$, Obtain $g_i = g_{x_m=0,\ldots,x_{i-1}=0,x_{i-1}=0,\ldots,x_n=0}$

For each $g_i, i \in [m, n]$ and $r' \in [r]$, Compute $\mathcal{S}^*_{r'}(g_i)$ using Algorithm 1 with $\Phi(\mathcal{W}_{r'})|_{x_m=0,\ldots,x_{i-1}=0,x_{i-1}=0,\ldots,x_n=0}$ as the LDICR space.

We now describe how to glue these spaces across g_i.

Consider an isomorphism Ψ obtained as follows. Sample $(n - m + 1) \times (m - r)$ random values $\beta_{i,j}$ where $i \in [m, n], j \in [r, m-1]$ uniformly from $\{1, \ldots, d^n\}$. For all $i \in [1, m-1]$, let $\Psi(x_i) = x_i$ and for all $i \in [m, n]$, let $\Psi(x_i) = x_i + \beta_{i,r}x_{r+1} + \cdots + \beta_{i,m-1}x_{m-1}$.

for $\mathbb{V}(x_1 - l_{m1}, \ldots, x_r - l_{mr}) \in \mathcal{S}^*_r(g_m)$ **do**

$l_{a1} := l_{m1}, l_{a2} := l_{m2}, \ldots, l_{ar} := l_{mr}$

for $i \in \{m+1, \ldots, n\}$ **do**

Search for $\mathbb{V}(x_1 - l_{i1}, \ldots, x_r - l_{ir})$ such that

$$\langle \Psi(x_1 - l_{m1})|_{x_m=0}, \ldots, \Psi(x_r - l_{mr})|_{x_m=0} \rangle = \langle \Psi(x_1 - l_{i1})|_{x_i=0}, \ldots \Psi(x_r - l_{ir})|_{x_i=0} \rangle$$

If multiple such spaces are found, break out of the loop, and go to the next space in the outer loop.

If only one such space is found then update $l_{a1} = l_{a1} - \alpha_1 x_i, \ldots, l_{ar} = l_{ar} - \alpha_r x_i$ where $\alpha_1, \ldots, \alpha_r$ are coefficients of x_i in $l_{i1}, \ldots, l_{ir}$ respectively.

Add $\mathbb{V}(x_1 - l_{a1}, x_2 - l_{a2}, \ldots, x_r - l_{ar})$ to $\mathcal{S}^*_r(g)$

For each $\mathbb{V}(l_1, \ldots, l_r) \in \mathcal{S}^*_r(g)$, Verify f vanishes on $\mathbb{V}(\Phi^{-1}(l_1), \ldots, \Phi^{-1}(l_r))$. Output the set thus obtained, as $\mathcal{S}^*_r(f)$.

6.3 Computing $\overline{\mathcal{S}}_r$

In this section, we will show how to compute $\overline{\mathcal{S}}_r(f)$ spaces as defined in Definition 15, given that we have already learned $\mathcal{W}_r, \mathcal{S}^*_1(f), \mathcal{S}^*_2(f), \ldots, \mathcal{S}^*_r(f)$.

Lemma 6.14. *Let $t = c_k + 2^{k^2}\mathcal{R}_{\mathbb{F}}(k, d) \log d$. Let f be a n-variate, degree-d polynomial in $\mathbb{F}[x_1, \ldots, x_n]$ that is computed by a $\Sigma\Pi\Sigma(k)$ circuit $C = G \times (T_1 + \cdots + T_k)$ with $\gcd(T_1, \ldots, T_k) = 1$ such that it has a $2t \log d + 3k$-cluster representation $C = G \times (C_1 + \cdots + C_s)$ as defined in Lemma 5.7. Let r be in $[s]$. Then, there exists an algorithm(Algorithm 3) that runs in time $\mathrm{poly}(n, d^{\mathrm{poly}(t)})$ and with probability $1 - o(1)$, it outputs a set of spaces $\overline{\mathcal{S}}_r$ of size $\mathrm{poly}(d)$ such that it satisfies the properties described in Definition 15.*

We will break this further into two separate lemmas which will show the first and second properties of $\overline{\mathcal{S}}_r$ spaces, respectively. As described in Algorithm 3, $\overline{\mathcal{S}}_r$ spaces are computed as a union of spaces $\overline{\mathcal{S}}_r := \overline{\mathcal{S}}^{(1)}_r \cup \overline{\mathcal{S}}^{(2)}_r \cup \cdots \cup \overline{\mathcal{S}}^{(r)}_r$. We will also make grow $\mathcal{W}_r$ slightly (which we will be able to do since we assume we have learnt $\mathcal{S}^*_r$, so that is also now intersects kernels of all regular spaces of codimension r (and not just those of codimension $r-1$). To not overload notation, we will continue to call this space $\mathcal{W}_r$.

Update of $\mathcal{W}_r$: Once we have computed $\mathcal{S}_r^*$, we can use it to update $\mathcal{W}_r$ so that any space in $\mathcal{S}_r^{reg}$ also has to intersect $\mathcal{W}_r$. To do this, we look at all maximal sets of independent spaces in $\mathcal{S}_r^*$ that span a space $\mathcal{W}_r'$ of dimension at most $r \cdot 2^{k^2}\mathcal{R}_{\mathbb{F}}(k,d)\log d$ which is at most $\text{poly}(\log d)$ for a suitable polynomial from Theorem 3.5. From Lemma 5.4, we know that there is right choice of $\mathcal{W}_r'$ that it would intersects all spaces in $\mathcal{S}_r^{reg}$. As $|\mathcal{S}_r^*|$ is $d^{O(1)}$, the number of choice of $\mathcal{W}_r'$ is $d^{\text{poly}(\log d)}$. Wlog we assume that we chose the right $\mathcal{W}_r'$ (our algorithm will iterate over all choices of $\mathcal{W}_r'$, and eventually all the wrong choices get pruned out by a final PIT step at the end. This adds a factor of $d^{\text{poly}(\log d)}$ to the running time, which we can handle). The new $\mathcal{W}_r$ is obtained by taking the span of $\mathcal{W}_r'$ with the original $\mathcal{W}_r$, and this increasing its dimension by at most $r \cdot 2^{k^2}\mathcal{R}_{\mathbb{F}}(k,d)\log d$. $\mathcal{W}_r$ now also intersects all kernels of spaces $\mathcal{S}_r^{reg}$. We will assume this going forward for computation of $\overline{\mathcal{S}}_r$.

Lemma 6.15. *Let f be a polynomial and C be a circuit computing it as described in Lemma 6.14. Then, there exists an algorithm that runs in time $\text{poly}(n, d^{\text{poly}(t)})$ and with probability $1-o(1)$, it outputs a set of spaces $\overline{\mathcal{S}}_r^{(1)}$ of size $\text{poly}(d)$ such that it satisfies the property 1 described in Definition 15, i.e. for every space $\mathbb{V}(l_1,\ldots,l_k)$ of codimension r, $l_i \in \text{Lin}(T_i)$, and such that $\text{span}(l_1,\ldots,l_k)\cap\mathcal{W}_r = \{0\}$. Then there is a space V in a $\overline{\mathcal{S}}_r^{(1)}$ such that $Ker(V) \subseteq \text{span}(l_1\ldots,l_k)$, $f \mod \langle l_1,\ldots,l_k\rangle = 0$ and $Ker(V)$ contains at least one of the l_i's.*

Proof. We will first show that there exists an algorithm that shows how to grow vanishing kernels to larger spaces which have interesting properties (such as containing a linear form of the circuit.)

Claim 6.16. *There exists an algorithm such that given any vanishing kernel W of dimension at most r such that $W\cap\mathcal{W}_r = \{0\}$, blackbox access to f and $\mathcal{W}_r$, it runs in $\text{poly}(n, d^{\text{poly}(t)})$-time and outputs a $\text{poly}(d)$-sized set $\overline{\mathcal{S}}$ of spaces of dimension at most r with the following properties:*

- $\forall X \in \overline{\mathcal{S}}, W \subset X$
- *Let U be any r dimension space which is of the form $\text{span}(l_1,\ldots,l_k)$ for $l_i \in \text{Lin}(T_i)$, such that $U\cap\mathcal{W}_r = \{0\}$. If $W \subset U$ with $C \mod \langle W\rangle = 0$ and if for all $i\in[k], l_i \notin W$ then $\exists X \in \overline{\mathcal{S}}, X \subseteq U$.*

Proof. Given access to W s.t $\dim(W) = r'$, we pick a random $(r'-1)$-dimensional subspace $V \subset W$ and a linear form l' such that $\text{span}(l', V) = W$. Next, consider the circuit $C \mod \langle V\rangle$. As W doesn't contain any regular space, as it doesn't intersect $\mathcal{W}_r$, l' will divide $\gcd(C \mod \langle V\rangle)$ and not $\text{sim}(C \mod \langle V\rangle)$. Let the multiplicity of l' in $C \mod \langle V\rangle$ be e, which we can find by factoring $C \mod \langle V\rangle$ using Lemma 3.9. Let $\overline{C} := C \mod \langle V\rangle/(l')^e$. We first argue that $\overline{C}$ will have a structure similar to that of C and in fact, $(\mathcal{W}_r \mod V)$ will satisfy the required properties of $\mathcal{W}_r$ for this. The algorithm will compute $\mathcal{S}_a^*(\overline{C})$ for all $a \in [1, r-r'+1]$ using Lemma 6.12, and then take the span of each of those spaces with W (in other words we grow W by dimension at least 1) and this will be the set of spaces $\overline{\mathcal{S}}$.

We first argue $\overline{C}$ **has a** $2t\log d + 2k$**-cluster representation** with the fan-in s. Since W is not a regular space (as it doesn't intersect $\mathcal{W}_r$), by definition $\text{sim}(C')$ will not be divisible by l' and hence $\overline{C}$ is still a $\Sigma\Pi\Sigma(k)$ circuit. Since V is chosen randomly from W, with high probability the linear forms in the T_i's won't vanish modulo V. Let C_i, C_j be 2 clusters in the cluster representation of C. We know $\Delta(C_i,C_j) \geq 2t\log d + 3k$. Dividing by l', doesn't decrease $\Delta(C_i,C_j)$ as l' only divides $gcd(C \mod V)$. Therefore, $\Delta(C_i,C_j)$ can decrease only if two independent linear forms become the same mod V. We argue this happens only if the two-dimensional space spanned by these linear forms lies inside W. If the space is outside and doesn't intersect V, they stay independent mod V.

If the space intersects W in a 1-dimensional line, then they become the same only if V contains this line, which happens with $o(1)$ probability. Thus, the only linear forms lost from $\text{sim}((C_i + C_j) \mod V)$ are those that are contained in W, which has dimension $r' \leq k$. Hence, for any 2 clusters i, j in $\overline{C}$, $\Delta(C_i, C_j) \geq 2t \log d + 2k$.

We will first show that any crashing space of dimension at most $r-2$ of $\overline{C}$ will intersect $\mathcal{W}_r \mod V$. Therefore, we can use the same $\mathcal{W}_r \mod V$ to play the role of $\mathcal{W}_r$ for computation of $\mathcal{S}^*$ in Lemma 6.12.

Any Crashing space of $\overline{C}$ intersects $\mathcal{W}_r \mod V$: We will show that any crashing space $X \in \mathscr{C}_{\leq r-r'-2,c_k}(\overline{C})$ would intersect $\mathcal{W}_r \mod V$. For the sake of contradiction, assume there is a crashing space $X \in \mathscr{C}_{\leq r-r'-2,c_k}(\overline{C})$ that doesn't intersect $\mathcal{W}_r \mod V$. Clearly, $\text{span}(X, V)$ (this is that subspace of $\mathbb{F}^n$ which contains V and such that when we go mod V it equals X) is a crashing space for C, i.e. $\text{span}(X, V) \in \mathscr{C}_{\leq r-2,c_k}(C)$, as l' doesn't divide the $\text{sim}(C \mod V)$ and only the gcd and therefore division by it, doesn't affect the rank of the simple part. Since, $\text{span}(X, V) \in \mathscr{C}_{\leq r-2,c_k}(C)$, it must intersect $\mathcal{W}_r$ from Lemma 6.6. Recall $V \cap \mathcal{W}_r = \{0\}$ as V is a subspace of W and W only intersects $\mathcal{W}_r$ at $\{0\}$. Since $\text{span}(X, V) \mod V = X$, thus it most hold that the space which means the intersection of X and $\mathcal{W}_r \mod V$ is non-zero, which is a contradiction to the assumption. Hence, all $X \in \mathscr{C}_{\leq r-r'-2,c_k}(\overline{C})$ would intersect $\mathcal{W}_r \mod V$.

Thus, we can compute the spaces $\mathcal{S}^*_a(\overline{C})$ for all $a \in [1, r-r'+1]$ using Lemma 6.12 with $\mathcal{W}_r \mod V$ playing the role of $\mathcal{W}_r$. We will now show that the conclusions of the claim hold. The first conclusion is immediate. Let U be as in the assumption of the second conclusion. We will now argue that one of these spaces in $\mathcal{S}^*_a(\overline{C})$ is a subspace of $U \mod \langle V \rangle$ or is equal to $U \mod \langle V \rangle$. Notice that this will suffice in proving the claim. Next, we show that $\overline{C}$ vanishes mod $(U \mod \langle V \rangle)$. This suffices as $U \cap \mathcal{W}_r = \{0\}$.

Notice that $C \mod V$ vanishes mod $(U \mod \langle V \rangle)$. To obtain $\overline{C}$, we divided $C \mod V$ by $(l')^e$. As argued earlier, $\overline{C}$ is a $\Sigma\Pi\Sigma(k)$ circuit.

We will show that this division does not impact the vanishing property. We know from definition of $U = \text{span}(l_1, \ldots, l_k)$, the linear forms l_i divide their respective T_i in C. We will show that $l_i \mod V$ continues to divide $T_i \mod V$ for each i even after the division. For this, it suffices to show that l_i does not become equal to l' when we go modulo V, and this is what we will show now.

Since by assumption, none of the l_i's lie inside W it immediately follows that when we go mod $\langle V \rangle$, they will not be in $\text{span}(l')$. Thus division by l' does not affect their presence in the T_i's and if we set $U \mod V$ to zero, each $l_i \mod V$ will get set to zero, and hence each of the gates in $\overline{C}$ vanishes and hence $\overline{C}$ vanishes mod $U \mod \langle V \rangle$.

Hence, $U \mod V$ is a space on which $\overline{C}$ vanishes on, and it doesn't intersect $\mathcal{W}_r \mod V$ which we use in the computation of $\mathcal{S}^*(\overline{C})$. Thus $U \mod V$ is a vanishing kernel, but it may not be a minimal vanishing kernel of $\overline{C}$. We conclude that either $U \mod V$ or some subspace of it must be computed in $\mathcal{S}^*(\overline{C})$.

As $\overline{C}$ doesn't vanish on l' (we divided out the multiplicity of l' in $C \mod V$), the spaces computed will be of dimension at least 1 outside W and when extended by W will contain W. Thus, the subspace computed of $U \mod V$ will extend to a subspace of U strictly containing W. Hence, we can lift W to a strictly larger subspace of U. The time complexity is the time complexity of computing $\mathcal{S}^*_{\leq r-r'+2}(\overline{C})$ which is $\text{poly}(n, d^{\text{poly}(t)})$ from Lemma 6.12. □

Our final algorithm will run the algorithm from Claim 6.16 using the spaces from $\mathcal{KS}^*_1, \ldots, \mathcal{KS}^*_{r-1}$ which don't intersect $\mathcal{W}_r$, playing the role of W, to learn a new set of "grown" spaces, and then run the algorithm again on this new set of slightly larger spaces. We do this iteratively $r-1$ times. For every space $U = \text{span}(l_1, \ldots, l_k)$, in each iteration we either learn a larger space in U or we learn a

subspace containing one of the l_i. Thus, at the end we would have either learned a subspace which contains one of the l_i's, or the space U itself, and hence proving the lemma. The runtime analysis easily follows from the runtime of the algorithm in the claim.

□

Lemma 6.17. *Let f be a n-variate, degree-d polynomial in $\mathbb{F}[x_1, \ldots, x_n]$, as described in Lemma 6.14. Then there exist an algorithm that computes a set $\overline{\mathcal{S}}_r$ such that it satisfies property 2 of Definition 15, i.e. If we consider any crashing space $W \in \mathscr{C}_{r-1,c_k}$ such that it doesn't intersect $\mathcal{W}_r$. Let l be any linear form and e a positive integer such that $l^e | \gcd(C \mod W)$, $\text{span}(W, l) \cap \mathcal{W}_r = \{0\}$, and $l \nmid \text{sim}(C \mod W)$. We say such an l is saturated by a space V, if it has the following properties.*

1. *$Ker(V) \subset \text{span}(W, l)$. Let $W' = Ker(V) \cap W$.*
2. *There exists e linear forms $l'_1, \ldots, l'_e$, each of the form $l + l'$, where $l' \in W$ such that $\gcd(C \mod W')$ is divisible by $\prod_{i \in [e]} (l'_i \mod W')$.*

Then, for any crashing space $W \in \mathscr{C}_{r-1,c_k}$ which doesn't intersect $\mathcal{W}_r$, the dimension of the span of linear forms in $\gcd(C \mod W)$ that are not saturated by any space in $\overline{\mathcal{S}}_r$ is at most k.

Proof. We will compute the set $\overline{\mathcal{S}}_r$ iteratively as a union of $\overline{\mathcal{S}}_r^{(1)}, \ldots, \overline{\mathcal{S}}_r^{(r)}$. We already saw how we can compute $\overline{\mathcal{S}}_r^{(1)}$ in Lemma 6.15. For every pair of spaces in $\mathcal{K}\overline{\mathcal{S}}_r^{(1)}$, the algorithm takes their intersection W' and considers $C \mod W'$. It then computes $\overline{\mathcal{S}}^{(1)}_{\leq r-\dim(W')}(C \mod W')$ and then takes the span of each space in $\mathcal{K}\overline{\mathcal{S}}^{(1)}_{\leq r-\dim(W')}(C \mod W')$ with W' to obtain $\mathcal{K}\overline{\mathcal{S}}_r^{(2)}(C)$ and hence $\overline{\mathcal{S}}_r^{(2)}$. We repeat this step r times, and use the union of these sets as our $\overline{\mathcal{S}}_r$.

In order to analyze why this algorithm works, we will first state and prove a claim which shows that $\overline{\mathcal{S}}_r^{(1)}$ already has some interesting saturation properties.

Let $W \in \mathscr{C}_{r-1,c_k}$. Thus $\text{rank}(\text{sim}(C \mod W)) \leq c_k$. When we go modulo W, some of the T_i might vanish. Wlog suppose that $T_1, \ldots, T_{k'}$ do not vanish mod W and $T_{k'+1}, \ldots, T_k$ do vanish mod W for some $k' \geq 2$.Thus $C \mod W$ is of the form $C' := G' \times (T'_1 + \cdots + T'_{k'})$.

For each $i \in [k']$, the linear forms in $\text{Lin}(T_i)$ can be partitioned into $A_i \cup B_i$ such that the linear forms that become part of G' are in A_i, and the linear forms that became part of T'_i are in B_i.

$\forall i \in [k'+1, k]$, as T_i vanishes mod W, there must exist a linear form l_i dividing T_i such that $l_i \in W$. There may be multiple such linear forms in every gate that vanish modulo W, but we fix any one such linear form per gate.

We focus on the linear forms in A_i's as they are the ones we are concerned for moving into the G'. Consider any $l \in \gcd(C \mod W)$, and let e be the multiplicity of l in $\gcd(C \mod W)$. We also assume that l is such that $\text{span}(W, l) \cap \mathcal{W}_r = \{0\}$, and $l \nmid \text{sim}(C \mod W)$. Then there are e different k'-tuples $(l_{11}, \ldots, l_{1k'}), \ldots, (l_{e1}, \ldots, l_{ek'})$, such that any $l_{ij} \in A_j$ and is of the form $\alpha_{ij} l + l'_{ij}$ where $l'_{ij} \in W$.

We consider such e different k-tuples of the form $(l_{i1}, \ldots, l_{ik'}, l_{k'+1}, \ldots, l_k)$ for $i \in [e]$.

We first show that for any such l (corresponding to the above tuples), either we have a space V in $\overline{\mathcal{S}}_r^{(1)}$ satisfying the properties of the lemma (i.e. V saturates l), or we have a space $V \in \mathcal{K}\overline{\mathcal{S}}_r^{(1)}$ such that at least one of $l_{k'+1}, \ldots, l_k$ is in V. Note that of V doesn't saturate l, we still make progress by learning one of the l_i within V, since distinct choices of V for distinct choices of l will allow us to start learning some of the l_i (we actually learn subspaces of W containing one of the l_i) by taking intersections.

Claim 6.18. *Let W be as above. Consider an $l \in \gcd(C \mod W)$ such that $\text{span}(W, l) \cap \mathcal{W}_r = \{0\}$ and $l \nmid \text{sim}(C \mod W)$. Then there exists a space V in $\mathcal{K}\overline{\mathcal{S}}_r^{(1)}$ such that*

1. *either V saturates l*

2. *or $\exists j \in [k'+1, k]$ s.t. $l_j \in W$ and $l_j \in V$.*

Proof. Observe that C vanishes modulo $\text{span}(W, l)$. Also, $\text{span}(l_{i1}, \ldots, l_{ik'}, l_{k'+1}, \ldots, l_k) \subseteq \text{span}(W, l)$ and C vanishes mod it (since every gate vanishes). Since this space also does not intersect $\mathcal{W}_r$, at least one subspace of $\text{span}(l_{i1}, \ldots, l_{ik'}, l_{k'+1}, \ldots, l_k)$ (any minimal vanishing kernel) must be in $\mathcal{KS}^*_{\leq r}$. Basically, all the tuples from $(l_{11}, \ldots, l_{1k'}, l_{k'+1}, \ldots, l_k), \ldots, (l_{e1}, \ldots, l_{ek'}, l_{k'+1}, \ldots, l_k)$ whose span is a minimal vanishing kernel, are in $\mathcal{KS}^*_{\leq r}$. Thus even if one of the tuples spans a minimal vanishing kernel, then since $\mathcal{K}\overline{\mathcal{S}}^{(1)}_r$ contains $\mathcal{KS}^*_{\leq r}$, thus choosing V to be span of the linear forms satisfies property (2) of the claim and we are done.

Therefore, we are left to deal with the case where $\mathbb{V}(l_{i1}, \ldots, l_{ik'}, l_{k'+1}, \ldots, l_k)$ is not a maximal vanishing subspace for all choices of $i \in [e]$.

Since $\text{span}(l_{i1}, \ldots, l_{ik'}, l_{k'+1}, \ldots, l_k)$ doesn't intersect $\mathcal{W}_r$ and $l \nmid \text{sim}(C \mod W)$, the span cannot be contained in a kernel of a regular space. Therefore, from Lemma 6.15, we know that there will be a subspace V of $\text{span}(l_{i1}, \ldots, l_{ik'}, l_{k'+1}, \ldots, l_k)$ in $\mathcal{K}\overline{\mathcal{S}}^{(1)}_r$, such that at least one of the linear forms in $(l_{i1}, \ldots, l_{ik'}, l_{k'+1}, \ldots, l_k)$ will be in V. Wlog, let V be the largest such subspace of $\text{span}(W, l)$ learned in $\mathcal{K}\overline{\mathcal{S}}^{(1)}_r$. If V is $\text{span}(W, l)$, then we are already done (property (2) is satisfied). So assume V is not $\text{span}(W, l)$. This means at some point in computation of $\mathcal{S}^{(1)}_r$, the computation learned V and did not learn a larger space even though the computation is iterated $r-1$ times. We will now argue that the only way this can happen is that V either satisfies property (1) or (2).

If V satisfies property (1) of the current claim, then we are done. If any of the $l_{k'+1}, \ldots, l_k$ are in V, then we are done as well. Let us assume that V doesn't satisfy either property. We will strengthen the analysis of Lemma 6.15 and show that, we will have learned a larger subspace of $\text{span}(W, l)$ in $\mathcal{K}\overline{\mathcal{S}}^{(1)}_r$ containing V, contradicting the maximality of V.

Consider a random $(\dim(V)-1)$-dimensional subspace of V, which we call V' and a linear form l' such that $V = \text{span}(V', l')$. Let e' be the multiplicity of l' in $C \mod V'$. Consider $C' = (C \mod V')/(l')^{e'}$ and look at it's $\mathcal{S}^*_{\leq r-\dim(V)+1}$ spaces. As discussed in Claim 6.16, we have that C' is computed by $\Sigma\Pi\Sigma(k)$ circuit with almost the same cluster structure as C and any crashing space of C' of dimension at most $r-2$ must also intersect $\mathcal{W}_r$. Therefore, we can use the same $\mathcal{W}_r$ for computation of $\mathcal{S}^*_{\leq r-\dim(V)+1}(C')$.

As property (1) is not satisfied by V, we must have $e' < e$. Note that $l'^{e'}$ divides the gcd and not the simple part of the circuit, and hence it divides each gate of the circuit. Since each gate T_j ($j \in [k']$) had e linear forms $\{l_{1j}, \ldots, l_{ej}\}$, when we divide by $l'^{e'}$ after going mod V', at least one of the e linear forms "survives" in the gate. In other words, it gets mapped modulo V', but does not get eliminated by division. WLOG let the surviving linear form in gate T_j for $j \in [k']$ be $l_{1j} \mod V'$. As $l_{k'+1}, \ldots, l_k$ are not in V by assumption, hence when we go modulo V', they do not get mapped to l'. Hence the division by $l'^{e'}$ does not impact these linear forms as well, and they survive. Thus for $i \in \{k'+1, \ldots, k\}$, T_i is is still divisible by $l_i \mod V'$ even after the division.

Consider $(l_{11}, \ldots, l_{1k'}, l_{k'+1}, \ldots, l_k)$. From the above discussion we know that in every gate T_j in C', we still have either l_{1j} ($j \in [k']$) or l_j ($j \in [k'+1, k]$) dividing the gate. Therefore, $\mathbb{V}(l_{11} \mod V', \ldots, l_{1k'} \mod V', l_{k'+1} \mod V', \ldots, l_k \mod V')$ must be a vanishing space of C'. Therefore, the vanishing space must have been combined with V to give a larger space than V (but still containing in $\text{span}(W, l)$) in $\overline{\mathcal{S}}^{(1)}_r$. That gives a contradiction to maximality of V. □

Consider any crashing space $W \in \mathscr{C}_{r-1,c_k}$ which doesn't intersect $\mathcal{W}_r$. Let $T_{k'+1}, \ldots, T_k$ be the gates which vanish modulo W and for each of these gates T_j, let us fix a linear form l_j dividing

T_j which vanishes on W. Let $l \in \gcd(C \mod W)$ such that $\text{span}(W, l) \cap \mathcal{W}_r = \{0\}$ and $l \nmid \text{sim}(C \mod W)$. By the above claim, we either learn $V \in \text{span}(W, l)$ that saturates l, or else V contains one of $\{l_{k'+1}, \ldots, l_k\}$. Suppose there are at most k independent linear forms l that are not saturated by some $V \in \overline{\mathcal{S}}_r^{(1)}$, then we are done. Otherwise there are $k+1$ distinct and linearly independent mod W choices of l $(l'_1, \ldots, l'_{k+1})$, such that for each $i \in [k]$ we learn some $V_i \in \overline{\mathcal{S}}_r^{(1)}$ such that $V_i \in \text{span}(W, l'_i)$ and V_i contains one of $\{l_{k'+1}, \ldots, l_k\}$. Observe that distinct i and j, V_i and V_j are distinct and their intersection is a subspace of W. By the pigeon hole principle, there must exist distinct V_i and V_j that both contain the same choice of linear form in $\{l_{k'+1}, \ldots, l_k\}$. Thus their intersection is a subspace of W' of W that contains one of $\{l_{k'+1}, \ldots, l_k\}$.

This we can learn W', by looking at intersections of spaces in $\mathcal{K}\overline{\mathcal{S}}_r^{(1)}$. Recall that $\mathcal{K}\overline{\mathcal{S}}_r^{(2)}$ is obtained by looking at every pair of spaces in $\mathcal{K}\overline{\mathcal{S}}_r^{(1)}$, taking their intersection W', and computing $\mathcal{K}\overline{\mathcal{S}}_{r-\dim(W')}^{(1)}$ for $C \mod W'$. Thus for the right choice of W' which corresponds to a subspace of W containing one of $\{l_{k'+1}, \ldots, l_k\}$, our algorithm will compute $\mathcal{K}\overline{\mathcal{S}}_r^{(1)}$ for $C \mod W'$. This will give us a space which either saturates all but k independent linear forms (when we consider its span along with W') or a subspace of $W \mod W'$ which contains another non-zero linear form from $\{l_{k'+1}, \ldots, l_k\} \mod W'$. Since we iterate this process r times, we will either recover a saturating space that saturates all but k independent linear forms or recover a subspace $\overline{W}$ of W that contains all of $\{l_{k'+1}, \ldots, l_k\}$. When we consider $C \mod \overline{W}$ then we get a saturating space by Claim 6.18. □

Algorithm 3 Computing codimension r Totally-Special vanishing spaces

Input: Black-box access to circuit C of form $\Sigma\Pi\Sigma(k)$ computing polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$, $\mathcal{W}_r$, $\mathcal{S}_1^*, \ldots, \mathcal{S}_r^*, \overline{\mathcal{S}}_1, \ldots, \overline{\mathcal{S}}_{r-1}$

function $\overline{\mathcal{S}}_r(C)$
  if $r \leq 2$ **then return** ϕ
  $\overline{\mathcal{S}}_r^{(1)} := \mathcal{S}_{\leq r}^*$
  for $r' \in [1, r-1]$ **do**
    for $\mathbb{V}(l_1, \ldots, l_{r'}) \in (\mathcal{S}_{r'}^*(f) \cup \overline{\mathcal{S}}_{r'}(f))$ **do**
      Pick a random vector space V of dimension $r'-1$ in $\text{span}(l_1, \ldots, l_{r'})$
      Let $C' := C \mod V$ and any l' such that $\text{span}(l', V) := \text{span}(l_1, \ldots, l_{r'})$.
      Let e be the multiplicity of l' in $\text{Lin}(C')$
      Let $\overline{C} := (C' / (l')^e)$.
      Compute $\mathcal{S}_{r-r'+1}^*(\overline{C})$ with $\mathcal{W}_r \mod V$ as the LDICR space.
      for $\mathbb{V}(l'_1, \ldots, l'_{r-r'+1}) \in \mathcal{S}_{r-r'+1}^*(\overline{C}))$ **do**
        if $l' \in \text{span}(l'_1, \ldots, l'_{r-r'+1})$ **then**
          Add $\mathbb{V}(l_1, \ldots, l_{r'}, l'_1, \ldots, l'_{r-r'})$ to $\overline{\mathcal{S}}_r^{(1)}$
  for $i \leftarrow [2, r]$ **do**
    for $V_1, V_2 \in \mathcal{K}\overline{\mathcal{S}}_r^{(i-1)}$ **do**
      Let $W = V_1 \cap V_2$. If $W = \{0\}$, continue.
      Compute $\overline{\mathcal{S}}_r^{(1)}$ for $C \mod W$.
      for $V \in \mathcal{K}\overline{\mathcal{S}}_r^{(1)}(C \mod W)$ **do**
        Add $\mathbb{V}(V, W)$ to $\overline{\mathcal{S}}_r^{(i)}$.
  Output $\overline{\mathcal{S}}_r := \overline{\mathcal{S}}_r^{(1)} \cup \cdots \cup \overline{\mathcal{S}}_r^{(r)}$

6.4 Learning linear forms when $SCS(r)$ property is satisfied

Lemma 6.19. *Let $t = 2^{2k^2}\mathcal{R}_{\mathbb{F}}(k,d)(k^4\log d)^{k-3}$. Let f be a n-variate, degree-d polynomial in $\mathbb{F}[x_1,\ldots,x_n]$ that is computed by a $\Sigma\Pi\Sigma(k)$ circuit $C = G\times(T_1+\cdots+T_k)$ with $gcd(T_1,\ldots,T_k)=1$ such that it has a $2t\log d+2k$-cluster representation $C = G\times(C_1+\cdots+C_s)$ as defined in Lemma 5.7. Let r be in $[s]$ and $t_r := 2^{2k^2}\mathcal{R}_{\mathbb{F}}(k,d)(k^4\log d)^{r-3}$. Also, for any c_{cand} and $\lambda = c_{cand}+2^k\cdot(2t_r\log d+2k)$, we have $\forall i\in[k], \dim(\text{span}(\text{Lin}(T_i)))\geq\lambda$. Assume the circuit satisfies the $SCS(r)$ property for some $r\in[s]$. Then, given access to $\mathcal{W}_{r-1}$ and the sets $\mathcal{KS}^*_1,\ldots,\mathcal{KS}^*_{r-1},\mathcal{K\overline{S}}_1,\ldots,\mathcal{K\overline{S}}_{r-1}$, there exists an algorithm that computes a set of linear forms $\mathcal{L}_{cand}$ such that $|\mathcal{L}_{cand}| = d^{O(1)}$ and $\exists j\in[k]$ such that $\dim(\text{span}(\text{Lin}(T_j)\cap\mathcal{L}_{cand}))\geq c_{cand}$ in time $\text{poly}(n,d)$.*

Proof. Let $r = \max(\dim(W_a),\dim(W_b))$, where W_a and W_b are the smallest possible totally independent spaces in $\mathscr{C}_{\leq r-2,t_r}$ contained in spaces in $\mathscr{C}_{\leq r-2,c_k}$ such that such that $C \mod \langle W_a\rangle = C_j \mod \langle W_a\rangle$ and $C \mod \langle W_b\rangle = C_j \mod \langle W_b\rangle$.

We can compute $\mathcal{P}_{r-2}$ as described in Lemma 6.3, and therefore W_a and W_b would be two spaces in $\mathcal{P}_{r-2}$. The rank requirement of Lemma 6.3, is satisfied by at least two clusters in the circuit with enough distance. We guess two spaces in $\mathcal{P}_{r-2}$, and for the correct guess we have access to W_a and W_b, such that $C \mod \langle W_a\rangle = C_j \mod \langle W_a\rangle$ and $C \mod \langle W_b\rangle = C_j \mod \langle W_b\rangle$. Using Lemma 3.9, we have access to the linear factors of $C_j \mod \langle W_a\rangle$ and $C_j \mod \langle W_b\rangle$. Consider any linear form in $l|\gcd(C_j)$. It can be written as $l = l' + l_a + l_b$ where l_a and l_b are linear forms in W_a and W_b. So, we find linear forms l_1, l_2 in $\text{Lin}(C_j \mod \langle W_a\rangle)$ and $\text{Lin}(C_j \mod \langle W_b\rangle)$ such that $l_1 \mod \langle W_b\rangle = l_2 \mod \langle W_a\rangle$. There will be $l_1 = l' + l_b$ and $l_2 = l' + l_a$ in the choices. From these, we can recover $l = l_1 + l_2 - (l_1 \mod \langle W_b\rangle)$. Thus, we can find all linear forms l such that $l|\gcd(C_j)$.

Since C has a $2t_r\log d+2k$ cluster representation, we know $\text{rank}(\text{sim}(C_j))\leq 2^k\cdot(2t_r\log d+2k)$. Since, $\dim(\text{span}(\text{Lin}(T_i)))\geq c_{cand}+2^k\cdot(2t_r\log d+2k)$, we have $\dim(\text{span}(\text{Lin}(\gcd(C_j))))\geq c_{cand}$ and hence, we learn c_{cand} independent linear forms from a gate in C. □

We will divide the computation of $\mathcal{S}^*_r(f)$ into two parts, where we first compute it when the number of variables (n) is small, such that $n = t_r$ and then do it in the general case.

7 Using $\mathcal{S}^*_r(f)$ and $\overline{\mathcal{S}}_r(f)$ to learn some linear forms appearing in C

Armed with $\mathcal{W}_r$, $\mathcal{S}^*_r$ and $\overline{\mathcal{S}}_r$ for various choices of r, we then show that considering intersections of the kernels of spaces in $\mathcal{S}^*_r$ and $\overline{\mathcal{S}}_r$ (for various choices of r) suffices in learning several linear forms from at least one multiplication gate of C. This step uses the assumption that all gates of C have "high rank". We show how to do this in Section 7.1. In the section after, we will show how to effectively reduce to the case where all ga†es have high rank.

7.1 All large rank gates

Lemma 7.1. *Let $t = 2^{2k^2}\mathcal{R}_{\mathbb{F}}(k,d)(k^4\log d)^{k-3}$. Let f be a n-variate, degree-d polynomial in $\mathbb{F}[x_1,\ldots,x_n]$ that is computed by a $\Sigma\Pi\Sigma(k)$ circuit $C = G\times(T_1+\cdots+T_k)$ with $\gcd(T_1,\ldots,T_k)=1$. Let $c_{cand}>0$ be an arbitrary constant (it can depend on d). Let $\lambda = 2^k\cdot(2t\log d+3k)+t+2k^2\cdot(c_{cand}+k)$. If it holds that $\forall i\in[k], \dim(\text{span}(\text{Lin}(T_i)))\geq\lambda$, then there exists an algorithm that runs in time $\text{poly}(n,d^{\text{poly}(t,c_{cand})})$ and computes a set of linear forms $\mathcal{L}_{cand}$ such that $|\mathcal{L}_{cand}| = d^{O(1)}$ and $\exists j\in[k]$ such that $\dim(\text{span}(\text{Lin}(T_j)\cap\mathcal{L}_{cand}))\geq c_{cand}$.*

Proof. Let f have a $2t\log d + 3k$-cluster representation $C = G \times (C_1 + \cdots + C_s)$ as defined in Lemma 5.7.

If there is some $r \in [s]$ for which the $SCS(r)$-property is not satisfied, then take the first r for which this holds, and by Theorem 6.2 we can learn the desired set of linear forms. Thus, we now assume that for all $r \in [s]$, the $SCS(r)$-property is not satisfied by all r. By applying Theorem 6.2 iteratively, we can thus compute the spaces $\mathcal{S}_r^*(f)$ and $\overline{\mathcal{S}}_r(f)$ for all $r \in [s]$ in $\text{poly}(n, d^{\text{poly}(t,c_{cand})})$ time.

We construct $\mathcal{L}_{cand}$ as follows. For each pair of spaces in the union of $\mathcal{S}_r^*(f)$ and $\overline{\mathcal{S}}_r(f)$ over all r, we consider the intersections of kernels of these spaces. If they intersect in a one-dimensional space, we add a linear form corresponding to the intersection to $\mathcal{L}_{cand}$. As both $|\mathcal{S}_r^*(f)|$ and $|\overline{\mathcal{S}}_r(f)|$ are $d^{O(1)}$ for all choices of r, clearly $|\mathcal{L}_{cand}|$ is $d^{O(1)}$ and the runtime of the algorithm is at most polynomial in the runtime of the algorithm from Theorem 6.2, which is $\text{poly}(n, d^{\text{poly}(t,c_{cand})})$.

We now show that the set $\mathcal{L}_{cand}$ has the desired properties.

Consider any set of independent linear forms $L_1 \subseteq \text{Lin}(\gcd(C_1))$ such that any $l \in L_1$, $l \notin \mathcal{W}_s$ and $|L_1| = 2k(c_{cand} + k) = \alpha$. Similarly for $i \in [s]$, let $L_i \subseteq \text{Lin}(\gcd(C_i))$ such that any $l \in L_i$, $l \notin \text{span}(\mathcal{W}_s, L_1, \ldots, L_{i-1})$ and $|L_i| = 2k(c_{cand} + k) = \alpha$. We will now justify that such L_i exist by the assumption of all gates being of high rank.

In any gate, the number of independent linear forms that can contribute to $\text{sim}(C_i)$ is at most $2^k \cdot (2t\log d + 3k)$, as shown in Lemma 5.7. As $\mathcal{W}_s$ is a LDICR space, we know $\dim(\mathcal{W}_s) \leq 2^{2k^2}\mathcal{R}_{\mathbb{F}}(k,d)(k^4\log d)^{s-3} \leq t$ from Definition 14. Since, we have $\dim(\text{span}(\text{Lin}(T_i))) \geq 2^k \cdot (2t\log d + 3k) + t + 2k^2 \cdot (c_{cand} + k)$, we can find enough independent linear forms to construct these sets $L_1, \ldots, L_s$.

Consider the set of spaces $\mathcal{S} = \{\mathbb{V}(l_1, \ldots, l_s) : \forall i \in [s], l_i \in L_i\}$. Note that each space in $\mathcal{S}$ is a vanishing space for C that does not intersect $\mathcal{W}_s$. We have $|\mathcal{S}| = (2k(c_{cand} + k))^s = \alpha^s$. We have learnt all the minimal vanishing kernels corresponding to each space in $\mathcal{S}$ in $\mathcal{S}_{\leq s}^*$ as shown in Lemma 6.12.

From Lemma 6.14, for any space $\mathbb{V}(l_1, \ldots, l_s) \in \mathcal{S}$ we can either learn $\mathbb{V}(l_1, \ldots, l_s)$ or we learn some space V' in $\overline{\mathcal{S}}_{\leq s}$, such that $\mathcal{K}V' \subseteq \text{span}(l_1, \ldots, l_s)$ and $\mathcal{K}V'$ contains at least one of the l_i's.

We divide our set $\mathcal{S}$ into $\mathcal{S}_{degen}$ and $\mathcal{S}_{non-degen}$, where for each space V in $\mathcal{S}_{non-degen}$, we learn V itself in $\overline{\mathcal{S}}_{\leq s}$ or in $\mathcal{S}_{\leq s}^*$, while for each space $V \in \mathcal{S}_{degen}$, we only learn a strict superspace in $\overline{\mathcal{S}}_{\leq s}$. Equivalently, we only learn a strict subspace of its kernel in $\mathcal{K}\overline{\mathcal{S}}_{\leq s}$. However the subspace kernel contains one of the l_i's that defined V.

For each space $V \in \mathcal{S}_{degen}$ such that V is of the form $\mathbb{V}(l_1, \ldots, l_s)$ we say that V is "associated" with the linear form l_i if the kernel of the learned subspace in $\overline{\mathcal{S}}_{\leq s}$ contains l_i.

For any linear form l that is in one of the L_i's, if there are two distinct spaces in $\mathcal{S}_{non-degen}$ whose kernels intersect in exactly l, then such an l gets learnt in $\mathcal{L}_{cand}$. We also learn l in $\mathcal{L}_{cand}$ if there are at least two spaces in $\mathcal{S}_{degen}$ that are both associated with l and the kernels of the spaces intersect in exactly l.

For any linear form $l \in L_j$ for some $j \in [s]$, there are $(\alpha)^{s-1}$ spaces in $\mathcal{S}$ whose kernels contain l. For a set $\mathcal{S}' \subset \mathcal{S}$, we say that $\mathcal{S}'$ is a matching with respect to l if it has the following properties. (1) Each element of $\mathcal{S}'$ contains l in the kernel, (2) $|\mathcal{S}'| = \alpha$ and each linear form in L_i (for any $L_i \neq L_j$),is in exactly one of the kernels of spaces in $\mathcal{S}'$.

For each $l \in L_j$ for some $j \in [s]$, we define $\mathcal{S}(l)$ to be a subset of $\mathcal{S}$ containing all those spaces whose kernel contains l. Observe that we can partition $\mathcal{S}(l)$ into α^{s-2} matchings with respect to l. This partition is not unique, but we fix any partition and thus obtain for any l, a set of α^{s-2} sets of spaces that are "matchings with respect to l".

Note that for any linear form l, if there is a single matching with respect to l that contains

two spaces in $\mathcal{S}_{non-degen}$, then their intersection recovers l, and thus we can learn l. We call such an l as "good". If there are more than kc_{cand} "good" linear forms, then we are done, as there will be at least one gate from which we have learned c_{cand} linear forms. Therefore, let us assume that for all except $kc_{cand} - 1$ linear forms, there is at most one space in $\mathcal{S}_{non-degen}$ in each matching corresponding to it. We call these linear forms "bad".

Thus we get the following bound on the size of $\mathcal{S}_{degen}$ (justification to follow).

$$|\mathcal{S}_{degen}| \geq (\alpha)^s - (kc_{cand}) \cdot (\alpha)^{s-1} - (s\alpha - kc_{cand}) \cdot (\alpha)^{s-2}$$

we get this expression because $(\alpha)^s$ is the size of $\mathcal{S}$. We first remove all elements of $\mathcal{S}$ that correspond to any "good" linear form. Thus, we remove at most $(kc_{cand}) \cdot (\alpha)^{s-1}$ elements. Thus, all remaining spaces correspond to all bad linear forms. Some of these spaces might be non-degenerate, and we would like to remove those. Since each bad linear form corresponds to at most $(\alpha)^{s-2}$ nondegenerate spaces, thus by eliminating at most a further $(s\alpha - kc_{cand}) \cdot (\alpha)^{s-2}$ spaces, we are left only with degenerate spaces.

If a linear form is associated with at least $T = (\alpha)^{s-2} + 1$ degenerate spaces, then we learn the linear form, as it will be associated with at least two spaces in the same matching corresponding to it. If we can lower bound the number of linear forms which are associated with at least $(\alpha)^{s-2} + 1$ degenerate spaces and show that it is at least kc_{cand}, then we will be done.

To do this, we upper bound the number of linear forms which are associated with at most $T - 1$ degenerate spaces. Let this quantity be a. Then we have $a \cdot (T-1) + (\alpha s - a) \cdot (\alpha)^{s-1}$ is at least $|\mathcal{S}_{degen}|$. Comparing the upper bounds and lower bound that we get on $|\mathcal{S}_{degen}|$, we get that

$$\begin{aligned} a \cdot \alpha^{s-2} + (\alpha s - a) \cdot \alpha^{s-1} &\geq \alpha^s - (kc_{cand}) \cdot \alpha^{s-1} - (s\alpha - kc_{cand}) \cdot \alpha^{s-2} \\ a &\leq \frac{\alpha^2 s - \alpha^2 + kc_{cand}\alpha + s\alpha - kc_{cand}}{\alpha - 1} \\ a &\leq (2kc_{cand} + 2k^2) \cdot (s-1) + kc_{cand} + s. \end{aligned} \quad \left.\begin{array}{l}\text{substituting} \\ \alpha = \\ 2k(c_{cand}+k)\end{array}\right)$$

Therefore, we get the number of linear forms associated with at least $\alpha^{s-2}+1$ degenerate spaces is at least $\alpha s - a \geq \alpha - kc_{cand} - s \geq kc_{cand}$. Thus, even in this case, we learn at least c_{cand} linear forms from one of the gates. □

7.2 Some small rank gates

In this section, we will show how to learn enough linear forms from some gate assuming there are some low rank gates. We still will require that the rank of the simple part of the circuit is high. If this is not the case, then as in previous works, we can recover the circuit as a sparse polynomial of linear forms (this part is quite standard), and we will handle this in the next section.

In the next lemma we will show how to learn enough linear forms when some of the gates are low rank. In proof we will show how to in some sense simulate blackbox access to the high rank part of the circuit and show how to compute $\mathcal{S}^*$ and $\overline{\mathcal{S}}$ spaces for the high rank gates. Once we can do this then we can learn the required linear forms as in Lemma 7.1.

Lemma 7.2. *Let $t = 2^{2k^2}\mathcal{R}_{\mathbb{F}}(k,d)(k^4 \log d)^{k-3}$. Let f be a n-variate, degree-d polynomial in $\mathbb{F}[x_1, \ldots, x_n]$ that is computed by a $\Sigma\Pi\Sigma(k)$ circuit $C = G \times (T_1 + \cdots + T_k)$ with $gcd(T_1, \ldots, T_k) = 1$. Let $c_{cand} > 0$ be an arbitrary constant (it can depend on d). Let $\lambda = 2^k \cdot (2t \log d + 3k) + t + 2k^2 \cdot (c_{cand} + k)$. If it holds that $\text{rank}(\text{sim}(C)) > 2^{8k} \cdot k \cdot \lambda$, then, there exists a randomized algorithm that runs in time $\text{poly}(n, d^{\text{poly}(t, c_{cand})})$ and computes with probability $1 - o_{\mathbb{F}}(1)$ a set of linear forms $\mathcal{L}_{cand}$ such that $|\mathcal{L}_{cand}| = d^{O(1)}$ and $\exists j \in [k]$ such that $\dim(\text{span}(\text{Lin}(T_j) \cap \mathcal{L}_{cand})) \geq c_{cand}$.*

Proof. We have already handled the case when all gates have rank at least λ in Lemma 7.1. Therefore, we only need to handle the case when there are some gates which have rank less than λ.

We will reduce this case to the case where all gates in the circuit have a rank at least λ. To do this, we will first partition the circuit into high rank and low rank gates. Gates with low rank, will have low number of essential variables (see Definition 7). Observe that if V is a vanishing space for the high rank gates, then when we consider the circuit restricted to V, the resulting polynomial has only few essential variables. This observation will motivate our algorithm. We will learn spaces of codimension upto r restricted to which the polynomial has few essential variables. We will show that these spaces are precisely the vanishing spaces for the high rank part of the circuit. Once we have a rich enough set of vanishing spaces ($\mathcal{S}^*$ and $\overline{\mathcal{S}}$) for the high rank gates, then we can learn c_{cand} independent linear forms in a gate using the previous lemma.

Partitioning the gates in high and low rank: The idea will be to partition the gates $T_i, i \in [k]$ by partitioning $[k]$ into two sets K and $K' = [k] \setminus K$ such that all low rank gates have indices in K' and there is a large separation between the ranks of gates with indices in K and the combined rank of gates with indices in K'.

Formally, $K' \subset [k]$ is such that $\forall j \in K, \dim(\text{span}(T_j)) \geq 8 \cdot \dim(\text{span}(\cup_{i \in K'} \text{Lin}(T_i))) + \lambda$. To find such a set K', we first choose (and add to K') all indices corresponding to gates T_i such that $\dim(\text{span}(\text{Lin}(T_i))) < \lambda$. Let the number of such chosen indices be a_1. Then, there must exist a gate in T_j (with index not in the current K'), such that $\dim(\text{span}(T_j)) < (8a_1 + 1) \cdot \lambda$, as otherwise, we are done. In this next step, we add indices corresponding to all such gates with $\dim(\text{span}(T_j)) < (8a_1+1) \cdot \lambda$ to K'. Suppose that the number of indices added is a_2. Similarly, now there must exist a gate (with index not in the current K') T_j, such that $\dim(\text{span}(T_j)) < (8a_1+1) \cdot (8a_2+1) \cdot \lambda$ (or again we are done). Suppose that this goes on for r steps and we get a_r new indices, and after that no gate remains, i.e. $K' = [k]$. We will now argue that this is not possible. If no gate remains after r steps, then $\dim(\text{span}(\cup_{i \in [k]} \text{Lin}(T_i))) \leq \sum_{j \in [r]} (1 + 8a_1) \dots (1 + 8a_{j-1}) \cdot a_j \cdot \lambda < (1 + 8a_1) \dots (1 + 8a_r) \cdot r \cdot \lambda$. Using $a_1 + \dots + a_r \leq k$ and $r \leq k$, we have $(1 + 8a_1) \dots (1 + 8a_r) \leq 2^{8k}$. Therefore, we have a contradiction as we know from assumption $\text{rank}(\text{sim}(C)) > 2^{8k} \cdot k \cdot \lambda$. Thus we conclude that when the process terminates, K' is not the full $[k]$ and thus its complement K (or high rank gates) is nonzero.

From the above argument, it immediately follows that $\dim(\text{span}(\cup_{i \in K'} \text{Lin}(T_i))) \leq 2^{8k}\lambda$. Let the rank of $\sum_{i \in K'} T_i$ (which is precisely $\dim(\text{span}(\cup_{i \in K'} \text{Lin}(T_i)))$) be m, for some integer m. We will assume henceforth that m is known to us. This is because we will run the algorithm for all $m \in [2^{8k} \cdot k \cdot \lambda]$ and our final set will be the union of all linear forms returned. Thus, it will include the linear forms for the correct choice of m.

Note that for every $i \in K$, the rank of T_i is at least $8m + \lambda$. Note that if for any choice of $\langle l_1, \dots, l_r \rangle$, if $C \mod \langle l_1, \dots, l_r \rangle$ has at most m essential variables, then it implies $\sum_{i \in K} T_i \mod \langle l_1, \dots, l_r \rangle$ has at most $2m$ essential variables. The second part of Lemma 5.6 thus implies it must be that $\sum_{i \in K} T_i \mod \langle l_1, \dots, l_r \rangle = 0$. In our algorithm, we will compute spaces $\mathbb{V}(l_1, \dots, l_r)$ such that restricted to them f has at most m essential variables. By the above reasoning, such spaces will be vanishing spaces for $\sum_{i \in K} T_i$.

We will discuss now how we can show an analog of Theorem 6.2 for $\sum_{i \in K} T_i$ (i.e. we will algorithmically compute a rich class of vanishing spaces) but only using blackbox access to C. First, we will show that if the $SCS(r)$ property doesn't hold for $\sum_{i \in K} T_i$ for any choice of $r \leq |K|$, we can compute $\mathcal{S}^*_r(\sum_{i \in K} T_i)$ and $\overline{\mathcal{S}}_r(\sum_{i \in K} T_i)$. We will also show that if $SCS(r)$ property does hold for some r, then we can use $\mathcal{S}^*_{\leq r-1}(\sum_{i \in K} T_i)$ and $\overline{\mathcal{S}}_{\leq r-1}(\sum_{i \in K} T_i)$, with blackbox access to C to compute c_{cand} independent linear forms from one gate.

Learning the $\mathcal{S}_r^*$ spaces of $\sum_{i\in K} T_i$ We will recursively learn these spaces. We will assume we have already learnt $\mathcal{W}_{r-1}, \mathcal{S}_1^*(\sum_{i\in K} T_i), \ldots, \mathcal{S}_{r-1}^*(\sum_{i\in K} T_i), \overline{\mathcal{S}}_1(\sum_{i\in K} T_i), \ldots, \overline{\mathcal{S}}_{r-1}(\sum_{i\in K} T_i)$ and we assume that the $SCS(r)$ property does not hold.

The algorithm for computation of a LDICR space, $\mathcal{W}_r$, corresponding to the circuit $\sum_{i\in K} T_i$ using the spaces $\mathcal{KS}_1^*(\sum_{i\in K} T_i), \ldots, \mathcal{KS}_{r-1}^*(\sum_{i\in K} T_i), \mathcal{K}\overline{\mathcal{S}}_1(\sum_{i\in K} T_i), \ldots, \mathcal{K}\overline{\mathcal{S}}_{r-1}(\sum_{i\in K} T_i)$ remains exactly the same as in Lemma 6.6. The algorithm for learning $\mathcal{S}_r^*(\sum_{i\in K} T_i)$ is almost identical to that of learning $\mathcal{S}_r^*(C)$ (Algorithm 2) with one key change - when we set up the system of equations, we don't set it up so that some polynomial vanishes, but set it up so that some polynomial has at most m essential variables. We elaborate below.

Similar to Algorithm 2, we take a random linear isomorphism Φ which is defined by $\Phi(x_i) = \sum_{j=1}^{n} \alpha_{i,j} x_j$(where $\alpha_{i,j}$ are chosen randomly from $[d^n]$) to get polynomial $g = \Phi(f) = f(\Phi(x))$. We then obtain polynomials g_i for $i \in [8m+1, n]$, by setting all but the first $8m$ variables and x_i to 0 in C. We can interpolate these (since they are only few variate) to get white-box access to the monomial representation of the g_i's. Let $\Phi(C)|_{x_{8m+1}=\cdots=x_{i-1}=x_{i+1}=\cdots=x_n=0} = G^{[i]} \times (T_1^{[i]} + \cdots + T_k^{[i]})$. For $j \in K$, As $\dim(\text{span}(\text{Lin}(T_j))) \geq 8m+\lambda$ (which is greater than $8m$), after projecting down the gates would still have high rank (at least $8m$) with high probability, similar to Lemma 6.8, while projected $\sum_{j\in K'} T_j^{[i]}$ will have at most m essential variables. Now, any $l_1, \ldots, l_r$ such that $C \mod \langle l_1, \ldots, l_r\rangle$ has at most m essential variables must be such that $(\sum_{j\in K} T_j^{[i]}) \mod \langle l_1, \ldots, l_r\rangle$ has at most $2m$ essential variables. From part 2 of Lemma 5.6, any $l_1, \ldots, l_r$ such that $(\sum_{j\in K} T_j^{[i]}) \mod \langle l_1, \ldots, l_r\rangle$ has at most $2m$ essential variables must be such that $(\sum_{j\in K} T_j^{[i]}) \mod \langle l_1, \ldots, l_r\rangle = 0$ (Since we know that all $T_i, i \in K$ have rank at least $8m + \lambda$).

We will first show that we can preprocess the circuit such that we reduce to the case where the linear forms in G are in the span of the linear forms appearing in $\sum_{i\in K'} T_i$. The argument is very similar to Corollary 5.3. We first observe that if l is any linear form dividing $\text{sim}(C)$ then $(\sum_{i\in[k]} T_i) \mod \langle l\rangle$ equals 0. By the rank bound Theorem 3.5 and the difference in ranks among the gates, it must hold that $(\sum_{i\in K'} T_i) \mod \langle l\rangle$ equals 0. Thus any such l is in the span of linear forms appearing in $\sum_{i\in K'} T_i$. We can look at all linear forms dividing C, guess the m-dimensional subset of linear forms corresponding to span of linear forms in $\sum_{i\in K'} T_i$ and divide out the remaining linear forms (which will not divide $\text{sim}(C)$. This follows the argument in Corollary 5.3, and thus we reduce to G having the required property.

Note, to learn $\mathcal{S}_r^*(\sum_{j\in K} T_j^{[i]})$, it suffices to learn the codimension r spaces on which $G^{[i]} \times (T_1^{[i]} + \cdots + T_k^{[i]})$ has at most m essential variables. Similar to Lemma 6.7, we have that if $\sum_{i\in K} T_i$ vanished mod $\langle l_1, \ldots, l_r\rangle$, then $\sum_{j\in K} T_j^{[i]}$ must vanish modulo $\langle x_1 - l_{1i}, \ldots, x_r - l_{ri}\rangle$ where $l_{ji} \in \mathbb{F}[x_{r+1}, \ldots, x_{8m}, x_i]$. Let $l_{ji} = \alpha_{j,r+1}x_{r+1} + \cdots + \alpha_{j,8m}x_{8m} + \alpha_{j,i}x_i$. For each $j \in r$, we substitute $x_j = \alpha_{j,r+1}x_{r+1} + \cdots + \alpha_{j,8m}x_{8m} + \alpha_{j,i}x_i$ into g_i which we have monomial access to (which we can obtain by interpolation since these polynomials are sparse). As seen in Lemma 3.14, the number of essential variables in any polynomial is the rank of the corresponding partial derivative matrix. Using white-box access to the g_i's, we can get access to its partial derivative matrix, and then we can set up a system of polynomial equations in for $j \in [r]$, $\alpha_{j,r+1}, \ldots, \alpha_{1,8m}, \alpha_{j,i}$ that bounds its rank to be at most m. We get this by equating all $(m+1)\times(m+1)$ minors of the matrix to 0, thus ensuring that the rank of the matrix is at most m. This ensures $g_i \mod \langle x_1 - l_{1i}, \ldots, x_r - l_{ri}\rangle$ has at most m essential variables. We will also add an equation ensuring that $\text{span}(x_1 - l_{1i}, \ldots, x_r - l_{ri})$ intersects $\Phi(\mathcal{W}_r)|_{x_{8m+1}=\ldots=x_{i-1}=x_{i+1}=\ldots=x_n=0}$ trivially, similar to Lemma 6.7. The system of equations is in $8m$ variables and has $d^{O(m)}$ equations with a degree at most $m+1$ and hence can be solved in $\text{poly}(d^{\text{poly}(m)})$ time using Theorem 3.10. We can then glue these solutions for g_i, similar to Lemma 6.12, by comparing coefficients in $x_{r+1}, \ldots, x_{8m}$, to get codimension r vanishing spaces

of $\sum_{j\in K} T_j^{[i]}$. Hence, we can compute $\mathcal{S}_r^*$ spaces of $\sum_{j\in K} T_j$ in time $d^{\text{poly}(m)}$ time with $1-o(1)$ probability.

Learning $\overline{\mathcal{S}}_r$ spaces: Next, we argue how to compute the $\overline{\mathcal{S}}_r$ spaces of $\sum_{j\in K} T_j$. We will again assume that $SCS(r)$ property is not satisfied and we have computed $\mathcal{W}_r, \mathcal{S}_1^*(\sum_{i\in K} T_i), \ldots, \mathcal{S}_r^*(\sum_{i\in K} T_i)$ and $\overline{\mathcal{S}}_1(\sum_{i\in K} T_i), \ldots, \overline{\mathcal{S}}_{r-1}(\sum_{i\in K} T_i)$.

Recall that to learn $\overline{\mathcal{S}}_r^{(1)}(C)$, we considered an r dimensional space $\text{span}(l_1, \ldots, l_k)$ for $(l_i \in \text{Lin}(T_i))$ which was not learned in $\mathcal{KS}_r^*$ because it contained a smaller space V in $\mathcal{KS}_{\le r}^*$. In Claim 6.16, we showed how to use V to learn a larger space in $\text{span}(l_1, \ldots, l_k)$. The steps after that for computation of $\overline{\mathcal{S}}_r^{(2)}, \ldots, \overline{\mathcal{S}}_r^{(r)}$ we consider the circuit mod intersections of spaces in $\overline{\mathcal{S}}_r^{(1)}$ and repeat the computation r times.

We need to now carry out these steps for $\sum_{i\in K} T_i$. All the later steps of computing $\overline{\mathcal{S}}_r^{(2)}, \ldots, \overline{\mathcal{S}}_r^{(r)}$ once we can compute $\overline{\mathcal{S}}_r^{(1)}$ are identical to Lemma 6.17. The only change required is in the procedure in Claim 6.16 to learn larger spaces that contained V but that are also in $\text{span}(l_1, \ldots, l_k)$.

To learn these larger spaces, in Algorithm 3, we consider a random $(\dim(V)-1)$-dimensional subspace V' of V and a linear form l' such that $\text{span}(V', l') = V$. Let e be the largest integer such that $(l')^e | C \mod V'$. We then considered the circuit $C' = C \mod V'/(l')^e$, and found $\mathcal{S}^*$ spaces of C' whose kernels contained l'. We appended the kernel of these spaces with V' to obtain larger vanishing spaces than V, and showed one of them was inside $\text{span}(l_1, \ldots, l_k)$. More details of the analysis are in Lemma 6.15 and Lemma 6.17.

The one step in all this that we cannot carry out is where we divide the circuit $C \mod V'$ by $(l')^e$ simply because we do not have black box access to the circuit we are computing $\overline{\mathcal{S}}_r$ spaces for, i.e. $\sum_{j\in K} T_j$, as we only have black-box access to the full circuit C.

We will show that we can still compute the $\overline{\mathcal{S}}_r^{(1)}$ spaces. We have black box access to the circuit $\sum_{j\in K} T_j + \sum_{j\in K'} T_j$, and we want to learn spaces of the form $\mathbb{V}(l_1, \ldots, l_{|K|})$, where $\dim(\text{span}(l_1, \ldots, l_{|K|})) = r$, $l_i \in \text{Lin}(T_i)$ for $i \in K$ and $\text{span}(l_1, \ldots, l_{|K|})$ intersects $\mathcal{W}_r(\sum_{j\in K} T_j)$ trivially. Let $\text{span}(l_1, \ldots, l_{|K|})$ contain a space V of smaller dimension such that $\sum_{j\in K} T_j \mod V = 0$ and hence $\mathbb{V}(l_1, \ldots, l_{|K|})$ couldn't be learned in $\mathcal{S}_r^*(\sum_{j\in K} T_j)$. Similar to the Claim 6.16, we consider a random $(\dim(V)-1)$-dimensional subspace V' of V and a linear form l' such that $\text{span}(V', l') = V$. Let e be the largest integer such that $(l')^e | (\sum_{j\in K} T_j) \mod V'$. Since $\text{span}(l_1, \ldots, l_{|K|})$ intersects $\mathcal{W}_r(\sum_{j\in K} T_j)$ trivially, we will have $\text{span}(l_1, \ldots, l_{|K|})$ doesn't contain any vanishing regular kernel, and hence l' doesn't divide $\text{sim}(\sum_{j\in K} T_j \mod V')$. The circuit $C \mod V'$ is of the form $(l')^e \cdot A + B$ where $A = ((\sum_{j\in K} T_j) \mod V')/(l')^e$ and $B = (\sum_{j\in K'} T_j) \mod V'$. We know the number of essential variables in A is at least $8m+\lambda-r$, while the number of essential variables in B is at most m. Wlog, we can consider an invertible linear isomorphism Φ from n variables to n variables such that $l' \to x_1$ and it keeps the rank of T_i's in K still high. After Φ, $C \mod V'$ looks like $(x_1)^e \cdot \Phi(A) + \Phi(B)$. Similar to Claim 6.16, we want to find $\mathcal{S}_{\le r-\dim(V')}^*(\Phi(A))$ such that it contains x_1 in the kernel, so we can append them to V and learn larger vanishing spaces. Assuming, we know e, this was easy to do in Claim 6.16, as we had black-box access to $\Phi(A)$. In our current case, we will simulate the same, by looking at partial derivatives of C wrt x_1, and then looking for spaces such that mod them the circuit has at most m essential variables.

To carry out the above plan, we first "guess" e from $\{1, \ldots, d\}$ (we will in fact run the algorithm for choices of e) and thus assume that we have the right choice of e. Then we look at the e-th order partial-derivative of $\Phi(C \mod V')$, i.e. $\frac{\partial^e \Phi(C \mod V')}{\partial x_1^e}$ with respect to x_1. Using product rule, it will be of the form $\Phi(A) + x_1 \cdot P + \frac{\partial^e \Phi(B)}{\partial x_1^e}$ where $P(x_1, \ldots, x_n)$ is some non-zero polynomial. Observe

that $\frac{\partial^e \Phi(B)}{\partial x_1^e}$ has almost m essential variables.

We will now compute the set of codimension upto $r - \dim V'$ spaces such that $\frac{\partial^e \Phi(C \mod V')}{\partial x_1^e}$ has almost m essential variables (similar to above) and their kernel contains x_1, which is equivalent to setting $x_1 = 0$ and finding vanishing spaces of $\Phi(A)$ and hence having no contribution from $x_1 \cdot P$. Thus, we find $\mathcal{S}^*_{\leq r-\dim(V)}(\Phi(A))$ containing x_1, and hence vanishing spaces of A such that it contains l' in the kernel by inverting Φ.

Therefore, we are able to learn the larger spaces from V, and hence learn $\overline{\mathcal{S}}^{(1)}_r(\sum_{j\in K} T_j)$ as well. Once, we can compute $\overline{\mathcal{S}}^{(1)}_r(\sum_{j\in K} T_j)$, we can look at intersections of these spaces and consider the circuit mod them, and repeat the computation of $\overline{\mathcal{S}}^{(1)}_r$, to learn $\overline{\mathcal{S}}^{(2)}_r, \ldots, \overline{\mathcal{S}}^{(r)}_r$ and hence $\overline{\mathcal{S}}_r(\sum_{j\in K} T_j)$.

Learning linear forms if $SCS(r)$ property holds Fix r to be the smallest such that $SCS(r)$ property holds. As seen above, we can compute $\mathcal{S}^*_{\leq r-1}(\sum_{i\in K} T_i)$ and $\overline{\mathcal{S}}_{\leq r-1}(\sum_{i\in K} T_i)$. In Lemma 6.19, we saw if the $SCS(r)$ property is satisfied for C then we can use $\mathcal{S}^*_{\leq r-1}(C)$ and $\overline{\mathcal{S}}_{\leq r-1}(C)$ to learn c_{cand} independent linear forms from a gate. To do this, we went mod the two independent spaces that satisfied the $SCS(r)$ property, learned the projections linear forms in the gcd of the cluster that survived by factoring, and then glued these projections back to learn the required linear forms. In current lemma, we are trying to learn linear forms from gates in $\sum_{i\in K} T_i$, and we have access to $\mathcal{S}^*_{\leq r-1}(\sum_{i\in K} T_i)$ and $\overline{\mathcal{S}}_{\leq r-1}(\sum_{i\in K} T_i)$, but we no longer have blackbox access to $\sum_{i\in K} T_i$ for factoring the surviving cluster. We will bypass this by finding codimension 1 spaces restricted to which the polynomial has at most m essential variables, instead of factoring.

Let W_a and W_b be the two completely independent spaces that come from the $SCS(r)$ property of $\sum_{i\in K} T_i$. When we go modulo W_a and W_b, only one cluster of the $2t\log d + 3k$-cluster representation of $\sum_{i\in K} T_i$ survives. We also know W_a and W_b are in the set $\mathcal{P}_{r-2}$ computed using $\mathcal{S}^*_{\leq r-1}(\sum_{i\in K} T_i)$ and $\overline{\mathcal{S}}_{\leq r-1}(\sum_{i\in K} T_i)$. Since $|\mathcal{P}_{r-2}| = d^{O(1)}$, we run the algorithm for all possible choices and output the set of linear forms as a union, and hence it will contain the linear forms computed when we picked the correct W_a and W_b.

We consider $C \mod W_a$ and $C \mod W_b$. We will then compute codimension 1 spaces modulo which the circuits have at most m essential variables, and these will be the linear forms mod which $(\sum_{i\in K} T_i) \mod W_a$ and $(\sum_{i\in K} T_i) \mod W_b$ vanish respectively. This computation is exactly the $\mathcal{S}^*_r(\sum_{i\in K} T_i)$ computation for $r = 1$, we discussed earlier. Since, only one cluster survives, the linear forms in gcd of the surviving cluster will also be in this list. Thus, we compute the linear forms in gcd of the cluster mod W_a and W_b. As discussed in Lemma 6.19, we can compute the linear forms in gcd of cluster exactly by gluing the two projections as the spaces W_a and W_b are completely independent.

Therefore, we have learned the linear forms in the gcd of the cluster. Since, it is $2t\log d + 3k$ cluster representation, the linear forms from any T_i that went to the simple part of cluster is at most $2^k \cdot (2t\log d + 3k)$ from Lemma 5.7. We know for any gate index $i \in K$, we have $\dim(\text{span}(\text{Lin}(T_i))) \geq 8m + \lambda$. And $\lambda = 2^k \cdot (2t\log d + 3k) + t + 2k^2 \cdot (c_{cand} + k)$. Thus, even after removing the linear forms lost in simple part of the cluster, we are still able to learn at least $8m + t + 2k^2 \cdot (c_{cand} + k)$ independent linear forms from gcd of the cluster and hence from one of the gates. □

8 Learning Circuit from few linear forms

We saw in Lemma 7.2, if the rank of the simple part of the circuit is large, then we learn a list $\mathcal{L}_{cand}$ with c_{cand} independent linear forms in one of the gates (for any $c_{cand} = \text{poly}(\log d)$) in $(nd)^{\text{poly}(\log d)}$

time with high probability. In this section, we will discuss how we can reconstruct the entire circuit if we have black-box access to a $\Sigma\Pi\Sigma(k)$ circuit $C = G \times (T_1 + \cdots + T_k)$ using the set $\mathcal{L}_{cand}$.

We will first see how to do reconstruction when the $\text{rank}(\text{sim}(C))$ is low in Lemma 8.1, i.e. in case when we cannot use Lemma 7.2 to learn linear forms in a gate. We will use Lemma 3.9 to factor out the linear factors of C and then learn the Non-linear part using the Carlini Algorithm (Lemma 3.14) to obtain a $\Sigma\Pi\Sigma(1, d, \text{poly}(\log d))$ generalized circuit computing the same polynomial as C.

Next, we will look at a clustering result from [KS09a], where they show that every $\Sigma\Pi\Sigma(k)$ circuit has a unique clustering if the clusters are far enough (Theorem 8.2). This gives a unique $\Sigma\Pi\Sigma(s, d, \text{poly}(\log d))$ $(s \leq k)$ representation for every $\Sigma\Pi\Sigma(k)$ circuit. We will focus on learning this unique representation.

Using the linear forms, we have learnt from Lemma 7.2, we will use them to get projections of sum of other clusters, which we can reconstruct recursively. Using these reconstructions, we get enough projections of a single cluster that we can combine them using Theorem 3.16 and Lemma 3.17. Once, we have learnt the cluster, we can subtract it from the circuit and learn the rest of the circuit with lower top fan-in. At the end we do a PIT check to ensure the output circuit computes same polynomial as C.

8.1 Low Rank Reconstruction

In this section, we will give a reconstruction algorithm for the case when the $\text{rank}(\text{sim}(C)) < 2^{k^k+9k^2}\mathcal{R}_{\mathbb{F}}(k, d)(\log d)^{k^k}$.

Lemma 8.1. *Given black-box access to a $\Sigma\Pi\Sigma(k)$ circuit $C = G \times (T_1 + \cdots + T_k)$ computing a degree-d polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ with $\text{rank}(\text{sim}(C)) < 2^{k^k+9k^2}\mathcal{R}_{\mathbb{F}}(k, d)(\log d)^{k^k}$, there exists an algorithm that runs in randomized time $(nd)^{\text{poly}(\log d)}$ and with probability $1 - o(1)$ outputs a $\Sigma\Pi\Sigma(1, d, 2^{k^k+9k^2}\mathcal{R}_{\mathbb{F}}(k, d)(\log d)^{k^k})$ generalized depth-3 circuit as defined in Definition 5.*

Proof. The input circuit is of the form $C = G \times (T_1 + \cdots + T_k)$ computing f where $\dim(\text{span}(\{l : l | (T_1 \times \cdots \times T_k)\})) < 2^{k^k+9k^2}\mathcal{R}_{\mathbb{F}}(k, d)(\log d)^{k^k}$. Clearly, the non-linear factor of f, $\text{NonLin}(f) = \frac{f}{\prod_{l \in \text{Lin}(f)} l}$ will divide $T_1 + \cdots + T_k$ and therefore, will have essential variables less than $\text{rank}(\text{sim}(C))$. So, we use Lemma 3.9 to get black-box access to $\text{NonLin}(f)$ and the linear factors $\text{Lin}(f)$ in randomized $\text{poly}(n, d)$ time. As $\text{NonLin}(f)$ has at most $\text{rank}(\text{sim}(C))$ essential variables, there exist a linear transformation A such that $\text{NonLin}(f)(A \cdot \overline{x})$ depends only on $\text{rank}(\text{sim}(C))$ variables. Using Theorem 3.12, we can compute A in randomized polynomial time. We can do polynomial interpolation in time $(nd)^{\text{rank}(\text{sim}(C))}$ from Lemma 3.8 to get monomial access to and hence learn $\text{NonLin}(f)(A \cdot \overline{x})$. We use A^{-1} to recover $\text{NonLin}(f)$, and then the circuit by multiplying it with $\text{Lin}(f)$. Notice that this would give us a $\Sigma\Pi\Sigma(1, d, 2^{k^k+9k^2}\mathcal{R}_{\mathbb{F}}(k, d)(\log d)^{k^k})$ circuit computing f. □

8.2 Unique Clustering

In this section, we will look at the κ-distant clustering as defined in [KS09a]. We will ask the reader to note that this is not the same as the clustering mentioned in Lemma 5.7.

Definition 18 (κ-distant circuit)**.** *Let C be a $\Sigma\Pi\Sigma(s, d, r)$ generalized depth-3 circuit(Definition 5) computing a polynomial f. We say that C is κ-distant if for any two multiplication gates of C, M and M', we have that $\Delta(M, M') \geq \kappa \cdot r$.*

The following theorem in [KS09a] showed that for every $\Sigma\Pi\Sigma(k)$ circuit and large enough κ, there exists a unique $\Sigma\Pi\Sigma(s,d,r)$ generalized depth 3 circuit, for $s \leq k$ and r in the suitable range.

Theorem 8.2 (Existence and Uniqueness,(Theorem 3.2,3.6[KS09a])). *Let f be a polynomial that can be computed by a $\Sigma\Pi\Sigma(k)$ circuit and let $\kappa \geq \frac{\mathcal{R}_{\mathbb{F}}(2k,d)}{\mathcal{R}_{\mathbb{F}}(k+1,d)+k} + k$. Then, there exists a $\Sigma\Pi\Sigma(s,d,r)$ generalized depth-3 circuit C' computing f for $s \leq k$ and $\mathcal{R}_{\mathbb{F}}(k+1,d)+k \leq r \leq (\mathcal{R}_{\mathbb{F}}(k+1,d)+k)\cdot(\kappa\cdot k+k)^{k-2}$. Moreover, this representation is unique.*

8.3 High Rank Reconstruction

We will focus on learning the unique $\Sigma\Pi\Sigma(s,d,r)$ configuration described in Theorem 8.2.

Lemma 8.3. *Let f be a n-variate degree-d polynomial in $\mathbb{F}[x_1,\ldots,x_n]$ such that it can be computed by a $\Sigma\Pi\Sigma(k)$ circuit with $\operatorname{rank}(\operatorname{sim}(C)) > 2^{k^k+9k^2}\mathcal{R}_{\mathbb{F}}(k,d)(\log d)^{k^k}$. Let $\kappa = 2^{(k-1)^{k-1}+9(k-1)^2+1}\mathcal{R}_{\mathbb{F}}(k,d)\cdot(\log d)^{(k-1)^{(k-1)}+1}$. Let $C = G\times(C_1+\cdots+C_s)$ be the κ-distant unique generalized $\Sigma\Pi\Sigma(s,d,r)$ circuit from Theorem 8.2 also computing f with $2\leq s\leq k$. Then, we can compute $C = G\times(C_1+\cdots+C_s)$ in $(nd)^{\operatorname{poly}(\log d)}$ time with $1-o(1)$ probability.*

Proof. We have $\kappa = 2^{(k-1)^{k-1}+9(k-1)^2+1}\cdot(\log d)^{(k-1)^{(k-1)}+1}$, therefore, from Theorem 8.2, there must exist a unique circuit $\Sigma\Pi\Sigma(s,d,r)$ κ-distance generalized depth-3 circuit $C = G\times(C_1+\cdots+C_s)$ computing f, where $\mathcal{R}_{\mathbb{F}}(k+1,d)+k \leq r \leq (\mathcal{R}_{\mathbb{F}}(k+1,d)+k)\cdot(\kappa\cdot k+k)^{k-2}$, i.e. $r \leq 2^{(k-1)^k}(\log d)^{(k-1)^k}$.

Let $c_{cand} = 2^{k^k}\mathcal{R}_{\mathbb{F}}(k,d)(\log d)^{k^k}$. Since, $\operatorname{rank}(\operatorname{sim}(C)) \geq 2^{k^k+9k^2}\mathcal{R}_{\mathbb{F}}(k,d)(\log d)^{k^k}$, from Lemma 7.2, we have a $\operatorname{poly}(d)$-sized list $\mathcal{L}_{cand}$ which has at least c_{cand} independent linear forms from one of the gates. Wlog, that gate is T_1, which is part of cluster C_1. Guess c_{cand} independent linear forms from $\mathcal{L}_{cand}$, and run the algorithm in parallel for all guesses. There will be at most $d^{c_{cand}}$ guesses, and for the right guess, we have $l_1,\ldots,l_{cand}$ independent linear forms from T_1.

At least $c_{cand}-r$ of these linear forms must be part of $\operatorname{Lin}(C_1)$, and C_1 vanishes mod these linear forms. Wlog these be $l_1,\ldots,l_{c_{cand}-r}$. We consider the circuit C mod these linear forms. Now consider the circuit $(C_2+\cdots+C_s) \mod l_i$ for $i\in[c_{cand}-r]$. The circuit has top fan-in at most $k-1$ as at least T_1 vanished. From Corollary 5.3, we have at most $2^{k^2}\mathcal{R}_{\mathbb{F}}(k,d)\log d$ linear forms might be such that all the clusters vanish mod them. We can learn projections of these clusters recursively, if $\operatorname{rank}(\operatorname{sim}((C_2+\cdots+C_s) \mod l_i)) \geq 2^{(k-1)^{(k-1)}+9(k-1)^2}\mathcal{R}_{\mathbb{F}}(k-1,d)(\log d)^{(k-1)^{(k-1)}}$. If only one cluster survives, we learn the projection using Lemma 8.1. Otherwise, we have the remaining clusters had distance at least $\kappa\cdot r \geq \kappa\cdot\mathcal{R}_{\mathbb{F}}(k+1,d)$. We will next exclude all linear forms such that mod them the rank of the simple part of remaining clusters dropped from $\kappa\cdot\mathcal{R}_{\mathbb{F}}(k+1,d)$ to $2^{(k-1)^{(k-1)}+9(k-1)^2}\mathcal{R}_{\mathbb{F}}(k-1,d)(\log d)^{(k-1)^{(k-1)}}$. This is exactly the setting, where we can use Lemma 5.1 with $r' = 2^{(k-1)^{(k-1)}+9(k-1)^2}\mathcal{R}_{\mathbb{F}}(k-1,d)(\log d)^{(k-1)^{(k-1)}}$. Therefore, Lemma 5.1, gives us that for a given pair of clusters surviving, the rank of the simple part of the circuit can go be low r' for at most $r'\log d$ linear forms. Doing this for all possible pairs, we have to exclude $k^2r'\log d + r + 2^{k^2}\mathcal{R}_{\mathbb{F}}(k,d)\log d$ linear forms from $l_1,\ldots,l_{cand}$. Since, $c_{cand} = 2^{k^k}\mathcal{R}_{\mathbb{F}}(k,d)(\log d)^{k^k}$ is much bigger, we easily have $k\cdot(100\log d + r + 2)$ independent linear forms l_i's such that which we can reconstruct $(C_2+\cdots+C_s) \mod l_i$ exactly due to uniqueness from Theorem 8.2, and get projections of clusters that did not vanish.

Therefore, we have at least $100\log d + r + 2$ many projections of a single cluster C_j. From Theorem 3.16, we need $100\log d$ independent projections of $\operatorname{Lin}(G\times C_j)$ to obtain $\operatorname{Lin}(G\times C_j)$. From Lemma 3.17 and Claim 7.5 of [SS25], we need $r+2$ projections of $\operatorname{NonLin}(C_j)$, to obtain $\operatorname{NonLin}(C_j)$ exactly. Since we have enough independent projections, we can use them to learn $G\times C_j$.

Once we have learned $G \times C_j$, we can simply subtract it from C, and hence learn the remaining circuit with smaller fan-in to obtain a circuit for $C' = C - G \times C_j$. At the end, we run a PIT check of Lemma 3.3 to see if $C' + G \times C_j = C$, and output $C' + G \times C_j$ if correct. □

8.4 Proof of Theorem 1.1

Finally using Lemma 8.3 and Lemma 8.1 let us finish the proof of Theorem 1.1.

Proof of Theorem 1.1. Let $c_1 = 2^{k^k}$ and $c_2 = k^k$. Then, if $\forall i \neq j \in [k], \text{rank}(\text{sim}(T_i + T_j)) \geq c_1(\log d)^{c_2}$, the cluster representation of Theorem 8.2 will be $G \times (T_1 + \cdots + T_k)$ as all the gates will be far enough and each cluster will have only one gate. In this case, Lemma 8.3 will learn the exact circuit, and we will have proper learning.

If the rank of the simple part is small, we learn a $\Sigma\Pi\Sigma(1, d, c_1(\log d)^{c_2})$ circuit in Lemma 8.1. If the rank is high, but not all gates are far apart, we learn a $\Sigma\Pi\Sigma(s, d, c_1(\log d)^{c_2})$ circuit in Lemma 8.3 for some $s \leq k - 1$. □

9 Future Work

The major open question that remains open is to understand the problem of proper learning for small rank $\Sigma\Pi\Sigma(k)$ circuits. This remains open even for top fan-in 2. Obtaining hardness results for the problem would also be very interesting.

It is also a very interesting question to de-randomize the current algorithm, as well as obtain reconstruction algorithms over low characteristic fields when top fan-in is greater than 2. Finally, it would also be interesting to obtain reconstruction algorithms for "generalized" depth-3 circuits and $\Sigma^k\Pi\Sigma\Pi^\delta$ circuits for constant k, δ as we have recently made good progress on PIT for these models.

References

[Ang88] D. Angluin. Queries and concept learning. *Machine Learning*, 2:319–342, 1988.

[AV08] 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.

[BBB+00] 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.

[BCC+18] Alessandra Bernardi, Enrico Carlini, Maria Virginia Catalisano, Alessandro Gimigliano, and Alessandro Oneto. The hitchhiker guide to: Secant varieties and tensor decomposition. *Mathematics*, 6(12):314, 2018.

[BDWY13] 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.

[BE67] W Bonnice and Michael Edelstein. Flats associated with finite sets in pd. *Niew. Arch. Wisk*, 15:11–14, 1967.

[BGKS22] 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)*. Schloss-Dagstuhl-Leibniz Zentrum für Informatik, 2022.

[BOT88] 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.

[BS25] Vishwas Bhargava and Devansh Shringi. Faster & Deterministic FPT Algorithm for Worst-Case Tensor Decomposition. In *52nd International Colloquium on Automata, Languages, and Programming (ICALP 2025)*, volume 334, pages 28:1–28:20, 2025.

[BSV20] Vishwas Bhargava, Shubhangi Saraf, and Ilya Volkovich. Reconstruction of depth-4 multilinear circuits. *SODA 2020*, 2020.

[BSV21] 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.

[Car06] Enrico Carlini. Reducing the number of variables of a polynomial. In *Algebraic geometry and geometric modeling*, pages 237–247. Springer, 2006.

[CGK+24] Pritam Chandra, Ankit Garg, Neeraj Kayal, Kunal Mittal, and Tanmay Sinha. Learning arithmetic formulas in the presence of noise: A general framework and applications to unsupervised learning. In *15th Innovations in Theoretical Computer Science Conference, ITCS*, volume 287 of *LIPIcs*, pages 25:1–25:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024.

[DDS21] 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.

[DGI+24] Pranjal Dutta, Fulvio Gesmundo, Christian Ikenmeyer, Gorav Jindal, and Vladimir Lysikov. Homogeneous algebraic complexity theory and algebraic formulas. In *15th Innovations in Theoretical Computer Science Conference (ITCS 2024)*, pages 43–1. Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2024.

[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.

[DSW14] 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.

[FS12] M. A. Forbes and A. Shpilka. Quasipolynomial-time identity testing of noncommutative and read-once oblivious algebraic branching programs. *Electronic Colloquium on Computational Complexity (ECCC)*, 19:115, 2012.

[GKKS13] 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.

[GKL12] 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.

[GKQ14] A. Gupta, N. Kayal, and Y. Qiao. Random arithmetic formulas can be reconstructed efficiently. *Computational Complexity*, 23(2):207–303, 2014.

[GKS20] 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.

[GOPS23] 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)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2023.

[GOS22] 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.

[GVJ88] D Yu Grigor'ev and Nicolai N Vorobjov Jr. Solving systems of polynomial inequalities in subexponential time. *Journal of symbolic computation*, 5(1-2):37–64, 1988.

[Han65] 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.

[Ier89] D. Ierardi. Quantifier elimination in the theory of an algebraically-closed field. In *Proceedings of the Twenty-First Annual ACM Symposium on Theory of Computing*, STOC '89, page 138–147, New York, NY, USA, 1989. Association for Computing Machinery.

[Kay11] 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.

[KNS19] 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.

[KNST17] 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.

[Koi10] P. Koiran. Arithmetic circuits: the chasm at depth four gets wider. *CoRR*, abs/1006.4700, 2010.

[KS01] 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.

[KS06] A. Klivans and A. Shpilka. Learning restricted models of arithmetic circuits. *Theory of computing*, 2(10):185–206, 2006.

[KS08] 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.

[KS09a] 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.

[KS09b] 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.

[KS19] 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.

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

[KT90] Erich Kaltofen and Barry M Trager. Computing with polynomials given by black boxes for their evaluations: Greatest common divisors, factorization, separation of numerators and denominators. *Journal of Symbolic Computation*, 9(3):301–320, 1990.

[Lan15] Joseph M Landsberg. Geometric complexity theory: an introduction for geometers. *Annali dell'universita'di Ferrara*, 61(1):65–117, 2015.

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

[LST22] 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.

[OS22] 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.

[OS24] 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.

[PS21] 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.

[PS22a] 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.

[PS22b] 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.

[PSV24] 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.

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

[Shp07] 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.

[Shp19] 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.

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

[Sin16b] 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.

[Sin22] Gaurav Sinha. Efficient reconstruction of depth three arithmetic circuits with top fan-in two. In *13th Innovations in Theoretical Computer Science Conference (ITCS 2022)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2022.

[SS11] 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.

[SS13] 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.

[SS25] Shubhangi Saraf and Devansh Shringi. Reconstruction of Depth 3 Arithmetic Circuits with Top Fan-In 3. In *40th Computational Complexity Conference (CCC 2025)*, volume 339, pages 21:1–21:22, 2025.

[Tav13] S. Tavenas. Improved bounds for reduction to depth 4 and depth 3. In *MFCS*, pages 813–824, 2013.

[Zip79] Richard Zippel. Probabilistic algorithms for sparse polynomials. In *International symposium on symbolic and algebraic manipulation*, pages 216–226. Springer, 1979.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il