Demystifying the border of depth-3 algebraic circuits

Pranjal Dutta[*] ✉ Prateek Dwivedi [†]✉ Nitin Saxena [‡]✉

Abstract

Border complexity of polynomials plays an integral role in GCT (Geometric complexity theory) approach to $\mathsf{P} \neq \mathsf{NP}$. It tries to formalize the notion of 'approximating a polynomial' via limits (Bürgisser FOCS'01). This raises the open question $\overline{\mathsf{VP}} \overset{?}{=} \mathsf{VP}$; 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 ($\overline{\Sigma^{[2]}\Pi\Sigma}$). Here we answer some of the related open questions. We show that the border of bounded top-fanin depth-3 circuits ($\overline{\Sigma^{[k]}\Pi\Sigma}$ 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 $\overline{\Sigma^{[k]}\Pi\Sigma}$ 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.

Contents

[*]Chennai Mathematical Institute, India (and Department of CSE, IIT Kanpur)
[†]Department of CSE, IIT Kanpur
[‡]Department of CSE, IIT Kanpur

1 Introduction: Border complexity, GCT and beyond

Algebraic circuit is a natural (& non-uniform) model of polynomial computation, which comprises the vast study of algebraic complexity [Val79]. 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, refer to Appendix A and [SY10, Mah13].

The problem of separating algebraic complexity classes has been a central theme of this study. Valiant [Val79] conjectured that VBP $\neq$ VNP, and even a stronger VP $\neq$ 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 [MS01] 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: VNP $\not\subseteq$ $\overline{\text{VBP}}$, 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 [Str74, Bin80, BCRL79, CW90, LO15]. The hope, in the GCT program, was to use available tools from 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 $\overline{\text{VP}}$ (or VBP and $\overline{\text{VBP}}$).

Outside VP vs. VNP implication, GCT has deep connections with computational invariant theory [FS13a, Mul12b, GGOW16, BGO+18, IQS18], algebraic natural proofs [GKSS17, BIL+21, CKR+20, KRST20], lower bounds [BI13, Gro15, LO15], optimization [AZGL+18, BFG+19] and many more. We refer to [BLMW11, Sec. 9] and [Mul12b, Mul12a] for expository references.

The simplest notion of the approximative closure comes from the following definition [Bür04, Bür20]: 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* (algebraic definition over any $\mathbb{F}$), 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 $\{\text{coefficient-vector of } g \mid g \in \mathcal{C}\}$; 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 $\mathbb{C}$ [Mum95, §2.C].

The size of the circuit computing g defines the *approximative* (or *border*) complexity of f, denoted $\overline{\text{size}}(f)$; evidently, $\overline{\text{size}}(f) \leq \text{size}(f)$. 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. $\overline{\text{size}}(f) \leq \text{size}(f) \leq \exp(\overline{\text{size}}(f))$ [Bür04, Thm. 5.7].

1.1 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 [BIM+20], 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; for definition and a formal proof, see Definition A.4 and Lemma A.21.

Why care about upper bounds? One of the fundamental questions in the GCT paradigm is whether $\overline{\text{VP}} \overset{?}{=} \text{VP}$ [Mul12a, GMQ16]. Confirmation or refutation of this question has multiple consequences, both in the algebraic complexity and at the frontier of algebraic geometry. If $\text{VP} = \overline{\text{VP}}$, then any proof of $\text{VP} \neq \text{VNP}$ will in fact also show that $\text{VNP} \not\subseteq \overline{\text{VP}}$, as conjectured in [Mul12b]; however a refutation would imply that any realistic approach to the VP vs. VNP conjecture would even have to separate the permanent from the families in $\overline{\text{VP}} \setminus \text{VP}$ (and for this, one needs a far better understanding than the current state of the art).

The other significance of the upper bound result arises from the *flip* [Mul10, Mul12b] whose basic idea in a nutshell is to understand the theory of upper bounds first, and then use this theory 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. For details, see Subsection 1.2.

De-bordering results in GCT are in a very nascent stage; for example, the boundary of 3×3 determinants was only recently understood [HL16]. 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 [Bür04, Bür20] suggests us to look for separation of restricted complexity classes or try to upper bound them by some other means. In [BIZ18], Bringmann et al. 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 $\mathsf{VBP}_2 \subsetneq \mathsf{VF} = \mathsf{VBP}_3$ [BC92, AW16]. 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 [BIM+20].

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 [Syl52, BCG11, BCC+18, GL19], 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* [Sax08], and ARO is closed under approximation [Nis91, For16]; for details see Lemma A.22.

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 [DS07, KS07, KS08, SS12, ASSS16]. In a recent twist, Kumar [Kum20] showed that border depth-3 fanin-2 circuits are 'universally' expressive; 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 [GMQ16, Edi18] 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 statement and its proof is first of its kind which eventually 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 the field $\mathbb{F}$ characteristic to be $= 0$, or large enough. For a detailed statement, see Theorem 3.2.

Theorem 1.1 (De-bordering 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 $\overline{\Pi\Sigma} = \Pi\Sigma$ [BIZ18, Prop. A.12] (see Lemma A.20).

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 [MS21, Thm. 1.31]. Separating the orbit and its closure of certain classes is the key difficulty in GCT. Theorem 1.1 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 $\deg(g_{ij}) \le \delta$. 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 $\overline{\Sigma^{[k]}\Pi\Sigma\wedge} \subseteq$ VBP; we sketch the proof in Theorem B.1.

1.2 Derandomizing the border: The blackbox PITs

Polynomial Identity Testing (PIT) is one of the fundamental decision problems in complexity theory. The Polynomial Identity Lemma [Ore22, DL78, Zip79, Sch80] 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 [For14, Sec. 3.2]. Such a set is called *hitting* set.

A trivial explicit hitting set for a class of degree d polynomial of size $O(d^n)$ can be obtained using the Polynomial Identity Lemma. Heintz and Schnorr [HS80] 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 theory, with numerous algorithmic applications in graph theory [Lov79, MVV87, FGT19], factoring [KSS14, DSS18], cryptography [AKS04], and hardness vs randomness results [HS80, NW94, Agr05, KI03, DSY09, DST21]. Moreover, a long line of depth reduction results [VSBR83, AV08, Koi12, Tav15, GKKS16] and the bootstrapping phenomenon [AGS19, KST19, GKSS19, And20] has justified the interest in hitting set construction for restricted classes; e.g. depth 3 [DS07, KS07, SS12, ASSS16], depth 4 [FS13b, BMS13, For15, Shp19, PS20, PS21, DDS21], ROABPs [AGKS15, GKS17, FS13b, GG20, BS21] and log-variate depth-3 diagonal circuits [FGS18]. We refer to [SY10, Sax14, KS19] for expositions.

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 [Mul12a] asked the question of constructing an efficient hitting set for $\overline{\mathsf{VP}}$. Forbes and Shpilka [FS18] gave a PSPACE algorithm over the field $\mathbb{C}$. In [GSS19], Guo et al. extended this result to *any* field. A very few better hitting set constructions are known for the restricted border classes, eg. poly-time hitting set for $\overline{\Pi\Sigma} = \Pi\Sigma$ [BOT88, KS01], quasi-poly hitting set for (resp.) $\overline{\Sigma\wedge\Sigma} \subseteq \overline{\mathsf{ARO}} \subseteq \overline{\mathsf{ROABP}}$ [FS13b, AGKS15, GKS17] and poly-time hitting set for the border of a restricted sum of log-variate ROABPs [BS21].

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 [Mul12a]. 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 $\overline{\mathsf{VP}}$ [Mul12a, FS13a]. Close connection between certain formulation of derandomization of NNL, and the problem of showing explicit circuit lower bounds is also known [Mul12a, Muk16].

The second motivation comes from the hope to find an explicit 'robust' hitting set for VP [FS18]; 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 $\overline{\mathsf{VP}}$ [FS18, MS21]. We know that small robust hitting set exists [CW01], but an explicit PSPACE construction was given in [FS18]. 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 $\Sigma^{[k]}\Pi^{[d]}\Sigma$ [SS11, SS12, ASSS16]. 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 are able to quasi-derandomize the class completely. For the detailed statement, see Theorem 4.1.

Theorem 1.2 (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 1.1 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. For details, see Appendix C.

The log*-variate regime.* In recent developments [AGS19, KST19, GKSS19, DST21] 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. A slightly *non-trivial* hitting set for trivariate $\Sigma\Pi\Sigma\wedge$-circuits [AGS19] would in fact imply quasi-efficient PIT for general circuits (optimized to poly-time in [GKSS19] with a hardness hypothesis). This motivation has pushed researchers to work on log-variate regime and design efficient PITs. In [FGS18], Forbes et al. 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, Bisht and Saxena [BS21] 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 [AGS19, GKSS19]. Motivated thus, we try to derandomize log-variate $\overline{\Sigma^{[k]}\Pi\Sigma}$-circuits. Unfortunately, direct application of Theorem 1.2 fails to give a polynomial-time PIT. Surprisingly, adapting techniques from [FGS18] to extend the existing result (Theorem 4.5), combined with our DiDIL technique, we prove the following. For details, see Theorem 4.6.

Theorem 1.3 (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.*

1.3 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 [Bür20, Thm. 5.7] which crucially uses [LL89, 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 [BCS13, 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 [Sax08] (see Lemma A.14) to make it an $\overline{\mathrm{ARO}}$ and finally using Nisan's characterization giving $\overline{\mathrm{ARO}} =$ ARO [Nis91, For16, GKS16] (Lemma A.21). 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 [For14, Lemma 8.6.4]. In fact, very recently, [BDI21, 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, [SSS13] 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 mod $\langle \varepsilon^M \rangle$; 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 [Kum20] 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, as we do not know efficient PIT for ABPs, known techniques would not yield an efficient PIT for the same. 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.

Dvir and Shpilka [DS07] gave a rank-based approach to design the first quasipolynomial time algorithm for $\Sigma^{[k]}\Pi\Sigma$. A series of works [KS09, SS11, SS12, SS13] 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 [MS21, ST21, BG21] which give efficient hitting sets for the orbit of sparse polynomials (which is in fact *dense* in $\Sigma\Pi\Sigma$). In particular, Medini and Shpilka [MS21] gave PIT for the orbits of variable disjoint monomials (see [MS21, Defn. 1.29]), under the affine group, but not the closure of it. Thus, they do not even give a subexponential PIT for $\overline{\Sigma^{[2]}\Pi\Sigma}$.

Recently, Guo [Guo21] 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, [ASSS16] 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, [DDS21] 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 polynomial-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 1.1 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 [DDS21], which now applies to uncharted borders.

1.4 Proof overview

In this section, we sketch the proof of Theorems 1.1-1.3. 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 1.1) or deduce directly about its nonzeroness (Theorem 1.2). Needless to say, these reductions and their consequences are startling and quite powerful.

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 [Niv69, DSS18, Sin19]. 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 $\text{dlog}_{z_1}(f) = (\partial_{z_1} f)/f$ plays a very crucial role in 'linearizing' the product gate, since $\text{dlog}_y(f\cdot g) = \partial_y(fg)/(fg) = (f\cdot\partial_y g + g\cdot\partial_y f)/(fg) = \text{dlog}_y(f) + \text{dlog}_y(g)$. 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 $R(\varepsilon, z_1)$, over certain ring R). The divisions do not come "free"— they require 'invertibility' wrt z_1 (and ε) throughout (again landing us in $R[[\varepsilon, z_1]]$, see Lem. A.17). 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 1.1: 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 $\text{size}_{\mathbb{F}(\varepsilon)}(g)\leq s$ (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.** [BIZ18, Prop. A.12] showed that f is *exactly* computable by $\Pi\Sigma$ of size s i.e., $\overline{\Pi\Sigma} = \Pi\Sigma$. Eventually, the proof relies on the fact that the product distributes the ε-powers showing $\overline{(\Pi\mathcal{C})}\subseteq\Pi(\overline{\mathcal{C}})$, for any reasonable class $\mathcal{C}$ (see Lemma A.19); here by $f\in\Pi\mathcal{C}$ we mean $f =: \prod f_i$, where $f_i\in\mathcal{C}$. 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 which we discuss in the next couple of paragraphs.

First we apply a homomorphism map $\Phi : \mathbb{F}(\varepsilon)[\boldsymbol{x}] \to \mathbb{F}(\varepsilon)[\boldsymbol{x}, z_1, z_2]$ that sends $x_i \mapsto z_1 \cdot x_i + \Psi(x_i)$. Here $\Psi : \mathbb{F}[\boldsymbol{x}] \to \mathbb{F}[z_2]$ is a map defined as $\Psi(x_i) = z_2^i$. Note that $\Psi(\ell) \neq 0$, for any nonzero linear polynomial ℓ; which essentially ensures that $\Phi(T_i)$ is invertible mod z_1^d. This makes z_1 the "degree counter" (as it helps track the degree of the polynomial and interpolate in the later stage) while z_2 the "non-zeroness preserving" variable. Moreover, Φ does not increase the complexity of f (over $\mathbb{F}(\boldsymbol{z})[\boldsymbol{x}]$)), where $\boldsymbol{z} = (z_1, z_2)$, since substituting *random* $\boldsymbol{z} = (a_1, a_2) \in \mathbb{F}^2$ 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 $\mathbb{F}(\boldsymbol{z})[\boldsymbol{x}]$, and thus giving the size upper bound of f.

Divide and Derive. Let, $\mathcal{R} := \mathbb{F}(z_2)[z_1]/\langle z_1^d \rangle$, 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 $v_2 := \mathrm{val}_{z_1}(\tilde{T}_2) \leq \mathrm{val}_{z_1}(\tilde{T}_1) =: v_1$, else we can rearrange. Divide both side by $\tilde{T}_2$ and take partial derivative with respect to z_1, to get:

$$\begin{aligned} \Phi(f)/\tilde{T}_2 \,+\, \varepsilon \cdot \Phi(S)/\tilde{T}_2 \;&=\; \varepsilon^{a_2} \,+\, \Phi(T_1)/\tilde{T}_2 \\ \implies \partial_{z_1}\left(\Phi(f)/\tilde{T}_2\right) \,+\, \varepsilon \cdot \partial_{z_1}\left(\Phi(S)/\tilde{T}_2\right) \;&=\; \partial_{z_1}\left(\Phi(T_1)/\tilde{T}_2\right) =: g_1 \,. \end{aligned} \tag{1.4}$$

First we argue that Equation 1.4 is well-defined over $\mathcal{R}'(\boldsymbol{x}, \varepsilon)$, where $\mathcal{R}' := \mathbb{F}(z_2)[z_1]/\langle z_1^{d-v_2-1} \rangle$. Think of this as going from the given relation $\Phi(T_1) + \Phi(T_2) = \Phi(f) + \varepsilon\Phi(S)$, which holds mod z_1^d, to Equation 1.4 which holds mod $z_1^{d-v_2-1}$; the loss of precision is due to division by $z_1^{v_2}$ and then one-time differentiation. Division by the minimum valuation helps to land us in the formal power series ring, thanks to Lemma A.17. Formally, we write g_1 as:

$$\mathrm{val}_{z_1}(\Phi(T_1)/\tilde{T}_2) \geq 0 \implies \Phi(T_1)/\tilde{T}_2 \in \mathbb{F}(\boldsymbol{x}, z_2, \varepsilon)[[z_1]] \implies g_1 \in \mathbb{F}(\boldsymbol{x}, z_2, \varepsilon)[[z_1]] \,.$$

Since, $\mathrm{val}_{z_1}(\tilde{T}_i) = \mathrm{val}_{z_1}(\Phi(T_i))$, for $i \in [2]$, it follows that $\mathrm{val}_{z_1}(\Phi(T_1) + \Phi(T_2)) \geq v_2$. Therefore, $\mathrm{val}_{z_1}(\Phi(f) + \varepsilon \cdot \Phi(S)) \geq v_2$. Setting $\varepsilon = 0$, implies $\mathrm{val}_{z_1}(\Phi(f)) \geq v_2$ as well, i.e. $\Phi(f)/\tilde{T}_2 \in \mathbb{F}(\boldsymbol{x}, z_2, \varepsilon)[[z_1]]$ (by Lemma A.17). This also implies the same for $\Phi(S)/\tilde{T}_2$, establishing the fact that both the LHS and RHS of Equation 1.4 are well-defined.

Moreover, $\lim_{\varepsilon \to 0} \tilde{T}_2 =: t_2$ exists as the maximum ε-power was extracted from T_2. Therefore, $\lim_{\varepsilon \to 0}(\Phi(f)/\tilde{T}_2) = \Phi(f)/t_2 \in \mathbb{F}(\boldsymbol{x}, \boldsymbol{z})$. Thus, $f_1 := \partial_{z_1}(\Phi(f)/t_2) \in \mathbb{F}(\boldsymbol{x}, z_2)[[z_1]]$. 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_1}-based divide, derive and taking 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,

$$\partial_{z_1}\left(\Phi(T_1)/\tilde{T}_2\right) \;=\; \Phi(T_1)/\tilde{T}_2 \cdot \mathrm{dlog}\left(\Phi(T_1)/\tilde{T}_2\right) \;=\; \Phi(T_1)/\tilde{T}_2 \cdot \left(\mathrm{dlog}(\Phi(T_1)) - \mathrm{dlog}(\tilde{T}_2)\right).$$

Note that the dlog operator distributes the product gate into summation giving $\mathrm{dlog}(\Pi\Sigma) =$

$\sum \text{dlog}(\Sigma)$, where Σ denotes linear polynomials and we observe that $\text{dlog}(\Sigma) = \Sigma/\Sigma \in \Sigma\wedge\Sigma$, 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:

$$1/(1 - a \cdot z_1) \equiv 1 + a \cdot z_1 + \cdots + a^{d-v_2-2} \cdot z_1^{d-v_2-2} \mod z_1^{d-v_2-1} .$$

We can assume each ℓ is invertible because of the map Ψ. Since $\Sigma\wedge\Sigma$ is 'closed' under taking product and addition (Subsection A.1), we obtain a final $\Sigma\wedge\Sigma$ circuit for $\text{dlog}\left(\Phi(T_1)/\tilde{T}_2\right)$. Details of this step can be found in Claim 3.8. Therefore, $\partial_{z_1}\left(\Phi(T_1)/\tilde{T}_2\right)$ 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 (Lemma A.15) and Nisan' characterization (Lemma A.21) and 3) de-bordering, for a product gate, is distributive (Lemma A.19). Thus,

$$f_1 = \lim_{\varepsilon \to 0} g_1 \in \overline{(\Pi\Sigma/\Pi\Sigma)\,(\Sigma\wedge\Sigma)} \subseteq (\Pi\Sigma/\Pi\Sigma) \cdot (\text{ARO}) \subseteq \mathsf{ABP}/\mathsf{ABP} .$$

Interpolate. We will now use the $f_1 = \partial_{z_1}(\Phi(f)/t_2)$ 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 $\mathbb{F}(\boldsymbol{x}, \boldsymbol{z})$ where the degree can be actually large ($> d$), however it can be shown to be at most $\text{poly}(s, d)$. Further, one can assume that $f_1 =: \sum_{i=0}^{d-v_2-2} C_i z_1^i$, over $\mathcal{R}'(\boldsymbol{x})$; we know such representations exist as $f_1 \in \mathbb{F}(z_2, \boldsymbol{x})[[z_1]]$. One can compute such expressions by using the inverse identity to expand $1/\mathsf{ABP}$ expression. We emphasize that we work with the 'reduced' ABP representation i.e. the denominator is not divisible by z_1; otherwise we can divide both numerator and denominator by the maximum power of z_1 and achieve such form (since it is a power series in z_1), to avoid $0/0$ expressions. Thus, the reduced expression must look like $\mathsf{ABP}_1/(\mathsf{ABP}_2 + z_1 \cdot \mathsf{ABP}_3)$, where ABP_2 is *non-zero* and z_1-free. Expanding it using the inverse identity and truncating till $d - v_2 - 2$, we get:

$$f_1 \equiv (\mathsf{ABP}_1/\mathsf{ABP}_2) \cdot (1/(1 + z_1 \cdot \mathsf{ABP}_3/\mathsf{ABP}_2)) \equiv \sum_{i=0}^{d-v_2-2} C_i\, z_1^i \mod z_1^{d-v_2-1} .$$

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); for details see Lemma A.2. Finally, by definite integration, we have

$$\Phi(f)/t_2 - \Phi(f)/t_2|_{z_1=0} \equiv \sum_{i=1}^{d-v_2-1} (C_i/i) \cdot z_1^i \mod z_1^{d-v_2} . \tag{1.5}$$

What is $\Phi(f)/t_2|_{z_1=0}$? As $\Phi(f)/t_2 \in \mathbb{F}(z_2, \boldsymbol{x})[[z_1]]$, $\Phi(f)/t_2\,|_{z_1=0} \in \mathbb{F}(z_2, \boldsymbol{x})$. Also, by assumption $\Phi(T_1)$ and $\tilde{T}_2$, evaluated at $z_1 = 0$ are non-zero elements in $\mathbb{F}(z_2, \varepsilon)$. Taking limit in Equation 1.4, we get:

$$\Phi(f)/t_2\,|_{z_1=0} \;=\; \lim_{\varepsilon\to 0}\left(\Phi(T_1)/\tilde{T}_2\,|_{z_1=0} + \varepsilon^{a_2}\right) \;\in\; \lim_{\varepsilon\to 0}\left(\mathbb{F}(z_2,\varepsilon) \,+\, \varepsilon^{a_2}\right) \;\subseteq\; \mathbb{F}(z_2)\,. \tag{1.6}$$

However, by assumption $\mathrm{val}_{z_1}(t_2) \geq v_2$ and moreover $t_2 \in \overline{\Pi\Sigma} = \Pi\Sigma$. Equation 1.5 yields

$$\Phi(f) \;\in\; \left(\sum_{i=1}^{d-v_2-1} C_i/i\,z_1^i + \mathbb{F}(z_2)\right)\cdot(\Pi\Sigma)\ \mathrm{mod}\ z_1^d \subseteq (\mathsf{ABP}/\mathsf{ABP})\ \ \mathrm{mod}\ \ z_1^d\,,$$

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 (Lemma A.2). 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 $z_1 = 0$ and taking limit, as seen in Equation 1.6, would be slightly more general than just an element in $\mathbb{F}(z_2)$; critically it will be of the form

$$\lim_{\varepsilon\to 0}\mathsf{Gen}(k,\cdot)|_{z_1=0} \;\in\; \lim_{\varepsilon\to 0}\sum\mathbb{F}(z_2,\varepsilon)\cdot(\Sigma\wedge\Sigma/\Sigma\wedge\Sigma) \;\subseteq\; \lim_{\varepsilon\to 0}(\Sigma\wedge\Sigma/\Sigma\wedge\Sigma) \;\subseteq\; \mathsf{ARO}/\mathsf{ARO}$$

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), see Lemma A.10. Here, we crucially use the fact that $\Pi\Sigma|_{z_1=0} \in \mathbb{F}(z_2,\varepsilon)$ (this remains so, even in the inductive steps!). For details, see Claim 3.11.

Remark. We point out that we needed to go to ABP, from ARO, as ARO is *not* closed under inverse, i.e. 1/ARO may not necessarily be an ARO. For more details, we refer to Section 3.

Extending to depth-4: Proof idea of Theorem B.1. 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. Firstly, we work with a different Φ. All we need to make sure is: the bottom $\Sigma\wedge$ circuits are 'invertible'; we can directly use the sparse-PIT (univariate) map Ψ [KS01], as $\Sigma\wedge$ are s-sparse.

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 essentially have a product

$$f_{k-1} \;\in\; \overline{(\Pi\Sigma\wedge/\Pi\Sigma\wedge)\cdot(\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge)} \;\subseteq\; (\Pi\Sigma\wedge/\Pi\Sigma\wedge)\cdot(\mathsf{ARO}/\mathsf{ARO}) \;\subseteq\; \mathsf{ABP}/\mathsf{ABP}\,.$$

We crucially use the fact that– 1) $\overline{\Pi\Sigma\wedge} = \Pi\Sigma\wedge$, as $\overline{\Sigma\wedge} = \Sigma\wedge$, and 2) $\overline{\Sigma\wedge\Sigma\wedge} \subseteq \mathsf{ARO}$, again using duality trick (Lemma A.15) and Nisan's characterization (Lemma A.21). Once, we have f_{k-1}, one can similarly interpolate and find f_0. For more details, see Theorem B.1 & its proof.

Proof idea of Theorem 1.2: 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.

$k = 1$ **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:

$$f_1 + \varepsilon \cdot S_1 \;=\; g_1\,,\text{over}\,\mathcal{R}'(\boldsymbol{x},\varepsilon)\;,\text{where}\,f_1 \;:=\; \partial_{z_1}(\Phi(f)/t_2),\;\text{and}\,g_1 \;:=\; \partial_{z_1}(\Phi(T_1)/\tilde{T}_2)\,.$$

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_1-free, or 2) $\mathrm{val}_{z_1}(f_1) \geq d - v_2 - 1$.

When is $\Phi(f)/t_2$ z_1-free? It is when $\Phi(f)/t_2 = \Phi(f)/t_2|_{z_1=0} \in \mathbb{F}(z_2)(\boldsymbol{x})$. However, by Equation 1.6, $\Phi(f)/t_2|_{z_1=0} \in \mathbb{F}(z_2)$. Of course, if $f \neq 0$, it must be a non-zero element in $\mathbb{F}(z_2)$ and checking it is easy, as $\deg(z_2)$ is polynomially bounded.

On the other hand, $\mathrm{val}_{z_1}(f_1) \geq d - v_2 - 1$, implies that $\mathrm{val}_{z_1}(\Phi(f)/t_2) \geq d - v_2$. However, $\mathrm{val}_{z_1}(t_2) \geq v_2$, as $\mathrm{val}_{z_1}(\tilde{T}_2) = v_2$. This means, $\mathrm{val}_{z_1}(\Phi(f)) \geq d$, which is a contradiction, as we assumed that $\deg(f) < d$.

The above discussion summarizes the following important identity testing branching:

$$\Phi(f) \;\neq\; 0\,,\,\text{over}\;\mathcal{R}(\boldsymbol{x}) \quad\Longleftrightarrow\quad f_1 \;\neq\; 0\,,\,\text{over}\;\mathcal{R}'(\boldsymbol{x})\,,\,\text{or}\;\Phi(f)/t_2 \in \mathbb{F}(z_2)\backslash\{0\}\,.$$

We remark that the $z_1 = 0$ substitution is a natural condition as the derivation *forgets* the mod z_1 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 $G(b,Z) \neq 0$; see Claim 4.4. This is crucial to get the final hitting set.

As discussed above, testing $\Phi(f)/t_2|_{z_1=0} \in \mathbb{F}(z_2)\backslash\{0\}$ 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, see Theorem C.3. 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 $\mathbb{F}(\boldsymbol{z})$); 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, $(U/V \cdot P)(\boldsymbol{a} + t \cdot \boldsymbol{b}) \in \mathbb{F}(t, \boldsymbol{z})\backslash\{0\}$. 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. For details, see Lemma C.4.

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_1, 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 Lemma C.4).

However, as seen before, the $z_1 = 0$ substitution, in the k case, i.e. $\lim_{\varepsilon\to 0} \mathsf{Gen}(k,\cdot)|_{z_1=0}$, gives an element of the form ARO/ARO (see Claim 3.11), 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 $\Phi(f)$, see Claim 4.4. 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: Proof idea of Theorem C.12 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 (univariate in z_2) map Ψ [KS01], yielding a different Φ.

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 $\mathrm{dlog}(\Pi\Sigma\Upsilon) \in \Sigma\wedge\Sigma\Upsilon$. So, one inducts on the bloated model $(\Pi\Sigma\Upsilon/\Pi\Sigma\Upsilon) \cdot (\Sigma\wedge\Sigma\Upsilon/\Sigma\wedge\Sigma\Upsilon)$, 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 $\overline{\Sigma\wedge\Sigma\wedge} \subseteq$ ARO. Moreover, we have poly-time hitting set for $\Pi\Sigma\wedge$. Therefore, after combining them (Lem. C.4), 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 $\Sigma\wedge\Sigma\Pi^{[\delta]}$ [For15]. 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 whether there is a non-zero small support trailing monomial in the border or not. For details, see Theorem C.11.

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 $\mathbb{F}^n$.

Thus, once we have a hitting set for $\overline{\Sigma\wedge\Sigma\Pi^{[\delta]}}$, the result follows as we know how to combine hitting set for $\Pi\Sigma\Pi^{[\delta]}$ and $\overline{\Sigma\wedge\Sigma\Pi^{[\delta]}}$, using Lemma C.4, yielding a hitting set $\mathcal{H}_j$, for each branch. Finally taking the union gives the final hitting set. For more details, see Subsection C.3.

Proof idea of Theorem 1.3: Derandomizing log**-variate** $\overline{\Sigma^{[k]}\Pi\Sigma}$**.** We adapt techniques from [FGS18] 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 $f_{k-1} \in (\Pi\Sigma/\Pi\Sigma) \cdot (\mathrm{ARO}/\mathrm{ARO})$. However, we point out that this is not any generic poly-sized ARO but the de-bordering 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 Lemma C.4), 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 [FGS18] 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 $\mathrm{poly}(s)$ 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 [CKW11, 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 Definition 2); this is directly from [For14, 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 [FGS18, Lemma 4]; see the statement in Lemma C.16, and (4) show that the low-cone-size monomials are $\mathrm{poly}(sd)$-many [FGS18, Lemma 5], see Lemma C.15) for the statement. For more details, see Subsection 4.2.

2 Preliminaries

In this section, we describe some of the assumptions and notations used throughout the paper.

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, $f = \sum_{i \geq 0} c_i x^i$, 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* $\mathrm{dlog}_y : R[y] \longrightarrow R(y)$ is defined as $\mathrm{dlog}_y(f) := \partial_y f/f$; here ∂_y denotes the partial derivative wrt variable y. One important property of dlog is that it is *additive* over a product as $\mathrm{dlog}_y(f \cdot g) = \partial_y(fg)/(fg) = (f \cdot \partial_y g + g \cdot \partial_y f)/(fg) = \mathrm{dlog}_y(f) + \mathrm{dlog}_y(g)$. [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 [BIZ18, Def. 2.1].

Definition 2.1 (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.

3 De-bordering depth-3 circuits

In this section we will discuss the proofs of de-bordering result (Theorem 1.1). As seen in Section 1.4, we will use our DiDIL technique. Moreover, we will induct over a more general circuit class as defined below.

Definition 3.1 (Bloated model). *We call a circuit $\mathcal{C} \in \mathsf{Gen}(k,s)$, over the fractional ring $\mathsf{R}(\boldsymbol{x})$, with parameter k and size s, if it computes $f \in \mathsf{R}(\boldsymbol{x})$ where $f = \sum_{i \in [k]} T_i$, such that $T_i = (U_i/V_i) \cdot P_i/Q_i$, with $U_i, V_i, P_i, Q_i \in \mathsf{R}[\boldsymbol{x}]$ such that $U_i, V_i \in \Pi\Sigma$ and $P_i, Q_i \in \Sigma\wedge\Sigma$.*

Further, $\mathrm{size}(\mathcal{C}) = \sum_{i \in [k]} \mathrm{size}(T_i)$, and $\mathrm{size}(T_i) = \mathrm{size}(U_i) + \mathrm{size}(V_i) + \mathrm{size}(P_i) + \mathrm{size}(Q_i)$.

It is easy to see that size-s $\Sigma^{[k]}\Pi\Sigma$ lies in $\mathsf{Gen}(k,s)$, which will be our general model of induction. Here is the main de-bordering theorem for depth-3 circuits.

Theorem 3.2 (De-bordering $\overline{\Sigma^{[k]}\Pi\Sigma}$). *Let $f(\boldsymbol{x}) \in \mathbb{F}[x_1, \ldots, x_n]$, such that f can be computed by a $\overline{\Sigma^{[k]}\Pi\Sigma}$-circuit of size s. Then f is also computable by an ABP (over $\mathbb{F}$), of size $s^{O(k \cdot 7^k)}$.*

Proof of Theorem 3.2. We will use DiDIL technique as discussed in Subsection 1.4. The $k = 1$ case is obvious, as $\overline{\Pi\Sigma} = \Pi\Sigma$ and trivially it has a small ABP. Further, as discussed before, $k = 2$ is already non-trivial. Eventually it involves de-bordering $\overline{\mathsf{Gen}(1, s)}$; as DiDIL technique reduces the $k = 2$ problem to $\overline{\mathsf{Gen}(1, s)}$ and then we interpolate.

Base step: De-bordering $\overline{\mathsf{Gen}(1, s)}$. Let $g(\boldsymbol{x}, \varepsilon) \in R(\boldsymbol{x}, \varepsilon)$ be approximating $f \in R(\boldsymbol{x})$; here R is a commutative ring (the ring will be clear later in the next few paragraphs). We also assume the syntactic degree bound, of the denominator and numerator computing g to be d. Here is the de-bordering result.

Claim 3.3. $\overline{\mathsf{Gen}(1, s)} \in \mathsf{ABP}/\mathsf{ABP}$*, of size* $O(sd^4n)$*, while the syntactic degree blows up to* $O(nd^2)$*.*

Proof. Using Definition 3.1,

$$g(\boldsymbol{x}, \varepsilon) \; =: \; (U(\boldsymbol{x}, \varepsilon)/V(\boldsymbol{x}, \varepsilon)) \cdot P(\boldsymbol{x}, \varepsilon)/Q(\boldsymbol{x}, \varepsilon) \; = \; f(\boldsymbol{x}) + \varepsilon \cdot S(\boldsymbol{x}, \varepsilon) \; ,$$

where $U, V, P, Q \in \mathbb{R}(\varepsilon)[\boldsymbol{x}]$ such that $U, V \in \Pi\Sigma$, $P, Q \in \Sigma\wedge\Sigma$. Let $a_1 := \mathrm{val}_\varepsilon(U)$, $a_2 := \mathrm{val}_\varepsilon(V)$, $b_1 := \mathrm{val}_\varepsilon(P)$ and $b_2 := \mathrm{val}_\varepsilon(Q)$. Extracting the maximum ε-power, we get

$$f \; + \; \varepsilon \cdot S \; = \; \varepsilon^{(a_1 - a_2) + (b_1 - b_2)} \cdot (\tilde{U}/\tilde{V}) \cdot (\tilde{P}/\tilde{Q}) \; ,$$

where $\widetilde{U}, \widetilde{V}, \widetilde{P}, \widetilde{Q} \in R(\varepsilon)[\boldsymbol{x}]$, and their valuations with respect to ε are zero i.e. $\lim_{\varepsilon \to 0} \tilde{U}$ exists (similarly for $\tilde{V}, \tilde{P}, \tilde{Q}$). Since, LHS is well-defined at $\varepsilon = 0$, it must happen that $(a_1 - a_2) + (b_1 - b_2) \geq 0$. If $(a_1 - a_2) + (b_1 - b_2) \geq 1$, then $f = 0$, and we have trivially de-bordered. Therefore, we can assume $(a_1 - a_2) + (b_1 - b_2) = 0$ which implies that

$$f = (\lim_{\varepsilon \to 0} \tilde{U} / \lim_{\varepsilon \to 0} \tilde{V}) \cdot (\lim_{\varepsilon \to 0} \tilde{P} / \lim_{\varepsilon \to 0} \tilde{Q}) \; \in \; (\Pi\Sigma/\Pi\Sigma) \cdot (\mathsf{ARO}/\mathsf{ARO}) \subseteq \mathsf{ABP}/\mathsf{ABP} \; .$$

We have used the fact that $\widetilde{U}, \widetilde{V} \in \Pi\Sigma$ and $\widetilde{P}, \widetilde{Q} \in \Sigma\wedge\Sigma$ of size at most s, over $R(\varepsilon)[\boldsymbol{x}]$. Further, by Lemma A.20 and Lemma A.22, we know that $\overline{\Pi\Sigma} = \Pi\Sigma$ and $\overline{\Sigma\wedge\Sigma} \subseteq \mathsf{ARO}$; therefore f is computable by a ratio of two ABPs of size at most $O(s \cdot d^4 n)$ and the degree gets blown up to atmost $O(nd^2)$. □

Bloat out: Reducing $\overline{\Sigma^{[k]}\Pi\Sigma}$ to de-bordering $\overline{\mathsf{Gen}(k-1, \cdot)}$. Let $f_0 := f$ be an arbitrary polynomial in $\overline{\Sigma^{[k]}\Pi\Sigma}$, approximated by $g_0 \in \mathbb{F}(\varepsilon)[\boldsymbol{x}]$, computed by a depth-3 circuit $\overline{C}$ of size s over $\mathbb{F}(\varepsilon)$, i.e. $g_0 := f_0 + \varepsilon \cdot S_0$. Further, assume that $\deg(f_0) < d_0 := d \leq s$; we keep the parameter d separately, to optimize the complexity later. Here, we also stress that one could think of homogeneous circuits and thus the degree can be assumed to be the syntactic degree as well. Then, $g_0 =: \sum_{i \in [k]} T_{i,0}$, such that $T_{i,0}$ is computable by a $\Pi\Sigma$-circuit of size at most s over $\mathbb{F}(\varepsilon)$. Moreover, define $U_{i,0} := T_{i,0}$ and $V_{i,0} := P_{i,0} := Q_{i,0} = 1$ as the base input case (of $\mathsf{Gen}(1, \cdot)$). As explained in the preliminaries, we do a safe division and derivation for reduction.

Φ *homomorphism.* To ensure invertibility and facilitate derivation, we define a homomorphism

$$\Phi : \mathbb{F}(\varepsilon)[\boldsymbol{x}] \to \mathbb{F}(\varepsilon)[\boldsymbol{x}, z_1, z_2] \, , \quad \text{such that} \;\; x_i \mapsto z_1 \cdot x_i + \Psi(x_i) \, , \text{and } \Psi : x_i \mapsto z_2^i \; .$$

It is easy to observe that $\Psi(T_{i,0}) \neq 0$ for all $i \in [k]$. Further, note that $\deg_{z_2}(\Phi(g_0)) \leq n \cdot d$. We will be working with different ring $\mathcal{R}_i(\boldsymbol{x})$, at i-th step of induction, with $\mathcal{R}_0 := \mathbb{F}(z_2)[z_1]/\langle z_1^d \rangle$, thinking of the z-variables as 'cost-free'. The map Φ can be thought of as a 'shift & scale' map. In a way, choosing random z and then shifting and scaling it back gives the original f. So, our target is to prove the size upper bound for $\Phi(f_0)$ over $\mathcal{R}(\boldsymbol{x})$, and thereby prove upper bound for f_0.

Divide and derive. Let $v_{i,0} := \mathrm{val}_{z_1}(\Phi(T_{i,0}))$. By Φ-map, $v_{i,0} \geq 0$, for each $i \in [k]$. Further, wrt ε-valuation, assume that $\Phi(T_{i,0}) =: \varepsilon^{a_{i,0}} \cdot \tilde{T}_{i,0}$, where $\tilde{T}_{i,0} =: t_{i,0} + \varepsilon \cdot \tilde{t}_{i,0}(\boldsymbol{x}, \boldsymbol{z}, \varepsilon)$ ($t_{i,0} = \tilde{T}_{i,0}|_{\varepsilon=0}$). Note that, $v_{i,0} = \mathrm{val}_{z_1}(\tilde{T}_{i,0})$. Without loss of generality, assume $\min_{i\in[k]} \mathrm{val}_{z_1}(\tilde{T}_{i,0}) = v_{k,0}$, i.e. wrt k, otherwise we can rearrange. Then, we divide $\Phi(g_0)$ by $\tilde{T}_{k,0}$ and derive wrt z_1:

$$\begin{aligned}
\Phi(f_0)/\tilde{T}_{k,0} + \varepsilon \cdot \Phi(S_0)/\tilde{T}_{k,0} &= \varepsilon^{a_{k,0}} + \sum_{i\in[k-1]} \Phi(T_{i,0})/\tilde{T}_{k,0} \quad [\textbf{Divide}] \\
\implies \partial_{z_1}\left(\Phi(f_0)/\tilde{T}_{k,0}\right) + \varepsilon \cdot \partial_{z_1}\left(\Phi(S_0)/\tilde{T}_{k,0}\right) &= \sum_{i\in[k-1]} \partial_{z_1}\left(\Phi(T_{i,0})/\tilde{T}_{k,0}\right) \quad [\textbf{Derive}] \\
&= \sum_{i\in[k-1]} \left(\Phi(T_{i,0})/\tilde{T}_{k,0}\right) \cdot \mathrm{dlog}\left(\Phi(T_{i,0})/\tilde{T}_{k,0}\right) \qquad (3.4) \\
&=: g_1 .
\end{aligned}$$

Definability. Let $\mathcal{R}_1 := \mathbb{F}(z_2)[z_1]/\langle z_1^{d_1}\rangle$, and $d_1 := d_0 - v_{k,0} - 1$. For $i \in [k-1]$, define

$$T_{i,1} := (\Phi(T_{i,0})/\tilde{T}_{k,0}) \cdot \mathrm{dlog}(\Phi(T_{i,0})/\tilde{T}_{k,0}), \text{ and } f_1 := \partial_{z_1}\left(\Phi(f_0)/t_{k,0}\right) .$$

Claim 3.5. *g_1 approximates f_1 correctly, i.e.* $\lim_{\varepsilon\to 0} g_1 = f_1$, *where g_1 (respec. f_1) are well-defined over $\mathcal{R}_1(\varepsilon, \boldsymbol{x})$ (respec. $\mathcal{R}_1(\boldsymbol{x})$).*

Proof. As we divide by the minimum valuation, by Lemma A.17 we have

$$\mathrm{val}_{z_1}(\Phi(T_{i,0})/\tilde{T}_{k,0}) \geq 0 \implies \Phi(T_{i,0})/\tilde{T}_{k,0} \in \mathbb{F}(\boldsymbol{x}, z_2, \varepsilon)[[z_1]] \implies T_{i,1} \in \mathbb{F}(\boldsymbol{x}, z_2, \varepsilon)[[z_1]] .$$

Note that $\mathrm{val}_{z_1}(\Phi(f_0) + \varepsilon \cdot S_0) = \mathrm{val}_{z_1}(\sum_{i\in[k]} \Phi(T_{i,0})) \geq v_{k,0}$. Setting, $\varepsilon = 0$, implies that $\mathrm{val}_{z_1}(\Phi(f_0)) \geq v_{k,0}$ and hence, $\Phi(f_0)/\tilde{T}_{k,0} \in \mathbb{F}(\boldsymbol{x}, z_2, \varepsilon)[[z_1]]$ (by Lemma A.17). Moreover, $(\Phi(f_0)/\tilde{T}_{k,0})|_{\varepsilon=0} = \Phi(f_0)/t_{k,0} \in \mathbb{F}(\boldsymbol{x}, \boldsymbol{z})$. Combining these it follows that

$$\Phi(f_0)/t_{k,0} \in \mathbb{F}(\boldsymbol{x}, z_2)[[z_1]] \implies f_1 \in \mathbb{F}(\boldsymbol{x}, z_2)[[z_1]] .$$

Once we know that each $T_{i,1}$ and f_1 are well-defined power-series, we claim that Eqn. (3.4) holds mod $z_1^{d_0-v_{k,0}-1}$. Note that, $\Phi(f_0) + \varepsilon \cdot \Phi(S_0) = \sum_{i\in[k]} T_i$, holds mod z_1^d. Thus after dividing by the minimum valuation element (with z_1-valuation $v_{k,0}$), it holds mod $z_1^{d_0-v_{k,0}}$; finally after differentiation it must hold mod $z_1^{d_0-v_{k,0}-1}$.

Further, as $\lim_{\varepsilon\to 0} \tilde{T}_{k,0}$ exists, we must have $\partial_{z_1}(\Phi(f_0)/t_{k,0}) = \lim_{\varepsilon\to 0} g_1$; i.e. g_1 approximates f_1 correctly, over $\mathcal{R}_1(\boldsymbol{x})$. □

However, we stress that we also think of these as elements over $\mathbb{F}(\boldsymbol{x}, \boldsymbol{z}, \varepsilon)$, with $\boldsymbol{z}$-degree being 'kept track of' (which could be $> d$). All these different 'lenses' of looking and computing will be important later.

Now what with the lower fanin? The main claim now is to show that– 1) $f_1 \in \overline{\mathsf{Gen}(k-1,\cdot)}$, and 2) assuming we know $\overline{\mathsf{Gen}(k-1,\cdot)}$ has small $\mathsf{ABP}/\mathsf{ABP}$, how to lift it for f_0 (we will show how to generally reduce fanin in the next few paragraphs).

To show that, we will eventually show that each $T_{i,1}$ has small $(\Pi\Sigma/\Pi\Sigma)\cdot(\Sigma\wedge\Sigma/\Sigma\wedge\Sigma)$-circuit over $\mathcal{R}_1(\boldsymbol{x},\varepsilon)$ and then we will interpolate. Once the degree of z is maintained to be *small*, this interpolation would not be costly, which will finally achieve our goal; as polynomially many sum of ratios of ABPs is still a ratio of small ABPs. We remark that these two steps are needed in the general reduction as well, and thus once we show the general inductive reduction, we will illustrate these steps.

Inductive step (j-th step): Reducing $\overline{\mathsf{Gen}(k-j,\cdot)}$ to $\overline{\mathsf{Gen}(k-j-1,\cdot)}$. Suppose, we are at the j-th ($j \geq 1$) step. Our induction hypothesis assumes–

1. $\sum_{i\in[k-j]} T_{i,j} =: g_j$, over $\mathcal{R}_j(\boldsymbol{x},\varepsilon)$, such that it approximates f_j correctly, where $f_j \in \mathcal{R}_j(\boldsymbol{x})$, where $\mathcal{R}_j := \mathbb{F}(z_2)[z_1]/\langle z_1^{d_j}\rangle$.

2. Here, $T_{i,j} =: (U_{i,j}/V_{i,j})\cdot(P_{i,j}/Q_{i,j})$, where $U_{i,j}, V_{i,j} \in \Pi\Sigma$ and $P_{i,j}, Q_{i,j} \in \Sigma\wedge\Sigma$, each in $\mathcal{R}_j(\varepsilon)[\boldsymbol{x}]$. Each can be thought as an element in $\mathbb{F}(\boldsymbol{x},\boldsymbol{z},\varepsilon)\bigcap\mathbb{F}(\boldsymbol{x},z_2,\varepsilon)[[z_1]]$ as well. Assume that the syntactic degree of each denominator and numerator of $T_{i,j}$ is bounded by D_j.

3. $v_{i,j} := \mathrm{val}_{z_1}(T_{i,j}) \geq 0$, for $i \in [k-j]$. Wlog, assume that $\min_i v_{i,j} = v_{k-j,j}$. Moreover, $U_{i,j}|_{z_1=0} \in \mathbb{F}(z_2,\varepsilon)\backslash\{0\}$ (similarly for $V_{i,j}$).

We do like the $j=0$-th step done above, without applying any new homomorphism. Similar to that reduction, we divide and derive to reduce the fanin further by 1.

Di*vide and* **D***erive.* Let $T_{k-j,j} =: \varepsilon^{a_{k-j,j}}\cdot\tilde{T}_{k-j,j}$, where $\tilde{T}_{k-j,j} =: (t_{k-j,j} + \varepsilon\cdot\tilde{t}_{k-j,j})$ is not divisible by ε. Divide $g_j =: f_j + \varepsilon\cdot S_j$, by $\tilde{T}_{k-j,j}$, to get:

$$\begin{aligned} f_j/\tilde{T}_{k-j,j} \;+\; \varepsilon\cdot S_j/\tilde{T}_{k-j,j} \;&=\; \varepsilon^{a_{k-j,j}} \;+\; \sum_{i\in[k-j-1]} T_{i,j}/\tilde{T}_{k-j,j} \\ \implies \partial_{z_1}\left(f_j/\tilde{T}_{k-j,j}\right) \;+\; \varepsilon\cdot\partial_{z_1}\left(S_j/\tilde{T}_{k-j,j}\right) \;&=\; \sum_{i\in[k-j-1]} \partial_{z_1}\left(T_{i,j}/\tilde{T}_{k-j,j}\right) \\ &=\; \sum_{i\in[k-j-1]} \left(T_{i,j}/\tilde{T}_{k-j,j}\right)\cdot \mathrm{dlog}\left(T_{i,j}/\tilde{T}_{k-j,j}\right) \qquad (3.6) \\ &=:\; g_{j+1}\,. \end{aligned}$$

Definability. Let $\mathcal{R}_{j+1} := \mathbb{F}(z_2)[z_1]/\langle z_1^{d_{j+1}}\rangle$, where $d_{j+1} := d_j - v_{k-j,j} - 1$. For $i \in [k-j-1]$, define

$$T_{i,j+1} \;:=\; \left(T_{i,j}/\tilde{T}_{k-j,j}\right)\cdot\mathrm{dlog}\left(T_{i,j}/\tilde{T}_{k-j,j}\right)\;,\text{ and } f_{j+1} \;:=\; \partial_{z_1}(f_j/t_{k-j,j})\;.$$

Claim 3.7 (Induction hypotheses)**.** *g_{j+1} approximates f_{j+1} correctly, i.e.* $\lim_{\varepsilon\to 0} g_{j+1} = f_{j+1}$*, where g_{j+1} (respec. f_{j+1}) are well-defined over $\mathcal{R}_{j+1}(\boldsymbol{x},\varepsilon)$ $\big($respec. , $\mathcal{R}_{j+1}(\boldsymbol{x})\big)$.*

Proof. Remember, f_j and $T_{i,j}$'s are elements in $\mathbb{F}(\boldsymbol{x},\boldsymbol{z},\varepsilon)$ which also belong to $\mathbb{F}(\boldsymbol{x},z_2,\varepsilon)[[z_1]]$. After dividing by the minimum valuation, by similar argument as in Claim 3.5, it follows that $T_{i,j+1}$ and f_{j+1} are elements in $\mathbb{F}(\boldsymbol{x},\boldsymbol{z},\varepsilon)\bigcap\mathbb{F}(\boldsymbol{x},z_2,\varepsilon)[[z_1]]$, proving the second part of induction-hypothesis-(2). In fact, trivially $v_{i,j+1}\geq 0$, for $i\in[k-j-1]$ proving induction-hypothesis-(3).

Similarly, Eqn. (3.6) holds over $\mathcal{R}_{j+1}(\varepsilon,\boldsymbol{x})$, or equivalently mod $z_1^{d_{j+1}}$; this is because of the division by z_1-valuation of $v_{k-j,j}$ and then differentiation, showing induction-hypothesis-(1). So, Eqn. (3.6) being computed mod $z_1^{d_{j+1}}$ is indeed valid. We also mention that using similar argument as in Claim 3.5, $f_{j+1}\in\mathbb{F}(\boldsymbol{x},z_2)[[z_1]]$.

Finally, as f_{j+1} exists, it is obvious to see that $\lim_{\varepsilon\to 0} g_{j+1}=f_{j+1}$. □

Invertibility of $\Pi\Sigma$*-circuits.* Before going into the size analysis, we want to remark that the dlog computation plays a crucial role here and the invertibility of the $\Pi\Sigma$-circuits are crucial for our arguments to go through. The action $\mathrm{dlog}(\Sigma\wedge\Sigma)\in\Sigma\wedge\Sigma/\Sigma\wedge\Sigma$, is of poly-size (Lemma A.13).

What is the action on $\Pi\Sigma$? As dlog distributes the product *additively*, so it suffices to work with $\mathrm{dlog}(\Pi\Sigma)$; and we show that $\mathrm{dlog}(\Pi\Sigma)\in\Sigma\wedge\Sigma$, is of poly-size. For the time being, assume these hold. Then, we simplify

$$T_{i,j}/\tilde{T}_{k-j,j} \;=\; \varepsilon^{-a_{k-j,j}}\cdot(U_{i,j}\cdot V_{k-j,j})/(V_{i,j}\cdot U_{k-j,j})\cdot(P_{i,j}\cdot Q_{k-j,j})/(Q_{i,j}\cdot P_{k-j,j})\;,$$

and its dlog. Therefore, one can define $U_{i,j+1}:=\varepsilon^{-a_{k-j,j}}\cdot U_{i,j}\cdot V_{k-j,j}$; similarly $V_{i,j+1}:=V_{i,j}\cdot U_{k-j,j}$. We stress that dlog computation will produce $\Sigma\wedge\Sigma/\Sigma\wedge\Sigma$ which will further multiply with $P's$ and Q's; it will be clear after the lemma. This directly means: $U_{i,j+1}|_{z_1=0}, V_{i,j+1}|_{z_1=0}\in\mathbb{F}(z_2,\varepsilon)\setminus\{0\}$. This proves the second part of induction-hypothesis-(3).

The overall size blowup. Finally, we show the main step: how to use dlog which is the crux of our reduction. We assume that at the j-th step, $\mathrm{size}(T_{i,j})\leq s_j$ and by assumption $s_0\leq s$.

Claim 3.8 (Size blowup from DiDIL)**.** *$T_{1,k-1}\in(\Pi\Sigma/\Pi\Sigma)\,(\Sigma\wedge\Sigma/\Sigma\wedge\Sigma)$ over $\mathcal{R}_{k-1}(\boldsymbol{x},\varepsilon)$ of size $s^{O(k7^k)}$. It is computed as an element in $\mathbb{F}(\varepsilon,\boldsymbol{x},\boldsymbol{z})$, with syntactic degree (in $\boldsymbol{x},\boldsymbol{z}$) $d^{O(k)}$.*

Proof. Steps $j=0$ vs $j>0$ are slightly different because of the homomorphism Φ. However the main idea of using dlog and expand it as a power-series is the same, which eventually shows that $\mathrm{dlog}(\Pi\Sigma)\in\Sigma\wedge\Sigma$ with a controlled blowup.

For $j=0$, we want to study dlog's effect on $\Phi(T_{i,0})/\tilde{T}_{k,0}$. As dlog distributes over product and thus it suffices to study $\mathrm{dlog}(\ell)$, where $\ell\in\mathcal{R}(\varepsilon)[\boldsymbol{x}]$. However, by the property of Φ, each ℓ must be of the form $\ell=A-z_1B$, where $A\in\mathbb{F}(z_2,\varepsilon)\backslash\{0\}$ and $B\in\mathbb{F}(\varepsilon)[\boldsymbol{x}]$. Using the power series expansion, we have the following, over $\mathcal{R}_1(\boldsymbol{x},\varepsilon)$:

$$\mathrm{dlog}(\ell)=-\frac{\partial_{z_1}(z_1\cdot B)}{A\,(1-z_1\cdot B/A)}=-\frac{\partial_{z_1}(z_1\cdot B)}{A}\cdot\sum_{j=0}^{d_1-1}\left(\frac{z_1\cdot B}{A}\right)^j. \tag{3.9}$$

Note,$\left(\partial_{z_1}(z_1\cdot B)/A\right)$ and $(-z_1\cdot B/A)^j$ have a trivial $\wedge\Sigma$ circuits, each of size $O(s)$. For all j use Lemma A.10 on $\left(\partial_{z_1}(z_1\cdot B)/A\right)\cdot(-z_1\cdot B/A)^j$ to obtain an equivalent $\Sigma\wedge\Sigma$ of size $O(j\cdot d\cdot s)$. Re-indexing gives us the final $\Sigma\wedge\Sigma$ circuit for $\mathrm{dlog}(\ell)$ of size $O(d^3\cdot s)$. We use the fact that $d_1\leq d_0=d$. Here the syntactic degree blowsup to $O(d^2)$.

For $j > 0$, the above equation holds over $\mathcal{R}_j(\boldsymbol{x})$. However, as mentioned before, the degree could be D_j (possibly $> d_j$) of the corresponding A and B. Thus, the overall size after the power-series expansion would be $O(D_j^2 d \,\mathrm{size}(\ell))$ [here again we use that $d_j \le d$].

Effect of dlog on $\Sigma\wedge\Sigma$ is, naturally, more straightforward because it is closed under differentiation, as shown in Lemma A.13. Using Lemma A.13, we obtain $\Sigma\wedge\Sigma/\Sigma\wedge\Sigma$ circuit for $\mathrm{dlog}(P_{i,j})$ of size $O\left(D_j^2 \cdot s_j\right)$. Similar claim can be made for $\mathrm{dlog}(Q_{i,j})$. Also, $\mathrm{dlog}(U_{i,j} \cdot V_{k-j,j}) \in \sum \mathrm{dlog}(\Sigma)$, which could be computed using the above Equation. Thus,

$$\mathrm{dlog}(T_{i,j}/\tilde{T}_{k-j,j}) \in \mathrm{dlog}(\Pi\Sigma/\Pi\Sigma) \pm \Sigma^{[4]}\,\mathrm{dlog}(\Sigma\wedge\Sigma) \subseteq \Sigma\wedge\Sigma + \Sigma^{[4]}\Sigma\wedge\Sigma/\Sigma\wedge\Sigma = \Sigma\wedge\Sigma/\Sigma\wedge\Sigma\,.$$

Here, $\Sigma^{[4]}$ means sum of 4-many expressions. The first containment is by linearization. Express $\mathrm{dlog}(\Pi\Sigma/\Pi\Sigma)$ as a single $\Sigma\wedge\Sigma$-expression of size $O(D_j^2 d_j s_j)$, by summing up the $\Sigma\wedge\Sigma$-expressions obtained from $\mathrm{dlog}(\Sigma)$. Next, there are 4-many $\Sigma\wedge\Sigma/\Sigma\wedge\Sigma$ expressions of size $O(D_j^2 s_j)$ as there are 4-many P's and Q's. Additionally, the syntactic degree of each denominator and numerator of $\Sigma\wedge\Sigma/\Sigma\wedge\Sigma$ grows up to $O(D_j)$. Finally, we club $\Sigma\wedge\Sigma/\Sigma\wedge\Sigma$ expressions (4 of them) to express it as a single $\Sigma\wedge\Sigma/\Sigma\wedge\Sigma$ expression using Lemma A.13, with size blowup of $O(D_j^{12} s_j^4)$. Finally, add the single $\Sigma\wedge\Sigma$ expression of size $O(D_j^3 s_j)$, and degree $O(dD_j)$, to get $O(s_j^5 D_j^{16} d)$ size representation.

Also, we need to multiply with $T_{i,j}/\tilde{T}_{k-j,j}$ which is of the form $(\Pi\Sigma/\Pi\Sigma) \cdot (\Sigma\wedge\Sigma/\Sigma\wedge\Sigma)$, where each $\Sigma\wedge\Sigma$ is basically product of two $\Sigma\wedge\Sigma$ expressions of size s_j and syntantic degree D_j and clubbed together, owing a blowup of $O(D_j s_j^2)$. Hence, multiplying this $(\Pi\Sigma/\Pi\Sigma) \cdot (\Sigma\wedge\Sigma/\Sigma\wedge\Sigma)$-expression with the $\Sigma\wedge\Sigma/\Sigma\wedge\Sigma$ expression obtained from dlog-compuation, gives a size blowup of $s_{j+1} := s_j^7 D_j^{O(1)} d$.

As mentioned before, the main blowup of syntactic degree in the dlog computation could be $O(dD_j)$ and clearing expressions and multiplying the without-dlog expression increases the syntactic degree only by a constant multiple. Therefore, $D_{j+1} := O(dD_j) \implies D_j = d^{O(j)}$. Hence, $s_{j+1} = s_j^7 \cdot d^{O(j)} \implies s_j \le (sd)^{O(j \cdot 7^j)}$. In particular, $s_{k-1} \le s^{O(k \cdot 7^k)}$; here we used that $d \le s$. This calculation quantitatively establishes induction-hypothesis-(2). □

Roadmap to trace back f_0. The above claim established that $g_{k-1} \in \mathsf{Gen}(1,\cdot)$ and approximates f_{k-1} correctly. We also know that $\overline{\mathsf{Gen}(1,\cdot)} \in \mathsf{ABP}/\mathsf{ABP}$, from Claim 3.3. Whence, g_{k-1} having $s^{O(k7^k)}$-size bloated-circuit implies: it can be computed as a ratio of ABPs with size $s^{O(k7^k)} \cdot D_{k-1}^4 \cdot n = s^{O(k7^k)}$, and syntactic degree $n \cdot D_{k-1}^2 = d^{O(k)}$. Now, we recursively 'lift' this quantity, via interpolation, to recover in order, $f_{k-2}, f_{k-3}, \ldots, f_0$; which we originally wanted.

Interpolation: To integrate and limit. As mentioned above, we will interpolate recursively. We know $f_{k-1} = \partial_{z_1}(f_{k-2}/t_{2,k-2})$ has a $\mathsf{ABP}/\mathsf{ABP}$ circuit over $\mathbb{F}(\boldsymbol{x}, \boldsymbol{z})$, i.e. each denominator and numerator is being computed in $\mathbb{F}[\boldsymbol{x}, \boldsymbol{z}]$, and size bounded by $\mathcal{S}_{k-1} := s^{O(k7^k)}$. Here is an important claim about the size of f_{k-2} (we denote it by $\mathcal{S}_{k-2}$).

Claim 3.10 (Tracing back one step). *f_{k-2} can be expressed as $\sum_{i=0}^{d_{k-2}-1}$ $(\mathsf{ABP}/\mathsf{ABP})$ z_1^i, of size $s^{O(k7^k)}$ and syntactic degree $d^{O(k)}$.*

Proof. Let the degree of f_{k-1} (both denominator and numerator) be bounded by $D'_{k-1} := d^{O(k)}$ and further we know that keeping information (of the power series) till mod $z_1^{d_{k-1}}$ suffices. While computing it, it may happen that valuation of each denominator and numerator is > 0, i.e. it is of the form $z_1^{e_1} \cdot (\mathsf{ABP})/z_1^{e_2} \cdot (\mathsf{ABP})$ (e_1, e_2 being valuations wrt $z1$). It must happen that $e_1 \geq e_2$, if it is indeed a power series in z_1; the e_i's are bounded by D'_{k-1}. Furthermore, these ABPs (after dividing by z_1-power) have similar size as z_1 is considered free [think of them being computed over $\mathbb{F}(z_1)[\boldsymbol{x}, z_2]$]. Therefore, $\mathsf{ABP}/\mathsf{ABP}$ can be expressed as $\sum_{i=0}^{d_{k-1}-1} C_{i,k-1} \cdot z_1^i$, by using the inverse identity: $1/(1-z_1) \equiv 1 + \ldots + z_1^{d_{k-1}-1} \bmod z_1^{d_{k-1}}$. Here, each $C_{i,k-1}$ has an $\mathsf{ABP}/\mathsf{ABP}$ of size at most $O(\mathcal{S}_{k-1} \cdot {D'_{k-1}}^2)$; for details see Lemma A.2.

Once we get $f_{k-1} = \sum_{i=0}^{d_{k-1}-1} C_{i,k-1} z_1^i$, definite-integration implies:

$$f_{k-2}/t_{2,k-2} - f_{k-2}/t_{2,k-2}|_{z_1=0} \equiv \sum_{i=1}^{d_{k-1}} (C_{i,k-1}/i) \cdot z_1^i \bmod z_1^{d_{k-1}+1} .$$

The final trick is to get $f_{k-2}/t_{2,k-2}|_{z_1=0}$ and 'reach' f_{k-2}. As, $f_{k-2}/t_{2,k-2} \in \mathbb{F}(\boldsymbol{x}, z_2)[[z_1]]$, substituting $z_1 = 0$ yields an element in $\mathbb{F}(\boldsymbol{x}, z_2)$. Recall the identity:

$$f_{k-2}/t_{2,k-2}|_{z_1=0} = \lim_{\varepsilon\to 0}(T_{1,k-2}/\tilde{T}_{2,k-2}|_{z_1=0} + \varepsilon^{a_{2,k-2}}) \in \lim_{\varepsilon\to 0}\left(\mathbb{F}(z_2,\varepsilon) \cdot (\Sigma\wedge\Sigma/\Sigma\wedge\Sigma) + \varepsilon^{a_{2,k-2}}\right) .$$

Since, $\mathbb{F}(z_2,\varepsilon) \cdot (\Sigma\wedge\Sigma/\Sigma\wedge\Sigma) + \varepsilon^{a_{2,k-2}} \in \Sigma\wedge\Sigma/\Sigma\wedge\Sigma$, over $\mathbb{F}(z_2,\varepsilon)(\boldsymbol{x})$. We know that the limit exists and is ARO/ARO ($\subseteq$ $\mathsf{ABP}/\mathsf{ABP}$) of syntactic degree $d^{O(k)}$ and size $s_{k-1} \cdot d^{O(k)}$. Thus, from the above equation, it follows:

$$f_{k-2}/t_{2,k-2} = f_{k-2}/t_{2,k-2}|_{z_1=0} + \sum_{i=1}^{d_{k-1}} (C_{i,k-1}/i) \cdot z_1^i \in \sum_{i=0}^{d_{k-1}} (\mathsf{ABP}/\mathsf{ABP}) \cdot z_1^i ,$$

of size $d_{k-1} \cdot \mathcal{S}_{k-1} D'^2_{k-1} + s_{k-1} \cdot d^{O(k)}$, and degree $D'_{k-1} + d^{O(k)}$. Lastly,

$$t_{2,k-2} \in \lim_{\varepsilon\to 0} (\Pi\Sigma/\Pi\Sigma) \cdot (\Sigma\wedge\Sigma/\Sigma\wedge\Sigma) \subseteq (\Pi\Sigma/\Pi\Sigma) \cdot (\mathrm{ARO}/\mathrm{ARO}) .$$

Thus, it has size s_{k-2}, by previous Claims and degree bound D_{k-2}. Moreover, we know that $\mathrm{val}_{z_1}(t_{2,k-2}) \geq v_{2,k-2} = d_{k-2} - d_{k-1} - 1$. Thus, multiply $t_{2,k-2}$ and truncate it till $d_{k-2} - 1$. This gives us the blowup: size $\mathcal{S}_{k-2} = d_{k-1} \cdot \mathcal{S}_{k-1} D'^2_{k-1} + s_{k-1} \cdot d^{O(k)}$ and degree $D'_{k-2} = D'_{k-1} + d^{O(k)}$.

So, we get: f_{k-2} has $\sum_{i=0}^{d_{k-2}-1} (\mathsf{ABP}/\mathsf{ABP}) z_1^i$ of size $\mathcal{S}_{k-2} = s^{O(k7^k)}$ and degree $D'_{k-2} = d^{O(k)}$. □

The $z_1 = 0$-evaluation. To trace back further, we imitate the step as above; and get f_j one by one. But we first need a claim about the $z_1 = 0$ evaluation of $f_j/t_{k-j,j}$.

Claim 3.11 (For definite integration). *$f_j/t_{k-j,j}|_{z_1=0} \in$ ARO/ARO $\subseteq$ $\mathsf{ABP}/\mathsf{ABP}$ of size $s^{O(k7^k)}$.*

Proof. Note that, $g_j/\tilde{T}_{k-j,j} = \sum_{i\in[k-j]} T_{i,j}/\tilde{T}_{k-j,j} \in \mathbb{F}(z_2, \boldsymbol{x})[[z_1, \varepsilon]]$, as valuation wrt z_1 respec. ε is non-negative. Therefore,

$$\begin{aligned} f_j/t_{k-j,j}|_{z_1=0} &= \lim_{\varepsilon\to 0} \sum_{i\in[k-j]} T_{i,j}/\tilde{T}_{k-j,j}|_{z_1=0} \\ &= \lim_{\varepsilon\to 0} \sum_{i\in[k-j]} \left(\varepsilon^{-a_{k-j,j}} \cdot (U_{i,j} \cdot V_{k-j,j})/(U_{k-j,j} \cdot V_{i,j}) \cdot (P_{i,j} \cdot Q_{k-j,j})/(P_{k-j,j} \cdot Q_{i,j})\right)|_{z_1=0} \end{aligned}$$

$$\in \lim_{\varepsilon\to 0} \sum_{i\in[k-j]} (\mathbb{F}(z_2,\varepsilon)\cdot \Sigma\wedge\Sigma/\Sigma\wedge\Sigma) = \lim_{\varepsilon\to 0} (\Sigma\wedge\Sigma/\Sigma\wedge\Sigma) \subseteq \text{ARO}/\text{ARO}\,.$$

Here we crucially used induction-hypothesis-(3) part: each $U_{i,j}, V_{i,j}$ at $z_1 = 0$, is an element in $\mathbb{F}(z_2,\varepsilon)$. Also, we used that $\Sigma\wedge\Sigma$ is *closed* under constant-fold multiplication (Lemma A.10). Finally, we take the limit to conclude that $\overline{\Sigma\wedge\Sigma/\Sigma\wedge\Sigma} \subseteq \text{ARO}/\text{ARO}$.

To show the ABP-size upper bound, let us denote the size$(f_j/t_{k-j,j}|_{z_1=0})$ =: S'_j, and the syntactic degree D'_j. We claim that $S'_j = O(s_j^{O(k-j)}\cdot D_j'^4 n)$. Because, we have a sum of $k-j$ many $\Sigma\wedge\Sigma/\Sigma\wedge\Sigma$ expressions each of size s_j; $\Sigma\wedge\Sigma$ is closed under multiplication (Lemma A.10) and $\Sigma\wedge\Sigma$ to ARO conversion introduces exponent 4 in the degree (Lemma A.15). Each time the syntactic degree blowup is only a constant multiple, thus $D'_j := d^{O(k)}$ (which is $\le s^{O(k)}$). Therefore, $S'_j = s^{O(k-j)\cdot j7^j} = s^{O(j(k-j)7^j)} = s^{O(k7^k)}$. Here, we use the fact that $\max_{j\in[k-1]} j(k-j)7^j = (k-1)7^{k-1}$ (see Lemma A.16). This finishes the proof. □

Size blowup. Suppose the ABP-size of f_j is $\mathcal{S}_j$; thus we need to estimate $\mathcal{S}_0$.

We *remark* that we do not need to eliminate division at each tracing-back-step (which we did to obtain f_{k-2}). Since once we have $\sum_{i=0}^{d_j-1}(\mathsf{ABP}/\mathsf{ABP})\cdot z_1^i$, it is easy to integrate (wrt z_1) without any blowup as we already have all the ABP/ABP's in hand (they are z_1-free). The main size blowup ($= S'_j$) happens due to $z_1 = 0$ computation which we calculated above (Claim 3.11). Thus, the final recurrence is $\mathcal{S}_j = \mathcal{S}_{j+1} + S'_j$. This gives $\mathcal{S}_0 = s^{O(k7^k)}$, which is the size of $\Phi(f)$, over $\mathbb{F}(\boldsymbol{z},\boldsymbol{x})$, being computed as an ABP/ABP.

Finally, plugging 'random' $\boldsymbol{z}$, shifting-and-scaling, gives us f; represented as an ABP/ABP of similar size. At the final stage, we eliminate the division-gate which gives us f represented as an ABP of size $s^{O(k7^k)}$. □

Remark. Our proof de-bordered $\mathsf{Gen}(k,s)$, and that too for any field of characteristic $= 0$ or $\ge d$.

4 Blackbox PIT for border depth-3 circuits

We divide the section into two parts. First subsection deals with proving Theorem 1.2, while the second subsection deals with optimally better hitting sets in the log-variate regime.

4.1 Quasi-derandomizing $\overline{\Sigma^{[k]}\Pi\Sigma}$ circuits

Induction step of DiDIL is important to give any meaningful upper bound of circuit complexity. However, hitting set construction demands less— each inductive step of fanin reduction must preserve non-zeroness. Eventually, we exploit this to give an efficient hitting set construction for $\overline{\Sigma^{[k]}\Pi\Sigma}$, and in the process of reducing the top fanin analyse the bloated model $\mathsf{Gen}(k,\cdot)$.

Theorem 4.1 (Efficient hitting set for $\overline{\Sigma^{[k]}\Pi\Sigma}$). *There exists an explicit $s^{O(k\cdot 7^k\cdot\log\log s)}$-time hitting set for $\overline{\Sigma^{[k]}\Pi\Sigma}$-circuits of size s.*

Proof. The basic reduction strategy is same as Section 3. Let $f_0 := f$ be an arbitrary polynomial in $\overline{\Sigma^{[k]}\Pi\Sigma}$, approximated by $g_0 \in \mathbb{F}(\varepsilon)[\boldsymbol{x}]$, computed by a depth-3 circuit $\overline{C}$ of size s over $\mathbb{F}(\varepsilon)$, i.e. $g_0 := f_0 + \varepsilon \cdot S_0$. Further, assume that $\deg(f_0) < d_0 := d \leq s$. Let $g_0 =: \sum_{i\in[k]} T_{i,0}$, such that $T_{i,0}$ is computable by a $\Pi\Sigma$-circuit of size atmost s over $\mathbb{F}(\varepsilon)$. As before, define $\mathcal{R}_0 := \mathbb{F}(z_2)[z_1]/\langle z_1^d \rangle$. Thus, $f_0 + \varepsilon \cdot S_0 = \sum_{i\in[k]} T_{i,0}$, holds over $\mathcal{R}_0(\boldsymbol{x}, \varepsilon)$.

Define $U_{i,0} := T_{i,0}$ and $V_{i,0} := P_{i,0} := Q_{i,0} = 1$ to set the input instance of $\mathsf{Gen}(k,s)$. Of course, we assume that each $T_{i,0} \neq 0$ (otherwise it is a smaller fanin than k).

Φ *homomorphism.* To ensure invertiblity and facilitate derivation, we define the same Φ as in Section 3, i.e. $\Phi : \mathbb{F}(\varepsilon)[\boldsymbol{x}] \to \mathbb{F}(\varepsilon)[\boldsymbol{x}, z_1, z_2]$ such that $x_i \mapsto z_1 \cdot x_i + \Psi(x_i)$, and $\Psi : x_i \mapsto z_2^i$.

0-*th step: Reduction from k to k* − 1. We will use the same notation as in Section 3. We know that g_1 approximates f_1 correctly over $\mathcal{R}_1(\boldsymbol{x}, \varepsilon)$. Rewriting the same, we have

$$f_0 + \varepsilon \cdot S_0 \;=\; \sum_{i\in[k]} T_{i,0}\,, \quad \text{over}\, \mathcal{R}_0(\boldsymbol{x}, \varepsilon) \quad \Longrightarrow \quad f_1 + \varepsilon \cdot S_1 \;=\; \sum_{i\in[k-1]} T_{i,1}\,, \quad \text{over } \mathcal{R}_1(\boldsymbol{x}, \varepsilon)\,. \tag{4.2}$$

Here, $T_{i,1} := (\Phi(T_{i,0})/\tilde{T}_{k,0}) \cdot \mathrm{dlog}(\Phi(T_{i,0})/\tilde{T}_{k,0})$, for $i \in [k-1]$ and $f_1 := \partial_{z_1}(\Phi(f_0)/t_{k,0})$, same as before. Also, we will consider $T_{i,1}$ as an element of $\mathbb{F}(\boldsymbol{x}, \boldsymbol{z}, \varepsilon)$ and keep track of $\deg(\boldsymbol{z})$.

The "iff" condition. Note that the equality in Equation 4.2 over $\mathcal{R}_1(\varepsilon, \boldsymbol{x})$ is only "one-sided". Whereas, to reduce identity testing, we need a necessary and sufficient condition: If $f_0 \neq 0$, we *would like* to claim that $f_1 \neq 0$ (over $\mathcal{R}_1(\boldsymbol{x})$). However, it may not be directly true because of the loss of z_1-free terms of f_0, due to differentiation. Note that $f_1 \neq 0$ implies $\mathrm{val}_{z_1}(f_1) < d =: d_1$. Further, $f_1 = 0$, over $R_1(\boldsymbol{x})$, implies–

either, (1) $\Phi(f_0)/t_{k,0}$ is z_1-free. This implies $\Phi(f_0)/t_{k,0} \in \mathbb{F}(z_2)(\boldsymbol{x})$, which further implies it is in $\mathbb{F}(z_2)$, because z_1-free implies $\boldsymbol{x}$-free, by substituting $z_1 = 0$, by the definition of Φ. Also, note that $f_0, t_{k,0} \neq 0$ implies $\Phi(f_0)/t_{k,0}$ is a *nonzero* element in $\mathbb{F}(z_2)$. Thus, it suffices to check whether $\Phi(f_0)|_{z_1=0} = \Psi(f_0)$ is non-zero or not. Further, the degree of z_2 in $\Psi(f_0)$ is polynomially bounded.

or, (2) $\partial_{z_1}(\Phi(f_0)/t_{k,0}) = z_1^{d_1} \cdot p$ where $p \in \mathbb{F}(z_2)(z_1, \boldsymbol{x})$ s.t. $\mathrm{val}_{z_1}(p) \geq 0$. By simple power series expansion, one can conclude that $p \in \mathbb{F}(z_2, \boldsymbol{x})[[z_1]]$ (Lemma A.17). Hence,

$$\Phi(f_0)/t_{k,0} = z_1^{d_1+1} \cdot \tilde{p}, \text{where } \tilde{p} \in \mathbb{F}(z_2, \boldsymbol{x})[[z_1]] \implies \mathrm{val}_{z_1}(\Phi(f_0)) \geq d\,,$$

a contradiction. Here we used the simple fact that differentiation decreases the valuation by 1.

Conversely, it is obvious that $f_0 = 0$ implies $f_1 = 0$. Thus, we have proved the following:

$$f_0 \neq 0 \text{ over } \mathbb{F}[\boldsymbol{x}] \iff f_1 \neq 0 \text{ over } \mathcal{R}_1(\boldsymbol{x}), \quad \text{or} \quad 0 \neq \Phi(f_0)|_{z_1=0} \in \mathbb{F}(z_2).$$

Recall, Claim 3.8 showed that $T_{i,1} \in (\Pi\Sigma/\Pi\Sigma)\ (\Sigma\wedge\Sigma/\Sigma\wedge\Sigma)$ with a polynomial blowup. Therefore, subject to $z_1 = 0$ test, we have reduced the identity testing problem to $k-1$. We will recurse over this until we reach $k = 1$.

Induction step. Assume that we are at the end of j-th step ($j \geq 1$). Our inductive hypothesis assumes the following invariants:

1. $\sum_{i\in[k-j]} T_{i,j} = f_j + \varepsilon \cdot S_j$ over $\mathcal{R}_j(\varepsilon, \boldsymbol{x})$, where $T_{i,j} \neq 0$ and $\mathcal{R}_j := \mathbb{F}(z_2)[z_1]/\langle z_1^{d_j} \rangle$.

2. Each $T_{i,j} = (U_{i,j}/V_{i,j}) \cdot (P_{i,j}/Q_{i,j})$ where $U_{i,j}, V_{i,j} \in \Pi\Sigma$ and $P_{i,j}, Q_{i,j} \in \Sigma\wedge\Sigma$.

3. $\text{val}_{z_1}(T_{i,j}) \geq 0$, for all $i \in [k-j]$. Moreover, $U_{i,j}|_{z_1=0} \in \mathbb{F}(z_2, \varepsilon) \backslash \{0\}$ (similarly $V_{i,j}$).

4. $f_0 \neq 0$ if and only if : $f_j \neq 0$ over $\mathcal{R}_j(\boldsymbol{x})$, or $\bigvee_{i=1}^{j-1} \, (f_i/t_{k-i,i}|_{z_1=0} \neq 0$, over $\mathbb{F}(z_2)(\boldsymbol{x}))$.

Reducing the problem to $k-j-1$. We will follow the $j=0$ case, *without* applying any homomorphism. Again, this reduction step is exactly the same as before, which yields:

$$f_j + \varepsilon \cdot S_j \;=\; \sum_{i \in [k-j]} T_{i,j}\,,\ \text{over}\ \mathcal{R}_j(\boldsymbol{x}, \varepsilon) \;\Longrightarrow\; f_{j+1} + \varepsilon \cdot S_{j+1} \;=\; \sum_{i \in [k-j-1]} T_{i,j+1}\,,\ \text{over}\ \mathcal{R}_{j+1}(\boldsymbol{x}, \varepsilon)\,. \tag{4.3}$$

Here, $T_{i,j+1} := \left(T_{i,j}/\tilde{T}_{k-j,j}\right) \cdot \text{dlog}(T_{i,j}/\tilde{T}_{k-j,j})$, and $f_{j+1} := \partial_{z_1}(f_j/t_{k-j,j})$, as before.

It remains to show that, all the invariants assumed are still satisfied for $j+1$. The first 3 invariants are already shown in Section 3. The 4-th invariant is the iff condition to be shown below.

The "iff" condition in the induction. The above Equation 4.3 pioneers to reduce from $k-j$-summands to $k-j-1$. But we want an 'iff' condition to efficiently reduce the identity testing. If $f_{j+1} \neq 0$, then $\text{val}_{z_1}(f_{j+1}) < d_{j+1}$. Further, $f_{j+1} = 0$, over $R_{j+1}(\boldsymbol{x})$ implies–

either, (1) $f_j/t_{k-j,j}$ is z_1-free, i.e. $f_j/t_{k-j,j} \in \mathbb{F}(z_2)(\boldsymbol{x})$. Now, if indeed $f_0 \neq 0$, then $t_{k-j,j}$ as well as f_j *must be non-zero* over $\mathbb{F}(z_2)(z_1, \boldsymbol{x})$, by induction hypothesis (assuming they are non-zero over $\mathcal{R}_j(\boldsymbol{x})$). We will eventually show that $f_j/t_{k-j,j}|_{z_1=0}$ has a small ARO/ARO circuit; which helps us to construct a quasi-polynomial size hitting set using Theorem C.3.

or, (2) $\partial_{z_1}(f_j/t_{k-j,j}) = z_1^{d_{j+1}} \cdot p$, where $p \in \mathbb{F}(z_2)(z_1, \boldsymbol{x})$ s.t. $\text{val}_{z_1}(p) \geq 0$. By simple power series expansion, one concludes that $p \in \mathbb{F}(z_2, \boldsymbol{x})[[z_1]]$ (Lemma A.17). Hence,

$$f_j/t_{k-j,j} \in z_1^{d_{j+1}+1} \cdot \tilde{p}, \text{ where } \tilde{p} \in \mathbb{F}(z_2, \boldsymbol{x})[[z_1]] \implies \text{val}_{z_1}(f_j) \geq d_j \implies f_j = 0\,, \text{over}\ \mathcal{R}_j(\boldsymbol{x})\,.$$

Conversely, $f_j = 0$, over $\mathcal{R}_j(\boldsymbol{x})$, implies $\text{val}_{z_1}(f_j/\tilde{T}_{k-j,j}) \geq d_j - v_{k-j,j} \implies \text{val}_{z_1}(\partial_{z_1}(f_j/\tilde{T}_{k-j,j})) \geq d_j - v_{k-j,j} - 1 = d_{j+1} \implies \partial_{z_1}(f_j/\tilde{T}_{k-j,j}) = 0$, over $\mathcal{R}_{j+1}(\varepsilon, \boldsymbol{x})$. Fixing $\varepsilon = 0$ we deduce $f_{j+1} = \partial_{z_1}(f_j/t_{k-j,j}) = 0$.

Thus, we have proved that $f_j \neq 0$ over $\mathcal{R}_j(\boldsymbol{x})$ iff

$$f_{j+1} \neq 0 \ \text{ over } R_{j+1}(\boldsymbol{x})\ , \text{ or},\ \ 0 \neq (f_j/t_{k-j,j})|_{z_1=0} \in \mathbb{F}(z_2)(\boldsymbol{x})\ .$$

This concludes the proof of the 4-th invariant.

Note: In the above substitution ($z_1 = 0$), $\Sigma\wedge\Sigma/\Sigma\wedge\Sigma$ maybe undefined by directly evaluating at numerator and denominator, i.e. $= 0/0$. But we can keep track of the z_1 degree of numerator and denominator, which will be polynomially bounded as seen in Claim 3.8. We can interpolate and cancel the z_1-powers to get the ratio.

Constructing the hitting set. The above discussion has reduced the problem of testing $\Phi(f)$ to testing f_{k-1} or $f_j/t_{k-j,j}|_{z_1=0}$, for $j \in [k-2]$. We know that $f_{k-1} \in (\Pi\Sigma/\Pi\Sigma) \cdot (\text{ARO}/\text{ARO})$, of size $s^{O(k7^k)}$, from Claim 3.8. We obtain the hitting set of $\Pi\Sigma$ from Lemma C.2, and for $\Sigma\wedge\Sigma$ we obtain the hitting set from Theorem C.3 (due to Lemma A.15). Finally we combine the two hitting sets using Lemma C.4 and use the fact that the syntactic degree is bounded by $s^{O(k)}$ to obtain a hitting set $\mathcal{H}_{k-1}$ of size $s^{O(k7^k \log\log s)}$.

However, it remains to show– (1) efficient hitting set for $f_j/t_{k-j,j}|_{z_1=0}$, for $j \in [k-2]$, and most importantly (2) how to translate these hitting sets to that of $\Phi(f)$.

Recall: Claim 3.11 shows that $f_k/t_{k-j,j}|_{z_1=0} \in \mathsf{ARO}/\mathsf{ARO}$, of size $s^{O(k7^k)}$ (over $\mathbb{F}(z_2)(\boldsymbol{x})$). Thus, it has a hitting set $\mathcal{H}_j$ of size $s^{O(k7^k \log\log s)}$ (Theorem C.3).

To translate the hitting set, we need a small property which will bridge the gap of lifting the hitting set to f_0.

Claim 4.4 (Fix $\boldsymbol{x}$)**.** *For $\boldsymbol{a} \in \mathbb{F}^n$, if $f_{j+1}|_{\boldsymbol{x}=\boldsymbol{a}} \neq 0$, over $\mathcal{R}_{j+1}$, and $\mathrm{val}_{z_1}(\tilde{T}_{k-j,j}|_{\boldsymbol{x}=\boldsymbol{a}}) = v_{k-j,j}$, then $f_j|_{\boldsymbol{x}=\boldsymbol{a}} \neq 0$, over $\mathcal{R}_j$.*

Proof. Suppose the hypothesis holds, and $f_j|_{\boldsymbol{x}=\boldsymbol{a}} = 0$, over $\mathcal{R}_j$. Then, $\mathrm{val}_{z_1}(f_j/\tilde{T}_{k-j,j}|_{\boldsymbol{x}=\boldsymbol{a}}) \geq d_j - v_{k-j,j} \implies \mathrm{val}_{z_1}(\partial_{z_1}(f_j/\tilde{T}_{k-j,j})|_{\boldsymbol{x}=\boldsymbol{a}}) \geq d_j - v_{k-j,j} - 1 = d_{j+1} \implies \partial_{z_1}(f_j/\tilde{T}_{k-j,j})|_{\boldsymbol{x}=\boldsymbol{a}} = 0$, over $\mathcal{R}_{j+1}(\boldsymbol{x})$. Fixing $\varepsilon = 0$ we deduce $f_{j+1}|_{\boldsymbol{x}=\boldsymbol{a}} = 0$. This is a contradiction! □

Finally, we have already shown in Section 3 that $\tilde{T}_{k-j,j} \in (\Pi\Sigma/\Pi\Sigma) \cdot (\Sigma\wedge\Sigma/\Sigma\wedge\Sigma)$, and $t_{k-j,j} \in (\Pi\Sigma/\Pi\Sigma) \cdot (\mathsf{ARO}/\mathsf{ARO})$, of size $s^{O(k7^k)}$, which is similar to f_{k-1}. Note: val_{z_1} of a $\Sigma\wedge\Sigma$ again reduces to a $\Sigma\wedge\Sigma$ question.

Joining the dots: The final hitting set. We now have all the ingredients to construct the hitting set for $\Phi(f_0)$. We know $\mathcal{H}_{k-1}$ works for f_{k-1} (as well as $t_{2,k-2}$, because they both are of the same size and belong to $(\Pi\Sigma/\Pi\Sigma) \cdot (\mathsf{ARO}/\mathsf{ARO})$). This lifts to f_{k-2}. But from the 4-th invariant, we know that $\mathcal{H}_{k-2}$ works for the $z_1 = 0$ part. Eventually, lifting this using Claim 4.4, the final hitting set (in $\boldsymbol{x}$) will be $\mathcal{H} := \bigcup_{j\in[k-1]} \mathcal{H}_j$. We remark that we do not need extra hitting set for each $t_{k-j,j}$, because it is already covered by $\mathcal{H}_{k-1}$. We have also kept track of $\deg(\boldsymbol{z})$ which is bounded by $s^{O(k)}$. We use a trivial hitting set for $\boldsymbol{z}$ which does not change the size. Thus, we have successfully constructed a $s^{O(k7^k \log\log s)}$-time hitting set for $\overline{\Sigma^{[k]}\Pi\Sigma}$. □

Remark. This is a PIT for $\overline{\mathsf{Gen}(k,s)}$, and that too for any field of characteristic $= 0$ or $\geq d$.

4.2 Border PIT for log-variate depth-3 circuits

In this section, we prove Theorem 1.3. This proof is dependent on adapting and extending [FGS18] proof, by showing that there is a $\mathrm{poly}(s)$-time hitting set for log-variate $\overline{\Sigma\wedge\Sigma}$-circuits.

Theorem 4.5 (Derandomizing log-variate $\overline{\Sigma\wedge\Sigma}$)**.** *There is a $\mathrm{poly}(s)$-time hitting set for $n = O(\log s)$ variate $\overline{\Sigma\wedge\Sigma}$-circuits of size s.*

Proof sketch. Let $g = f + \varepsilon \cdot Q$, such that $g \in \Sigma\wedge\Sigma$, over $\mathbb{F}(\varepsilon)$, approximates $f \in \overline{\Sigma\wedge\Sigma}$. The idea is the same as [FGS18]— (1) show that f has $\mathrm{poly}(s,d)$ partial derivative space, (2) low partial derivative space implies low cone-size monomials, (3) we can extract low cone-size monomials efficiently, (4) number of low cone-size monomials is $\mathrm{poly}(sd)$-many.

We remark that (2) is direct from [For14, Corollary 4.14] (with origins in [FS13a]); see Theorem C.14. (4) is also directly taken from [FGS18, Lemma 5] once we assume (1); for the full statement we refer to Lemma C.15.

To show (1), we know that g has $\mathrm{poly}(s,d)$ partial-derivative space over $\mathbb{F}(\varepsilon)$. Denote

$$V_\varepsilon := \left\langle \frac{\partial g}{\partial \boldsymbol{x}^{\boldsymbol{a}}} \mid \boldsymbol{a} < \infty \right\rangle_{\mathbb{F}(\varepsilon)}, \text{ and } V := \left\langle \frac{\partial f}{\partial \boldsymbol{x}^{\boldsymbol{a}}} \mid \boldsymbol{a} < \infty \right\rangle_{\mathbb{F}}.$$

Consider the matrix M_ε, where we index the rows by $\partial_{\boldsymbol{x}^{\boldsymbol{a}}}$, while columns are indexed by monomials (say supporting g), and the entries are the operator-values. Suppose, $\dim(V_\varepsilon) =: r \le \mathrm{poly}(s,d)$ (because of $\Sigma\wedge\Sigma$). That means, any $(r+1)$-many polynomials $\frac{\partial g}{\partial \boldsymbol{x}^{\boldsymbol{a}}}$ are linearly dependent. In other words, determinant of any $(r+1)\times(r+1)$ minor of M_ε is 0. Note that $\lim_{\varepsilon\to 0} M_\varepsilon = M$, the corresponding partial-derivative matrix for f. Crucially, the zeroness of the determinant of any $(r+1)\times(r+1)$ minor of M_ε translates to the corresponding $(r+1)\times(r+1)$ submatrix of M as well [one can also think of det as a "continuous" function, yielding this property]. In particular, $\dim(V) \le r \le \mathrm{poly}(s,d)$.

Finally, to show (3), we note that the coefficient extraction lemma [FGS18, Lemma 4] also holds over $\mathbb{F}(\varepsilon)$. Thus, given the circuit of g, we can decide whether the coefficient of $m =: \boldsymbol{x}^{\boldsymbol{a}}$ is zero or not, in $\mathrm{poly}(\mathrm{cs}(m),s,d)$-time; see Lemma C.16. Note: the coefficient is an arbitrary element in $\mathbb{F}(\varepsilon)$; however we are only interested in its non-zeroness, which is merely 'unit-cost' for us.

We only extract monomials with cone-size $\mathrm{poly}(s,d)$ (property (2)) and there are only $\mathrm{poly}(s,d)$ many such monomials. Therefore, we have a $\mathrm{poly}(s)$-time hitting set for $\overline{\Sigma\wedge\Sigma}$. □

Once we have Theorem 4.5, we argue that this polynomial-time hitting set can be used to give a poly-time hitting set for $\overline{\Sigma^{[k]}\Pi\Sigma}$. We restate Theorem 1.3 with proper complexity below.

Theorem 4.6 (Efficient hitting set for log-variate $\overline{\Sigma^{[k]}\Pi\Sigma}$)**.** *There exists an explicit $s^{O(k7^k)}$-time hitting set for $n = O(\log s)$ variate, size-s, $\overline{\Sigma^{[k]}\Pi\Sigma}$ circuits.*

Proof sketch. We proceed similarly as in Subsection 4.1, with same notations. The reduction and branching out remains exactly the same; in the end, we get that $f_{k-1} \in (\Pi\Sigma/\Pi\Sigma)\cdot(\mathrm{ARO}/\mathrm{ARO})$. Crucially, observe that this ARO is not a generic poly-sized ARO; these AROs are de-bordered log-variate $\overline{\Sigma\wedge\Sigma}$ circuits. From Theorem 4.5, we know that there is a $s^{O(k7^k)}$-time hitting set (because of the size blowup, as seen in Section 3). Combining this hitting set with $\Pi\Sigma$-hitting set is easy, by Lemma C.4.

Moreover, $t_{k-j,j}$ are also of the form $(\Pi\Sigma/\Pi\Sigma)\cdot(\mathrm{ARO}/\mathrm{ARO})$, where again these AROs are de-bordered log-variate $\overline{\Sigma\wedge\Sigma}$ circuits and $s^{O(k7^k)}$-time hitting set exists. Therefore, take the union of the hitting sets (as before), each of size $s^{O(k7^k)}$. This gives the final hitting set which is again $s^{O(k7^k)}$-time constructible! □

5 Conclusion & future direction

This work introduces the DiDIL-technique and successfully de-borders as well as derandomizes $\overline{\Sigma^{[k]}\Pi\Sigma}$. Further we extend this to depth-4 as well. 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? Note that we already have quasi-derandomized the class (Theorem C.12).

5. Can we show that constant border-waring rank is polynomially bounded by waring rank, the degree and the number of variables? i.e. $\overline{\Sigma^{[k]}\wedge\Sigma} \subseteq \Sigma\wedge\Sigma$ 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, we have successfully de-bordered and (quasi)-derandomized $\overline{\Sigma^{[k]}\Pi\Sigma}$. Here, we remark that de-bordering did not directly give us a hitting set, since the de-bordering result was more general than the models where explicit hitting sets are known. However, we were still able to do it 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.

Acknowledgement. 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

[AB03] Manindra Agrawal and Somenath Biswas. Primality and identity testing via Chinese remaindering. *J. ACM*, 50(4):429–443, 2003. 46

[AGKS15] Manindra Agrawal, Rohit Gurjar, Arpita Korwar, and Nitin Saxena. Hitting-Sets for ROABP and Sum of Set-Multilinear Circuits. *SIAM J. Comput.*, 44(3):669–697, 2015. 5, 6

[Agr05] Manindra Agrawal. Proving Lower Bounds Via Pseudo-random Generators. In *Proceedings of the 25th International Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2005)*, pages 92–105, 2005. 5

[AGS19] Manindra Agrawal, Sumanta Ghosh, and Nitin Saxena. Bootstrapping variables in algebraic circuits. *Proceedings of the National Academy of Sciences*, 116(17):8107–8118, 2019. Pre-print available at `arXiv:https://www.pnas.org/content/116/17/8107.full.pdf`. 5, 7

[AKS04] Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. PRIMES is in P. *Annals of mathematics*, pages 781–793, 2004. 5

[And20] Robert Andrews. Algebraic Hardness Versus Randomness in Low Characteristic. In *Proceedings of the 35th Annual Computational Complexity Conference (CCC 2020)*, 2020. 5

[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. 4, 5, 6, 8

[AV08] Manindra Agrawal and V. Vinay. Arithmetic Circuits: A Chasm at Depth Four. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2008)*, pages 67–75, 2008. 5

[AW16] Eric Allender and Fengming Wang. On the power of algebraic branching programs of width two. *Computational Complexity*, 25(1):217–253, 2016. 4

[AZGL+18] Zeyuan Allen-Zhu, Ankit Garg, Yuanzhi Li, Rafael Oliveira, and Avi Wigderson. Operator scaling via geodesically convex optimization, invariant theory and polynomial identity testing. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 172–181, 2018. 2

[BC92] Michael Ben-Or and Richard Cleve. Computing Algebraic Formulas Using a Constant Number of Registers. *SIAM J. Comput.*, 21(1):54–58, 1992. 4

[BCC+18] Alessandra Bernardi, Enrico Carlini, Maria Virginia Catalisano, Alessandro Gimigliano, and Alessandro Oneto. The hitchhiker guide to: Secant varieties and tensor decomposition. *Mathematics*, 6(12):314, 2018. 4

[BCG11] Mats Boij, Enrico Carlini, and Anthony Geramita. Monomials as sums of powers: the real binary case. *Proceedings of the American Mathematical Society*, 139(9):3039–3043, 2011. 4

[BCRL79] Dario Bini, Milvio Capovani, Francesco Romani, and Grazia Lotti. $O(n^{2.7799})$ Complexity for n*n Approximate Matrix Multiplication. *Inf. Process. Lett.*, 8(5):234–235, 1979. 2

[BCS13] Peter Bürgisser, Michael Clausen, and Mohammad A Shokrollahi. *Algebraic complexity theory*, volume 315. Springer Science & Business Media, 2013. 7

[BDI21] Markus Bläser, Julian Dörfler, and Christian Ikenmeyer. On the complexity of evaluating highest weight vectors. *Accepted in the 36th Conference on Computational Complexity (CCC 2021)*, 2021. 7

[BFG$^+$19] Peter Bürgisser, Cole Franks, Ankit Garg, Rafael Oliveira, Michael Walter, and Avi Wigderson. Towards a theory of non-commutative optimization: Geodesic 1st and 2nd order methods for moment maps and polytopes. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 845–861. IEEE, 2019. 2

[BG21] Vishwas Bhargava and Sumanta Ghosh. Improved Hitting Set for Orbit of ROABPs. *Electronic Colloquium on Computational Complexity (ECCC)*, 2021. `eccc:TR21-062`. 8

[BGO$^+$18] Peter Bürgisser, Ankit Garg, Rafael Oliveira, Michael Walter, and Avi Wigderson. Alternating minimization, scaling algorithms, and the null-cone problem from invariant theory. *Proceedings of Innovations in Theoretical Computer Science (ITCS 2018)*, 2018. 2

[BI13] Peter Bürgisser and Christian Ikenmeyer. Explicit lower bounds via geometric complexity theory. In *Proceedings of the 45th Annual ACM symposium on Theory of Computing*, pages 141–150, 2013. 2

[BIL$^+$21] Markus Bläser, Christian Ikenmeyer, Vladimir Lysikov, Anurag Pandey, and Frank-Olaf Schreyer. On the Orbit Closure Containment Problem and Slice Rank of Tensors. In Dániel Marx, editor, *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA 2021, Virtual Conference, January 10 - 13, 2021*, pages 2565–2584. SIAM, 2021. 2

[BIM$^+$20] Markus Bläser, Christian Ikenmeyer, Meena Mahajan, Anurag Pandey, and Nitin Saurabh. Algebraic Branching Programs, Border Complexity, and Tangent Spaces. In Shubhangi Saraf, editor, *35th Computational Complexity Conference (CCC 2020)*, volume 169 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 21:1–21:24, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. 3, 4

[Bin80] Dario Bini. Relations between exact and approximate bilinear algorithms. Applications. *Calcolo*, 17(1):87–97, 1980. 2

[BIZ18] Karl Bringmann, Christian Ikenmeyer, and Jeroen Zuiddam. On Algebraic Branching Programs of Small Width. *J. ACM*, 65(5):32:1–32:29, 2018. 4, 9, 16, 44

[BLMW11] Peter Bürgisser, J. M. Landsberg, Laurent Manivel, and Jerzy Weyman. An Overview of Mathematical Issues Arising in the Geometric Complexity Theory Approach to VP$\neq$VNP. *SIAM J. Comput.*, 40(4):1179–1209, 2011. 2

[BMS13] Malte Beecken, Johannes Mittmann, and Nitin Saxena. Algebraic independence and blackbox identity testing. *Inf. Comput.*, 222:2–19, 2013. 5

[BOT88] Michael Ben-Or and Prasoon Tiwari. A deterministic algorithm for sparse multivariate polynomial interpolation. In *Proceedings of the* 20*th* *Annual ACM symposium on Theory of computing (STOC'88)*, pages 301–309, 1988. 6

[BS21] Pranav Bisht and Nitin Saxena. Blackbox identity testing for sum of special ROABPs and its border class. *Journal of Comput. Compl.*, 2021. 5, 6, 7

[Bür04] Peter Bürgisser. The Complexity of Factors of Multivariate Polynomials. *Found. Comput. Math.*, 4(4):369–396, 2004. 2, 3, 4

[Bür20] Peter Bürgisser. Correction To: The Complexity of Factors of Multivariate Polynomials. *Found. Comput. Math.*, 20(6):1667–1668, 2020. 2, 4, 7

[CCG12] Enrico Carlini, Maria Virginia Catalisano, and Anthony V. Geramita. The solution to the Waring problem for monomials and the sum of coprime monomials. *Journal of Algebra*, 370:5–14, 2012. 41

[CKR+20] Prerona Chatterjee, Mrinal Kumar, C Ramya, Ramprasad Saptharishi, and Anamay Tengse. On the Existence of Algebraically Natural Proofs. 61*st* *Annual IEEE Symposium on the Foundations of Computer Science (FOCS 2020)*, 2020. 2

[CKW11] Xi Chen, Neeraj Kayal, and Avi Wigderson. *Partial derivatives in arithmetic complexity and beyond*. Now Publishers Inc, 2011. 15

[CW90] Don Coppersmith and Shmuel Winograd. Matrix Multiplication via Arithmetic Progressions. *J. Symb. Comput.*, 9(3):251–280, 1990. 2

[CW01] Anthony Carbery and James Wright. Distributional and L^q norm inequalities for polynomials over convex bodies in $\mathbb{R}^n$. *Mathematical research letters*, 8(3):233–248, 2001. 6

[DDS21] Pranjal Dutta, Prateek Dwivedi, and Nitin Saxena. Deterministic identity testing paradigms for bounded top-fanin depth-4 circuits. *Accepted in the* 36*th* *Conference on Computational Complexity (CCC 2021)*, 2021. 5, 9

[DL78] Richard A DeMillo and Richard J Lipton. A Probabilistic Remark on Algebraic Program Testing. *Information Processing Letters*, 7(4):193–195, 1978. 5, 46

[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. 4, 5, 8

[DSS18] Pranjal Dutta, Nitin Saxena, and Amit Sinhababu. Discovering the roots: uniform closure results for algebraic classes under factoring. In Ilias Diakonikolas, David Kempe, and Monika Henzinger, editors, *Proceedings of the 50th Annual ACM Symposium on Theory of Computing (STOC 2018)*, pages 1152–1165. ACM, 2018. 5, 9

[DST21] Pranjal Dutta, Nitin Saxena, and Thomas Thierauf. A Largish Sum-Of-Squares Implies Circuit Hardness and Derandomization. In *Proceedings of the 11th Innovations in Theoretical Computer Science Conference (ITCS 2021)*, 2021. 5, 7

[DSY09] Zeev Dvir, Amir Shpilka, and Amir Yehudayoff. Hardness-Randomness Tradeoffs for Bounded Depth Arithmetic Circuits. *SIAM Journal of Computing*, 39(4):1279–1293, 2009. Preliminary version in the *40th Annual ACM Symposium on Theory of Computing (STOC 2008)*. 5

[Edi18] Editors. Open Problems. *Algebraic Methods, Simons Institute for the Theory of Computing*, 2018. Emailed by Igor Carboni Oliveira. 4

[FGS18] Michael A. Forbes, Sumanta Ghosh, and Nitin Saxena. Towards Blackbox Identity Testing of Log-Variate Circuits. In *Proceedings of the 45th International Colloquium on Automata, Languages and Programming (ICALP 2018)*, 2018. 5, 7, 15, 26, 27, 51

[FGT19] Stephen A. Fenner, Rohit Gurjar, and Thomas Thierauf. A deterministic parallel algorithm for bipartite perfect matching. *Commun. ACM*, 62(3):109–115, 2019. 5

[For14] Michael A. Forbes. Polynomial Identity Testing of Read-Once Oblivious Algebraic Branch-ing Programs. *PhD thesis, Massachusetts Institute of Technology*, 2014. 5, 7, 15, 26, 38, 50

[For15] Michael A. Forbes. Deterministic Divisibility Testing via Shifted Partial Derivatives. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2015)*, pages 451–465. IEEE Computer Society, 2015. 5, 14, 47, 48

[For16] Michael 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. 4, 7, 44

[FS13a] Michael A Forbes and Amir Shpilka. Explicit noether normalization for simultaneous conjugation via polynomial identity testing. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 527–542. Springer, 2013. 2, 6, 26

[FS13b] Michael A. Forbes and Amir Shpilka. Quasipolynomial-Time Identity Testing of Non-commutative and Read-Once Oblivious Algebraic Branching Programs. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2013)*, pages 243–252, 2013. 5, 6, 40

[FS18] Michael A. Forbes and Amir Shpilka. A PSPACE construction of a hitting set for the closure of small algebraic circuits. In *Proceedings of the 50th Annual ACM Symposium on Theory of Computing (STOC 2018)*, 2018. 6

[GG20] Zeyu Guo and Rohit Gurjar. Improved Explicit Hitting-Sets for ROABPs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020. 5, 46

[GGOW16] Ankit Garg, Leonid Gurvits, Rafael Oliveira, and Avi Wigderson. A deterministic polynomial time algorithm for non-commutative rational identity testing. In *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 109–117. IEEE, 2016. 2

[Gho19] Sumanta Ghosh. *Low Variate Polynomials: Hitting Set and Bootstrapping*. PhD thesis, PhD thesis. Indian Institute of Technology Kanpur, 2019. 51

[GKKS14] Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Approaching the Chasm at Depth Four. *Journal of the ACM*, 61(6):33:1–33:16, 2014. Preliminary version in the *28th Annual IEEE Conference on Computational Complexity (CCC 2013)*. 47

[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. Preliminary version in the *54th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2013)*. 5

[GKS16] Rohit Gurjar, Arpita Korwar, and Nitin Saxena. Identity Testing for Constant-Width, and Commutative, Read-Once Oblivious ABPs. In Ran Raz, editor, *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, volume 50 of *LIPIcs*, pages 29:1–29:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. 7, 44

[GKS17] Rohit Gurjar, Arpita Korwar, and Nitin Saxena. Identity Testing for Constant-Width, and Any-Order, Read-Once Oblivious Arithmetic Branching Programs. *Theory Comput.*, 13(1):1–21, 2017. 5, 6, 46

[GKSS17] Joshua A Grochow, Mrinal Kumar, Michael Saks, and Shubhangi Saraf. Towards an algebraic natural proofs barrier via polynomial identity testing. *arXiv preprint arXiv:1701.01717*, 2017. 2

[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. 5, 7

[GL19] Fulvio Gesmundo and Joseph M. Landsberg. Explicit Polynomial Sequences with Maximal Spaces of Partial Derivatives and a Question of K. Mulmuley. *Theory of Computing*, 15(3):1–24, 2019. 4

[GMQ16] Joshua A. Grochow, Ketan D. Mulmuley, and Youming Qiao. Boundaries of VP and VNP. In *Proceedings of the 43rd International Colloquium on Automata, Languages and Programming (ICALP 2016)*, pages 34:1–34:14, 2016. 3, 4

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

[GSS19] Zeyu Guo, Nitin Saxena, and Amit Sinhababu. Algebraic Dependencies and PSPACE Algorithms in Approximative Complexity over Any Field. *Theory Comput.*, 15:1–30, 2019. 6

[Guo21] Zeyu Guo. Variety Evasive Subspace Families. *Accepted in the* 36^{th} *Conference on Computational Complexity (CCC 2021)*, 2021. 8

[Gur15] Rohit Gurjar. *Derandomizing PIT for ROABP and Isolation Lemma for Special Graphs*. PhD thesis, Indian Institute of Technology Kanpur, 2015. 43

[HL16] Jesko Hüttenhain and Pierre Lairez. The boundary of the orbit of the 3-by-3 determinant polynomial. *Comptes Rendus Mathematique*, 354(9):931–935, 2016. 3

[HS80] Joos Heintz and Claus-Peter Schnorr. Testing Polynomials which Are Easy to Compute (Extended Abstract). In *Proceedings of the 12th Annual ACM Symposium on Theory of Computing (STOC 1980)*, pages 262–272, 1980. 5

[IQS18] Gábor Ivanyos, Youming Qiao, and K. V. Subrahmanyam. Non-commutative Edmonds' problem and matrix semi-invariants. *Comput. Complex.*, 26(3):717–763, 2018. 2

[Kay12] Neeraj Kayal. An exponential lower bound for the sum of powers of bounded degree polynomials. *Electronic Colloquium on Computational Complexity (ECCC)*, 19:81, 2012. 47

[KI03] Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. In Lawrence L. Larmore and Michel X. Goemans, editors, *Proceedings of the 35th Annual ACM Symposium on Theory of Computing, June 9-11, 2003, San Diego, CA, USA*, pages 355–364. ACM, 2003. 5

[Koi12] Pascal Koiran. Arithmetic circuits: The chasm at depth four gets wider. *Theor. Comput. Sci.*, 448:56–65, 2012. 5

[KRST20] Mrinal Kumar, C Ramya, Ramprasad Saptharishi, and Anamay Tengse. If VNP is hard, then so are equations for it. *arXiv preprint arXiv:2012.07056*, 2020. 2

[KS01] Adam R. Klivans and Daniel A. Spielman. Randomness efficient identity testing of multivariate polynomials. In Jeffrey Scott Vitter, Paul G. Spirakis, and Mihalis Yannakakis, editors, *Proceedings on 33rd Annual ACM Symposium on Theory of Computing, July 6-8, 2001, Heraklion, Crete, Greece*, pages 216–223. ACM, 2001. 6, 12, 14, 45, 46, 49

[KS07] Neeraj Kayal and Nitin Saxena. Polynomial identity testing for depth 3 circuits. *computational complexity*, 16(2):115–138, 2007. 4, 5

[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. 4

[KS09] Neeraj Kayal and Shubhangi Saraf. Blackbox polynomial identity testing for depth 3 circuits. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 198–207. IEEE, 2009. 8

[KS19] Mrinal Kumar and Ramprasad Saptharishi. Hardness-randomness tradeoffs for algebraic computation. *Bulletin of EATCS*, 3(129), 2019. 5

[KSS14] Swastik Kopparty, Shubhangi Saraf, and Amir Shpilka. Equivalence of Polynomial Identity Testing and Deterministic Multivariate Polynomial Factorization. In *Proceedings of the 29th Annual IEEE Conference on Computational Complexity (CCC 2014)*, pages 169–180. IEEE Computer Society, 2014. 5

[KST19] Mrinal Kumar, Ramprasad Saptharishi, and Anamay Tengse. Near-optimal Bootstrapping of Hitting Sets for Algebraic Circuits. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2019)*. SIAM, 2019. 5, 7

[Kum20] Mrinal Kumar. On the Power of Border of Depth-3 Arithmetic Circuits. *ACM Trans. Comput. Theory*, 12(1):5:1–5:8, 2020. 4, 8

[LL89] Thomas Lehmkuhl and Thomas Lickteig. On the order of approximation in approximative triadic decompositions of tensors. *Theoretical Computer Science*, 66(1):1–14, 1989. 7

[LO15] Joseph M. Landsberg and Giorgio Ottaviani. New Lower Bounds for the Border Rank of Matrix Multiplication. *Theory Comput.*, 11:285–298, 2015. 2

[Lov79] László Lovász. On determinants, matchings, and random algorithms. In *Fundamentals of Computation Theory (FCT 1979)*, pages 565–574, 1979. 5

[Mah13] Meena Mahajan. Algebraic Complexity Classes. *CoRR*, abs/1307.3863, 2013. Preprint available at `arXiv:1307.3863`. 2

[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. 2

[MS21] Dori Medini and Amir Shpilka. Hitting Sets and Reconstruction for Dense Orbits in VP_e and $\Sigma\Pi\Sigma$ Circuits. *36th Computational Complexity Confernece (CCC 2021)*, 2021. 5, 6, 8

[Muk16] Partha Mukhopadhyay. Depth-4 Identity Testing and Noether's Normalization Lemma. In *Proceedings of the 11th International Computer Science Symposium in Russia (CSR 2016)*, 2016. 6

[Mul10] Ketan Mulmuley. Geometric complexity theory VI: The flip via positivity. *arXiv preprint arXiv:0704.0229*, 2010. 3

[Mul12a] Ketan Mulmuley. Geometric Complexity Theory V: Equivalence between Blackbox Derandomization of Polynomial Identity Testing and Derandomization of Noether's Normalization Lemma. In *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS 2012)*, pages 629–638, 2012. 2, 3, 6

[Mul12b] Ketan D Mulmuley. The GCT program toward the P vs. NP problem. *Communications of the ACM*, 55(6):98–107, 2012. 2, 3

[Mum95] David Mumford. *Algebraic geometry I: complex projective varieties*. Springer Science & Business Media, 1995. 3

[MVV87] Ketan Mulmuley, Umesh V. Vazirani, and Vijay V. Vazirani. Matching is as easy as matrix inversion. *Comb.*, 7(1):105–113, 1987. 5

[Nis91] Noam Nisan. Lower bounds for non-commutative computation. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing (STOC 1991)*, pages 410–418, 1991. Available on `citeseer:10.1.1.17.5067`. 4, 7, 40

[Niv69] Ivan Niven. Formal Power Series. *The American Mathematical Monthly*, 76(8):871–889, 1969. 9

[NW94] Noam Nisan and Avi Wigderson. Hardness vs Randomness. *Journal of Computer and System Sciences*, 49(2):149–167, 1994. Available on `citeseer:10.1.1.83.8416`. 5

[Ore22] Øystein Ore. Über höhere kongruenzen. *Norsk Mat. Forenings Skrifter*, 1(7):15, 1922. 5, 46

[PS20] Shir Peleg and Amir Shpilka. A generalized Sylvester-Gallai type theorem for quadratic polynomials. In 35^{th} *Computational Complexity Conference (CCC'20)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2020. 5

[PS21] Shir Peleg and Amir Shpilka. Polynomial time deterministic identity testing algorithm for $\sum^{[3]}\prod\sum\prod^{[2]}$ circuits via Edelstein-Kelly type theorem for quadratic polynomials. In 53^{rd} *Annual ACM symposium on Theory of computing (STOC'21)*, 2021. 5

[Sap13] Ramprasad Saptharishi. *Unified Approaches to Polynomial Identity Testing and Lower Bounds*. PhD thesis, Chennai Mathematical Institute, 2013. 46

[Sap19] Ramprasad Saptharishi. A survey of lower bounds in arithmetic circuit complexity. *Github Survey*, 2019. 38

[Sax08] Nitin Saxena. Diagonal Circuit Identity Testing and Lower Bounds. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP 2008)*, pages 60–71, 2008. 4, 7, 42

[Sax14] Nitin Saxena. Progress on polynomial identity testing-II. In *Perspectives in Computational Complexity*, pages 131–146. Springer, 2014. 5

[Sch80] J. T. Schwartz. Fast Probabilistic Algorithms for Verification of Polynomial Identities. *J. ACM*, 27(4):701–717, October 1980. 5, 46

[Shp19] Amir Shpilka. Sylvester-Gallai type theorems for quadratic polynomials. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 1203–1214, 2019. 5

[Sin19] Amit Kumar Sinhababu. *Power series in complexity: Algebraic Dependence, Factor Conjecture and Hitting Set for Closure of VP*. PhD thesis, PhD thesis, Indian Institute of Technology Kanpur, 2019. 9

[SS11] Nitin Saxena and C. Seshadhri. An Almost Optimal Rank Bound for Depth-3 Identities. *SIAM J. Comput.*, 40(1):200–224, 2011. 6, 8

[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. 4, 5, 6, 8

[SS13] Nitin Saxena and C. Seshadhri. From Sylvester-Gallai configurations to rank bounds: Improved blackbox identity test for depth-3 circuits. *J. ACM*, 60(5):33:1–33:33, 2013. 8

[SSS13] Chandan Saha, Ramprasad Saptharishi, and Nitin Saxena. A case of depth-3 identity testing, sparse factorization and duality. *Computational Complexity*, 22(1):39–69, 2013. 8

[ST21] Chandan Saha and Bhargav Thankey. Hitting Sets for Orbits of Circuit Classes and Polynomial Families. *Electron. Colloquium Comput. Complex.*, 2021. 8

[Str74] Volker Strassen. Polynomials with Rational Coefficients Which are Hard to Compute. *SIAM J. Comput.*, 3(2):128–149, 1974. 2

[SY10] Amir Shpilka and Amir Yehudayoff. Arithmetic Circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science*, 5:207–388, March 2010. 2, 5, 38

[Syl52] James Joseph Sylvester. *On the principles of the calculus of forms*. 1852. 4

[Tav15] Sébastien Tavenas. Improved bounds for reduction to depth 4 and depth 3. *Information and Computation*, 240:2–11, 2015. MFCS 2013. 5

[Val79] Leslie G. Valiant. Completeness Classes in Algebra. In *Proceedings of the 11h Annual ACM Symposium on Theory of Computing*, pages 249–261, 1979. 2

[VSBR83] Leslie G. Valiant, Sven Skyum, S. Berkowitz, and Charles Rackoff. Fast Parallel Computation of Polynomials Using Few Processors. *SIAM Journal of Computing*, 12(4):641–644, 1983. Preliminary version in the *6th Internationl Symposium on the Mathematical Foundations of Computer Science (MFCS 1981)*. 5

[Zip79] Richard Zippel. Probabilistic algorithms for sparse polynomials. In *International symposium on symbolic and algebraic manipulation*, pages 216–226. Springer, 1979. 5, 46

A Basics of algebraic complexity

We will give a brief definition of various computational models and tools used in our results. Interested readers can refer [SY10, For14, Sap19] for more refined versions.

Algebraic Circuits, defined over a field $\mathbb{F}$, are directed acyclic graphs with a unique root node. The leaf nodes of the graph is labelled by variables or field constants and internal nodes are either labelled with $+$ or $\times$. Further the edges can bear field constants. The output of the circuit, through root, is the polynomial it computes. The *size* and *depth* of circuit is the size and depth of the underlying graph.

Definition A.1 (Algebraic Branching Program (ABP))**.** *ABP is a computational model which is described using a layered graph with a source vertex s and a sink vertex t. All edges connect vertices* *from layer i to $i+1$. Further, edges are labelled by univariate polynomials. The polynomial computed by the ABP is defined as*

$$f = \sum_{\text{path } \gamma: s \rightsquigarrow t} \mathrm{wt}(\gamma)$$

where $\mathrm{wt}(\gamma)$ is product of labels over the edges in path γ. Number of layers (Δ) defines the *depth* and the maximum number of vertices in any layer (w) defines the *width* of an ABP. The *size* (s) of an ABP is the sum of the graph-size and the degree of the univariate polynomials that label. If d is the maximum degree of univariates then $s \leq dw^2\Delta$; in fact, we will take the latter as the ABP-size bound in our calculations.

We remark that ABP is closed under addition and multiplication, which is straightforward from the definition. In fact, we also need to eliminate division in ABPs. Here is an important lemma stated below.

Lemma A.2 (Strassen's division elimination)**.** *Let $g(\boldsymbol{x}, y)$ and $h(\boldsymbol{x}, y)$ be computed by ABPs of size s and degree $< d$. Further, assume $h(\boldsymbol{x}, 0) \neq 0$. Then, $g/h \mod y^d$ can be written as $\sum_{i=0}^{d-1} C_i \cdot y^i$, where each C_i is of the form ABP/ABP of size $O(sd^2)$.*

Moreover, in case g/h is a polynomial, then it has an ABP of size $O(sd^2)$.

Proof. ABPs are closed under multiplication, which makes interpolation, wrt y, possible. Interpolating the coefficient C_i, of y^i, gives a sum of d ABP/ABP's; which can be rewritten as a single ABP/ABP of size $O(sd^2)$.

Next, assume that g/h is a polynomial. For a random $(\boldsymbol{a}, a_0) \in \mathbb{F}^{n+1}$, write $h(\boldsymbol{x}+\boldsymbol{a}, y+a_0) =: h(\boldsymbol{a}, a_0) - \tilde{h}(\boldsymbol{x}, y)$ and define $g' := g(\boldsymbol{x}+\boldsymbol{a}, y+a_0)$. Clearly $0 \neq h(\boldsymbol{a}, a_0) \in \mathbb{F}$ and $\tilde{h} \in \langle \boldsymbol{x}, y \rangle$. Of course, $\tilde{h}$ has a small ABP. Using the inverse identity in $\mathbb{F}[[\boldsymbol{x}, y]]$, we have $g(\boldsymbol{x}+\boldsymbol{a}, y+a_0)/h(\boldsymbol{x}+\boldsymbol{a}, y+a_0) =$

$$(g'/h(\boldsymbol{a}, a_0))/(1 - \tilde{h}/h(\boldsymbol{a}, a_0)) \equiv (g'/h(\boldsymbol{a}, a_0)) \cdot \left(\sum_{0 \leq i < d} (\tilde{h}/h(\boldsymbol{a}, a_0))^i \right) \mod \langle \boldsymbol{x}, y \rangle^d .$$

Note that, the degree blowsup in the above summands to $O(d^2)$ and the ABP-size is $O(sd)$. ABPs are closed under addition/ multiplication; thus, we get an ABP of size $O(sd^2)$ for the polynomial $g(\boldsymbol{x}+\boldsymbol{a}, y+a_0)/h(\boldsymbol{x}+\boldsymbol{a}, y+a_0)$. This implies the ABP-size for g/h as well. □

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

Definition A.3 (Read-once Oblivious Algebraic Branching Program (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 A.4 (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.*

A.1 Properties of any-order ROABP (ARO)

We will start with defining the *partial coefficient space* of a polynomial f to 'characterise' the width of ARO. We can work over any field $\mathbb{F}$.

Let $A(\boldsymbol{x})$ be a polynomial over $\mathbb{F}$ in n variables with individual degree d. Denote the set $M := \{0, \ldots, d\}^n$. Note that, one can write $A(\boldsymbol{x})$ as

$$A(\boldsymbol{x}) \;=\; \sum_{\boldsymbol{a} \in M} \mathrm{coef}_A(\boldsymbol{x}^{\boldsymbol{a}}) \cdot \boldsymbol{x}^{\boldsymbol{a}} \,.$$

Consider a partition of the variables $\boldsymbol{x}$ into two parts $\boldsymbol{y}$ and $\boldsymbol{z}$, with $|\boldsymbol{y}| = k$. Then, $A(\boldsymbol{x})$ can be viewed as a polynomial in variables $\boldsymbol{y}$, where the coefficients are polynomials in $\mathbb{F}[\boldsymbol{z}]$. For monomial $\boldsymbol{y}^{\boldsymbol{a}}$, let us denote the coefficient of $\boldsymbol{y}^{\boldsymbol{a}}$ in $A(\boldsymbol{x})$ by $A_{(\boldsymbol{y},\boldsymbol{a})} \in \mathbb{F}[\boldsymbol{z}]$. The coefficient $A_{(\boldsymbol{y},\boldsymbol{a})}$ can also be expressed as a partial derivative $\partial A / \partial \boldsymbol{y}^{\boldsymbol{a}}$, evaluated at $\boldsymbol{y} = \boldsymbol{0}$ (and multiplied by an appropriate constant), see [FS13b, Section 6]. Moreover, we can also write $A(\boldsymbol{x})$ as

$$A(\boldsymbol{x}) \;=\; \sum_{\boldsymbol{a} \in \{0,\ldots,d\}^k} A_{(\boldsymbol{y},\boldsymbol{a})} \cdot \boldsymbol{y}^{\boldsymbol{a}} \,.$$

One can also capture the space by the coefficient matrix (also known as the partial derivative matrix) where the rows are indexed by monomials p_i from $\boldsymbol{y}$, columns are indexed by monomials q_j from $\boldsymbol{z} = \boldsymbol{x} \backslash \boldsymbol{y}$ and (i, j)-th entry of the matrix is $\mathrm{coef}_{p_i \cdot q_j}(f)$.

The following lemma formalises the connection between ARO width and dimension of the coefficient space (or the rank of the coefficient matrix).

Lemma A.5 ([Nis91])**.** *Let $A(\boldsymbol{x})$ be a polynomial of individual degree d, computed by an* ARO *of width w. Let $k \leq n$ and $\boldsymbol{y}$ be any prefix of length k of $\boldsymbol{x}$. Then*

$$\dim_{\mathbb{F}}\{A_{(\boldsymbol{y},\boldsymbol{a})} \mid \boldsymbol{a} \in \{0, \ldots, d\}^k\} \;\leq\; w \,.$$

We remark that the original statement was for a fixed variable order. Since, ARO affords any-order, the above holds for any-order as well. The following lemma is the converse of the above lemma and shows us that the dimension of the coefficient space is rightly captured by the width.

Lemma A.6 (Converse lemma [Nis91])**.** *Let $A(\boldsymbol{x})$ be a polynomial of individual degree d with $\boldsymbol{x} = (x_1, \ldots, x_n)$, such that for some w, for any $1 \leq k \leq n$, and $\boldsymbol{y}$, any-order-prefix of length k, we have*

$$\dim_{\mathbb{F}}\{A_{(\boldsymbol{y},\boldsymbol{a})} \mid \boldsymbol{a} \in \{0, \ldots, d\}^k\} \;\leq\; w \,.$$

Then, there exists an ARO *of width w for $A(\boldsymbol{x})$.*

With the width characterisation, we can show that ARO is closed under addition and multiplication.

Lemma A.7 (ARO closed under addition)**.** *Consider polynomials $f_i \in \mathbb{F}[\boldsymbol{x}]$ of degree d_i, computable by* ARO *of size s_i, for $i \in [k]$. Then $\sum_i f_i$ is computable by* ARO *of size at most $\sum_i s_i$.*

Proof sketch. Fix $\boldsymbol{y}$ as some any-order prefix, of length ℓ, of $\boldsymbol{x}$. Let $V_i := \langle (f_i)_{(\boldsymbol{y},\boldsymbol{a})} \mid \boldsymbol{a} \in \{0, \ldots, d\}^{\ell} \rangle_{\mathbb{F}}$, the vector space spanned by the coefficient polynomials. We know that $\dim V_i \leq w_i \leq s_i$. Let $f := \sum_i f_i$ and V is the corresponding coefficient-space. Note that V is a subspace of $V_1 + \ldots + V_k$. Further, as dimension is subadditive, it directly follows that $\dim(V) \leq \sum_{i \in [k]} s_i$.

As this holds for any variable order and any prefix, using the converse Lemma A.6, we conclude the desired result. □

The next lemma establishes that ARO is closed under *constant*-fold multiplication as the size blowup is multiplicative.

Lemma A.8 (ARO is closed under multiplication). *Consider polynomials $f_i \in \mathbb{F}[\boldsymbol{x}]$ of degree d_i, computable by* ARO *of width s_i, for $i \in [k]$. Then $\prod_i f_i$ is computable by* ARO *of width at most $\prod_i s_i$.*

Proof sketch. We will prove this for two polynomials f_1 and f_2 which suffices to generalize to any k. Let V_1 and V_2 be the correponding spaces for f_1 and f_2. Fix $\boldsymbol{y}$, any-order-prefix of ℓ-length, (wrt which we want to show the dimension upper bound for $f_1 \cdot f_2$). Further, suppose $(f_1)_{(\boldsymbol{y},\boldsymbol{a}_1)}, \ldots, (f_1)_{(\boldsymbol{y},\boldsymbol{a}_p)}$ are the basis elements for V_1, while $(f_2)_{(\boldsymbol{y},\boldsymbol{b}_1)}, \ldots, (f_2)_{(\boldsymbol{y},\boldsymbol{b}_q)}$ are the basis elements for V_2 where $p \le w_1, q \le w_2$. We multiply them to get

$$f_1 \cdot f_2 = \left(\sum_{\boldsymbol{a} \in \{0,\ldots,d_1\}^\ell} (f_1)_{(\boldsymbol{y},\boldsymbol{a})} \cdot \boldsymbol{y}^{\boldsymbol{a}} \right) \cdot \left(\sum_{\boldsymbol{b} \in \{0,\ldots,d_2\}^\ell} (f_2)_{(\boldsymbol{y},\boldsymbol{b})} \cdot \boldsymbol{y}^{\boldsymbol{b}} \right) .$$

The above equation implies that $(f_1)_{(\boldsymbol{y},\boldsymbol{a}_i)} \cdot (f_2)_{(\boldsymbol{y},\boldsymbol{b}_j)}$, for $i \in [p], j \in [q]$ make the basis for the coefficient-space of $f_1 \cdot f_2$, which has dimension at most $w_1 \cdot w_2$ (wrt $\boldsymbol{y}$). We also note that $\max(\boldsymbol{a}+\boldsymbol{b}) = d_1 + d_2$, the maximum individual degree of $f_1 \cdot f_2$.

Doing this for every $\boldsymbol{y}$, the 'converse' Lemma A.6 implies an ARO of size $(w_1 w_2)^2 n(d_1+d_2)$ and width $\le s_1 s_2$. □

A.2 Properties of depth-3 diagonal circuits

In this section we will discuss various properties of $\Sigma\wedge\Sigma$ circuits and basic waring-rank. The corresponding bloated model is $\Sigma\wedge\Sigma/\Sigma\wedge\Sigma$, that computes elements of the form f/g, where $f, g \in \Sigma\wedge\Sigma$. The following lemma gives us a sum of powers representation of monomial. For proofs see [CCG12, Proposition 4.3].

Lemma A.9 (Waring identity for a monomial [CCG12]). *Let $M = x_1^{b_1} \cdots x_k^{b_k}$, where $1 \le b_1 \le \cdots \le b_k$, and* roots of unity *$\mathcal{Z}(i) := \{z \in \mathbb{C} : z^{b_i+1} = 1\}$. Then,*

$$M = \sum_{\varepsilon(i) \in \mathcal{Z}(i) : i=2,\cdots,k} \gamma_{\varepsilon(2),\ldots,\varepsilon(k)} \cdot (x_1 + \varepsilon(2)x_2 + \ldots + \varepsilon(k)x_k)^d \ ,$$

where $d := \deg(M) = b_1 + \cdots + b_k$, and $\gamma_{\varepsilon(2),\cdots,\varepsilon(k)}$ are scalars ($\mathrm{rk}(M) :=$) $\prod_{i=2}^k (b_i+1)$ many.

Remark. For fields other than $\mathbb{F} = \mathbb{C}$: We can go to a small extension (at most d^k), for a monomial of degree d, to make sure that $\varepsilon(i)$ exists.

Using this, we show that $\Sigma\wedge\Sigma$ is *closed* under *constant*-fold multiplication.

Lemma A.10 ($\Sigma\wedge\Sigma$ closed under multiplication). *Let $f_i \in \mathbb{F}[\boldsymbol{x}]$, of syntactic degree $\le d_i$, be computed by a $\Sigma\wedge\Sigma$ circuit of size s_i, for $i \in [k]$. Then, $f_1 \cdots f_k$ has $\Sigma\wedge\Sigma$ circuit of size $O((d_2+1)\cdots(d_k+1)\cdot s_1 \cdots s_k)$.*

Proof. Let $f_i =: \sum_j \ell_{ij}^{e_{ij}}$; by assumption $e_{ij} \le d_i$. Each summand of $\prod_i f_i$ after expanding can be expressed as $\Sigma\wedge\Sigma$ using Lemma A.9 of size at most $(d_2+1)\cdots(d_k+1)\cdot \left(\sum_{i\in[k]} \mathrm{size}(\ell_{ij_i})\right)$.

Summing up, for all $s_1 \cdots s_k$ many products, gives the upper bound. □

Remark. The above lemma, and its proof, hold good for the more general $\Sigma\wedge\Sigma\wedge$ circuits.

Using the additive and multiplicative closure of $\Sigma\wedge\Sigma$, we can show that $\Sigma\wedge\Sigma/\Sigma\wedge\Sigma$ is closed under constant-fold addition.

Lemma A.11 ($\Sigma\wedge\Sigma/\Sigma\wedge\Sigma$ closed under addition). *Let $f_i \in \mathbb{F}[\boldsymbol{x}]$, of syntactic degree d_i, be computable by $\Sigma\wedge\Sigma/\Sigma\wedge\Sigma$ of size s_i, for $i \in [k]$. Then, $\sum_{i\in[k]} f_i$ has a $(\Sigma\wedge\Sigma/\Sigma\wedge\Sigma)$ representation of size $O((\prod_i d_i)\cdot\prod_i s_i)$.*

Proof. Let $f_i =: u_{i1}/u_{i2}$, where $u_{ij} \in \Sigma\wedge\Sigma$ of size at most s_i. Then

$$f \;=\; \sum_{i\in[k]} f_i \;=\; \left(\sum_{i\in[k]} u_{i1} \prod_{j\neq i} u_{j2}\right) \Big/ \left(\prod_{i\in[k]} u_{i2}\right).$$

Use Lemma A.10 on each product-term in the numerator to obtain $\Sigma\wedge\Sigma$ of size $O((\prod_i d_i)\cdot\prod_i s_i)$. Trivially, $\Sigma\wedge\Sigma$ is closed under addition; so the size of the numerator is $O((\prod_i d_i)\cdot\prod_i s_i)$. Similar argument can be given for the denominator. □

Remark. The above holds for $\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge$ circuits as well.

Using a simple interpolation, the coefficient of y^e can be extracted from $f(\boldsymbol{x},y) \in \Sigma\wedge\Sigma$ again as a small $\Sigma\wedge\Sigma$ representation.

Lemma A.12 ($\Sigma\wedge\Sigma$ coefficient extraction). *Let $f(\boldsymbol{x},y) \in \mathbb{F}[\boldsymbol{x}][y]$ be computed by a $\Sigma\wedge\Sigma$ circuit of size s and degree d. Then, $\text{coef}_{y^e}(f) \in \mathbb{F}[\boldsymbol{x}]$ is a $\Sigma\wedge\Sigma$ circuit of size $O(sd)$, over $\mathbb{F}[\boldsymbol{x}]$.*

Proof sketch. Let $f =: \sum_i \alpha_i \cdot \ell_i^{e_i}$, with $e_i \le s$ and $\deg_y(f) \le d$. Thus, write $f =: \sum_{i=0}^{d} f_i \cdot y^i$, where $f_i \in \mathbb{F}[\boldsymbol{x}]$. Interpolate using $(d+1)$-many distinct points $y \mapsto \alpha \in \mathbb{F}$, and conclude that f_i has a $\Sigma\wedge\Sigma$ circuit of size $O(sd)$. □

Like coefficient extraction, differentiation of $\Sigma\wedge\Sigma$ circuit is easy too.

Lemma A.13 ($\Sigma\wedge\Sigma$ differentiation). *Let $f(\boldsymbol{x},y) \in \mathbb{F}[\boldsymbol{x}][y]$ be computed by a $\Sigma\wedge\Sigma$ circuit of size s and degree d. Then, $\partial_y(f)$ is a $\Sigma\wedge\Sigma$ circuit of size $O(sd^2)$, over $\mathbb{F}[\boldsymbol{x}][y]$.*

Proof sketch. Lemma A.12 shows that each f_e has $O(sd)$ size circuit where $f =: \sum_e f_e y^e$. Doing this for each $e \in [0,d]$ gives a blowup of $O(sd^2)$ and the representation: $\partial_y(f) = \sum_e f_e \cdot e \cdot y^{e-1}$. □

Remark. Same property holds for $\Sigma\wedge\Sigma\wedge$ circuits.

Lastly, we show that $\Sigma\wedge\Sigma$ circuit can be converted into ARO. In fact, we give the proof for a more general model $\Sigma\wedge\Sigma\wedge$. The key ingredient for the lemma is the *duality trick*.

Lemma A.14 (Duality trick [Sax08]). *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.

We remark that the above proof works for fields of characteristic $= 0$, or $> d$.

Now, the basic idea is to convert $\wedge\Sigma\wedge$ into $\Sigma\Pi\Sigma^{\{1\}}\wedge$ (i.e. sum-of-product-of-univariates) which is subsumed by ARO [Gur15, Section 2.5.2].

Lemma A.15 ($\Sigma\wedge\Sigma\wedge$ as ARO). *Let $f \in \mathbb{F}[\boldsymbol{x}]$ be an n-variate polynomial computable by $\Sigma\wedge\Sigma\wedge$ circuit of size s and syntactic degree D. Then f is computable by an* ARO *of size $O(sn^2D^2)$.*

Proof sketch. Let $g^e = (g_1(x_1) + \cdots + g_n(x_n))^e$, where $\deg(g_i) \cdot e \leq D$. Using Lemma A.14 we get $g^e = \sum_{i=1}^{O(ne)} h_{i1}(x_1) \cdots h_{in}(x_n)$, where each h_{ij} is of degree at most D.

We do this for each power (i.e. each summand of f) individually, to get the final sum-of-product-of-univariates; of top-fanin $O(sne)$ and individual degree at most D. This is an ARO of size $O(sne) \cdot n \cdot D \leq O(sn^2D^2)$. □

A.3 Basic mathematical tools

For the time-complexity bound, we need to optimize the following function:

Lemma A.16. *Let $k \in \mathbb{N}_{\geq 4}$, and $h(x) := x(k-x)7^x$. Then, $\max_{i \in [k-1]} h(i) = h(k-1)$.*

Proof sketch. Differentiate to get $h'(x) = (k-x)7^x - x7^x + x(k-x)(\log 7)7^x = 7^x \cdot [x^2(-\log 7) + x(k\log 7 - 2) + k]$. It vanishes at $x = \left(\frac{k}{2} - \frac{1}{\log 7}\right) + \sqrt{\left(\frac{k}{2} - \frac{1}{\log 7}\right)^2 - \frac{k}{\log 7}}$. Thus, h is maximized at the integer $x = k-1$. □

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 A.17 (Valuation). *Let $f \in \mathbb{F}(\boldsymbol{x}, y)$ such that $\mathrm{val}_y(f) \geq 0$. Then, $f \in \mathbb{F}(\boldsymbol{x})[[y]] \bigcap \mathbb{F}(\boldsymbol{x}, y)$.*

Proof sketch. Let $f = g/h$ such that $g, h \in \mathbb{F}[\boldsymbol{x}, y]$. Now, $\mathrm{val}_y(f) \geq 0$, implies $\mathrm{val}_y(g) \geq \mathrm{val}_y(h)$. Let $\mathrm{val}_y(g) = d_1$ and $\mathrm{val}_y(h) = d_2$, where $d_1 \geq d_2 \geq 0$. Further, write $g = y^{d_1} \cdot \tilde{g}$ and $h = y^{d_2} \cdot \tilde{h}$. Write, $\tilde{h} = h_0 + h_1\, y + h_2\, y^2 + \cdots + h_d\, y^d$, for some d; with $h_i \in \mathbb{F}[\boldsymbol{x}]$. Note that $h_0 \neq 0$. Thus

$$\begin{aligned} f &= y^{d_1 - d_2} \cdot \tilde{g}/(h_0 + h_1 y + \cdots + h_d y^d) \\ &= y^{d_1-d_2} \cdot (\tilde{g}/h_0) \cdot \left((h_1/h_0) + (h_2/h_0)y + \cdots + (h_d/h_0)y^d\right)^{-1} \in \mathbb{F}(\boldsymbol{x})[[y]] \end{aligned}$$

□

Claim A.18. *For our linear-map Ψ, and $g \in \Sigma\Pi^{[\delta]}$: $\Psi(g) \in \Sigma\Pi^{[\delta]}$ of size $3^\delta \cdot \mathrm{size}(g)$ (for $n \gg \delta$).*

Proof sketch. Each monomial $\boldsymbol{x}^{\boldsymbol{a}}$ of degree δ, can produce $\prod_i (a_i + 1) \leq ((\sum_i a_i + n)/n)^n \leq (\delta/n + 1)^n$-many monomials, by AM-GM inequality as $\sum_i a_i \leq \delta$. As $\delta/n \to 0$, we have $(1 + \delta/n)^n \to e^\delta$. As $e < 3$, the upper bound follows. □

A.4 De-bordering simple models

In this section we will discuss known de-bordering results of restricted models like product of sum of univariates and ARO.

Polynomials approximated by $\Pi\Sigma$ can be easily de-bordered [BIZ18, Prop.A.12]. In fact, it is the only constructive de-bordering result known so far. We extend it to show that same holds for polynomials approximated by $\Pi\Sigma\wedge$ circuits. In fact, we start it by showing a much more general theorem.

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 A.19. $\overline{(\mathcal{C}/\mathcal{C}) \cdot (\mathcal{D}/\mathcal{D})} \subseteq (\overline{\mathcal{C}}/\overline{\mathcal{C}}) \cdot (\overline{\mathcal{D}}/\overline{\mathcal{D}})$.

Proof. Suppose $(g_1/g_2) \cdot h_1/h_2 = f + \varepsilon \cdot Q$, where $Q \in \mathbb{F}(\boldsymbol{x}, \varepsilon)$ and $f \in \mathbb{F}(\boldsymbol{x})$. Let $\mathrm{val}_\varepsilon(g_i) =: a_i$ and $\mathrm{val}_\varepsilon(h_i) =: b_i$. Denote, $g_i =: \varepsilon^{a_i} \cdot \tilde{g}_i$, similarly $\tilde{h}_i$. Further, assume $\tilde{g}_i =: \hat{g}_i + \varepsilon \cdot \hat{g}_i'$; similarly for $\tilde{h}_i$, we define $\hat{h}_i \in \mathbb{F}[\boldsymbol{x}]$. Note that $\hat{g}_i \in \overline{\mathcal{C}}$, similarly $\hat{h}_i \in \overline{\mathcal{D}}$.

So, LHS $= \varepsilon^{a_1 - a_2 + b_1 - b_2} \cdot (\tilde{g}_1/\tilde{g}_2) \cdot (\tilde{h}_1/\tilde{h}_2)$. This has a limit $\lim_{\varepsilon \to 0}$, so $a_1 + b_1 - a_2 - b_2 \geq 0$. If it is ≥ 1, the limit in RHS is 0 and so $f = 0$. If $a_1 + b_1 - a_2 - b_2 = 0$, then

$$f \;=\; (\hat{g}_1/\hat{g}_2) \cdot (\hat{h}_1/\hat{h}_2) \;\in\; (\overline{\mathcal{C}}/\overline{\mathcal{C}}) \cdot (\overline{\mathcal{D}}/\overline{\mathcal{D}}) \;.$$

□

Now, we show an important de-bordering result on $\Pi\Sigma\wedge$ circuits.

Lemma A.20 (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 $f(\boldsymbol{x})$.*

Proof. We will show that $\overline{\Pi\Sigma\wedge} = \Pi\Sigma\wedge \subseteq \mathrm{ARO}$. From Lemma A.19 (and its proof), it follows that $\overline{\Pi\Sigma\wedge} \subseteq \prod(\overline{\Sigma\wedge})$. However, we note that $\overline{\Sigma\wedge} = \Sigma\wedge$ and it does not change the size (as it can not increase the sparsity). Therefore, the size does not increase and further it is an ARO. Thus, the conclusion follows. □

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 [For16]; also implicitly in [GKS16].

Lemma A.21 (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 $f(\boldsymbol{x})$.*

Proof. By definition, there exists a polynomial $g = f + \varepsilon Q$ computable by width w ARO over $\mathbb{F}(\varepsilon)[\boldsymbol{x}]$. Note that $w \leq s$. In this proof, we will use the partial derivative matrix. With respect to any-order-prefix $\boldsymbol{y} \subset \boldsymbol{x}$, consider the partial derivative matrix $N(g)$. Using Lemma A.5 and A.6, we know $\mathrm{rk}_{\mathbb{F}(\varepsilon)}(N(g)) \leq w$. This means determinant of *any* $(w+1) \times (w+1)$ minor of $N(g)$ is identically zero. One can see that the entries of the minor are coefficients of monomials of g which are in $\mathbb{F}[\varepsilon][\boldsymbol{x} \setminus \boldsymbol{y}]$. Thus, determinant polynomial will remain zero even under the limit of $\varepsilon = 0$. Since, $\lim_{\varepsilon \to 0} g = f$, each minor (under limit) captures partial derivative matrix of f of corresponding rows and columns. Thus, we get $\mathrm{rk}_{\mathbb{F}}(N(f)) \leq w$. Lemma A.6 shows that there exists an ARO, of width w over $\mathbb{F}$, which *exactly* computes f. □

An obvious consequence of Lemma A.15 and Lemma A.21 is the following de-bordering result.

Lemma A.22 (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 $f(\boldsymbol{x})$.*

B Gentle leap into depth-4: De-bordering $\overline{\Sigma^{[k]}\Pi\Sigma\wedge}$ circuits

The main content of this section is to sketch the de-bordering theorem for $\overline{\Sigma^{[k]}\Pi\Sigma\wedge}$. We intend to extend DiDIL and induct on the bloated model, as sketched in Subsection 1.4.

Theorem B.1 ($\overline{\Sigma^{[k]}\Pi\Sigma\wedge}$ upper bound)**.** *Let $f(\boldsymbol{x}) \in \mathbb{F}[x_1, \ldots, x_n]$, such that f can be computed by a $\overline{\Sigma^{[k]}\Pi\Sigma\wedge}$-circuit of size s. Then f is also computable by an ABP (over $\mathbb{F}$), of size $s^{O(k \cdot 7^k)}$.*

Proof sketch. We will go through the proof of Theorem 3.2 (see Section 3), while reusing the notations, and point out the important maneuvering for DiDIL to work on this more general bloated-model $(\Pi\Sigma\wedge/\Pi\Sigma\wedge) \cdot (\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge)$.

Base case. The analysis remains unchanged. We merely have to de-border $\Pi\Sigma\wedge$ and $\Sigma\wedge\Sigma\wedge$ for numerator and denominator separately using Lemma A.20 and Lemma A.22. Then use the product lemma (Lemma A.19) to conclude:

$$\overline{(\Pi\Sigma\wedge/\Pi\Sigma\wedge) \cdot (\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge)} \subseteq (\Pi\Sigma\wedge/\Pi\Sigma\wedge) \cdot (\mathsf{ARO}/\mathsf{ARO}) \subseteq \mathsf{ABP}/\mathsf{ABP}\,.$$

Reducing the problem to $k-1$. To facilitate DiDIL, we need a slightly more general $\Phi : \mathbb{F}(\varepsilon)[\boldsymbol{x}] \longrightarrow \mathbb{F}(\varepsilon)[\boldsymbol{x}, z_1, z_2]$ such that the bottom $\Sigma\wedge$ circuits are 'invertible' (mod z_1^d). So, we directly use the sparse-PIT (univariate in z_2) map Ψ [KS01], as $\Sigma\wedge$ are at most s-sparse. By similar argument, it suffices to upper bound $\Phi(f)$.

We will apply again divide and derive to reduce the fanin step by step. We just need to understand $T_{i,j}$. Similar to Claim 3.8, we claim the following.

Claim B.2. *$T_{1,k-1} \in (\Pi\Sigma\wedge/\Pi\Sigma\wedge)\ (\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge)$ over $\mathcal{R}_{k-1}(\boldsymbol{x}, \varepsilon)$ of size at most $s^{O(k7^k)}$.*

Proof. The main part is to show that dlog acts on $\Pi\Sigma\wedge$ circuits "well". To elaborate, we note that Equation 3.9 can be written for $\Sigma\wedge$ circuits, giving a $\Sigma\wedge\Sigma\wedge$ circuit. To elaborate, let $A - z_1 \cdot B =: h \in \Sigma\wedge$, such that $0 \neq A \in \mathbb{F}(z_2, \varepsilon)$. Therefore, over $\mathcal{R}_1(\boldsymbol{x})$, we have

$$\mathrm{dlog}(h) = -\frac{\partial_{z_1}(z_1 \cdot B)}{A\,(1 - z_1 \cdot B/A)} = -\frac{\partial_{z_1}(z_1 \cdot B)}{A} \cdot \sum_{j=0}^{d_1-1} \left(\frac{z_1 \cdot B}{A}\right)^j .$$

Once we use the fact that $\Sigma\wedge\Sigma\wedge$ is closed under multiplication (Lemma A.10), it readily follows that $\mathrm{dlog}(\Pi\Sigma\wedge) \in \Sigma\wedge\Sigma\wedge$. Moreover, the derivative of $\Sigma\wedge\Sigma\wedge$ is again a $\Sigma\wedge\Sigma\wedge$ circuit, due to easy interpolation (Lemma A.13). Following the same proof arguments (as for Theorem 3.2), we can establish the above claim.

It was already remarked that properties shown in Subsection A.2 hold for $\Sigma\wedge\Sigma\wedge$ circuits as well. Therefore, the rest of the calculations remain unchanged, and the size claim holds. □

Interpolation & Definite integration. It is again not hard to see that

$$f_j/t_{k-j,j}|_{z_1=0} \in \lim_{\varepsilon\to 0} \sum_{i\in[k-j]} \mathbb{F}(z_2,\varepsilon)\cdot(\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge) \subseteq \mathsf{ARO}/\mathsf{ARO} \subseteq \mathsf{ABP}/\mathsf{ABP}\ .$$

Here, we have used the facts that $\Sigma\wedge\Sigma\wedge$ is closed under multiplication (Lemma A.10) and $\overline{\Sigma\wedge\Sigma\wedge} \subseteq \mathsf{ARO}$ (Lemma A.22). The remaining steps also follow similarly once we have the $\mathsf{ABP}/\mathsf{ABP}$ form of de-bordered expressions.

We remark that in all the steps the size and degree claims remain the same and hence the final size of the circuit for $\Phi(f)$ immediately follows. □

C Blackbox PIT for border depth-4 circuits

C.1 Known useful PITs

We dedicate this section to discuss some known blackbox PIT results for *exact* computation. We will start with the simplest one obtained using PIT lemma of [Sch80, Zip79, DL78, Ore22].

Lemma C.1 (Trivial hitting set). *For a class of n-variate, individual degree $< d$ polynomial $f \in \mathbb{F}[x_1,\ldots,x_n]$ there exists an explicit hitting-set $\mathcal{H} \subseteq \mathbb{F}^n$ of size d^n+1. In other words, there exists a point $\overline{\alpha} \in \mathcal{H}$ such that $f(\overline{\alpha}) \neq 0$ (if $f \neq 0$).*

The above result becomes interesting when $n = O(1)$ as it yields a polynomial-time explicit hitting set. For general n, we have better results for restricted circuits, for eg. sparse circuits $\Sigma\Pi$, [AB03, KS01] gave a map which reduces multivariate sparse polynomial into univariate polynomial of small degree, while preserving the non-identity. Since testing (low-degree) univariate polynomial is trivial, we get a simple PIT algorithm for sparse polynomials.

Indeed if identity of sparse polynomial can be tested efficiently, product of sparse polynomials $\Pi\Sigma\Pi$ can be tested efficiently. We formalise this in the following lemma.

Lemma C.2 ([Sap13, Lemma 2.3]). *For the class of n-variate, degree d polynomial $f \in \mathbb{F}[x_1,\ldots,x_n]$ computable by $\Pi\Sigma\Pi$ of size s, there exist an explicit hitting set of size* $\mathrm{poly}(s,d)$.

Finally, we state the best known PIT result for ARO, see [GKS17, GG20] for more details.

Theorem C.3 (ARO hitting set). *For the class of d-degree n-variate polynomials $f \in \mathbb{F}[\boldsymbol{x}]$ computable by size s ARO, there exists an explicit hitting set of size $s^{O(\log\log s)}$.*

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 C.4. *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 $s_1 \cdot s_2 \cdot O(d)$.*

Proof. Let $f = f_1 \cdot f_2 \in \mathcal{C}_1 \cdot \mathcal{C}_2$ such that $f_1 \in \mathcal{C}_1$ and $f_2 \in \mathcal{C}_2$. For each $\boldsymbol{a}_i \in \mathcal{H}_1$, $\boldsymbol{b}_j \in \mathcal{H}_2$ define a 'formal-sum' evaluation point (over $\mathbb{F}[t]$) $\boldsymbol{c} := (c_\ell)_{1 \le \ell \le n}$ such that $c_\ell := a_{i\ell} + t \cdot b_{j\ell}$; where t is a formal variable. Collect these points, going over i, j, in a set H. It can be seen, by shifting and scaling, that non-zeroness is preserved: there exists $\boldsymbol{c} \in H$ such that $0 \neq f(\boldsymbol{c}) \in \mathbb{F}[t]$ and $\deg f(\boldsymbol{c}) = O(d)$. Using trivial hitting set from Lemma C.1 we obtain the final hitting set $\mathcal{H}$ of size $O(s_1 \cdot s_2 \cdot d)$. □

Remark. The above argument easily extends to circuit classes $(\mathcal{C}_1/\mathcal{C}_1) \cdot (\mathcal{C}_2/\mathcal{C}_2)$, which compute rationals of the form $(g_1/g_2) \cdot (h_1/h_2)$, where $g_i \in \mathcal{C}_1$ and $h_i \in \mathcal{C}_2$ $(g_2 h_2 \neq 0)$.

C.2 Efficient hitting set for $\overline{\Sigma\wedge\Sigma\Pi^{[\delta]}}$

Forbes [For15] gave quasipolynomial-time blackbox PIT for $\Sigma\wedge\Sigma\Pi^{[\delta]}$; this was basically a *rank*-based method. We will make some small observations to extend the same for $\overline{\Sigma\wedge\Sigma\Pi^{[\delta]}}$ as well. We encourage interested readers to refer [For15] for details. First, we need some definitions and properties.

Shifted Partial Derivative measure $\boldsymbol{x}^{\le \ell}\boldsymbol{\partial}_{\le m}$ is a linear operator first introduced in [Kay12, GKKS14] as:

$$\boldsymbol{x}^{\le \ell}\boldsymbol{\partial}_{\le m}(g) \;:=\; \{\boldsymbol{x}^{\boldsymbol{c}} \partial_{\boldsymbol{x}^{\boldsymbol{b}}}(g)\}_{\deg \boldsymbol{x}^{\boldsymbol{c}} \le \ell, \deg \boldsymbol{x}^{\boldsymbol{b}} \le m} \, .$$

It was shown in [For15] that the rank of shifted partial derivatives of a polynomial computed by $\Sigma\wedge\Sigma\Pi^{[\delta]}$ is small. We state the result formally in the next lemma. Consider the fractional field $\mathcal{R} := \mathbb{F}(\varepsilon)$.

Lemma C.5 (Measure upper bound). *Let $g(\varepsilon, \boldsymbol{x}) \in \mathcal{R}[x_1, \ldots, x_n]$ be computable by $\Sigma\wedge\Sigma\Pi^{[\delta]}$ circuit of size s. Then*

$$\operatorname{rk} \boldsymbol{x}^{\le \ell}\boldsymbol{\partial}_{\le m}(g) \le s \cdot m \cdot \binom{n + (\delta - 1)m + \ell}{(\delta - 1)m + \ell}.$$

Further they observed that, rank can be lower bounded using *Trailing Monomial*. Under any *monomial ordering*, the trailing monomial of g denoted by $\mathrm{TM}(g)$ is the smallest monomial in the set $\mathrm{Supp}(g) := \{\boldsymbol{x}^{\boldsymbol{a}} : \mathrm{coef}_{\boldsymbol{x}^{\boldsymbol{a}}}(g) \neq 0\}$.

Proposition C.6 (Measure the trailing monomial). *Consider $g \in \mathcal{R}[\boldsymbol{x}]$. For any $\ell, m \ge 0$,*

$$\operatorname{rk} \operatorname{span} \boldsymbol{x}^{\le \ell}\boldsymbol{\partial}_{\le m}(g) \ge \operatorname{rk} \operatorname{span} \boldsymbol{x}^{\le \ell}\boldsymbol{\partial}_{\le m}\left(\mathrm{TM}(g)\right).$$

For a large enough characteristic, lower bound on a monomial was obtained.

Lemma C.7 (Monomial lowerbound). *Consider a monomial $\boldsymbol{x}^{\boldsymbol{a}} \in \mathcal{R}[x_1, \ldots, x_n]$. Then,*

$$\operatorname{rk} \operatorname{span}\left(\boldsymbol{x}^{\le \ell}\boldsymbol{\partial}_{\le m}\left(\boldsymbol{x}^{\boldsymbol{a}}\right)\right) \ge \binom{\eta}{m}\binom{\eta - m + \ell}{\ell}$$

where $\eta := |\mathrm{Supp}\left(\boldsymbol{x}^{\boldsymbol{a}}\right)|$.

In [For15] the above results were combined to show that the trailing monomial of polynomials computed by $\Sigma\wedge\Sigma\Pi^{[\delta]}$ circuits have log-small support size. Using the same idea we show that if such a polynomial approximates f, then support of $\mathrm{TM}(f)$ is also small. We formalize this in the next lemma.

Lemma C.8 (Trailing monomial support)**.** *Let $g(\varepsilon, \boldsymbol{x}) \in \mathcal{R}[x_1, \ldots, x_n]$ be computable by a $\Sigma\wedge\Sigma\Pi^{[\delta]}$ circuit of size s such that $g = f + \varepsilon \cdot Q$ where $f \in \mathbb{F}[\boldsymbol{x}]$ and $Q \in \mathbb{F}[\varepsilon, \boldsymbol{x}]$. Let $\eta := |\operatorname{Supp}(\mathrm{TM}(f))|$. Then $\eta = O(\delta \log s)$.*

Proof. Let $\boldsymbol{x}^{\boldsymbol{a}} := \mathrm{TM}(f)$ and $S := \{i \,|\, a_i \neq 0\}$. Define a substitution map ρ such that $x_i \to y_i$ for $i \in S$ and $x_i \to 0$ for $i \notin S$. It is easy to observe that $\mathrm{TM}(\rho(f)) = \rho(\mathrm{TM}(f)) = \boldsymbol{y}^{\boldsymbol{a}}$. Using Lemma C.5 we know:

$$\operatorname{rk}_{\mathcal{R}} \boldsymbol{y}^{\leq \ell} \boldsymbol{\partial}_{\leq m}(\rho(g)) \;\leq\; s \cdot m \cdot \binom{\eta + (\delta-1)m + \ell}{(\delta-1)m + \ell} =: R\,.$$

To obtain the upper bound for $\rho(f)$ we use the following claim.

Claim C.9. $\operatorname{rk}_{\mathbb{F}} \boldsymbol{y}^{\leq \ell} \boldsymbol{\partial}_{\leq m}(\rho(f)) \leq R$.

Proof. Define *coefficient matrix* $N(\rho(g))$ with respect to $\boldsymbol{y}^{\leq \ell} \boldsymbol{\partial}_{\leq m}(\rho(g))$ as follows: the rows are indexed by the operators $\boldsymbol{y}^{=\ell_i} \partial_{\boldsymbol{y}^{=m_i}}$, while the columns are indexed by the terms present in $\rho(g)$; and the entries are the respective operator-action on the respective term in $\rho(g)$. Note that $\operatorname{rk}_{\mathbb{F}(\varepsilon)} N(\rho(g)) \leq R$. Similarly define $N(\rho(f))$ with respect to $\boldsymbol{y}^{\leq \ell} \boldsymbol{\partial}_{\leq m}(\rho(f))$, then it suffices to show that $\operatorname{rk}_{\mathbb{F}} N(\rho(f)) \leq R$.

For any $r > R$, let $\mathcal{N}(\rho(g))$ be a $r \times r$ sub-matrix of $N(\rho(g))$. The rank bound ensures: $\det \mathcal{N}(\rho(g)) = 0$. This will remain true under the limit $\varepsilon = 0$; thus, $\det(\mathcal{N}(\rho(f))) = 0$. Since $r > R$ was arbitrary and linear dependence is preserved, we deduce: $\operatorname{rk}_{\mathbb{F}} N(\rho(f)) \leq R$. □

For lower bound, recall $\boldsymbol{y}^{\boldsymbol{a}} = \mathrm{TM}(\rho(f))$. Then using Proposition C.6 and Lemma C.7 we get:

$$\operatorname{rk}_{\mathbb{F}} \boldsymbol{y}^{\leq \ell} \boldsymbol{\partial}_{\leq m}(\rho(f)) \geq \binom{\eta}{m}\binom{\eta - m + \ell}{\ell}. \tag{C.10}$$

Comparing Claim C.9 and Equation C.10 we get:

$$s \geq \frac{1}{m} \cdot \binom{\eta}{m} \cdot \binom{\eta - m + \ell}{\ell} \Big/ \binom{\eta + (\delta-1)m + \ell}{(\delta-1)m + \ell}.$$

For $\ell := (\delta-1)(\eta + (\delta-1)m)$ and $m := \lfloor n/e^3\delta \rfloor$, [For15, Lem.A.6] showed $\eta \leq O(\delta \log s)$. □

Existence of a small support monomial in a polynomial, which is being approximated, is a structural result which will help in constructing a hitting set for this larger class. The idea is to use a map that reduces the number of variables to support-size, and then invoke Lemma C.1.

Theorem C.11 (Hitting set for $\overline{\Sigma\wedge\Sigma\Pi^{[\delta]}}$)**.** *For the class of n-variate, degree d polynomials* approximated *by $\Sigma\wedge\Sigma\Pi^{[\delta]}$ circuits of size s, there is an explicit set $H \subseteq \mathbb{F}^n$ of size $s^{O(\delta \log s)}$ i.e., for every such nonzero polynomial f there exists an $\boldsymbol{\alpha} \in H$ for which $f(\boldsymbol{\alpha}) \neq 0$.*

Proof. Let $g(\varepsilon, \boldsymbol{x}) \in \mathcal{R}[x_1, \ldots, x_n]$ be computable by a $\Sigma\wedge\Sigma\Pi^{[\delta]}$ circuit of size s such that $g =: f + \varepsilon \cdot Q$, where $f \in \mathbb{F}[\boldsymbol{x}]$ and $Q \in \mathbb{F}[\varepsilon, \boldsymbol{x}]$. Then Lemma C.8 shows that there exists a monomial $\boldsymbol{x}^{\boldsymbol{a}}$ of f such that $\eta := |\operatorname{Supp}(\boldsymbol{x}^{\boldsymbol{a}})| = O(\delta \log s)$.

Let $S \in \binom{[n]}{\eta}$. Define a substitution map ρ_S such that $x_i \to y_i$ for $i \in S$ and $x_i \to 0$ for $i \notin S$. Note that, under this substitution non-zeroness of f is preserved for some S; because monomials of support $S \supseteq \operatorname{Supp}(\boldsymbol{x}^{\boldsymbol{a}})$ will survive for instance. Essentially $\rho_S(f)$ is an η-variate degree-d polynomial. For which Lemma C.1 gives a trivial hitting set of size $O(d^\eta)$. Therefore, with respect to S we get a hitting set $\mathcal{H}_S$ of size $O(d^\eta)$. To finish, we do this for all such S, to obtain the final hitting set $\mathcal{H}$ of size:

$$\binom{n}{\eta} \cdot O\left(d^\eta\right) \le O((nd)^\eta).$$

□

Remark. Unlike border-depth-3 PIT result, we obtain this result without de-bordering the circuit at all.

C.3 DiDIL on depth-4 models

We sketched in Subsection 1.4 that DiDIL-paradigm along with the branching idea, can be used to give hitting set for border depth-4 $\overline{\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}}$ and $\overline{\Sigma^{[k]}\Pi\Sigma\wedge}$ circuits. For brevity, we denote these two types of (non-border) depth-4 circuits by $\Sigma^{[k]}\Pi\Sigma\Upsilon$ circuits where $\Upsilon \in \{\wedge, \Pi^{[\delta]}\}$. We will give separate hitting set for the border of each class, while analysing them together.

Theorem C.12 (Hitting set for bounded border depth-4)**.** *There exists an explicit $s^{O(k \cdot 7^k \cdot \log\log s)}$ (respec. $s^{O(\delta^2 k 7^k \log s)}$)-time hitting set for $\overline{\Sigma^{[k]}\Pi\Sigma\wedge}$ (respec. $\overline{\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}}$)-circuits of size s.*

Proof sketch. We will again follow the same notation as Subsection 4.1. Let $g_0 := \sum_{i \in [k]} T_{i,0} = f_0 + \varepsilon S_0$ such that g_0 is computable by $\Sigma^{[k]}\Pi\Sigma\Upsilon$ over $\mathbb{F}(\varepsilon)$. As earlier, we will instead work with bloated model that preserves the structure on applying the DiDIL technique. The bloated model we consider is $\Sigma^{[k]}\left(\Pi\Sigma\Upsilon/\Pi\Sigma\Upsilon\right)\left(\Sigma\wedge\Sigma\Upsilon/\Sigma\wedge\Sigma\Upsilon\right)$. Define a map $\Phi : \mathbb{F}(\varepsilon)[\boldsymbol{x}] \to \mathbb{F}(\varepsilon)[\boldsymbol{x}, z_1, z_2]$ such that $x_i \to z_1 \cdot x_i + \Psi(x_i)$. Essentially, our $\Sigma\Upsilon$ circuits are at most s-sparse, so it suffices to consider the sparse-PIT (univariate) map Ψ [KS01], yielding a different Φ. The invertible map implies: $f_0 \neq 0$ if and only if $\Phi(f_0) \neq 0$; and $\deg_{z_2} \Phi(f_0) \le \operatorname{poly}(s)$.

The next steps are essentially the same: reduce k to the bloated $k-1$, and inductively to the bloated $k = 1$ case. There will be 'branches' and for each branch we will give efficient hitting sets; taking their union will give the final hitting set.

By **Di**vide and **D**erive, we will eventually show that

$$f_0 \neq 0 \iff f_{k-1} \neq 0 \,\text{over}\, \mathcal{R}_j(\boldsymbol{x}), \text{ or } \bigvee_{i=1}^{k-2} \left(f_i / t_{k-i,i}|_{z_1=0} \neq 0, \text{ over}\, \mathbb{F}(z_2)(\boldsymbol{x})\right).$$

Now, $T_{1,k-1} \in \left(\Pi\Sigma\Upsilon/\Pi\Sigma\Upsilon\right)\left(\Sigma\wedge\Sigma\Upsilon/\Sigma\wedge\Sigma\Upsilon\right)$, over $\mathcal{R}_{k-1}(\boldsymbol{x}, \varepsilon)$ (similar to Claim B.2). The trick is again to use dlog and show that $\operatorname{dlog}(\Pi\Sigma\Upsilon) \in \Sigma\wedge\Sigma\Upsilon$. However the size blowup behaves slightly differently. We point this out in the next claim.

Claim C.13. *For $\Sigma^{[k]}\Pi\Sigma\wedge$, respectively $\Sigma^{[k]}\Pi\Sigma\Pi^{[\delta]}$, we have $T_{1,k-1} \in (\Pi\Sigma\wedge/\Pi\Sigma\wedge)\,(\Sigma\wedge\Sigma\wedge/\Sigma\wedge\Sigma\wedge)$, respec. $\left(\Pi\Sigma\Pi^{[\delta]}/\Pi\Sigma\Pi^{[\delta]}\right)\left(\Sigma\wedge\Sigma\Pi^{[\delta]}/\Sigma\wedge\Sigma\Pi^{[\delta]}\right)$ over $\mathcal{R}_{k-1}(\boldsymbol{x},\varepsilon)$, of size $s^{O(k7^k)}$ respec. $(s3^\delta)^{O(k7^k)}$.*

Proof sketch. We explain it for one step i.e. over $\mathcal{R}_1(\boldsymbol{x},\varepsilon)$. Let $A - z_1\cdot B = h \in \Sigma\Upsilon$, such that $A \in \mathbb{F}(z_2,\varepsilon)$ (we have already shifted). Therefore, over $\mathcal{R}_1(\boldsymbol{x})$, we have

$$\operatorname{dlog}(h) = -\frac{\partial_{z_1}(z_1\cdot B)}{A\,(1 - z_1\cdot B/A)} = -\frac{\partial_{z_1}(z_1\cdot B)}{A}\cdot\sum_{j=0}^{d_1-1}\left(\frac{z_1\cdot B}{A}\right)^j .$$

Here, use the fact that $\Sigma\wedge\Sigma\Upsilon$ is closed under multiplication. For $\Sigma\wedge\Sigma\wedge$ circuits, the calculations remains the same as in Appendix B. However, for $\Sigma\wedge\Sigma\Pi^{[\delta]}$ circuits, note that as h is shifted, $\text{size}(B)$ is no longer $\text{poly}(s)$; but it is at most $3^\delta\cdot s$, see Claim A.18. Therefore, the claim follows. □

Eventually, one can show (using Lemma A.19 to distribute):

$$f_{k-1} \in \overline{(\Pi\Sigma\Upsilon/\Pi\Sigma\Upsilon)\cdot(\Sigma\wedge\Sigma\Upsilon/\Sigma\wedge\Sigma\Upsilon)} \subseteq (\Pi\Sigma\Upsilon/\Pi\Sigma\Upsilon)\cdot\overline{(\Sigma\wedge\Sigma\Upsilon/\Sigma\wedge\Sigma\Upsilon)} .$$

When $\Upsilon = \wedge$, we know $\overline{\Sigma\wedge\Sigma\wedge} \subseteq$ ARO and thus this has a hitting set of size $s^{O(k7^k\log\log s)}$ (Theorem C.3). We also know hitting set for $\Pi\Sigma\wedge$ (Lemma C.2). Combining them using Lemma C.4, we have a quasipolynomial-time hitting set of size $s^{O(k7^k\log\log s)}$.

As seen before, we also need to understand $z_1 = 0$ evaluation. By similar argument, it will follow that

$$f_j/t_{k-j,j}|_{z_1=0} \in \lim_{\varepsilon\to 0}\sum_{i\in[k-j]}\mathbb{F}(z_2,\varepsilon)\cdot(\Sigma\wedge\Sigma\Upsilon/\Sigma\wedge\Sigma\Upsilon) \subseteq \overline{\Sigma\wedge\Sigma\Upsilon} .$$

When $\Upsilon = \wedge$, we can de-border and this can be shown to be an ARO. Thus, in that case $f_j/t_{k-j,j}|_{z_1=0} \in$ ARO/ARO, where hitting set is known (similarly as before) giving hitting set for each branch. Once we have hitting set for each branch, we can take union (similar to Claim 4.4) to finally give the desired hitting set.

Unfortunately, we do not know $\overline{\Sigma\wedge\Sigma\Upsilon}$, when $\Upsilon = \Pi^{[\delta]}$, as the duality trick cannot be directly applied. However, as we know hitting set for $\overline{\Sigma\wedge\Sigma\Pi^{[\delta]}}$, from Theorem C.11; we will use it to get the final hitting set. To see why this works, note that we need to 'hit' $f_{k-1} \in \left(\Pi\Sigma\Pi^{[\delta]}/\Pi\Sigma\Pi^{[\delta]}\right)\cdot\overline{\Sigma\wedge\Sigma\Pi^{[\delta]}}/\overline{\Sigma\wedge\Sigma\Pi^{[\delta]}}$. We know hitting sets for both $\Pi\Sigma\Pi^{[\delta]}$ (Lemma C.2) and $\overline{\Sigma\wedge\Sigma\Pi^{[\delta]}}$ (Theorem C.11), thus combining them is easy Lemma C.4.

To get the final estimate, define $s' := s^{O(\delta k7^k)}$; which signifies the size blowup due to DiDIL. Next, the hitting set $\mathcal{H}_{k-1}$ for f_{k-1} has size $(nd)^{O(\delta\log s')} \leq s^{O(\delta^2 k7^k\log s)}$. We know that similar bound also holds for each branch. Taking their union gives the final hitting set of the size as claimed. □

C.4 Cone-size and coefficient extraction: Tools for log-variate depth-3 circuits

Here is an important lemma, originally from [For14, Corollary 4.14], which shows that small partial derivative space implies existence of small cone-size monomial. For a detailed proof, we refer [Gho19, Lemma 2.3.15]

Theorem C.14 (Cone-size concentration). *Let $\mathbb{F}$ be a field of characteristic 0 or greater than d. Let $\mathcal{P}$ be a set of n-variate d-degree polynomials over $\mathbb{F}$ such that for all $P \in \mathcal{P}$, the dimension of the partial derivative space of P is at most k. Then every nonzero $P \in \mathcal{P}$ has a cone-size-k monomial with nonzero coefficient.*

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

Lemma C.15 (Counting low-cones, [FGS18, 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 [FGS18, Lemma 4]. It is proved by multivariate interpolation.

Lemma C.16 (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 $\mathrm{poly}(\mathrm{size}(C), cs(m), d)$, where $cs(m)$ denotes the cone-size of m.*