Electronic Colloquium on Computational Complexity, Report No. 14 (2021)

Hitting Sets and Reconstruction for Dense Orbits in $\mathrm{VP_e}$ and $\Sigma\Pi\Sigma$ Circuits

Dori Medini* Amir Shpilka*

Abstract

In this paper we study polynomials in $\mathrm{VP_e}$ (polynomial-sized formulas) and in $\Sigma\Pi\Sigma$ (polynomial-size depth-3 circuits) whose orbits, under the action of the affine group $\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})$,[1] are *dense* in their ambient class. We construct hitting sets and interpolating sets for these orbits as well as give reconstruction algorithms. Specifically, we obtain the following results:

1. For $\mathrm{C}_n\left(\ell_1(\boldsymbol{x}),\ldots,\ell_n(\boldsymbol{x})\right) \triangleq \mathrm{Trace}\left(\begin{pmatrix}\ell_1(\boldsymbol{x}) & 1\\ 1 & 0\end{pmatrix}\cdot\ldots\cdot\begin{pmatrix}\ell_n(\boldsymbol{x}) & 1\\ 1 & 0\end{pmatrix}\right)$, where the ℓ_is are linearly independent linear functions, we construct a polynomial-sized interpolating set, and give a polynomial-time reconstruction algorithm. By a result of Bringmann, Ikenmeyer and Zuiddam, the set of all such polynomials is dense in $\mathrm{VP_e}$ [BIZ18], thus our construction gives the first polynomial-size interpolating set for a dense subclass of $\mathrm{VP_e}$.
2. For polynomials of the form $\mathrm{ANF}_\Delta\left(\ell_1(\boldsymbol{x}),\ldots,\ell_{4^\Delta}(\boldsymbol{x})\right)$, where $\mathrm{ANF}_\Delta(\boldsymbol{x})$ is the canonical read-once formula in *alternating normal form*, of depth 2Δ, and the ℓ_is are linearly independent linear functions, we provide a quasipolynomial-size interpolating set. We also observe that the reconstruction algorithm of [GKQ14] works for *all* polynomials in this class. This class is also dense in $\mathrm{VP_e}$.
3. Similarly, we give a quasipolynomial-sized hitting set for read-once formulas (not necessarily in alternating normal form) composed with a set of linearly independent linear functions. This gives another dense class in $\mathrm{VP_e}$.
4. We give a quasipolynomial-sized hitting set for polynomials of the form $f\left(\ell_1(\boldsymbol{x}),\ldots,\ell_m(\boldsymbol{x})\right)$, where f is an m-variate s-sparse polynomial and the ℓ_is are linearly independent linear functions in $n \geq m$ variables. This class is dense in $\Sigma\Pi\Sigma$.
5. For polynomials of the form $\sum_{i=1}^{s}\prod_{j=1}^{d}\ell_{i,j}(\boldsymbol{x})$, where the $\ell_{i,j}$s are linearly independent linear functions, we construct a polynomial-sized interpolating set. We also observe that the reconstruction algorithm of [KNS19] works for *every* polynomial in the class. This class is dense in $\Sigma\Pi\Sigma$.

As $\mathrm{VP} = \mathrm{VNC}^2$, our results for $\mathrm{VP_e}$ translate immediately to VP with a quasipolynomial blow up in parameters.

If any of our hitting or interpolating sets could be made *robust* then this would immediately yield a hitting set for the superclass in which the relevant class is dense, and as a consequence also a lower bound for the superclass. Unfortunately, we also prove that the kind of constructions that we have found (which are defined in terms of k-independent polynomial maps) do not necessarily yield robust hitting sets.

*Department of Computer Science, Tel Aviv University, Tel Aviv, Israel, E-mail: dorimedini@gmail.com, shpilka@tauex.tau.ac.il. The research leading to these results has received funding from the Israel Science Foundation (grant number 514/20) and from the Len Blavatnik and the Blavatnik Family foundation.

[1] The action of $(A,\boldsymbol{b}) \in \mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})$ on a polynomial $f \in \mathbb{F}[\boldsymbol{x}]$ is defined as $(A,\boldsymbol{b}) \circ f = f(A^T\boldsymbol{x}+\boldsymbol{b})$.

ISSN 1433-8092

Contents

1 Introduction

Proving lower bounds on the size of algebraic circuits (also called arithmetic circuits), is an outstanding open problem in algebraic complexity. In spite of much effort, only a handful of lower bounds are known (a detailed account of most known lower bounds can be found in the excellent survey of Saptharishi [Sap15]). One common theme of most known lower bounds is that they are proved using *algebraic arguments*. That is, a proof of a lower bound for a class of circuits $\mathcal{C}$, usually has the following structure: one comes up with a set of (nonzero) polynomials $F_1, \ldots, F_m$, in $N = \binom{n+d}{d}$ many variables, such that the coefficient vector of every n-variate, degree-d polynomial that can be computed in $\mathcal{C}$, is a common zero of all the F_is (such F_is are called *separating polynomials*). Then, one exhibits a polynomial f whose coefficient vector is not a common zero, thus proving $f \notin \mathcal{C}$. As an example one can immediately see that the well known partial derivative technique, and its predecessor, shifted partial derivative technique, are algebraic. Grochow [Gro15] demonstrated this for most of the known lower bound proofs. As the set of common zeros of a set of polynomials is closed,[2] this immediately implies that if we prove that $f \notin \mathcal{C}$ using an algebraic argument, then the same argument also implies that $f \notin \overline{\mathcal{C}}$, the closure of $\mathcal{C}$. Recall that, in characteristic zero, the closure of a class $\mathcal{C}$ is the set of all polynomials that are limit points of sequences of polynomials from $\mathcal{C}$, where convergence is coefficient-wise (see Definition 1.5 for a general definition over arbitrary characteristic). As most known techniques are algebraic, we see that for proving a lower bound for a class $\mathcal{C}$ one actually has to consider the larger, and less structured class, $\overline{\mathcal{C}}$.

Geometric Complexity Theory (GCT for short), which was initiated by Mulmuley and Sohoni [MS01, MS08], approaches the lower bound question from a different angle. GCT also looks for an algebraic lower bound proof, but rather than exhibiting an algebraic argument, it aims to prove the existence of a separating polynomial. Specifically, GCT attempts to prove Valiant's hypothesis, that VP≠VNP, over $\mathbb{C}$, via *representation theory*. Valiant's hypothesis is, more or less, equivalent to showing that the permanent of a symbolic $n \times n$ matrix is not a *projection* of the symbolic $m \times m$ determinant for any $m = m(n)$ polynomial in n.[3] Recall that a projection of a polynomial is a restriction of the polynomial to an affine subspace of its inputs. Observe that a restriction of an n-variate polynomial $f(\boldsymbol{x})$ to a subspace of its inputs, is equivalent to considering the polynomial $f(A\boldsymbol{x} + \boldsymbol{b})$, where A is an $n \times n$ matrix and $\boldsymbol{b} \in \mathbb{C}^n$. As any matrix is a limit point of a sequence of invertible matrices, an algebraic proof that the permanent is not a projection of the $m \times m$ determinant, over $\mathbb{C}$, is equivalent to an algebraic proof showing that the permanent is not in the closure of the set of polynomials $\{\mathrm{Det}(AX + \boldsymbol{b}) \mid A \in \mathrm{GL}_m(\mathbb{C}),\ \boldsymbol{b} \in \mathbb{C}^m\}$, where $\mathrm{GL}_m(\mathbb{C})$ is the group of invertible $m \times m$ matrices (this is true for every field of characteristic $\neq 2$). The set $\{\mathrm{Det}(AX + \boldsymbol{b}) \mid A \in \mathrm{GL}_m(\mathbb{C}),\ \boldsymbol{b} \in \mathbb{C}^m\}$ is called the *orbit* of the determinant under the action of the affine group (we denote the affine group over $\mathbb{C}^m$ with $\mathrm{GL}_m^{\mathrm{aff}}(\mathbb{C})$). GCT considers the linear space of polynomials that vanish on every coefficient vector in the orbit of the determinant, and similarly the linear space of polynomials that vanish on every coefficient vector in the orbit of the permanent. There is a natural action of $\mathrm{GL}_m^{\mathrm{aff}}(\mathbb{C})$ on those linear spaces, thus defining two representations of $\mathrm{GL}_m^{\mathrm{aff}}(\mathbb{C})$. GCT wishes to find a separating polynomial by showing that some irreducible representation of $\mathrm{GL}_m^{\mathrm{aff}}(\mathbb{C})$ has strictly larger multiplicity when considering the representation corresponding to the determinant. This approach bypasses the barrier given in [FSV18, GKSS17] as it does not exhibit any efficiently computable separating polynomial but rather just proves the existence of one. However, the representation theory questions arising in this program are quite difficult, even when considering the analog

[2] It is closed in the Zariski topology. Over $\mathbb{R}$ or $\mathbb{C}$ this is the same as being closed in the Euclidean topology.

[3] A super-quasipolynomial lower bound would imply that VP≠VNP whereas a super-polynomial lower bound would imply that permanent does not have polynomial-size algebraic formulas or algebraic branching programs.

questions for restricted classes. For an introduction to GCT see the lecture notes of Bläser and Ikenmeyer [BI19].

Another possible approach for proving lower bounds against a class of polynomials $\mathcal{C}$, is via the construction of a *hitting set* for $\mathcal{C}$. Recall that a hitting set $\mathcal{H}$ for a class $\mathcal{C}$ is a set of points such that for any nonzero polynomial f, that can be computed by a circuit from $\mathcal{C}$, there is $\boldsymbol{v} \in \mathcal{H}$ such that $f(\boldsymbol{v}) \neq 0$. In [HS80] Heintz and Schnorr observed that if we have such a hitting set $\mathcal{H}$ then any nonzero polynomial g that vanishes on $\mathcal{H}$ cannot be computed in $\mathcal{C}$. It is also not hard to see that this way of obtaining lower bounds also bypasses the natural proof barrier of [FSV18, GKSS17]. The problem is that in most cases we obtained a hitting set for a class only after proving a lower bound for it.

In [FS18] Forbes and Shpilka defined the notion of a *robust* hitting set for a circuit class $\mathcal{C}$. Over fields of characteristic zero, a hitting set $\mathcal{H}$ for a class $\mathcal{C}$ is c-robust if it also satisfies that for every $f \in \mathcal{C}$ there is $\boldsymbol{v} \in \mathcal{H}$ such that $|f(\boldsymbol{v})| \geq c \cdot \|f\|$, where $\|\cdot\|$ is some fixed norm on $\mathbb{C}[\boldsymbol{x}]$ (see Definition 1.9 for a definition over arbitrary fields). It is not hard to see that if $\mathcal{H}$ is a robust hitting set for a class $\mathcal{C}$ then it also hits the closure of $\mathcal{C}$.

In this work we focus on depth-3 algebraic circuits, known as $\Sigma\Pi\Sigma$, and on $\mathrm{VP_e}$, the class of algebraic formulas, two classes for which we lack strong lower bounds, and in particular we do not have hitting sets for them. For $\Sigma\Pi\Sigma$ circuits the best lower bound is the near cubic lower bound of Kayal, Saha and Tavenas [KST16], and for $\mathrm{VP_e}$ the best lower bound is the quadratic lower bound of Kalarkoti [Kal85]. Recall that by the result of Valiant et al. [VSBR83], a super-quasipolynomial lower bound against $\mathrm{VP_e}$ implies a super-polynomial lower bound against VP. Similarly, a hitting set for $\mathrm{VP_e}$ implies a hitting set for VP. We also note that by a result of Gupta et al. [GKKS16], a strong enough lower bound or a hitting set for $\Sigma\Pi\Sigma$ imply both a lower bound for general circuits and a hitting set for them. This result also implies that a polynomial-time reconstruction algorithm for $\Sigma\Pi\Sigma$ circuits would give rise to a sub-exponential time *reconstruction algorithm* for general circuits. Recall that a reconstruction algorithm for a class $\mathcal{C}$ is an algorithm that, given black-box access to a circuit from $\mathcal{C}$, outputs a circuit in $\mathcal{C}$ that computes the same polynomial.

Instead of viewing robust hitting sets as a way to obtain hitting sets for the closure of circuit classes, we suggest to find subclasses of interesting classes, $\tilde{\mathcal{C}} \subset \mathcal{C}$, such that $\mathcal{C}$ is contained in the closure of $\tilde{\mathcal{C}}$, and aim to construct a robust hitting set for the subclass $\tilde{\mathcal{C}}$. This offers a new approach for constructing hitting sets for known classes and for obtaining lower bounds. Specifically, we consider subclasses of $\Sigma\Pi\Sigma$ and $\mathrm{VP_e}$ that are dense in their superclasses. Each of these subclasses is the orbit of some simple polynomial under the group of invertible affine transformations.

For $\mathrm{VP_e}$, we first consider a subclass that was defined by Bringmann, Ikenmeyer and Zuiddam [BIZ18]–the orbit of the so called *continuant* polynomial (see Definition 1.16). We give a polynomial-sized interpolating set[4] for this subclass as well as a polynomial-time deterministic reconstruction algorithm that uses as oracle a *root-finding algorithm*.[5] In particular, this implies a polynomial-time randomized reconstruction algorithm, and, in some cases, a polynomial-time deterministic algorithm.

In addition, we exhibit two other subclasses that are dense in $\mathrm{VP_e}$. The first class is defined as the orbit of read-once formulas (ROF for short, see Definition 5.1) and the second as the orbit of read-once formulas in *alternating normal form* (ROANF for short, see Definition 5.3). We obtain hitting sets for both classes and

[4]Recall that an interpolating set for a class $\mathcal{C}$ of polynomials in n variables, over a field $\mathbb{F}$, is a set of points $\mathcal{H} \subset \mathbb{F}^n$ such that for every $f \in \mathcal{C}$, the list of values $f(\mathcal{H})$ uniquely determines f. See Definition 1.11.

[5]A root-finding algorithm, over a field $\mathbb{F}$, when given black-box access to a univariate polynomial, outputs a root of that polynomial in $\mathbb{F}$, if such a root exists.

an interpolating set for the second. We also observe that the reconstruction algorithm of [GKQ14] works for the polynomials in the orbit of ROANFs. Although the results that we obtain for the subclass defined by the continuant polynomial are stronger, we think that every such dense subclass can shed more light on $\mathrm{VP_e}$ and may eventually be used in order to obtain new lower bounds.

For $\Sigma\Pi\Sigma$ we consider two subclasses. One is based on orbits of *sparse* polynomials (polynomials having polynomially many monomials) and the other on orbits of *diagonal* tensors (see Definition 1.29). We give a hitting set for the first, an interpolation set for the second, and we also observe that a slight modification of the randomized reconstruction algorithm of [KNS19] applies for the second class.

In particular, our results give the first dense subclasses inside $\mathrm{VP_e}$ and $\Sigma\Pi\Sigma$ for which a polynomial-size interpolating set is known as well as a polynomial-time reconstruction algorithm. By [VSBR83] our result immediately translate to VP, giving a dense subclass of for which a quasipolynomial-sized interpolating set is known as well as a quasipolynomial-time reconstruction algorithm.

If we could transform the interpolating sets that we have found to *robust hitting sets* for the orbits, then this will immediately give hitting sets for the closure of the orbits, i.e. for $\Sigma\Pi\Sigma$ and $\mathrm{VP_e}$, which, by [HS80] gives a lower bound for the class. Thus, our work raises an intriguing problem:

Problem 1.1. *Given an interpolating set for a class $\mathcal{C}$ construct a robust hitting set for $\mathcal{C}$.*

We stress that by our results, solving this problem would lead to hitting sets, and lower bounds, for $\mathrm{VP_e}$ and VP.

Another advantage for having small interpolating sets for dense subclasses is the following: One approach for searching for separating polynomials for a class, is by considering the map from circuits in the class to the coefficient vectors of the polynomials that they compute. That is, once we fix a computation graph, an assignment to the constants appearing in the circuit determines the output polynomial. Each coefficient is a polynomial in those constants, and as there are "few" constants (polynomially many for polynomially sized circuits), and there are exponentially many coefficients, there should be many polynomials vanishing on the closure of the image of this map. If we could get a good understanding of this map then perhaps we could use it to construct a polynomial that vanishes on all such coefficient vectors. This polynomial will vanish on all coefficient vectors of the superclass in which the subclass is dense. A different approach is to find a coefficient vector that is not in the closure of the image of this map (this is the approach of Raz in [Raz10]). Now, assume that $\mathcal{H}$ is an interpolating set for a dense subclass $\tilde{\mathcal{C}} \subset \mathcal{C}$. We know that the map $f \to f\big|_{\mathcal{H}}$ is one-to-one on $\tilde{\mathcal{C}}$. Thus, the list of values $f\big|_{\mathcal{H}}$ can be viewed as an efficient encoding that is given in terms of values of the computed polynomial. This provides a different encoding of a circuit – instead of the constants in it, use the evaluations on $\mathcal{H}$. Thus, by studying the closure of this map (i.e. the closure of the set of points on $\mathbb{F}^{|\mathcal{H}|}$ that can be obtained as evaluation vectors of polynomials in the subclass) we may be able to find a separating polynomial, or, as in Raz's approach, find an evaluation vector that is not obtained by any polynomial in the superclass. It is clear that one can also try this approach even if $\mathcal{H}$ is not an interpolating set, however, as interpolating sets "preserve information" of a dense set, we believe that such sets are better suited for this approach.

To conclude, focusing on dense subclasses and studying their properties could lead to better understanding of their superclasses and perhaps to breakthrough results in algebraic complexity.

To formally state our results we need some definitions that we give next.

1.1 Basic definitions

1.1.1 Circuit classes

Definition 1.2. *An algebraic formula (also called arithmetic formula) over a field $\mathbb{F}$, is a rooted tree whose leaves are labeled with either variable or scalars from $\mathbb{F}$, and whose root and internal nodes (called gates) are labeled with either "+" (addition) or "×" (multiplication). An algebraic formula computes a polynomial in the natural way. Each leaf computes the polynomial that labels it, and each gate computes either the sum or product of its children, depending on its label. The output of the formula is the polynomial computed at its root. The size of a formula is the number of wires in it. The depth of a formula is the length of the longest simple leaf-root path in it. The formula size of a polynomial f is defined as the smallest size of a formula that outputs f.*

A sequence $m(n)$ of natural numbers is called polynomially bounded if there exists a univariate polynomial q such that $m(n) \leq q(n)$ for all n.

The complexity class $\mathrm{VP_e}$ is defined as the set of all families of polynomials $(f_n)_n$, with $f_n \in \mathbb{F}[x_1, \ldots, x_n]$, whose formula size is polynomially bounded.

Definition 1.3. *An arithmetic circuit Φ is a $\Sigma^{[s]}\Pi^{[d]}$ circuit if it is a layered graph of depth-2, has a top gate labeled + with fan-in $\leq s$ and its second layer is comprised entirely of × gates with fan-in $\leq d$. In other words, $\Sigma^{[s]}\Pi^{[d]}$ compute polynomials of degree d with at most s monomials.*

Definition 1.4. *An arithmetic circuit Φ in n variables is a $\Sigma^{[s]}\Pi^{[d]}\Sigma$ circuit if it is a layered graph of depth-3, has a top gate labeled + with fan-in $\leq s$, its second layer is comprised entirely of × gates with fan-in $\leq d$, and its bottom layer is comprised of linear functions in $x_1, \ldots, x_n$. In other words, $\Sigma^{[s]}\Pi^{[d]}\Sigma$ circuit compute polynomials of the form*

$$f(\boldsymbol{x}) = \sum_{i=1}^{s} \prod_{j=1}^{d} (\alpha_{i,j,0} + \sum_{k=1}^{n} \alpha_{i,j,k} x_k) \ .$$

Given a family of circuits $\mathcal{C}$, we will sometime denote it as $\mathcal{C}(\mathbb{F})$ to stress that we allow coefficients to come from the field $\mathbb{F}$. Observe that the definitions of the classes above do not depend on the field and so we can define them over any field of our choice.

1.1.2 Approximate complexity

The following definition gives sense to the notion of approximation over arbitrary fields. In what follows we let ε be a new formal variable.[6] For a field $\mathbb{F}$ we denote with $\mathbb{F}[\varepsilon]$ the ring of polynomial expressions in ε over $\mathbb{F}$, and with $\mathbb{F}(\varepsilon)$ the fraction field of $\mathbb{F}[\varepsilon]$, i.e. the field of rational expressions in ε.

Definition 1.5. *Let $\mathcal{C}(\mathbb{F})$ be a circuit class over a field $\mathbb{F}$. The closure of $\mathcal{C}$, denoted $\overline{\mathcal{C}(\mathbb{F})}$, is defined as follows: A family of functions $(f_n)_n$, where $f_n \in \mathbb{F}[x_1, \ldots, x_n]$, is in $\overline{\mathcal{C}(\mathbb{F})}$ if there is a polynomially bounded function $m : \mathbb{N} \to \mathbb{N}$, and a family of functions $(g_{m(n)})_n \in \mathcal{C}(\mathbb{F}(\varepsilon))$, with $g_{m(n)} \in \mathbb{F}[\varepsilon][x_1, \ldots, x_{m(n)}]$, such that for all $n \in \mathbb{N}$,*

$$g_{m(n)}(x_1, \ldots, x_{m(n)}) = f_n(x_1, \ldots, x_n) + \varepsilon \cdot g_{n,0}(x_1, \ldots, x_{m(n)}) \ , \tag{1}$$

[6] Intuitively, one should think of ε as an infinitesimal quantity.

for some polynomial $g_{n,0} \in \mathbb{F}[\varepsilon][x_1, \ldots, x_{m(n)}]$. *Whenever an equality as in* (1) *holds we say that*

$$g_{m(n)} = f_n + O(\varepsilon) \quad or \quad f_n = g_{m(n)} + O(\varepsilon) \ .$$

In that case we think of $g_{m(n)}$ *as an "approximation" of* f_n, *and we say that the family* $(g_{m(n)})_n$ *approximates the family* $(f_n)_n$.

Alder [Ald84] have shown that over $\mathbb{C}$ it holds that $(f_n) \in \overline{\mathcal{C}(\mathbb{C})}$, in the sense of Definition 1.5, if and only if it is in the closure of $\mathcal{C}(\mathbb{C})$ in the usual sense. That is, if for every n there exists a sequence of polynomials $g_{n,k} \in \mathcal{C}(\mathbb{C})$ such that $\lim_{k\to\infty} g_{n,k} = f_n$, where convergence is taken coefficient wise. This result holds over $\mathbb{R}$ as well, see [LL89, Bür04].

Finally, we note that every matrix is approximable (in the sense of Definition 1.5) by a non-singular matrix (which is equivalent to being a limit of a sequence of non-singular matrices, in characteristic zero).

Observation 1.6. *For every* $A \in \mathbb{F}^{n\times n}$ *there exists a non-singular matrix* $B \in \mathbb{F}(\varepsilon)^{n\times n}$ *such that* $A = B + O(\varepsilon)$.

1.1.3 Hitting and interpolating sets

Definition 1.7. *A set of points* $\mathcal{H} \subseteq \mathbb{F}^n$ *is called a* hitting set *for a circuit class* $\mathcal{C}$ *(we also say that* $\mathcal{H}$ hits $\mathcal{C}$*) if for every circuit* $\Phi \in \mathcal{C}$, *computing a non-zero polynomial, there exists some* $\boldsymbol{a} \in \mathcal{H}$ *such that* $\Phi(\boldsymbol{a}) \neq 0$.

We next give the definition of a robust hitting set, a notion first defined in [FS18]. Here we extend the definition for arbitrary characteristic. We start by giving the definition of [FS18], over characteristic zero (and focus on $\mathbb{C}$) and then the more general definition.

Definition 1.8 (Following Definition 5.1 of [FS18])**.** *Let* $\|\cdot\|$ *be some norm on* $\mathbb{C}[\boldsymbol{x}]$. *A hitting set* $\mathcal{H}$ *for a circuit class* $\mathcal{C} \subseteq \mathbb{C}[\boldsymbol{x}]$ *is called* robust *if there exists some constant* $c > 0$ *such that, for every* $0 \neq f \in \mathcal{C}$,[7] *there exists some* $\boldsymbol{a} \in \mathcal{H}$ *such that* $|f(\boldsymbol{a})| \geq c \cdot \|f\|$.

For arbitrary characteristic we use the same approach as in Definition 1.5.

Definition 1.9. *Let* $\mathbb{F}$ *be a field of arbitrary characteristic. A hitting set* $\mathcal{H} \subset \mathbb{F}^n$ *for a circuit class* $\mathcal{C}(\mathbb{F})$ *is called* robust *if for every circuit* $\Phi \in \mathcal{C}(\mathbb{F}(\varepsilon))$ *computing a polynomial* $f(\boldsymbol{x}) = h(\boldsymbol{x}) + \varepsilon \cdot g(\boldsymbol{x})$, *where* $h(\boldsymbol{x}) \in \mathbb{F}[\boldsymbol{x}]$ *and* $g(\boldsymbol{x}) \in \mathbb{F}[\varepsilon][\boldsymbol{x}]$, *there exists some* $\boldsymbol{a} \in \mathcal{H}$ *such that* $f(\boldsymbol{a}) \notin \varepsilon \cdot \mathbb{F}[\varepsilon]$.

It is not hard to prove using the result of [Ald84] that for $\mathbb{F} = \mathbb{C}$, Definitions 1.8 and 1.9 are equivalent.

Observation 1.10. *If* $\mathcal{H}$ *is a finite robust hitting set for* $\mathcal{C}(\mathbb{F})$, *then* $\mathcal{H}$ *hits* $\overline{\mathcal{C}(\mathbb{F})}$ *as well.*

Proof. Consider $0 \neq f \in \overline{\mathcal{C}(\mathbb{F})}$. By Definition 1.5 there is $g \in \mathcal{C}(\mathbb{F}(\varepsilon))$, such that $f = g + O(\varepsilon)$. Clearly $g \neq 0$. Let $\boldsymbol{a} \in \mathcal{H}$ be such that $g(\boldsymbol{a}) \notin \varepsilon \cdot \mathbb{F}[\varepsilon]$. It follows that $f(\boldsymbol{a}) \notin \varepsilon \cdot \mathbb{F}[\varepsilon]$. In particular, $f(\boldsymbol{a}) \neq 0$. □

We next define the notion of an interpolating set.

Definition 1.11. *Let* $\mathcal{C}$ *be a class of* n*-variate polynomials. A set* $\mathcal{H} \subseteq \mathbb{F}^n$ *is called an* interpolating set *for* $\mathcal{C}$ *if, for every* $f \in \mathcal{C}$, *the evaluations of* f *on* $\mathcal{H}$ *uniquely determine* f.

[7] We abuse notation and write $f \in \mathcal{C}$ when f is the output of some circuit from $\mathcal{C}$.

Observation 1.12. *If $\mathcal{H}$ is a hitting set for $\mathcal{C}(\mathbb{F}) + \mathcal{C}(\mathbb{F}) \triangleq \{\alpha f + \beta g : f, g \in \mathcal{C}, \alpha, \beta \in \mathbb{F}\}$, then $\mathcal{H}$ is an interpolating set for $\mathcal{C}$.*

A common method for designing hitting and interpolating sets is via hitting set generators.

Definition 1.13. *A polynomial mapping $\mathcal{G} : \mathbb{F}^k \to \mathbb{F}^n$ is called a* hitting set generator *(or simply a generator) for a circuit class $\mathcal{C}(\mathbb{F})$ if for any non-zero n-variate polynomial $f \in \mathcal{C}$, the k-variate polynomial $f \circ \mathcal{G}$ is non-zero.*

Similarly, we call $\mathcal{G} : \mathbb{F}^k \to \mathbb{F}^n$ an interpolating set generator *for a circuit class $\mathcal{C}(\mathbb{F})$ if for any two different n-variate polynomials $f_1, f_2 \in \mathcal{C}$, the k-variate polynomial $(f_1 - f_2) \circ \mathcal{G}$ is non-zero.*

Generators immediately give rise to hitting sets.

Observation 1.14. *Let $\mathcal{G} : \mathbb{F}^k \to \mathbb{F}^n$ be a generator for $\mathcal{C}(\mathbb{F})$ such that the individual degree of each coordinate of $\mathcal{G}$ is at most r. Let $W \subset \mathbb{F}$ be any set of size $|W| = d \cdot r + 1$. Let $\mathcal{H} = \mathcal{G}\left(W^k\right)$. Then $\mathcal{H}$ hits every n-variate polynomial $f \in \mathcal{C}$ of degree at most d.*

Proof. As $\mathcal{G}$ is a generator, the k-variate polynomial $f \circ \mathcal{G}$ is nonzero. As its individual degrees are bounded by $d \cdot r$ it follows that at least one of the values in $(f \circ \mathcal{G})\left(W^k\right) = f(\mathcal{H})$ is not zero. □

1.1.4 k-independent maps

Our constructions rely on polynomial mappings $\mathcal{G}_k$, parameterized by some integer $k \leq n$, with the property that the image of $f \circ \mathcal{G}_k$ contains all projections of f to k variables. We call such a map a k-independent map.

Definition 1.15. *We call a polynomial mapping $\mathcal{G}(y_1, \ldots, y_t, z_1) : \mathbb{F}^{t+1} \to \mathbb{F}^n$ a* 1-independent polynomial map *if for every index $i \in [n]$ there exists an assignment $\boldsymbol{a}_i \in \mathbb{F}^t$ to $y_1, \ldots, y_t$ such that the ith coordinate of $\mathcal{G}(\boldsymbol{a}_i, z_1)$ is z_1, and the rest of the coordinates are 0. For $k > 1$, a polynomial mapping $\mathcal{G}(y_1, \ldots, y_{tk}, z_1, \ldots, z_k) : \mathbb{F}^{k(t+1)} \to \mathbb{F}^n$ is called a* k-independent polynomial map *(or a k-independent map) if $\mathcal{G}$ is a sum of k variable-disjoint 1-independent polynomial maps. We denote k-independent polynomial maps as $\mathcal{G}(\boldsymbol{y}, \boldsymbol{z})$ when k, t are implicit. The $\boldsymbol{y}$ variables are called* control variables*.*

A k-independent polynomial map $\mathcal{G}$ is called uniform *if all n coordinates of $\mathcal{G}$ are homogeneous polynomials of the same degree.*

1.1.5 The linear and affine groups and their actions

Given a matrix $A \in \mathbb{F}^{n \times n}$ and a tuple of variables $\boldsymbol{x} = (x_1, \ldots, x_n)$, we denote

$$A\boldsymbol{x} = \left(\sum_{i=1}^{n} A_{1,i} x_i, \sum_{i=1}^{n} A_{2,i} x_i, \ldots, \sum_{i=1}^{n} A_{n,i} x_i \right).$$

Let $n \geq m \in \mathbb{N}$. For an m-variate polynomial $f(x_1, \ldots, x_m) \in \mathbb{F}[x_1, \ldots, x_m]$, a matrix $A = (A_{i,j})_{i,j=1}^{n} \in \mathbb{F}^{n \times n}$ and a vector $\boldsymbol{b} = (b_1, \ldots, b_n) \in \mathbb{F}^n$, we define the n-variate polynomial $f(A\boldsymbol{x} + \boldsymbol{b})$ to be

$$f(A\boldsymbol{x} + \boldsymbol{b}) \triangleq f\left(\sum_{i=1}^{n} A_{1,i} x_i + b_1, \sum_{i=1}^{n} A_{2,i} x_i + b_2, \ldots, \sum_{i=1}^{n} A_{m,i} x_i + b_m \right). \tag{2}$$

Note that we ignored the last $n-m$ coordinates of $A\boldsymbol{x}+\boldsymbol{b}$.

We denote with $\mathrm{GL}_n(\mathbb{F})$ the group of invertible $n \times n$ matrices over $\mathbb{F}$, and with $\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})$ the group of invertible affine transformation, i.e. all the maps $\boldsymbol{x} \to A\boldsymbol{x}+\boldsymbol{b}$, where $A \in \mathrm{GL}_n(\mathbb{F})$ and $\boldsymbol{b} \in \mathbb{F}^n$.

For an m-variate polynomial f over $\mathbb{F}$, and $n \geq m$ we denote with $f^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})}$ the orbit of f under the natural action of $\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})$:[8]

$$f^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})} \triangleq \{f(A\boldsymbol{x}+\boldsymbol{b}) \mid A \in \mathrm{GL}_n(\mathbb{F}),\ \boldsymbol{b} \in \mathbb{F}^n\}\ .$$

We similarly define $f^{\mathrm{GL}_n(\mathbb{F})}$. More generally, for a class of m-variate polynomials $\mathcal{C}(\mathbb{F})$, we denote the *orbit* of $\mathcal{C}$ under $\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})$ by

$$\mathcal{C}^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})} \triangleq \{f(A\boldsymbol{x}+\boldsymbol{b}) \mid f \in \mathcal{C},\ A \in \mathrm{GL}_n(\mathbb{F}),\ \boldsymbol{b} \in \mathbb{F}^n\}\ .$$

We similarly define $\mathcal{C}^{\mathrm{GL}_n(\mathbb{F})}$. When we want to speak about orbits of families of polynomials from $\mathcal{C}(\mathbb{F})$, with arbitrary number of variables, we use the notation $\mathcal{C}^{\mathrm{GL}(\mathbb{F})}$ or $\mathcal{C}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$.

1.2 Our results

We first give our results for the class $\mathrm{VP_e}$ and then for the class of depth-3 circuits, for which it may be easier to obtain a robust hitting set, or prove super-polynomial lower bounds.

1.2.1 The continuant polynomial

Bringmann, Ikenmeyer and Zuiddam [BIZ18] defined the following polynomial (in Remark 3.14 of their paper), which they called the continuant polynomial:

Definition 1.16. *The continuant polynomial on n variables, $C_n(x_1,\ldots,x_n)$, is defined as the* trace *of the following matrix product:*

$$C_n(x_1,\ldots,x_n) \triangleq \mathit{Trace}\left(\begin{pmatrix} x_1 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_2 & 1 \\ 1 & 0 \end{pmatrix} \cdot \ldots \cdot \begin{pmatrix} x_n & 1 \\ 1 & 0 \end{pmatrix}\right). \tag{3}$$

We denote with $C^{GL^{aff}(\mathbb{F})}$ the class of families of polynomials $(f_n)_n$ such that $f_n \in \mathbb{F}[x_1,\ldots,x_n]$ and for some $m \leq n$, $f_n \in C_m^{GL_n^{aff}(\mathbb{F})}$.

A result of Allender and Wang implies that the polynomial $x_1 \cdot y_1 + \cdots + x_8 \cdot y_8$ is not in $\mathrm{C}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$ [AW16]. Thus, as a computational class it is very weak. However, Theorem 3.12 of [BIZ18] states that for every field $\mathbb{F}$ of characteristic different than 2, it holds that

$$\overline{\mathrm{C}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}} = \overline{\mathrm{VP_e}}\ . \tag{4}$$

We give a polynomial-size interpolating set for the class $\mathrm{C}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$ as well as a polynomial-time reconstruction algorithm for it. We first state a simple result that gives a hitting set for the class.

Theorem 1.17. *Let $f(x_1,\ldots,x_n) \in C_m^{GL_n^{aff}(\mathbb{F})}$, for $m \leq n$, and arbitrary $\mathbb{F}$. Then, for any uniform 1-independent polynomial map $\mathcal{G}$ over $\mathbb{F}$, $f \circ \mathcal{G} \neq 0$.*

[8] To be precise, the action is $((A,\boldsymbol{b}) \circ f)(\boldsymbol{x}) = f(A^T\boldsymbol{x}+\boldsymbol{b})$. This is required in order to make the action a homomorphism, however, for the groups that we consider it does not change the orbit.

As immediate corollary we get a hitting set for the class.

Corollary 1.18. *For every field $\mathbb{F}$, there is an explicit hitting set $\mathcal{H} \subset \mathbb{F}^n$, of size $|\mathcal{H}| = O\left(n^6\right)$, that hits every $0 \neq f \in C_m^{GL_n^{aff}(\mathbb{F})}$. If $|\mathbb{F}| < n^2$ then $\mathcal{H}$ is defined over a polynomial-sized extension field of $\mathbb{F}$, $\mathbb{K}$ such that $|\mathbb{K}| \geq n^2$.*

Theorem 1.19. *For every field $\mathbb{F}$, there is an explicit interpolating set $\mathcal{H} \subset \mathbb{F}^n$, of size $|\mathcal{H}| = O\left(n^{10}\right)$, for $\bigcup_{m=1}^{n} C_m^{GL_n^{aff}(\mathbb{F})}$. If $|\mathbb{F}| < n^2$ then $\mathcal{H}$ is defined over a polynomial-sized extension field of $\mathbb{F}$, $\mathbb{K}$ such that $|\mathbb{K}| \geq n^2$.*

Theorem 1.20. *There is a deterministic algorithm that given $\mathbb{F}$, an integer n, oracle access to a root-finding algorithm over $\mathbb{F}$, and black-box access to a polynomial $f(x_1, \ldots, x_n) \in C_m^{GL_n^{aff}(\mathbb{F})}$ (for any $m \leq n$), runs in polynomial-time and outputs linear functions $(\ell_1(x_1, \ldots, x_n), \ldots, \ell_m(x_1, \ldots, x_n))$ such that*

$$f(x_1, \ldots, x_n) = C_m\left(\ell_1(\boldsymbol{x}), \ldots, \ell_m(\boldsymbol{x})\right) .$$

If $|\mathbb{F}| < n^3$ then the algorithm will make queries from a polynomial-sized extension field of $\mathbb{F}$, $\mathbb{K}$, such that $|\mathbb{K}| \geq n^3$, and it also requires oracle access to a root-finding algorithm over $\mathbb{K}$.

1.2.2 Orbits of read-once formulas

Roughly, a read-once formula (ROF) is a formula in which every variable labels at most one leaf. However, following [SV15, SV14] we also allow gates of the formula to pass on their output wire a linear function of their polynomial (see Definition 5.1). We denote with $\mathrm{ROF}^{\mathrm{GL}(\mathbb{F})}$ the class of families of polynomials $(f_n)_n$, such that for every n there exists a ROF Φ, on $m \leq n$ variables, such that $f_n(x_1, \ldots, x_n) \in \Phi^{\mathrm{GL}_n(\mathbb{F})}$.

A ROF is in *alternating normal form* (ROANF) if it is a full binary tree of depth 2Δ with alternating layers of addition and multiplication gates. In particular, it is a ROF on 4^Δ many variables (see Definition 5.3).

We denote with ANF_Δ the canonical ROANF of depth 2Δ in which the leaves are labeled with the variables $x_1, \ldots, x_{4^\Delta}$ according to their order (see Definition 5.4). We denote with $\mathrm{ANF}^{\mathrm{GL}^{\mathrm{aff}}[\mathbb{F}]}$ the class of families of polynomials $(f_n)_n$, such that for every n there exists Δ such that $4^\Delta \leq n$ and $f_n(x_1, \ldots, x_n) \in \mathrm{ANF}_\Delta^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})}$.

We first make the following simple observation.

Theorem 1.21. *For every field $\mathbb{F}$, it holds that*

$$ANF^{GL^{aff}(\mathbb{F})} \subsetneq ROF^{GL(\mathbb{F})} \subsetneq VP_e(\mathbb{F}) . \tag{5}$$

However, when taking closures we get

$$\overline{ANF^{GL^{aff}(\mathbb{F})}} = \overline{ROF^{GL(\mathbb{F})}} = \overline{VP_e(\mathbb{F})} . \tag{6}$$

Our main results for ROFs and ROANFs are a construction of a hitting set for the orbit of ROFs, and an interpolating set for the orbit of ROANFs. Both constructions are obtained using independent polynomial maps (Definition 1.15).

Theorem 1.22. *Let $0 \neq f \in ROF^{GL_n^{aff}(\mathbb{F})}$ where the underlying ROF depends on 2^t variables, for $2^t \leq n$. Then, for any $(t+1)$-independent polynomial map $\mathcal{G}$, over $\mathbb{F}$, $f \circ \mathcal{G} \neq 0$.*

Corollary 1.23. *For every field $\mathbb{F}$, there is a hitting set $\mathcal{H} \subset \mathbb{F}^n$, of size $|\mathcal{H}| = n^{O(\log n)}$, that hits every $0 \neq f \in ROF^{GL_n^{aff}(\mathbb{F})}$. If $|\mathbb{F}| < n^2$ then $\mathcal{H}$ is defined over a polynomial-sized extension field of $\mathbb{F}$, $\mathbb{K}$ such that $|\mathbb{K}| \geq n^2$.*

Since a hitting set for all polynomials of the form $g - h$ where $g, h \in \mathcal{C}$ is the same as an interpolating set for $\mathcal{C}$, the following theorem gives an interpolating set for the orbit of ROANFs.

Theorem 1.24. *Let $f_1 = ANF_{\Delta_1}(A_1\boldsymbol{x} + \boldsymbol{b}_1), f_2 = ANF_{\Delta_2}(A_2\boldsymbol{x} + \boldsymbol{b}_2) \in ANF^{GL_n^{aff}(\mathbb{F})}$ and $f = f_1 - f_2$. Set $k \triangleq 2\max\{\Delta_1, \Delta_2\} + 7$ and let $\mathcal{G}$ be any uniform k-independent polynomial map, over $\mathbb{F}$. If $f \neq 0$ then $f \circ \mathcal{G} \neq 0$.*

Corollary 1.25. *For any field $\mathbb{F}$, the class $ANF_{\Delta}^{GL_n^{aff}(\mathbb{F})}$, for $4^{\Delta} \leq n$, admits an interpolating set $\mathcal{H} \subset \mathbb{F}^n$, of size $|\mathcal{H}| = n^{O(\Delta)}$. If $|\mathbb{F}| < n^2$ then $\mathcal{H}$ is defined over a polynomial-sized extension field of $\mathbb{F}$, $\mathbb{K}$, such that $|\mathbb{K}| \geq n^2$.*

Finally, we observe that the randomized algorithm of Gupta, Kayal And Qiao [GKQ14], for reconstructing *random algebraic formula* (for a natural definition of a random formula), yields a randomized reconstruction algorithm for $\mathrm{ANF}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{C})}$. Naturally, the reconstruction is up to the symmetry group of ROANFs.

Theorem 1.26 (A special case of Theorem 1.1 of [GKQ14])**.** *Let T be a finite subset of $\mathbb{C}$. Let $n, \Delta \geq 1$ be integers such that $s \triangleq 4^{\Delta} \leq n$. Given black-box access to the output f of a circuit $\Phi \in ANF^{GL_n^{aff}(\mathbb{C})}$, with probability at least $1 - \frac{n^2 s^{O(1)}}{|T|}$ (on internal randomness), Algorithm 6.9 of [GKQ14] successfully computes a tuple of s linearly independent linear functions $L = (\ell_1, \ldots, \ell_s) \in (\mathbb{C}[\boldsymbol{x}])^s$ such that $f = ANF_{\Delta}(\ell_1, \ldots, \ell_s)$, and the ℓ_is are identical to the labels of the leaves of Φ up to $TS_n(\mathbb{C})$-equivalence (see Definition 2.3). Moreover, the running time of the algorithm is poly$(n, s, \log(|T|))$.*

Remark 1.27. *Theorem 1.1 of [GKQ14] is stated only for characteristic zero fields. However, in Remark 6.10 they explain how to make the algorithm work over any characteristic, for a large enough field. Thus, Theorem 1.26 also holds over large enough fields in arbitrary characteristic.*

Remark 1.28. *As a direct implication of Theorem 1.24, the reconstruction algorithm of Theorem 1.26 can be converted into a zero-error algorithm, with expected quasipolynomial running time: Given black-box access to some $f_1 \in ANF^{GL^{aff}(\mathbb{F})}$, we define f_2 to be the output of the algorithm of Theorem 1.26 on input f_1, and then verify $f_1 = f_2$ using Corollary 1.25.*

1.2.3 Dense subclasses of $\Sigma\Pi\Sigma$

We start by defining the canonical diagonal tensor of degree d and rank s, $\mathrm{T}_{s,d} \in \mathbb{F}[x_{1,1}, \ldots, x_{s,d}]$, and the resulting class of polynomials $\mathcal{T}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$.

Definition 1.29. *Let $T_{s,d} \triangleq \sum_{i=1}^{s} \prod_{j=1}^{d} x_{i,j}$. I.e., it is a sum of s variable-disjoint monomials. For $n \geq s \cdot d$, we denote with $T_{s,d}^{GL_n^{aff}(\mathbb{F})}$ the orbit of $T_{s,d}$ over $\mathbb{F}$, under the action of the affine group. Finally, we denote with $\mathcal{T}^{GL^{aff}(\mathbb{F})}$ the class of families of polynomials $(f_n)_n$, such that for every n there exist s and d such that $n \geq s \cdot d$ and $f_n(x_1, \ldots, x_n) \in T_{s,d}^{GL_n^{aff}(\mathbb{F})}$.*

Clearly, $\mathrm{T}_{s,d}^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})} \subset \Sigma^{[s]}\Pi^{[d]}\Sigma$. We next define the class consisting of orbits of sparse polynomials.

Definition 1.30. *Let $\Sigma\Pi^{GL^{\mathit{aff}}(\mathbb{F})}$ denote the class of families of polynomials that are computed by orbits of depth-2 circuits, of polynomially bounded size, over $\mathbb{F}$. I.e., it is all families $(f_n)_n$, of polynomially bounded degree, such that for some polynomially bounded $m(n)$, there exist $\Sigma^{m(n)}\Pi^{\deg(f_n)}$ circuits Φ_m, in $k \le n$, many variables, such that $f_n \in \Phi_m^{GL_n^{\mathit{aff}}(\mathbb{F})}$.*

As before we first give the basic observation connecting all three classes.

Theorem 1.31. *For every field $\mathbb{F}$ it holds that*

$$\mathcal{T}^{GL^{\mathit{aff}}(\mathbb{F})} \subsetneq \Sigma\Pi^{GL^{\mathit{aff}}(\mathbb{F})} \subseteq \Sigma\Pi\Sigma(\mathbb{F}) \,,$$

and for fields of size $|\mathbb{F}| \ge n+1$

$$\Sigma\Pi^{GL^{\mathit{aff}}(\mathbb{F})} \subsetneq \Sigma\Pi\Sigma(\mathbb{F}) \,.$$

In addition,

$$\overline{\mathcal{T}^{GL^{\mathit{aff}}(\mathbb{F})}} = \overline{\Sigma\Pi^{GL^{\mathit{aff}}(\mathbb{F})}} = \overline{\Sigma\Pi\Sigma(\mathbb{F})} \,. \tag{7}$$

Our main results for this section are a quasipolynomial-size hitting set for the class $\Sigma\Pi^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$, and a polynomial-size interpolating set for $\mathcal{T}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$.

Theorem 1.32. *Let $0 \neq g \in \mathbb{F}[\boldsymbol{x}]$ have sparsity $\le 2^t$. Let $(A, \boldsymbol{b}) \in GL_n^{\mathit{aff}}(\mathbb{F})$, and $f(\boldsymbol{x}) = g(A\boldsymbol{x} + \boldsymbol{b})$. Then, for any $(t+1)$-independent polynomial map $\mathcal{G}$, $f \circ \mathcal{G} \neq 0$.*

Corollary 1.33. *For any integers s, d, n, there exists an explicit hitting set $\mathcal{H} \subset \mathbb{F}^n$, of size $|\mathcal{H}| = (nd)^{O(\log s)}$, such that $\mathcal{H}$ hits every nonzero polynomial $f \in \left(\Sigma^{[s]}\Pi^{[d]}\right)^{GL_n^{\mathit{aff}}(\mathbb{F})}$. If $|\mathbb{F}| \le n \cdot d$ then we let $\mathcal{H}$ be defined over an extension field $\mathbb{K}$ of $\mathbb{F}$ of size $|\mathbb{K}| > n \cdot d$.*

We next state our result concerning an interpolating set for $\mathcal{T}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$.

Theorem 1.34. *Let $n, s_1, s_2, d_1, d_2 \in \mathbb{N}$ be such that $n \ge s_1 \cdot d_1, s_2 \cdot d_2$. For $i \in \{1,2\}$ let $f_i \in T_{s_i,d_i}^{GL_n(\mathbb{F})}$, and let $f = f_1 - f_2$. If $f \neq 0$, then any uniform 6-independent polynomial map $\mathcal{G}$ satisfies $f \circ \mathcal{G} \neq 0$.*

Finally we note that the randomized reconstruction algorithm of Kayal and Saha [KS19a], which works for (as it is termed in their paper) "non-degenerate" homogeneous depth-3 circuits, works for $\mathcal{T}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$. This follows from the observation that $\mathcal{T}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$ circuits are always non-degenerate.

Theorem 1.35 (special case of Theorem 1 of [KS19a])**.** *Let $n, d, s \in \mathbb{N}$, $n \ge (3d)^2$ and $s \le (\frac{n}{3d})^{\frac{d}{3}}$. Let $\mathbb{F}$ be a field of characteristic zero or greater than ds^2. There is a randomized $\mathrm{poly}(n,d,s) = \mathrm{poly}(n,s)$ time algorithm which takes as input black-box access to a polynomial f that is computable by a $T_{s,d}^{GL_n^{\mathit{aff}}(\mathbb{F})}$ circuit, and outputs a $T_{s,d}^{GL_n^{\mathit{aff}}(\mathbb{F})}$ circuit Φ computing f with high probability. Furthermore, Φ is unique up to $TPS_{s,d}(\mathbb{F})$-equivalence (see Definition 2.6).*

Remark 1.36. *As in remark 1.28, Theorem 1.34 enables us to convert the reconstruction algorithm of Theorem 1.35 to a zero-error algorithm, with expected polynomial running time. Given black-box access to some $f_1 \in \mathcal{T}^{GL^{\mathit{aff}}(\mathbb{F})}$, we define f_2 to be the output of the algorithm of Theorem 1.35 on input f_1, and then verify $f_1 \equiv f_2$ by applying Theorem 1.34 to $f = f_1 - f_2$.*

1.2.4 Robust hitting sets?

As we showed in Observation 1.10, if a hitting set $\mathcal{H}$ for a circuit class $\mathcal{C}$ is *robust*, then $\mathcal{H}$ hits $\overline{\mathcal{C}}$ as well. It is thus natural to ask whether our interpolating sets are already robust. Our next result shows that the property of being a t-independent map, which was sufficient for the constructions in Theorems 1.17, 1.19, 1.22, 1.24, 1.32 and 1.34 (for the appropriate values of t), by itself is not sufficient for obtaining robust hitting sets. We prove this by constructing an independent polynomial map which gives rise to a provably non-robust hitting set. Our construction is the same as the one given by Forbes et al. [FSTW16] (Construction 6.3 in the full version).

Theorem 1.37. *Let $\mathbb{F}$ be of characteristic zero. For every t, there exists a uniform t-independent polynomial map $\mathcal{G}$ and a nonzero polynomial f such that $f \circ \mathcal{G} \equiv 0$, and f can be computed by a $\Sigma\Pi\Sigma$ formula of size $t^{O(\sqrt{t})}$. If $\mathbb{F}$ has a positive characteristic then f can be computed by a $\Sigma\Pi\Sigma$ formula of size t^t, or by a general formula of size $t^{O(\log t)}$. Furthermore, for a certain arrangement of the variables in a $\sqrt{n} \times \sqrt{n}$ matrix, f can be taken to be the determinant of any $(t+1) \times (t+1)$ minor.*

1.3 Polynomial Identity Testing

So far we discussed our work from the perspective of dense subclasses of classes for which no strong lower bounds are known. Here we put our work in the context of the polynomial identity testing problem.

Polynomial Identity Testing (PIT for short) is the problem of designing efficient deterministic algorithms for deciding whether a given arithmetic circuit computes the identically zero polynomial. PIT has many applications, e.g. deciding primality [AKS02], finding a perfect matching in parallel [FGT19, ST17] etc., and strong connection to circuit lower bounds [KI04, DSY09, CKS18, GKSS19]. See [SY10, Sax09, Sax14] for surveys on PIT and [KS19b] for a survey of algebraic hardness-randomness tradeoffs.

PIT is considered both in the white-box model, in which we get access to the graph of computation of the circuit, and in the black-box model in which we only get query access to the polynomial computed by the circuit. Clearly, a deterministic PIT algorithm in the black-box model is equivalent to a hitting set for the circuit class. In this work we only focus on the black-box model.

The continuant polynomial and algebraic branching programs: The continuant polynomial is trivially computed by width-2 *Algebraic Branching Programs* (ABPs). Recall that an ABP of depth-d and width-w computes polynomials of the form $\mathrm{Trace}\left(M_1(\boldsymbol{x}) \cdot \ldots \cdot M_d(\boldsymbol{x})\right)$, where each M_i is a $w \times w$ matrix whose entries contain variables or field elements. Ben-Or and Cleve proved that every polynomial in $\mathrm{VP_e}$ can be computed by a width-3 ABP of polynomial-size [BC92].

Raz and Shpilka gave the first polynomial-time white-box PIT algorithm for read-once ABPs (ABPs in which every variable can appear in at most one matrix) [RS05]. Forbes, Saptharishi and Shpilka gave the first quasipolynomial-sized hitting set for read-once ABPs (ROABPs) [FSS14]. This result was slightly improved in [GG20] for the case where the width of the ROABP is small. Anderson et al. gave a subexponential hitting set for read-k ABPs [AFS$^+$18]. We note that none of these models is strong enough to contain the orbit $\mathrm{C}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$. For ABPs that are not constant-read we do not have sub-exponential time PIT algorithms. Thus, the following is an interesting open problem (recall that by the result of Ben-Or and Cleve a PIT algorithm for width-3 ABPs works for $\mathrm{VP_e}$ as well).

Problem 1.38. *Give a sub-exponential time PIT algorithm for ABPs of width-2.*

Although we do not have a PIT algorithm for general branching programs, in [KNST18] Kayal et al. gave an average-case reconstruction algorithm for low width ABPs. Kayal, Nair and Saha obtained a significantly better algorithm in [KNS19]. Their algorithm succeeds w.h.p, provided the ABP satisfies four non-degeneracy conditions (these conditions are defined in Section 4.3 of [KNS19]). However, the ABP computing the continuant polynomial does not satisfy the non-degeneracy conditions that are required for their algorithm to work. Thus, Theorem 1.20 does not follow from [KNS19].

To the best of our knowledge, $\mathrm{C}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$ is the first natural[9] computational class that is dense in $\mathrm{VP_e}$ for which a polynomial (or even sub-exponential)-sized interpolating set (or a hitting set) is known.

Read-Once formulas: Hitting sets for read-once formulas were first constructed by Volkovich and Shpilka [SV15], who gave quasipolynomial-sized hitting set for the model, as well as a deterministic reconstruction algorithm of the same running time (earlier randomized reconstruction algorithms were known [BHH95, BB98]). Minahan and Volkovich obtained a polynomial-sized hitting set for the class, which led to a similar improvement in the running time of the reconstruction algorithm [MV18]. Anderson, van Melkebeek and Volkovich constructed a hitting set of size $n^{k^{O(k)}+O(k\log n)}$ for read-k formulas [AvMV15]. All these results work in a slightly stronger model in which we allow to label leaves with univariate polynomials, of polynomial degree, such that every variable appears in at most one polynomial, or with sparse polynomials on disjoint sets of variables.

The read-once models that we consider here, $\mathrm{ANF}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$ and $\mathrm{ROF}^{\mathrm{GL}(\mathbb{F})}$, can be viewed as read-once formulas composed with a layer of addition gates with the restriction that the bottom layer of additions computes linearly independent linear functions. We note that these models do not fall into any of the previously studied models, as a variable can appear in all the linear functions.

As is the case with $\mathrm{C}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$, our hitting sets for $\mathrm{ANF}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$ and $\mathrm{ROF}^{\mathrm{GL}(\mathbb{F})}$ are the first sub-exponential-sized hitting sets for natural dense subclasses of $\mathrm{VP_e}$.

Small depth circuits: The class of $\Sigma\Pi$ circuits was considered in many works, see e.g. [BT88, KS01] and polynomial-sized hitting sets were constructed. The class of $\Sigma\Pi\Sigma$ circuits also received a lot of attention but with lesser success. Dvir and Shpilka [DS07] and Karnin and Shpilka [KS08] gave the first quasipolynomial-time white-box and black-box PIT algorithms for $\Sigma^{[k]}\Pi^{[d]}\Sigma$ circuits, respectively. Currently, the best result is by Saxena and Seshadhri who gave a hitting set of size $(nd)^{O(k)}$ for such circuits [SS12]. In [dOSV16] a subexponential-size hitting set for *multilinear* $\Sigma\Pi\Sigma$ circuits was given. In [ASSS16], Agrawal et al. gave a hitting set of size $n^{O(1)}\cdot(kd)^{O(r)}$ for $\Sigma^{[k]}\Pi^{[d]}\Sigma$ circuits, where r is an upper bound on the *algebraic rank* of the multiplication gates in the circuit. Thus, known quasipolynomial-size hitting sets for subclasses of $\Sigma\Pi\Sigma$ circuits are known when the fan-in of the top gate is poly-logarithmic, or when the algebraic rank of the set of multiplication gates is poly-logarithmic. In contrast, polynomials in $\mathcal{T}^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})}$ and $\Sigma\Pi^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$, when viewed as $\Sigma\Pi\Sigma$ circuits, can have polynomially many multiplication gates and their algebraic rank can be n. On the other hand, the corresponding $\Sigma\Pi\Sigma$ circuits are such that the *different* linear functions that are computed at their bottom layer are linearly independent (when we view linear functions that are a constant multiple of each other as the same function). Thus, our Corollary 1.33 provides a hitting set for a new subclass of $\Sigma\Pi\Sigma$ circuits.

[9] It is hard to define what a natural class means, but, for example the set of all polynomials in $\mathrm{VP_e}$ with a nonzero free term has a trivial hitting set, but is not a "computational" subclass.

To the best of our knowledge, our results for $\mathcal{T}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$ and $\Sigma\Pi^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$ give the first sub-exponential size hitting sets for natural subclasses that are dense in $\Sigma\Pi\Sigma$.

1.4 More related work

Approximations in algebraic complexity were first studied by Bini et al. in the context of algorithms for matrix multiplication [BCRL79]. For more on the history of border rank in the context of matrix multiplication see notes of chapter 15 in [BCS13]. More recently, influenced by the GCT program, a lot of research was invested in trying to find polynomials characterizing tensors of small rank. See [Lan17] for a discussion on this approach. More recently, Kumar proved that *every* polynomial over $\mathbb{C}$ can be approximated by a $\Sigma^{[2]}\Pi\Sigma$ circuit (of exponential degree) [Kum20].

Very little is known about the closure of circuit classes. Forbes observed that the class of ROABPs is closed [For16]. I.e. ROABP = $\overline{\text{ROABP}}$. We are not aware of other collapses or separation between general "natural" classes and their closures.

Beside the reconstruction algorithms mentioned earlier, reconstruction algorithms are known for $\Sigma\Pi$ circuits [BT88, KS01]; for random depth three *powering* circuits [Kay12]; for set-multilinear $\Sigma\Pi\Sigma$ and ROABPs [BBB+00, KS06]; for $\Sigma\Pi\Sigma$ circuits with bounded top fan-in [Shp09, KS09, Sin16]; and for multilinear depth-4 circuits with a constant top fan-in [GKL12, BSV20].

In general, we do not expect the reconstruction problem to be solvable efficiently, as the problem of finding the minimal circuit computing a given polynomial is a notoriously hard problem. A detailed discussion on the hardness of reconstruction can be found in [KNS19].

1.5 Proof technique

Our proofs are based on the following simple yet important, and as far as we know novel, observations concerning k-independent polynomial maps. Specifically, our proofs are based on the following two claims:

1. If we have a hitting-set generator H for nonzero polynomials of the form $\frac{\partial f}{\partial x_1}$, for $f \in \mathcal{C}$, and if $\mathcal{G}$ is a 1-independent map then $H + \mathcal{G}$ hits every nonzero $f \in \mathcal{C}$. This is proved in Lemma 3.9.

2. Similarly, we prove that if we have a hitting-set generator H for nonzero polynomials of the form $f\big|_{\ell=0}(A\boldsymbol{x} + \boldsymbol{b})$, for $f \in \mathcal{C}$, a linear function ℓ, and an invertible affine transformation $(A, \boldsymbol{b})$, and if $\mathcal{G}$ is a 1-independent map then $H + \mathcal{G}$ hits every nonzero $f \in \mathcal{C}$. This follows from Lemma 3.10.

By applying these claims $k + r$ times we get that composition with a $(k + r)$-independent map allows to reduce the problem of hitting a class $\mathcal{C}$ to hitting polynomials of the form $\left.\frac{\partial^k f}{\partial x_{i_1} \partial x_{i_2} \cdots \partial x_{i_k}}\right|_{\ell_1 = \ldots = \ell_r = 0}$. Thus, if we could prove that for a class $\mathcal{C}$, there is such a sequence of derivatives and restrictions that simplifies the polynomials in it to a degree that they can be easily hit by some map H, then we conclude that $H + \mathcal{G}_{k+r}$, for a $(k + r)$-independent map $\mathcal{G}_{k+r}$, is a hitting set generator for $\mathcal{C}$.

It seems that all that is left to do is prove that for each of the orbits that we consider in Section 1.2 that is such small k and r. However, a potential problem is that a partial derivative of the polynomial $g(\boldsymbol{x}) = f(A\boldsymbol{x} + \boldsymbol{b})$ gives $\frac{\partial g}{\partial x_1} = \sum_{i=1}^{n} \frac{\partial f}{\partial y_i} \cdot \frac{\partial \ell_i}{\partial x_1}$, where ℓ_i is the ith coordinate of $A\boldsymbol{x} + \boldsymbol{b}$. Thus, it is no longer a derivative composed with an affine transformation but rather a sum of such derivatives, which could lead to polynomials outside of our class. For example, it is not hard ot prove that if we compose the ROF

$y_1 \cdot y_2 \cdot y_3$ with (x_1, x_1+x_2, x_1+x_3) and then take a derivative according to x_1, then the resulting polynomial, $\frac{\partial(x_1\cdot(x_1+x_2)\cdot(x_1+x_3))}{\partial x_1} = 3x_1^2 + 2x_1\cdot(x_2+x_3) + x_2\cdot x_3$, is not in the orbit of any ROF. The solution to this problem is to take a *directional derivative* in a direction coming from a *dual basis*. For example if $\ell_i(\boldsymbol{v}_j) = \delta_{i,j}$ then $\frac{\partial g}{\partial \boldsymbol{v}_1} = \frac{\partial f}{\partial x_1}(A\boldsymbol{x}+\boldsymbol{b})$ (see Lemma 3.8). Now, comes another important observation: If H is a hitting-set generator for nonzero polynomials of the form $\frac{\partial f}{\partial \boldsymbol{v}}$, for $f \in \mathcal{C}$ and a direction $\boldsymbol{v}$, and if $\mathcal{G}$ is a 1-independent map then $H+\mathcal{G}$ hits every nonzero $f \in \mathcal{C}$. The point is that if $\frac{\partial f}{\partial \boldsymbol{v}} \circ H \neq 0$ then for some i, $\frac{\partial f}{\partial x_i} \circ H \neq 0$ and the claim follows from the first claim above. Thus, composition with $(k+r)$-independent maps allows us to reduce the problem of hitting a class $\mathcal{C}$ to finding a generator for polynomials that are obtained as a restriction to a subspace of co-dimension r of a directional partial derivative of order k of polynomials in $\mathcal{C}$.

Let us demonstrate this idea for the case of orbits of sparse polynomials. I.e. to polynomials of the form $g(\boldsymbol{x}) = f(A\boldsymbol{x}+\boldsymbol{b})$, where the number of monomials in f is at most 2^t. It is not hard to see that there is a variable x_i such that if we consider $f\big|_{x_i=0}$ and $\frac{\partial f}{\partial x_i}$ then one of these polynomials has at most 2^{t-1} monomials.[10] Thus, after a a sequence of at most t partial derivatives and restrictions, we get to a polynomial with only one monomial that we can easily hit. Hence after at most t directional derivatives and restrictions to a subspace, we get that g is a product of linear forms, which we can easily hit. This proves that any $(t+1)$-independent map hits such nonzero polynomials g.

To obtain interpolating sets for our classes (and also a reconstruction algorithm for the orbit of the continuant polynomial), we prove that if two polynomials in the orbit, of any of the classes that we consider, are different, then there is a sequence of a few (directional) partial derivatives and restrictions that makes one of them zero while keeping the other nonzero. Using this and the ideas from above we construct our interpolating sets.

1.6 Discussion

As Theorem 1.37 shows, our hitting sets are not necessarily robust. It is thus an outstanding open problem to find a way to convert a hitting set to a robust one (recall Problem 1.1).

The following toy example demonstrates that converting a hitting set for a class $\mathcal{C}$ to a robust hitting set for $\mathcal{C}$, cannot be done in a black-box manner and one has to use information about $\mathcal{C}$ for that: let $\mathcal{C}(\mathbb{F})$ be the class of all polynomials with non-zero free term. A trivial hitting set for $\mathcal{C}$ would simply be the singleton set $\mathcal{H} = \{\mathbf{0}\}$. On the other hand, it is clear that $\overline{\mathcal{C}} = \mathbb{F}[\boldsymbol{x}]$, so making $\mathcal{H}$ robust would yield a hitting set for *all* polynomials. Note, however, that this is not a "computational class."

Another potential approach for obtaining robust hitting sets follows from the observation that the set of queries made by a non-adaptive deterministic black-box reconstruction algorithm, $\mathcal{A}$, for $\mathcal{C}$, which is *continuous* at 0 (i.e. at the identically zero polynomial) is a robust hitting set for $\mathcal{C}$. The reason is, that if $0 \neq f \in \overline{\mathcal{C}}$ and $\{f_k\}_{k=1}^{\infty} \subseteq \mathcal{C}$ converges to f, then for large enough k: $\|f_k\|_2 \geq \frac{1}{2}\|f\|_2 > 0$. As the f_k sequence converges and polynomial evaluation is continuous (and their evaluation vectors are bounded), the sequence $\boldsymbol{v}_k = f_k\big|_{\mathcal{H}} \subseteq \mathbb{C}^{|\mathcal{H}|}$ must also converge to some vector $\boldsymbol{v} = f\big|_{\mathcal{H}} \in \mathbb{C}^{|\mathcal{H}|}$. If $\boldsymbol{v} = \mathbf{0}$ then the continuity of $\mathcal{A}$ at $\mathbf{0}$ implies the coefficients of the polynomials $f_k(\boldsymbol{x})$ must also converge to zero, as $\mathcal{A}(\mathbf{0}) = 0$. This would contradict $\|f_k\|_2 \geq \frac{1}{2}\|f\|_2 > 0$ for large enough k, so $\boldsymbol{v} \neq \mathbf{0}$ and thus $\mathcal{H}$ hits $\overline{\mathcal{C}}$.

Thus, an interesting challenge is to derandomize the reconstruction algorithms given in Theorems 1.20, 1.26 and 1.35, hoping that the resulting algorithms are continuous at $\mathbf{0}$. We note however, that currently we do

[10] This is not exactly accurate – it only holds if f is not divisible by some variable x_i. However, the case where there is a monomial dividing f is also quite easy to handle as it is enough to hit the polynomial obtained after dividing by that monomial (since a composition with a 1-independent map keeps any nonzero linear function nonzero).

not even have efficient deterministic root-finding algorithms over $\mathbb{C}$. It is also known that in general, finding the minimal circuit for a polynomial can be very difficult. E.g., in [Hås90, Swe18] it was shown that the question of computing, or even approximating, tensor rank, for degree 3 tensors, is NP hard, over any field.

Remark 1.39. *In Theorem 1.34, we have seen that any uniform $O(\log(sn))$-independent polynomial map $\mathcal{G}$ is an interpolating set generator for $\mathcal{T}^{GL^{\text{aff}}(\mathbb{C})}$; i.e, $\mathcal{G}$ induces an interpolating set $\mathcal{H}$ for $\mathcal{T}^{GL^{\text{aff}}(\mathbb{C})}$. On the other hand, in Theorem 1.37, we constructed such a map $\mathcal{G}$, with the additional property that $\mathcal{G}$ is* not *a hitting set generator for $\Sigma\Pi\Sigma$ circuits. In particular, this implies that the induced (non-efficient) reconstruction map $\mathcal{A}$ (that takes $f(\mathcal{H})$ and returns a circuit computing f) is not continuous at $\mathbf{0}$.*

We conclude this section with a somewhat vague question.

Problem 1.40. *Find a "computational" class of polynomials $\mathcal{C}$ with a known hitting set $\mathcal{H}$, such that $\overline{\mathcal{C}} \neq \mathcal{C}$, and convert $\mathcal{H}$ to a robust hitting set.*

We note that the closure of $\Sigma \bigwedge \Sigma$ circuits (i.e. circuits computing polynomials of the form $\sum_i \ell_i(\boldsymbol{x})^d$, for linear functions ℓ_i) is contained in the class of commutative read-once algebraic branching programs (see [FSS14]). Thus, the hitting set for the latter class gives a robust hitting set for the former [FSS14]. However, we seek an example in which there is an "interesting" conversion of a hitting set to a robust one.

1.7 Organization

The paper is organized as follows. Section 2 contains some more basic notations and definitions as well as characterization of the groups of symmetries of ANF_Δ and of $\mathrm{T}_{s,d}$. In Section 3 we give properties and constructions of k-independent polynomial maps and prove Theorem 1.37. In Section 4 we study the continuant polynomial and prove Theorems 1.17, 1.19 and 1.20. In Section 5 we study orbits of ROFs and ROANFs and prove Theorems 1.21, 1.22, 1.24 and 1.26. Section 6 contains our results for subclasses of $\Sigma\Pi\Sigma$ circuits (Theorems 1.31, 1.32, 1.34 and 1.35). The appendix contains missing definitions that are required for explaining the reconstruction algorithm of [GKQ14].

2 Preliminaries

2.1 Notation

For $k \in \mathbb{N}$, we denote $[k] \triangleq \{1, 2, 3, \ldots, k\}$ and $[k]_0 \triangleq \{0, 1, 2, \ldots, k-1\}$. We use boldface lowercase letters to denote tuples of variables or vectors, as in $\boldsymbol{x} = (x_1, \ldots, x_n)$, $\boldsymbol{a} = (a_1, \ldots, a_m)$, when the dimension is clear from the context. For any two elements i, j coming from some set S (usually i and j will be numbers), $\delta_{i,j}$ equals 1 when $i = j$ and 0 otherwise. For every $m \in \mathbb{N}$ we denote with I_m the $m \times m$ identity matrix. When we wish to treat the entries of a matrix A as formal variables, we use boldface $\boldsymbol{A}$. We will note use capital bold face letters other than to denote such matrices.

For an exponent vector $\boldsymbol{a} = (a_1, \ldots, a_n) \in \mathbb{N}^n$, we denote $\boldsymbol{x}^{\boldsymbol{a}} \triangleq \prod_{i=1}^n x_i^{a_i}$. In some cases we shall consider "monomials" with respect to set of linear functions $\{\ell_i\}_{i=1}^m$: for an exponent vector $\boldsymbol{e} = (e_1, \ldots, e_m) \in \mathbb{N}^m$ we denote $\boldsymbol{\ell}^{\boldsymbol{e}} = \prod_{i=1}^m \ell_i^{e_i}$ and refer to it as an $\{\ell_i\}$-*monomial*. For a polynomial $f(\boldsymbol{x})$ we define the *monomial support* of f, denoted $\mathrm{mon}(f)$, as the set of monomials with non-zero coefficient in f. The *variable set* of f, denoted $\mathrm{var}(f)$, is the set of variables that f depends on. I.e., all variables that appear in $\mathrm{mon}(f)$. The individual degree of a variable x_i in $f(\boldsymbol{x})$ is the degree of f as a polynomial in x_i. A polynomial $f \in \mathbb{F}[\boldsymbol{x}]$

of $\deg(f) \leq 1$ is called a linear function, and if f is homogeneous then it is called a *linear form*. For a polynomial $f \in \mathbb{F}[\boldsymbol{x}]$ and an integer $k \in \mathbb{N}$ we denote by $f^{[k]}$ the degree-k homogeneous part of $f(\boldsymbol{x})$,i.e. the sum of all monomials of f of degree exactly k. In particular,

$$f(\boldsymbol{x}) = f^{[0]}(\boldsymbol{x}) + f^{[1]}(\boldsymbol{x}) + \ldots + f^{[\deg(f)]}(\boldsymbol{x}) \ .$$

Note that for a linear function f, $f^{[1]}$ is a linear form. We say that a polynomial f is homogeneous of degree k or that f is k-homogeneous if $f = f^{[k]}$. We say a set of linear functions $\{\ell_1(\boldsymbol{x}), \ldots, \ell_n(\boldsymbol{x})\} \subset \mathbb{F}[\boldsymbol{x}]$ is *linearly independent* if the set $\left\{\ell_i^{[1]}\right\}$ is linearly independent.[11] Given a polynomial $f(\boldsymbol{x})$, a subset of variables $\boldsymbol{y} \subseteq \{x_1, \ldots, x_n\}$ and an assignment to those variables $\boldsymbol{a} \in \mathbb{F}^{|\boldsymbol{y}|}$, we denote by $f\big|_{\boldsymbol{y}=\boldsymbol{a}} \in \mathbb{F}[\boldsymbol{x} \setminus \boldsymbol{y}]$ the polynomial resulting from assigning the values of $\boldsymbol{a}$ to the variables of $\boldsymbol{y}$ in $f(\boldsymbol{x})$. We sometimes abuse notation and write $\boldsymbol{y} \subseteq [n]$ to indicate the indices of the assigned variables instead of the variables themselves.

Given an arithmetic circuit Φ, we frequently denote by $\Phi(\boldsymbol{x})$ or, abusing notation, by Φ, the polynomial computed at the output node of Φ. Given a class of arithmetic circuits $\mathcal{C}$ and a polynomial $f \in \mathbb{F}[\boldsymbol{x}]$, we say $f \in \mathcal{C}$ if f can be computed by some circuit from $\mathcal{C}$. For a circuit class $\mathcal{C}(\mathbb{F})$ we denote by $\overline{\mathcal{C}}(\mathbb{F})$ the *closure* of $\mathcal{C}(\mathbb{F})$, as in Definition 1.5.

2.2 Groups of matrices and their action

We first list some simple properties of composition with a linear (or affine) transformation that we shall use implicitly.

Observation 2.1. *For any m variate polynomial $f(x_1, \ldots, x_m)$ and $n \geq m$:*

- *For any $A \in GL_n(\mathbb{F})$ and $d \in \mathbb{N}$, $f^{[d]}(A\boldsymbol{x})$ is the d-homogeneous part of $f(A\boldsymbol{x})$.*
- *For any $A \in GL_n^{aff}(\mathbb{F})$, $f(\boldsymbol{x})$ is irreducible if and only if $f(A\boldsymbol{x})$ is irreducible.*
- *The set of matrices A for which $f(\boldsymbol{x}) = f(A\boldsymbol{x})$ forms a multiplicative subgroup of $GL_n(\mathbb{F})$ and a similar claim holds for $GL_n^{aff}(\mathbb{F})$.*

We next define some special groups that serve as group of symmetries of some of the models that we consider. We first define the group of symmetries of $\mathrm{ANF}_\Delta(\boldsymbol{x})$.

Definition 2.2. *For $m, \Delta \in \mathbb{N}$ such that $m = 2^\Delta$, the* tree-symmetry group *$TR_m(\mathbb{F})$ denotes the automorphisms of a rooted complete binary tree of depth Δ. It is defined recursively as follows.*

- *For $m = 1$, $TR_1(\mathbb{F})$ consists only of the identity matrix.*
- *For $m > 0$, $TR_m(\mathbb{F})$ is generated by matrices of the form*

$$\begin{pmatrix} A & 0 \\ 0 & B \end{pmatrix} \quad \text{and} \quad \begin{pmatrix} 0 & I_{\frac{m}{2}} \\ I_{\frac{m}{2}} & 0 \end{pmatrix}$$

where $A, B \in TR_{\frac{m}{2}}(\mathbb{F})$.

[11] Note that by our definition, x and $x+1$ are linearly dependent.

Definition 2.3. *For any $m = 4^{\Delta}$, the* tree-scale group *$TS_m(\mathbb{F})$ is the group generated by elements of $TR_m(\mathbb{F})$ and matrices of the form*

$$\begin{pmatrix} \alpha I_{\frac{m}{4}} & 0 & 0 & 0 \\ 0 & \alpha^{-1} I_{\frac{m}{4}} & 0 & 0 \\ 0 & 0 & \beta I_{\frac{m}{4}} & 0 \\ 0 & 0 & 0 & \beta^{-1} I_{\frac{m}{4}} \end{pmatrix}$$

where $0 \neq \alpha, \beta \in \mathbb{F}$.

The importance of the group $\mathrm{TS}_m(\mathbb{F})$ stems from the fact that it is the symmetry group of ANF_{Δ}. To intuitively see why this is the case, notice that in any representation of an ANF one may swap children of any node without changing the output polynomial. We call such symmetries "tree-symmetries" and they are captured by the group $\mathrm{TR}_n(\mathbb{F})$. A second source of ambiguity comes from the fact that we can rescale the formula. Recall that the output polynomial is of the form $f_1 \cdot f_2 + f_3 \cdot f_4$ (Definition 5.3). Clearly, the output does not change if we replace f_1 by, say, $2f_1$ and f_2 by $f_2/2$. Such rescaling symmetries are captured by the group $\mathrm{TS}_n(\mathbb{F})$. Finally, another source for ambiguity comes from the fact that the quadratic polynomials computed at the bottom two layers of the ANF may have different representations. For example,

$$4xy + 4wz = (x + y + w - z) \cdot (x + y - w + z) + (w + z + x - y) \cdot (w + z - x + y) \,.$$

As there is an infinite number of representations for each quadratic polynomial (over infinite fields), we can expect to characterize the symmetries in term of the quadratics computed at the bottom two layers of the ANF.

Fact 2.4 (Special case of Theorem 5.43(iii) of [GKQ14])**.** *Let $m, \Delta, n \in \mathbb{N}$ such that $m = 4^{\Delta-1} \leq n/4$. Let $f = ANF_{\Delta}(\ell_1, \ldots, \ell_{4m}) \in ANF_{\Delta}^{GL_n^{aff}(\mathbb{F})}$. Let $Q = (q_1, \ldots, q_m)$ be the list of quadratic polynomials that are computed at the bottom two layers of the formula $ANF_{\Delta}(\ell_1, \ldots, \ell_{4m})$. In particular, $f = ANF_{\Delta-1}(q_1, \ldots, q_m)$. If $Q' = (q'_1, \ldots, q'_m)$ is any other m-tuple of quadratic polynomials for which $f = ANF_{\Delta-1}(q'_1, \ldots, q'_m)$ then Q is $TS_m(\mathbb{F})$-equivalent to Q'.*

Next, we define the group of symmetries of $\mathrm{T}_{s,d}(\boldsymbol{x})$.

Definition 2.5. *For any $n \in \mathbb{N}$ the* permutation-scale group*, denoted $PS_n(\mathbb{F})$, is the set of all matrices $A \in GL_n(\mathbb{F})$ which are row-permutations of non-singular diagonal matrices with determinant one.*

For example, $\begin{pmatrix} 0 & -2 & 0 \\ 0 & 0 & -1 \\ 1/2 & 0 & 0 \end{pmatrix} \in \mathrm{PS}_3(\mathbb{C})$.

Definition 2.6. *Let $s, d, n \in \mathbb{N}$ such that $n = s \cdot d$. A matrix $A \in GL_n(\mathbb{F})$ is a member of the* tensor permutation-scale group*, denoted $TPS_{s,d}(\mathbb{F})$, if $A = (P \otimes I_d) \cdot B$, where P is an $s \times s$ permutation matrix and $B = \begin{pmatrix} B_1 & 0 & \ldots & 0 \\ 0 & B_2 & \ldots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \ldots & 0 & B_d \end{pmatrix}$ is a block diagonal matrix such that each block B_i of B satisfies $B_i \in PS_d(\mathbb{F})$.*

For example, for $s = d = 2$ the matrix $A = \begin{pmatrix} 0 & 0 & 0 & 2 \\ 0 & 0 & 1/2 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \end{pmatrix}$ is in $\mathrm{TPS}_{2,2}(\mathbb{C})$, as for $P = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$ and

$B = \begin{pmatrix} -1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 2 \\ 0 & 0 & 1/2 & 0 \end{pmatrix}$, we have $A = (P \otimes I_2) \cdot B$, and clearly each block of B is in $\mathrm{PS}_2(\mathbb{C})$.

Another way of defining the group is as follows: index rows and columns of A with pairs $(i,j) \in [s] \times [d]$. Then, $A \in \mathrm{TPS}_{s,d}(\mathbb{F})$ if and only if there exists a permutation $\pi : [s] \to [s]$, and for all $i \in [s]$ permutations $\theta_i : [d] \to [d]$ and constants $\alpha_{i,j}$ satisfying $\prod_{j=1}^{d} \alpha_{i,j} = 1$, such that $A_{(i,j),(i',j')} = \delta_{\pi(i),i'} \cdot \delta_{\theta_i(j),j'} \cdot \alpha_{i,j}$ for all i,j. We next prove that $\mathrm{TPS}_{s,d}(\mathbb{F})$ is the group of symmetries of $\mathrm{T}_{s,d}(\boldsymbol{x})$. In other words, we show that $\mathrm{T}_{s,d}(\boldsymbol{x}) = \mathrm{T}_{s,d}(A\boldsymbol{x})$ if and only if $A \in \mathrm{TPS}_{s,d}(\mathbb{F})$. Intuitively, $\mathrm{T}_{s,d}$ admits no symmetries other than the trivial ones: permutations on the product gates, and internal permutation-scale of each product gate such that the product of the scale coefficients is 1. This is exactly captured by the group $\mathrm{TPS}_{s,d}(\mathbb{F})$, which is therefore contained in the group of symmetries of $\mathrm{T}_{s,d}(\boldsymbol{x})$.

Lemma 2.7. *Let $s,d,n \in \mathbb{N}$, such that $d > 2$ and $n = s \cdot d$. If $A \in GL_n(\mathbb{F})$ satisfies $T_{s,d}(\boldsymbol{x}) = T_{s,d}(A\boldsymbol{x})$, then $A \in TPS_{s,d}(\mathbb{F})$.*

Proof of Lemma 2.7. Fix linear forms $\ell_{1,1}, \ldots, \ell_{s,d}$ such that the (i,j)th coordinate of $A\boldsymbol{x}$ (using the indexing $[n] = [s] \times [d]$) is $\ell_{i,j}(\boldsymbol{x})$, and $\mathrm{T}_{s,d}(A\boldsymbol{x}) = \sum_{i=1}^{s} \prod_{j=1}^{d} \ell_{i,j}(\boldsymbol{x})$. By the discussion above, our goal is to prove that there exists a permutation $\pi : [s] \to [s]$, and for all $i \in [s]$ permutations $\theta_i : [d] \to [d]$ and constants $\alpha_{i,j}$ satisfying $\prod_{j=1}^{d} \alpha_{i,j} = 1$, such that $\ell_{i,j}(\boldsymbol{x}) = \alpha_{i,j} \cdot x_{\pi(i),\theta_i(j)}$ for all i,j. Fix some $i \in [s]$ and take a derivative of the equation $\mathrm{T}_{s,d}(\boldsymbol{x}) = \mathrm{T}_{s,d}(A\boldsymbol{x})$ by $x_{i,1}$:

$$\prod_{j \in \{2,\ldots,d\}} x_{i,j} = \frac{\partial \mathrm{T}_{s,d}(\boldsymbol{x})}{\partial x_{i,1}} = \frac{\partial \mathrm{T}_{s,d}(A\boldsymbol{x})}{\partial x_{i,1}} = \sum_{r=1}^{s} \frac{\partial}{\partial x_{i,1}} \left(\prod_{j=1}^{d} \ell_{r,j}(\boldsymbol{x}) \right). \tag{8}$$

For $r \in [s]$, denote $h_{i,r}(\boldsymbol{x}) \triangleq \frac{\partial}{\partial x_{i,1}} \left(\prod_{j=1}^{d} \ell_{r,j}(\boldsymbol{x}) \right)$. As $d > 2$, the LHS of Equation (8) is a reducible polynomial, so $\sum_{r=1}^{s} h_{i,r}(\boldsymbol{x})$ is also reducible. Composition with a non-singular matrix preserves reducibility, so $\sum_{r=1}^{s} h_{i,r}(A^{-1}\boldsymbol{x})$ is also reducible. However, $h_{i,1}(A^{-1}\boldsymbol{x}), \ldots, h_{i,s}(A^{-1}\boldsymbol{x})$ are s variable-disjoint, multilinear polynomials, each of which is either $(d-1)$-homogeneous or zero. Thus, by Observation 2.8 below, at most one $h_{i,r}(A^{-1}x)$ can be non-zero. Accordingly, for every variable $x_{i,j}$ there exists a unique i' such that $x_{i,j} \in \mathrm{var}\left(\prod_{j'=1}^{d} \ell_{i',j'}(\boldsymbol{x}) \right)$. Thus, for some i' we have

$$\prod_{j \in \{2,\ldots,d\}} x_{i,j} = \frac{\partial}{\partial x_{i,1}} \left(\prod_{j=1}^{d} \ell_{i',j}(\boldsymbol{x}) \right). \tag{9}$$

For any $j > 1$, if we take a derivative of (9) by $x_{i,j}$ then the LHS is clearly non-zero. Thus, both $x_{i,1}$ and $x_{i,j}$ exist in $\mathrm{var}\left(\prod_{j'=1}^{d} \ell_{i',j'}(\boldsymbol{x}) \right)$, proving variables in the same product gate of $\mathrm{T}_{s,d}(\boldsymbol{x})$ are mapped to the same product gate of $\mathrm{T}_{s,d}(A\boldsymbol{x})$. A similar argument shows that variables from distinct product gates of $\mathrm{T}_{s,d}(\boldsymbol{x})$ are mapped to different product gates of $\mathrm{T}_{s,d}(A\boldsymbol{x})$. It follows that product gates of $\mathrm{T}_{s,d}(A\boldsymbol{x})$ are variable-disjoint and that there exists a permutation $\pi : [s] \to [s]$ satisfying

$$\forall i \in [s]: \qquad \mathrm{var}\left(\prod_{j=1}^{d} \ell_{i,j}(\boldsymbol{x}) \right) = \{ x_{\pi(i),1}, \ldots, x_{\pi(i),d} \}.$$

In particular, there can be no cancellations between different product gates of $\mathrm{T}_{s,d}(A\boldsymbol{x})$. Therefore, by

multilinearity, for every $i \in [s]$, the linear forms $\ell_{i,1}(\boldsymbol{x}), \ldots, \ell_{i,d}(\boldsymbol{x})$ must be variable-disjoint. Exactly d variables appear in $\prod_{j=1}^{d} \ell_{i,j}(\boldsymbol{x})$, so for every $i \in [s]$ and $j \in [d]$ there exists a permutation $\theta_i : [d] \to [d]$ and a non-zero constant $\alpha_{i,j} \in \mathbb{F}$ such that $\ell_{i,j}(\boldsymbol{x}) = \alpha_{i,j} x_{\pi(i),\theta_i(j)}$. As $\prod_{j=1}^{d} \alpha_{i,j}$ is the coefficient of $\prod_{j=1}^{d} x_{\pi(i),j}$ in $\mathrm{T}_{s,d}(A\boldsymbol{x})$, this product must be 1, which completes the proof. □

Observation 2.8. *If f, g are non-constant, variable-disjoint, multilinear polynomials, then for every $c \in \mathbb{F}$ the polynomial $f(\boldsymbol{x}) + g(\boldsymbol{x}) + c$ is irreducible.*

3 k-independent polynomial maps and their properties

All the hitting and interpolating sets that we construct are based on k-independent polynomial maps (Definition 1.15). We next give some simple properties of independent polynomial maps, that follow immediately from the definition.

Observation 3.1. *It holds that*

1. *If $\mathcal{G}(\boldsymbol{y}, \boldsymbol{z})$ is a $(k+1)$-independent polynomial map, then there exists a subset of variables S and an assignment $\boldsymbol{\alpha} \in \mathbb{F}^{|S|}$ such that $\mathcal{G}\big|_{S=\boldsymbol{\alpha}}$ is a k-independent polynomial map.*
2. *For any $k \geq 1$, the n coordinates of any k-independent polynomial map are $\mathbb{F}$-linearly independent.*
3. *Let $\ell_1(\boldsymbol{x})$ and $\ell_2(\boldsymbol{x})$ be linearly independent linear functions in $\mathbb{F}[\boldsymbol{x}]$. Let $\mathcal{G}(\mathbf{y}, z_1, z_2)$ be any 2-independent polynomial map. Consider $\ell_1 \circ \mathcal{G}$ and $\ell_2 \circ \mathcal{G}$ as polynomials in z_1, z_2 over $\mathbb{F}(\mathbf{y})$. Then, $(\ell_1 \circ \mathcal{G})^{[1]}$ and $(\ell_2 \circ \mathcal{G})^{[1]}$ are linearly independent, as linear forms in z_1, z_2 over $\mathbb{F}(\mathbf{y})$.*

We next give the construction of [SV15] of a k-independent polynomial map (denoted G_k in [SV15]).

Definition 3.2. *Fix n and a set of n distinct field elements $\mathcal{A} = \{\alpha_1, \ldots, \alpha_n\} \subseteq \mathbb{F}$.[12] For every $i \in [n]$ let $L_i(w) : \mathbb{F} \to \mathbb{F}$ be the ith Lagrange Interpolation polynomial for the set $\mathcal{A}$. That is, each $L_i(w)$ is polynomial of degree $n-1$ that satisfies $L_i(\alpha_j) = \delta_{i,j}$. We define $\mathcal{G}_1^{SV}(y_1, z_1) : \mathbb{F}^2 \to \mathbb{F}^n$ as:*

$$\mathcal{G}_1^{SV}(y_1, z_1) \triangleq (L_1(y_1) \cdot z_1, L_2(y_1) \cdot z_1, \ldots, L_n(y_1) \cdot z_1),$$

and for any $k \geq 1$, we define $\mathcal{G}_k^{SV} : \mathbb{F}^{2k} \to \mathbb{F}^n$ as:

$$\mathcal{G}_k^{SV}(\boldsymbol{y}, \boldsymbol{z}) \triangleq \mathcal{G}_1^{SV}(y_1, z_1) + \mathcal{G}_1^{SV}(y_2, z_2) + \ldots + \mathcal{G}_1^{SV}(y_k, z_k) = \left(\sum_{j=1}^{k} L_1(y_j) \cdot z_j, \sum_{j=1}^{k} L_2(y_j) \cdot z_j, \ldots, \sum_{j=1}^{k} L_n(y_j) \cdot z_j \right).$$

Observation 3.3. *$\mathcal{G}_k^{SV}$ is a k-independent polynomial map, in which each variable has degree at most $n-1$.*

The generator $\mathcal{G}_k^{\mathrm{SV}}$ can be converted to a uniform k-independent polynomial map by adding another k control variables $y_{k+1}, \ldots, y_{2k}$, and swapping out the $L_i(y_j)$s for their homogenizations $y_{j+k}^{n-1} L_i\left(\frac{y_j}{y_{j+k}}\right)$:

Definition 3.4. *With the notation used in Definition 3.2, define the* uniform SV-generator *with k independence $\mathcal{G}_k^{SV\text{-}hom} : \mathbb{F}^{3k} \to \mathbb{F}^n$ as:*

$$\mathcal{G}_k^{SV\text{-}hom}(y_1, \ldots, y_{2k}, z_1, \ldots, z_k) \triangleq y_{1+k}^{n-1} \cdot \mathcal{G}_1^{SV}\left(\frac{y_1}{y_{1+k}}, z_1\right) + y_{2+k}^{n-1} \cdot \mathcal{G}_1^{SV}\left(\frac{y_2}{y_{2+k}}, z_2\right) + \ldots + y_{2k}^{n-1} \cdot \mathcal{G}_1^{SV}\left(\frac{y_k}{y_{2k}}, z_k\right)$$

[12] If $|\mathbb{F}| < n$ then we take these elements from an appropriate extension field of $\mathbb{F}$.

$$= \left(\sum_{j=1}^{k} y_{j+k}^{n-1} L_1\left(\frac{y_j}{y_{j+k}}\right) \cdot z_j, \sum_{j=1}^{k} y_{j+k}^{n-1} L_2\left(\frac{y_j}{y_{j+k}}\right) \cdot z_j, \ldots, \sum_{j=1}^{k} y_{j+k}^{n-1} L_n\left(\frac{y_j}{y_{j+k}}\right) \cdot z_j \right).$$

Observation 3.5. *$\mathcal{G}_k^{SV\text{-}hom}$ is a uniform k-independent polynomial map, with individual degrees at most $n-1$.*

We next show how we can use k-independent polynomial maps in order to, roughly, simulate a kth order directional derivative or, project a polynomial to a subspace of co-dimension k. We first need to define the notion of a directional derivative.

Definition 3.6. *For an n-variate polynomial $f \in \mathbb{F}[\boldsymbol{x}]$ and $\boldsymbol{v} = (v_1, \ldots, v_n) \in \mathbb{F}^n$, the* derivative of $f(\boldsymbol{x})$ in the direction $\boldsymbol{v}$ *is defined as:*

$$\frac{\partial f}{\partial \boldsymbol{v}} = \sum_{i=1}^{n} v_i \cdot \frac{\partial f}{\partial x_i}.$$

If $\mathbb{F}$ has positive characteristic then by $\frac{\partial F}{\partial x_i}$ we refer to the formal derivative (which in the case of fields of characteristic zero is equal to the analytical definition). Observe that we still have that

$$\frac{\partial^2 f}{\partial y \partial x} = \frac{\partial^2 f}{\partial x \partial y}, \quad \frac{\partial (fg)}{\partial x} = \frac{\partial f}{\partial x} \cdot g + \frac{\partial g}{\partial x} \cdot f \quad \text{and} \quad \frac{\partial f\left(g_1(\boldsymbol{x}), \ldots, g_m(\boldsymbol{x})\right)}{\partial x_k} = \sum_{i=1}^{m} \frac{\partial f}{\partial y_i}\left(g_1(\boldsymbol{x}), \ldots, g_m(\boldsymbol{x})\right) \cdot \frac{\partial g_i}{\partial x_k},$$

where in the last expression f is an m variate polynomial, and $g_1, \ldots, g_m$ are n variate polynomials.

We shall often take derivatives according to a *dual set* to a set of linearly independent linear functions:

Definition 3.7. *A dual set for m linearly independent linear functions (recall that we say that linear functions are linearly independent if and only if their degree-1 homogeneous parts are linearly independent) in $n \geq m$ variables, $\ell_1(\boldsymbol{x}), \ldots, \ell_m(\boldsymbol{x})$ is a set of m vectors $\{\boldsymbol{v}_i\} \subset \mathbb{F}^n$ such that $\ell_i^{[1]}(\boldsymbol{v}_j) = \delta_{i,j}$.*

Lemma 3.8. *Let $\ell_1, \ldots, \ell_m \in \mathbb{F}[x_1, \ldots, x_n]$, for $n \geq m$, be linearly independent linear functions. Let $\{\boldsymbol{v}_i\} \subset \mathbb{F}^n$ be a dual set. Let $g \in \mathbb{F}[y_1, \ldots, y_m]$ be a polynomial. Then, for $f(\boldsymbol{x}) = g\left(\ell_1(\boldsymbol{x}), \ldots, \ell_m(\boldsymbol{x})\right)$ it holds that*

$$\frac{\partial f}{\partial \boldsymbol{v}_i}(\boldsymbol{x}) = \frac{\partial g}{\partial y_i}\left(\ell_1(\boldsymbol{x}), \ldots, \ell_m(\boldsymbol{x})\right) .$$

Proof.

$$\begin{aligned}\frac{\partial f}{\partial \boldsymbol{v}_i}(\boldsymbol{x}) &= \sum_j v_{i,j} \cdot \frac{\partial f}{\partial x_j}(\boldsymbol{x}) = \sum_{j,k} v_{i,j} \cdot \frac{\partial \ell_k}{\partial x_j} \cdot \frac{\partial g}{\partial y_k}\left(\ell_1(\boldsymbol{x}), \ldots, \ell_m(\boldsymbol{x})\right) \\ &= \sum_k \ell_k^{[1]}(\boldsymbol{v}_i) \cdot \frac{\partial g}{\partial y_k}\left(\ell_1(\boldsymbol{x}), \ldots, \ell_m(\boldsymbol{x})\right) = \frac{\partial g}{\partial y_i}\left(\ell_1(\boldsymbol{x}), \ldots, \ell_m(\boldsymbol{x})\right) .\end{aligned}$$

□

Lemma 3.9. *Let $f \in \mathbb{F}[\boldsymbol{x}]$ where $\boldsymbol{x} = (x_1, \ldots, x_n)$. Let $H(\boldsymbol{w}) : \mathbb{F}^t \to \mathbb{F}^n$ be a polynomial map in variables $\boldsymbol{w}$, and let $\mathcal{G}(\boldsymbol{y}, \boldsymbol{z})$ be a k-independent polynomial map such that $\mathrm{var}(H) \cap \mathrm{var}(\mathcal{G}) = \varnothing$. Then, for any $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k \in \mathbb{F}^n$:*

$$\frac{\partial^k f}{\partial \boldsymbol{v}_1 \partial \boldsymbol{v}_2 \cdots \partial \boldsymbol{v}_k} \circ H \neq 0 \quad \Rightarrow \quad f \circ (\mathcal{G} + H) \neq 0 .$$

Proof. By definition of k-independent polynomial maps, $\mathcal{G} = \mathcal{G}_1(\boldsymbol{y_1}, z_1) + \ldots + \mathcal{G}_k(\boldsymbol{y_k}, z_k)$ for some variable-disjoint 1-independent polynomial maps $\mathcal{G}_1, \ldots, \mathcal{G}_k$. It is therefore enough to prove the lemma for $k = 1$, as

we can replace f with $\frac{\partial^{k-1} f}{\partial \boldsymbol{v}_2 \cdots \partial \boldsymbol{v}_k}$, H with $H + \mathcal{G}_2 + \ldots + \mathcal{G}_k$ and $\mathcal{G}$ with $\mathcal{G}_1$; by iterative application of the result for $k = 1$, we will get the general result for an arbitrary $k \in \mathbb{N}$.

Denote $H = (H_1, H_2, \ldots, H_n)$. By Definition 3.6, the condition $\frac{\partial f}{\partial \boldsymbol{v}} \circ H \neq 0$ implies that there exists some $i \in [n]$ such that $\frac{\partial f}{\partial x_i} \circ H \neq 0$. Assume, WLOG, $\frac{\partial f}{\partial x_1} \circ H \neq 0$. As $\mathcal{G}$ is a 1-independent polynomial map, there exists some $\boldsymbol{\alpha} \in \mathbb{F}^{|\boldsymbol{y_1}|}$ such that $f \circ (\mathcal{G} + H)\big|_{\boldsymbol{y_1} = \boldsymbol{\alpha}} = f(z_1 + H_1, H_2, \ldots, H_n)$; denote $g \triangleq f \circ (\mathcal{G} + H)\big|_{\boldsymbol{y_1} = \boldsymbol{\alpha}}$. As no coordinate of H depends on z_1:

$$\frac{\partial g}{\partial z_1} = \frac{\partial (z_1 + H_1)}{\partial z_1} \cdot \frac{\partial f}{\partial x_1}(z_1 + H_1, H_2, \ldots, H_n) = 1 \cdot \left(\frac{\partial f}{\partial x_1}\right)(z_1 + H_1, H_2, \ldots, H_n)$$

and therefore:

$$\left.\frac{\partial g}{\partial z_1}\right|_{z_1 = 0} = 1 \cdot \left(\frac{\partial f}{\partial x_1}\right)(0 + H_1, H_2, \ldots, H_n) = \left(\frac{\partial f}{\partial x_1}\right) \circ H \neq 0 \,.$$

As g is a projection of $f \circ (\mathcal{G} + H)$, it follows that $f \circ (\mathcal{G} + H) \neq 0$. □

The next lemma shows how to use k-independent maps in order to project a polynomial to a subset of its coordinates.

Lemma 3.10. *Let $m \leq n \in \mathbb{N}$ and $g(\boldsymbol{w}) \in \mathbb{F}[w_1, \ldots, w_m]$. Let $f(\boldsymbol{x}) = g(\ell_1(\boldsymbol{x}), \ldots, \ell_m(\boldsymbol{x}))$ for linearly independent linear functions $\ell_1(\boldsymbol{x}), \ldots, \ell_m(\boldsymbol{x})$. Let $\mathcal{G}(\boldsymbol{y}, \boldsymbol{z})$ be a k-independent polynomial map. For a set $S \subseteq [n]$ of size k denote by $\tilde{g}(x_i : i \in [m] \setminus S) = g\big|_{S=0}$ the projection of g to the variables outside of S. Then, there exist linearly independent linear functions $\{\tilde{\ell}_i(\boldsymbol{x}) : i \in [m] \setminus S\}$, additional linear functions $\boldsymbol{L}(\boldsymbol{x}) = (L_1(\boldsymbol{x}), \ldots, L_k(\boldsymbol{x}))$ and an assignment $\boldsymbol{\alpha} \in \mathbb{F}^{|\boldsymbol{y}|}$ such that:*

$$f(\boldsymbol{x} + \mathcal{G}(\boldsymbol{\alpha}, \boldsymbol{L}(\boldsymbol{x}))) = \tilde{g}(\tilde{\ell}_i(\boldsymbol{x}) : i \in [m] \setminus S) \,.$$

Proof. It is enough to prove the lemma for the case $k = 1$, as we may then define $\tilde{f}(\boldsymbol{x}) \triangleq f(\boldsymbol{x} + \mathcal{G}(\boldsymbol{\alpha}, L_1(\boldsymbol{x}))) = \tilde{g}(\tilde{\ell}_1(\boldsymbol{x}), \ldots, \tilde{\ell}_{m-1}(\boldsymbol{x}))$ and apply the result iteratively. Thus, assume $k = 1$, and WLOG assume $S = \{x_1\}$ (thus, $\tilde{g}(w_2, \ldots, w_m) = g(0, w_2, \ldots, w_m)$).

Let x_i be some variable with a non-zero coefficient in $\ell_1(\boldsymbol{x})$. Such a variable exists as the ℓ_js are linearly independent. For $j \in [m]$, denote $\beta_j = \frac{\partial \ell_j}{\partial x_i}$, i.e. β_j is the coefficient of x_i in ℓ_j. By our choice of i, $\beta_1 \neq 0$. Choose some $\boldsymbol{\alpha} \in \mathbb{F}^{|\boldsymbol{y}|}$ such that $\mathcal{G}(\boldsymbol{\alpha}, z_1)$ has z_1 in the ith coordinate, and 0 in all other coordinates. Define $L(\boldsymbol{x}) \triangleq -\frac{\ell_1(\boldsymbol{x})}{\beta_1}$, so we get:

$$f(\boldsymbol{x} + \mathcal{G}(\boldsymbol{\alpha}, L(\boldsymbol{x})) = f\left(x_1, x_2, \ldots, x_{i-1}, x_i - \frac{\ell_1(\boldsymbol{x})}{\beta_1}, x_{i+1}, \ldots, x_n\right).$$

Observe that for every i,

$$\ell_i\left(\boldsymbol{x} + \mathcal{G}(\boldsymbol{\alpha}, L(\boldsymbol{x})\right) = \ell_i\left(x_1, x_2, \ldots, x_{i-1}, x_i - \frac{\ell_1(\boldsymbol{x})}{\beta_1}, x_{i+1}, \ldots, x_n\right) = \ell_i(\boldsymbol{x}) - \frac{\beta_i}{\beta_1} \cdot \ell_1(\boldsymbol{x}) \,.$$

In particular, $\ell_1\left(\boldsymbol{x} + \mathcal{G}(\boldsymbol{\alpha}, L(\boldsymbol{x})\right) = 0$. For $i = 2, \ldots, m$, define:

$$\tilde{\ell}_i(\boldsymbol{x}) \triangleq \ell_i(\boldsymbol{x}) - \frac{\beta_i}{\beta_1} \cdot \ell_1(\boldsymbol{x}) \,.$$

As $\ell_1, \ldots, \ell_m$ are linearly independent, it follows that $\tilde{\ell}_2, \ldots, \tilde{\ell}_m$ are also linearly independent. We get that

$$f(\boldsymbol{x} + \mathcal{G}(\boldsymbol{\alpha}, L(\boldsymbol{x}))) = g(0, \tilde{\ell}_2(\boldsymbol{x}), \ldots, \tilde{\ell}_m(\boldsymbol{x})) = \tilde{g}(\tilde{\ell}_2(\boldsymbol{x}), \ldots, \tilde{\ell}_m(\boldsymbol{x})) \,. \qquad \square$$

3.1 Proof of Theorem 1.37

We next prove that there are k-independent maps that are provably not robust. The proof is by giving a different construction of such maps that, for an appropriate arrangement of the n variables in a matrix, is guaranteed to output matrices of rank at most k. Thus, a determinant of any $(k+1) \times (k+1)$ minor, a polynomial that has small formulas for small values of k, vanishes on the output of any such map.

The fact that such a construction exists was already noticed in [FSTW16] (Construction 6.3 of the full version of the paper). For completeness we repeat the construction here.

Proof. (of Theorem 1.37) Fix the number of variables n and assume WLOG n is a perfect square, i.e., $n = m^2$. We index the variables as $x_{i,j}$ for $i, j \in [m]$. We let $f = \mathrm{Det}_{t+1}$. By [GKKS16], over fields of characteristic zero, f has a $t^{O(\sqrt{t})} = O(n)$ sized $\Sigma\Pi\Sigma$ formula, which is polynomial in n for $t = O\left((\log n / \log\log n)^2\right)$. Over fields of positive characteristic the formula size is quasipolynomial in t, and the $\Sigma\Pi\Sigma$ complexity is at most $t!$, which is polynomial in n for $t = O(\log n / \log\log n)$.

Denote by $\boldsymbol{M}$ the $(t+1) \times (t+1)$ symbolic matrix of variables $\boldsymbol{M}_{i,j} = x_{i,j}$. We first construct a uniform 1-independent polynomial map $\mathcal{G}_1$ such that $\boldsymbol{M} \circ \mathcal{G}_1$ is of rank 1, and define $\mathcal{G}$ to be a sum of t variable-disjoint copies of $\mathcal{G}_1$. As $rank(\boldsymbol{M} \circ \mathcal{G}_1) = 1$, we have $rank(\boldsymbol{M} \circ \mathcal{G}) \leq t$ so $\mathrm{Det}_{t+1}(\boldsymbol{M} \circ \mathcal{G}) = 0$, as required. We now focus on $\mathcal{G}_1$.

Fix n distinct field elements $\{\alpha_{i,j}\}_{i,j=1}^{m} \subseteq \mathbb{F}$ and let w, y, z be new variables. Define two vectors of polynomials of degree $n-1$, $R = (R_1, \ldots, R_m), C = (C_1, \ldots, C_m) \in \mathbb{F}[y]^m$, such that for every $k \in [m]$ R_k and C_k satisfy

$$R_k(\alpha_{i,j}) = \delta_{i,k} \quad \text{and} \quad C_k(\alpha_{i,j}) = \delta_{j,k}.$$

Define $\mathcal{G}_1(w, y, z)$ as the $m \times m$ matrix $z \cdot (w^{2n-2} R(\frac{y}{w}) \cdot C(\frac{y}{w})^T)$ (the (i,j) entry of $\mathcal{G}_1$ is $z \cdot w^{2n-2} \cdot R_i(\frac{y}{w}) \cdot C_j(\frac{y}{w})$). As every coordinate of $\mathcal{G}_1$ is a homogeneous polynomial of degree $2n-1$, $\mathcal{G}_1$ is a uniform polynomial map. For any $i, j \in [m]$ we have that

$$\mathcal{G}_1(1, \alpha_{i,j}, z) = z \cdot (R_{i'}(\alpha_{i,j}) \cdot C_{j'}(\alpha_{i,j}))_{i',j' \in [m]} = z \cdot (\delta_{i,i'} \delta_{j,j'})_{i',j' \in [m]} \,.$$

The above matrix has z in entry (i,j) and 0 everywhere else, so $\mathcal{G}_1$ is a uniform 1-independent polynomial map. The resulting matrix $\boldsymbol{M} \circ \mathcal{G}_1$ is of rank 1 since it is a product of vectors $R \cdot C^T$, so the variable-disjoint sum $\mathcal{G} = \sum_1^t \mathcal{G}_1(w_i, y_i, z_i)$ is a uniform t-independent polynomial map satisfying $f \circ \mathcal{G} = 0$. $\square$

4 Interpolation and reconstruction for orbits of the continuant polynomial

We start by proving that any uniform 1-independent map hits $\mathrm{C}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$ (Theorem 1.17).

Proof of Theorem 1.17. Let $f(x_1, \ldots, x_n) = \mathrm{C}_m(\ell_1(\boldsymbol{x}) + b_1, \ldots, \ell_m(\boldsymbol{x}) + b_m)$, where the ℓ_is are linear forms. Observe that $\mathrm{C}_m(y_1, \ldots, y_m)$ is a multilinear polynomial that has a unique monomial of degree m and all

other monomials are of smaller degree. Thus,

$$\mathrm{C}_m(y_1,\ldots,y_m) = \prod_{i=1}^{m} y_i + \tilde{\mathrm{C}}_{m-1}(y_1,\ldots,y_m)\ ,$$

where $\deg\left(\tilde{\mathrm{C}}_{m-1}\right) \le m-1$. Hence,

$$f(\boldsymbol{x}) = \mathrm{C}_m\left(\ell_1(\boldsymbol{x})+b_1,\ldots,\ell_m(\boldsymbol{x})+b_m\right) = \prod_{i=1}^{m}\ell_i + \tilde{f}\left(\ell_1,\ldots,\ell_m\right)\ ,$$

where $\deg\left(\tilde{f}\right) \le m-1$.

Let $\mathcal{G}_1$ be a uniform 1-independent polynomial map into $\mathbb{F}^n$. Let d be the degree of the different components of $\mathcal{G}_1$. Observation 3.1(2) implies that $\left(\prod_{i=1}^{m}\ell_i\right)\circ\mathcal{G}_1 \neq 0$ and hence it is a nonzero homogeneous polynomial of degree $m\cdot d$. As $\deg\left(\tilde{f}\circ\mathcal{G}_1\right) \le (m-1)\cdot d < \deg\left(\left(\prod_{i=1}^{m}\ell_i\right)\circ\mathcal{G}_1\right)$, we have that

$$f\circ\mathcal{G}_1 = \left(\prod_{i=1}^{m}\ell_i\right)\circ\mathcal{G}_1 + \tilde{f}\circ\mathcal{G}_1 \neq 0$$

and the claim follows. □

Corollary 1.18 follows immediately from Theorem 1.17, Observation 1.14 and the construction of a uniform generator in Definition 3.4.

Remark 4.1. *A similar argument would show that* $\mathcal{G}(y,z) \triangleq \left(y^{n-1}, y^{n-2}z, \ldots, z^{n-1}\right)$ *is a hitting set generator for* $C_m^{GL_n^{aff}(\mathbb{F})}$*, which leads to a hitting set of size* n^4*.*

We now turn to giving a reconstruction algorithm for $\mathrm{C}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$. We start by proving some simple lemmas that will be used for constructing an interpolating set.

Definition 4.2. *We call an ordered triplet* $(i,j,k) \in \mathbb{Z}_m^3$ *a* consecutive triplet *if* $j = i+1$ *and* $k = i+2$*, or* $j = k+1$ *and* $i = k+2$*, where all equalities are taken modulo* m*.*

Lemma 4.3. *Let* $m \ge 3$*. Then* (i,j,k) *is a consecutive triplet if and only if every monomial in* $C_m(x_0,\ldots,x_{m-1})$ *that contains both* x_i *and* x_k*, also contains* x_j*.*

Proof. Observe that a polynomial $f(\boldsymbol{x})$ has a monomial containing x_i and x_k but not x_j, if and only if this is also the case when we set $x_j = 0$. Assume that (i,j,k) is a consecutive triplet. Then,

$$\begin{aligned}\mathrm{C}_m(x_0,\ldots,x_i,0,x_{i+2},\ldots,x_{m-1}) &= \mathrm{Trace}\left(\begin{pmatrix}x_0 & 1\\ 1 & 0\end{pmatrix}\cdot\ldots\cdot\begin{pmatrix}x_i & 1\\ 1 & 0\end{pmatrix}\cdot\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}\cdot\begin{pmatrix}x_{i+2} & 1\\ 1 & 0\end{pmatrix}\cdot\ldots\cdot\begin{pmatrix}x_{m-1} & 1\\ 1 & 0\end{pmatrix}\right)\\ &= \mathrm{Trace}\left(\begin{pmatrix}x_0 & 1\\ 1 & 0\end{pmatrix}\cdot\ldots\cdot\begin{pmatrix}x_{i-1} & 1\\ 1 & 0\end{pmatrix}\cdot\begin{pmatrix}x_i+x_{i+2} & 1\\ 1 & 0\end{pmatrix}\cdot\begin{pmatrix}x_{i+3} & 1\\ 1 & 0\end{pmatrix}\cdot\ldots\cdot\begin{pmatrix}x_{m-1} & 1\\ 1 & 0\end{pmatrix}\right).\end{aligned}$$

It immediately follows that no monomial of $\mathrm{C}_m(x_0,\ldots,x_i,0,x_{i+2},\ldots,x_{m-1})$ contains both x_i and x_{i+2}.

We now prove the second direction in the claim. Since C_m is a trace of a matrix product, by properties of trace we can assume WLOG that $i<j<k$, by first rotating the order of the matrices until we have $i<j<k$ or $k<j<i$ (where $a<b$ means that the matrix corresponding to a comes before that of b). As both cases

are equivalent we can assume that $i < j < k$. We next handle this case. Assume WLOG that $j - i > 1$. Set $x_r = 0$ for every $i + 2 \le r < k$, to 0. We get that the new polynomial has the form

$$\mathrm{Trace}\left(\begin{pmatrix} x_0 & 1 \\ 1 & 0 \end{pmatrix} \cdot \ldots \cdot \begin{pmatrix} x_i & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} x_{i+1} & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}^{k-i-2} \cdot \begin{pmatrix} x_k & 1 \\ 1 & 0 \end{pmatrix} \cdot \ldots \begin{pmatrix} x_{m-1} & 1 \\ 1 & 0 \end{pmatrix}\right)$$
$$= \begin{cases} \mathrm{C}_{m-k+i+2}\left(x_0, \ldots, x_i, x_{i+1}, x_k, \ldots, x_{m-1}\right), & \text{for } k-i \text{ even} \\ \mathrm{C}_{m-k+i+1}\left(x_0, \ldots x_{i-1}, x_i, x_{i+1} + x_k, \ldots, x_{m-1}\right), & \text{for } k-i \text{ odd} \end{cases},$$

and a monomial of maximal degree in this polynomial contains both x_i and x_k (when $k - i$ is even there is a unique monomial of maximal degree, and when $k - i$ is odd there are two such monomials). □

Corollary 4.4. *Let $m \ge 3$. Then (i, j, k) is a consecutive triplet if and only if $\frac{\partial^2 C_m}{\partial x_i \partial x_k}\Big|_{x_j=0} = 0$.*

For every list of three distinct indices $(i, j, k) \in [m]_0^3$ denote

$$\mathrm{C}_m^{(i,j,k)}(\boldsymbol{x}) \triangleq \left.\frac{\partial^2 \mathrm{C}_m}{\partial x_i \partial x_k}\right|_{x_j=0}.$$

Lemma 4.5. *Let $n \ge m \ge 3$ and t be integers. Assume $H(\boldsymbol{w}) : \mathbb{F}^t \to \mathbb{F}^n$ is a hitting-set generator for $C_m^{(i,j,k)^{GL_n^{aff}(\mathbb{F})}}$, for every list of three distinct indices $(i, j, k) \in [m]_0^3$. Let $\mathcal{G}_3(\boldsymbol{y}, \boldsymbol{z})$ be a 3-independent polynomial map (into $\mathbb{F}^n$) that each of its coordinates is a homogeneous linear function in $\boldsymbol{z}$, over $\mathbb{F}(\boldsymbol{y})$ (for example, $\mathcal{G}_k^{SV}$ has this property, for every k). Then, for every m_1, m_2 and n and every two polynomials $f_1 \in C_{m_1}^{GL_n^{aff}(\mathbb{F})}$ and $f_2 \in C_{m_2}^{GL_n^{aff}(\mathbb{F})}$ it holds that $f_1 = f_2$ if and only if $f_1 \circ (H + \mathcal{G}_3) = f_2 \circ (H + \mathcal{G}_3)$.*

Roughly, what the lemma claims is that if $\mathcal{G}_3$ is a 3-independent map and H hits $\mathrm{C}_m^{(i,j,k)^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})}}$, then $H + \mathcal{G}_3$ is an interpolating-set generator.

Proof. Denote $f_1 = \mathrm{C}_{m_1}(\ell_{1,0}, \ldots, \ell_{1,m_1-1})$ and $f_2 = \mathrm{C}_{m_2}(\ell_{2,0}, \ldots, \ell_{2,m_2-1})$. The proof has three steps. We first prove that if $f_1 \circ (H + \mathcal{G}_3) = f_2 \circ (H + \mathcal{G}_3)$ then $m_1 = m_2$ and there exists a permutation $\pi : [m]_0 \to [m]_0$, and constants α_j, such that for every j it holds that $\ell_{1,j} = \alpha_j \cdot \ell_{2,\pi(j)}$. We then show that, possibly after rotating the order and taking a transpose, we can assume WLOG that π is the identity permutation. At the last step we prove that either $\alpha_j = 1$ for every j, or that m is even, $\alpha_0 \cdot \alpha_1 = 1$ and for every j, $\alpha_{2j} = \alpha_0$ and $\alpha_{2j+1} = \alpha_1$.

Step 1: As in the proof of Theorem 1.17, $\deg(f_i) = m_i$ and the homogeneous part of degree m_i in f_i is given by

$$f_i^{[m_i]} = \prod_{j=0}^{m_i-1} \ell_{i,j}^{[1]}.$$

Observe that since $f_i^{[m_i]} \circ (H + \mathcal{G}_3)$ is nonzero (e.g. by Observation 3.1(2)), and its degree, as a polynomial in $\boldsymbol{z}$, is exactly m_i (and every other term in $f_i \circ (H + \mathcal{G}_3)$ has degree strictly smaller as a polynomial in $\boldsymbol{z}$), it must hold that $m_1 = m_2$. To simplify the notation let $m = m_1 = m_2$. Again by comparing terms of maximal degree in $\boldsymbol{z}$ we see that

$$\left(\prod_{j=0}^{m-1} \ell_{1,j}^{[1]}\right) \circ \mathcal{G}_3 = \left(\prod_{j=0}^{m-1} \ell_{2,j}^{[1]}\right) \circ \mathcal{G}_3. \tag{10}$$

As both $\{\ell_{1,i}\}$ and $\{\ell_{2,i}\}$ are linearly independent sets, we get from unique factorization and from Observation 3.1(3), that there exists a permutation $\pi : [m]_0 \to [m]_0$ and constants $\{\alpha_j\}$ so that $\ell_{1,j} = \alpha_j \ell_{2,\pi(j)}$, for every j. This completes the first step.

Step 2: We wish to show that the permutation π is an "ordered" cycle of length m. That is, that it either has the form $(i, i+1, \ldots, m-1, 0, \ldots, i-1)$, or $(i, i-1, \ldots, 0, m-1, \ldots, i+1)$, for some i. Indeed, assume for a contradiction that this is not the case. Then, there must be an index i such that $(\pi(i), \pi(i+1), \pi(i+2))$ is not a consecutive triplet. Let $\{\boldsymbol{v}_j\}_j$ be a dual set to $\{\ell_{2,j}\}_j$. Corollary 4.4 and Lemma 3.8 imply that

$$\left.\frac{\partial^2 f_1}{\partial \boldsymbol{v}_i \partial \boldsymbol{v}_{i+2}}\right|_{\ell_{2,i+1}(\boldsymbol{x})=0} = 0 \quad \text{and} \quad \left.\frac{\partial^2 f_2}{\partial \boldsymbol{v}_i \partial \boldsymbol{v}_{i+2}}\right|_{\ell_{2,i+1}(\boldsymbol{x})=0} \neq 0 \,.$$

In particular

$$-\mathrm{C}_m^{(\pi(i),\pi(i+1),\pi(i+2))}(\ell_{2,0}, \ldots, \ell_{2,m-1}) = \left.\frac{\partial^2 (f_1 - f_2)}{\partial \boldsymbol{v}_i \partial \boldsymbol{v}_{i+2}}\right|_{\ell_{2,i+1}(\boldsymbol{x})=0} \neq 0 \,.$$

By the assumption on H we get that

$$-\mathrm{C}_m^{(\pi(i),\pi(i+1),\pi(i+2))} \circ H = \left(\left.\frac{\partial^2 (f_1 - f_2)}{\partial \boldsymbol{v}_i \partial \boldsymbol{v}_{i+2}}\right|_{\ell_{2,i+1}(\boldsymbol{x})=0}\right) \circ H = -\left(\left.\frac{\partial^2 f_2}{\partial \boldsymbol{v}_i \partial \boldsymbol{v}_{i+2}}\right|_{\ell_{2,i+1}(\boldsymbol{x})=0}\right) \circ H \neq 0 \,.$$

Applying Lemma 3.9 for $k = 2$ and Lemma 3.10 for $k = 1$ we get that $(f_1 - f_2) \circ (H + \mathcal{G}_3) \neq 0$, in contradiction.

Step 3: To simplify notation, assume, WLOG, that π is the identity permutation. Observe that $\prod_{i=0}^{m-1} \ell_{1,i}^{[1]} \circ \mathcal{G}_3 = \prod_{i=0}^{m-1} \alpha_i \cdot \prod_{i=0}^{m-1} \ell_{2,i}^{[1]} \circ \mathcal{G}_3$. Hence, Equation (10) implies that $\prod_{i=0}^{m-1} \alpha_i = 1$. If there is i such that $\alpha_i \cdot \alpha_{i+1} \neq 1$ then use $\mathcal{G}_3$ to restrict to the subspace $\ell_{1,i} = \ell_{1,i+1} = 0$ (using Lemma 3.10). Denote with $\mathcal{G}_3'$, the map $\mathcal{G}_3$ after we used two of the z_is for the restriction ($\mathcal{G}_3'$ is a 1-independent map). As $\mathrm{C}_m(x_0, \ldots, x_{i-1}, 0, 0, x_{i+2}, \ldots, x_{m-1}) = \mathrm{C}_{m-2}(x_0, \ldots, x_{i-1}, x_{i+2}, \ldots, x_{m-1})$, we get a contradiction by considering the terms of maximal degrees (as polynomials in the remaining z) in $f_1 \circ \mathcal{G}_3$ and $f_2 \circ \mathcal{G}_3$ as follows:

$$\begin{aligned}
\left(\prod_{j \in [m]_0 \setminus \{i,i+1\}} \ell_{1,j}^{[1]}\right) \circ \mathcal{G}_3' &= \left(\prod_{j \in [m]_0 \setminus \{i,i+1\}} \ell_{2,j}^{[1]}\right) \circ \mathcal{G}_3' = \left(\prod_{j \in [m]_0 \setminus \{i,i+1\}} \alpha_j \cdot \ell_{1,j}^{[1]}\right) \circ \mathcal{G}_3' \\
&= \left(\prod_{j \in [m]_0 \setminus \{i,i+1\}} \alpha_j\right) \cdot \left(\prod_{j \in [m]_0 \setminus \{i,i+1\}} \cdot \ell_{1,j}^{[1]}\right) \circ \mathcal{G}_3' \\
&= \frac{1}{\alpha_i \cdot \alpha_{i+1}} \cdot \left(\prod_{j \in [m]_0 \setminus \{i,i+1\}} \cdot \ell_{1,j}^{[1]}\right) \circ \mathcal{G}_3' \neq \left(\prod_{j \in [m]_0 \setminus \{i,i+1\}} \ell_{1,j}^{[1]}\right) \circ \mathcal{G}_3' \,,
\end{aligned}$$

where the first equality follows from the assumption that $f_1 \circ \mathcal{G}_3 = f_2 \circ \mathcal{G}_3$ and the last inequality uses the assumption $\alpha_i \cdot \alpha_{i+1} \neq 1$. Consequently, either for every i, $\alpha_i = 1$, which means that $f_1 = f_2$, as we wanted to prove, or m is even and for every i, $\alpha_{2,i} = \alpha_0$ and $\alpha_{2,i+1} = \alpha_1$, and that $\alpha_0 \cdot \alpha_1 = 1$. We next show that in this case as well the polynomials are equal. Indeed, observe that $\begin{pmatrix} 1 & 0 \\ 0 & \alpha_0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & \alpha_1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$. Hence,

$$\begin{aligned}
f_1 &= \mathrm{Trace}\left(\begin{pmatrix} \ell_{1,0} & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} \ell_{1,1} & 1 \\ 1 & 0 \end{pmatrix} \cdot \ldots \cdot \begin{pmatrix} \ell_{1,m-1} & 1 \\ 1 & 0 \end{pmatrix}\right) \\
&= \mathrm{Trace}\left(\begin{pmatrix} 1 & 0 \\ 0 & \alpha_0 \end{pmatrix} \cdot \begin{pmatrix} \ell_{1,0} & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & \alpha_0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & \alpha_1 \end{pmatrix} \cdot \begin{pmatrix} \ell_{1,1} & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & \alpha_1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & \alpha_0 \end{pmatrix}\right)
\end{aligned}$$

$$\cdot \begin{pmatrix} \ell_{1,2} & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & \alpha_0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & \alpha_1 \end{pmatrix} \cdot \ldots \cdot \begin{pmatrix} 1 & 0 \\ 0 & \alpha_0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & \alpha_1 \end{pmatrix} \cdot \begin{pmatrix} \ell_{1,m-1} & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & \alpha_1 \end{pmatrix} \Bigg)$$

$$= \text{Trace}\left(\begin{pmatrix} \ell_{1,0} & \alpha_0 \\ \alpha_0 & 0 \end{pmatrix} \cdot \begin{pmatrix} \ell_{1,1} & \alpha_1 \\ \alpha_1 & 0 \end{pmatrix} \cdot \ldots \cdot \begin{pmatrix} \ell_{1,m-1} & \alpha_1 \\ \alpha_1 & 0 \end{pmatrix} \right) \tag{11}$$

$$= \text{Trace}\left(\begin{pmatrix} \alpha_0 \cdot \ell_{2,0} & \alpha_0 \\ \alpha_0 & 0 \end{pmatrix} \cdot \begin{pmatrix} \alpha_1 \cdot \ell_{2,1} & \alpha_1 \\ \alpha_1 & 0 \end{pmatrix} \cdot \ldots \cdot \begin{pmatrix} \alpha_1 \cdot \ell_{2,m-1} & \alpha_1 \\ \alpha_1 & 0 \end{pmatrix} \right)$$

$$= (\alpha_0 \cdot \alpha_1)^{m/2} \cdot \text{Trace}\left(\begin{pmatrix} \ell_{2,0} & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} \ell_{2,1} & 1 \\ 1 & 0 \end{pmatrix} \cdot \ldots \cdot \begin{pmatrix} \ell_{2,m-1} & 1 \\ 1 & 0 \end{pmatrix} \right) = 1 \cdot f_2 \ .$$

This concludes the proof of the lemma. □

From Lemma 4.5 we see that all that we have to do in order to construct an interpolating set for $\mathrm{C}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$, is to find a map H as in the statement of the lemma.

Lemma 4.6. *Let $n \geq m$ be integers. Let $\mathcal{G}_2(\boldsymbol{y}, \boldsymbol{z})$ be a 2-independent polynomial map into $\mathbb{F}^n$, that is linear in $\boldsymbol{z}$. Then, For every list of three distinct indices $(i,j,k) \in [m]_0^3$ and for every m n-variate linearly independent linear functions $\ell_0(\boldsymbol{x}), \ldots, \ell_{m-1}(\boldsymbol{x}) \in \mathbb{F}[\boldsymbol{x}]$ it holds that if $C_m^{(i,j,k)}(\ell_0, \ldots, \ell_{m-1}) \neq 0$ then $C_m^{(i,j,k)}(\ell_0, \ldots, \ell_{m-1}) \circ \mathcal{G}_2 \neq 0$.*

Proof. As $\mathrm{C}_m^{(i,j,k)}(\ell_0, \ldots, \ell_{m-1}) \neq 0$ it follows that (i,j,k) is not a consecutive triplet. Assume WLOG that $i < j-1 < j < k$. Use $\mathcal{G}_2$ to further restrict the polynomial to the subspace $\ell_{j-1} = 0$ (using Lemma 3.10). Let $\mathcal{G}_2'$ denote $\mathcal{G}_2$ after the restriction. Lemma 3.10 guarantees that $\mathcal{G}_2'$ is 1-independent. Observe that the homogeneous term of maximal degree in $\mathrm{C}_m^{(i,j,k)}(\ell_0, \ldots, \ell_{m-1})\Big|_{\ell_{j-1}(\boldsymbol{x})=0}$ is equal to $\prod_{t \in [m]_0 \setminus \{i,j-1,j,k\}} \ell_t^{[1]}$. It follows that the term of maximal degree, as a polynomial in $\boldsymbol{z}$, in $\mathrm{C}_m^{(i,j,k)}(\ell_0, \ldots, \ell_{m-1})\Big|_{\ell_{j-1}(\boldsymbol{x})=0} \circ \mathcal{G}_2'$ is $\left(\prod_{t \in [m]_0 \setminus \{i,j-1,j,k\}} \ell_t^{[1]}\right) \circ \mathcal{G}_2'$, which is nonzero by Observation 3.1(2). □

Combining Lemmas 4.5 and 4.6 we get the following corollary:

Corollary 4.7. *Let $\mathcal{G}_5(\boldsymbol{y}, \boldsymbol{z}) : \mathbb{F}^t \to \mathbb{F}^n$ be a 5-independent polynomial map that is linear in $\boldsymbol{z}$. Then, for every $m_1, m_2 \leq n$ and every two polynomials $f_1 \in C_{m_1}^{GL_n^{aff}(\mathbb{F})}$ and $f_2 \in C_{m_2}^{GL_n^{aff}(\mathbb{F})}$, it holds that $f_1 = f_2$ if and only if $f_1 \circ \mathcal{G}_5 = f_2 \circ \mathcal{G}_5$.*

Theorem 1.19 follows immediately from Corollary 4.7 and Observation 1.14.

4.1 Reconstruction algorithm for $\mathrm{C}^{\mathbf{GL}^{\mathbf{aff}}(\mathbb{F})}$

The reconstruction algorithm is given in Page 27.

Analysis of Algorithm 1:

Claim 4.8. *Step 1 can be executed in polynomial-time.*

Proof. Let $\mathcal{G}_1(y,z)$ be a 1-independent map. Let w be a new variable and consider $\mathcal{G} = w \cdot \mathcal{G}_1$. I.e., we multiply each coordinate of $\mathcal{G}_1$ with w. Observe that the degree of w and of z in $(f \circ \mathcal{G})$ is exactly $\deg(f) = m$. As in the proof of Theorem 1.17, we see that the m-homogeneous component of $(f \circ \mathcal{G})$, when viewed as a

input : Integer n, black-box access to $f = \mathrm{C}_m^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})}$
output: Linear functions $\tilde{\ell}_0, \ldots, \tilde{\ell}_{m-1} \in \mathbb{F}[\boldsymbol{x}]$ such that $f = \mathrm{C}_m\left(\tilde{\ell}_0, \ldots, \tilde{\ell}_{m-1}\right)$

Compute m using interpolation and the hitting set constructed in Theorem 1.17 ;
Factor $f^{[m]}$; /* Using univariate root-finding */
/* We found linear functions $L_0^{[1]}, \ldots, L_{m-1}^{[1]}$, such that for some permutation π and scalars α_i, $\alpha_i \cdot L_i^{[1]} = \ell_{\pi(i)}^{[1]}$ */
Compute a dual set $\{\boldsymbol{v}_i\}_i$ to $\left\{L_i^{[1]}\right\}_i$;
/* Next we compute the free terms */
for $i = 0$ **to** $m-1$ **do**
  Define $f_i'(\boldsymbol{x}) \triangleq \frac{\partial f}{\partial \boldsymbol{v}_i}(\boldsymbol{x})$;
  Set $g_i(\boldsymbol{x}) = f(\boldsymbol{x}) - L_i^{[1]}(\boldsymbol{x}) \cdot f_i'(\boldsymbol{x})$; /* We can simulate queries to g_i */
  Compute $\deg(g_i)$;
  if $\deg(g_i) = m-2$ **then**
    set $\lambda_i = 0$
  else
    Find $\boldsymbol{u} \in \mathbb{F}^n$ such that $f^{[m]}(\boldsymbol{u}) \neq 0$;
    Set $\lambda_i = \left(L_i^{[1]}(\boldsymbol{u}) \cdot {g_i}^{[m-1]}(\boldsymbol{u})\right) / f^{[m]}(\boldsymbol{u})$;
  end
  Set $L_i = L_i^{[1]} + \lambda_i$;
end
/* There is a permutation π and scalars α_i such that $\alpha_i \cdot L_i = \ell_{\pi(i)}$ */
Find all consecutive triplets and recover the permutation π ;
/* WLOG π is the identity permutation */
Find $\{\boldsymbol{u}_i\}$ such that $L_i(\boldsymbol{u}_j) = \delta_{i,j}$;
/* We now recover the α_is */
if *m is odd* **then**
  for $i = 0$ **to** $m-1$ **do**
    Set $\tilde{\ell}_i = f(\boldsymbol{u}_i) \cdot L_i$;
  end
else
  Set $\beta_0 = \alpha_0 = 1$ and $\tilde{\ell}_0 = L_0$;
  for $i = 1$ **to** $m-1$ **do**
    Set $\beta_i = \left(f(\boldsymbol{u}_{i-1} + \boldsymbol{u}_i) - 2\right) / \beta_{i-1}$ and $\tilde{\ell}_i = \beta_i \cdot L_i$;
  end
end
return $\tilde{\ell}_0, \ldots, \tilde{\ell}_{m-1}$;

Algorithm 1: reconstruction algorithm for $\mathrm{C}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$

polynomial in w, is $\left(\prod_{i=0}^{m-1} \ell_i^{[1]}\right) \circ \mathcal{G}_1 \neq 0$. As we know that $m \leq n$, using interpolation (over w) we get black-box access to $(f \circ \mathcal{G})^{[k]}$, for every $0 \leq k \leq n$. We look for the first k, starting from n and going down, such that $(f \circ \mathcal{G})^{[k]} \neq 0$. This can be done, for example, by interpolation (over y, z). □

Claim 4.9. *Step 2 can be done with polynomially many queries to a root-finding algorithm over $\mathbb{F}$ (assuming $|\mathbb{F}| \geq n^3$).*

We assume some knowledge with known factoring algorithms. For good a reference see [vzGG03] (the lecture notes of Madhu Sudan are also a great resource on the subject [Sud99]).

Proof sketch. Observe that $f^{[m]} = \prod_{i=0}^{m-1} \ell_i^{[1]}$, and all its linear factors are linearly independent. Known factoring algorithms require that we reduce the polynomial that we wish to factor to a square-free, bivariate polynomial. This can be easily done using 2-independent maps. Let $\mathcal{G}_2(\boldsymbol{y}, z_1, z_2)$ be a 2-independent map that is a linear form in z_1 and z_2 (e.g., $\mathcal{G}_2^{\mathrm{SV}}$). Observation 3.1(3) shows that composing $f^{[m]}$ with $\mathcal{G}_2(\boldsymbol{y}, \boldsymbol{z})$, keeps all factors linearly independent, when viewed as linear polynomials in $\boldsymbol{z}$. Each assignment to $\boldsymbol{y}$ gives a different polynomial whose factors are homogeneous linear functions in z_1, z_2. Observe that there is an assignment to $\boldsymbol{y}$ from the set $[n^3]^{|\boldsymbol{y}|}$, that maintains the property that the factors are linearly independent. Indeed, for every two factors we need the assignment to be a nonzero of the determinant of the coefficient-matrix of the two factors. There are $\binom{m}{2}$ such determinant, each has degree $2(n-1)$ as a polynomial in $\boldsymbol{y}$ (hence the requirement for a field of size n^3). By going over all such assignments to $\boldsymbol{y}$, we are guaranteed to find one that maintains this property.

Once we reduced to the square-free, bivariate case, factoring algorithms proceed by reducing to factoring of univariate polynomials. In our case the univariate completely splits as a product of linear factors, hence the univariate factorization step only need oracle access to a root-finding algorithm. □

Observe that we have found irreducible linear functions $L_i^{[1]}$, each is a scalar product of some $\ell_{\pi(i)}^{[1]}$, for some permutation π. Let $\{\alpha_i\}$ be such that $\alpha_i \cdot L_i^{[1]} = \ell_{\pi(i)}^{[1]}$.

Claim 4.10. *For every i, the for-loop in Step 6 returns L_i such that $\alpha_i \cdot L_i = \ell_{\pi(i)}$.*

Proof. For $i \in [m]_0$, denote $\mathrm{C}_m(y_0, \ldots, y_{m-1}) = y_{\pi(i)} \cdot F_{i,1}(\boldsymbol{y} \setminus y_{\pi(i)}) + F_{i,0}(\boldsymbol{y} \setminus y_{\pi(i)})$. Observe that $\deg(F_{i,1}) = m-1$ (since it contains the product of all y_j except $y_{\pi(i)}$) and that $\deg(F_{i,0}) = m-2$. Indeed,

$$\begin{aligned}
F_{i,0}(\boldsymbol{y}) &= \mathrm{C}_m(y_0, \ldots, y_{m-1})\big|_{y_{\pi(i)}=0} \\
&= \mathrm{Trace}\left(\begin{pmatrix} y_0 & 1 \\ 1 & 0 \end{pmatrix} \cdot \ldots \cdot \begin{pmatrix} y_{\pi(i)-1} & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} y_{\pi(i)+1} & 1 \\ 1 & 0 \end{pmatrix} \cdot \ldots \cdot \begin{pmatrix} y_{m-1} & 1 \\ 1 & 0 \end{pmatrix}\right) \\
&= \mathrm{Trace}\left(\begin{pmatrix} y_0 & 1 \\ 1 & 0 \end{pmatrix} \cdot \ldots \cdot \begin{pmatrix} y_{\pi(i)-2} & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} y_{\pi(i)-1} + y_{\pi(i)+1} & 1 \\ 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} y_{\pi(i)+2} & 1 \\ 1 & 0 \end{pmatrix} \cdot \ldots \cdot \begin{pmatrix} y_{m-1} & 1 \\ 1 & 0 \end{pmatrix}\right) \\
&= \mathrm{C}_{m-2}\left(y_0, \ldots, y_{\pi(i)-2}, y_{\pi(i)-1} + y_{\pi(i)+1}, y_{\pi(i)+2}, \ldots, y_{m-1}\right) .
\end{aligned}$$

We now note that

$$f_i'(\ell_0, \ldots, \ell_{m-1}) = \frac{\partial \ell_{\pi(i)}}{\partial \boldsymbol{v}_i} \cdot F_{i,1}(\boldsymbol{\ell} \setminus \ell_{\pi(i)}) = \alpha_i \cdot F_{i,1}(\boldsymbol{\ell} \setminus \ell_{\pi(i)}) .$$

As $g_i = f - L_i^{[1]} \cdot f_i'$, we get that

$$g_i = \left(\ell_{\pi(i)} \cdot F_{i,1}(\boldsymbol{\ell} \setminus \ell_{\pi(i)}) + F_{i,0}(\boldsymbol{\ell} \setminus \ell_{\pi(i)})\right) - L_i^{[1]} \cdot \left(\alpha_i \cdot F_{i,1}(\boldsymbol{\ell} \setminus \ell_{\pi(i)})\right)$$

$$= \left(\ell_{\pi(i)} - \alpha_i \cdot L_i^{[1]}\right) \cdot F_{i,1}(\boldsymbol{\ell} \setminus \ell_{\pi(i)}) + F_{i,0}(\boldsymbol{\ell} \setminus \ell_{\pi(i)}) \ .$$

Thus, $\deg(g_i) = m-2$ if and only if $\ell_{\pi(i)} - \alpha_i \cdot L_i^{[1]} = 0$. In other words, $\deg(g_i) = m-2$ if and only if $\ell_{\pi(i)}$ is homogeneous and $L_i = L_i^{[1]}$. As $\ell_{\pi(i)}^{[1]} = \alpha_i \cdot L_i^{[1]}$, it holds that $\left(\ell_{\pi(i)} - \alpha_i \cdot L_i^{[1]}\right) \in \mathbb{F}$. Therefore, if $\deg(g_i) = m-1$ we get that

$$g_i^{[m-1]} = \left(\ell_{\pi(i)} - \alpha_i \cdot L_i^{[1]}\right) \cdot F_{i,1}(\boldsymbol{\ell}^{[1]} \setminus \ell_i^{[1]})^{[m-1]} = \left(\ell_{\pi(i)} - \alpha_i \cdot L_i^{[1]}\right) \cdot \prod_{j \neq \pi(i)} \ell_j^{[1]} \ . \tag{12}$$

Hence,

$$\begin{aligned}\lambda_i &= L_i^{[1]}(\boldsymbol{u}) \cdot g_i^{[m-1]}(\boldsymbol{u}) / f^{[m]}(\boldsymbol{u}) = L_i^{[1]}(\boldsymbol{u}) \cdot \left(\ell_{\pi(i)} - \alpha_i \cdot L_i^{[1]}\right) \cdot \prod_{j \neq \pi(i)} \ell_j^{[1]} / \prod_j \ell_j^{[1]} \\ &= \left(L_i^{[1]}(\boldsymbol{u}) \cdot \left(\ell_{\pi(i)} - \alpha_i \cdot L_i^{[1]}\right)\right) / \ell_{\pi(i)}^{[1]}(\boldsymbol{u}) = \left(\ell_{\pi(i)} - \alpha_i \cdot L_i^{[1]}\right) / \alpha_i \ .\end{aligned}$$

It follows that

$$\alpha_i \cdot L_i = \alpha_i \cdot \left(L_i^{[1]} + \lambda_i\right) = \alpha_i \cdot L_i^{[1]} + \alpha_i \cdot \lambda_i = \alpha_i \cdot L_i^{[1]} + \left(\ell_{\pi(i)} - \alpha_i \cdot L_i^{[1]}\right) = \ell_{\pi(i)}$$

as claimed.

An important point to notice is that we can check whether $\deg(g_i) = m-1$ in the same manner in which we computed $\deg(f)$ (thanks to Equation (12)). □

Note that Step 19 can be executed using Corollary 4.4 and Lemma 4.6. Indeed, as $\ell_{\pi(i)} = \alpha_i L_i$, it follows that $\{\boldsymbol{v}_i/\alpha_i\}$ is a dual set for $\left\{\ell_{\pi(i)}^{[1]}\right\}$. That is, $\ell_{\pi(i)}^{[1]}(\boldsymbol{v}_j/\alpha_j) = \delta_{i,j}$. Therefore, $\frac{\partial^2 f}{\partial(\boldsymbol{v}_i/\alpha_i)\partial(\boldsymbol{v}_k/\alpha_k)}\Big|_{\ell_{\pi(j)}=0} = 0$ if and only if $\frac{\partial^2 f}{\partial \boldsymbol{v}_i \partial \boldsymbol{v}_k}\Big|_{L_j=0} = 0$. Hence, with the help of Lemma 4.6 and interpolation, we can find all consecutive triplets. Once we have that information, construction of π (up to reversal, which does not change the resulting polynomial) is immediate. Since we know π we can assume WLOG that π is the identity permutation.

Step 21 is possible as the L_is are linearly independent. Note that $\ell_{\pi(i)}^{[1]}(\boldsymbol{u}_j) = \delta_{i,j} \cdot \alpha_j$.

Claim 4.11. *The linear functions $\tilde{\ell}_i$ that were computed in Steps 23-31 satisfy $C_m\left(\tilde{\ell}_0, \ldots, \tilde{\ell}_{m-1}\right) = f$.*

Proof. First, observe that $\ell_i(\boldsymbol{u}_j) = \alpha_i \cdot \delta_{i,j}$. Assume first that m is odd. Then

$$\begin{aligned}f(\boldsymbol{u}_i) &= \mathrm{Trace}\left(\begin{pmatrix}\ell_0(\boldsymbol{u}_i) & 1\\ 1 & 0\end{pmatrix} \cdot \begin{pmatrix}\ell_1(\boldsymbol{u}_i) & 1\\ 1 & 0\end{pmatrix} \cdot \ldots \cdot \begin{pmatrix}\ell_{m-1}(\boldsymbol{u}_i) & 1\\ 1 & 0\end{pmatrix}\right) \\ &= \mathrm{Trace}\left(\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}^i \cdot \begin{pmatrix}\alpha_i & 1\\ 1 & 0\end{pmatrix} \cdot \begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}^{m-i-1}\right) = \\ &= \mathrm{Trace}\begin{pmatrix}\alpha_i & 1\\ 1 & 0\end{pmatrix} = \alpha_i \ .\end{aligned}$$

In this case we get that $\tilde{\ell}_i = f(\boldsymbol{u}_i) \cdot L_i = \alpha_i L_i = \ell_i$. In particular, we recovered the original ℓ_is.

Next, assume that m is even. Observe that since m is even we can replace each ℓ_{2i} with ℓ_{2i}/α_0 and each ℓ_{2i+1} with $\ell_{2i+1} \cdot \alpha_0$ and still get the same f (recall Equation (11)). Therefore, we may assume WLOG that $\alpha_0 = 1$.

The first iteration gives

$$f(\boldsymbol{u}_0+\boldsymbol{u}_1) = \mathrm{Trace}\left(\begin{pmatrix}\ell_0(\boldsymbol{u}_0+\boldsymbol{u}_1) & 1\\ 1 & 0\end{pmatrix}\cdot\begin{pmatrix}\ell_1(\boldsymbol{u}_0+\boldsymbol{u}_1) & 1\\ 1 & 0\end{pmatrix}\cdot\ldots\cdot\begin{pmatrix}\ell_{m-1}(\boldsymbol{u}_0+\boldsymbol{u}_1) & 1\\ 1 & 0\end{pmatrix}\right)$$
$$= \mathrm{Trace}\left(\begin{pmatrix}1 & 1\\ 1 & 0\end{pmatrix}\cdot\begin{pmatrix}\alpha_1 & 1\\ 1 & 0\end{pmatrix}\cdot\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}^{m-2}\right) = \mathrm{Trace}\begin{pmatrix}\alpha_1+1 & 1\\ \alpha_1 & 1\end{pmatrix} = \alpha_1+2\,.$$

Hence, $\beta_1 = (f(\boldsymbol{u}_0+\boldsymbol{u}_1)-2)/\alpha_0 = \alpha_1/1 = \alpha_1$, and therefore, $\tilde{\ell}_1 = \ell_1$. We proceed to show by induction that for every i, $\beta_i = \alpha_i$.

$$f(\boldsymbol{u}_i+\boldsymbol{u}_{i+1}) = \mathrm{Trace}\left(\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}\cdot\ldots\cdot\begin{pmatrix}\ell_i(\boldsymbol{u}_i+\boldsymbol{u}_{i+1}) & 1\\ 1 & 0\end{pmatrix}\cdot\begin{pmatrix}\ell_{i+1}(\boldsymbol{u}_i+\boldsymbol{u}_{i+1}) & 1\\ 1 & 0\end{pmatrix}\cdot\ldots\cdot\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}\right)$$
$$= \mathrm{Trace}\left(\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}^{i}\cdot\begin{pmatrix}\alpha_i & 1\\ 1 & 0\end{pmatrix}\cdot\begin{pmatrix}\alpha_{i+1} & 1\\ 1 & 0\end{pmatrix}\cdot\begin{pmatrix}0 & 1\\ 1 & 0\end{pmatrix}^{m-i-2}\right)$$
$$= \mathrm{Trace}\begin{pmatrix}\alpha_i\cdot\alpha_{i+1}+1 & \alpha_i\\ \alpha_{i+1} & 1\end{pmatrix} = \alpha_i\cdot\alpha_{i+1}+2\,,$$

and we conclude, from the induction hypothesis, that $\beta_{i+1} = \alpha_{i+1}$ and that $\tilde{\ell}_{i+1} = \ell_{i+1}$. □

Thus, algorithm 1 correctly outputs linear functions $\{\tilde{\ell}_i\}$ so that $\mathrm{C}_m(\tilde{\ell}_0,\ldots,\tilde{\ell}_{m-1}) = f$.
The claim regarding the running time is also obvious given the analysis above. We thus see that Theorem 1.20 holds.

Remark 4.12. *As Theorem 1.37 shows that t-independent maps do not necessarily lead to robust hitting sets, our reconstruction algorithm is not continuous at* $\mathbf{0}$ *(recall the discussion in section 1.6): Intuitively, around* $\mathbf{0}$*, there is no way to break the tie between the different polynomials* $C_m^{(j,i,k)}(\boldsymbol{x})$ *and decide which are the consecutive triplets.*

5 Orbits of read-once formulas

In this section we discuss the circuit classes $\mathrm{ANF}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$ and $\mathrm{ROF}^{\mathrm{GL}(\mathbb{F})}$ (see Definitions 5.1 and 5.3 below), which are dense in $\mathrm{VP_e}$. We construct a hitting set for $\mathrm{ROF}^{\mathrm{GL}(\mathbb{F})}$ and an interpolating set for $\mathrm{ANF}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$. Finally we observe that the randomized reconstruction algorithm of [GKQ14] works for every polynomial in $\mathrm{ANF}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{C})}$.

We start with basic definitions concerning ROFs and ROANFs and prove Theorem 1.21.

Definition 5.1. *An* arithmetic read-once formula *(ROF for short)* Φ *over a field* $\mathbb{F}$ *in the variables* $\boldsymbol{x} = (x_1,\ldots,x_n)$ *is a binary tree* T *whose leaves are labeled with input variables and a pairs of field elements* $(\alpha,\beta)\in\mathbb{F}^2$*, and whose internal nodes are labeled with the arithmetic operations* $\{+,\times\}$ *and a field element* $\alpha\in\mathbb{F}$*. Each input variable can label at most one leaf. The computation is performed in the following way: A leaf labeled with the variable* x_i *and with* (α,β)*, computes the polynomial* $\alpha x_i+\beta$*. If a node* v *is labeled with the operation* $*\in\{+,\times\}$ *and with* $\alpha\in\mathbb{F}$*, and its children compute the polynomials* Φ_{v_1} *and* Φ_{v_2}*, then the polynomial computed at* v *is* $\Phi_v = \Phi_{v_1} * \Phi_{v_2} + \alpha$*. A polynomial* $f(\boldsymbol{x})$ *is called a* read-once polynomial *(ROP for short) if* $f(\boldsymbol{x})$ *can be computed by a ROF.*

Observation 5.2. *Read-once polynomials are always multilinear polynomials.*

We next define formulas in alternating normal form, as was first defined in [GKQ14].

Definition 5.3 (Section 3.2 in [GKQ14])**.** *We say that an arithmetic formula Φ, over $\mathbb{F}$, is in* alternating normal form *(Φ is called an* ANF *for short) if:*

1. *The underlying tree of Φ is a complete rooted binary tree (the root node is called the output node). In particular,* $\mathrm{size}(\Phi) = 2^{\mathrm{depth}(\Phi)+1} - 1$*, where* $\mathrm{size}(\Phi)$ *is the number of nodes in the tree of Φ and* $\mathrm{depth}(\Phi)$ *is the maximum distance of a leaf node from the output node of Φ.*
2. *The internal nodes consist of alternating layers of $+$ and $\times$ gates. In particular, the label of an internal node at distance d from the closest leaf node is $+$ if d is even and $\times$ otherwise. So if the root node is a $+$ node, its children are all $\times$ nodes, its grandchildren are all $+$ etc.*
3. *The leaves of the tree are labeled with linear functions. That is, each leaf is labeled with $\ell(\boldsymbol{x}) = a_0 + \sum_{i=1}^{n} a_i x_i$, where each $a_i \in \mathbb{F}$ is a scalar.*

The product depth Δ *of Φ is the number of layers of product gates. The number of leaves of Φ is therefore always 4^{Δ} if the top gate is $+$, and $\frac{1}{2} \cdot 4^{\Delta}$ if the top gate is $\times$.*

The class $\mathrm{ANF}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$ mentioned in section 1.2.2 is defined in terms of the following canonical read-once ANF formula (ROANF for short):

Definition 5.4 (Notation from Fact 3.4 of [GKQ14])**.** *We denote the canonical ROANF polynomial, of product depth Δ on 4^{Δ} variables, as $ANF_{\Delta}(\boldsymbol{x})$. It is defined recursively as follows:*

$$\begin{aligned} ANF_0(\boldsymbol{x}) &= x_1 \\ ANF_{\Delta+1}(\boldsymbol{x}) &= ANF_{\Delta}\left(\boldsymbol{x}^{(1)}\right) ANF_{\Delta}\left(\boldsymbol{x}^{(2)}\right) + ANF_{\Delta}\left(\boldsymbol{x}^{(3)}\right) ANF_{\Delta}\left(\boldsymbol{x}^{(4)}\right) , \end{aligned}$$

where $\boldsymbol{x}^{(i)}$ is the 4^{Δ}-tuple of variables $\{x_{(i-1)\cdot 4^{\Delta}+1}, \ldots, x_{i \cdot 4^{\Delta}}\}$.

For example, $\mathrm{ANF}_1(\boldsymbol{x}) = x_1 x_2 + x_3 x_4$.

Observe that any polynomial in $\mathrm{ANF}_{\Delta}^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})}$ is an ANF according to Definition 5.3, but not vice versa.

Next we give some basic definitions concerning the underlying tree of a ROF, or of a ROANF.

Definition 5.5. *Let Φ be a ROF and v_i, v_j nodes of Φ. The* first common gate *of v_i, v_j (denoted* $\mathrm{fcg}(v_i, v_j)$*) is the first gate in Φ common to all the paths from v_i and v_j to the root of the formula.*

Definition 5.6. *Let T be the computation tree of some ROP polynomial $g \in \mathbb{F}[\boldsymbol{x}]$. For a node $v \in T$ that is not the root, we denote by* $\mathrm{sib}(v) \in T$ *the unique sibling of v in T. When clear from context,* $\mathrm{sib}(v) \in \mathbb{F}[\boldsymbol{x}]$ *denotes the polynomial computed at node* $\mathrm{sib}(v)$*.*

We may characterize $\mathrm{mon}(\mathrm{ANF}_{\Delta}(\boldsymbol{x}))$ by the first common gates of pairs of variables appearing in the monomials:

Observation 5.7. *$\boldsymbol{x}^{\boldsymbol{e}} \in \mathrm{mon}(ANF_{\Delta}(\boldsymbol{x}))$ if and only if $\boldsymbol{x}^{\boldsymbol{e}}$ is multilinear of degree 2^{Δ}, and for every $x_i \neq x_j \in \mathrm{var}(\boldsymbol{x}^{\boldsymbol{e}})$ it holds that* $\mathrm{fcg}(x_i, x_j)$ *is a product gate.*

Observation 5.8. *Let $n = 4^\Delta$. Let T be the computation tree of $ANF_\Delta(\boldsymbol{x})$ (from Definition 5.4 above). Fix some variable $x_i \in \boldsymbol{x}$ and let $\{v_1, \ldots, v_\Delta\} \subseteq T$ be the addition gates on the path from x_i to the root of T, where v_Δ is the root. Denote with $v_0 \in T$ the leaf labeled x_i. Then, recalling Definition 5.6,*

$$\frac{\partial ANF_\Delta}{\partial x_i} = \prod_{k=0}^{\Delta-1} \mathrm{sib}(v_k) = \prod_{k=0}^{\Delta-1} ANF_k(\mathrm{var}(\mathrm{sib}(v_k))) \,.$$

Corollary 5.9. *For any set of variables $S \subseteq \boldsymbol{x}$, $\frac{\partial ANF_\Delta}{\partial S}$ is either zero, or a product of variable-disjoint ROANFs.*

Corollary 5.10. *For any $\mathbf{0} \neq \boldsymbol{u} \in \mathbb{F}^{4^\Delta}$, $\frac{\partial ANF_\Delta}{\partial \boldsymbol{u}}$ is non-zero.*

Proof. Denote $\boldsymbol{u} = (u_1, \ldots, u_n)$. By Observation 5.8, every monomial of $\frac{\partial \mathrm{ANF}_\Delta}{\partial x_i}$ is divisible by $\mathrm{sib}(x_i)$ and is not divisible by x_i. Furthermore, for every $j \neq i$, any monomial of $\frac{\partial \mathrm{ANF}_\Delta}{\partial x_j}$ that contains $\mathrm{sib}(x_i)$, must also contain x_i. Thus, in any linear combination $\frac{\partial \mathrm{ANF}_\Delta}{\partial \boldsymbol{u}} = \sum_{i=1}^n u_i \frac{\partial \mathrm{ANF}_\Delta}{\partial x_i}$, no cancellations can occur as the monomial sets in the summed polynomials are disjoint. □

We first give the simple proof of Theorem 1.21, that separates $\mathrm{ANF}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$, $\mathrm{ROF}^{\mathrm{GL}(\mathbb{F})}$ and $\mathrm{VP_e}$, and that shows that their closures are equal.

Proof of Theorem 1.21. From the definition it is obvious that $\mathrm{ANF}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})} \subseteq \mathrm{ROF}^{\mathrm{GL}(\mathbb{F})}$. It is also clear that the classes are different as the degree of every polynomial in $\mathrm{ANF}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$ is always a power of 2, which is not necessarily the case for polynomials in $\mathrm{ROF}^{\mathrm{GL}(\mathbb{F})}$. As polynomials in $\mathrm{ROF}^{\mathrm{GL}(\mathbb{F})}$ are multilinear with respect to some basis, it is also clear that $\mathrm{ROF}^{\mathrm{GL}(\mathbb{F})} \subsetneq \mathrm{VP_e}$, as the example $f(x) = x^2$ shows. It is also not hard to demonstrate a multilinear polynomial in $\mathrm{VP_e}$ that is not in $\mathrm{ROF}^{\mathrm{GL}(\mathbb{F})}$. The next claim follows example 3.8 of [SV14].

Claim 5.11. $f(\boldsymbol{x}) = x_1x_2 + x_2x_3 + x_3x_1 \notin ROF^{GL(\mathbb{F})}$.

Proof. Assume for a contradiction that there is some ROF formula containing f in its orbit. As f is irreducible, the top gate of Φ is an addition gate. As there cannot be any cancellations in Φ, the children of the root must compute homogeneous degree 2 polynomials. It is not hard to see that this means that the polynomial computed cannot be written as a ROF in only three linear functions, as one child of the root must compute a linear function. □

To show that the closures are equal, we note that Proposition 3.2 of [GKQ14] states that any polynomial that is computed by a size s formula, can be computed by an ANF formula of size $O(s^4)$. As the leaves of an ANF formula are labeled with linear functions, we can approximate these linear functions with linearly independent linear functions and thus conclude that $\mathrm{VP_e} \subseteq \overline{\mathrm{ANF}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}}$. The claim about the closures immediately follows. □

5.1 A hitting set generator for orbits of read-once formulas

In this section we prove Theorem 1.22 that gives a hitting set for $\mathrm{ROF}^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})}$. Our proof follows the proof of [SV15], who constructed such a generator for ROFs. We note that Minahan and Volkovich significantly improved upon the result of [SV15], namely, they achieved a polynomial-sized hitting set for ROFs. However, we do not know how to adapt their approach to orbits of ROFs and instead use the method of [SV15] that

is based on taking partial derivatives, an operation that works well when composing the ROF with a k-independent map (recall Lemma 3.9). We now turn to proving Theorem 1.22.

Proof of Theorem 1.22. The proof of the theorem is by induction on the number of variables in the underlying ROF, which we denote by m. In fact, we claim something stronger:

Let Φ be a ROF on $m \leq 2^t$ many variables that computes a non-constant polynomial. Then, for $f \in \Phi^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})}$ and any $(t+1)$-independent polynomial map $\mathcal{G}$, over $\mathbb{F}$, $f \circ \mathcal{G}$ is a non-constant polynomial.

For $m \leq 2$ the claim follows from Observation 3.1.

Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ be in the orbit of some ROF, on m many variables, $\Phi(w_1, \ldots, w_m)$. Let t be the smallest integer such that $m \leq 2^t$. By definition, for some linearly independent n-variate linear functions $\ell_1, \ldots, \ell_m$, $f(\boldsymbol{x}) = \Phi(\ell_1(\boldsymbol{x}), \ldots, \ell_m(\boldsymbol{x}))$ (where we abuse notation and identify Φ with the polynomial that it computes). Let $\{\boldsymbol{v}_i\}$ be a dual set to $\{\ell_i\}$.

As in the proof of Lemma 5.1 of [SV15], we split the proof into cases depending on the top gate of Φ. Let $\mathcal{G}_1, \mathcal{G}_t$ be a 1-independent polynomial map and a t-independent polynomial map, respectively, such that $\mathcal{G} = \mathcal{G}_1 + \mathcal{G}_t$.

Case $\Phi = \Phi_1 + \Phi_2 + \alpha$: As Φ_1 and Φ_2 are variable disjoint, we can assume, WLOG that $|\mathrm{var}(\Phi_1)| \leq m/2 \leq 2^{t-1}$. Assume further, WLOG, that $\frac{\partial \Phi_1}{\partial w_1} \neq 0$. As Φ_2 does not depend on w_1, we get from Lemma 3.8 that $\frac{\partial f}{\partial \boldsymbol{v}_1} = \frac{\partial \Phi_1}{\partial w_1}(\ell_1, \ldots, \ell_m) \neq 0$. By our induction hypothesis, $\left(\frac{\partial f}{\partial \boldsymbol{v}_1}\right) \circ \mathcal{G}_t = \left(\frac{\partial f_1}{\partial \boldsymbol{v}_1}\right) \circ \mathcal{G}_t$ is a non-constant polynomial. Lemma 3.9 implies that $f \circ \mathcal{G} = f \circ (\mathcal{G}_1 + \mathcal{G}_t) \neq 0$, and it is clearly not a constant polynomial.

Case $\Phi = \Phi_1 \times \Phi_2 + \alpha$: As we can assume that both Φ_1 and Φ_2 are non-constant (there is always such formula computing $\Phi(\boldsymbol{w})$ if it is not the constant polynomial), they both contain less than m variables. Denote $f_i = \Phi_i(\ell_1, \ldots, \ell_m)$, so that $f = f_1 \cdot f_2 + \alpha$. The induction hypothesis implies that $f_1 \circ \mathcal{G}_{t+1}$ and $f_2 \circ \mathcal{G}_{t+1}$ are both non-constant. Hence, $f \circ \mathcal{G}_{t+1} = (f_1 \circ \mathcal{G}_{t+1}) \cdot (f_2 \circ \mathcal{G}_{t+1}) + \alpha$ is also non-constant, as we wanted to prove. □

As before, Corollary 1.23 follows immediately from Theorem 1.22 and Observation 1.14.

5.2 An interpolating set generator for $\mathbf{ANF}^{\mathbf{GL}^{\mathbf{aff}}(\mathbb{F})}$

In this section, we construct an interpolating set generator for $\mathrm{ANF}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$, thus proving Theorem 1.24. We restate the theorem to ease the reading.

Theorem 1.24. *Let $f_1 = ANF_{\Delta_1}(A_1\boldsymbol{x} + \boldsymbol{b}_1), f_2 = ANF_{\Delta_2}(A_2\boldsymbol{x} + \boldsymbol{b}_2) \in ANF^{GL_n^{aff}(\mathbb{F})}$ and $f = f_1 - f_2$. Set $k \triangleq 2\max\{\Delta_1, \Delta_2\} + 7$ and let $\mathcal{G}$ be any uniform k-independent polynomial map, over $\mathbb{F}$. If $f \neq 0$ then $f \circ \mathcal{G} \neq 0$.*

The first step in the proof is a reduction to the case where f_1 and f_2 are "almost the same". Recall that by Fact 2.4, f_1 and f_2 can be equal and still compute different linear functions at their bottom layer. The next lemma (roughly) shows that composing $\mathrm{ANF}_\Delta(\boldsymbol{x})$ with an $O(\Delta)$-independent map, preserves equivalence of different ANFs while not introducing any new equivalences.

Lemma 5.12. *Let $f_1 = ANF_{\Delta_1}(A_1\boldsymbol{x} + \boldsymbol{b}_1), f_2 = ANF_{\Delta_2}(A_2\boldsymbol{x} + \boldsymbol{b}_2) \in ANF^{GL_n^{aff}(\mathbb{F})}$ and $f = f_1 - f_2$. For $i = 1, 2$, denote by $h_i \triangleq x_0^{\deg(f_i)} f_i(\frac{x_1}{x_0}, \ldots, \frac{x_n}{x_0})$ the homogenization of f_i, and let $\tilde{A}_i$ be an extension of A_i such that $\tilde{A}_i \in GL_{n+1}(\mathbb{F})$ and $h_i = ANF_{\Delta_i}(\tilde{A}_i\boldsymbol{x})$. Set $k = 2\max\{\Delta_1, \Delta_2\} + 7$ and let $\mathcal{G}$ be any uniform k-independent polynomial map. If $f \neq 0$ then at least one of the following holds:*

1. $f \circ \mathcal{G} \neq 0$.

2. $\Delta_1 = \Delta_2$, *and there is a* $1-1$ *map between the quadratic forms of* $h_2(\tilde{A}_1^{-1}\boldsymbol{x})$ *and those of* $ANF_{\Delta_1}(\boldsymbol{x})$, *such that any two quadratics that were matched have the same monomials,* possibly with different coefficients.[13] *Furthermore, the map between the quadratics is a* $TR_{4^{\Delta_1-1}}(\mathbb{F})$ *symmetry (see Definition 2.2).*

Observe that if $\{\ell_{i,j}\}$ are linear functions such that $f_i = \mathrm{ANF}_{\Delta_i}\left(\ell_{i,1}(\boldsymbol{x}), \ldots, \ell_{i,4^{\Delta_i}}\right)$, then the condition "the monomials appearing in the quadratic forms of $h_2\left((\tilde{A}_1)^{-1}\boldsymbol{x}\right)$ are identical to the monomials of the quadratic forms of $\mathrm{ANF}_{\Delta_1}(\boldsymbol{x})$, up to $\mathrm{TR}_{4^{\Delta_1}}(\mathbb{F})$ symmetry" is equivalent to saying that there exists a permutation $\pi \in \mathrm{TR}_{4^{\Delta_1-1}}(\mathbb{F})$, matching quadratics in f_2 to those of f_1, such that when we represent the ith quadratic $q_i^{(2)}$ of f_2 according to the linear functions $\left\{\ell_{1,1}, \ldots, \ell_{1,4^{\Delta_1}}\right\}$, then $q_i^{(2)}$ has the same set of $\left\{\ell_{1,1}, \ldots, \ell_{1,4^{\Delta_1}}\right\}$-monomials as $q_{\pi(i)}^{(1)}$, the $\pi(i)$th quadratic in f_1. In general, whenever we say "up to $\mathrm{TR}_{4^{\Delta_1-1}}(\mathbb{F})$ symmetry" we mean that there exists a permutation $\pi \in \mathrm{TR}_{4^{\Delta_1-1}}(\mathbb{F})$ such that the statement holds when we apply π to the quadratics computed at the bottom layers.

Once we have this in mind we can see that the only "bad" case is when, for every i, $\ell_{2,i} = \alpha_i \cdot \ell_{1,i}$, for scalars $\alpha_i \in \mathbb{F}$ (possibly after applying some $\mathrm{TR}_{4^{\Delta_1-1}}(\mathbb{F})$ symmetry). Thus, the proof of Theorem 1.24 would follow from the next lemma.

Lemma 5.13. *Let* $\ell_1(\boldsymbol{x}), \ldots, \ell_n(\boldsymbol{x})$ *be linearly independent linear forms, and let* $\alpha_1, \ldots, \alpha_n \in \mathbb{F}$ *be non-zero constants. Let* $f = ANF_\Delta(\ell_1, \ldots, \ell_n)$ *and* $g = ANF_\Delta(\alpha_1\ell_1, \ldots, \alpha_n\ell_n)$, *and let* $\mathcal{G}$ *be a* $(2\Delta+2)$*-independent polynomial map. It holds that if* $f - g \neq 0$ *then* $(f-g) \circ \mathcal{G} \neq 0$.

We first give the formal proof of the theorem and then prove the main lemmas.

Proof of Theorem 1.24. Let h_1, h_2 be the homogenizations of f_1, f_2 as in the premise of Lemma 5.12. Assume Case 2 of Lemma 5.12 holds, as otherwise we are done. Then, for $n = 4^{\Delta_1}$, this assumption implies that for some linearly independent linear forms $\ell_1, \ldots, \ell_n$ and non-zero constants $\alpha_1, \ldots, \alpha_n \in \mathbb{F}$, $h_1 = \mathrm{ANF}_{\Delta_1}(\ell_1, \ldots, \ell_n)$ and $h_2 = \mathrm{ANF}_{\Delta_1}(\alpha_1\ell_1, \ldots, \alpha_n\ell_n)$. By Lemma 5.13, if $f \neq 0$ then $(h_1 - h_2) \circ \mathcal{G} \neq 0$; and by the following lemma (Lemma 5.14), we may conclude $f \circ \mathcal{G} \neq 0$. □

Lemma 5.14. *Let* $\boldsymbol{x} = (x_1, \ldots, x_n)$ *and* $f \in \mathbb{F}[\boldsymbol{x}]$ *be a polynomial of degree* d. *Let* $g(x_0, \boldsymbol{x}) = x_0^d f(\frac{x_1}{x_0}, \ldots, \frac{x_n}{x_0})$ *be the homogenization of* f, *and let* $\mathcal{G} : \mathbb{F}^t \to \mathbb{F}^{n+1}$ *be a polynomial map such that the coordinates of* $\mathcal{G}$ *are homogeneous polynomials of identical degree. Let* $H : \mathbb{F}^t \to \mathbb{F}^n$ *be the restriction of* $\mathcal{G}$ *to the coordinates in* $[n]$ *(i.e., we ignore the* 0*th coordinate). If* $g \circ \mathcal{G} \neq 0$ *then* $f \circ H \neq 0$.

Proof. Write $g(x_0, \boldsymbol{x}) = \sum_{i=0}^{d} x_0^i f^{[d-i]}(\boldsymbol{x})$, and denote by $\mathcal{G}_0$ the 0th coordinate of $\mathcal{G}$ (such that $\mathcal{G} = (\mathcal{G}_0, H)$). We get:

$$g \circ \mathcal{G} = \sum_{i=0}^{d} (\mathcal{G}_0)^i \cdot (f^{[d-i]} \circ H).$$

Fix $i \in [d+1]_0$ to be the minimal index such that $f^{[d-i]} \circ H \neq 0$. Such an index must exist, because $g \circ \mathcal{G} \neq 0$. As all coordinates of $\mathcal{G}$ are homogeneous and of identical degree, for any $i < i' \in [d]$ such that $f^{[d-i']} \circ H$ is non-zero, we must have $\deg(f^{[d-i]} \circ H) > \deg(f^{[d-i']} \circ H)$. Thus, nothing can cancel $f^{[d-i]} \circ H$ in $f \circ H$, proving $f \circ H \neq 0$. □

[13] Thus, composition with $\mathcal{G}$ does not exactly preserve equivalence.

5.2.1 Proof of Lemma 5.12

The high-level strategy for proving Lemma 5.12 is as follows: first, we show that if Case 2 of the lemma is false, then there are $\boldsymbol{v}, \boldsymbol{u} \in \mathbb{F}^n$ such that $\frac{\partial^2 f}{\partial \boldsymbol{v} \partial \boldsymbol{u}} = \frac{\partial^2 f_1}{\partial \boldsymbol{v} \partial \boldsymbol{u}} \neq 0$. This is proven in Lemma 5.16, based on the structural result of Lemma 5.15. After that, we prove that $(k-2)$-independent polynomial maps hit $\frac{\partial^2 f_1}{\partial \boldsymbol{v} \partial \boldsymbol{u}}$, in Lemma 5.18.

To prove Lemma 5.12, we first set out to prove that inclusion of monomial sets is enough to deduce that Case 2 of Lemma 5.12 holds:

Lemma 5.15. *Let $g(\boldsymbol{x}) = ANF_\Delta(A\boldsymbol{x}+\boldsymbol{b})$ for some $(A, \boldsymbol{b}) \in GL_n^{aff}(\mathbb{F})$. Let $q_1, \ldots, q_{4^{\Delta-1}}$ denote the quadratic forms of ANF_Δ such that $g = ANF_{\Delta-1}(q_1(A\boldsymbol{x}+\boldsymbol{b}), \ldots, q_{4^{\Delta-1}}(A\boldsymbol{x}+\boldsymbol{b}))$. If $\mathrm{mon}(g) \subseteq \mathrm{mon}(ANF_\Delta(\boldsymbol{x}))$, then $\boldsymbol{b} = 0$ and $\mathrm{mon}(q_i(A\boldsymbol{x})) = \mathrm{mon}(q_i(\boldsymbol{x}))$, up to $TR_{4^{\Delta-1}}(\mathbb{F})$ symmetry. In particular, $\mathrm{mon}(g) = \mathrm{mon}(ANF_\Delta(\boldsymbol{x}))$.*

Proof. The proof is by induction on Δ.

For $\Delta = 1$, we know $\mathrm{mon}(g) \subseteq \{x_1x_2, x_3x_4\}$. $\mathrm{ANF}_1(\boldsymbol{x})$ is irreducible, so $\mathrm{mon}(g) \neq \{x_1x_2\}$ or $\{x_3x_4\}$, and g is non-constant so $\mathrm{mon}(g) = \{x_1x_2, x_3x_4\}$. Now, let $\ell_1(\boldsymbol{x}), \ldots, \ell_4(\boldsymbol{x})$ denote linearly independent linear functions such that $g = \ell_1(\boldsymbol{x})\ell_2(\boldsymbol{x}) + \ell_3(\boldsymbol{x})\ell_4(\boldsymbol{x})$, and denote $\alpha_i \triangleq \ell_i(0)$. The 1-homogeneous part of g is given by:

$$g^{[1]} = \alpha_1\ell_2^{[1]}(\boldsymbol{x}) + \alpha_2\ell_1^{[1]}(\boldsymbol{x}) + \alpha_3\ell_4^{[1]}(\boldsymbol{x}) + \alpha_4\ell_3^{[1]}(\boldsymbol{x}).$$

As g is 2-homogeneous, $g^{[1]} = 0$. As the $\ell_i^{[1]}$s are linearly independent, this implies $\alpha_1 = \ldots = \alpha_4 = 0$, and therefore $\boldsymbol{b} = 0$, proving the base case.

Assume $\Delta > 1$ and denote $\mathrm{ANF}_\Delta(\boldsymbol{x}) = F_1F_2 + F_3F_4$, where $F_1, \ldots, F_4$ are the grandchildren of the root of ANF_Δ. In particular, each F_i is an $\mathrm{ANF}_{\Delta-1}(\boldsymbol{x})$ formula (on one quarter of the variables). We note that g is 2^Δ homogeneous because $\mathrm{mon}(g) \subseteq \mathrm{mon}(\mathrm{ANF}_\Delta)$, so $g = \mathrm{ANF}_\Delta(A\boldsymbol{x})$ (because $\mathrm{ANF}_\Delta(A\boldsymbol{x}+\boldsymbol{b})^{[2^\Delta]} = \mathrm{ANF}_\Delta(A\boldsymbol{x})$). Denote $g = g_1g_2 + g_3g_4$ where $g_i(\boldsymbol{x}) = F_i(A\boldsymbol{x})$.

First, note that $\mathrm{var}(g) = \mathrm{var}(\mathrm{ANF}_\Delta(\boldsymbol{x}))$: we already know $\mathrm{var}(g) \subseteq \mathrm{var}(\mathrm{ANF}_\Delta(\boldsymbol{x}))$, and g must depend on at least 4^Δ variables, or the 4^Δ linear functions on the leaves cannot be linearly independent.

Next, observe that g_1g_2 and g_3g_4 must be variable disjoint: if $x_i \in \mathrm{var}(g_1g_2) \cap \mathrm{var}(g_3g_4)$, then $\left(\frac{\partial g}{\partial x_i}\right)(A^{-1}\boldsymbol{x})$ is a sum of non-constant, variable-disjoint, multilinear polynomials, and $\left(\frac{\partial g}{\partial x_i}\right)(\boldsymbol{x})$ is therefore irreducible (recall Observation 2.8). However, if we denote by x_j the sibling of x_i in $\mathrm{ANF}_\Delta(\boldsymbol{x})$, the fact that $\mathrm{mon}(g) \subseteq \mathrm{mon}(\mathrm{ANF}_\Delta(\boldsymbol{x}))$ implies that every monomial of $\frac{\partial g}{\partial x_i}(\boldsymbol{x})$ is divisible by x_j. As $\Delta > 1$, we have $\deg\left(\frac{\partial g}{\partial x_i}\right) \geq 3$, and therefore $\frac{\partial g}{\partial x_i}$ must be reducible, in contradiction. Thus, $\mathrm{var}(g_1g_2) \cap \mathrm{var}(g_3g_4) = \varnothing$, and in particular $\mathrm{mon}(g_1g_2), \mathrm{mon}(g_3g_4) \subseteq \mathrm{mon}(\mathrm{ANF}_\Delta)$.

Next, assume, WLOG, there exist some monomial $\boldsymbol{x}^{\boldsymbol{e}} \in \mathrm{mon}(F_1F_2)$ such that $\boldsymbol{x}^{\boldsymbol{e}} \in \mathrm{mon}(g_1g_2)$. If g_1g_2 contains a monomial of F_3F_4, then g_1g_2 can be partitioned into a sum of two variable-disjoint, non-constant, multilinear polynomials; which would contradict reducibility of g_1g_2. Thus, $\mathrm{mon}(g_1g_2) \subseteq \mathrm{mon}(F_1F_2)$. As we showed that $\mathrm{var}(g) = \mathrm{var}(\mathrm{ANF}_\Delta(\boldsymbol{x}))$, the conditions on the monomials implies that there must exist some monomial of F_3F_4 in g, so we may conclude $\mathrm{mon}(g_3g_4) \subseteq \mathrm{mon}(F_3F_4)$, and in addition, $\mathrm{var}(g_1g_2) = \mathrm{var}(F_1F_2)$ and $\mathrm{var}(g_3g_4) = \mathrm{var}(F_3F_4)$.

To apply induction, it remains to prove that $\mathrm{mon}(g_i) \subseteq \mathrm{mon}(F_i)$ for $i \in [4]$ (up to $\mathrm{TR}(\mathbb{F})$); focus on g_1g_2 and WLOG assume $\mathrm{var}(g_1) \cap \mathrm{var}(F_1) \neq \varnothing$.

As all monomials of g_1g_2 are multilinear, $\mathrm{var}(g_1) \cap \mathrm{var}(g_2) = \varnothing$. As $\Delta > 1$, we may denote by p_1, p_2, p_3, p_4

the variable-disjoint polynomials such that $F_1 = p_1 + p_2$ and $F_2 = p_3 + p_4$:

$$F_1F_2 = (p_1 + p_2)(p_3 + p_4) = p_1p_3 + p_1p_4 + p_2p_3 + p_2p_4 \ .$$

We now show that g_1 cannot contain variables from both F_1 and F_2. Assume there exist monomials $\boldsymbol{x}^{\boldsymbol{e}_1}, \boldsymbol{x}^{\boldsymbol{e}_2} \in \mathrm{mon}(g_1)$ such that $\boldsymbol{x}^{\boldsymbol{e}_1}$ contains variables from $\mathrm{var}(F_1)$ and $\boldsymbol{x}^{\boldsymbol{e}_2}$ contains variables from $\mathrm{var}(F_2)$ ($\boldsymbol{x}^{\boldsymbol{e}_1}$ and $\boldsymbol{x}^{\boldsymbol{e}_2}$ may be the same monomial). WLOG assume $\mathrm{var}(\boldsymbol{x}^{\boldsymbol{e}_1}) \cap \mathrm{var}(p_1) \neq \varnothing$, and likewise $\mathrm{var}(\boldsymbol{x}^{\boldsymbol{e}_2}) \cap \mathrm{var}(p_3) \neq \varnothing$. Let $\boldsymbol{x}^{\boldsymbol{c}} \in \mathrm{mon}(g_2)$, and let $x_i | \boldsymbol{x}^{\boldsymbol{c}}$. If $x_i \in \mathrm{var}(p_2)$, then $\boldsymbol{x}^{\boldsymbol{e}_1} \cdot \boldsymbol{x}^{\boldsymbol{c}} \in \mathrm{mon}(g_1g_2)$ is a monomial involving variables from both p_1 and p_2, in contradiction; by a symmetric argument, we cannot have $x_i \in \mathrm{var}(p_4)$. Thus, all monomials of g_2 may involve only variables of p_1 and p_3, i.e., $\mathrm{var}(g_2) \subseteq \mathrm{var}(p_1) \uplus \mathrm{var}(p_3)$. Therefore, the only way to get monomials involving variables of p_2 or p_4 is via monomials of g_1, so g_1 must contain monomials $\boldsymbol{x}^{\boldsymbol{e}_1'}, \boldsymbol{x}^{\boldsymbol{e}_2'}$ containing variables of p_2 and p_4, respectively (here we use the fact that $\mathrm{var}(g_1g_2) = \mathrm{var}(F_1F_2)$). As before, we get $\mathrm{var}(g_2) \subseteq \mathrm{var}(p_2) \uplus \mathrm{var}(p_4)$, in contradiction.

We can therefore conclude that $\mathrm{var}(g_1) \subseteq \mathrm{var}(F_1)$. Using $\mathrm{var}(g_1g_2) = \mathrm{var}(F_1F_2)$, we deduce $\mathrm{var}(g_2) \cap \mathrm{var}(F_2) \neq \varnothing$, and repeating the argument of the previous paragraph we conclude $\mathrm{var}(g_2) \subseteq \mathrm{var}(F_2)$, which implies $\mathrm{var}(g_i) = \mathrm{var}(F_i)$ for $i = 1, 2$.

As $\mathrm{mon}(g_1g_2) \subseteq \mathrm{mon}(F_1F_2)$, we may conclude $\mathrm{mon}(g_i) \subseteq \mathrm{mon}(F_i)$ (for $i = 1, 2$):

$$\mathrm{mon}(F_i) = \{\boldsymbol{x}^{\boldsymbol{e}}\big|_{(\boldsymbol{x} \setminus \mathrm{var}(F_i))=1} : \boldsymbol{x}^{\boldsymbol{e}} \in \mathrm{mon}(F_1F_2)\} \supseteq \{\boldsymbol{x}^{\boldsymbol{e}}\big|_{(\boldsymbol{x} \setminus \mathrm{var}(g_i))=1} : \boldsymbol{x}^{\boldsymbol{e}} \in \mathrm{mon}(g_1g_2)\} = \mathrm{mon}(g_i).$$

Finally, we may apply the induction hypothesis and conclude $\boldsymbol{b} = \boldsymbol{0}$ and $\mathrm{mon}(q_i(A\boldsymbol{x})) = \mathrm{mon}(q_i(\boldsymbol{x}))$, up to $\mathrm{TR}_{4^{\Delta-1}}(\mathbb{F})$ symmetry. I.e., there is a permutation $\pi \in \mathrm{TR}_{4^{\Delta-1}}(\mathbb{F})$ such that $\mathrm{mon}(q_i(A\boldsymbol{x})) = \mathrm{mon}(q_{\pi(i)}(\boldsymbol{x}))$ ($\mathrm{TR}_{4^{\Delta-1}}(\mathbb{F})$ symmetry enters every time we use "WLOG" in the proof). □

The next step is showing that, if Case 2 of Lemma 5.12 does not hold, then we may choose a pair of vectors by which to take a derivative of $f = f_1 - f_2$ such that $\frac{\partial^2 f_1}{\partial \boldsymbol{v}_1 \boldsymbol{v}_2} = 0$ and $\frac{\partial^2 f_2}{\partial \boldsymbol{v}_1 \boldsymbol{v}_2} \neq 0$. This is formalized in Lemma 5.16 below, and is proved by applying Lemma 5.15.

Lemma 5.16. *Let $f = ANF_\Delta(A_1\boldsymbol{x})$ and $g = ANF_\Delta(A_2\boldsymbol{x})$, for some $A_1, A_2 \in GL_n(\mathbb{F})$. Denote $\tilde{g} \triangleq g(A_1^{-1}\boldsymbol{x})$. If $\mathrm{mon}(\tilde{g}) \neq \mathrm{mon}(ANF_\Delta(\boldsymbol{x}))$, then there exist $\boldsymbol{v}, \boldsymbol{u} \in \mathbb{F}^n$ such that $\frac{\partial^2 f}{\partial \boldsymbol{v} \partial \boldsymbol{u}} = 0$ and $\frac{\partial^2 g}{\partial \boldsymbol{v} \partial \boldsymbol{u}} \neq 0$.*

Proof. Let $\ell_1(\boldsymbol{x}), \ldots, \ell_n(\boldsymbol{x})$ be linearly independent linear forms such that $f = \mathrm{ANF}_\Delta(\ell_1(\boldsymbol{x}), \ldots, \ell_{4^\Delta}(\boldsymbol{x}))$, and let $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{4^\Delta}\}$ be a dual set.

By Lemma 5.15, the fact that $\mathrm{mon}(\tilde{g}) \neq \mathrm{mon}(\mathrm{ANF}_\Delta(\boldsymbol{x}))$ implies $\mathrm{mon}(\tilde{g}) \nsubseteq \mathrm{mon}(\mathrm{ANF}_\Delta(\boldsymbol{x}))$. Fix some monomial $\boldsymbol{x}^{\boldsymbol{e}} \in \mathrm{mon}(\tilde{g}) \setminus \mathrm{mon}(\mathrm{ANF}_\Delta(\boldsymbol{x}))$, and choose $\boldsymbol{v}, \boldsymbol{u}$ as follows:

- If $\boldsymbol{x}^{\boldsymbol{e}}$ is not a multilinear monomial, let x_i be such that $x_i^2 | \boldsymbol{x}^{\boldsymbol{e}}$. Set $\boldsymbol{v} = \boldsymbol{u} \triangleq \boldsymbol{v}_i$. In this case, we get from Lemma 3.8 that $\frac{\partial^2 f}{\partial \boldsymbol{v} \partial \boldsymbol{u}} = \frac{\partial^2 \mathrm{ANF}_\Delta}{\partial x_i^2}(\ell_1, \ldots, \ell_{4^\Delta}) = 0$, as ANF_Δ is multilinear. Clearly $\frac{\partial^2 g}{\partial \boldsymbol{v} \partial \boldsymbol{u}} \neq 0$.
- If $\boldsymbol{x}^{\boldsymbol{e}}$ is multilinear, then let $x_i, x_j \in \mathrm{var}(\boldsymbol{x}^{\boldsymbol{e}})$ be such that $\mathrm{fcg}(x_i, x_j)$ is an addition gate (all monomials of $\tilde{g}$ are of degree exactly 2^Δ, so Observation 5.7 implies the existence of such a pair of variables). Set $\boldsymbol{v} \triangleq \boldsymbol{v}_i, \boldsymbol{u} \triangleq \boldsymbol{v}_j$. Lemma 3.8 again implies that $\frac{\partial^2 f}{\partial \boldsymbol{v} \partial \boldsymbol{u}} = \frac{\partial^2 \mathrm{ANF}_\Delta}{\partial x_i \partial x_j} = 0$, because $\mathrm{fcg}(x_i, x_j)$ is an addition gate in ANF_Δ. As before, it is clear that $\frac{\partial^2 g}{\partial \boldsymbol{v} \partial \boldsymbol{u}} \neq 0$. □

Looking back at Lemma 5.12, Lemma 5.16 allows us to separate f_1 from f_2, provided Case 2 of Lemma 5.12 does not hold. We still need to provide a hitting set for $\frac{\partial^2 f_1}{\partial \boldsymbol{v} \partial \boldsymbol{u}}$, where $\boldsymbol{v}, \boldsymbol{u}$ are arbitrary, and satisfy $\frac{\partial^2 f_1}{\partial \boldsymbol{v} \partial \boldsymbol{u}} \neq 0$.

To do so, we reduce $\frac{\partial^2 f_1}{\partial \boldsymbol{v}\partial \boldsymbol{u}}$ to a single, non-zero product of variable-disjoint ROPs composed with affine transformations (Lemma 5.18). For simplicity, we first reduce to a product of ROPs in the standard basis in Lemma 5.17, and subsequently extend the result to affine orbits in Lemma 5.18.

Lemma 5.17. *Let $\Delta \geq 2$, and let $f(\boldsymbol{x}) = \sum_{i,j} \alpha_{i,j} \frac{\partial^2 ANF_\Delta(\boldsymbol{x})}{\partial x_i \partial x_j}$ be some non-zero linear combination of second derivatives of $ANF_\Delta(\boldsymbol{x})$. Then, there exist variables x_i, x_j, sets $D, Z \subseteq \boldsymbol{x}$ such that $|D| \leq 2$ and $|Z| = 2$, and a constant $\beta_{i,j}$ such that*

$$\left(\frac{\partial^{|D|} f}{\partial D}\right)\Big|_{Z=\mathbf{0}} = \beta i,j \left(\frac{\partial^{2+|D|} ANF_\Delta(\boldsymbol{x})}{\partial x_i \partial x_j \partial D}\right)\Big|_{Z=\mathbf{0}} \neq 0.$$

Proof. First, assume there exist some i, j such that $\alpha_{i,j} \frac{\partial^2 \mathrm{ANF}_\Delta(\boldsymbol{x})}{\partial x_i \partial x_j} \neq 0$ and $x_i \neq \mathrm{sib}(x_j)$. Set $D = \{\mathrm{sib}(x_i), \mathrm{sib}(x_j)\}$. By Observation 5.8, $\frac{\partial^4 \mathrm{ANF}_\Delta(\boldsymbol{x})}{\partial x_i \partial x_j \partial D} \neq 0$ and is a product of variable-disjoint ROPs that do not depend on x_i nor on x_j.

Consider any pair $\{i', j'\} \neq \{i, j\}$ and set $h = \frac{\partial^2 \mathrm{ANF}_\Delta(\boldsymbol{x})}{\partial x_{i'} \partial x_{j'}}$. Note that if $\frac{\partial^2 h}{\partial D} \neq 0$ then $\frac{\partial^2 h}{\partial D}$ is divisible by x_i or by x_j (or both, if $\{i, j\} \cap \{i', j'\} = \varnothing$). If we set $Z \triangleq \{x_i, x_j\}$, then $\left(\frac{\partial^2 h}{\partial D}\right)\Big|_{Z=\mathbf{0}} = 0$. This is true for any $\{i', j'\} \neq \{i, j\}$, and as $\frac{\partial^4 \mathrm{ANF}_\Delta(\boldsymbol{x})}{\partial x_i \partial x_j \partial D}$ does not depend on x_i nor on x_j we get

$$\left(\frac{\partial^2 f}{\partial D}\right)\Big|_{Z=\mathbf{0}} = \left(\frac{\partial^4 \mathrm{ANF}_\Delta}{\partial x_i \partial x_j \partial D}\right)\Big|_{Z=\mathbf{0}} \neq 0 \,.$$

Next, assume all non-zero summands of f, $\alpha_{i,j} \frac{\partial^2 \mathrm{ANF}_\Delta(\boldsymbol{x})}{\partial x_i \partial x_j}$, satisfy $x_i = \mathrm{sib}(x_j)$. Note that if $x_i x_j + x_{i'} x_{j'}$ is a quadratic form of $\mathrm{ANF}_\Delta(\boldsymbol{x})$, then $\frac{\partial^2 \mathrm{ANF}_\Delta(\boldsymbol{x})}{\partial x_i \partial x_j} = \frac{\partial^2 \mathrm{ANF}_\Delta(\boldsymbol{x})}{\partial x_{i'} \partial x_{j'}}$ (Observation 5.8). Therefore,

$$f = \sum_{\substack{x_i x_j + x_{i'} x_{j'} \text{ is a} \\ \text{quadratic of } \mathrm{ANF}_\Delta}} (\alpha_{i,j} + \alpha_{i',j'}) \frac{\partial^2 \mathrm{ANF}_\Delta(\boldsymbol{x})}{\partial x_i \partial x_j} \,.$$

Fix some i, j, i', j' such that $q_1 = x_i x_j + x_{i'} x_{j'}$ is a quadratic of $\mathrm{ANF}_\Delta(\boldsymbol{x})$, and $(\alpha_{i,j} + \alpha_{i',j'}) \frac{\partial^2 \mathrm{ANF}_\Delta(\boldsymbol{x})}{\partial x_i \partial x_j} \neq 0$. As $\Delta \geq 2$, q_1 has a sibling quadratic form; denote it by $q_2 \triangleq \mathrm{sib}(q_1) = x_k x_\ell + x_{k'} x_{\ell'}$ and set $D \triangleq \{x_k\}$. Note that by Observation 5.8, $(\alpha_{i,j} + \alpha_{i',j'}) \frac{\partial^3 \mathrm{ANF}_\Delta(\boldsymbol{x})}{\partial x_i \partial x_j \partial D} \neq 0$, does not depend on $x_i, x_j, x_{i'}, x_{j'}$, and is a product of variable-disjoint ROPs.

Set $Z = \{x_i, x_{i'}\}$. Consider any pair $\{s, t\}$ such that $\{s, t\} \notin \{\{i, j\}, \{i', j'\}\}$ and $x_s = \mathrm{sib}(x_t)$. Set $h = \frac{\partial^2 \mathrm{ANF}_\Delta}{\partial x_t \partial x_s}$. If $\frac{\partial h}{\partial x_k} \neq 0$ then it is divisible by the quadratic form $q_1 = x_i x_j + x_{i'} x_{j'}$ (by Observation 5.8), and thus $\left(\frac{\partial h}{\partial D}\right)\Big|_{Z=\mathbf{0}} = 0$. Hence,

$$\left(\frac{\partial f}{\partial D}\right)\Big|_{Z=\mathbf{0}} = \left((\alpha_{i,j} + \alpha_{i',j'}) \frac{\partial^3 \mathrm{ANF}_\Delta(\boldsymbol{x})}{\partial x_i \partial x_j \partial D}\right)\Big|_{Z=\mathbf{0}} \neq 0 \,. \qquad \square$$

Lemma 5.18. *Let $\Delta \geq 2$, let $f = ANF_\Delta(A\boldsymbol{x} + \boldsymbol{b})$ for some $(A, \boldsymbol{b}) \in GL_n^{aff}(\mathbb{F})$, and let $\boldsymbol{w}, \boldsymbol{u} \in \mathbb{F}^n$. Then, for any $(2\Delta + 5)$-independent polynomial map $\mathcal{G}$, if $\frac{\partial^2 f}{\partial \boldsymbol{w}\partial \boldsymbol{u}} \neq 0$ then $\frac{\partial^2 f}{\partial \boldsymbol{w}\partial \boldsymbol{u}} \circ \mathcal{G} \neq 0$.*

Proof. Let $\ell_1(\boldsymbol{x}), \ldots, \ell_{4\Delta}(\boldsymbol{x})$ be linearly independent linear functions such that $f = \mathrm{ANF}_\Delta(\ell_1, \ldots, \ell_{4\Delta})$. Let $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_{4\Delta}\}$ be a dual set. There exist constants $\alpha_{i,j}$ such that:

$$0 \neq \frac{\partial^2 f}{\partial \boldsymbol{w}\partial \boldsymbol{u}}(\boldsymbol{x}) = \sum_{i,j} \alpha_{i,j} \frac{\partial^2 \mathrm{ANF}_\Delta}{\partial x_i \partial x_j}(A\boldsymbol{x} + \boldsymbol{b}) \,.$$

Denote $g(\boldsymbol{x}) \triangleq \frac{\partial^2 f}{\partial \boldsymbol{w}\partial \boldsymbol{u}}(A^{-1}\boldsymbol{x} - A^{-1}\boldsymbol{b}) = \sum_{i,j}\alpha_{i,j}\frac{\partial^2 \mathrm{ANF}_\Delta}{\partial x_i \partial x_j}(\boldsymbol{x})$, and let x_{i_0}, x_{j_0}, $D = \{x_k, x_\ell\}$, $Z = \{x_r, x_m\}$ and $\beta_{i_0 j_0}$ be as promised by Lemma 5.17. Thus,[14]

$$\left(\frac{\partial^2 g}{\partial D}(\boldsymbol{x})\right)\Big|_{Z=\mathbf{0}} = \left(\beta_{i_0 j_0}\frac{\partial^4 \mathrm{ANF}_\Delta}{\partial x_{i_0}\partial x_{j_0}\partial D}(\boldsymbol{x})\right)\Big|_{x_r=x_m=0} \neq 0\,. \tag{13}$$

From Lemma 3.8 and Equation (13) we deduce that

$$\left(\frac{\partial^4 f}{\partial \boldsymbol{w}\partial \boldsymbol{u}\partial \boldsymbol{v}_k\partial \boldsymbol{v}_\ell}(\boldsymbol{x})\right)\Big|_{\ell_r=\ell_m=0} = \left(\frac{\partial^2 g}{\partial D}(A\boldsymbol{x}+\boldsymbol{b})\right)\Big|_{\ell_r=\ell_m=0} = \left(\beta_{i_0 j_0}\frac{\partial^2}{\partial \boldsymbol{v}_k\partial \boldsymbol{v}_\ell}\left(\frac{\partial^2 \mathrm{ANF}_\Delta}{\partial x_i \partial x_j}(A\boldsymbol{x}+\boldsymbol{b})\right)\right)\Big|_{\ell_r=\ell_m=0} \neq 0\,.$$

Let $\mathcal{G} = \mathcal{G}_1 + \mathcal{G}_2 + \mathcal{G}_{2\Delta+1}$ be a $(2\Delta+5)$-independent map where $\mathcal{G}_1, \mathcal{G}_2$ are 2-independent polynomial maps, $\mathcal{G}_{2\Delta+1}$ is a $(2\Delta+1)$-independent polynomial map, and $\mathcal{G}_1, \mathcal{G}_2, \mathcal{G}_{2\Delta+1}$ are variable-disjoint. As $\left(\frac{\partial^4 f}{\partial \boldsymbol{w}\partial \boldsymbol{u}\partial \boldsymbol{v}_k\partial \boldsymbol{v}_\ell}(\boldsymbol{x})\right)\Big|_{\ell_r=\ell_m=0}$ is a non-zero product of ROPs composed with an affine transformation, where the underlying ROPs depend on at most 4^Δ variables, we get from Theorem 1.22 that $\left(\frac{\partial^4 f}{\partial \boldsymbol{w}\partial \boldsymbol{u}\partial \boldsymbol{v}_k\partial \boldsymbol{v}_\ell}(\boldsymbol{x})\right)\Big|_{\ell_r=\ell_m=0} \circ \mathcal{G}_{2\Delta+1} \neq 0$. Lemma 3.10 implies that $\frac{\partial^4 f}{\partial \boldsymbol{w}\partial \boldsymbol{u}\partial \boldsymbol{v}_k\partial \boldsymbol{v}_\ell}(\mathcal{G}_{2\Delta+1} + \mathcal{G}_2) \neq 0$. Finally, from Lemma 3.9 it follows that $\frac{\partial^2 f}{\partial \boldsymbol{w}\partial \boldsymbol{u}}(\mathcal{G}_{2\Delta+1} + \mathcal{G}_2 + \mathcal{G}_1) \neq 0$, as required. □

We are now ready to prove Lemma 5.12.

Proof of Lemma 5.12. First, assume $\Delta_1 \neq \Delta_2$. WLOG assume $\Delta_1 > \Delta_2$. Let $\ell_1, \ldots, \ell_{4^{\Delta_1}}$ be linearly independent linear functions such that $f_1 = \mathrm{ANF}_{\Delta_1}(\ell_1, \ldots, \ell_{4^{\Delta_1}})$. There must exist some i such that ℓ_i is *not* spanned by the linear functions at the leaves of f_2. Fix some vector $\boldsymbol{v}$ such that $\ell^{[1]}(\boldsymbol{v}) = 0$ for every linear function ℓ labeling a leaf of f_2, and such that $\ell_i^{[1]}(\boldsymbol{v}) = 1$. By Lemma 3.8 and Corollary 5.10, $\frac{\partial f_2}{\partial \boldsymbol{v}} = 0$ and $\frac{\partial f_1}{\partial \boldsymbol{v}} \neq 0$; thus, $0 \neq \frac{\partial f}{\partial \boldsymbol{v}} = \frac{\partial f_1}{\partial \boldsymbol{v}}$. From Lemma 5.18 it follows that any $(2\Delta_1+5)$-independent polynomial map $\mathcal{G}'$ satisfies $\frac{\partial f}{\partial \boldsymbol{v}} \circ \mathcal{G}' \neq 0$; and therefore, using Lemma 3.9, we get $f \circ \mathcal{G} \neq 0$, so Case 1 of the lemma holds.

Next, assume $\Delta_1 = \Delta_2$ and denote $h \triangleq h_1 - h_2$ (recall that h_i is the homogenization of f_i). As $\mathcal{G}$ is uniform, Lemma 5.14 implies that it suffices to prove that either $h \circ \mathcal{G} \neq 0$ (where we extend $\mathcal{G}$ to $n+1$ coordinates such that $\mathcal{G}$ is still a uniform k-independent polynomial map) or that Case 2 of the lemma holds.

Assume that $h \circ \mathcal{G} = 0$. Lemmas 5.16 and 5.18 imply that $\mathrm{ANF}_\Delta(\boldsymbol{x})$ and $h_2\left(\tilde{A_1}^{-1}(\boldsymbol{x})\right)$ have the same set of monomials. From Lemma 5.15 we conclude that Case 2 holds.

□

5.2.2 Proof of Lemma 5.13

Finally, we conclude the proof of Theorem 1.24 by proving Lemma 5.13 that gives a hitting set for the difference of two polynomials in $\mathrm{ANF}_\Delta^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})}$ that, up to constant factors, have the same linear functions on the leaves.

Proof of Lemma 5.13. First, if $f = \alpha g$ for some $\alpha \in \mathbb{F}$, then $f - g \in \mathrm{ANF}^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})}$ and the lemma follows from Theorem 1.22. We therefore assume that f is not a multiple of g, and denote that by $f \not\propto g$.

For any node u in the complete binary tree of depth 2Δ, denote by u_f the polynomial computed at node u in $\mathrm{ANF}_\Delta(\ell_1, \ldots, \ell_n)$, and by u_g the polynomial computed at node u in $\mathrm{ANF}_\Delta(\alpha_1\ell_1, \ldots, \alpha_n\ell_n)$. Fix a node u satisfying $u_f(\boldsymbol{x}) \not\propto u_g(\boldsymbol{x})$, such that u is a deepest node with that property. In particular, each child of

[14] Note that by Lemma 5.17 we may have $|D| = 1$, but we may add some other variable x_ℓ to simplify the notation.

u_f is a multiple of the corresponding child of u_g. Note that, as $f \not\propto g$, such a node u must exist; and by the premise of the lemma, u_f and u_g are not leaves. In addition, u_f and u_g must be addition gates, otherwise we may choose a child u' of u such that $u'_f(\boldsymbol{x}) \not\propto u'_g(\boldsymbol{x})$.

Let $\{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_n\}$ be a dual set to $\{\ell_1, \ldots, \ell_n\}$. Denote $u_f = f_1 f_2 + f_3 f_4$ and $u_g = g_1 g_2 + g_3 g_4$, where the f_is are the grandchildren of u_f and the g_is are the grandchildren of u_g. By choice of u, there exist constants $\alpha, \beta \in \mathbb{F}$ such that $f_1 f_2 = \alpha \cdot g_1 g_2$ and $f_3 f_4 = \beta \cdot g_3 g_4$, and $\alpha \neq \beta$ (otherwise $u_f = \alpha \cdot u_g$). WLOG, assume f_1, g_1 are ancestors of the leaf labeled ℓ_1 (or $\alpha_1 \ell_1$), and f_3, g_3 are ancestors of the leaf labeled ℓ_3 (or $\alpha_3 \ell_3$). By Observation 5.8, there exist polynomials $F(\boldsymbol{x}), G(\boldsymbol{x})$ such that:

$$\begin{aligned} \frac{\partial f}{\partial \boldsymbol{v}_1} &= F(\boldsymbol{x}) f_2(\boldsymbol{x}) \tfrac{\partial f_1}{\partial \boldsymbol{v}_1}(\boldsymbol{x}) \,, & \frac{\partial f}{\partial \boldsymbol{v}_3} &= F(\boldsymbol{x}) f_4(\boldsymbol{x}) \frac{\partial f_3}{\partial \boldsymbol{v}_3}(\boldsymbol{x}) \,, \\ \frac{\partial g}{\partial \boldsymbol{v}_1} &= G(\boldsymbol{x}) g_2(\boldsymbol{x}) \tfrac{\partial g_1}{\partial \boldsymbol{v}_1}(\boldsymbol{x}) \quad \text{and} & \frac{\partial g}{\partial \boldsymbol{v}_3} &= G(\boldsymbol{x}) g_4(\boldsymbol{x}) \frac{\partial g_3}{\partial \boldsymbol{v}_3}(\boldsymbol{x}) \,. \end{aligned}$$

Observe that

$$\frac{\partial (f-g)}{\partial \boldsymbol{v}_1} = F(\boldsymbol{x}) f_2(\boldsymbol{x}) \frac{\partial f_1}{\partial \boldsymbol{v}_1}(\boldsymbol{x}) - G(\boldsymbol{x}) g_2(\boldsymbol{x}) \frac{\partial g_1}{\partial \boldsymbol{v}_1}(\boldsymbol{x}) = (\alpha \cdot F(\boldsymbol{x}) - G(\boldsymbol{x})) g_2(\boldsymbol{x}) \frac{\partial g_1}{\partial \boldsymbol{v}_1}(\boldsymbol{x}) \,, \tag{14}$$

and

$$\frac{\partial (f-g)}{\partial \boldsymbol{v}_3} = F(\boldsymbol{x}) f_4(\boldsymbol{x}) \frac{\partial f_3}{\partial \boldsymbol{v}_3}(\boldsymbol{x}) - G(\boldsymbol{x}) g_4(\boldsymbol{x}) \frac{\partial g_3}{\partial \boldsymbol{v}_3}(\boldsymbol{x}) = (\beta \cdot F(\boldsymbol{x}) - G(\boldsymbol{x})) g_4(\boldsymbol{x}) \frac{\partial g_3}{\partial \boldsymbol{v}_3}(\boldsymbol{x}) \,. \tag{15}$$

Let $\mathcal{G}_1, \mathcal{G}_{2\Delta+1}$ be a 1-independent polynomial map and a $(2\Delta+1)$-independent polynomial map, respectively, such that $\mathcal{G} = \mathcal{G}_1 + \mathcal{G}_{2\Delta+1}$. Theorem 1.22 and Observation 5.8 imply that $\left(g_2(\boldsymbol{x}) \frac{\partial g_1}{\partial \boldsymbol{v}_1}(\boldsymbol{x})\right) \circ \mathcal{G}_{2\Delta+1} \neq 0$, so if $\alpha \cdot F(\mathcal{G}_{2\Delta+1}) \neq G(\mathcal{G}_{2\Delta+1})$ then we get from Equation (14) that $\frac{\partial (f-g)}{\partial \boldsymbol{v}_1} \circ \mathcal{G}_{2\Delta+1} \neq 0$ and thus $(f-g) \circ \mathcal{G} \neq 0$ (using Lemma 3.9). On the other hand, if $\alpha \cdot F(\mathcal{G}_{2\Delta+1}) = G(\mathcal{G}_{2\Delta+1})$, then, since $\alpha \neq \beta$, a similar argument, relying on Equation (15), shows that $\frac{\partial (f-g)}{\partial \boldsymbol{v}_3} \circ \mathcal{G}_{2\Delta+1} \neq 0$ and thus $(f-g) \circ \mathcal{G} \neq 0$, as claimed. □

5.3 Reconstruction for $\mathbf{ANF}^{\mathbf{GL^{aff}}(\mathbb{C})}$

In this section, we argue that the reconstruction algorithm of Gupta et al. [GKQ14], when given oracle access to a polynomial $f \in \mathrm{ANF}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{C})}$, w.h.p. successfully reconstructs an $\mathrm{ANF}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{C})}$ formula computing f. We do so by explaining why the different steps of their algorithm succeed w.h.p. on any input $f \in \mathrm{ANF}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{C})}$. To ease the reading we give their algorithm (`AFR`) and its main subroutine (`LDR`) in the appendix (Algorithms 2 and 3). We remind that their result, with minor changes, can be adapted to any large enough field, see remark 1.27.

Before quoting the original result, we define the distribution on ANF formulas used in [GKQ14]. To this end, we define the *universal* ANF:

Definition 5.19. *Let $\Delta, n \in \mathbb{N}$. Let $\boldsymbol{x} = (x_1, \ldots, x_n)$ and $\boldsymbol{y} = \{y_{i,j} : i = 1, \ldots, 4^\Delta, j = 1, \ldots, n+1\}$ be formal variables. The* universal Δ, n ANF*, denoted $\mathcal{U}_{\Delta,n}(\boldsymbol{x}, \boldsymbol{y})$, is an ANF formula of product depth Δ in which leaf i is labeled $\sum_{j=1}^{n} x_j y_{i,j} + y_{i,n+1}$.*

Trivially, for any ANF formula $f(\boldsymbol{x})$ of product depth Δ on n variables, there exists an assignment $\boldsymbol{v} \in \mathbb{C}^{(n+1)\cdot 4^\Delta}$ to the $\boldsymbol{y}$ variables of $\mathcal{U}_{\Delta,n}(\boldsymbol{x}, \boldsymbol{y})$ such that $f(\boldsymbol{x}) = \mathcal{U}_{\Delta,n}(\boldsymbol{x}, \boldsymbol{v})$. Given the number of variables n, the size $s = 2 \cdot 4^\Delta - 1$ of the ANF we wish to sample, and a finite set of field elements $S \subseteq \mathbb{C}$, we define the distribution $\mathcal{D}(n, s, S)$ on ANF formulas by uniformly sampling an assignment $\boldsymbol{v}$ from $S^{4^\Delta (n+1)}$. This is the distribution used in the main result of [GKQ14]:

Theorem 5.20 (Theorem 1.1 of [GKQ14]). *Let $\mathbb{F}$ be a field of characteristic 0 and S be a finite subset of $\mathbb{F}$. Assume there is a black box holding an ANF formula Φ of size s sampled from $\mathcal{D}(n,s,S)$, and Φ computes a polynomial $f \in \mathbb{F}[x_1,\ldots,x_n]$. There is a randomized algorithm that, given this black box, either outputs an ANF formula Φ' of size $\leq s$ computing f, or outputs* `Fail`. *The algorithm succeeds for a $(1-\frac{n^2\cdot s^{O(1)}}{|S|})$ fraction of the ANF formulas from $\mathcal{D}(n,s,S)$. Moreover, the running time of the algorithm is at most $(ns)^{O(1)}$.*

We note that, although it is not mentioned in their main theorem, the output formula is unique up to $\mathrm{TS}_n(\mathbb{C})$-equivalence, and this fact is stated when needed in intermediate results of [GKQ14] (recall Fact 2.4). We prove Theorem 1.26 by going over the different steps of Algorithm 2. We do not repeat all the arguments and claims of [GKQ14], but rather give high level explanations, referring to theorems, algorithms and tools of [GKQ14].

Sketch of proof of Theorem 1.26. We shall use the following notation in the proof. We wish to reconstruct $f \in \mathrm{ANF}^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{C})}$ that is computed by the ANF formula Φ. We define the homogenization of f, f^h, as usual: $f^h(x_0,\ldots,x_n) = x_0^{\deg(f)} \cdot f(x_1/x_0,\ldots,x_n/x_0)$. Denote by $\boldsymbol{A}$ an $(n+1)\times(n+1)$ matrix of formal variables $a_{i,j}$. For $i \neq j \in \{r+1, r+2,\ldots,n\}$ we denote by $\boldsymbol{A_r^{i,j}}$ the matrix $\boldsymbol{A}$ where all columns except those indexed by $\{0,1,2,\ldots,r\}\cup\{i,j\}$ are set to zero (generic projection matrix to the variables $x_0,x_1,\ldots,x_r,x_i,x_j$). We denote by $A \in \mathbb{C}^{n\times n}$ an assignment to $\boldsymbol{A}$, and likewise $A_r^{i,j}$ would be an assignment to the $n\cdot(r+3)$ variables of $\boldsymbol{A_r^{i,j}}$. Note that $\mathrm{ANF}_\Delta(\boldsymbol{A_r^{i,j}x})$ is a *universal* homogeneous $(r+3)$-variate ANF (in $\{x_0,x_1,\ldots,x_r,x_i,x_j\}$) in the sense that for every $(r+3)$-variate homogeneous ANF $f(x_0,x_1,\ldots,x_r,x_i,x_j)$, of depth 2Δ, there exists an assignment $A_r^{i,j}$ such that $f(\boldsymbol{x}) = \mathrm{ANF}_\Delta(A_r^{i,j}\boldsymbol{x})$. Finally, following [GKQ14], we denote $\sigma_{A_r^{i,j}}(f) \triangleq f^h(A_r^{i,j}\boldsymbol{x})$ (where now we think of $\boldsymbol{x}$ as $\boldsymbol{x} = (x_0,\ldots,x_n)$).

Looking at Algorithm 2, it is clear that except for Step `AFR3`, the rest of the algorithm works without any assumptions on the input ANF. Hence, the proof of correctness boils down to proving that Step `AFR3` works w.h.p.; and more importantly, proving that the `LDR` algorithm (Algorithm 3, the subroutine invoked in Step `AFR3`) succeeds w.h.p. on random projections of *any* $\mathrm{ANF}^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{C})}$ instance. Specifically, we need to prove that for *any* $f \in \mathrm{ANF}^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{C})}$, step `AFR3` succeeds with probability $\geq 1-\frac{|\Phi|^{O(1)}}{|T|}$ on a random linear projection to $r+3 = 128$ variables (see remark A.1) of the *homogenization* of f, f^h (where the coefficients of the projection are sampled from $T \subseteq \mathbb{C}$).

Gupta et al. define two conditions on internal nodes of an ANF $\mathcal{U}_{\Delta,n}(\boldsymbol{x},v)$: *formulaic independence* (FI, see Definition A.5) and *pairwise singular independence* (PSI, see Definition A.7). These conditions are defined in terms of dimensions of certain algebraic varieties $V_1,\ldots,V_k$. In Lemmas 5.10, 5.11, 5.16 and 5.26 of their paper, they show that if every node of Φ satisfies FI, then the `LDR` algorithm correctly reconstructs the polynomial computed at each node of Φ (up to an appropriate group of symmetries). Moreover, part (2) of their Lemma 5.16 shows that when a node u of Φ satisfies FI and PSI, then the polynomials computed at the grandchildren of u are computed up to $\mathrm{TS}_n(\mathbb{C})$ equivalence. Overall, this means that all the quadratic forms are computed correctly up to $\mathrm{TS}_n(\mathbb{C})$-equivalence.

Thus, if the projected polynomials $\sigma_{A_r^{i,j}}(f)$ that we compute in Step `AFR3` satisfy FI and PSI, then the algorithm will correctly reconstruct our $\mathrm{ANF}^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{C})}$ formula.

To prove that (w.h.p.) $\sigma_{A_r^{i,j}}(f)$ satisfies FI and PSI, Gupta et al. prove that these conditions are captured by a set of polynomial equations. Intuitively, this is not a surprising result as FI and PSI are algebraic conditions.

Observation 5.21. *For every $i,j \in \{r+1, r+2,\ldots,n\}$ there exists a set of nonzero polynomials*

$p_1, \ldots, p_k \in \mathbb{C}[\boldsymbol{A_r^{i,j}}]$ *with the property that* $ANF_\Delta(A_r^{i,j}\boldsymbol{x})$ *satisfies FI and PSI if* $A_r^{i,j}$ *is not a point on the variety* $V\left(p_1(\boldsymbol{A_r^{i,j}}), \ldots, p_k(\boldsymbol{A_r^{i,j}})\right) \triangleq \left\{A_r^{i,j} \mid p_1(A_r^{i,j}) = \ldots = p_k(A_r^{i,j}) = 0\right\}$. *Furthermore, the degree of each* p_i *is* $2^{O(\Delta)}$*, which is polynomial in the size of the formula.*

This observation is not stated as is in [GKQ14] but it can be immediately deduced from the proofs of Corollaries 5.31 and 5.32 of [GKQ14].

Thus, we wish to show that a random $A_r^{i,j}$ does not belong to the variety defined in Observation 5.21. For this we follow the same approach as Gupta et al. We prove that *there exist* good projections $A_r^{i,j}$ that do not belong to the variety, and then using DeMillo-Lipton-Schwartz-Zippel lemma we conclude that such a random projection is not on the variety.

Claim 5.22. *Let* $r \geq 125$ *and* $n \geq r$*. For any* n*-variate* $f \in ANF^{GL_n^{aff}(\mathbb{C})}$*, computed by the ANF formula* Φ*, and any* $i, j \in \{r+1, r+2, \ldots, n\}$*, there exists some projection* $A_r^{i,j}$ *such that* $\sigma_{A_r^{i,j}}(f)$ *satisfies FI and PSI at* every internal node *of* Φ*.*

Proof. To prove the existence of a "good" projection for an arbitrary $f \in \mathrm{ANF}^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{C})}$, we use an explicit ANF g, on 128 variables, that can be described as a projection of *any* $f \in \mathrm{ANF}^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{C})}$ (more accurately, of f^h). The definition of g comes from the proof of Lemma 5.30 of [GKQ14]:

$$\begin{aligned}
\forall i,j \in [4]: \quad g_{i,j}(\boldsymbol{x}) \triangleq & (x^e_{32(i-1)+8(j-1)} + x^e_{32(i-1)+8(j-1)+1}) \cdot (x^e_{32(i-1)+8(j-1)+2} + x^e_{32(i-1)+8(j-1)+3}) \\
& + (x^e_{32(i-1)+8(j-1)+4} + x^e_{32(i-1)+8(j-1)+5}) \cdot (x^e_{32(i-1)+8(j-1)+6} + x^e_{32(i-1)+8(j-1)+7}) \\
\forall i \in [4]: \quad g_i(\boldsymbol{x}) \triangleq & g_{i,1}(\boldsymbol{x}) g_{i,2}(\boldsymbol{x}) + g_{i,3}(\boldsymbol{x}) g_{i,4}(\boldsymbol{x}) \qquad (16) \\
g(\boldsymbol{x}) \triangleq & g_1(\boldsymbol{x}) g_2(\boldsymbol{x}) + g_3(\boldsymbol{x}) g_4(\boldsymbol{x}). \qquad (17)
\end{aligned}$$

The exponent $e \in \mathbb{N}$ is chosen such that the degree of g is 2^Δ for the given Δ, i.e. $e = 2^{\Delta-3}$. Gupta et al. prove that g satisfies PSI in Lemma 5.30. In Lemma 5.29, the FI condition is proven to hold for a slightly different polynomial (specifically, they prove g_i as defined in equation (16) satisfies FI), but the proof for formulaic independence of g itself works exactly the same (relies on variable-disjointness of $g_1, \ldots, g_4$), so we get:

Fact 5.23. *The polynomial* g *defined in Equation* (17) *satisfies FI and PSI (and so does* $g(x_{\pi(0)}, \ldots, x_{\pi(127)})$*, for any permutation* π*).*

Let $g(\boldsymbol{x})$ be as defined in equation (17) above. Our goal here is, given an unknown $f \in \mathrm{ANF}^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{C})}$ and indices $i, j \in [n]$, to prove there exists some projection $A_r^{i,j}$ such that $\sigma_{A_r^{i,j}}(f) = g(\boldsymbol{x})$ (possibly up to a permutation of the variables); as we only care about projections up to permutations of the variables, we can WLOG assume $i = r+1, j = r+2$. The correctness of Algorithm 3 is proven for a number of variables ≥ 128 and g is a 128-variate polynomial, so for sake of simplicity we may assume $r = 125$ such that projections of f^h have the same number of variables as g.

For an ANF Ψ computing g such that each leaf is labeled by a single variable from $\{x_1, \ldots, x_{128}\}$ (times some constant), denote by $\tilde{\Psi}$ a new formula constructed as follows: for every $i \in [4^\Delta]$, if leaf number i in Ψ is labeled $\alpha_i \cdot x_j$, relabel it to $\alpha_i \cdot x_j + \ell_i(\boldsymbol{x})$, where ℓ_i is some linear form depending on the variables $x_{129}, \ldots, x_n$. Choose the coefficients of the ℓ_is so that all the leaves of $\tilde{\Psi}$ are linearly independent (thus, $\tilde{\Psi}(\boldsymbol{x}) \in \mathrm{ANF}^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{C})}$). As f^h and $\tilde{\Psi}$ are two polynomials in the $\mathrm{GL}_{n+1}(\mathbb{C})$-orbit of ANF_Δ, there exists some $B \in \mathrm{GL}_{n+1}(\mathbb{C})$ such that $f^h(B\boldsymbol{x}) = \tilde{\Psi}(\boldsymbol{x})$, and by construction $\tilde{\Psi}\big|_{x_{129}=0, x_{130}=0, \ldots, x_n=0}(\boldsymbol{x}) = \Psi(\boldsymbol{x}) = g(\boldsymbol{x})$. By defining $A_r^{i,j}$ to be

the matrix B with columns $129, \ldots, n$ set to zero, we get $\sigma_{A_r^{i,j}}(f) = \tilde{\Psi}\big|_{x_{129}=0, x_{130}=0, \ldots, x_n=0}(\boldsymbol{x}) = g(\boldsymbol{x})$. Since $A_r^{i,j}$ is a projection, this is what we wanted to prove. □

Thus, by applying the DeMillo-Lipton-Schwartz-Zippel lemma, we can conclude that a random projection (sampled from a set $T \subseteq \mathbb{C}$) of the homogenization of any $f \in \mathrm{ANF}^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{C})}$ satisfies FI and PSI with probability at least $\left(1 - \frac{|\Phi|^{O(1)}}{|T|}\right)$, thanks to the upper bound on the degree of the p_is of Observation 5.21. For Step `AFR3` to work, we need all n^2 projections to yield "good" polynomials, and by a simple application of the union bound we deduce that `AFR3` succeeds with probability at least $\left(1 - \frac{n^2 \cdot |\Phi|^{O(1)}}{|T|}\right)$.

This completes the proof of Theorem 1.26 □

Remark 5.24. *The original theorem of [GKQ14] uses two sets of field elements: the set S, used to sample random ANFs from the distribution $\mathcal{D}(n, s, S)$, and the set T, used to sample random projections $A_r^{i,j}$ of the input ANF. As their algorithm works for any $f \in ANF^{GL_n^{aff}(\mathbb{C})}$, we do not need the set S. Thus,we only use T, and we add run-time dependence on $\log(|T|)$ so we can sample the uniform distribution on T.*

6 Dense orbits for ΣΠΣ circuits

In this section we prove our claims regarding dense orbits in ΣΠΣ. We start by proving Theorem 1.31 regarding the relation between $\mathcal{T}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$, $\Sigma\Pi^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$ and ΣΠΣ.

Proof of Theorem 1.31. The claim regarding the closures follows immediately from the fact that every matrix can be approximated by invertible matrices and from the simple observation that for any n-variate polynomial $f(\boldsymbol{x}) \in \Sigma^{[s]}\Pi^{[d]}\Sigma(\mathbb{F})$, there exist $A \in \mathbb{F}^{n \times n}, \boldsymbol{b} \in \mathbb{F}^n$ such that $\mathrm{T}_{s,d}(A\boldsymbol{x} + \boldsymbol{b}) = f(\boldsymbol{x})$.

To prove the separation we first note that the polynomial $f(\boldsymbol{x}) = x_1^2$ is in ΣΠ, but not in $\mathcal{T}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$: if $f(\boldsymbol{x}) \in \mathcal{T}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$, then there exists $(A, \boldsymbol{b}) \in \mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})$ such that $f(A\boldsymbol{x} + \boldsymbol{b}) = \mathrm{T}_{s,d}$, for some s and d (as we compose with invertible affine maps). However, $f(A\boldsymbol{x} + \boldsymbol{b}) = (\ell(\boldsymbol{x}))^2$ for some non-constant linear function $\ell(\boldsymbol{x})$, which is obviously not a multilinear polynomial. The second separation will follow from the next simple claim.

Claim 6.1. *If $f \in \Sigma\Pi^{GL^{aff}(\mathbb{F})}$ is d-homogeneous, then it is in the $GL_n(\mathbb{F})$ orbit of some d-homogeneous ΣΠ circuit (i.e. no affine translation is needed).*

Proof. Let $(A, \boldsymbol{b}) \in \mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})$ and let Ψ be a ΣΠ circuit such that $f(\boldsymbol{x}) = \Psi(A\boldsymbol{x} + \boldsymbol{b})$. Observe that for every i it holds that $\Psi(\boldsymbol{x})^{[i]} \neq 0$ if and only if $\Psi(A\boldsymbol{x})^{[i]} \neq 0$, since A is invertible. In particular, if $\Psi(\boldsymbol{x})$ had a monomial of degree larger than d then the degree of $f(\boldsymbol{x}) = \Psi(A\boldsymbol{x} + \boldsymbol{b})$ would have been larger than d in contradiction. Thus, all gates in Ψ have degree at most d. Similarly, we now see that $f(\boldsymbol{x}) = (\Psi(Ax + \boldsymbol{b}))^{[d]} = (\Psi(\boldsymbol{x}))^{[d]}(A\boldsymbol{x})$. Thus, $\Psi^{[d]}$ is the claimed ΣΠ circuit. □

Let $\sigma_d(\boldsymbol{x})$ be the nth elementary symmetric polynomial. I.e. the sum over all degree-d multilinear monomials in n-variables. Theorem 0 of [NW97] shows that any homogeneous ΣΠΣ circuit computing σ_d must have size $\Omega(\frac{n}{2d})^d$. As any homogeneous polynomial in $\Sigma\Pi^{\mathrm{GL}_n(\mathbb{F})}$ can be computed by a homogeneous ΣΠΣ circuit of the same complexity, we get an exponential lower bound on the sparsity of any $\Sigma\Pi^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$ circuit computing σ_d, over any field. To get an upper bound on the ΣΠΣ complexity, note that, over any field of size $|\mathbb{F}| \geq n + 1$, σ_d has a ΣΠΣ circuit of size $O\left(n^2\right)$ (see [SW01]), that is obtained by interpolating the polynomial $f(Y) = \prod_{i=1}^{n}(Y + x_i)$. □

We devote the rest of this section to proving Theorems 1.32, 1.34 and 1.35.

6.1 A hitting-set generator for $\Sigma\Pi^{\mathbf{GL^{aff}}(\mathbb{F})}$ circuits

In this section, we prove Theorem 1.32. The main idea is that given some $f \in \Sigma\Pi^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$, where $f(\boldsymbol{x}) = g(A\boldsymbol{x}+\boldsymbol{b}) = g(\ell_1(\boldsymbol{x}),\ldots,\ell_n(\boldsymbol{x}))$ for an s-sparse polynomial g, composing f with a 1-independent polynomial map allows us to "halve" the number of monomials appearing in the underlying $\Sigma\Pi$ circuit $g(x)$. Depending on the structure of g, this can be done by either taking a derivative of f at the direction of an appropriately chosen dual vector, or by restricting f to a linear subspace in which some $\ell_i(\boldsymbol{x}) = 0$ and other linear functions remain linearly independent. By Lemmas 3.9 and 3.10, both tasks can be simulated using a 1-independent generator.

As a reminder, we restate Theorem 1.32 before giving its proof.

Theorem 1.32. *Let $0 \neq g \in \mathbb{F}[\boldsymbol{x}]$ have sparsity $\leq 2^t$. Let $(A,\boldsymbol{b}) \in GL_n^{aff}(\mathbb{F})$, and $f(\boldsymbol{x}) = g(A\boldsymbol{x}+\boldsymbol{b})$. Then, for any $(t+1)$-independent polynomial map $\mathcal{G}$, $f \circ \mathcal{G} \neq 0$.*

Proof. By induction on t. For $t = 0$, $0 \neq f(\boldsymbol{x})$ is either a non-zero constant, or a product of non-zero linear functions. A non-zero linear function composed with a 1-independent polynomial map $\mathcal{G}$ is non-zero because the n entries of G are linearly independent (Observation 3.1(2)), so $f \circ \mathcal{G} \neq 0$.

Let $t > 0$ and let $\mathcal{G}_1(\boldsymbol{y_1}, z_1)$ and $\mathcal{G}_t(\boldsymbol{y_2}, z_2, \ldots, z_{t+1})$ be a 1-independent polynomial map and a t-independent polynomial map, respectively, such that $\mathcal{G} = \mathcal{G}_1 + \mathcal{G}_t$. Let $\ell_1,\ldots,\ell_n$ be linear functions such that the ith coordinate of $A\boldsymbol{x}$ is $\ell_i(\boldsymbol{x})$, and let $\boldsymbol{b} = (b_1, ..., b_n)$.

First, we note that WLOG we can assume that no variable x_i divides g; otherwise we can take some $\tilde{g} \in \mathbb{F}[\boldsymbol{x}]$ such that $g(\boldsymbol{x}) = x_i^k \tilde{g}(\boldsymbol{x})$, x_i does not divide $\tilde{g}$ and both g and $\tilde{g}$ have the same sparsity. By the base case (sparsity 1), $(\ell_i(\boldsymbol{x}) + b_i)^k \circ \mathcal{G} \neq 0$, so $f \circ \mathcal{G} \neq 0$ if and only if $(\tilde{g}(A\boldsymbol{x}+\boldsymbol{b})) \circ \mathcal{G} \neq 0$.

Now that we know $g(\boldsymbol{x})$ is not divisible by any variable, we consider two cases:

Case 1: There exists a variable $x_i \in \mathrm{var}(g)$ that appears in $\leq 2^{t-1}$ monomials of $g(\boldsymbol{x})$. Choose $\boldsymbol{v} \in \mathbb{F}^n$ such that $\ell_i(\boldsymbol{v}) = 1$, and for all $j \neq i$, $\ell_j(\boldsymbol{v}) = 0$. By Lemma 3.8, $\frac{\partial f}{\partial \boldsymbol{v}}(\boldsymbol{x}) = \left(\frac{\partial g}{\partial x_i}\right)(A\boldsymbol{x}+\boldsymbol{b})$. By choice of x_i, $\frac{\partial g}{\partial x_i}$ is non-zero and of sparsity $\leq 2^{t-1}$, so by induction: $\left(\frac{\partial f}{\partial \boldsymbol{v}}\right)(\mathcal{G}_t) \neq 0$. Lemma 3.9 implies that $f \circ \mathcal{G} = f \circ (\mathcal{G}_1 + \mathcal{G}_t) \neq 0$.

Case 2: Every variable $x_i \in \mathrm{var}(g)$ appears in at least 2^{t-1} monomials of g. Assume, WLOG, that $x_1 \in \mathrm{var}(g)$, and define $\tilde{g}(\boldsymbol{x}) \triangleq g(0, x_2, x_3, \ldots, x_n)$. As x_1 does not divide g, $\tilde{g} \neq 0$ and is of sparsity $\leq 2^{t-1}$. By Lemma 3.10, there exist linearly independent linear functions $\tilde{\ell}_2,\ldots,\tilde{\ell}_n$, an assignment $\boldsymbol{\alpha} \in \mathbb{F}^{|\boldsymbol{y_1}|}$ and some linear function $L(\boldsymbol{x})$ such that $f\left(\boldsymbol{x} + \mathcal{G}_1\left(\boldsymbol{\alpha}, L(\boldsymbol{x})\right)\right) = \tilde{g}\left(\ell_2(\boldsymbol{x}),\ldots,\ell_n(\boldsymbol{x})\right) \neq 0$. As $\tilde{g}$ is non-zero and has sparsity $\leq 2^{t-1}$, we get from the induction hypothesis that $f\left(\boldsymbol{x} + \mathcal{G}_1\left(\boldsymbol{\alpha}, L(\boldsymbol{x})\right)\right) \circ \mathcal{G}_t \neq 0$, and therefore $f\left(\boldsymbol{x} + \mathcal{G}_1\left(\boldsymbol{y_1}, z_1\right)\right) \circ \mathcal{G}_t \neq 0$. Hence, $f \circ \mathcal{G} = f \circ (\mathcal{G}_t + \mathcal{G}_1) = f\left(\boldsymbol{x} + \mathcal{G}_1\left(\boldsymbol{y_1}, z_1\right)\right) \circ \mathcal{G}_t \neq 0$. □

Corollary 1.33 follows immediately from Theorem 1.32 and Observation 1.14.

6.2 An interpolating set generator for $\mathcal{T}^{\mathbf{GL^{aff}}(\mathbb{F})}$

To construct an interpolating set generator for $\mathcal{T}^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})} \triangleq \mathcal{T}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})} \cap \mathbb{F}[x_1,\ldots,x_n]$ we need a generator that hits the difference of two polynomials of $\mathcal{T}^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})}$. As this class is closed under multiplication by scalars, such a generator hits every nonzero sum of two $\mathcal{T}^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})}$ polynomials. The main idea can be described as follows: the tensor $\mathrm{T}_{s,d}$ on variables $\{x_{1,1},\ldots,x_{s,d}\}$ has the property that for any two variables in distinct

product gates, $x_{i,j}$ and $x_{i',j'}$ ($i \neq i'$), it holds that $\frac{\partial^2 \mathrm{T}_{s,d}}{\partial x_{i,j} \partial x_{i',j'}} = 0$. We prove that for a sum of distinct $\mathcal{T}^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})}$ polynomials, there is always a pair of "dual" vectors such that if we take a derivative in their direction then one of the $\mathcal{T}^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})}$ polynomials of the sum vanishes. Once we prove this, all that is left is to hit the remaining polynomial (or actually, its derivative).

If $f \in \mathcal{T}^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})}$ and $\boldsymbol{v} \in \mathbb{F}^n$ is arbitrary, then $\frac{\partial f}{\partial \boldsymbol{v}}$ need not be in $\mathcal{T}^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})}$. We thus begin by constructing a hitting set generator for directional derivatives of $\mathcal{T}^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})}$ polynomials.

Lemma 6.2. *Let $f \in \mathcal{T}^{GL_n^{aff}(\mathbb{F})}$, $k \in \mathbb{N}$ and $\boldsymbol{v}_1, \ldots, \boldsymbol{v}_k \in \mathbb{F}^n$. Then, for any $(k+2)$-independent polynomial map $\mathcal{G}$:*

$$\frac{\partial^k f}{\partial \boldsymbol{v}_1 \partial \boldsymbol{v}_2 \cdots \partial \boldsymbol{v}_k} \neq 0 \Rightarrow \frac{\partial^k f}{\partial \boldsymbol{v}_1 \partial \boldsymbol{v}_2 \cdots \partial \boldsymbol{v}_k} \circ \mathcal{G} \neq 0 \,.$$

Proof. Let $\mathcal{G}_1^{(1)}, \mathcal{G}_1^{(2)}, \mathcal{G}_k$ be a pair of 1-independent polynomial maps and a k-independent polynomial map, respectively, such that $\mathcal{G} = \mathcal{G}_1^{(1)} + \mathcal{G}_1^{(2)} + \mathcal{G}_k$. Let $\{\ell_{1,1}, \ldots, \ell_{s,d}\}$ be linearly independent linear functions such that $f(\boldsymbol{x}) = \sum_{i=1}^{s} \prod_{j=1}^{d} \ell_{i,j}$. Let $\{\boldsymbol{u}_{i,j}\}$ be a dual set to $\left\{\ell_{i,j}^{[1]}\right\}$. I.e., $\ell_{i,j}^{[1]}(\boldsymbol{u}_{i',j'}) = \delta_{i,i'} \cdot \delta_{j,j'}$.

Set $g(\boldsymbol{x}) \triangleq \frac{\partial^k f}{\partial \boldsymbol{v}_1 \partial \boldsymbol{v}_2 \cdots \partial \boldsymbol{v}_k}(\boldsymbol{x})$. For every i, let $Q_i(w_{i,1}, \ldots, w_{i,d})$ be a polynomial satisfying $Q_i(\ell_{i,1}(\boldsymbol{x}), \ldots, \ell_{i,d}(\boldsymbol{x})) = \frac{\partial^k (\prod_{j=1}^{d} \ell_{i,j}(\boldsymbol{x}))}{\partial \boldsymbol{v}_1 \partial \boldsymbol{v}_2 \cdots \partial \boldsymbol{v}_k}$. In particular, $g(\boldsymbol{x}) = \sum_{i=1}^{s} Q_i(\ell_{i,1}, \ldots, \ell_{i,d})$. Fix some $i \in [s]$ such that Q_i is non-constant (if no such i exists, then g is a non-zero constant and thus $g \circ \mathcal{G} \neq 0$). Assume, WLOG, that Q_i depends non-trivially on $w_{i,1}$ and consider the derivative in direction $\boldsymbol{u}_{i,1}$. From Lemma 3.8 We get

$$\frac{\partial Q_i(\ell_{i,1}(\boldsymbol{x}), \ldots, \ell_{i,d}(\boldsymbol{x}))}{\partial \boldsymbol{u}_{i,1}} = \frac{\partial Q_i}{\partial w_{i,1}} \left(\ell_{i,1}(\boldsymbol{x}), \ldots, \ell_{i,d}(\boldsymbol{x})\right) \neq 0 \,,$$

and for $i' \neq i$

$$\frac{\partial Q_{i'}(\ell_{i,1}(\boldsymbol{x}), \ldots, \ell_{i,d}(\boldsymbol{x}))}{\partial \boldsymbol{u}_{i,1}} = \frac{\partial Q_{i'}}{\partial w_{i,1}} (\ell_{i,1}(\boldsymbol{x}), \ldots, \ell_{i,d}(\boldsymbol{x})) = 0 \,.$$

Thus

$$\frac{\partial g}{\partial \boldsymbol{u}_{i,1}} = \frac{\partial Q_i}{\partial w_{i,1}} (\ell_{i,1}(\boldsymbol{x}), \ldots, \ell_{i,d}(\boldsymbol{x})) \neq 0 \,.$$

As $Q_i(\ell_{i,1}(\boldsymbol{x}), \ldots, \ell_{i,d}(\boldsymbol{x}))$ is a kth order directional derivative of the product $\ell_{i,1}(\boldsymbol{x}) \cdots \ell_{i,d}(\boldsymbol{x})$ we have that

$$Q_i(\ell_{i,1}(\boldsymbol{x}), \ldots, \ell_{i,d}(\boldsymbol{x})) = \sum_{\substack{S \subseteq [d] \\ |S|=k}} \alpha_S \left(\prod_{j \in [d] \setminus S} \ell_{i,j}(\boldsymbol{x}) \right) \,,$$

for some constants $\alpha_S \in \mathbb{F}$. Thus,

$$\frac{\partial g}{\partial \boldsymbol{u}_{i,1}} = \sum_{\substack{S \subseteq \{2,\ldots,d\} \\ |S|=k}} \alpha_S \left(\prod_{j \in \{2,\ldots,d\} \setminus S} \ell_{i,j}(\boldsymbol{x}) \right) \,.$$

Assume, WLOG, that for $T = \{2, \ldots, k+1\}$, $\alpha_T \neq 0$. Observe that except for the term $\alpha_T \left(\prod_{j \in \{2,\ldots,d\} \setminus T} \ell_{i,j}(\boldsymbol{x}) \right)$, every other term is divisible by one of the functions $\ell_{i,j}$, for $j \in T$. Let $V = \{\boldsymbol{v} \mid \ell_{i,j}(\boldsymbol{v}) = 0, \ \forall j \in T\}$. It follows that $\frac{\partial g}{\partial \boldsymbol{u}_{i,1}}\Big|_V = \alpha_T \left(\prod_{j \in \{2,\ldots,d\} \setminus T} \ell_{i,j}(\boldsymbol{x}) \right)\Big|_V \neq 0$. Lemma 3.10 implies that there exist linear functions

$L_1(\boldsymbol{x}),\ldots,L_k(\boldsymbol{x})$ and an assignment $\boldsymbol{\beta}$ such that for $\boldsymbol{L}=(L_1,\ldots,L_k)$:

$$\frac{\partial g}{\partial \boldsymbol{u}_{i,1}}\left(\boldsymbol{x}+\mathcal{G}_k(\boldsymbol{\beta},\boldsymbol{L}(\boldsymbol{x}))\right)=\alpha_T\left(\prod_{j\in\{2,\ldots,d\}\setminus T}\ell_{i,j}\left(\boldsymbol{x}+\mathcal{G}_k(\boldsymbol{\beta},\boldsymbol{L}(\boldsymbol{x}))\right)\right)\neq 0\,.$$

As the right term is a product of linear functions, we get from Observation 3.1(2) that

$$\frac{\partial g}{\partial \boldsymbol{u}_{i,1}}\left(\boldsymbol{x}+\mathcal{G}_k(\boldsymbol{\beta},\boldsymbol{L}(\boldsymbol{x}))\right)\circ\mathcal{G}_1^{(2)}\neq 0\,.$$

Therefore, $\frac{\partial g}{\partial \boldsymbol{u}_{i,1}}\circ(\mathcal{G}_1^{(2)}+\mathcal{G}_k)\neq 0$. The claim now follows from Lemma 3.9. □

It is not hard to see that the proof above implies the following hitting set generator for $\mathcal{T}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$:

Corollary 6.3. *If $0\neq f\in\mathcal{T}^{GL_n^{aff}(\mathbb{F})}$, then for any 2-independent polynomial map $\mathcal{G}$: $f\circ\mathcal{G}\neq 0$.*

We are now prepared to a construct hitting set generator for $\mathcal{T}^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})}+\mathcal{T}^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})}$. We recall the statement of Theorem 1.34.

Theorem 1.34. *Let $n,s_1,s_2,d_1,d_2\in\mathbb{N}$ be such that $n\geq s_1\cdot d_1,s_2\cdot d_2$. For $i\in\{1,2\}$ let $f_i\in T_{s_i,d_i}^{GL_n(\mathbb{F})}$, and let $f=f_1-f_2$. If $f\neq 0$, then any uniform 6-independent polynomial map $\mathcal{G}$ satisfies $f\circ\mathcal{G}\neq 0$.*

Proof. Let $\mathcal{G}_6$ be a uniform 6-independent polynomial map and let $\left\{\ell_{i,j,k}\right\}$ be linear functions such that $f_i=\mathrm{T}_{s_i,d_i}\left(\ell_{i,1,1},\ldots,\ell_{i,s_i,d_i}\right)$.

We first prove that if $f\circ\mathcal{G}_6=0$ then $d_1=d_2$. Assume for a contradiction that $d_1>d_2$. Observe that $f_1^{[d_1]}=\mathrm{T}_{s_1,d_1}\left(\ell_{1,1,1}^{[1]},\ldots,\ell_{1,s_1,d_1}^{[1]}\right)$ (recall that $\ell^{[1]}$ is the degree 1 homogeneous part of ℓ). As the $\ell_{1,i,j}^{[1]}$s are linearly independent, it follows that $f_1^{[d_1]}\neq 0$. Corollary 6.3 implies that $f_1^{[d_1]}\circ\mathcal{G}_6\neq 0$, and as $\mathcal{G}_6$ is uniform, we get that $\deg\left(f_1\circ\mathcal{G}_6\right)=d_1\cdot\deg\left(\mathcal{G}_6\right)$. On the other hand, $\deg\left(f_2\circ\mathcal{G}_6\right)\leq d_2\cdot\deg\left(\mathcal{G}_6\right)<\deg\left(f_1\circ\mathcal{G}_6\right)$. It follows that $f\circ\mathcal{G}_6\neq 0$, in contradiction. From now on we denote $d=d_1=d_2$.

Next, we note that we can assume that f is homogeneous. Let $\tilde{\ell}_{i,j,k}=x_0\cdot\ell_{i,j,k}(\boldsymbol{x}/x_0)$ be the homogenization of $\ell_{i,j,k}$. Observe that the homogenization of f is $\tilde{f}(x_0,\boldsymbol{x})\triangleq x_0^d f(\boldsymbol{x}/x_0)=\mathrm{T}_{s_1,d}\left(\tilde{\ell}_{1,1,1},\ldots,\tilde{\ell}_{1,s_1,d_1}\right)+\mathrm{T}_{s_2,d}\left(\tilde{\ell}_{2,1,1},\ldots,\tilde{\ell}_{2,s_2,d_2}\right)$, which is a homogeneous polynomial in $\mathcal{T}^{\mathrm{GL}_{n+1}(\mathbb{F})}+\mathcal{T}^{\mathrm{GL}_{n+1}(\mathbb{F})}$. By Lemma 5.14, it is enough to prove that $\tilde{f}\circ\mathcal{G}_6'\neq 0$, where $\mathcal{G}_6'$ is a uniform 6-independent map into $\mathbb{F}^{n+1}$. Hence, to simplify notation and WLOG, we assume from now on that f is homogeneous and that $\ell_{i,j,k}=\ell_{i,j,k}^{[1]}$. Next, we handle the case $s_1\neq s_2$.

Assume, WLOG, that $s_1>s_2$. As the $s_1\cdot d$ linear functions $\{\ell_{1,i,j}\}_{i,j}$ are linearly independent, there must exist a linear form, WLOG, $\ell_{1,1,1}$, such that $\ell_{1,1,1}\notin\mathrm{span}\left(\{\ell_{2,i,j}\}_{i,j}\right)$. As before, fix a vector $\boldsymbol{v}$ such that $\ell_{1,1,1}(\boldsymbol{v})=1$ and $\ell_{2,i,j}(\boldsymbol{v})=0$ for all $i,j\in[s_2]\times[d]$. Lemma 3.8 implies that $\frac{\partial f_2}{\partial\boldsymbol{v}}=0$. On the other hand, from linear independence we get that $\frac{\partial(\prod_{j=1}^d\ell_{1,1,j})}{\partial\boldsymbol{v}}\neq 0$ and, the same argument also gives $\frac{\partial f_1}{\partial\boldsymbol{v}}\neq 0$. Thus $\frac{\partial f}{\partial\boldsymbol{v}}\neq 0$. From Lemmas 6.2 and 3.9 we conclude that any uniform 4-independent polynomial map hits f. Observe that the proof above also shows that it must be the case that $\mathrm{span}\left(\{\ell_{1,i,j}\}_{i,j}\right)=\mathrm{span}\left(\{\ell_{2,i,j}\}_{i,j}\right)$, or else any uniform 4-independent polynomial map hits f.

From this point on, we assume that $s_1=s_2=s$ and that $\mathrm{span}\left(\{\ell_{1,i,j}\}_{i,j}\right)=\mathrm{span}\left(\{\ell_{2,i,j}\}_{i,j}\right)$.

As $\mathrm{span}\left(\{\ell_{1,i,j}\}_{i,j}\right)=\mathrm{span}\left(\{\ell_{2,i,j}\}_{i,j}\right)$, we can represent f_2 as a polynomial in $\{\ell_{1,i,j}\}_{i,j}$ (recall this notion from Section 2.1). We split the proof into two cases, depending on the $\{\ell_{1,i,j}\}_{i,j}$-monomials appearing in f_2:

1. The set of $\{\ell_{1,i,j}\}_{i,j}$-monomials appearing in f_2 is a subset of the $\{\ell_{1,i,j}\}_{i,j}$-monomials in f_1. I.e., $f_2(\boldsymbol{x}) = \sum_{i=1}^{s} \alpha_i \cdot \prod_{j=1}^{d} \ell_{1,i,j}$. This means that $f = \sum_{i=1}^{s}(1+\alpha_i) \cdot \prod_{j=1}^{d} \ell_{1,i,j} \in \mathrm{T}_{s,d}^{\mathrm{GL}_n(\mathbb{F})}$, and the theorem follows from Corollary 6.3.

2. There exists an $\{\ell_{1,i,j}\}_{i,j}$-monomial $\prod_{i,j} \ell_{i,j}^{a_{i,j}}$ in f_2 that is not an $\{\ell_{1,i,j}\}_{i,j}$-monomial of f_1. Let $\{\boldsymbol{v}_{i,j}\}$ be a dual set to $\{\ell_{1,i,j}\}$. We proceed to show we can choose two vectors $\boldsymbol{u}, \boldsymbol{w} \in \{\boldsymbol{v}_{1,1}, \ldots, \boldsymbol{v}_{s,d}\}$ such that $\frac{\partial^2 f_1}{\partial \boldsymbol{u} \partial \boldsymbol{w}} = 0$ and $\frac{\partial^2 f_2}{\partial \boldsymbol{u} \partial \boldsymbol{w}} \neq 0$. We again consider two cases:

   - There exists some $a_{i,j} \geq 2$: Let $\boldsymbol{u} = \boldsymbol{w} = \boldsymbol{v}_{i,j}$. By Lemma 3.8:

     $$\frac{\partial^2 f_1}{\partial \boldsymbol{u} \partial \boldsymbol{w}}(\boldsymbol{x}) = \frac{\partial \mathrm{T}_{s,d}}{\partial^2 x_{i,j}}(\ell_{1,1,1}, \ldots, \ell_{1,s,d}) = 0$$

     and

     $$\frac{\partial^2 f_2}{\partial \boldsymbol{u} \partial \boldsymbol{w}}(\boldsymbol{x}) \neq 0 \,,$$

     as the $\{\ell_{1,i,j}\}$-monomial $\prod_{i,j} \ell_{i,j}^{a_{i,j}}$ exists in f_2.
   - $a_{i,j} \leq 1$ for every i, j: In this case, since f_2 is homogeneous, there must be some $i \neq i'$ such that for some j and j', $a_{i,j}, a_{i',j'} \neq 0$. Now choose $\boldsymbol{u} = \boldsymbol{v}_{i,j}$ and $\boldsymbol{w} = \boldsymbol{v}_{i',j'}$. As before, it is easy to verify that

     $$\frac{\partial^2 f_1}{\partial \boldsymbol{u} \partial \boldsymbol{w}}(\boldsymbol{x}) = 0 \quad \text{and} \quad \frac{\partial^2 f_2}{\partial \boldsymbol{u} \partial \boldsymbol{w}}(\boldsymbol{x}) \neq 0 \,.$$

   Thus, in either cases, there exist $\boldsymbol{u}, \boldsymbol{w}$ such that

   $$\frac{\partial^2 f}{\partial \boldsymbol{u} \partial \boldsymbol{w}} = \frac{\partial^2 f_2}{\partial \boldsymbol{u} \partial \boldsymbol{w}} \neq 0.$$

   By Lemma 6.2, any 4-independent polynomial map hits $\frac{\partial^2 f}{\partial \boldsymbol{u} \partial \boldsymbol{w}}$; so by Lemma 3.9, any uniform 6-independent polynomial map hits f.

□

6.3 Reconstruction of $\mathcal{T}^{\mathbf{GL^{aff}}(\mathbb{C})}$ circuits

In [KS19a], Kayal and Saha gave a polynomial-time, randomized reconstruction algorithm that, given black-box access to a homogeneous $\Sigma\Pi\Sigma$ circuits satisfying a *non-degeneracy* condition (Definition 6.5), reconstructs the circuit with high probability. To prove Theorem 1.35 all we have to do is show that any homogeneous polynomial $f \in \mathcal{T}^{\mathrm{GL}^{\mathrm{aff}}(\mathbb{F})}$ satisfies the non-degeneracy condition of Definition 6.5.

To explain the condition we first need to define the *partial derivative* space of a polynomial:

Definition 6.4. *For an n-variate polynomial $f(\boldsymbol{x}) \in \mathbb{F}[\boldsymbol{x}]$, of degree d, and for any $k \in [d]$, the* partial derivative space of order k *of f (PD_k space for short), denoted $\partial^k f$, is the $\mathbb{F}$-span of all partial derivatives of f of order k:*

$$\partial^k f = \mathrm{span}_{\mathbb{F}} \left\{ \frac{\partial^k f}{\partial x_{i_1} \partial x_{i_2} \cdots \partial x_{i_k}} : i_1, \ldots, i_k \in [n] \right\} .$$

Definition 6.5 (Non-degeneracy condition [KS19a])**.** *Let $f(\boldsymbol{x}) = f_1(\boldsymbol{x}) + \ldots + f_s(\boldsymbol{x})$, where $f_i = \prod_{j=1}^{d} \ell_{i,j}$ for some linear forms $\ell_{i,j}$, be an n-variate d-homogeneous polynomial, which can be computed by a depth-3*

circuit of top fan-in s. Fix $k \triangleq \left\lceil \frac{\log(s)}{\log(\frac{n}{\mathsf{e}\cdot d})} \right\rceil$, where e is the base of the natural logarithm. We say $f(\boldsymbol{x})$ is non-degenerate *if $\dim(\partial^k f) = s\cdot\binom{d}{k}$, and for every $i \in [s]$ there exist $2k+1$ linear forms $\ell_{i,r_1},\ldots,\ell_{i,r_{2k+1}}$ such that:*

$$\dim\left(\partial^k\left(\sum_{j\in[s]\setminus\{i\}} f_j\right) \bmod \operatorname{span}_{\mathbb{C}}\{\ell_{i,r_1},\ldots,\ell_{i,r_{2k+1}}\}\right) = (s-1)\cdot\binom{d}{k}$$

Theorem 6.6 (Theorem 1 of [KS19a])**.** *Let $n,d,s \in \mathbb{N}$, $n \geq (3d)^2$ and $s \leq (\frac{n}{3d})^{\frac{d}{3}}$. Let $\mathbb{F}$ be a field of characteristic zero or greater than ds^2.*[15] *There is a randomized, poly(n,d,s) = poly(n,s) time algorithm which takes as input black-box access to an n-variate d-homogeneous polynomial f that can be computed by a non-degenerate (Definition 6.5) $\Sigma\Pi\Sigma$ circuit of top fan-in s, and outputs a non-degenerate, n-variate, d-homogeneous $\Sigma\Pi\Sigma$ circuit of top fan-in s computing f.*

For our proof we will need the following simple fact.

Fact 6.7. *Let $f(\boldsymbol{x})$ be a polynomial of degree d and $(A,\boldsymbol{b}) \in GL_n^{aff}(\mathbb{F})$. Then, for any $k \in [d]$:*

$$\partial^k f(A\boldsymbol{x}+\boldsymbol{b}) = \left\{g(A\boldsymbol{x}+\boldsymbol{b}) : g \in \partial^k f(\boldsymbol{x})\right\} .$$

Proof of Theorem 1.35. As given a non-homogeneous $\mathcal{T}^{\mathrm{GL}_n^{\mathrm{aff}}(\mathbb{F})}$ circuit we can easily get query access to its homogenization, $f^h = x_0^d f(\frac{x_1}{x_0},\ldots,\frac{x_n}{x_0})$, which is a homogeneous polynomial in $\mathcal{T}^{\mathrm{GL}_{n+1}(\mathbb{F})}$, we can assume WLOG that the black-box polynomial is homogeneous. It should also be clear that a polynomial satisfies the condition in Definition 6.5 if and only if its homogenization does.

It is clear that $\dim\left(\partial^k \mathrm{T}_{s,d}\right) = s\binom{d}{k}$, and since composing with an invertible linear transformation does not affect the dimension of the PD_k space (Fact 6.7), it follows that $\dim\left(\partial^k f\right) = s\binom{d}{k}$ for *any* d-homogeneous, s-sparse $f \in \mathcal{T}^{\mathrm{GL}_n(\mathbb{F})}$. It is also clear that $\mathrm{T}_{s,d}$ satisfies the second condition and that this condition too is invariant under invertible linear transformations.

We still need to argue that the output of the algorithm of Theorem 6.6 is a $\mathcal{T}^{\mathrm{GL}(\mathbb{F})}$ circuit. Theorem 6.6 guarantees that the output circuit $\Phi = \sum_1^s \prod_1^d \ell_{i,j}$ is a non-degenerate d-homogeneous, $\Sigma\Pi\Sigma$ circuit computing f. We claim the linear forms $\ell_{i,j}$ on the leaves are linearly independent, and conclude that it is indeed a $\mathcal{T}^{\mathrm{GL}(\mathbb{F})}$ circuit. Indeed, as $f(\boldsymbol{x})$ is $\mathrm{GL}_n(\mathbb{F})$-equivalent to $\mathrm{T}_{s,d}(\boldsymbol{x})$ and $\partial^{d-1}\mathrm{T}_{s,d}(\boldsymbol{x}) = \operatorname{span}_{\mathbb{F}}\{x_{1,1},\ldots,x_{s,d}\}$, it follows that $\partial^{d-1}\Phi$ has dimension $s\cdot d$. The space $\partial^{d-1}\Phi$ is contained in $\operatorname{span}_{\mathbb{F}}\{\ell_{1,1},\ldots,\ell_{s,d}\}$, so by dimension argument the set $\{\ell_{1,1},\ldots,\ell_{s,d}\}$ must be linearly independent.

Finally, we note that by Lemma 2.7 the representation that was found is unique up to $\mathrm{TPS}_{s,d}(\mathbb{F})$-equivalence. This concludes the proof of Theorem 1.35. □

References

[AFS+18] Matthew Anderson, Michael A. Forbes, Ramprasad Saptharishi, Amir Shpilka, and Ben Lee Volk. Identity testing and lower bounds for read-k oblivious algebraic branching programs. *ACM Trans. Comput. Theory*, 10(1):3:1–3:30, 2018.

[AKS02] Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. Primes is in p. *Ann. of Math*, 2:781–793, 2002.

[15] This requirement appears before the statement of their theorem.

[Ald84] A. Alder. *Grenzrang und Grenzkomplexität aus algebraischer und topologischer Sicht*. PhD thesis, Universität Zürich, Philosophische Fakultät II, 1984.

[ASSS16] Manindra Agrawal, Chandan Saha, Ramprasad Saptharishi, and Nitin Saxena. Jacobian hits circuits: Hitting sets, lower bounds for depth-d occur-k formulas and depth-3 transcendence degree-k circuits. *SIAM J. Comput.*, 45(4):1533–1562, 2016.

[AvMV15] Matthew Anderson, Dieter van Melkebeek, and Ilya Volkovich. Deterministic polynomial identity tests for multilinear bounded-read formulae. *Computational Complexity*, 24(4):695–776, 2015.

[AW16] Eric Allender and Fengming Wang. On the power of algebraic branching programs of width two. *Computational Complexity*, 25(1):217–253, 2016.

[BB98] Daoud Bshouty and Nader H. Bshouty. On interpolating arithmetic read-once formulas with exponentiation. *J. Comput. Syst. Sci.*, 56(1):112–124, 1998.

[BBB+00] Amos Beimel, Francesco Bergadano, Nader H. Bshouty, Eyal Kushilevitz, and Stefano Varricchio. Learning functions represented as multiplicity automata. *J. ACM*, 47(3):506–530, 2000.

[BC92] Michael Ben-Or and Richard Cleve. Computing algebraic formulas using a constant number of registers. *SIAM J. Comput.*, 21(1):54–58, 1992.

[BCRL79] Dario Bini, Milvio Capovani, Francesco Romani, and Grazia Lotti. $O(n^{2.7799})$ complexity for $n \times n$ approximate matrix multiplication. *Information Processing Letters*, 8(5):234 – 235, 1979.

[BCS13] Peter Bürgisser, Michael Clausen, and Mohammad A Shokrollahi. *Algebraic complexity theory*, volume 315. Springer Science & Business Media, 2013.

[BHH95] Nader H. Bshouty, Thomas R. Hancock, and Lisa Hellerstein. Learning arithmetic read-once formulas. *SIAM J. Comput.*, 24(4):706–735, 1995.

[BI19] Markus Bläser and Christian Ikenmeyer. Introduction to geometric complexity theory. `https://pcwww.liv.ac.uk/~iken/teaching_sb/summer17/introtogct/gct.pdf`, 2019.

[BIZ18] Karl Bringmann, Christian Ikenmeyer, and Jeroen Zuiddam. On algebraic branching programs of small width. *J. ACM*, 65(5):32:1–32:29, 2018.

[BSV20] Vishwas Bhargava, Shubhangi Saraf, and Ilya Volkovich. Reconstruction of depth-4 multilinear circuits. In Shuchi Chawla, editor, *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 2144–2160. SIAM, 2020.

[BT88] Michael Ben-Or and Prasoon Tiwari. A deterministic algorithm for sparse multivariate polynomial interpolation (extended abstract). In Janos Simon, editor, *Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA*, pages 301–309. ACM, 1988.

[Bür04] Peter Bürgisser. The complexity of factors of multivariate polynomials. *Found. Comput. Math.*, 4(4):369–396, 2004.

[CKS18] Chi-Ning Chou, Mrinal Kumar, and Noam Solomon. Hardness vs randomness for bounded depth arithmetic circuits. In Rocco A. Servedio, editor, *33rd Computational Complexity Conference, CCC 2018, June 22-24, 2018, San Diego, CA, USA*, volume 102 of *LIPIcs*, pages 13:1–13:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

[dOSV16] Rafael Mendes de Oliveira, Amir Shpilka, and Ben lee Volk. Subexponential size hitting sets for bounded depth multilinear formulas. *Computational Complexity*, 25(2):455–505, 2016.

[DS07] Zeev Dvir and Amir Shpilka. Locally decodable codes with two queries and polynomial identity testing for depth 3 circuits. *SIAM Journal on Computing*, 36(5):1404–1434, 2007.

[DSY09] Zeev Dvir, Amir Shpilka, and Amir Yehudayoff. Hardness-Randomness Tradeoffs for Bounded Depth Arithmetic Circuits. *SIAM J. Comput.*, 39(4):1279–1293, 2009.

[FGT19] Stephen A. Fenner, Rohit Gurjar, and Thomas Thierauf. A deterministic parallel algorithm for bipartite perfect matching. *Commun. ACM*, 62(3):109–115, 2019.

[For16] Michael A. Forbes. Some concrete questions on the border complexity of polynomials. `https://www.youtube.com/watch?v=1HMogQIHT6Q`, 2016.

[FS18] Michael A. Forbes and Amir Shpilka. A PSPACE construction of a hitting set for the closure of small algebraic circuits. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 1180–1192. ACM, 2018.

[FSS14] Michael A. Forbes, Ramprasad Saptharishi, and Amir Shpilka. Hitting sets for multilinear read-once algebraic branching programs, in any order. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 867–875. ACM, 2014.

[FSTW16] Michael A. Forbes, Amir Shpilka, Iddo Tzameret, and Avi Wigderson. Proof complexity lower bounds from algebraic circuit complexity. In Ran Raz, editor, *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, volume 50 of *LIPIcs*, pages 32:1–32:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. Full version at `http://arxiv.org/abs/1606.05050`.

[FSV18] Michael A. Forbes, Amir Shpilka, and Ben Lee Volk. Succinct hitting sets and barriers to proving lower bounds for algebraic circuits. *Theory of Computing*, 14(1):1–45, 2018.

[GG20] Zeyu Guo and Rohit Gurjar. Improved explicit hitting-sets for roabps. In Jaroslaw Byrka and Raghu Meka, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2020, August 17-19, 2020, Virtual Conference*, volume 176 of *LIPIcs*, pages 4:1–4:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020.

[GKKS16] Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Arithmetic circuits: A chasm at depth 3. *SIAM J. Comput.*, 45(3):1064–1079, 2016.

[GKL12] Ankit Gupta, Neeraj Kayal, and Satya Lokam. Reconstruction of depth-4 multilinear circuits with top fan-in 2. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 625–642, 2012.

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

[GKSS17] Joshua A. Grochow, Mrinal Kumar, Michael E. Saks, and Shubhangi Saraf. Towards an algebraic natural proofs barrier via polynomial identity testing. *CoRR*, abs/1701.01717, 2017.

[GKSS19] Zeyu Guo, Mrinal Kumar, Ramprasad Saptharishi, and Noam Solomon. Derandomization from algebraic hardness: Treading the borders. In David Zuckerman, editor, *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 147–157. IEEE Computer Society, 2019.

[Gro15] Joshua A. Grochow. Unifying known lower bounds via geometric complexity theory. *Computational Complexity*, 24(2):393–475, 2015.

[Har13] Joe Harris. *Algebraic geometry: a first course*, volume 133. Springer Science & Business Media, 2013.

[Hås90] Johan Håstad. Tensor rank is NP-complete. *J. Algorithms*, 11(4):644–654, 1990.

[HS80] Joos Heintz and Claus-Peter Schnorr. Testing polynomials which are easy to compute (extended abstract). In Raymond E. Miller, Seymour Ginsburg, Walter A. Burkhard, and Richard J. Lipton, editors, *Proceedings of the 12th Annual ACM Symposium on Theory of Computing, April 28-30, 1980, Los Angeles, California, USA*, pages 262–272. ACM, 1980.

[Kal85] Kyriakos Kalorkoti. A lower bound for the formula size of rational functions. *SIAM J. Comput.*, 14(3):678–687, 1985.

[Kay12] Neeraj Kayal. Affine projections of polynomials. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 643–662, 2012.

[KI04] Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Computational Complexity*, 13(1-2):1–46, 2004.

[KNS19] Neeraj Kayal, Vineet Nair, and Chandan Saha. Average-case linear matrix factorization and reconstruction of low width algebraic branching programs. *Computational Complexity*, 28(4):749–828, 2019.

[KNST18] Neeraj Kayal, Vineet Nair, Chandan Saha, and Sébastien Tavenas. Reconstruction of full rank algebraic branching programs. *ACM Transactions on Computation Theory (TOCT)*, 11(1):1–56, 2018.

[KS01] Adam R Klivans and Daniel Spielman. Randomness efficient identity testing of multivariate polynomials. In *Proceedings of the thirty-third annual ACM symposium on Theory of computing*, pages 216–223, 2001.

[KS06] Adam R. Klivans and Amir 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.

[KS09] Zohar Shay Karnin and Amir Shpilka. Reconstruction of generalized depth-3 arithmetic circuits with bounded top fan-in. In *Proceedings of the 24th Annual IEEE Conference on Computational Complexity, CCC 2009, Paris, France, 15-18 July 2009*, pages 274–285. IEEE Computer Society, 2009.

[KS19a] Neeraj Kayal and Chandan Saha. Reconstruction of non-degenerate homogeneous depth three circuits. In Moses Charikar and Edith Cohen, editors, *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing, STOC 2019, Phoenix, AZ, USA, June 23-26, 2019*, pages 413–424. ACM, 2019. Full version at `https://eccc.weizmann.ac.il/report/2018/191`.

[KS19b] Mrinal Kumar and Ramprasad Saptharishi. Hardness-Randomness tradeoffs for algebraic computation. *Bull. EATCS*, 129, 2019.

[KST16] Neeraj Kayal, Chandan Saha, and Sébastien Tavenas. An almost cubic lower bound for depth three arithmetic circuits. In Ioannis Chatzigiannakis, Michael Mitzenmacher, Yuval Rabani, and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming, ICALP 2016, July 11-15, 2016, Rome, Italy*, volume 55 of *LIPIcs*, pages 33:1–33:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.

[Kum20] Mrinal Kumar. On the power of border of depth-3 arithmetic circuits. *ACM Trans. Comput. Theory*, 12(1):5:1–5:8, 2020.

[Lan17] Joseph M. Landsberg. *Geometry and Complexity Theory*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2017.

[LL89] Thomas Lehmkuhl and Thomas Lickteig. On the order of approximation in approximative triadic decompositions of tensors. *Theor. Comput. Sci.*, 66(1):1–14, 1989.

[MS01] Ketan Mulmuley and Milind A. Sohoni. Geometric complexity theory I: an approach to the P vs. NP and related problems. *SIAM J. Comput.*, 31(2):496–526, 2001.

[MS08] Ketan Mulmuley and Milind A. Sohoni. Geometric complexity theory II: towards explicit obstructions for embeddings among class varieties. *SIAM J. Comput.*, 38(3):1175–1206, 2008.

[MV18] Daniel Minahan and Ilya Volkovich. Complete derandomization of identity testing and reconstruction of read-once formulas. *ACM Transactions on Computation Theory (TOCT)*, 10(3):1–11, 2018.

[NW97] Noam Nisan and Avi Wigderson. Lower bounds on arithmetic circuits via partial derivatives. *Computational Complexity*, 6(3):217–234, 1997.

[Raz10] Ran Raz. Elusive functions and lower bounds for arithmetic circuits. *Theory of Computing*, 6(1):135–177, 2010.

[RS05] Ran Raz and Amir Shpilka. Deterministic polynomial identity testing in non-commutative models. *Computational Complexity*, 14(1):1–19, 2005.

[Sap15] Ramprasad Saptharishi. A survey of lower bounds in arithmetic circuit complexity. *Github survey*, 2015. Available at `https://github.com/dasarpmar/lowerbounds-survey`.

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

[Sax14] Nitin Saxena. *Progress on Polynomial Identity Testing-II*, volume 26 of *Progress in Computer Science and Applied Logic*, pages 131–146. Birkhäuser Basel, 2014.

[Shp09] Amir Shpilka. Interpolation of depth-3 arithmetic circuits with two multiplication gates. *SIAM Journal on Computing*, 38(6):2130–2161, 2009.

[Sin16] Gaurav Sinha. Reconstruction of real depth-3 circuits with top fan-in 2. In Ran Raz, editor, *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, volume 50 of *LIPIcs*, pages 31:1–31:53. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016.

[SS12] Nitin Saxena and C. Seshadhri. Blackbox Identity Testing for Bounded Top-Fanin Depth-3 Circuits: The Field Doesn't Matter. *SIAM J. Comput.*, 41(5):1285–1298, 2012.

[ST17] Ola Svensson and Jakub Tarnawski. The matching problem in general graphs is in quasi-nc. In Chris Umans, editor, *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 696–707. IEEE Computer Society, 2017.

[Sud99] Madhu Sudan. Algebra and computation. http://madhu.seas.harvard.edu/MIT/FT98/course.html, 1999. Lecture notes.

[SV14] Amir Shpilka and Ilya Volkovich. On reconstruction and testing of read-once formulas. *Theory of Computing*, 10(18):465–514, 2014.

[SV15] Amir Shpilka and Ilya Volkovich. Read-once polynomial identity testing. *Computational Complexity*, 24(3):477–532, 2015.

[SW01] Amir Shpilka and Avi Wigderson. Depth-3 arithmetic circuits over fields of characteristic zero. *Computational Complexity*, 10(1):1–27, 2001.

[Swe18] Joseph Swernofsky. Tensor rank is hard to approximate. In Eric Blais, Klaus Jansen, José D. P. Rolim, and David Steurer, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2018, August 20-22, 2018 - Princeton, NJ, USA*, volume 116 of *LIPIcs*, pages 26:1–26:9. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018.

[SY10] Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Found. Trends Theor. Comput. Sci.*, 5(3-4):207–388, 2010.

[VSBR83] Leslie G. Valiant, Sven Skyum, S. Berkowitz, and Charles Rackoff. Fast parallel computation of polynomials using few processors. *SIAM J. Comput.*, 12(4):641–644, 1983.

[vzGG03] Joachim von zur Gathen and Jürgen Gerhard. *Modern computer algebra (2. ed.).* Cambridge University Press, 2003.

A The reconstruction algorithm of [GKQ14]

For Algorithm 2 we introduce the following notation. Given integers $0 < r < i < j \le n$ we denote $\boldsymbol{x}_{r,i,j} \triangleq (x_0, \ldots, x_r, 0, \ldots, 0, x_i, 0, \ldots, 0, x_j, 0, \ldots, 0)$ a vector of variables of length $n+1$. To be consistent with the notation of [GKQ14] we also use the following notation: given an $(n+1)\times(n+1)$ matrix A and a polynomial $f(x_1, \ldots, x_n)$ we denote $\sigma_A(f) \triangleq f^h(A\boldsymbol{x})$, where f^h is the homogenization of f. Finally, we define the rank of a homogeneous quadratic polynomial q to be the minimal k such that for some linear forms $\{\ell_i\}_{i=1}^k$, $q = \ell_1^2 + \ldots + \ell_t^2 - \ell_{t+1}^2 - \ldots - \ell_k^2$.

input : Black-box access to an n-variate polynomial $f \in \mathbb{F}[\boldsymbol{x}]$ of degree at most $d = 2^\Delta$
output: Either a set of 4^Δ linear functions $\ell_1, \ldots, \ell_{4^\Delta}$ such that $f = \mathrm{ANF}_\Delta(\ell_1, \ldots, \ell_{4^\Delta})$ or `Fail`

AFR1 If $\Delta = 0$ then f is a linear function. Compute f via interpolation and return the linear function;
AFR2 **Homogenization**. Homogenize f (i.e. obtain query access to f^h);
AFR3 **Reduction to** `LDR`. Pick $(n+1)$ vectors $\boldsymbol{a}_0, \ldots, \boldsymbol{a}_n$, each of whose coordinates are chosen uniformly at random from a large enough subset $T \subseteq \mathbb{F}$. Let $r = 127$ and $m = 4^{\Delta-1}$. For $r < i < j \le n$, let $A_r^{i,j}$ be the $(n+1)\times(n+1)$ matrix whose kth column ($k \in [n+1]_0$) is $\delta_{ijk} \cdot \boldsymbol{a}_k$ where δ_{ijk} is 1 if $k \in \{0, 1, \ldots, r\} \cup \{i, j\}$ and 0 otherwise. For each $A_r^{i,j}$ invoke the LDR algorithm on $\sigma_{A_r^{i,j}}(f)$ (which is an $r+3$-variate polynomial) to obtain an m-tuple, $Q_{i,j} = (q_{i,j,1}, \ldots, q_{i,j,m})$, of quadratic polynomials satisfying
   - $\mathrm{rank}(q_{i,j,l}) \le 4$ for each $\{i, j\} \in \binom{\{r+1, r+2, \ldots, n\}}{2}$ and $l \in [m]$, and
   - $\sigma_{A_r^{i,j}}(f) = \mathrm{ANF}_{\Delta-1}(Q_{i,j})$
AFR4 **Patchwork**. Invoke the algorithm of Lemma 6.6 of [GKQ14] on input $((\boldsymbol{a}_0, \ldots, \boldsymbol{a}_n), (q_{i,j})_{r<i<j\le n})$ and obtain an m-tuple of quadratic forms $Q = (q_1, q_2, \ldots, q_m)$;
AFR5 For each $i \in [m]$, find linear forms $\ell_{i,1}, \ell_{i,2}, \ell_{i,3}, \ell_{i,4}$ such that $q_i = \ell_{i,1} \cdot \ell_{i,2} + \ell_{i,3} \cdot \ell_{i,4}$;
AFR6 **return** $(\ell_{1,1}, \ldots, \ell_{1,4}, \ell_{2,1}, \ldots, \ell_{m,3}, \ell_{m,4})$;

Algorithm 2: ANF Formula Reconstruction $\mathtt{AFR}(f(\boldsymbol{x}), \Delta)$ (Algorithm 6.9 of [GKQ14])

Remark A.1. *We note that in Algorithm 5.1 of [GKQ14] (Algorithm 3) they treat f as an $(r+1)$-variate polynomial. However the r in their Algorithm 6.9 (Algorithm 2) is not the same r as in Algorithm 3, specifically, $r_{AFR} = r_{LDR} + 2$. Hence, to avoid confusion, we decided to denote the number of variables in Algorithm 3 with $r+3$.*

input : An $r+3$-variate homogeneous polynomial $f \in \mathbb{F}[\boldsymbol{Y}]$ of degree $d = 2^{\Delta}$ given as a list of coefficients

output: Either a tuple of $m = 4^{\Delta-1}$ quadratic forms $(q_1, \ldots, q_m)$, each of rank 4, such that $f = \mathrm{ANF}_{\Delta-1}(q_1, \ldots, q_m)$, or `Fail`

LDR1 If $\Delta = 1$ then return f itself;

LDR2 Let $\mathrm{Sing}(f)$ be the ideal generated by the first order derivatives of f - i.e., the ideal

$$\left\langle \frac{\partial f}{\partial Y_0}, \frac{\partial f}{\partial Y_1}, \ldots, \frac{\partial f}{\partial Y_{r+2}} \right\rangle.$$

Use Proposition 4.8 of [GKQ14] to determine the dimension of $\mathrm{Sing}(f)$. If codimension of $\mathrm{Sing}(f)$ is not 4, output `Fail`. Else, compute a set of generators $g_1, g_2, \ldots, g_l$ for the top dimensional component (of codimension 4) of $\mathrm{Sing}(f)$ using the algorithm of Theorem 4.14 of [GKQ14];

LDR3 Compute a basis $\{\tilde{g}_1, \ldots, \tilde{g}_t\}$ for the vector space $V \subseteq \mathbb{F}[\boldsymbol{Y}]$ consisting of all the homogeneous components of degree $\frac{d}{2}$ of each g_i above. If $t = \dim(V) \neq 4$, output `Fail`;

LDR4 By solving an appropriate system of polynomial equations in 4 unknowns, compute another basis $\{h_1, h_2, h_3, h_4\}$ of V such that the singularities of each h_i has a component of codimension 4;

LDR5 By going over all permutations $\pi : [4] \to [4]$, find one such that f is an $\mathbb{F}$-linear combination of $h_{\pi(1)} \cdot h_{\pi(2)}$ and $h_{\pi(3)} \cdot h_{\pi(4)}$. Compute α, β such that $f = \alpha h_{\pi(1)} h_{\pi(2)} + \beta h_{\pi(3)} h_{\pi(4)}$. Let $\tilde{h}_1 = \alpha h_{\pi(1)}$, $\tilde{h}_2 = h_{\pi(2)}$, $\tilde{h}_3 = \beta h_{\pi(3)}$, $\tilde{h}_4 = h_{\pi(4)}$;

LDR6 For each $i \in [4]$, make a recursive call to $\mathtt{LDR}(\tilde{h}_i, \Delta - 1)$ and obtain $Q_i = (q_{i,1}, q_{i,2}, \ldots, q_{i,4^{\Delta-2}})$ such that $\tilde{h}_i = \mathrm{ANF}_{\Delta-2}(q_{i,1}, q_{i,2}, \ldots, q_{i,4^{\Delta-2}})$;

LDR7 **return** $Q = Q_1 \circ Q_2 \circ Q_3 \circ Q_4$, where '$\circ$' denotes list concatenation ;

Algorithm 3: Low-dimensional formula reconstruction $\mathtt{LDR}(f(\boldsymbol{Y}), \Delta)$ (Algorithm 5.1 of [GKQ14])

A.1 Definition of Formulaic Independence and Pairwise Singular Independence

In [GKQ14] Gupta et al. characterize "bad" inputs to their average-case, randomized algorithm in terms of points in a specific variety. As we only stated their algorithm over the complex numbers, we define varieties only over $\mathbb{C}$. However, all definitions can be easily extended to other fields as well.

For any set of n-variate polynomials $\mathcal{F} \subseteq \mathbb{C}[\boldsymbol{x}]$, we define the *zero set* of $\mathcal{F}$ as:

$$V(\mathcal{F}) \triangleq \{\boldsymbol{a} \in \mathbb{C}^n \mid \forall f \in \mathcal{F} : f(\boldsymbol{a}) = 0\} .$$

Any set $V \subseteq \mathbb{C}^n$ that can be defined as a zero set $V = V(\mathcal{F})$ for some set of polynomials $\mathcal{F} \subseteq \mathbb{C}[\boldsymbol{x}]$ is called a *variety*, or an *algebraic set*.

The notions "Formulaic Independence" and "Pairwise Singular Independence" are defined in terms of dimensions of *projective varieties*, as the polynomials in question are always homogeneous.

Let $r \in \mathbb{N}$. The *r-dimensional projective space* $\mathbb{P}^r$ is the space $\mathbb{C}^{r+1} \setminus \{\boldsymbol{0}\}$ with the equivalence relation $\sim$, where $\boldsymbol{v}, \boldsymbol{u} \in \mathbb{C}^{r+1} \setminus \{\boldsymbol{0}\}$ satisfy $\boldsymbol{v} \sim \boldsymbol{u}$ if and only if there exists some $\lambda \in \mathbb{C}$ such that $\lambda \boldsymbol{v} = \boldsymbol{u}$.

If $V = V(f_1, \ldots, f_k)$ is a variety where every f_i is an $r+1$-variate homogeneous polynomial, and if $\boldsymbol{v} \in \mathbb{C}^{r+1}$ satisfies $f_1(\boldsymbol{v}) = \ldots = f_k(\boldsymbol{v}) = 0$, then for every $\lambda \in \mathbb{C}$: $f_1(\lambda \cdot \boldsymbol{v}) = \ldots = f_k(\lambda \cdot \boldsymbol{v}) = 0$. Thus, the set $V \setminus \{\boldsymbol{0}\}$ can be viewed as a subset of $\mathbb{P}^r$. In this case we call V a *projective variety*, and define its dimension as follows:

Definition A.2 (Proposition 11.4 in [Har13])**.** *The dimension of a projective variety $V \subseteq \mathbb{P}^r$, denoted* $\dim(V)$*, is the largest integer k such that any linear space of dimension $\geq r - k$ intersects V nontrivially.*

The definition of *formulaic independence* involves the algebraic set of *singularities* of a polynomial f, and the *Jacobian matrix* of a tuple of polynomials: For a polynomial $f \in \mathbb{C}[\boldsymbol{x}]$, the set of *singularities* of f is the set of points $\boldsymbol{v} \in \mathbb{C}^n$ such that $f(\boldsymbol{v}) = \left(\frac{\partial f}{\partial x_1}\right)(\boldsymbol{v}) = \left(\frac{\partial f}{\partial x_2}\right)(\boldsymbol{v}) = \ldots = \left(\frac{\partial f}{\partial x_n}\right)(\boldsymbol{v}) = 0$. In other words,

$$\mathrm{Sing}(f) = V\left(f, \frac{\partial f}{\partial x_1}, \ldots, \frac{\partial f}{\partial x_n}\right) .$$

Given a tuple of polynomials $\boldsymbol{f} = (f_1, \ldots, f_m) \in \mathbb{C}[\boldsymbol{x}]^m$, the *Jacobian* of $\boldsymbol{f}$ is the following matrix of partial derivatives of $f_1, \ldots, f_m$:

$$J(\boldsymbol{f}, \boldsymbol{x}) = \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \ldots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \ldots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \ldots & \frac{\partial f_m}{\partial x_n} \end{pmatrix} \in \mathbb{C}[\boldsymbol{x}]^{m \times n} .$$

Definition A.3 (Definition from Section 3.1 of [GKQ14])**.** *Let $M(\boldsymbol{x}) \in \mathbb{C}[\boldsymbol{x}]^{s \times r}$ be a matrix whose entries are polynomials in $\boldsymbol{x}$, and let $t \in \mathbb{N}$. We denote by* $\mathrm{Minors}(M(\boldsymbol{x}), t) \subseteq \mathbb{C}[\boldsymbol{x}]$ *the set of determinants of all $t \times t$ submatrices of $M(\boldsymbol{x})$.*

Definition A.4 (Definition 5.2 of [GKQ14])**.** *Let $\boldsymbol{g} = (g_1(\boldsymbol{x}), \ldots, g_k(\boldsymbol{x})) \in \mathbb{C}[\boldsymbol{x}]$ be a k-tuple of homogeneous polynomials. The algebraic set $V_J(g_1, \ldots, g_k)$ ($V_J(\boldsymbol{g})$ for short) is defined to be the set of common zeroes of polynomials in* $\mathrm{Minors}(J(\boldsymbol{g}, \boldsymbol{x}), k)$*. In other words, $V_J(\boldsymbol{g})$ consists of all points $\boldsymbol{v} \in \mathbb{P}^r$ for which the rank of the Jacobian matrix $J(\boldsymbol{g}, \boldsymbol{x})$ is less than k.*

Definition A.5 (Formulaic Independence, Definition 5.3 of [GKQ14])**.** *Let $\boldsymbol{x} = (x_0, x_1, \ldots, x_r)$ and let $f, f_1, f_2, f_3, f_4 \in \mathbb{C}[\boldsymbol{x}]$ such that $f = f_1 \cdot f_2 + f_3 \cdot f_4$. Denote $\boldsymbol{f} \triangleq (f_1, f_2, f_3, f_4)$. We say that f_1, f_2, f_3, f_4 are*

formulaically independent *if* $\dim(V(\boldsymbol{f})) = r-4$ *and* $\dim(\mathrm{Sing}(f)\cap V_J(\boldsymbol{f})) < r-4$. *We say that a homogeneous ANF formula* Φ satisfies formulaic independence *at node* v *if* v *is a* $+$ *gate, and the four polynomials computed at the grandchildren of* v *are formulaically independent.*

To define *pairwise singular independence*, we must first define the *iterated Jacobian matrix*:

Definition A.6 (The Iterated Jacobian and the variety V_I, Definition 5.19 of [GKQ14])**.** *Let* $\boldsymbol{x} = (x_0, x_1, \ldots, x_r)$, *and let* $\boldsymbol{g_1}, \ldots, \boldsymbol{g_k} \in (\mathbb{C}[\boldsymbol{x}])^m$ *be* m*-tuples of homogeneous,* $(r+1)$*-variate polynomials:* $\boldsymbol{g_i} = g_{i,1}, \ldots, g_{i,m}$. *The* iterated Jacobian *of* $(\boldsymbol{g_1}, \ldots, \boldsymbol{g_k})$, *denoted* $I(\boldsymbol{g_1}, \ldots, \boldsymbol{g_k})$, *is defined to be the following matrix:* $I(\boldsymbol{g_1}, \ldots, \boldsymbol{g_k}) \in \mathbb{C}[\boldsymbol{x}]^{\binom{r+1}{k}\times m^k}$ *has its rows indexed by* k*-sized subsets of indices of variables* $\{j_1, \ldots, j_k\} \in \binom{[r+1]_0}{k}$ *and its columns indexed by tuples* $(i_1, \ldots, i_k) \in [m]^k$. *The* $(\{j_1, \ldots, j_k\}, (i_1, \ldots, i_k))$*th entry of* $I(\boldsymbol{g_1}, \ldots, \boldsymbol{g_k}, \boldsymbol{x})$ *is the polynomial*

$$Det\begin{pmatrix} \frac{\partial g_{1,i_1}}{\partial x_{j_1}} & \frac{\partial g_{2,i_2}}{\partial x_{j_1}} & \cdots & \frac{\partial g_{k,i_k}}{\partial x_{j_1}} \\ \frac{\partial g_{1,i_1}}{\partial x_{j_2}} & \frac{\partial g_{2,i_2}}{\partial x_{j_2}} & \cdots & \frac{\partial g_{k,i_k}}{\partial x_{j_2}} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial g_{1,i_1}}{\partial x_{j_k}} & \frac{\partial g_{2,i_2}}{\partial x_{j_k}} & \cdots & \frac{\partial g_{k,i_k}}{\partial x_{j_k}} \end{pmatrix}.$$

The algebraic set $V_I(\boldsymbol{g_1}, \ldots, \boldsymbol{g_k})$ *is defined to be the common zeroes of the polynomials in* $\mathrm{Minors}(I(\boldsymbol{g_1}, \ldots, \boldsymbol{g_k}), \ell^k)$.

Definition A.7 (Pairwise Singular Independence, Definition 5.20 of [GKQ14])**.** *Let* $\{f_{i,j}\}_{i,j=1}^4 \subseteq \mathbb{C}[\boldsymbol{x}]$ *be sixteen homogeneous,* $(r+1)$*-variate polynomials of the same degree. For every* $i \in [4]$, *let* $f_i = f_{i,1}\cdot f_{i,2} + f_{i,3}\cdot f_{i,4}$ *and* $\boldsymbol{f_i} = (f_{i,1}, f_{i,2}, f_{i,3}, f_{i,4})$. *For a set* $S = \{i_1, \ldots, i_k\} \subseteq [4]$, *denote:* $W_S \triangleq V_J(f_{i_1}, \ldots, f_{i_k}) \cap V_I(\boldsymbol{f_{i_1}}, \ldots, \boldsymbol{f_{i_k}})$. *We say that* $(\boldsymbol{f_1}, \boldsymbol{f_2}, \boldsymbol{f_3}, \boldsymbol{f_4})$ *are* pairwise singularly independent *if*

1. *for all* $1 \le i < j \le 4$: $\dim(\mathrm{Sing}(f_i) \cap \mathrm{Sing}(f_j)) \le r-6$, *and*
2. *for all* $S \subseteq [4]$ *such that* $|S| \ge 2$: $\dim(W_S) \le r-6$.

We say that a homogeneous ANF formula Φ *satisfies* pairwise singular independence *at a node* v *if the node* v *is a* $+$ *gate, and* $(\boldsymbol{f_{v_1}}, \boldsymbol{f_{v_2}}, \boldsymbol{f_{v_3}}, \boldsymbol{f_{v_4}})$ *are pairwise singularly independent, where* v_1, v_2, v_3, v_4 *are nodes which are the grandchildren of* v *and* $\boldsymbol{f_{v_i}}$ *is the* 4*-tuple of polynomials computed at the grandchildren of the node* v_i.

ECCC ISSN 1433-8092
https://eccc.weizmann.ac.il