Demystifying the border of depth-3 algebraic circuits

Pranjal Dutta
Department of Computer Science
Chennai Mathematical Institute
Chennai, India
pranjal@cmi.ac.in

Prateek Dwivedi
Department of CSE
Indian Institute of Technology, Kanpur
Kanpur, India
pdwivedi@cse.iitk.ac.in

Nitin Saxena
Department of CSE
Indian Institute of Technology, Kanpur
Kanpur, India
nitin@cse.iitk.ac.in

***Abstract*—Border complexity of polynomials plays an integral role in GCT (Geometric Complexity Theory) approach to P versus NP. It tries to formalize the notion of 'approximating a polynomial' via limits (Bürgisser FOCS'01). This raises the open question whether border of VP is same as VP or not; as the approximation involves exponential precision, which may not be efficiently simulable. Recently (Kumar ToCT'20) proved the universal power of the border of top-fanin-2 depth-3 circuits. Here we answer some of the related open questions. We show that the border of bounded top-fanin-k depth-3 circuits, for constant k, is relatively easy– it can be computed by a polynomial size algebraic branching program (ABP). There were hardly any de-bordering results known for prominent models before our result.**

Moreover, we give the first quasipolynomial-time blackbox identity test for the same. Prior best was in PSPACE (Forbes,Shpilka STOC'18). Also, with more technical work, we extend our results to depth-4. Our de-bordering paradigm is a multi-step process; in short we call it DiDIL –divide, derive, induct, with limit. It 'almost' reduces border top-fanin-k depth-3 circuits to special cases of read-once oblivious algebraic branching programs (ROABPs) in any-order.

***Keywords*- approximative; border; depth-3; depth-4; circuits; de-border; derandomize; blackbox; PIT; GCT; any-order; ROABP; ABP; VBP; VP; VNP.**

Full version: https://www.cse.iitk.ac.in/users/nitin/papers/border-depth3.pdf

I. INTRODUCTION

Algebraic circuit is a natural and non-uniform model of polynomial computation, which comprises the vast study of algebraic complexity [1]. We say that a polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$, over a field $\mathbb{F}$ is computable by a circuit of size s and depth d if there exists a directed acyclic graphs of size s (nodes + edges) and depth d such that its leaf nodes are labelled by variables or field constants, internal nodes are labelled with $+$ and $\times$, and the polynomial computed at the root is f. Further, if the output of a gate is never re-used then it is a *Formula*. Any formula can be converted into a layered graph called *Algebraic Branching Program* (ABP). Various complexity measures can be defined on the computational model to classify polynomials in different complexity classes. For eg. VP (respec. VBP, respec. VF) is the class of polynomials of polynomial degree, computable by polynomial-sized circuits (respec. ABPs, respec. formulas). Finally, VNP is the class of polynomials, each of which can be expressed as an exponential-sum of projection of a VP circuit family. For more details, see [2], [3].

The problem of separating algebraic complexity classes has been a central theme of this study. Valiant [1] conjectured that $\mathsf{VBP} \neq \mathsf{VNP}$, and even a stronger $\mathsf{VP} \neq \mathsf{VNP}$, as an algebraic analog of P vs. NP problem. Over the years, an impressive progress has been made towards resolving this, however, the existing tools have not been able to resolve this conclusively. In this light, Mulmuley and Sohoni [4] introduced *Geometric Complexity Theory* (GCT) program, where they studied the border (or approximative) complexity, with the aim of approaching Valiant's conjecture and *strengthening* it to: $\mathsf{VNP} \not\subseteq \overline{\mathsf{VBP}}$, i.e. (padded) permanent does not lie in the orbit closure of 'small' determinants. This notion was already studied in the context of designing matrix multiplication algorithms [5], [6], [7]. The hope, in the GCT program, was to use algebraic geometry and representation theory, and possibly settle the question once and for all. This also gave a natural reason to understand the relationship between VP and $\overline{\mathsf{VP}}$ (or VBP and $\overline{\mathsf{VBP}}$).

Outside VP vs. VNP implication, GCT has deep connections with computational invariant theory [8], [9], [10], algebraic natural proofs [11], [12], [13], lower bounds [14], [15], optimization [16], [17] and many more. We refer to [9], [18] for expository references.

The simplest notion of the approximative closure comes from the following definition [19]: a polynomial $f(\boldsymbol{x}) \in \mathbb{F}[x_1, \ldots, x_n]$ is approximated by $g(\boldsymbol{x}, \varepsilon) \in \mathbb{F}(\varepsilon)[\boldsymbol{x}]$ if there exists a $Q(\boldsymbol{x}, \varepsilon) \in \mathbb{F}[\varepsilon][\boldsymbol{x}]$ such that $g = f + \varepsilon Q$. We can also think analytically (in $\mathbb{F} = \mathbb{R}$ Euclidean topology) that $\lim_{\varepsilon \to 0} g = f$. If g belongs to a circuit class $\mathcal{C}$ (over $\mathbb{F}(\varepsilon)$, i.e. any *arbitrary* ε-power is allowed as 'cost-free' constants), then we say that $f \in \overline{\mathcal{C}}$, the approximative closure of $\mathcal{C}$. Further, one could also think of the closure as *Zariski closure*, i.e. taking the closure of the set of polynomials (considered as points) of $\mathcal{C}$: Let $\mathcal{I}$ be the smallest (annihilating) ideal whose zeros cover $\{$coefficient-vector of $g \mid g \in \mathcal{C}\}$; then put in $\overline{\mathcal{C}}$ each polynomial f with coefficient-vector being a zero of $\mathcal{I}$. Interestingly, all these notions are *equivalent* over the algebraically closed field $\mathbb{C}$ [20, §2.C].

The size of the circuit computing g defines the *approxima-*

tive (or *border*) complexity of f, denoted $\overline{\mathrm{size}}(f)$; evidently, $\overline{\mathrm{size}}(f) \leq \mathrm{size}(f)$. Due to the possible $1/\varepsilon^M$ terms in the circuit computing g, evaluating it at $\varepsilon = 0$ may not be necessarily valid (though limit exists). Hence, given $f \in \overline{\mathcal{C}}$, does not immediately reveal anything about the *exact* complexity of f. Since $g(\boldsymbol{x}, \varepsilon) = f(\boldsymbol{x}) + \varepsilon \cdot Q(\boldsymbol{x}, \varepsilon)$, we could extract the coefficient of ε^0 from g using standard interpolation trick, by setting random ε-values from $\mathbb{F}$. However, the trivial bound on the circuit size of f would depend on the degree M of ε, which could provably be *exponential* in the size of the circuit computing g, i.e. $\overline{\mathrm{size}}(f) \leq \mathrm{size}(f) \leq \exp(\overline{\mathrm{size}}(f))$ [19, Thm. 5.7].

A. De-bordering: The upper bound results

The major focus of this paper is to address the power of approximation in the restricted circuit classes. Given a polynomial $f \in \overline{\mathcal{C}}$, for an interesting class $\mathcal{C}$, we want to upper bound the exact complexity of f (we call it 'de-bordering'). If $\mathcal{C} = \overline{\mathcal{C}}$, then $\mathcal{C}$ is said to be closed under approximation: Eg. 1) $\Sigma\Pi$, the sparse polynomials (with complexity measure being sparsity), 2) Monotone ABPs [21], and 3) ROABP (read-once ABP) respec. ARO (*any-order ROABP*), with measure being the width. ARO is an ABP with a natural restriction on the use of variables per layer; see Definition 3 and Lemma 4.

Why care about upper bounds? One of the fundamental questions in the GCT is whether $\overline{\mathsf{VP}} \stackrel{?}{=} \mathsf{VP}$ [18], [22]. Confirmation or refutation of this has multiple consequences, both in the algebraic complexity and at the frontier of algebraic geometry. If $\mathsf{VP} = \overline{\mathsf{VP}}$, then any proof of $\mathsf{VP} \neq \mathsf{VNP}$ will in fact also show that $\mathsf{VNP} \not\subseteq \overline{\mathsf{VP}}$, as conjectured in [9]; however a refutation would imply that any realistic approach to the VP vs. VNP conjecture would even have to separate VNP from the families in $\overline{\mathsf{VP}} \setminus \mathsf{VP}$, requiring a far better understanding than the current state of the art.

The other significance of the upper bound result arises from the *flip* [23], [9] whose basic idea in a nutshell is to understand the theory of upper bounds first, and then use it to prove lower bounds later. Taking this further to the realm of algorithms: showing de-bordering results, for even restricted classes (eg. depth-3, small-width ABPs), could have potential identity testing implications; see Section I-B.

De-bordering results in GCT are in a very nascent stage; for e.g., the boundary of 3×3 determinants was only recently understood [24]. Note that here both the number of variables n and the degree d are constant. In this work, however, we target polynomial families with both n and d unbounded. So getting exact results about such border models is highly nontrivial considering the current state of the art.

De-bordering small-width ABPs. The exponential degree dependence of ε [19], [25] suggests us to look for separation of restricted complexity classes or try to upper bound them by some other means. In [26], the authors showed that $\mathsf{VBP}_2 \subsetneq \overline{\mathsf{VBP}_2} = \overline{\mathsf{VF}}$; here VBP_2 denotes the class of polynomials computed by width-2 ABP. Surprisingly, we also know that $\mathsf{VBP}_2 \subsetneq \mathsf{VF} = \mathsf{VBP}_3$ [27], [28]. Very recently, polynomial gap between ABPs and border-ABPs, in the trace model, for noncommutative and also for commutative monotone settings (along with $\mathsf{VQP} \neq \overline{\mathsf{VNP}}$) have also been established [21].

Quest for de-bordering depth-3 circuits. Outside such ABP results and depth-2 circuits, we understand very little about the border of other important models. Thus, it is natural to ask the same for depth-3 circuits, plausibly starting with depth-3 diagonal circuits ($\Sigma\wedge\Sigma$), i.e. polynomials of the form $\sum_{i\in[s]} c_i \cdot \ell_i^d$, where ℓ_i are linear polynomials. Interestingly, the relation between waring rank (minimum s to compute f) and border-waring rank (minimum s, to approximate f) has been studied in mathematics since ages [29], [30], [31], [32], yet it is not clear whether the measures are polynomially related or not. However, we point out that $\overline{\Sigma\wedge\Sigma}$ has a small ARO; this follows from the fact that $\Sigma\wedge\Sigma$ has small ARO by *duality trick* [33], and ARO is closed under approximation [34], [35]; see Lemma 5.

This pushes us further to study depth-3 circuits $\Sigma^{[k]}\Pi^{[d]}\Sigma$; these circuits compute polynomials of the form $f = \sum_{i\in[k]} \prod_{j\in[d]} \ell_{ij}$ where ℓ_{ij} are linear polynomials. This model with bounded fanin has been a source of great interest for derandomization [36], [37], [38], [39], [40]. In a recent twist, Kumar [41] showed that border depth-3 fanin-2 circuits are 'universal'; i.e. $\overline{\Sigma^{[2]}\Pi^{[D]}\Sigma}$ over $\mathbb{C}$ can approximate *any* homogeneous d-degree, n-variate polynomial; though his expression requires an exceedingly large $D = \exp(n, d)$.

Our upper bound results. The universality result of border depth-3 fanin-2 circuits makes it imperative to study $\overline{\Sigma^{[2]}\Pi^{[d]}\Sigma}$, for $d = \mathrm{poly}(n)$ and understand its computational power. To start with, are polynomials in this class even 'explicit' (i.e. the coefficients are efficiently computable)? If yes, is $\overline{\Sigma^{[2]}\Pi^{[d]}\Sigma} \subseteq \mathsf{VNP}$? (See [22], [42] for more general questions in the same spirit.) To our surprise, we show that the class is very explicit; in fact every polynomial in this class has a small ABP. The proof uses analytic approach and 'reduces' the Π-gate to $\wedge$-gate. We remark that it does not reveal the polynomial dependence on the ε-degree. However, this positive result could be thought as a baby step towards $\overline{\mathsf{VP}} = \mathsf{VP}$. We assume $\mathrm{char}(\mathbb{F}) = 0$, or large enough.

Theorem 1 (De-bordering constant top-fanin depth-3 circuits): For any constant k, $\overline{\Sigma^{[k]}\Pi\Sigma} \subseteq \mathsf{VBP}$, i.e. any polynomial in the border of constant top-fanin size-s depth-3 circuits, can also be computed by a $\mathrm{poly}(s)$-size algebraic branching program (ABP).

Remarks. 1. When $k = 1$, it is easy to show that $\overline{\Pi\Sigma} = \Pi\Sigma$ [26, Prop. A.12] (see Lemma 6).

2. The size of the ABP turns out to be $s^{\exp(k)}$. It is an interesting open question whether $f \in \overline{\Sigma^{[k]}\Pi\Sigma}$ has a subexponential ABP when $k = \Theta(\log s)$.

3. $\overline{\Sigma^{[k]}\Pi\Sigma}$ is the *orbit closure* of k-sparse polynomials

[43, Thm. 1.31]. Separating the orbit and its closure of certain classes is the key difficulty in GCT. Theorem 1 is one of the first such results to demystify orbit closures (of constant-sparse polynomials).

Extending to depth-4. Once we have dealt with depth-3 circuits, it is natural to ask the same for constant top-fanin depth-4 circuits. Polynomials computed by $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$ circuits are of the form $f = \sum_{i\in[k]} \prod_j g_{ij}$ where $\deg(g_{ij}) \leq \delta$. Unfortunately, our technique cannot be generalised to this model, primarily due to the inability to de-border $\overline{\Sigma\wedge\Sigma\Pi^{[\delta]}}$. However, when the bottom Π is replaced by $\wedge$, we can show $\overline{\Sigma^{[k]}\Pi\Sigma\wedge} \subseteq \mathsf{VBP}$.

B. Derandomizing the border: The blackbox PITs

Polynomial Identity Testing (PIT) is one of the fundamental decision problems in complexity theory. The Polynomial Identity Lemma [44], [45], [46], [47] gives an efficient randomized algorithm to test the zeroness of a given polynomial, even in the blackbox settings (known as Blackbox PIT), where we are not allowed to see the internal structure of the model (unlike the 'whitebox' setting), but evaluations at points are allowed. It is still an open problem to derandomize blackbox PIT. Designing a *deterministic* blackbox PIT algorithm for a circuit class is equivalent to finding a set of points such that for every nonzero circuit, the set contains a point where it evaluates to a nonzero value [48, Sec. 3.2]. Such a set is called *hitting* set.

A trivial $O(d^n)$-size explicit hitting set for a class of degree d n-variate polynomials can be obtained using the Polynomial Identity Lemma. Heintz and Schnorr [49] showed that $\mathrm{poly}(s, n, d)$ size hitting set*exists* for d-degree, n-variate polynomials computed (as well as approximated) by circuits of size s. However, the real challenge is to efficiently obtain such an *explicit* set.

Constructing small size explicit hitting set for VP is a long standing open problem in algebraic complexity, with numerous applications in graph theory [50], [51], factoring [52], [53], cryptography [54], and hardness vs randomness results [49], [55], [56], [57], [58], [59]. Moreover, a long line of depth reduction results [60], [61], [62], [63], [64] and the bootstrapping phenomenon [65], [66], [67] has justified the interest in hitting set construction for restricted classes; e.g. depth 3 [36], [37], [39], [40], depth 4 [68], [69], [70], [71], [72], ROABPs [73], [74], [68] and log-variate depth-3 diagonal circuits [75]. For exposition, see [2], [76], [77].

PIT in the border. In this paper we address the question of constructing hitting set for restrictive border circuits. $\mathcal{H}$ is a hitting set for a class $\overline{\mathcal{C}}$, if $g(\boldsymbol{x}, \varepsilon) \in \mathcal{C}_{\mathbb{F}(\varepsilon)}$, approximates a *non-zero* polynomial $f(\boldsymbol{x}) \in \overline{\mathcal{C}}$, then $\exists \boldsymbol{a} \in \mathcal{H}$ such that $g(\boldsymbol{a}, \varepsilon) \notin \varepsilon \cdot \mathbb{F}[\varepsilon]$, i.e. $f(\boldsymbol{a}) \neq 0$. Note that, as $\mathcal{H}$ will also 'hit' polynomials of class $\mathcal{C}$, construction of hitting set for the border classes (we call it 'border PIT') is a natural and possibly a different avenue to derandomize PIT. Here, we emphasize that $\boldsymbol{a} \in \mathbb{F}^n$ such that $g(\boldsymbol{a}, \varepsilon) \neq 0$, *may not* hit the limit polynomial f since $g(\boldsymbol{a}, \varepsilon)$ might still lie in $\varepsilon \cdot \mathbb{F}[\varepsilon]$; because f could have really high complexity compared to g. Intrinsically, this property makes it harder to construct an explicit hitting set for $\overline{\mathsf{VP}}$.

We also remark that there is no 'whitebox' setting in the border and thus we cannot really talk about 't-time algorithm'; rather we would only be using the term 't-time hitting set', since the given circuit after evaluating on $\boldsymbol{a} \in \mathbb{F}^n$, may require *arbitrarily* high-precision in $\mathbb{F}(\varepsilon)$.

Prior known border PITs. Mulmuley [18] asked the question of constructing an efficient hitting set for $\overline{\mathsf{VP}}$. Forbes and Shpilka [78] gave a PSPACE algorithm over the field $\mathbb{C}$. In [79], it was extended to *any* field. A very few better hitting set constructions are known for the restricted border classes, eg. poly-time hitting set for $\overline{\Pi\Sigma} = \Pi\Sigma$ [80], [81], quasi-poly hitting set for (resp.) $\overline{\Sigma\wedge\Sigma} \subseteq \overline{\mathrm{ARO}} \subseteq \overline{\mathrm{ROABP}}$ [68], [73], [74] and poly-time hitting set for the border of a restricted sum of $\log$-variate ROABPs [82].

Why care about border PIT? PIT for $\overline{\mathsf{VP}}$ has a lot of applications in the context of borderline geometry and computational complexity, as observed by Mulmuley [18]. For eg. Noether's Normalization Lemma (NNL); it is a fundamental result in algebraic geometry where the computational problem of constructing explicit *normalization map* reduces to constructing small size hitting set of $\overline{\mathsf{VP}}$ [18], [8]. Close connection between certain formulation of derandomization of NNL, and the problem of showing explicit circuit lower bounds is also known [18], [83].

The second motivation comes from the hope to find an explicit 'robust' hitting set for VP [78]; this is a hitting set $\mathcal{H}$ such that after an adequate normalization, there will be a point in $\mathcal{H}$ on which f evaluates to (say) 1. This notion overcomes the discrepancy between a hitting set for VP and a hitting set for $\overline{\mathsf{VP}}$ [78], [43]. We know that small robust hitting set exists [84], but [78] gave an explicit PSPACE construction. It is not at all clear whether the efficient hitting sets known for restricted depth-3 circuits are robust or not.

Our border PIT results. We continue our study on $\overline{\Sigma^{[k]}\Pi^{[d]}\Sigma}$ and ask for a better than PSPACE constructible hitting set. Already a polynomial-time hitting set is known for $\Sigma^{[k]}\Pi^{[d]}\Sigma$ [85], [39], [40]. But, the border class seems to be more powerful, and the known hitting sets seem to fail. However, using our structural understanding and the analytic DiDIL technique, we can *quasi*-derandomize the class.

Theorem 2 (Quasi-derandomizing depth-3*):* There exists an explicit $s^{O(\log\log s)}$-time hitting set for $\overline{\Sigma^{[k]}\Pi\Sigma}$-circuits of size s and constant k.

Remarks. 1. For $k = 1$, as $\overline{\Pi\Sigma} = \Pi\Sigma$, there is an explicit polynomial-time hitting set.

2. Our technique *necessarily* blows up the size to $s^{\exp(k)\cdot\log\log s}$. Therefore, it would be interesting to design a

subexponential time algorithm when $k = \Theta(\log s)$; or poly-time for $k = O(1)$.

3. We can not directly use the de-bordering result of Theorem 1 and try to find efficient hitting set, as we do not know explicit good hitting set for general ABPs.

4. One can extend this technique to construct quasi-polynomial time hitting set for depth-4 classes: $\overline{\Sigma^{[k]}\Pi\Sigma\wedge}$ and $\overline{\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}}$, when k and δ are constants.

The $\log$-*variate regime.* Recently, low-variate polynomials, even in highly restricted models, have gained a lot of clout for their general implications in the context of derandomization and hardness results [65], [66], [67], [59]. A slightly *non-trivial* hitting set for trivariate $\Sigma\Pi\Sigma\wedge$-circuits [65] would in fact imply quasi-efficient PIT for general circuits (optimized to poly-time in [67] with a hardness hypothesis). This motivation has pushed researchers to design efficient PITs in the $\log$-variate regime. In [75], the authors showed a $\mathrm{poly}(s)$-time blackbox identity test for $n = O(\log s)$ variate size-s circuits that have $\mathrm{poly}(s)$-dimensional partial derivative space; eg. $\log$-variate depth-3 diagonal circuits. Very recently, [82] gave the first $\mathrm{poly}(s)$-time blackbox PIT for sum of constant-many, size-s, $O(\log s)$-variate constant-width ROABPs (and its border).

We remark that non-trivial border-PIT in the low-variate bootstraps to non-trivial PIT for $\overline{\mathsf{VP}}$ as well [65], [67]. Motivated thus, we try to derandomize $\log$-variate $\overline{\Sigma^{[k]}\Pi\Sigma}$-circuits. Unfortunately, direct application of Theorem 2 fails to give a polynomial-time PIT. Surprisingly, adapting techniques from [75] to extend the existing result, combined with our DiDIL technique, we prove the following.

Theorem 3 (Derandomizing $\log$-*variate depth*-3*):* There exists an explicit $\mathrm{poly}(s)$-time hitting set for $n = O(\log s)$ variate, size-s, $\overline{\Sigma^{[k]}\Pi\Sigma}$ circuits, for constant k.

C. Limitation of standard techniques

In this section, we briefly discuss about the standard techniques for both the upper bounds and PITs, in the border sense, and point out why they fail to yield our results.

Why known upper bound techniques fail? One of the most obvious way to de-border restricted classes is to essentially show a polynomial ε-degree bound and interpolate. In general, the bound is known to be exponential [25, Thm. 5.7] which crucially uses [86, Prop. 1]. This proposition essentially shows the existence of an irreducible curve C whose degree is bounded in terms of the degree of the affine variety, that we are interested in. The degree is in general exponentially upper bounded by the size [87, Thm. 8.48]. Unless and until, one improves these bounds for varieties induced by specific models (which seems hard), one should not expect to improve the ε-degree bound, and thus interpolation trick seems useless.

As mentioned before, $\overline{\Sigma\wedge\Sigma}$-circuits could be de-bordered using the duality trick [33] (Theorem 7) to make it an $\overline{\mathrm{ARO}}$ and finally using Nisan's characterization giving $\overline{\mathrm{ARO}} = \mathrm{ARO}$ [34], [35], [88] (Theorem 4). But this trick is directly inapplicable to our models with the Π-gate, due to large waring rank & ROABP-width, as one could expect 2^d-blowup in the top fanin while converting Π-gate to $\wedge$. We also remark that the duality trick was made *field independent* in [48, Lemma 8.6.4]. In fact, very recently, [89, Theorem 4.3] gave an *improved* duality trick with no size blowup, independent of degree and number of variables.

Moreover, all the non-trivial current upper bound methods, for limit, seem to need an auxiliary linear space, which even for $\overline{\Sigma^{[2]}\Pi\Sigma}$ is not clear, due to the possibility of heavy cancellation of ε-powers. To elaborate, one of the major bottleneck is that individually $\lim_{\varepsilon\to 0} T_i$, for $i \in [2]$ *may not* exist, however, $\lim_{\varepsilon\to 0}(T_1 + T_2)$ does exist, where $T_i \in \Pi\Sigma$ (over $\mathbb{F}(\varepsilon)[\boldsymbol{x}]$). For eg. $T_1 := \varepsilon^{-1}(x + \varepsilon^2 y)y$ and $T_2 := -\varepsilon^{-1}(y + \varepsilon x)x$. No generic tool is available to 'capture' such cancellations, and may even suggest a non-linear algebraic approach to tackle the problem.

Furthermore, [90] explicitly classified certain factor polynomials to solve non-border $\Sigma^{[2]}\Pi\Sigma\wedge$ PIT. This factoring-based idea seems to fail miserably when we study factoring $\bmod \langle\varepsilon^M\rangle$; in that case, we get non-unique, usually exponentially-many, factorizations. For eg. $x^2 \equiv (x - a\cdot\varepsilon^{M/2})\cdot(x + a\cdot\varepsilon^{M/2}) \mod \langle\varepsilon^M\rangle$; for all $a \in \mathbb{F}$. In this case, there are, in fact, infinitely many factorizations. Moreover, $\lim_{\varepsilon\to 0} 1/\varepsilon^M\cdot\left(x^2 - (x - a\cdot\varepsilon^{M/2})\cdot(x + a\cdot\varepsilon^{M/2})\right) = a^2$. Therefore, infinitely many factorizations may give infinitely many limits. To top it all, Kumar's result [41] hinted a possible hardness of border-depth-3 (top-fanin-2). In that sense, ours is a very non-linear algebraic proof for restricted models which successfully opens up a possibility of finding non-representation-theoretic, and elementary, upper bounds.

Why known PIT techniques fail? Once we understand $\overline{\Sigma^{[k]}\Pi\Sigma}$, it is natural to look for efficient derandomization. However, we do not know efficient PIT for ABPs! Further, in a nutshell—1) limited (almost non-existent) understanding of linear/algebraic dependence under limit, 2) exponential upper bound on ε, and 3) not-good-enough understanding of restricted border classes make it really hard to come up with an efficient hitting set. We elaborate these points below.

[36] gave a rank-based approach to design the first quasipolynomial time algorithm for $\Sigma^{[k]}\Pi\Sigma$. A series of works [91], [85], [39], [92] finally gave a $s^{O(k)}$-time algorithm for the same. Their techniques depend on either generalizing Chinese remaindering (CR) via ideal-matching or certifying paths, or via efficient variable-reduction, to obtain a good enough rank-bound on the multiplication $(\Pi\Sigma)$ terms. Most of these approaches required a linear space, but possibility of exponential ε-powers and non-trivial cancellations make these methods fail miserably in the limit. Similar obstructions also hold for [43], [93], [94] which give efficient hitting sets for the orbit of sparse polynomials

(which is in fact *dense* in $\Sigma\Pi\Sigma$). In particular, [43] gave PIT for the orbits of variable disjoint monomials (see [43, Defn. 1.29]), under the affine group, but not the closure of it. Thus, they do not even give a subexp. PIT for $\overline{\Sigma^{[2]}\Pi\Sigma}$.

Recently, [95] gave a s^{δ^k}-time PIT, for non-SG (Sylvester–Gallai) $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$ circuits, by constructing explicit variety evasive subspace families; but to apply this idea to border PIT, one has to devise a radical-ideal based PIT idea. Currently, this does not work in the border, as $\varepsilon \mod \langle \varepsilon^M \rangle$ has an exponentially high nilpotency. Since radical$\langle \varepsilon^M \rangle = \langle \varepsilon \rangle$, it 'kills' the necessary information unless we can show a polynomial upper bound on M.

Finally, [40] came up with *faithful* map by using Jacobian + certifying path technique, which is more about algebraic rank rather than linear-rank. However, it is not at all clear how it behaves wrt $\lim_{\varepsilon \to 0}$. For eg. $f_1 = x_1 + \varepsilon^M \cdot x_2$, and $f_2 = x_1$, where M is arbitrary large. Note that the underlying Jacobian $J(f_1, f_2) = \varepsilon^M$ is nonzero; but it flips to zero in the limit. This makes the whole Jacobian machinery collapse in the border setting; as it cannot possibly give a variable reduction for the border model. (Eg. one needs to keep both x_1 and x_2 above.)

Very recently, [72] gave a quasipolynomial time hitting set for exact $\Sigma^{[k]}\Pi\Sigma\wedge$ and $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$ circuits, when k and δ are constant. This result is dependent on the Jacobian technique which fails under taking limit, as mentioned above. However, a poly-time whitebox PIT for $\Sigma^{[k]}\Pi\Sigma\wedge$ circuits was shown using DiDI-technique (Divide, Derive and Induct). This cannot be directly used because there was no ε (i.e. without limit) and $\overline{\Sigma^{[k]}\Pi\Sigma\wedge}$ has only blackbox access. Further, Theorem 1 gives an ABP, where DiDI-technique cannot be directly applied. Therefore, our DiDIL-technique can be thought of as a *strict* generalization of the DiDI-technique, first introduced in [72].

D. *Proof overview*

In this section, we sketch the proof of Theorems 1-3. The proofs are recursive and analytic. They use *logarithmic derivative*, and its power-series expansion; we call the unifying technique as DiDIL (Di=Divide, D=Derive, I=Induct, L=Limit). In both the cases, we *essentially* reduce to the well-known 'wedge' models (as fractions, with unbounded top-fanin) and then 'interpolate' it (for Theorem 1) or deduce directly about its nonzeroness (Theorem 2).

The analytic tool that we use, appears in algebra (& complexity theory) through the ring of *formal power series* $R[[x_1, \ldots, x_n]]$ (in short $R[[\boldsymbol{x}]]$), see [96], [53]. One of the advantages of the ring $R[[\boldsymbol{x}]]$ emerges from the following *inverse* identity: $(1 - x_1)^{-1} = \sum_{i \geq 0} x_1^i$, which *does not* make sense in $R[x]$, but is available now. Lastly, the logarithmic derivative operator $\mathrm{dlog}_z(f) = (\partial_z f)/f$ plays a very crucial role in 'linearizing' the product gate, since $\mathrm{dlog}_y(f \cdot g) = \partial_y(fg)/(fg) = (f \cdot \partial_y g + g \cdot \partial_y f)/(fg) = \mathrm{dlog}_y(f) + \mathrm{dlog}_y(g)$. Essentially, this operator enables us to use power-series expansion and converts the $\prod$-gate to $\wedge$.

Moreover, we will be working with the division operator (eg. over $R(\varepsilon, z)$, over certain ring R). The divisions do not come "free"— they require 'invertibility' wrt z (and ε) throughout (again landing us in $R[[\varepsilon, z]]$, see Lem. 8). We define the class $\mathcal{C}/\mathcal{D} := \{f/g \mid f \in \mathcal{C}, 0 \neq g \in \mathcal{D}\}$, for circuit classes $\mathcal{C}, \mathcal{D}$, (similarly $\mathcal{C} \cdot \mathcal{D}$ denotes the class taking respective products).

Proof idea of Theorem 1: De-bordering $\overline{\Sigma^{[k]}\Pi\Sigma}$. Consider a polynomial $f \in \mathbb{F}[\boldsymbol{x}]$ where $\boldsymbol{x} = x_1, \ldots, x_n$, such that $f \in \overline{\Sigma^{[k]}\Pi^{[d]}\Sigma}$ of size s, i.e. $g = f + \varepsilon \cdot S$ such that $\mathrm{size}_{\mathbb{F}(\varepsilon)}(g) \leq s$ (as a $\Sigma^{[k]}\Pi\Sigma$-circuit), $S \in \mathbb{F}[\varepsilon, \boldsymbol{x}]$. We want to understand the complexity of f.

$k = 1$ **case.** [26, Prop. A.12] showed that f is *exactly* computable by $\Pi\Sigma$ of size s i.e., $\overline{\Pi\Sigma} = \Pi\Sigma$. Unfortunately, due to possible heavy cancellation among linear terms of $\overline{\Sigma^{[k]}\Pi\Sigma}$, the idea directly fails for all $k > 1$.

$k = 2$ **case (almost a detailed analysis).** Our remaining focus would be to sketch the $k = 2$ proof, which would give a fair idea about generalizing the same to general k. Recall from the definition, $g := T_1 + T_2 = f + \varepsilon \cdot S$ where T_1, T_2 are multiplication terms ($\Pi\Sigma$-circuits over $\mathbb{F}(\varepsilon)[\boldsymbol{x}]$). The sum gate makes it hard to give any relevant information for de-bordering. However, if we can somehow reduce it to $k = 1$ case carefully, it can give some structural information to upper bound the size of circuit computing f. This is where the DiDIL technique comes into picture.

First we apply a homomorphism map $\Phi : \mathbb{F}(\varepsilon)[\boldsymbol{x}] \to \mathbb{F}(\varepsilon)[\boldsymbol{x}, z]$ that sends $x_i \mapsto z \cdot x_i + \alpha_i$. One can think of α_i being 'random' elements from $\mathbb{F}$; essentially it suffices to ensure that $\Phi(T_i)$ is invertible $\mod z^d$. This makes z the "degree counter" (as it helps track the degree of the polynomial and interpolate in the later stage). Moreover, Φ does not increase the complexity of f (over $\mathbb{F}(z)[\boldsymbol{x}]$)), since substituting *random* $z = a \in \mathbb{F}$ and then shifting and scaling it back gives the original f. Thus, all our efforts will be towards finding $\lim_{\varepsilon \to 0} \Phi(g) = \Phi(f)$, over $\mathbb{F}(z)[\boldsymbol{x}]$, and thus giving the size upper bound of f.

Divide and Derive. Let, $\mathcal{R} := \mathbb{F}[z]/\langle z^d \rangle$, where $\deg(f) < d$. Let $a_1 := \mathrm{val}_\varepsilon(\Phi(T_1))$ and similarly a_2 with respect to $\Phi(T_2)$; here $\mathrm{val}_\varepsilon(\cdot)$ denotes the highest power of ε dividing it. Let $\Phi(T_i) =: \varepsilon^{a_i} \cdot \tilde{T}_i$, for $i \in [2]$. Wlog also assume that $v_2 := \mathrm{val}_z(\tilde{T}_2) \leq \mathrm{val}_z(\tilde{T}_1) =: v_1$, else we can rearrange. For this particular case, $v_1 = v_2 = 0$; but we keep v_2 for the broader picture since the division by min valuation will be crucial for the general k case. Now, divide both side by $\tilde{T}_2$ and take partial derivative with respect to z, to get:

$$
\begin{aligned}
&\Phi(f)/\tilde{T}_2 + \varepsilon \cdot \Phi(S)/\tilde{T}_2 = \varepsilon^{a_2} + \Phi(T_1)/\tilde{T}_2 \\
\implies &\partial_z \left(\Phi(f)/\tilde{T}_2\right) + \varepsilon \cdot \partial_z \left(\Phi(S)/\tilde{T}_2\right) = \partial_z \left(\Phi(T_1)/\tilde{T}_2\right) =: g_1 \,. \qquad (1)
\end{aligned}
$$

First we argue that Equation 1 is well-defined over $\mathcal{R}'(\boldsymbol{x}, \varepsilon)$, where $\mathcal{R}' := \mathbb{F}[z]/\langle z^{d-v_2-1}\rangle$. Think of this as going from the given relation $\Phi(T_1) + \Phi(T_2) = \Phi(f) + \varepsilon\Phi(S)$, which holds $\bmod\ z^d$, to Equation 1 which holds $\bmod\ z^{d-v_2-1}$; the loss of precision is due to division by z^{v_2} and then one-time differentiation. Division by the minimum valuation helps to land us in the formal power series ring (Theorem 8). Formally, we write g_1 as: $\mathrm{val}_z(\Phi(T_1)/\tilde{T}_2) \geq 0 \implies \Phi(T_1)/\tilde{T}_2 \in \mathbb{F}(\boldsymbol{x}, \varepsilon)[[z]] \implies g_1 \in \mathbb{F}(\boldsymbol{x}, \varepsilon)[[z]]$.

Since, $\mathrm{val}_z(\tilde{T}_i) = \mathrm{val}_z(\Phi(T_i))$, for $i \in [2]$, it follows that $\mathrm{val}_z(\Phi(T_1) + \Phi(T_2)) \geq v_2$. Therefore, $\mathrm{val}_z(\Phi(f) + \varepsilon \cdot \Phi(S)) \geq v_2$. Setting $\varepsilon = 0$, implies $\mathrm{val}_z(\Phi(f)) \geq v_2$ as well, i.e. $\Phi(f)/\tilde{T}_2 \in \mathbb{F}(\boldsymbol{x}, \varepsilon)[[z]]$ (by Theorem 8). This also implies the same for $\Phi(S)/\tilde{T}_2$, establishing the fact that both the LHS and RHS of Equation 1 are well-defined.

Moreover, the maximum ε-power was extracted from T_2, $t_2 := \lim_{\varepsilon\to 0} \tilde{T}_2$ exists. Therefore, $\lim_{\varepsilon\to 0}(\Phi(f)/\tilde{T}_2) = \Phi(f)/t_2 \in \mathbb{F}(\boldsymbol{x}, z)$. Thus, $f_1 := \partial_z(\Phi(f)/t_2) \in \mathbb{F}(\boldsymbol{x},)[[z]]$. This establishes that g_1 approximates f_1 correctly, over $\mathcal{R}'(\boldsymbol{x})$. Essentially, the ε-definition of border is such that it allows us val_z-based divide, derive and take limit (wrt ε).

Logarithmic derivative strikes. Though it seems to reduce the fanin to 1, we have completely disfigured the model by introducing a division gate. This is exactly where *logarithmic derivative* (aka dlog) enters with bunch of helpful properties. In particular, $\partial_z\left(\Phi(T_1)/\tilde{T}_2\right) = \Phi(T_1)/\tilde{T}_2 \cdot \mathrm{dlog}\left(\Phi(T_1)/\tilde{T}_2\right) = \Phi(T_1)/\tilde{T}_2 \cdot \left(\mathrm{dlog}(\Phi(T_1)) - \mathrm{dlog}(\tilde{T}_2)\right)$.

Note that the dlog operator distributes the product gate into summation giving $\mathrm{dlog}(\Pi\Sigma) = \sum \mathrm{dlog}(\Sigma)$, where Σ denotes linear polynomials and we observe that $\mathrm{dlog}(\Sigma) = \Sigma/\Sigma \in \Sigma\wedge\Sigma$, the depth-3 powering circuits, over $\mathcal{R}'(\varepsilon, \boldsymbol{x})$. The idea is to expand $1/\ell$, where ℓ is a linear polynomial, as sum of powers of linear terms using the inverse identity: $1/(1 - a\cdot z) \equiv 1 + a\cdot z + \cdots + a^{d-v_2-2}\cdot z^{d-v_2-2} \mod z^{d-v_2-1}$.

We can assume each ℓ is invertible because of the choice of α_i's. Since $\Sigma\wedge\Sigma$ is 'closed' under taking product and addition, we obtain a final $\Sigma\wedge\Sigma$ circuit for $\mathrm{dlog}\left(\Phi(T_1)/\tilde{T}_2\right)$. Therefore, $\partial_z\left(\Phi(T_1)/\tilde{T}_2\right)$ is actually in a bloated class– $(\Pi\Sigma/\Pi\Sigma)\cdot(\Sigma\wedge\Sigma)$ over $\mathcal{R}'(\varepsilon, \boldsymbol{x})$; they compute elements of the form $(A/B)\cdot C$ where $A, B \in \Pi\Sigma$ while $C \in \Sigma\wedge\Sigma$. In particular, we get that $g_1 \in (\Pi\Sigma/\Pi\Sigma)\cdot\Sigma\wedge\Sigma$, over $\mathcal{R}'(\varepsilon, \boldsymbol{x})$.

Limit: The 'L' of DiDIL. The appealing thing about this bloated class $(\Pi\Sigma/\Pi\Sigma)\,(\Sigma\wedge\Sigma)$ is that it can be easily de-bordered using known results mainly because 1) $\overline{\Pi\Sigma} = \Pi\Sigma$, 2) $\overline{\Sigma\wedge\Sigma} \subseteq$ ARO, using duality trick and Nisan' characterization (Theorem 4) and 3) de-bordering, for a product gate, is distributive (Theorem 9). Thus, $f_1 = \lim_{\varepsilon\to 0} g_1 \in \overline{(\Pi\Sigma/\Pi\Sigma)\,(\Sigma\wedge\Sigma)} \subseteq (\Pi\Sigma/\Pi\Sigma)\cdot(\mathrm{ARO}) \subseteq \mathsf{ABP}/\mathsf{ABP}$.

Interpolate. We will now use the $f_1 = \partial_z(\Phi(f)/t_2)$ circuit (ratio of ABPs) to make our upper bound claim on $\Phi(f)$. At the core, the idea of the interpolation is very primal: to 'find' a polynomial $g(x)$, it suffices to know $g'(x)$ (which has all the information about the coefficients of g except the constant term) and $g(0)$ (the constant term).

We can think of f_1 being computed as an element in $\mathbb{F}(\boldsymbol{x}, z)$ where the degree can be actually large ($> d$), however it can be shown to be at most $\mathrm{poly}(s, d)$. Further, one can assume that $f_1 =: \sum_{i=0}^{d-v_2-2} C_i z^i$, over $\mathcal{R}'(\boldsymbol{x})$; we know such representations exist as $f_1 \in \mathbb{F}(\boldsymbol{x})[[z]]$. One can compute such expressions by using the inverse identity to expand 1/ABP expression. We emphasize that we work with the 'reduced' ABP representation i.e. the denominator is not divisible by z; otherwise we can divide both numerator and denominator by the maximum power of z and achieve such form (since it is a power series in z), to avoid $0/0$ expressions. Thus, the reduced expression must look like $\mathsf{ABP}_1/(\mathsf{ABP}_2 + z\cdot\mathsf{ABP}_3)$, where ABP_2 is *non-zero* and z-free. Expanding it using the inverse identity and truncating till $d - v_2 - 2$, we get: $f_1 \equiv (\mathsf{ABP}_1/\mathsf{ABP}_2)\cdot(1/(1 + z\cdot\mathsf{ABP}_3/\mathsf{ABP}_2)) \equiv \sum_{i=0}^{d-v_2-2} C_i\, z^i \mod z^{d-v_2-1}$.

One can show that each C_i has a small $\mathsf{ABP}/\mathsf{ABP}$ by simple interpolation and using the fact that ABPs are closed under many-time multiplication (and addition). Finally, by definite integration, we have

$$\Phi(f)/t_2 - \Phi(f)/t_2|_{z=0} \;\equiv\; \sum_{i=1}^{d-v_2-1} (C_i/i)\cdot z^i \mod z^{d-v_2}\,. \tag{2}$$

What is $\Phi(f)/t_2|_{z=0}$? As $\Phi(f)/t_2 \in \mathbb{F}(\boldsymbol{x})[[z]]$, $\Phi(f)/t_2\,|_{z=0} \in \mathbb{F}(\boldsymbol{x})$. Also, by assumption $\Phi(T_1)$ and $\tilde{T}_2$, evaluated at $z = 0$ are non-zero elements in $\mathbb{F}(\varepsilon)$. Taking limit in Equation 1, we get:

$$\Phi(f)/t_2\,|_{z=0} \;=\; \lim_{\varepsilon\to 0}\left(\Phi(T_1)/\tilde{T}_2\,|_{z=0} + \varepsilon^{a_2}\right) \;\in\; \mathbb{F}\,. \tag{3}$$

However, by assumption $\mathrm{val}_z(t_2) \geq v_2$ and moreover $t_2 \in \overline{\Pi\Sigma} = \Pi\Sigma$. Equation 2 yields $\Phi(f) \in \left(\sum_{i=1}^{d-v_2-1} C_i/i\, z^i + \mathbb{F}\right)\cdot(\Pi\Sigma) \bmod z^d \subseteq (\mathsf{ABP}/\mathsf{ABP}) \mod z^d$, of polynomial size. Finally, as $\Phi(f)$ is a $<d$-degree polynomial, we can eliminate the division gate to finally get a poly-sized ABP. This implies that f has a small ABP.

Generalizing it to k. The idea is inductive, natural and easily scales to show de-bordering result for constant k. However, for our main proof we will instead give an upper bound for a more general bloated class (it is in depth-5): $\mathsf{Gen}(k, s) := \Sigma^{[k]}\,(\Pi\Sigma/\Pi\Sigma)\,(\Sigma\wedge\Sigma/\Sigma\wedge\Sigma)$; they compute elements of the form $\sum_{i=1}^{k}(U_i/V_i)\cdot(P_i/Q_i)$, where $U_i, V_i \in \Pi\Sigma$, and $P_i, Q_i \in \Sigma\wedge\Sigma$, and the circuit (with division allowed) has size s. Of course, it trivially subsumes $\Sigma^{[k]}\Pi\Sigma$. Colloquially, we will show that this bloated model is *closed* under DiDIL operations, which is the reason we could obtain an interesting upper bound. We also emphasize that the last step of substituting $z = 0$ and taking limit, as seen in Equation 3, would be slightly more general than just an element in $\mathbb{F}$; critically it will be of the form $\lim_{\varepsilon\to 0}\mathsf{Gen}(k,\cdot)|_{z=0} \in \lim_{\varepsilon\to 0}\sum \mathbb{F}(\varepsilon)\cdot(\Sigma\wedge\Sigma/\Sigma\wedge\Sigma) \subseteq \lim_{\varepsilon\to 0}(\Sigma\wedge\Sigma/\Sigma\wedge\Sigma) \subseteq \overline{\mathrm{ARO}/\mathrm{ARO}}$, which overall gives an $\mathsf{ABP}/\mathsf{ABP}$. Here, the size blowup is only polynomial, as $\Sigma\wedge\Sigma$ is closed under multiplication (blowup being multiplicative, though). Here, we crucially use the fact that $\Pi\Sigma|_{z=0} \in \mathbb{F}(\varepsilon)$ (this remains so, even in the inductive steps!).

Remark. We point out that we needed to go to ABP, from ARO, as ARO is *not* closed under inverse, i.e. $1/\mathrm{ARO}$ may not necessarily be an ARO.

Extending to $\overline{\textit{depth-}4}$. One can extend the above techniques to de-border $\overline{\Sigma^{[k]}\Pi\Sigma\wedge}$. We point out the necessary differences to generalize the above idea. Again, we work with a Φ such that the bottom $\Sigma\wedge$ circuits are 'invertible'.

Once we divide and derive, the analytic nature remains the same. But action of dlog is more involved. Using the inverse identity, one sees that $1/\Sigma\wedge \in \Sigma\wedge\Sigma\wedge$, yielding $\mathrm{dlog}(\Pi\Sigma\wedge) = \sum \mathrm{dlog}(\Sigma\wedge) \subseteq \sum(\Sigma\wedge/\Sigma\wedge) \subseteq \sum(\Sigma\wedge)\cdot(\Sigma\wedge\Sigma\wedge) \subseteq \Sigma\wedge\Sigma\wedge$.

Thus, one has to induct on the bloated model $(\Pi\Sigma\wedge/\Pi\Sigma\wedge)\cdot(\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge)$. At the end of $(k-1)$-th step, we have: $f_{k-1} \in \overline{(\Pi\Sigma\wedge/\Pi\Sigma\wedge)\cdot(\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge)} \subseteq (\Pi\Sigma\wedge/\Pi\Sigma\wedge)\cdot(\mathrm{ARO}/\mathrm{ARO}) \subseteq \mathsf{ABP}/\mathsf{ABP}$.

We crucially use the fact that– 1) $\overline{\Pi\Sigma\wedge} = \Pi\Sigma\wedge$, as $\overline{\Sigma\wedge} = \Sigma\wedge$, and 2) $\overline{\Sigma\wedge\Sigma\wedge} \subseteq \mathrm{ARO}$, again using duality trick and Nisan's characterization (Theorem 4). Once, we have f_{k-1}, one can similarly interpolate and find f_0.

Proof idea of Theorem 2: Quasi-derandomizing $\overline{\Sigma^{[k]}\Pi\Sigma}$**.** The previous proof overview gives an idea about de-bordering $\overline{\Sigma^{[k]}\Pi\Sigma}$; unfortunately it only yields a small ABP for which efficient PIT is not known. However, we will show that DiDIL-reduction eventually lands us to identity test a few smaller cases, for which fortunately efficient PITs are known. We will follow same reduction strategy (and hence same notation) as the above. Here in Φ, we do not use random α_i, instead $\alpha_i = a^i$, for some $a \in \mathbb{F}$ (we try polynomially many a) must make $\Phi(T_i) \mod z^d$ invertible.

$k = 1$ **case.** From the previous proof, we know that $\overline{\Pi\Sigma} = \Pi\Sigma$. The idea is to use $\boldsymbol{x} \mapsto (z, z^2, \ldots, z^n)$, for a new variable z, and observe that this map preserves non-zeroness. Finally, as this is a sn-degree univariate polynomial in z, a trivial $(sn+1)$-size explicit hitting set exists.

$k = 2$ **case.** We will mainly focus on constructing an efficient hitting set for $k = 2$, which will set the path to generalize it to k. Recall, after divide and derive, we got the identity: $f_1 + \varepsilon\cdot S_1 = g_1$, over $\mathcal{R}'(\boldsymbol{x},\varepsilon)$, where $f_1 := \partial_z(\Phi(f)/t_2)$, and $g_1 := \partial_z(\Phi(T_1)/\tilde{T}_2)$.

We *would* like to have the property that $f \neq 0$, over $\mathcal{R}(\boldsymbol{x})$, if and only if $f_1 \neq 0$, over $\mathcal{R}'(\boldsymbol{x})$. Unfortunately, this *may not* necessarily hold. When can $f_1 = 0$? Either when– 1) $\Phi(f)/t_2$ is z-free, or 2) $\mathrm{val}_z(f_1) \geq d - v_2 - 1$.

When is $\Phi(f)/t_2$ z-free? It is when $\Phi(f)/t_2 = \Phi(f)/t_2|_{z=0} \in \mathbb{F}(\boldsymbol{x})$. However, by Equation 3, $\Phi(f)/t_2|_{z=0} \in \mathbb{F}$. Of course, if $f \neq 0$, it must be a non-zero element in $\mathbb{F}$ and checking it is easy.

On the other hand, $\mathrm{val}_z(f_1) \geq d - v_2 - 1$, implies that $\mathrm{val}_z(\Phi(f)/t_2) \geq d - v_2$. However, $\mathrm{val}_z(t_2) \geq v_2$, as $\mathrm{val}_z(\tilde{T}_2) = v_2$. This means, $\mathrm{val}_z(\Phi(f)) \geq d$, which is a contradiction, as we assumed that $\deg(f) < d$.

The above discussion summarizes the following important identity testing branching: $\Phi(f) \neq 0$, over $\mathcal{R}(\boldsymbol{x}) \iff f_1 \neq 0$, over $\mathcal{R}'(\boldsymbol{x})$, or $\Phi(f)/t_2 \in \mathbb{F}\backslash\{0\}$.

We remark that the $z = 0$ substitution is a natural condition as the derivation *forgets* the $\mathrm{mod}\ z$ part. At the core, the idea is really "primal". If a bivariate polynomial $G(X,Z) \neq 0$, then either its derivative $\partial_Z G(X,Z) \neq 0$, or its constant-term $G(X,0) \neq 0$ (note: $G(X,0) = G \mod Z$). So, if $G(a,0) \neq 0$ or $\partial_Z G(b,Z) \neq 0$, then the union-set $\{a,b\}$ hits $G(X,Z)$, i.e. either $G(a,Z) \neq 0$ or $G(b,Z) \neq 0$. This is crucial to get the final hitting set.

As discussed above, testing $\Phi(f)/t_2|_{z=0} \in \mathbb{F}\backslash\{0\}$ is easy, let us call this hitting set $\mathcal{H}_1$. To check $f_1 \neq 0$, note that we already have shown $f_1 \in (\Pi\Sigma/\Pi\Sigma)\cdot(\mathrm{ARO})$. Individually, we have efficient polynomial-time hitting set for $\Pi\Sigma$ (as seen in $k = 1$ case) and quasipolynomial-time hitting set for ARO.It remains to combine these hitting sets to find a final hitting set (wrt only $\boldsymbol{x}$) for $(\Pi\Sigma/\Pi\Sigma)\cdot(\mathrm{ARO})$.

Let $f_1 = (U/V)\cdot P$, where $U, V \in \Pi\Sigma$ and $P \in \mathrm{ARO}$. Let $\boldsymbol{a} \in \mathbb{F}^n$ such that $U(\boldsymbol{a}), V(\boldsymbol{a}) \neq 0$ (over $\mathbb{F}(z)$); this you find by noting $U\cdot V \in \Pi\Sigma$. Further, let $\boldsymbol{b} \in \mathbb{F}^n$ such that $P(\boldsymbol{b}) \neq 0$. Then, consider the formal sum of points $\boldsymbol{a} + t\cdot\boldsymbol{b}$, where t is a new variable. Note that, $(U/V\cdot P)(\boldsymbol{a} + t\cdot\boldsymbol{b}) \in \mathbb{F}(t,z)\backslash\{0\}$. Further, degree of t is polynomially bounded. Thus, we have a $s^{O(\log\log s)}$-time hitting set $\mathcal{H}_2$ for f_1 (Theorem 10).

Once we have individual hitting set for both cases, as discussed above, $\mathcal{H} := \mathcal{H}_1 \bigcup \mathcal{H}_2$, is indeed a hitting set (in $\boldsymbol{x}$) for $\Phi(f)$. Finally, as we have poly-degree bound on z, trying a trivial hitting set gives finally a $s^{O(\log\log s)}$-time hitting set for $\overline{\Sigma^{[2]}\Pi\Sigma}$.

Generalizing to k**.** As before, the general model of induction will be on $\mathsf{Gen}(k,s)$. The core idea of branching-out remains the same. We know that at the end of $k-1$ steps, $f_{k-1} \in (\Pi\Sigma/\Pi\Sigma)\cdot(\mathrm{ARO}/\mathrm{ARO})$. Using similar ideas as above, it is possible to construct a hitting set (for details, see Theorem 10).

However, as seen before, the $z = 0$ substitution, in the k case, i.e. $\lim_{\varepsilon\to 0}\mathsf{Gen}(k,\cdot)|_{z=0}$, gives an element of the form $\mathrm{ARO}/\mathrm{ARO}$, for which we have a quasipolynomial-time hitting set. As seen before, we know it suffices to hit each branch separately, since their union can be shown to be hitting set for the original $\Phi(f)$,. Moreover, the syntactic degree can be shown to be bounded by $s^{O(k)}$, which finally gives a quasipolynomial-time hitting set for the general k.

Extending to depth-4. To derandomize the two types of $\overline{\Sigma^{[k]}\Pi\Sigma\Upsilon}$ circuits, where $\Upsilon = \{\wedge, \Pi^{[\delta]}\}$, we again follow DiDIL and branching-out strategy as above. We point out the main differences in generalizing it to depth-4. As $\Sigma\Upsilon$ circuits are at most s-sparse, it suffices to consider the sparse-PIT map to construct the α_i in Φ [81].

Once we divide and derive, the action of dlog becomes different. However, using the inverse identity, one can show that $1/\Sigma\Upsilon \in \Sigma\wedge\Sigma\Upsilon$, which finally yields that $\text{dlog}(\Pi\Sigma\Upsilon) \in \Sigma\wedge\Sigma\Upsilon$. So, one inducts on the bloated model $\overline{(\Pi\Sigma\Upsilon/\Pi\Sigma\Upsilon) \cdot (\Sigma\wedge\Sigma\Upsilon/\Sigma\wedge\Sigma\Upsilon)}$, and at the end, we have $f_{k-1} \in \overline{(\Pi\Sigma\Upsilon/\Pi\Sigma\Upsilon) \cdot (\Sigma\wedge\Sigma\Upsilon/\Sigma\wedge\Sigma\Upsilon)} \subseteq (\Pi\Sigma\Upsilon/\Pi\Sigma\Upsilon) \cdot \overline{(\Sigma\wedge\Sigma\Upsilon/\Sigma\wedge\Sigma\Upsilon)}$.

Note that $\Sigma\Upsilon$ is closed under de-bordering (and so is $\Pi\Sigma\Upsilon$). When $\Upsilon = \wedge$, we know $\overline{\Sigma\wedge\Sigma\wedge} \subseteq \text{ARO}$. Moreover, we have poly-time hitting set for $\Pi\Sigma\wedge$. Therefore, after combining them (Lem. 10), we have hitting sets $\mathcal{H}_j$ at each j-th branch. Their union gives the final hitting set.

However, when $\Upsilon = \Pi^{[\delta]}$, we currently do not know how to de-border, as we can no longer apply duality trick to conclude that $\overline{\Sigma\wedge\Sigma\Pi^{[\delta]}}$ has small ARO. Nonetheless, we know quasipolynomial-time hitting set for $\Sigma\wedge\Sigma\Pi^{[\delta]}$ [70]. This method is *rank*-based and eventually shows that a small-support (of size $O(\delta \log s)$) trailing monomial exists. Think of this monomial as the 'last' monomial in a polynomial (under a monomial ordering) where the variables used is really 'few'. This proof is based on bounding shifted-partial-derivative space. However, rank behaves 'well' wrt limit and thus this method can be extended to border; to eventually show that small support trailing monomial exists in a nonzero $P \in \overline{\Sigma\wedge\Sigma\Pi^{[\delta]}}$ of size s. We can then use trivial hitting set of size $s^{O(\delta \log s)}$ to conclude existence of a non-zero small support trailing monomial in the border.

We would like to stress that the given circuit g, at point $\boldsymbol{x} = \boldsymbol{a} \in \mathbb{F}^n$, takes value in $\mathbb{F}(\varepsilon)$, though $f(\boldsymbol{a}) \in \mathbb{F}$. However we do not count the (potentially very-high) precision of $g(\boldsymbol{a})$ in our time-complexity; because we only care about hitting set design within $\mathbb{F}^n$.

Once we have a hitting set for $\overline{\Sigma\wedge\Sigma\Pi^{[\delta]}}$, the result follows as we can combine hitting set for $\Pi\Sigma\Pi^{[\delta]}$ and $\overline{\Sigma\wedge\Sigma\Pi^{[\delta]}}$, using Theorem 10, yielding a hitting set $\mathcal{H}_j$, for each branch. Finally taking a union gives the final hitting set.

Proof idea of Theorem 3: Derandomizing log-variate $\overline{\Sigma^{[k]}\Pi\Sigma}$. We adapt techniques from [75] and argue that eventually the same proof works to give a poly-time hitting set for log-variate $\overline{\Sigma\wedge\Sigma}$-circuits. First, let us argue that why poly-time hitting set for $\overline{\Sigma\wedge\Sigma}$ translates to giving a polynomial-time hitting set for $\overline{\Sigma^{[k]}\Pi\Sigma}$.

To argue, we follow the DiDIL technique as shown in the depth-3 circuits, and eventually arrive at the 'end' where $f_{k-1} \in (\Pi\Sigma/\Pi\Sigma) \cdot (\text{ARO}/\text{ARO})$. However, we point out that this is not any generic poly-sized ARO but the debordering of log-variate $\overline{\Sigma\wedge\Sigma}$. If there is a poly-time hitting set for this class, after combining this with poly-time hitting set of $\Pi\Sigma$ (using Theorem 10), we again get a polynomial-time hitting set $\mathcal{H}_{k-1}$ for f_{k-1}. Eventually, at each branch, we will similarly get a polynomial-time hitting set $\mathcal{H}_j$, at the j-th step. Taking a union finally yields a polynomial-time hitting set as we wanted.

Thus, it remains to argue that one can extend the idea of [75] to give a polynomial-time hitting set for log-variate $\overline{\Sigma\wedge\Sigma}$-circuits. The flow of the proof goes as follows— (1) show that $f \in \overline{\Sigma\wedge\Sigma}$ has $\text{poly}(s)$ partial-derivative space; this is a vector space spanned by all partial-derivatives of f; this follows from the fact that $\Sigma\wedge\Sigma$, over $\mathbb{F}(\varepsilon)$ has polynomial partial-derivative space [97, Lemma 10.2], and rank behaves "well" under limit yielding the same for f, (2) show that low partial-derivative space implies low cone-size monomials (for definition see the Def. II); this is directly from [48, Corollary 4.14], (3) decide the non-zeroness of the coefficient of a low cone-size monomial efficiently, over $\mathbb{F}(\varepsilon)$; this can be done by general-interpolation, similar to [75, Lemma 4]; see the statement in Theorem 12, and (4) show that the low-cone-size monomials are $\text{poly}(sd)$-many [75, Lemma 5], see Theorem 11) for the statement.

II. Notations and Preliminaries

Notation. Denote $[n] = \{1, \ldots, n\}$, and $\boldsymbol{x} = (x_1, \ldots, x_n)$. For, $\boldsymbol{a} = (a_1, \ldots, a_n), \boldsymbol{b} = (b_1, \ldots, b_n) \in \mathbb{F}^n$, and a variable t, we denote $\boldsymbol{a} + t \cdot \boldsymbol{b} := (a_1 + tb_1, \ldots, a_n + tb_n)$.

We also use $\mathbb{F}[[x]]$, to denote the ring of formal power series over $\mathbb{F}$. Formally, $f = \sum_{i \ge 0} c_i x^i$, with $c_i \in \mathbb{F}$, is an element in $\mathbb{F}[[x]]$. Further, $\mathbb{F}(\boldsymbol{x})$ denotes the function field, where the elements are of the form f/g, where $f, g \in \mathbb{F}[\boldsymbol{x}]$ $(g \neq 0)$.

Logarithmic derivative. Over a ring R and a variable y, the *logarithmic derivative* $\text{dlog}_y : R[y] \longrightarrow R(y)$ is defined as $\text{dlog}_y(f) := \partial_y f/f$; here ∂_y denotes the partial derivative wrt variable y. One important property of dlog is that it is *additive* over a product as $\text{dlog}_y(f \cdot g) = \partial_y(fg)/(fg) = (f \cdot \partial_y g + g \cdot \partial_y f)/(fg) = \text{dlog}_y(f) + \text{dlog}_y(g)$. [dlog *linearizes product*]

Circuit size. Some of the complexity parameters of a circuit are *depth* (number of layers), *syntactic degree* (the maximum degree polynomial computed by any node), *fanin* (maximum number of inputs to a node).

Operation on Complexity Classes. For class $\mathcal{C}$ and $\mathcal{D}$ defined over ring R, our bloated model is any combination of sum, product, and division of polynomials from respective classes. For instance, $\mathcal{C}/\mathcal{D} = \{f/g : f \in \mathcal{C}, 0 \neq g \in \mathcal{D}\}$ similarly $\mathcal{C}\cdot\mathcal{D}$ for products, $\mathcal{C}+\mathcal{D}$ for sum, and other possible combinations. Also we use $\mathcal{C}_{\mathsf{R}}$ to denote the basic ring R on which $\mathcal{C}$ is being computed over.

Hitting set. A set of points $\mathcal{H} \subseteq \mathbb{F}^n$ is called a *hitting-set* for a class $\mathcal{C}$ of n-variate polynomials if for any nonzero polynomial $f \in \mathcal{C}$, there exists a point in $\mathcal{H}$ where f evaluates to a nonzero value. A $T(s)$-time hitting-set would mean that the hitting-set can be generated in time $\leq T(s)$, for input size s.

Valuation. Valuation is a map $\mathrm{val}_y : R[y] \longrightarrow \mathbb{Z}_{\geq 0}$, over a ring R, such that $\mathrm{val}_y(\cdot)$ is defined to be the maximum power of y dividing the element. It can be easily extended to fraction field $R(y)$, by defining $\mathrm{val}_y(p/q) := \mathrm{val}_y(p) - \mathrm{val}_y(q)$; where it can be negative.

Field. We denote the underlying field as $\mathbb{F}$ and assume that it is of characteristic 0 (eg. $\mathbb{Q}, \mathbb{Q}_p$). All our results hold for other fields (eg. $\mathbb{F}_{p^e}$) of *large* characteristic p.

Approximative closure. For an algebraic complexity class $\mathcal{C}$, the approximation is defined as follows [26, Def. 2.1].

Definition 1 (Approximative closure of a class): Let $\mathcal{C}_{\mathbb{F}}$ be a class of polynomials defined over a field $\mathbb{F}$. Then, $f(\boldsymbol{x}) \in \mathbb{F}[x_1, \ldots, x_n]$ is said to be in Approximative Closure $\overline{\mathcal{C}}$ if and only if there exists polynomial $Q \in \mathbb{F}[\varepsilon, \boldsymbol{x}]$ such that $\mathcal{C}_{\mathbb{F}(\varepsilon)} \ni g(\boldsymbol{x}, \varepsilon) = f(\boldsymbol{x}) + \varepsilon \cdot Q(\boldsymbol{x}, \varepsilon)$.

Cone-size of monomials. For a monomial $\boldsymbol{x}^{\boldsymbol{a}}$, the cone of $\boldsymbol{x}^{\boldsymbol{a}}$ is the set of all sub-monomials of $\boldsymbol{x}^{\boldsymbol{a}}$. The cardinality of this set is called *cone-size* of $\boldsymbol{x}^{\boldsymbol{a}}$. It equals $\prod_{i \in [n]} (a_i + 1)$, where $\boldsymbol{a} = (a_1, \ldots, a_n)$. We will denote $\mathrm{cs}(m)$, as the cone-size of the monomial m.

III. Basics from algebraic complexity

Our interest primarily is in the following two ABP-variants: ROABP and ARO.

Definition 2 (Read-once Oblivious ABP (ROABP)): An ABP is defined as Read-Once Oblivious Algebraic Branching Program (ROABP) in a variable order $(x_{\sigma(1)}, \ldots, x_{\sigma(n)})$ for some permutation $\sigma : [n] \to [n]$, if edges of i-th layer of ABP are univariate polynomials in $x_{\sigma(i)}$.

Definition 3 (Any-order ROABP (ARO)): A polynomial $f \in \mathbb{F}[\boldsymbol{x}]$ is computable by ARO of size s if for all possible permutation of variables there exists a ROABP of size at most s in that variable order.

Next we show that polynomials approximated by ARO can be easily de-bordered. To the best of our knowledge the following lemma was sketched in [35] (implicit in [88]).

Lemma 4 (De-bordering ARO): Consider a polynomial $f \in \mathbb{F}[\boldsymbol{x}]$ which is approximated by ARO of size s over $\mathbb{F}(\varepsilon)[\boldsymbol{x}]$. Then, there exists an ARO of size s which exactly computes $f(\boldsymbol{x})$.

Lemma 5 (De-bordering $\Sigma\wedge\Sigma\wedge$): Consider a polynomial $f \in \mathbb{F}[\boldsymbol{x}]$ which is approximated by $\Sigma\wedge\Sigma\wedge$ of size s over $\mathbb{F}(\varepsilon)[\boldsymbol{x}]$ and syntactic degree D. Then there exists an ARO of size $O(sn^2D^2)$ which exactly computes $f(\boldsymbol{x})$.

Lemma 6 (De-bordering $\Pi\Sigma\wedge$): Consider a polynomial $f \in \mathbb{F}[\boldsymbol{x}]$ which is approximated by $\Pi\Sigma\wedge$ of size s over $\mathbb{F}(\varepsilon)[\boldsymbol{x}]$. Then there exists a $\Pi\Sigma\wedge$ (hence an ARO) of size s which exactly computes $f(\boldsymbol{x})$.

Lemma 7 (Duality trick [33]): The polynomial $f = (x_1 + \ldots + x_n)^d$ can be written as

$$f = \sum_{i \in [t]} f_{i1}(x_1) \cdots f_{in}(x_n),$$

where $t = O(nd)$, and f_{ij} is a univariate polynomial of degree at most d.

Here is an important lemma to show that positive valuation with respect to y, lets us express a function as a power-series of y.

Lemma 8 (Valuation): Let $f \in \mathbb{F}(\boldsymbol{x}, y)$ such that $\mathrm{val}_y(f) \geq 0$. Then, $f \in \mathbb{F}(\boldsymbol{x})[[y]] \bigcap \mathbb{F}(\boldsymbol{x}, y)$.

Let $\mathcal{C}$ and $\mathcal{D}$ be two classes over $\mathbb{F}[\boldsymbol{x}]$. Consider the bloated-class $(\mathcal{C}/\mathcal{C}) \cdot (\mathcal{D}/\mathcal{D})$, which has elements of the form $(g_1/g_2) \cdot (h_1/h_2)$, where $g_i \in \mathcal{C}$ and $h_i \in \mathcal{D}$ ($g_2 h_2 \neq 0$). One can also similarly define its border (which will be an element in $\mathbb{F}(\boldsymbol{x})$). Here is an important observation.

Lemma 9: $\overline{(\mathcal{C}/\mathcal{C}) \cdot (\mathcal{D}/\mathcal{D})} \subseteq (\overline{\mathcal{C}}/\overline{\mathcal{C}}) \cdot (\overline{\mathcal{D}}/\overline{\mathcal{D}})$.

The following lemma is useful to construct hitting set for product of two circuit classes when the hitting set of individual circuit is known.

Lemma 10: Let $\mathcal{H}_1, \mathcal{H}_2 \subseteq \mathbb{F}^n$ of size s_1 and s_2 respectively be the hitting set of the class of n-variate degree d polynomials computable by $\mathcal{C}_1$ and $\mathcal{C}_2$ respectively. Then, for the class of polynomials computable by $\mathcal{C}_1 \cdot \mathcal{C}_2$ there is an explicit hitting set $\mathcal{H}$ of size $s_1 \cdot s_2 \cdot O(d)$.

The next lemma shows that there are only few low-cone monomials in a non-zero n-variate polynomial.

Lemma 11 (Counting low-cones, [75, Lem 5]): The number of n-variate monomials with cone-size at most k is $O(rk^2)$, where $r := (3n/\log k)^{\log k}$.

The following lemma is the same as [75, Lemma 4]. It is proved by multivariate interpolation.

Lemma 12 (Coefficient extraction): Given a circuit C, over the underlying field $\mathbb{F}(\varepsilon)$, we can 'extract' the coefficient of monomial m in C; in time $\mathrm{poly}(\mathrm{size}(C), \mathrm{cs}(m), d)$, where $\mathrm{cs}(m)$ denotes the cone-size of m.

IV. Conclusion

This opens a variety of questions which would enrich border-complexity theory.

1) Does $\overline{\Sigma^{[k]}\Pi\Sigma} \subseteq \Sigma\Pi\Sigma$; or $\overline{\Sigma^{[k]}\Pi\Sigma} \subseteq \mathsf{VF}$, i.e. does it have a small formula?
2) Can we show that $\mathsf{VBP} \neq \overline{\Sigma^{[k]}\Pi\Sigma}$?
3) Can we improve the current hitting set of $s^{\exp(k)\cdot\log\log s}$ to $s^{O(\mathrm{poly}(k)\cdot\log\log s)}$, or even a $\mathrm{poly}(s)$-time hitting set? The current technique seems to blowup the exponent.
4) Can we de-border $\overline{\Sigma\wedge\Sigma\Pi^{[\delta]}}$, or $\overline{\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}}$, for constant k and δ?
5) Can we show that $\overline{\Sigma^{[k]}\wedge\Sigma} \subseteq \Sigma\wedge\Sigma$ for constant k?
6) Can we de-border $\overline{\Sigma^{[2]}\Pi\Sigma\wedge^{[2]}}$? i.e. the bottom-layer has variable mixing.

De-bordering vs. Derandomization. In this work, de-bordering results did not directly give us hitting sets, since we end up getting more general models where explicit hitting sets are unknown. However, we were still able derandomize because of the DiDIL-technique. Moreover, while extending this to depth-4, we could quasi-derandomize $\overline{\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}}$, because eventually hitting set for $\overline{\Sigma\wedge\Sigma\Pi^{[\delta]}}$ is known. However we could not de-border $\overline{\Sigma\wedge\Sigma\Pi^{[\delta]}}$, because the duality-trick *fails* to give an ARO. This whole paradigm suggests that de-bordering *may be* harder than its derandomization.

Acknowledgment

Pranjal thanks Department of CSE, IIT Kanpur for the hospitality, and acknowledges the support of Google Ph. D. Fellowship. Nitin thanks the funding support from DST (DST/SJF/MSA-01/2013-14) and N. Rama Rao Chair.

References

[1] L. G. Valiant, "Completeness classes in algebra," in *STOC 1979*, 1979, pp. 249–261.

[2] A. Shpilka and A. Yehudayoff, "Arithmetic circuits: A survey of recent results and open questions," *Foundations and Trends in Theoretical Computer Science*, vol. 5, pp. 207–388, 2010.

[3] M. Mahajan, "Algebraic complexity classes," *CoRR*, vol. abs/1307.3863, 2013.

[4] K. Mulmuley and M. A. Sohoni, "Geometric complexity theory I: an approach to the P vs. NP and related problems," *SIAM J. Comput.*, vol. 31, no. 2, pp. 496–526, 2001.

[5] V. Strassen, "Polynomials with rational coefficients which are hard to compute," *SIAM J. Comput.*, vol. 3, no. 2, pp. 128–149, 1974.

[6] D. Bini, "Relations between exact and approximate bilinear algorithms. Applications," *Calcolo*, vol. 17, no. 1, pp. 87–97, 1980.

[7] D. Coppersmith and S. Winograd, "Matrix multiplication via arithmetic progressions," *J. Symb. Comput.*, vol. 9, no. 3, pp. 251–280, 1990.

[8] M. A. Forbes and A. Shpilka, "Explicit noether normalization for simultaneous conjugation via polynomial identity testing," in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*. Springer, 2013, pp. 527–542.

[9] K. D. Mulmuley, "The GCT program toward the P vs. NP problem," *Communications of the ACM*, vol. 55, no. 6, pp. 98–107, 2012.

[10] G. Ivanyos, Y. Qiao, and K. V. Subrahmanyam, "Non-commutative edmonds' problem and matrix semi-invariants," *Comput. Complex.*, vol. 26, no. 3, pp. 717–763, 2018.

[11] J. A. Grochow, M. Kumar, M. Saks, and S. Saraf, "Towards an algebraic natural proofs barrier via polynomial identity testing," *arXiv preprint arXiv:1701.01717*, 2017.

[12] M. Bläser, C. Ikenmeyer, V. Lysikov, A. Pandey, and F. Schreyer, "On the orbit closure containment problem and slice rank of tensors," in *SODA 2021*, 2021, pp. 2565–2584.

[13] P. Chatterjee, M. Kumar, C. Ramya, R. Saptharishi, and A. Tengse, "On the Existence of Algebraically Natural Proofs," *FOCS 2020*, 2020.

[14] P. Bürgisser and C. Ikenmeyer, "Explicit lower bounds via geometric complexity theory," in *Proceedings of the 45^{th} Annual ACM symposium on Theory of Computing*, 2013, pp. 141–150.

[15] J. A. Grochow, "Unifying known lower bounds via geometric complexity theory," *Computational Complexity*, vol. 24, no. 2, pp. 393–475, 2015.

[16] Z. Allen-Zhu, A. Garg, Y. Li, R. Oliveira, and A. Wigderson, "Operator scaling via geodesically convex optimization, invariant theory and polynomial identity testing," in *STOC 2018*, 2018, pp. 172–181.

[17] P. Bürgisser, C. Franks, A. Garg, R. Oliveira, M. Walter, and A. Wigderson, "Towards a theory of non-commutative optimization: Geodesic 1st and 2nd order methods for moment maps and polytopes," in *FOCS 2019*, 2019, pp. 845–861.

[18] K. Mulmuley, "Geometric complexity theory V: equivalence between blackbox derandomization of polynomial identity testing and derandomization of noether's normalization lemma," in *FOCS 2012*, 2012, pp. 629–638.

[19] P. Bürgisser, "The complexity of factors of multivariate polynomials," *Found. Comput. Math.*, vol. 4, no. 4, pp. 369–396, 2004.

[20] D. Mumford, *Algebraic geometry I: complex projective varieties*. Springer Science & Business Media, 1995.

[21] M. Bläser, C. Ikenmeyer, M. Mahajan, A. Pandey, and N. Saurabh, "Algebraic Branching Programs, Border Complexity, and Tangent Spaces," in *CCC 2020*, vol. 169, 2020, pp. 21:1–21:24.

[22] J. A. Grochow, K. D. Mulmuley, and Y. Qiao, "Boundaries of VP and VNP," in *ICALP 2016*, 2016, pp. 34:1–34:14.

[23] K. Mulmuley, "Geometric complexity theory vi: The flip via positivity," *arXiv preprint arXiv:0704.0229*, 2010.

[24] J. Hüttenhain and P. Lairez, "The boundary of the orbit of the 3-by-3 determinant polynomial," *Comptes Rendus Mathematique*, vol. 354, no. 9, pp. 931–935, 2016.

[25] P. Bürgisser, "Correction to: The complexity of factors of multivariate polynomials," *Found. Comput. Math.*, vol. 20, no. 6, pp. 1667–1668, 2020.

[26] K. Bringmann, C. Ikenmeyer, and J. Zuiddam, "On algebraic branching programs of small width," *J. ACM*, vol. 65, no. 5, pp. 32:1–32:29, 2018.

[27] M. Ben-Or and R. Cleve, "Computing algebraic formulas using a constant number of registers," *SIAM J. Comput.*, vol. 21, no. 1, pp. 54–58, 1992.

[28] E. Allender and F. Wang, "On the power of algebraic branching programs of width two," *Computational Complexity*, vol. 25, no. 1, pp. 217–253, 2016.

[29] J. J. Sylvester, *On the principles of the calculus of forms*, 1852.

[30] M. Boij, E. Carlini, and A. Geramita, "Monomials as sums of powers: the real binary case," *Proceedings of the American Math. Society*, vol. 139, no. 9, pp. 3039–3043, 2011.

[31] A. Bernardi, E. Carlini, M. V. Catalisano, A. Gimigliano, and A. Oneto, "The hitchhiker guide to: Secant varieties and tensor decomposition," *Mathematics*, vol. 6, no. 12, p. 314, 2018.

[32] F. Gesmundo and J. M. Landsberg, "Explicit polynomial sequences with maximal spaces of partial derivatives and a question of k. mulmuley," *Theory of Computing*, vol. 15, no. 3, pp. 1–24, 2019.

[33] N. Saxena, "Diagonal Circuit Identity Testing and Lower Bounds," in *ICALP 2008*, 2008, pp. 60–71.

[34] N. Nisan, "Lower bounds for non-commutative computation," in *STOC 1991*, 1991, pp. 410–418.

[35] M. Forbes, "Some concrete questions on the border complexity of polynomials. presentation given at the workshop on algebraic complexity theory wact 2016 in tel aviv," 2016.

[36] Z. Dvir and A. Shpilka, "Locally decodable codes with two queries and polynomial identity testing for depth 3 circuits," *SIAM Journal on Computing*, vol. 36, no. 5, pp. 1404–1434, 2007.

[37] N. Kayal and N. Saxena, "Polynomial identity testing for depth 3 circuits," *computational complexity*, vol. 16, no. 2, pp. 115–138, 2007.

[38] Z. S. Karnin and A. Shpilka, "Black box polynomial identity testing of generalized depth-3 arithmetic circuits with bounded top fan-in," in *CCC 2008*, 2008, pp. 280–291.

[39] N. Saxena and C. Seshadhri, "Blackbox identity testing for bounded top-fanin depth-3 circuits: The field doesn't matter," *SIAM J. Comput.*, vol. 41, no. 5, pp. 1285–1298, 2012.

[40] M. Agrawal, C. Saha, R. Saptharishi, and N. Saxena, "Jacobian hits circuits: Hitting sets, lower bounds for depth-d occur-k formulas and depth-3 transcendence degree-k circuits," *SIAM J. Comput.*, vol. 45, no. 4, pp. 1533–1562, 2016.

[41] M. Kumar, "On the power of border of depth-3 arithmetic circuits," *ACM Trans. Comput. Theory*, vol. 12, no. 1, pp. 5:1–5:8, 2020.

[42] Editors, "Open Problems," *Algebraic Methods, Simons Institute for the Theory of Computing*, 2018.

[43] D. Medini and A. Shpilka, "Hitting Sets and Reconstruction for Dense Orbits in VP_e and $\Sigma\Pi\Sigma$ Circuits," *CCC'21*, 2021.

[44] Ø. Ore, "Über höhere kongruenzen," *Norsk Mat. Forenings Skrifter*, vol. 1, no. 7, p. 15, 1922.

[45] R. A. DeMillo and R. J. Lipton, "A Probabilistic Remark on Algebraic Program Testing," *Information Processing Letters*, vol. 7, no. 4, pp. 193–195, 1978.

[46] R. Zippel, "Probabilistic algorithms for sparse polynomials," in *International symposium on symbolic and algebraic manipulation*. Springer, 1979, pp. 216–226.

[47] J. T. Schwartz, "Fast probabilistic algorithms for verification of polynomial identities," *J. ACM*, vol. 27, no. 4, pp. 701–717, 1980.

[48] M. A. Forbes, "Polynomial Identity Testing of Read-Once Oblivious Algebraic Branch-ing Programs," *PhD thesis, Massachusetts Institute of Technology*, 2014.

[49] J. Heintz and C. Schnorr, "Testing polynomials which are easy to compute (extended abstract)," in *STOC 1980*, 1980, pp. 262–272.

[50] L. Lovász, "On determinants, matchings, and random algorithms," in *Fundamentals of Computation Theory (FCT 1979)*, 1979, pp. 565–574.

[51] S. A. Fenner, R. Gurjar, and T. Thierauf, "A deterministic parallel algorithm for bipartite perfect matching," *Commun. ACM*, vol. 62, no. 3, pp. 109–115, 2019.

[52] S. Kopparty, S. Saraf, and A. Shpilka, "Equivalence of polynomial identity testing and deterministic multivariate polynomial factorization," in *CCC 2014*, 2014, pp. 169–180.

[53] P. Dutta, N. Saxena, and A. Sinhababu, "Discovering the roots: uniform closure results for algebraic classes under factoring," in *STOC 2018*. ACM, 2018, pp. 1152–1165.

[54] M. Agrawal, N. Kayal, and N. Saxena, "PRIMES is in P," *Annals of mathematics*, pp. 781–793, 2004.

[55] N. Nisan and A. Wigderson, "Hardness vs Randomness," *Journal of Computer and System Sciences*, vol. 49, no. 2, pp. 149–167, 1994.

[56] M. Agrawal, "Proving Lower Bounds Via Pseudo-random Generators," in *FSTTCS 2005*, 2005, pp. 92–105.

[57] V. Kabanets and R. Impagliazzo, "Derandomizing polynomial identity tests means proving circuit lower bounds," in *STOC 2003*, 2003, pp. 355–364.

[58] Z. Dvir, A. Shpilka, and A. Yehudayoff, "Hardness-randomness tradeoffs for bounded depth arithmetic circuits," *SIAM J. Comput.*, vol. 39, no. 4, pp. 1279–1293, 2009.

[59] P. Dutta, N. Saxena, and T. Thierauf, "A largish sum-of-squares implies circuit hardness and derandomization," in *ITCS'21*, 2021.

[60] L. G. Valiant, S. Skyum, S. Berkowitz, and C. Rackoff, "Fast Parallel Computation of Polynomials Using Few Processors," *SIAM J. Comput.*, vol. 12, no. 4, pp. 641–644, 1983.

[61] M. Agrawal and V. Vinay, "Arithmetic circuits: A chasm at depth four," in *FOCS 2008*, 2008, pp. 67–75.

[62] P. Koiran, "Arithmetic circuits: The chasm at depth four gets wider," *Theor. Comput. Sci.*, vol. 448, pp. 56–65, 2012.

[63] S. Tavenas, "Improved bounds for reduction to depth 4 and depth 3," *Inf. Comput.*, vol. 240, pp. 2–11, 2015.

[64] A. Gupta, P. Kamath, N. Kayal, and R. Saptharishi, "Arithmetic circuits: A chasm at depth 3," *SIAM J. Comput.*, vol. 45, no. 3, pp. 1064–1079, 2016.

[65] M. Agrawal, S. Ghosh, and N. Saxena, "Bootstrapping variables in algebraic circuits," *PNAS*, vol. 116, no. 17, pp. 8107–8118, 2019.

[66] M. Kumar, R. Saptharishi, and A. Tengse, "Near-optimal bootstrapping of hitting sets for algebraic circuits," in *SODA'19*, 2019.

[67] Z. Guo, M. Kumar, R. Saptharishi, and N. Solomon, "Derandomization from algebraic hardness: Treading the borders," in *FOCS'19*, 2019, pp. 147–157.

[68] M. A. Forbes and A. Shpilka, "Quasipolynomial-time identity testing of non-commutative and read-once oblivious algebraic branching programs," in *FOCS'13*, 2013, pp. 243–252.

[69] M. Beecken, J. Mittmann, and N. Saxena, "Algebraic independence and blackbox identity testing," *Inf. Comput.*, vol. 222, pp. 2–19, 2013.

[70] M. A. Forbes, "Deterministic divisibility testing via shifted partial derivatives," in *FOCS'15*, 2015, pp. 451–465.

[71] S. Peleg and A. Shpilka, "Polynomial time deterministic identity testing algorithm for $\sum^{[3]}\prod\sum\prod^{[2]}$ circuits via Edelstein-Kelly type theorem for quadratic polynomials," in *STOC'21*, 2021.

[72] P. Dutta, P. Dwivedi, and N. Saxena, "Deterministic identity testing paradigms for bounded top-fanin depth-4 circuits," *CCC'21*, 2021.

[73] M. Agrawal, R. Gurjar, A. Korwar, and N. Saxena, "Hitting-sets for ROABP and sum of set-multilinear circuits," *SIAM J. Comput.*, vol. 44, no. 3, pp. 669–697, 2015.

[74] R. Gurjar, A. Korwar, and N. Saxena, "Identity testing for constant-width, and any-order, read-once oblivious arithmetic branching programs," *Theory Comput.*, vol. 13, no. 1, pp. 1–21, 2017.

[75] M. A. Forbes, S. Ghosh, and N. Saxena, "Towards blackbox identity testing of log-variate circuits," in *ICALP'18*, 2018.

[76] N. Saxena, "Progress on polynomial identity testing-II," in *Perspectives in Comput. Comp.*, 2014, pp. 131–146.

[77] M. Kumar and R. Saptharishi, "Hardness-randomness tradeoffs for algebraic computation," *Bulletin of EATCS*, vol. 3, no. 129, 2019.

[78] M. A. Forbes and A. Shpilka, "A PSPACE construction of a hitting set for the closure of small algebraic circuits," in *STOC'18*, 2018.

[79] Z. Guo, N. Saxena, and A. Sinhababu, "Algebraic dependencies and PSPACE algorithms in approximative complexity over any field," *Theory Comput.*, vol. 15, pp. 1–30, 2019.

[80] M. Ben-Or and P. Tiwari, "A deterministic algorithm for sparse multivariate polynomial interpolation," in *STOC'88*, 1988, pp. 301–309.

[81] A. R. Klivans and D. A. Spielman, "Randomness efficient identity testing of multivariate polynomials," in *STOC'01*, 2001, pp. 216–223.

[82] P. Bisht and N. Saxena, "Blackbox identity testing for sum of special ROABPs and its border class," *Comput. Compl.*, 2021.

[83] P. Mukhopadhyay, "Depth-4 identity testing and noether's normalization lemma," in *CSR'16*, 2016.

[84] A. Carbery and J. Wright, "Distributional and L^q norm inequalities for polynomials over convex bodies in $\mathbb{R}^n$," *Math. Res. Let.*, vol. 8, no. 3, pp. 233–248, 2001.

[85] N. Saxena and C. Seshadhri, "An almost optimal rank bound for depth-3 identities," *SIAM J. Comput.*, vol. 40, no. 1, pp. 200–224, 2011.

[86] T. Lehmkuhl and T. Lickteig, "On the order of approximation in approximative triadic decompositions of tensors," *Theoretical Computer Science*, vol. 66, no. 1, pp. 1–14, 1989.

[87] P. Bürgisser, M. Clausen, and M. A. Shokrollahi, *Algebraic complexity theory*. Springer Science & Business Media, 2013, vol. 315.

[88] R. Gurjar, A. Korwar, and N. Saxena, "Identity testing for constant-width, and commutative, read-once oblivious abps," in *CCC 2016*, vol. 50, 2016, pp. 29:1–29:16.

[89] M. Bläser, J. Dörfler, and C. Ikenmeyer, "On the complexity of evaluating highest weight vectors," *CCC'21*, 2021.

[90] C. Saha, R. Saptharishi, and N. Saxena, "A case of depth-3 identity testing, sparse factorization and duality," *Computational Complexity*, vol. 22, no. 1, pp. 39–69, 2013.

[91] N. Kayal and S. Saraf, "Blackbox polynomial identity testing for depth 3 circuits," in *FOCS'09*, 2009, pp. 198–207.

[92] N. Saxena and C. Seshadhri, "From sylvester-gallai configurations to rank bounds: Improved blackbox identity test for depth-3 circuits," *J. ACM*, vol. 60, no. 5, pp. 33:1–33:33, 2013.

[93] C. Saha and B. Thankey, "Hitting sets for orbits of circuit classes and polynomial families," *Electron. Colloquium Comput. Complex.*, 2021. [Online]. Available: https://eccc.weizmann.ac.il/report/2021/015

[94] V. Bhargava and S. Ghosh, "Improved Hitting Set for Orbit of ROABPs," *RANDOM'21*, 2021.

[95] Z. Guo, "Variety Evasive Subspace Families," *CCC'21*, 2021.

[96] I. Niven, "Formal power series," *The American Mathematical Monthly*, vol. 76, no. 8, pp. 871–889, 1969. [Online]. Available: http://www.jstor.org/stable/2317940

[97] X. Chen, N. Kayal, and A. Wigderson, *Partial derivatives in arithmetic complexity and beyond*. Now Publishers Inc, 2011.