Deterministic Divisibility Testing via Shifted Partial Derivatives

Michael A. Forbes [*]

April 16, 2015

Abstract

Kayal [Kay12] has recently introduced the method of *shifted partial derivatives* as a way to give the first exponential lower bound for computing an explicit polynomial as a *sum of powers of quadratic polynomials*. This method has garnered further attention because of the work of Gupta, Kamath, Kayal and Saptharishi [GKKS14] who used this method to obtain lower bounds that approach the "chasm at depth-4" ([AV08, Koi12, Tav13]).

In this work, we investigate to what extent this method can be used to obtain deterministic *polynomial identity testing (PIT)* algorithms, which are algorithms that decide whether a given algebraic circuit computes the zero polynomial. In particular, we give a $\mathsf{poly}(s)^{\mathcal{O}(\lg s)}$-time deterministic black-box PIT algorithm for a size-s sum of powers of quadratic polynomials. This is the first sub-exponential deterministic PIT algorithm for this model of computation and the first[1] PIT algorithm based on the method of shifted partial derivatives.

We also study the problem of *divisibility testing*, which when given polynomials $f(\overline{x})$ and $g(\overline{x})$ (as algebraic circuits) asks to decide whether $f(\overline{x})$ divides $g(\overline{x})$. Using Strassen's [Str73] technique for the elimination of divisions, we show that one can obtain deterministic divisibility testing algorithms via deterministic PIT algorithms, and this reduction does not dramatically increase the complexity of the underlying algebraic circuits.

Using this reduction, we show that deciding divisibility of a quadratic polynomial f into a sparse polynomial g reduces to PIT of *sums of a monomial multiplied by a power of quadratic polynomials*. We then extend the method of shifted partial derivatives to give a $\mathsf{poly}(s)^{\mathcal{O}(\lg s)}$-time deterministic black-box PIT algorithm for this model of computation, and thus derive a corresponding deterministic divisibility algorithm. This is the first non-trivial deterministic algorithm for this problem.

Previous work on multivariate divisibility testing due to Saha, Saptharishi and Saxena [SSS13] gave algorithms for when f is linear and g is sparse, and essentially worked via PIT algorithms for *read-once (oblivious) algebraic branching programs (roABPs)*. We give explicit sums of powers of quadratic polynomials that require exponentially-large roABPs in a strong sense, showing that techniques known for roABPs have limited applicability in our regime.

Finally, by combining our results with the algorithm of Forbes, Shpilka and Saptharishi [FSS14] we obtain $\mathsf{poly}(s)^{\mathcal{O}(\lg \lg s)}$-time deterministic black-box PIT algorithms for various models (translations of sparse polynomials, and sums of monomials multiplied by a power of a linear polynomial) where only $\mathsf{poly}(s)^{\Theta(\lg s)}$-time such algorithms were previously known.

[*] Email: `miforbes@csail.mit.edu`. School of Mathematics, Institute for Advanced Study. This work was performed when the author was a graduate student at MIT CSAIL (supported by Scott Aaronson's Waterman award, NSF CCF-1249349), when the author was a Google Research Fellow at the Simons Institute for the Theory of Computing, and when the author was a member of the School of Mathematics at the Institute for Advanced Study. This material is based upon work supported by the National Science Foundation under agreement No. CCF-1412958. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

[1] Kumar and Saraf (personal communication) have independently obtained PIT algorithms via applying the hardness versus randomness paradigm to lower bounds obtained via shifted partial derivatives. Their results seem related but incomparable to the results obtained here.

Contents

1 Introduction

We consider two problems from computational algebra: *polynomial identity testing (PIT)* and *divisibility testing*. In these problems, one receives algebraic circuits as inputs and the problem is to decide whether the multivariate polynomials that these circuits compute satisfy various properties. That is, the input to these problems is a directed acyclic graph whose internal nodes are labeled with algebraic operations (addition and multiplication) and whose leaves are labeled by variables x_i or scalars from a field $\mathbb{F}$. Algebraic circuits naturally compute polynomials in the ring $\mathbb{F}[x_1, \ldots, x_n]$ and are one of the most natural and succinct methods to describe such polynomials. However, the succinctness of this description creates challenges in algorithmically understanding the computed polynomial. In particular, while there are often randomized algorithms for deciding properties of the polynomials computed by algebraic circuits, derandomizing such algorithms is an active area of research, in particular because of its connections to derandomizing other well-known problems such as computing perfects matchings in NC ([Lov79, KUW86, MVV87]).

Polynomial Identity Testing: The first such problem we consider, polynomial identity testing, asks whether a given algebraic circuit computes the zero polynomial. This problem has a simple randomized algorithm: evaluate the circuit at random point and declare the polynomial non-zero if the evaluation is non-zero. The correctness of this algorithm follows from the Schwartz-Zippel [Sch80, Zip79, DL78] lemma. This randomized algorithm also has the property of being *black-box* in that it only uses the algebraic circuit as a method to implement an evaluation oracle $\overline{\alpha} \mapsto f(\overline{\alpha})$ for the underlying polynomial $f(\overline{x})$ computed by this circuit. In contrast, a *white-box* algorithm is allowed to use the structure of the computation of this circuit. Thus, as the black-box model allows weaker access, deriving results in this model are correspondingly stronger.

Creating deterministic PIT algorithms is a significant challenge, as it is known to have implications for the existence of explicit polynomials that require large algebraic circuits for their computation ([HS80, KI04, Agr05]), which is a long standing open question. As such, attention has focused on designing deterministic PIT algorithms for restricted models of algebraic computation. This focus on restricted models has in particular yielded a long line of work for PIT of bounded top-fan-in depth-3 and depth-4 circuits [DS07, KS07, KS11, KS09, SS11, KMSV13, SV11, ASSS12, SSS13, SS12, SS13, Gup14]. Recently, this focus has been further justified by *depth-reduction* results ([VSBR83, AV08, Koi12, Tav13, GKKS13]) that in particular show that polynomial-time deterministic black-box PIT algorithms for depth-3 algebraic circuits (of exponential degree) imply a deterministic *quasi*polynomial-time black-box PIT algorithm for general algebraic circuits. These results show that depth-3 circuits essentially capture the full complexity of general algebraic circuits. For more on algebraic circuits, PIT, and depth reduction, see the recent surveys of Shpilka-Yehudayoff [SY10], Saxena [Sax09, Sax14] or Saptharishi [Sap14].

Another direction of study has considered *algebraic branching programs (ABPs)*. As this model can simulate formulas (and thus low depth circuits), this direction of study ([RS05, AMS10, FS12, FS13b, ASS13, FS13a, FSS14, AGKS14, GKST15]) has focused on restricted classes of ABPs such as those that are *non-commutative*, *set-multilinear*, or *read-once (and oblivious)*. These three restrictions are essentially the same, and thus we will focus on read-once (oblivious) ABPs (roABPs). This model is partly interesting because it subsumes various other natural models (such as sums of powers of linear forms ([Sax08, FS13b])), and because its complexity is exactly characterized by the rank of Nisan's [Nis91] partial derivative matrix (which as a technique is also used for multilinear formula lower bounds ([Raz06, Raz09, RY09])). Further, developing deterministic polytime black-box PIT has applications to other questions such as derandomizing Noether Normalization ([Mul12, FS13a]) and can be seen as an algebraic analogue to derandomizing RL (see the discussion in Forbes-

Shpilka [FS13b]). For more on read-once ABPs, see the thesis of Forbes [For14].

While the inquiry into roABPs remains unfinished (in that we lack deterministic polytime black-box PIT for roABPs), we qualitatively understand this model well. That is, the techniques of Nisan [Nis91] produces explicit polynomials requiring exponentially large roABPs, and we have a deterministic white-box PIT algorithm for roABPs due to Raz and Shpilka [RS05] [2], as well as another white-box PIT algorithm (in characteristic zero) due to Arvind, Joglekar and Srinivasan [AJS09]. To some extent, many of the above papers sought to replicate the Raz-Shpilka [RS05] algorithm in the black-box model with varying degrees of success [3].

Given the above state of affairs, it is then a natural question as to which other restricted models of algebraic computation could designing PIT algorithms be potentially tractable. Historically, our understanding of a restricted model of algebraic computation is first demonstrated by finding a polynomial which is hard to to compute in this model, then with more work an efficient deterministic white-box PIT algorithm is developed, and finally additional work develops a corresponding black-box algorithm. While the hardness versus randomness paradigm (instantiated in the algebraic setting by Kabanets-Impagliazzo [KI04] and Dvir-Shpilka-Yehudayoff [DSY09]) shows that for general computation one can go directly from lower bounds to black-box PIT, it is not readily applicable for many restricted models of computation. Further, results of Heintz-Schnorr [HS80], Agrawal [Agr05] and Kabanets-Impagliazzo [KI04] show that non-trivial PIT algorithms *imply* lower bounds. As such, it seems that to obtain further progress in PIT algorithms we must examine the existing lower bound techniques and attempt to develop corresponding algorithms out of them (or develop new lower bounds).

In particular, the multilinear formula lower bounds of Raz and Raz-Yehudayoff [Raz06, Raz09, RY09] were only recently developed into a deterministic PIT algorithm by Oliveira, Shpilka, and Volk [OSV15]. While this represents progress in that their algorithm takes $\exp(n^{1-\Omega(1)})$-time, their algorithm requires $\exp(n^{\Omega(1)})$-time even for polynomial size depth-3 multilinear formulas. At present, it is unclear whether this is a limitation of their techniques or whether this is inherent to the lower bound techniques of Raz and Raz-Yehudayoff [Raz06, Raz09, RY09].

In this work, we study the more recent lower bound technique of *shifted partial derivatives* from the seminal papers of Kayal [Kay12] and Gupta-Kamath-Kayal-Saptharishi [GKKS14]. While these works have spawned many follow-up lower bound results ([KSS14a, FLMS14, KLSS14a, KS14a, KS14b, KLSS14b, KS14c, KS15]), there are no PIT algorithms (white-box or black-box) based on these ideas. In this work, we take the first paper of Kayal [Kay12] on this subject and translate his ideas into a black-box PIT algorithm by *scaling down* the lower bound and making it *robust*. That is, Kayal [Kay12] gave an exponential lower bound for the top-fan-in s when the monomial $x_1 \cdots x_n$ is expressed as $\sum_{i=1}^{s} f_i(\overline{x})^{d_i}$ where $\deg f_i \leq 2$ (a sum of powers of quadratic polynomials, denoted $\sum \bigwedge \sum \prod^2$) and we give a quasipolynomial-time deterministic black-box PIT algorithm for this same model. While in retrospect our new algorithm is a rather simple extension of the methods of Forbes-Shpilka [FS13a] (who did the same for sums of powers of *linear* polynomials (denoted $\sum \bigwedge \sum$), using the partial derivative method of Nisan-Wigderson [NW96]), there were conceptual reasons to believe this extension was not possible (as we discuss below).

To further demonstrate the novelty of the method, note that sums of powers of linear polynomials are a sub-model of roABPs and as such have efficient deterministic PIT algorithms. It thus is natural to ask how the complexity changes when going from powers of linear polynomials to powers of

[2] These results were originally phrased in terms of non-commutative ABPs. For the explicit translation to roABPs, see the thesis of Forbes [For14].

[3] An exception is the paper of Gurjar, Korwar, Saxena, and Thierauf [GKST15], who design a new PIT algorithm for sums of roABPs (in varying variable orders). While this algorithm requires new ideas beyond those of Raz-Shpilka [RS05], it is to some extent in the same spirit.

quadratic polynomials. To address this question, we give an explicit power of a quadratic polynomial which requires exponentially large roABPs even under partial substitutions (and in any variable order). While weaker versions of this result are folklore, this result more strongly shows that the techniques of roABPs cannot address sums of powers of quadratic polynomials.

Divisibility Testing: We now turn to the second computational question we study, that of divisibility testing. In this problem, we are given *two* algebraic circuits computing polynomials $f(\overline{x})$ and $g(\overline{x})$ respectively, and the problem is to decide whether $f(\overline{x})$ divides $g(\overline{x})$ (denoted $f|g$). As in PIT, this question can also be asked in the black-box model where we are only allowed to use the circuits to gain access to the evaluation oracles of f and g. Note that unlike PIT, it is not immediately obvious whether there is *any* efficient randomized algorithm for this question, much less an algorithm in the black-box model. However, such an algorithm can be derived from the works of Kaltofen and Trager [Kal89, KT90]. That is, in the black-box model they gave a randomized algorithm that takes an evaluation oracle for a polynomial $h(\overline{x})$ and produces evaluation oracles for the irreducible factors of h. In the white-box model where h is given as an algebraic circuit they showed that they can even compute small algebraic circuits for the irreducible factors of h. Given these algorithms, one can decide whether f divides g by computing their respective irreducible factors and then checking that the multiplicities of each factor is at least as large in g as in f [4].

While the above provides a randomized algorithm (by solving the harder problem of factorization), it leaves open the question of obtaining a deterministic algorithm. However, as PIT efficiently reduces to divisibility testing[5] it is clear that to provide deterministic divisibility algorithms one first needs deterministic PIT algorithms. Conversely, the recent work of Kopparty, Saraf and Shpilka [KSS14b] showed that one can derandomize the factorization algorithm of Kaltofen and Trager [Kal89, KT90] using a deterministic PIT algorithm. As such, this shows that divisibility testing and PIT are *equivalent* in computational complexity. Indeed, many other works have shown similar results showing that derandomizing PIT suffices for other problems in computational algebra ([SV10, SSS13, DOS14, Vol14]).

While the above seemingly suggests that we understand the complexity of divisibility testing, the above equivalence with PIT only works for *general* algebraic circuits. That is, even if one asks for divisibility testing of restricted algebraic circuits then the above reduction yields PIT instances of seemingly difficult complexity (that is, involving determinants). While Shpilka and Volkovich [SV10] have shown a reduction from factoring multilinear polynomials to PIT that roughly preserves the complexity of the original computation, this reduction is highly tailored to multilinear polynomials. Indeed, even deterministically factoring sparse polynomials is an open question of von zur Gathen and Kaltofen [vzGK85] because it is not known whether the factors of sparse polynomials are sparse (in large characteristic).

To the best of our knowledge, it is even an open question to deterministically test whether f divides g when both f, g are sparse[6]. The only deterministic divisibility testing algorithm we are aware of[7] is a polytime algorithm for deciding whether the linear polynomial f divides the

[4] This can also be viewed as computing the greatest common divisor of f and g, and then checking that this is f.

[5] Note that $0|g(\overline{x})$ iff $g(\overline{x}) = 0$. Perhaps less trivially, $y|(g(\overline{x}) + y)$ iff $g(\overline{x}) = 0$ if $\overline{x}$ and y are variable disjoint.

[6] Dvir and Oliviera [DO14] showed that if factors of sparse polynomials are sparse then one can efficiently (within quasipolynomial time) do divisibility testing when f, g are sparse. They also claimed to prove this conjecture about the sparsity of factors of sparse polynomials, but withdrew their claims due to an error.

[7] There are also several works ([KK05, KK06, CGK+13a, Gre14a] and references there-in) giving deterministic algorithms for determining low-degree factors of sparse multivariate polynomials of *exponential* degree (known as *lacunary* polynomials), thus implying similar divisibility testing algorithms. However, these deterministic algorithms have a runtime which has an exponential dependence on the number of variables. As such, their results seem incomparable to the ones of this paper.

sparse polynomial g [8]. This follows from the work of Saha, Saptharishi and Saxena [SSS13] who reduced this question (via long-division) to PIT of expressions of the form $\sum_{i=1}^{s} \overline{x}^{\overline{a}_i} f_i(\overline{x})^{d_i}$, where $\overline{x}^{\overline{a}}$ is the monomial $\prod_{i=1}^{n} x_i^{a_i}$ and $\deg f_i \leq 1$ (which we denote $\sum \mathrm{m} \bigwedge \sum \prod^1$). They then (in the reinterpretation of Forbes-Shpilka [FS13b]) reduced this question to PIT of roABPs from which the above mentioned PIT algorithms apply.

In this work, we ask for divisibility testing of when f is quadratic and g is sparse. Via long-division (or the reduction mentioned below), one can show that this reduces to PIT of expressions of the form $\sum_{i=1}^{s} \overline{x}^{\overline{a}_i} f_i(\overline{x})^{d_i}$ where $\deg f_i \leq 2$ (which we denote $\sum \mathrm{m} \bigwedge \sum \prod^2$). Aside from the monomials $\overline{x}^{\overline{a}_i}$, this is exactly the $\sum \bigwedge \sum \prod^2$ mentioned above. However, the known hard polynomial of Kayal [Kay12] for $\sum \bigwedge \sum \prod^2$ is the monomial $\overline{x}^{\overline{1}}$, whereas in this new $\sum \mathrm{m} \bigwedge \sum \prod^2$ model this monomial is trivial to compute. Thus, we use the *translation* idea of Agrawal, Saha, and Saxena [ASS13] to instead consider the monomial $(\overline{x} + \overline{1})^{\overline{1}} := \prod_i (x_i + 1)$ to be the hard polynomial for $\sum \mathrm{m} \bigwedge \sum \prod^2$ formulas. Using a variant of the shifted partial derivative method we are able to give exponential lower bounds for $(\overline{x} + \overline{1})^{\overline{1}}$ as a $\sum \mathrm{m} \bigwedge \sum \prod^2$ formula, and following the recipe mentioned before (of scaling-down the lower bound and making it robust) we obtain quasipolynomial black-box PIT algorithms for this model. Plugging this into our reduction, we obtain a deterministic quasipolynomial-time black-box algorithm to decide whether the quadratic f divides the sparse g.

The above reduction from divisibility testing to PIT using long division is highly dependent on the fact that f is low-degree, and does not give a general reduction from divisibility of sparse f into sparse g to a polynomial-size PIT problem. However, we remedy this situation by providing such a general reduction from divisibility testing to PIT that avoids the need for factorization and also roughly preserves the complexity of the original algebraic circuits. Instead of using long-division, we consider Strassen's [Str73] procedure for removing division gates from algebraic circuits. That is, if h, f, g are polynomials where $h = {}^{g}\!/\!_{f}$, Strassen [Str73] showed how to derive an algebraic circuit for h from circuits for f and g. We observe here that this procedure is still well-defined even if $f \nmid g$, in which case it produces some polynomial $\widetilde{{}^{g}\!/\!_{f}}$. Thus, one can check whether $f|g$ by checking that $g - f \cdot \widetilde{{}^{g}\!/\!_{f}} = 0$. By a careful inspection of Strassen's [Str73] procedure this shows that the complexity of this computation is not too far above that of f and g. Unfortunately, the PIT problems that arise when f and g are sparse are still beyond what is known how to perform deterministically.

1.1 Our Results and Techniques

We now more formally discuss our results and techniques.

PIT for $\sum \bigwedge \sum \prod^t$ formulas: We begin by defining $\sum \bigwedge \sum \prod^t$ formulas (in general we are most interested in $t = \mathcal{O}(1)$), one of the main classes of algebraic computation of interest in this paper.

Definition 1.1. *A polynomial $f(\overline{x}) \in \mathbb{F}[x_1, \ldots, x_n]$ is **computable by a $\sum \bigwedge \sum \prod^t$ formula** if*

$$f(\overline{x}) = \sum_{i=1}^{s} f_i(\overline{x})^{d_i} \ ,$$

*where $\deg f_i \leq t$. The **size** of the formula is $\sum_i (d_i + 1) \cdot \binom{n+t}{t}$. When $t = 1$ we write this as $\sum \bigwedge \sum$.* ◊

We use the notation '$\bigwedge$' to denote *powering*, following Gupta, Kamath, Kayal, and Saptharishi [GKKS13]. This notation is meant to be suggestive of the exponentiation character '^'. Note that one could also consider constants α_i in front of the $f_i^{d_i}$ and modify our results essentially

[8] One can actually show that this algorithm works if f is a sum of univariates and g is computable by small roABP.

without change. However since our algorithms will mostly be black-box, we can think of these constants as being absorbed into the $f_i^{d_i}$ by taking d_i-th roots.

The notion of a $\sum\bigwedge\sum$ formula is well-studied but not completely understood. This notion has been previously studied under the name of a "*depth-3 diagonal formula*", as named by Saxena [Sax08]. These formulas are also classically studied in mathematics, where the size of these formulas is known as *symmetric tensor rank* or *Waring rank* of a polynomial, see for example Landsberg [Lan12]. These formulas are in some sense the weakest complete model of algebraic computation for which we do not have a complete understanding; in particular, we lack polynomial-time deterministic black-box PIT algorithms. We now briefly summarize what is known about these formulas, in particular how this class lies in the intersection of two techniques in algebraic complexity theory (roABPs and the partial derivative method of Nisan-Wigderson [NW96]).

We begin by reviewing the definition of roABPs. While there are equivalent definitions that are more flexible (see for example Forbes [For14, Section 4.4]), the definition we use is a normal form for this type of computation.

Definition 1.2. *Let $\mathbb{F}$ be a field, $n \geq 1$, and let $\pi : [n] \to [n]$ be a permutation. The polynomial $f(\overline{x}) \in \mathbb{F}[x_1, \ldots . x_n]$ is computable by a* ***read-once (oblivious) algebraic branching program (roABP) with variable order*** *π, with depth-n, individual degree-$(\leq d)$ and width-$(\leq w)$, if there exist matrices $M_i(x_{\pi(i)}) \in \mathbb{F}[x_{\pi(i)}]^{(\leq w)\times(\leq w)}$ of (individual) degree $\leq d$ such that*

$$f(\overline{x}) = \left(\prod_{i=1}^{n} M_i\left(x_{\pi(i)}\right)\right)_{1,1} ,$$

where the sizes of the matrices is such that this product is well-defined. The ***size*** *of such a roABP is defined to be nw^2d. The roABP computes in a* ***known order*** *if π is a fixed known permutation (such as the identity permutation), and computes in an* ***unknown order*** *if π exists but is unknown. We say that $f(\overline{x})$ is computed by a* ***commutative roABP*** *if it is computed by an roABP in* every *variable order π.* ♢

One can also force (by padding) the M_i to *all* be width w.

Saxena [Sax08] (using his *duality trick*) implicitly (made explicit in Forbes-Shpilka [FS13b]) showed that $\sum\bigwedge\sum$ formulas are computable by small *commutative* roABPs, and this is stated more generally in the following result.

Lemma 1.3 (Saxena [Sax08], Forbes-Shpilka [FS13b])**.** *Let $\mathbb{F}$ be any field. Let $f(x_1, \ldots, x_n)$ be computed by a $\sum\bigwedge\sum\bigwedge$ formula (diagonal depth-4 formula), so that*

$$f(\overline{x}) = \sum_{i=1}^{s} f_i(\overline{x})^{a_i} ,$$

where each f_i is a sum of univariate polynomials, so that $f_i(\overline{x}) = \sum_{j\in[n]} f_{i,j}(x_j)$, and where $\deg f_i \leq d$. *Then for any variable order $\pi : [n] \to [n]$, f is computed by a $\mathsf{poly}(\sum_{i\in[s]} a_i, n)$-explicit* [9] *width-$\left(\sum_{i\in[s]} (a_i + 1)\right)$ roABP, of individual degree $\leq d \max_{i\in[s]} a_i$, in the variable order π.* □

Nisan [Nis91] essentially gave exponential lower bounds for computing the determinant as a roABP, so this lower bound similarly extends to $\sum\bigwedge\sum$. Further, as the above reduction is explicit and Raz-Shpilka [RS05] gave a polytime white-box PIT algorithm for roABPs, these results combine to give such a corresponding algorithm for $\sum\bigwedge\sum$ (reinterpreting the paper of Saxena [Sax08]).

[9] In this paper, *explicit* means computable in a Turing machine model where there are special registers holding elements of $\mathbb{F}$ and all $\mathbb{F}$-operations on these registers are unit cost.

Initially, black-box PIT algorithms for $\sum\bigwedge\sum$ formulas all essentially worked also for commutative roABPs ([FS12, ASS13, FS13b]) and had complexity $\mathsf{poly}(nwd)^{\log n}$ for width-w, n-variate, degree $\le d$ commutative roABPs. Obtaining better algorithms for roABPs (or even commutative roABPs) seems challenging.

However, $\sum\bigwedge\sum$ formulas are also simple with respect to the *partial derivative* measure of Nisan and Wigderson [NW96]. That is, to a polynomial f one considers the space $\boldsymbol{\partial}_{\overline{x}^{<\infty}}(f) := \{\partial_{\overline{x}^{\overline{a}}} f\}_{\overline{a}}$, where $\partial_{\overline{x}^{\overline{a}}} f$ is the derivative $\partial_{x_1^{a_1}\cdots x_n^{a_n}}$ of f, and the exponent vector $\overline{a}$ ranges over all derivatives. In particular, one looks at the *dimension* of $\boldsymbol{\partial}_{\overline{x}^{<\infty}}(f)$ as a vector space. Kayal (see Saxena [Sax08]) observed that polynomial-size $\sum\bigwedge\sum$ formulas have a partial derivative space of low-dimension, while the monomial $x_1\cdots x_n$ has $\dim\boldsymbol{\partial}_{\overline{x}^{<\infty}}(x_1\cdots x_n) = 2^n$ showing that $x_1\cdots x_n$ requires exponential size as a $\sum\bigwedge\sum$ formula (which is tight by Fischer [Fis94]).

In contrast, roABPs very easily compute the monomial $x_1\cdots x_n$ and thus the partial derivative method seems to give more insight into $\sum\bigwedge\sum$ formulas. Based on this insight, and using the *hardness of representation* idea of Shpilka and Volkovich [SV09], Forbes-Shpilka [FS13a] gave a $\mathsf{poly}(s)^{\log s}$-time black-box PIT algorithm for size-s $\sum\bigwedge\sum$ formulas. As our techniques are a generalization of theirs, we briefly describe their approach. We begin by *scaling down* the above lower bound. That is, observe that *if* a size-s $\sum\bigwedge\sum$ computes a monomial $\overline{x}^{\overline{a}}$, which involves $\|\overline{a}\|_0 := \{i \mid a_i \neq 0\}$) many variables, then it must be (by the above lower bound) that $\|\overline{a}\|_0 \le \mathcal{O}(\lg s)$. The next observation is that the above lower bound is *robust*. That is, if $f(\overline{x}) = \overline{x}^{\overline{a}} + o(\overline{x}^{\overline{a}})$ in that we mean "$o(\overline{x}^{\overline{a}})$" to consist of lower order terms (say, with respect to the lexicographic ordering), then it follows that the measure of f is at least that of its leading monomial $\overline{x}^{\overline{a}}$, that is, $\dim\boldsymbol{\partial}_{\overline{x}^{<\infty}}(f) \ge \dim\boldsymbol{\partial}_{\overline{x}^{<\infty}}(\overline{x}^{\overline{a}})$. The measure $\dim\boldsymbol{\partial}_{\overline{x}^{<\infty}}$ is thus *robust* in that it ignores lower order terms. From these facts, Forbes-Shpilka [FS13a] deduced that any size-s $\sum\bigwedge\sum$ formula must compute a monomial (in fact, its leading monomial) that involves $\mathcal{O}(\lg s)$ variables. That is, any such $\sum\bigwedge\sum$ formula computing a non-zero polynomial must compute a polynomial with a *small-support monomial.* A brute force algorithm on this small-support monomial (see Corollary 3.15) then yields the desired PIT algorithm.

Thus, the above two methodologies offer different ways to obtain quasipolynomial-time PIT algorithms for $\sum\bigwedge\sum$ formulas. However, somewhat surprisingly, Forbes, Shpilka and Saptharishi [FSS14] showed how to *combine* these two approaches to obtain $\mathsf{poly}(s)^{\mathcal{O}(\lg\lg s)}$-time black-box PIT for size-s $\sum\bigwedge\sum$ formulas (see Theorem 3.16). We will further discuss this approach below, as our variant of shifted partial derivatives along with this method allows us to derive $\mathsf{poly}(s)^{\mathcal{O}(\lg\lg s)}$-time black-box PIT for size-s *translations of sparse polynomials*, which improves on the previous best runtime of $\mathsf{poly}(s)^{\mathcal{O}(\lg s)}$ ([FS12, ASS13, FS13b]) which comes from viewing such polynomials as a subclass of commutative roABPs.

We now return to discuss $\sum\bigwedge\sum\prod^t$ formulas for $t > 1$. One may ask to what extent the above two approaches (roABPs and the partial derivative measure) generalize to this case. We address these techniques in order. It seems to be a folklore result (Lemma 8.5) that the $\sum\bigwedge\sum\prod^2$ formula $(\sum_{i=1}^n x_i y_i)^n$ requires an exponentially large roABP in any variable order where $\overline{x}$ precedes $\overline{y}$. However, this result is only for these special variable orders, and in fact this formula *is* computable by a small roABP in the variable order $x_1 < y_1 < \cdots < x_n < y_n$ (Lemma 8.7). Thus, while black-box PIT algorithms for roABPS in a fixed order (such as Forbes-Shpilka [FS13b]) would not work on this formula (without knowing the order), the black-box PIT algorithms for roABPs that work in an unknown order (such as Agrawal, Gurjar, Korwar, and Saxena [AGKS14]) would succeed, and thus this formula does not give a convincing example that roABP methods cannot succeed for PIT of $\sum\bigwedge\sum\prod^2$. Thus, we provide such an example.

Theorem (Informal Version of Corollary 8.21)**.** *Let $\mathbb{F}$ be a field of characteristic $\ge \mathsf{poly}(n)$. Then* *there is an explicit n-variate $\sum\bigwedge\sum\prod^2$ formula of degree n, such that under any partial substitution that leaves $\Omega(n)$ variables untouched, the resulting polynomial requires $\exp(\Omega(n))$-size roABPs in any variable order.* □

The explicit polynomial is of the form $(\overline{x}^t A\overline{x})^n$, where $A \in \mathbb{F}^{n\times n}$ is a totally non-singular matrix. Thus, this polynomial essentially embeds the hard polynomial $(\sum_{i=1}^n x_i y_i)^n$ under any partition of the variables, but more work is required to formalize this intuition.

We now turn to the partial derivative method. As it is folklore that $\sum\bigwedge\sum\prod^2$ formulas (such as $(\sum_{i=1}^n x_i^2)^n$) can have exponentially large partial derivative space, it follows that this method alone will not generalize from $\sum\bigwedge\sum$ to $\sum\bigwedge\sum\prod^2$. However, Kayal's [Kay12] *shifted partial derivatives* were discovered exactly for this purpose. This operator maps a polynomial f to the space $\boldsymbol{\overline{x}}^{\le \ell}\boldsymbol{\partial}_{\overline{x}^{\le k}}(f) := \{\overline{x}^{\overline{b}}\partial_{\overline{x}^{\overline{a}}}(f)\}_{\overline{b},\overline{a}}$ where the exponent vectors $\overline{a},\overline{b}$ are chosen so that $\deg \overline{x}^{\overline{b}} \le \ell$ and $\deg \overline{x}^{\overline{a}} \le k$. For $\ell = 0$ one recovers the partial derivative method of Nisan and Wigderson [NW96]. By carefully choosing parameters via the analysis of Gupta-Kamath-Kayal-Saptharishi [GKKS14], Kayal [Kay12] obtained the following theorem (which as Kayal [Kay12] notes, is tight).

Theorem (Kayal [Kay12], Gupta-Kamath-Kayal-Saptharishi [GKKS14], see also Corollary 4.19)**.** *For any field $\mathbb{F}$, computing $x_1 \cdots x_n$ as a $\sum\bigwedge\sum\prod^t$ formula requires top-fan-in $\ge \exp(\Omega(n/t))$.* □

While the above seems very similar to the analogous lower bound for $\sum\bigwedge\sum$ formulas, there is a very tangible difference. That is, for polynomial-size $\sum\bigwedge\sum$ formulas the $\dim \boldsymbol{\partial}_{\overline{x}^{<\infty}}$ measure is polynomially-bounded. In contrast, for the values of k, ℓ chosen for the above lower bound, the dimension $\dim \boldsymbol{\overline{x}}^{\le\ell}\boldsymbol{\partial}_{\overline{x}^{\le k}}(\sum\bigwedge\sum\prod^t)$ is *exponentially* large. That is, to obtain the above lower bound Kayal [Kay12, GKKS14] had to show that this exponential is exponentially smaller than the corresponding measure of the monomial $x_1 \cdots x_n$.

The largeness of this measure seems very problematic for designing PIT algorithms. That is, PIT algorithms often seek to reduce PIT of n-variate polynomials to polynomials on m-variate polynomials for $m \ll n$. Sometimes one arrives at $m = 1$ directly in which case univariate interpolation is then applied, and other times one gets $m = n/2$ to which recursion is applied. However, in order for these reductions to succeed one needs to argue that non-zero polynomials *remain non-zero*. For inductive purposes this often requires preserving more than just non-zeroness alone, and the amount of information to preserve is often quantified by the measure of complexity of the computation (for example, in roABPs one needs to preserve $\approx w$ amount of information in width-w roABPs). However, in shifted partial derivatives this measure is exponentially large and thus there is no efficient way to preserve this amount of information while reducing the amount of variables.

However, despite this obstacle we show how to extend the methods of Forbes-Shpilka [FS13a] for the partial derivative space to shifted partials, in particular by scaling down Kayal's [Kay12, GKKS14] bound and making it robust. As such, we arrive at the following theorem.

Theorem (Informal version of Proposition 4.18, Corollary 4.20)**.** *Let $\mathbb{F}$ be a field of characteristic $> d$. Then the "leading monomial" of a size-s $\sum\bigwedge\sum\prod^t$ formula involves $\mathcal{O}(t \lg s)$ variables. In particular, there is a deterministic $\mathsf{poly}(s)^{\mathcal{O}(t\lg s)}$-time black-box PIT algorithm for size-s $\sum\bigwedge\sum\prod^t$ n-variate, degree-$(\le d)$ formulas.* □

This is the first non-trivial (white-box or black-box) PIT algorithm for this class of computations. In particular, the above algorithm is black-box and no better white-box algorithm is known.

PIT of $\sum \mathrm{m} \bigwedge\sum\prod^t$: Given the above results on $\sum\bigwedge\sum\prod^t$, we now seek to generalize this to a larger class of formulas, called $\sum \mathrm{m} \bigwedge\sum\prod^t$, which we now define. We motivate this class in two ways. The first way is that this class naturally contains both $\sum\bigwedge\sum\prod^t$ and sparse polynomials. While the monomial $x_1 \cdots x_n$ is hard for $\sum\bigwedge\sum\prod^t$, it is very easy to compute as a sparse polynomial. As such, obtaining lower bounds simultaneously against *both* $\sum\bigwedge\sum\prod^t$ and sparse polynomials seems to be a challenge. While for $t = 1$ both of these models are subsumed by roABPs so that the methods from that literature apply, our results show these methods are not relevant for $t > 1$. The second motivation comes from the mentioned connections with divisibility testing. That is, we will describe in the next section how testing whether a degree-t polynomial divides a sparse polynomial reduces to PIT of this $\sum \mathrm{m}\bigwedge\sum\prod^t$ class. Given this motivation, we now define this class.

Definition 1.4. *A polynomial $f(\overline{x}) \in \mathbb{F}[x_1, \ldots, x_n]$ is* ***computable by a*** $\sum \mathrm{m}\bigwedge\sum\prod^t$ ***formula*** *if*

$$f(\overline{x}) = \sum_{i=1}^{s} \overline{x}^{\overline{a}_i} f_i(\overline{x})^{d_i} \ ,$$

where $\deg f_i \le t$. *The* ***size*** *of the formula is* $\sum_i (\deg \overline{x}^{\overline{a}_i} + (d_i + 1)\binom{n+t}{t})$. *When $t = 1$ we write this as* $\sum \mathrm{m}\bigwedge\sum$. ◊

When $t = 1$, the notion of a $\sum \mathrm{m}\bigwedge\sum$ formula has been previously studied under the name of a *semi-diagonal depth-3 formula* by Saha, Saptharishi, and Saxena [SSS13]. We rename this class here to mimic the above notation for a $\sum\bigwedge\sum$ formula. For arbitrary t, a more globally consistent name for this class would be "$\sum(\prod) \cdot (\bigwedge\sum^t)$", as the "$(\prod)$" here indicates a monomial and "$(\bigwedge\sum^t)$" indicates the power of a degree-$(\le t)$ polynomial. However, this notation seems cumbersome, and thus we instead opt for "m" to indicate the presence of the extra monomial added to the $\sum\bigwedge\sum\prod^t$ formula.

We begin by discussing the $t = 0$ case, that is, sparse polynomials. The partial derivative method does not yield good results for these polynomials because the monomial $\overline{x}^{\overline{1}} := x_1 \cdots x_n$ has a large space of partial derivatives. However, observe that sparse polynomials are not closed under translation, so that $(\overline{x} + \overline{1})^{\overline{1}} := \prod_i (x_i + 1)$ is very *non-sparse.* Thus, even though the monomial $\overline{x}^{\overline{1}}$ is easy for sparse polynomials to compute, the *translated* monomial $(\overline{x} + \overline{1})^{\overline{1}}$ is hard to compute. While this lower bound is trivial to obtain, we now seek to scale it down and make it robust as done for $\sum\bigwedge\sum\prod^t$ formulas. That is, consider a s-sparse polynomial $f(\overline{x}) = \sum_{i=1}^{s} \alpha_i \overline{x}^{\overline{a}_i}$. Suppose the translation $f(\overline{x} + \overline{1})$ computes the monomial $\overline{x}^{\overline{a}}$. Scaling down the previous argument it follows that $\overline{x}^{\overline{a}}$ must involve $\mathcal{O}(\lg s)$ variables, as $f(\overline{x}) = (\overline{x} - \overline{1})^{\overline{a}}$ must have sparsity $\le s$. Now consider robustness. For size-s $\sum\bigwedge\sum\prod^t$ formulas we argued that in some sense the *first* monomial must involve few variables as this monomial is the "dominant term". However, for sparse polynomials this is false as the first monomial of $(\overline{x} + \overline{1})^{\overline{1}}$ is $\overline{x}^{\overline{1}}$, which involves many variables. However, the key insight here is that the *last* monomial (which in the previous example is 1) must involve few variables. Indeed, this is what we can show.

Theorem (Corollary 5.14)**.** *Let $f(\overline{x}) \in \mathbb{F}[\overline{x}]$ be $(\le s)$-sparse. Then the "last" monomial of $f(\overline{x} + \overline{1})$ involves $\le \lg s$ variables.* □

We prove the above using a variant of the partial derivative method. That is, consider the differential operator $(x + 1)\partial_x$ which maps $f \mapsto (x + 1) \cdot \partial_x(f)$. Note that for a sparse polynomial in $x + 1$ basis such as $(x + 1)^2$, this operator leaves the polynomial unchanged (up to a constant). However, for a polynomial sparse in the x-basis such as x^2, it gets mapped to the "different" $2x(x+1)$. In general, we will consider the space $((\boldsymbol{\overline{x}} + \boldsymbol{\overline{\alpha}}) \circ \boldsymbol{\partial}_{\overline{x}})^{\le k}(f) := \{(\overline{x} + \overline{1})^{\overline{b}} \cdot \partial_{\overline{x}^{\overline{b}}}(f)\}_{\overline{b}}$ where the exponent $\overline{b}$ ranges over monomials where $\deg \overline{x}^{\overline{b}} \le k$. Note that while this superficially seems similar to the space of shifted partial derivatives, there are some tangible differences. In particular, this space of differential operators *correlates* the "shift" $(\overline{x} + \overline{1})^{\overline{b}}$ with the derivative $\partial_{\overline{x}^{\overline{b}}}$. In shifted partial derivatives the shift and derivatives are uncorrelated. In particular, the dimension of shifted partial derivatives is translation invariant (Lemma 5.2) and as such *cannot* separate polynomials sparse in the $\overline{x}$ basis from polynomials sparse in the translated $\overline{x}+\overline{1}$ basis.

By the above arguments, if $f(\overline{x})$ is s-sparse (in the $\overline{x}$ basis) then $\dim((\boldsymbol{\overline{x}}+\boldsymbol{\overline{\alpha}})\circ\boldsymbol{\partial}_{\overline{x}})^{\le k}(f(\overline{x}+\overline{1}))$ is at most s. Further, we can show the following robustness property, that $\dim((\boldsymbol{\overline{x}}+\boldsymbol{\overline{\alpha}})\circ\boldsymbol{\partial}_{\overline{x}})^{\le k}(f(\overline{x}))\ge \dim\boldsymbol{\partial}_{\overline{x}^{\le k}}(\mathrm{TM}(f))$, where $\mathrm{TM}(f)$ is the "trailing monomial" or "last monomial" of f. As the usual partial derivative method shows that $\dim\boldsymbol{\partial}_{\overline{x}^{\le k}}(\overline{x}^{\overline{a}})$ is at least $2^{\|\overline{a}\|_0}$ where $\|\overline{a}\|_0$ is the number of variables in the monomial $\overline{x}^{\overline{a}}$, this gives the above theorem.

While polynomial-time black-box PIT is already known for sparse polynomials (for example, see Klivans and Spielman [KS01]), those results crucially exploit sparsity as a *combinatorial* criteria instead of an *algebraic* one. In particular, these results do not work for the class of translated sparse polynomials, that is, the class of $\{f(\overline{x}+\overline{\alpha})|f(\overline{x})\ s\text{-sparse}\ ,\overline{\alpha}\in\mathbb{F}^n\}$. However, as the above methods are algebraic they are somewhat insensitive to translations. As such, we can obtain the following black-box PIT algorithm by combining our results with those for roABPs of Forbes, Shpilka, Saptharishi [FSS14] (Theorem 3.16).

Theorem (Corollary 5.16)**.** *Let $|\mathbb{F}|\ge \mathsf{poly}(n,d,s)$. There is a $\mathsf{poly}(n,d,s)^{\mathcal{O}(\lg\lg s)}$-time black-box PIT algorithm for the class of polynomials $f(\overline{x})\in\mathbb{F}[x_1,\ldots,x_n]$ that are translations of some s-sparse polynomial.* □

Prior to this work the best such black-box algorithm came from PIT of roABPs and thus took $\mathsf{poly}(n,d,s)^{\Theta(\lg n)}$ time.

Given that the above variant of the partial derivative method sufficed to understand $\sum\mathrm{m}\bigwedge\sum\prod^t$ for $t=0$, it seems logical that applying this variant to the *shifted* partial derivative method would yield similar results for $t>0$ when combined with our results for $\sum\bigwedge\sum\prod^t$ formulas. Indeed, we show the following result.

Theorem (Informal version of Proposition 6.5, Corollary 6.7)**.** *Let $\mathbb{F}$ be a field of characteristic $>d$. If $f(\overline{x})$ is a size-s $\sum\mathrm{m}\bigwedge\sum\prod^t$ formula then the "trailing/last monomial" of $f(\overline{x}+\overline{1})$ involves $\mathcal{O}(t\lg s)$ variables. In particular, there is a deterministic $\mathsf{poly}(s)^{\mathcal{O}(t\lg s)}$-time black-box PIT algorithm for size-s $\sum\mathrm{m}\bigwedge\sum\prod^t$ n-variate, degree-$(\le d)$ formulas.* □

One can also obtain such a result for $\sum\mathrm{m}\bigwedge\sum\prod^t$ under translation[10]. Now, noting that $\sum\mathrm{m}\bigwedge\sum$ formulas are computable as commutative roABPs (Lemma 3.18) we can obtain the following black-box PIT algorithm for $\sum\mathrm{m}\bigwedge\sum$ by combining our results with those for roABPs of Forbes, Shpilka, Saptharishi [FSS14] (Theorem 3.16).

Theorem (Informal version of Corollary 6.8)**.** *Let $\mathbb{F}$ be a field of characteristic $>d$. There is a deterministic $\mathsf{poly}(s)^{\mathcal{O}(\lg\lg s)}$-time black-box PIT algorithm for size-s $\sum\mathrm{m}\bigwedge\sum$ n-variate, degree-$(\le d)$ formulas.* □

As with translations of sparse polynomials, the previous best black-box PIT algorithm for this class required $\mathsf{poly}(s)^{\Theta(\lg n)}$ time.

Reducing Divisibility Testing to PIT: As sketched above, we use Strassen's [Str73] elimination of divisions to give a deterministic reduction in the black-box model from testing whether $f(\overline{x})$ divides $g(\overline{x})$ to a PIT problem of a polynomial $h(\overline{x})$ which is not too much more complicated than f and g. As an example of such a reduction, consider the following lemma.

Lemma 1.5. *Let $f(\overline{x})\in\mathbb{F}[\overline{x}]$ and $g(\overline{x},y)\in\mathbb{F}[\overline{x},y]$. Then $(y-f(\overline{x}))|g(\overline{x},y)$ iff $g(\overline{x},f(\overline{x}))=0$.* □

[10] In a future version of this work, we will show that these results also hold for a sum of a constant number of $\sum\mathrm{m}\bigwedge\sum\prod^t$ formula under *different* translations.

This lemma can be proven via long-division of multivariate polynomials (see for example Cox-Little-O'Shea [CLO07]). In particular, if h is a linear polynomial then without loss of generality it is of the form $y - f(\overline{x})$ for some distinguished variable y. If $g(\overline{x}, y)$ is sparse, then $g(\overline{x}, f(\overline{x}))$ is then a $\sum \mathrm{m} \bigwedge \sum$ formula as noted by Saha, Saptharishi and Saxena [SSS13]. With more work, one can push[11] the long-division method to reduce divisibility of a degree-t f into a sparse g to PIT of $\sum \mathrm{m} \bigwedge \sum \prod^t$. However, the resulting formula has size exponential in t and thus this approach seems limited to $t = \mathcal{O}(1)$.

Instead, as sketched above, we reduce "$f|g$?" to PIT of a polynomial h, where h is constructed using Strassen's [Str73] elimination of divisions. By carefully inspecting this procedure we can bound the complexity of h, yielding the following result.

Theorem (Informal verison of Corollary 7.11)**.** *Let $\mathbb{F}$ be a field with $|\mathbb{F}| \geq \mathsf{poly}(d)$. Let $\mathcal{C}_1, \mathcal{C}_2 \subseteq \mathbb{F}[x_1, \ldots, x_n]$ be two classes of n-variate degree-($\leq d$) polynomials which are closed under the operations $f(\overline{x}) \mapsto f(\alpha \cdot \overline{x} + \overline{\beta})$ for any $\alpha \in \mathbb{F}$ and $\overline{\beta} \in \mathbb{F}^n$. Then testing divisibility of $\mathcal{C}_1$ polynomials into $\mathcal{C}_2$ polynomials is efficiently reducible to polynomial identity testing of $\sum^{\mathsf{poly}(d)} \mathcal{C}_1 \cdot \mathcal{C}_2 \cdot \bigwedge^{\mathsf{poly}(d)} \mathcal{C}_1$ polynomials in both the black-box and white-box models of computation, where $\sum^{\mathsf{poly}(d)} \mathcal{C}_1 \cdot \mathcal{C}_2 \cdot \bigwedge^{\mathsf{poly}(d)} \mathcal{C}_1$ is the class of polynomials $\left\{\sum_{i=1}^s \alpha_i \cdot f_i(\overline{x}) \cdot g_i(\overline{x}) \cdot h_i(\overline{x})^{d_i} \mid f_i, h_i \in \mathcal{C}_1, g_i \in \mathcal{C}_2, s, d_i \leq \mathsf{poly}(d)\right\}$. In particular, there is an efficient randomized algorithm for divisibility testing.* □

However, the above theorem as stated is limited in that it needs the closure of $\mathcal{C}_1$ and $\mathcal{C}_2$ under the maps $f(\overline{x}) \mapsto f(\alpha \cdot \overline{x} + \overline{\beta})$. In particular, sparse polynomials are not closed under this operation. However, by tailoring our reduction to the specific divisibility testing problem of whether a constant-degree polynomial divides a sparse polynomial and applying our PIT results we obtain the following.

Theorem (Main Result, Informal Version of Corollary 7.17)**.** *Let $\mathbb{F}$ be a field $\mathrm{char}(\mathbb{F}) \geq \mathsf{poly}(d)$. Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ where $\deg f \leq \mathcal{O}(1)$. Let $g \in \mathbb{F}[\overline{x}]$ be of degree $\leq d$ and be computable by a size-s $\sum \mathrm{m} \bigwedge \sum \prod^{\mathcal{O}(1)}$ formula. Then computing the multiplicity of f as a factor of g can be deterministically computed in $\mathsf{poly}(s, n, d)^{\mathcal{O}(\lg s)}$-time in the black-box model. In particular, one can take g to be a s-sparse polynomial.* □

2 Notation

We briefly summarize some notation used in this paper.

Denote $[n] := \{1, \ldots, n\}$. For a vector $\overline{a} \in \mathbb{Z}^n$, we define $\mathrm{Supp}(\overline{a}) = \{i \mid a_i \neq 0\}$, $\|\overline{a}\|_0 := \sum_{i \in [n]} |a_i|^0 = |\mathrm{Supp}(\overline{a})|$, $\|\overline{a}\|_1 := \sum_{i \in [n]} |a_i|$, $\|\overline{a}\|_\infty := \max_{i \in [n]} |a_i|$.

Polynomials will often be written out in their monomial expansion. When "taking the coefficient of $\overline{y}^{\overline{b}}$ in $f \in \mathbb{F}[\overline{x}, \overline{y}]$" we mean that both $\overline{x}$ and $\overline{y}$ are treated as variables and thus the coefficient returned is a scalar in $\mathbb{F}$. However, when "taking the coefficient of $\overline{y}^{\overline{b}}$ in $f \in \mathbb{F}[\overline{x}][\overline{y}]$" we mean that $\overline{x}$ is now part of the ring of scalars, so the coefficient will be an element of $\mathbb{F}[\overline{x}]$.

The **support** of a polynomial $f(\overline{x}) = \sum_{\overline{a}} \alpha_{\overline{a}} \overline{x}^{\overline{a}}$ is the set $\{\overline{a} \mid \alpha_{\overline{a}} \neq 0\}$, which we sometimes identify with the set $\{\overline{x}^{\overline{a}} \mid \alpha_{\overline{a}} \neq 0\}$ The **(total) degree** of a polynomial is defined as $\deg f := \max_{\overline{a} \in \mathrm{Supp}(f)} \|\overline{a}\|_1$. The **individual degree** of a polynomial is defined as $\mathrm{ideg}\, f := \max_{\overline{a} \in \mathrm{Supp}(f)} \|\overline{a}\|_\infty$. Given a polynomial f in $\mathbb{F}[\overline{x}]$, we write $\mathrm{H}_k(f)$ for the **homogeneous part** of f of degree k.

The binomial coefficient $\binom{\overline{a}}{\overline{b}}$ is defined by $\binom{\overline{a}}{\overline{b}} := \prod_i \binom{a_i}{b_i}$, so that the binomial operator is applied coordinate-wise and then the coordinates are multiplied. Similarly for $\overline{f} \in \mathbb{F}[\overline{x}]^n$ be a vector of

[11] An exposition of this will be contained in a future version of this work, but this technique is essentially subsumed by the use of Strassen's [Str73] elimination of divisions.

polynomials and $\overline{a} \in \mathbb{N}^n$ an exponent vector. Then $\overline{f}^{\overline{a}} := \prod_{i=1}^n f_i^{a_i}$. This notation will be most often used as $\overline{x}^{\overline{a}}$ to denote the monomial $\prod_{i=1}^n x_i^{a_i}$.

3 Preliminaries

In this section we give various background definitions and results that will drive the rest of the paper. First, we give the definition of a *hitting set*. As it is well-known (see for example the thesis of Forbes [For14]) that deterministic black-box PIT algorithms are equivalent to the construction of explicit hitting sets, we will construct hitting sets to achieve our PIT algorithms.

Definition 3.1. *Let $\mathcal{C} \subseteq \mathbb{F}[x_1, \ldots, x_n]$ be a set of polynomials. A multi-set $\mathcal{H} \subseteq \mathbb{F}^n$ is a **hitting set for** $\mathcal{C}$ if for all $f \in \mathcal{C}$,*

$$f \equiv 0 \text{ iff } f|_{\mathcal{H}} \equiv 0 \, .$$

That is, $f(\overline{x}) = 0$ in $\mathbb{F}[\overline{x}]$ iff $f(\overline{\alpha}) = 0$ for all $\overline{\alpha} \in \mathcal{H}$.

*The hitting set $\mathcal{H}$ is $t(n)$-**explicit** if there is an algorithm such that given an index into $\mathcal{H}$, the corresponding element of $\mathcal{H}$ can be computed in $t(n)$-time in the unit-cost computation model over $\mathbb{F}$.* ◊

We now review Hasse derivatives, monomial orders and their properties, and how one can obtain hitting sets for polynomials with small-support monomials.

3.1 Hasse Derivatives

In this section we review the definition and properties of *Hasse derivatives*. These are like partial derivatives except that they work better in low characteristic fields, which will be important for the results of Section 5.

Definition 3.2. *The **Hasse derivative with respect to a monomial** $\overline{x}^{\overline{a}}$, denoted $\partial_{\overline{x}^{\overline{a}}}$, is the operator $\partial_{\overline{x}^{\overline{a}}} : \mathbb{F}[\overline{x}] \to \mathbb{F}[\overline{x}]$ that sends*

$$f(\overline{x}) \mapsto \mathrm{Coeff}_{\overline{z}^{\overline{a}}} \left(f(\overline{x} + \overline{z}) \right) \, .$$

That is, we take the coefficient of $\overline{z}^{\overline{a}}$ of $f(\overline{x} + \overline{z})$, when we view $f(\overline{x} + \overline{z})$ in the ring $\mathbb{F}[\overline{x}][\overline{z}]$. ◊

We now recall some standard properties of Hasse derivatives.

Lemma 3.3 (see for example Dvir, Koppary, Saraf, and Sudan [DKSS13] or Forbes [For14])**.** *Hasse derivatives have the following properties.*

- ***Linearity:*** *$\partial_{\overline{x}^{\overline{a}}}(\alpha f + \beta g) = \alpha \partial_{\overline{x}^{\overline{a}}}(f) + \beta \partial_{\overline{x}^{\overline{a}}}(g)$.*
- ***Product Rule:*** *$\partial_{\overline{x}^{\overline{a}}}(fg) = \sum_{\overline{b}+\overline{c}=\overline{a}} \partial_{\overline{x}^{\overline{b}}}(f) \partial_{\overline{x}^{\overline{c}}}(f)$.*
- ***Action on a Monomial:*** *$\partial_{\overline{x}^{\overline{b}}}(\overline{x}^{\overline{a}}) = \binom{\overline{a}}{\overline{b}} \overline{x}^{\overline{a}-\overline{b}}$, where this expression is still valid if $\overline{a} \not\geq \overline{b}$ (point-wise) in which case $\binom{\overline{a}}{\overline{b}} = 0$.* □

3.2 Monomial Orders

We recall here the definition and properties of a *monomial order*, following Cox, Little and O'Shea [CLO07]. We first fix the definition of a *monomial* in our context.

Definition 3.4. *A **monomial** in $\mathbb{F}[x_1, \ldots, x_n]$ is a polynomial of the form $\overline{x}^{\overline{a}} = x_1^{a_1} \cdots x_n^{a_n}$ for $\overline{a} \in \mathbb{N}^n$. We will sometimes abuse notation and identify the monomial $\overline{x}^{\overline{a}}$ with its exponent vector $\overline{a}$.*

*A **term** in $\mathbb{F}[\overline{x}]$ is a polynomial of the form $\alpha \cdot \overline{x}^{\overline{a}}$ for $\alpha \in \mathbb{F} \setminus \{0\}$ and $\overline{x}^{\overline{a}}$ a monomial.* ◊

Note that in this definition "1" is a monomial. We now define a monomial order, which will be total order on monomials with certain natural properties.

Definition 3.5. *A* ***monomial ordering*** *is a total order $\prec$ on the monomials in $\mathbb{F}[\overline{x}]$ such that*

- *For all $\overline{a} \in \mathbb{N}^n \setminus \{\overline{0}\}$, $1 \prec \overline{x}^{\overline{a}}$.*
- *For all $\overline{a}, \overline{b}, \overline{c} \in \mathbb{N}^n$, $\overline{x}^{\overline{a}} \prec \overline{x}^{\overline{b}}$ implies $\overline{x}^{\overline{a}+\overline{c}} \prec \overline{x}^{\overline{b}+\overline{c}}$.*

For $f \in \mathbb{F}[\overline{x}]$, the ***leading monomial of*** *f* ***(with respect to a monomial order*** *$\prec$***)***, denoted $\mathrm{LM}(f)$, is the largest monomial in $\mathrm{Supp}(f) := \{\overline{x}^{\overline{a}} : \mathrm{Coeff}_{\overline{x}^{\overline{a}}}(f) \neq 0\}$ with respect to the monomial order $\prec$. The* ***trailing monomial of*** *f, denoted $\mathrm{TM}(f)$, is defined analogously to be the smallest monomial in $\mathrm{Supp}(f)$. The zero polynomial has no leading or trailing monomial.*

We will often abuse notation and extend this total order to terms, so that for $\alpha, \beta \in \mathbb{F} \setminus \{0\}$, $\alpha\overline{x}^{\overline{a}} \prec \beta\overline{x}^{\overline{b}}$ iff $\overline{x}^{\overline{a}} \prec \overline{x}^{\overline{b}}$. We thus also have the notions of a ***leading term*** *and* ***trailing term*** *of a polynomial f, denoted $\mathrm{LT}(f)$ and $\mathrm{TT}(f)$ respectively, where $\mathrm{LT}(f) = \mathrm{Coeff}_{\mathrm{LM}(f)}(f) \cdot \mathrm{LM}(f)$ and $\mathrm{TT}(f) = \mathrm{Coeff}_{\mathrm{TM}(f)}(f) \cdot \mathrm{TM}(f)$.* $\diamondsuit$

Henceforth in this paper we will assume $\mathbb{F}[\overline{x}]$ is equipped with some monomial order $\prec$. The results in this paper will hold for *any* monomial order. However, for concreteness, one can consider the lexicographic ordering on monomials, which is easily seen to be a monomial ordering (see also Cox, Little and O'Shea [CLO07]).

While the notion of a leading monomial is well-studied because of its importance in Gröbner basis theory, trailing monomials lack such connections. Despite this, the trailing monomial will be more important for this work as further explained in Section 5.

3.3 Monomial Order Properties

We now list some basic properties of monomial orders. Our focus will be on trailing monomials and their interaction with basic operations such as multiplication and differentiation. Many of these properties will also hold for leading monomials, but as mentioned below some will not.

We begin with a simple lemma about taking trailing monomials is homomorphic with respect to multiplication.

Lemma 3.6. *Let $f, g \in \mathbb{F}[\overline{x}]$ be non-zero so that $fg \neq 0$. Then the trailing monomial is homomorphic with respect to multiplication, that is,*

$$\mathrm{TM}(fg) = \mathrm{TM}(f)\,\mathrm{TM}(g)\ .$$

Proof: Let $f(\overline{x}) = \sum_{\overline{a}} \alpha_{\overline{a}}\overline{x}^{\overline{a}}$ and $g(\overline{x}) = \sum_{\overline{b}} \beta_{\overline{b}}\overline{x}^{\overline{b}}$. Isolating the trailing monomials,

$$f(\overline{x}) = \alpha_{\mathrm{TM}(f)} \cdot \mathrm{TM}(f) + \sum_{\overline{x}^{\overline{a}} \succ \mathrm{TM}(f)} \alpha_{\overline{a}}\overline{x}^{\overline{a}}, \qquad g(\overline{x}) = \beta_{\mathrm{TM}(g)} \cdot \mathrm{TM}(g) + \sum_{\overline{x}^{\overline{b}} \succ \mathrm{TM}(g)} \beta_{\overline{b}}\overline{x}^{\overline{b}},$$

with $\alpha_{\mathrm{TM}(f)}, \beta_{\mathrm{TM}(g)} \neq 0$. Thus,

$$\begin{aligned} f(\overline{x})g(\overline{x}) = \alpha_{\mathrm{TM}(f)}\beta_{\mathrm{TM}(g)}\,\mathrm{TM}(f)\,\mathrm{TM}(g) + \alpha_{\mathrm{TM}(f)}\,\mathrm{TM}(f)\left(\sum_{\overline{x}^{\overline{b}} \succ \mathrm{TM}(g)} \beta_{\overline{b}}\overline{x}^{\overline{b}}\right) \\ + \beta_{\mathrm{TM}(g)}\,\mathrm{TM}(g)\left(\sum_{\overline{x}^{\overline{a}} \succ \mathrm{TM}(f)} \alpha_{\overline{a}}\overline{x}^{\overline{a}}\right) + \left(\sum_{\overline{x}^{\overline{a}} \succ \mathrm{TM}(f)} \alpha_{\overline{a}}\overline{x}^{\overline{a}}\right)\left(\sum_{\overline{x}^{\overline{b}} \succ \mathrm{TM}(g)} \beta_{\overline{b}}\overline{x}^{\overline{b}}\right). \end{aligned}$$

Using that $\overline{x}^{\overline{a}}\overline{x}^{\overline{b}} \succ \overline{x}^{\overline{a}} \operatorname{TM}(g), \operatorname{TM}(f)\overline{x}^{\overline{b}} \succ \operatorname{TM}(f) \operatorname{TM}(g)$ shows that $\operatorname{TM}(f) \operatorname{TM}(g)$ is indeed the minimal monomial in the above expression with non-zero coefficient. □

We now recall the monomial order's homomorphism with respect to (Hasse) differentiation, as given by Forbes-Shpilka [FS13a]. Note that this is somewhat subtle in that in a monomial order where $x \prec y$ we do not have that "$\partial_x(x) \prec \partial_x(y)$" as $\partial_x(y) = 0$ and "0" is not part of the ordering defined by '$\prec$'. However, whenever 0 does not occur the desired property holds.

Lemma 3.7 (Forbes-Shpilka [FS13a])**.** *Let $\overline{x}^{\overline{a}} \prec \overline{x}^{\overline{b}}$ be monomials. Then for any derivative $\partial_{\overline{x}^{\overline{c}}}$, if $\partial_{\overline{x}^{\overline{c}}}(\overline{x}^{\overline{a}}), \partial_{\overline{x}^{\overline{c}}}(\overline{x}^{\overline{b}}) \neq 0$ then $\partial_{\overline{x}^{\overline{c}}}(\overline{x}^{\overline{a}}) \prec \partial_{\overline{x}^{\overline{c}}}(\overline{x}^{\overline{b}})$ (as terms).* □

Forbes-Shpilka [FS13a] applied this to show that taking leading monomials (essentially) commutes with differentiation. We repeat this lemma here, but now with trailing monomials.

Lemma 3.8 (Forbes-Shpilka [FS13a])**.** *Let $f \in \mathbb{F}[\overline{x}]$. If $\partial_{\overline{x}^{\overline{a}}}(\operatorname{TM}(f)) \neq 0$, then $\operatorname{TM}(\partial_{\overline{x}^{\overline{a}}}(\operatorname{TM}(f))) = \operatorname{TM}\left(\partial_{\overline{x}^{\overline{a}}}(f)\right)$.* □

Note that $\partial_{\overline{x}^{\overline{a}}}(\operatorname{TM}(f))$ is only a *term* and not a *monomial* which is why we again took the trailing monomial to remove the scalar.

We now recall the fact that for any set of polynomials the dimension of their span in $\mathbb{F}[\overline{x}]$ is equal to the number of distinct trailing monomials in their span. This has been used in several papers on shifted partial derivatives, but where leading monomials are used instead.

Lemma 3.9 (see for example Forbes [For14, Lemma 8.4.12])**.** *Let $S \subseteq \mathbb{F}[\overline{x}]$ be a set of polynomials. Then $\dim \operatorname{span} S = |\operatorname{TM}(\operatorname{span} S)|$. In particular, $\dim \operatorname{span} S \geq |\operatorname{TM}(S)|$.* □

Note that when the set S of polynomials are all terms (or zero), we have that the trailing monomials of the span of the terms in S are simply the underlying monomials of the terms in S.

Lemma 3.10. *Let $S \subseteq \mathbb{F}[\overline{x}]$ be a set of terms (or zero). Then $\operatorname{TM}(\operatorname{span} S) = \operatorname{TM}(S)$.*

Proof: $\underline{\operatorname{TM}(\operatorname{span} S) \supseteq \operatorname{TM}(S):}$ Thus follows as $\operatorname{span} S \supseteq S$.

$\underline{\operatorname{TM}(\operatorname{span} S) \subseteq \operatorname{TM}(S):}$ First note that $\operatorname{span} S = \operatorname{span} \operatorname{TM}(S)$, as (even if $0 \in S$) $\operatorname{span} S$ is spanned by those (non-zero) terms $\alpha\overline{x}^{\overline{a}} \in S$ and $\operatorname{span} \operatorname{TM}(S)$ is spanned by those monomials $\{\overline{x}^{\overline{a}} \mid \exists \alpha \in \mathbb{F}\setminus\{0\}, \alpha\overline{x}^{\overline{a}} \in S\}$, and that these spanning sets span each other by non-zero re-weightings.

Now consider the possibly zero polynomial $f \in \operatorname{span} \operatorname{TM}(S)$, so that $f = \sum_{\overline{x}^{\overline{a}} \in \operatorname{TM}(S)} \alpha_{\overline{a}} \overline{x}^{\overline{a}}$. Thus, $\operatorname{Supp}(f) \subseteq \operatorname{TM}(S)$. If f is zero then it has no trailing monomial, otherwise $\operatorname{TM}(f) \in \operatorname{Supp}(f)$ by definition. □

Combining with the above lemmas yields the following.

Corollary 3.11. *Let $S \subseteq \mathbb{F}[\overline{x}]$ be a set of terms (or zero). Then $\dim \operatorname{span} S = |\operatorname{TM}(S)|$.* □

We now relate the trailing monomials of more complicated differential operators using Lemma 3.6 and Lemma 3.8 as building blocks.

Corollary 3.12. *Let $f, h \in \mathbb{F}[\overline{x}]$. Suppose that $\partial_{\overline{x}^{\overline{b}}}(\operatorname{TM}(f)) \neq 0$ and that $h(\overline{0}) \neq 0$, so that $\operatorname{TM}(h) = 1$. Then*

$$\operatorname{TM}\left(\overline{x}^{\overline{c}} \cdot h \cdot \partial_{\overline{x}^{\overline{b}}}(f)\right) = \operatorname{TM}\left(\overline{x}^{\overline{c}} \cdot \partial_{\overline{x}^{\overline{b}}}(\operatorname{TM}(f))\right) .$$

In particular,

$$\operatorname{TM}\left(\overline{x}^{\overline{c}} \cdot \partial_{\overline{x}^{\overline{b}}}(f)\right) = \operatorname{TM}\left(\overline{x}^{\overline{c}} \cdot \partial_{\overline{x}^{\overline{b}}}(\operatorname{TM}(f))\right) .$$

Proof: By Lemma 3.6, we see that

$$\operatorname{TM}(\overline{x}^{\overline{c}} \cdot h \cdot \partial_{\overline{x}^{\overline{b}}}(f)) = \operatorname{TM}(\overline{x}^{\overline{c}}) \operatorname{TM}(h) \operatorname{TM}\left(\partial_{\overline{x}^{\overline{b}}}(f)\right)$$

as $\mathrm{TM}(h) = 1$,

$$= \mathrm{TM}(\overline{x}^{\overline{c}})\,\mathrm{TM}\left(\partial_{\overline{x}^{\overline{b}}}(f)\right)$$

applying Lemma 3.8 as $\partial_{\overline{x}^{\overline{b}}}(\mathrm{TM}(f)) \neq 0$,

$$= \mathrm{TM}(\overline{x}^{\overline{c}})\,\mathrm{TM}\left(\partial_{\overline{x}^{\overline{b}}}(\mathrm{TM}(f))\right)$$

applying Lemma 3.6 in reverse,

$$= \mathrm{TM}\left(\overline{x}^{\overline{c}} \cdot \partial_{\overline{x}^{\overline{b}}}(\mathrm{TM}(f))\right) .$$

The second part of the claim follows from taking $h = 1$. □

3.4 Hitting Sets for Polynomials with Small-Support Monomials

In this section, we give results showing that certain structural results are sufficient to derive explicit hitting sets. In particular, we review the notion of a polynomial having *small-support* monomial, and state constructions of hitting sets known for such polynomials. This structural condition was introduced in the works of Shpilka-Volkovich [SV09] and Agrawal-Saha-Saxena [ASS13], and further developments were made in Forbes-Shpilka-Saptharishi [FSS14]. We begin with the definition.

Definition 3.13. *The set of n-**variate degree-**$(\leq d)$ **polynomials with a support-**$(\leq \ell)$ **monomial** in $\mathbb{F}[x_1, \ldots, x_n]$ is the set of polynomials $f \in \mathbb{F}[\overline{x}]$ of degree $\leq d$ which (when non-zero) have some support-$(\leq \ell)$ monomial $\overline{x}^{\overline{a}}$ with a non-zero coefficient. That is, there is a $\overline{a}$ such that* $\mathrm{Coeff}_{\overline{x}^{\overline{a}}}(f) \neq 0$ *and* $\|\overline{a}\|_0 \leq \ell$. ◊

Note that the small-support monomial that is guaranteed to have a non-zero coefficient can vary between polynomials in this class. As an example of the definition, $x_1 \cdots x_n + 1$ has a support-ℓ monomial for $\ell = 0$, but $x_1 \cdots x_n$ requires $\ell \geq n$.

Showing that a polynomial with a small-support monomial can be thought of as a structural result showing that this polynomial is in some sense simple. This simplicity allows the following lemma, which shows that non-zero polynomials with small-support monomials have some non-root with small-support.

Lemma 3.14. *Let $S \subseteq \mathbb{F}^n$ have size $|S| = d + 1$. Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ be a n-variate degree-$(\leq d)$ polynomial with somewhere-support-$(\leq \ell)$. Then if $f \neq 0$ then there is an $\overline{\alpha} \in S^n$ with $\|\overline{\alpha}\|_0 \leq \ell$ such that $f(\overline{\alpha}) \neq 0$.* □

When $\ell = n$ the above result is just usual interpolation, but the above gains for $\ell \ll n$ by zero-ing out variables which do not contribute to the small-support monomial of f. By taking all such sparse vectors we then get a hitting set for the class of polynomials with small-support monomials.

Corollary 3.15 (Shpilka-Volkovich [SV09])**.** *There is a* $\mathsf{poly}(n, d, \ell)$*-explicit hitting set $\mathcal{H}^{ss}_{n,d,\ell}$ for the class of n-variate degree-$(\leq d)$ polynomials with a support-$(\leq \ell)$ monomial, with size $|\mathcal{H}^{ss}_{n,d,\ell}| = (n(d+1))^\ell$,* □

One can roughly think of this hitting set as a ball around $\overline{0}$, where the ball has radius ℓ in the hamming distance. This perspective is useful as it naturally motivates centering this ball around other points (such as $\overline{1}$), and as well shall see sometimes picking a different center leads to new results.

Note that this result "beats" the probabilistic method in that a naïve union bound cannot even furnish the *existence* of a hitting set with the above parameters. This is because *most* polynomials have small-support monomials and thus there are too many such polynomials to apply a union bound over. Indeed, such union-bound arguments typically yield *interpolation sets* (where a polynomial f is *entirely determined* by its evaluations on such a set), and because the above class of polynomials is large it cannot have small interpolation sets. While intuitively interpolation sets and hitting sets are similar, this is only true for classes of polynomials closed under subtraction, which the class of polynomials with small-support monomials is not.

The above hitting set has two parts: identifying the support $S \subseteq [n]$ of the small-support monomial, and then zeroing-out all variables outside S and brute-forcing the polynomial on the variables inside S. Forbes, Shpilka, Saptharishi [FSS14] observed that one can do better if the original polynomial has a commutative roABP structure. That is, consider the second step of this argument. As roABPs are closed under partial substitutions (Lemma 3.17), zeroing-out the variables retains this roABP structure. As hitting sets for roABPs such as Forbes-Shpilka [FS13b] have size $\mathsf{poly}(nwd)^{\mathcal{O}(\lg n)}$ for n-variate, width-w, degree-d roABPs, applying these hitting sets for the zeroed-out polynomial results in a complexity of $\mathsf{poly}(|S|wd)^{\mathcal{O}(\lg |S|)}$ instead of the brute-force $d^{|S|}$. Now consider the step of guessing S. Instead of guessing S directly, one can use hashing to push the complexity of this guess into the roABP itself so that the hitting set for the roABP derandomizes the guess. Combining these ideas, they obtained the following theorem.

Theorem 3.16 (Forbes, Shpilka, Saptharishi [FSS14])**.** *Let $|\mathbb{F}| \ge \mathsf{poly}(n, d, w, \ell)$. There is a $\mathsf{poly}(n, d, w, \ell)$-explicit hitting set of size $\mathsf{poly}(n, d, w, \ell)^{\mathcal{O}(\lg \ell)}$ for the class of polynomials $f(\overline{x}) \in \mathbb{F}[\overline{x}]$ computed by a width-w, individual degrees $\le d$ commutative roABPs which have a support-$(\le \ell)$ monomial.* □

We have the following closure result for roABPs which will be useful below.

Lemma 3.17. *Let $f, g \in \mathbb{F}[x_1, \ldots, x_n]$ be computable by width-r and width-w roABPs respectively, each of individual degree $\le d$, each in the variable order $\pi : [n] \to [n]$.*

1. ***Homogenization:*** *f is also computable by a width-r roABP of individual degree $\le \deg f$.*

2. ***Partial Substitution:*** *For any $S \subseteq [n]$ and $\overline{\alpha} \in \mathbb{F}^{[n]\setminus S}$ define the substitution map $\varphi : \mathbb{F}[\overline{x}] \to \mathbb{F}[\overline{x}|_S]$ by the substitutions $x_i \mapsto x_i$ for $i \in S$ and $x_i \mapsto \alpha_i$ for $i \notin S$. Then $\varphi(f(\overline{x}))$ is computed by a width-r roABP in the variable order π' on the variables $\overline{x}|_S$ induced by π.*

3. ***Addition:*** *The addition $f(\overline{x}) + g(\overline{x})$ is computable by a width-$(w + r)$ in the variable order π.*

4. ***Multiplication:*** *The multiplication $f(\overline{x})g(\overline{x})$ is computable by a width-wr roABP in the variable order π.*

5. ***Monomials:*** *The monomial $\overline{x}^{\overline{a}}$ is computable by a width-1 roABP in any order π.*

6. ***Translation:*** *For any $\overline{\alpha} \in \mathbb{F}^n$, $f(\overline{x} + \overline{\alpha})$ is computable by a width-r roABP in the variable order π.* □

Applying the above closure result, along with the simulation of $\sum \bigwedge \sum$ formulas by commutative roABPs (Lemma 1.3), we have the following additional simulation results.

Lemma 3.18. *The following classes of n-variate, degree-$(\le d)$ polynomials are computable by $\mathsf{poly}(n, d, s)$-size commutative roABPs.*

1. *s-sparse polynomials.*

2. *s-sparse polynomials under any translation.*

3. *$\sum \mathrm{m} \bigwedge \sum$ formulas of size s.*

4. *$\sum \mathrm{m} \bigwedge \sum$ formulas of size s under any translation.* □

Thus, if one can show that the above classes of computation must compute a $\mathcal{O}(\lg s)$ monomial (perhaps in at least one of $\mathsf{poly}(s)$-many translated bases) then the above hitting set of Forbes, Shpilka, Saptharishi [FSS14] can be applied to obtain a $\mathsf{poly}(s)^{\mathcal{O}(\lg \lg s)}$-size hitting set. Note that Agrawal, Saha, and Saxena [ASS13] (as made explicit by Forbes, Shpilka, Saptharishi [FSS14]) showed that one can induce such monomials using a "$\mathcal{O}(\lg s)$-wise independent map". However, this would require trying $\mathsf{poly}(s)^{\Theta(\lg s)}$-many possible translations which would make the improvement via the results of Forbes, Shpilka, Saptharishi [FSS14] negligible.

4 Small-Support Monomials for $\sum \bigwedge \sum \prod^t$ Formulas

In this section we first review the definitions and properties of the shifted partial derivative measure, as given by Kayal [Kay12] and Gupta, Kamath, Kayal, and Saptharishi [GKKS14]. We first focus on the definition of the measure and upper bounds on the measure for composed functions and $\sum \bigwedge \sum \prod^t$ formulas in particular. We then extend Kayal's [Kay12, GKKS14] lower bound on this measure to all monomials. Finally, we show that the shifted partial derivative measure of a polynomial f is at least that of its trailing monomial $\mathrm{TM}(f)$. Combining this with Kayal's [Kay12, GKKS14] logic we obtain a bound on the number of variables involved in the trailing monomials of $\sum \bigwedge \sum \prod^t$ formulas.

4.1 The Measure and Upper Bounds for Computation

We now define the shift and derivative operators that comprise the shifted partial derivative method.

Definition 4.1. *The **order-k (Hasse) derivative** operator $\boldsymbol{\partial}_{\overline{x}^{\le k}}$ is the map $\boldsymbol{\partial}_{\overline{x}^{\le k}} : \mathbb{F}[\overline{x}] \to 2^{\mathbb{F}[\overline{x}]}$ defined by*

$$\boldsymbol{\partial}_{\overline{x}^{\le k}}(f) := \left\{ \partial_{\overline{x}^{\overline{b}}}(f) \right\}_{\deg \overline{x}^{\overline{b}} \le k} .$$

Define $\boldsymbol{\partial}_{\overline{x}^{<\infty}}(f)$ to be the union of $\boldsymbol{\partial}_{\overline{x}^{\le k}}(f)$ over all $k \in \mathbb{N}$.

*The **shift** operator $\overline{\boldsymbol{x}}^{\le \ell}$ is the map $\overline{\boldsymbol{x}}^{\le \ell} : \mathbb{F}[\overline{x}] \to 2^{\mathbb{F}[\overline{x}]}$ defined by*

$$\overline{\boldsymbol{x}}^{\le \ell}(f) := \left\{ \overline{x}^{\overline{c}} \cdot f \right\}_{\deg \overline{x}^{\overline{c}} \le \ell} . \qquad \Diamond$$

We will use bold to emphasize that these operators are collections of set-valued. Note that these operators can be composed in a natural way, so that $\overline{\boldsymbol{x}}^{\le \ell} \boldsymbol{\partial}_{\overline{x}^{\le k}}(f)$ denotes $\{\overline{x}^{\overline{c}} \partial_{\overline{x}^{\overline{b}}}(f)\}_{\deg \overline{x}^{\overline{b}} \le \ell, \deg \overline{x}^{\overline{c}} \le k}$.

We have the following trivial bounds for the dimension of the spaces these operators produce.

Lemma 4.2. *Let $f \in \mathbb{F}[x_1, \ldots, x_n]$. Then*

1. $\dim \boldsymbol{\partial}_{\overline{x}^{\le k}} f \le |\boldsymbol{\partial}_{\overline{x}^{\le k}} f| = \binom{n+k}{k}$.

2. $\dim \overline{\boldsymbol{x}}^{\le \ell} f \le |\overline{\boldsymbol{x}}^{\le \ell} f| = \binom{n+\ell}{\ell}$.

Proof: Both statements follow by bounding the dimension by the number of spanning elements, and then counting these elements by counting the number of monomials in the n variables $\overline{x}$ of the appropriate degree. □

Note that the maps $f \mapsto \partial_{\overline{x}^{\overline{b}}} f$ and $f \mapsto \overline{x}^{\overline{c}} f$ are linear maps over the $\mathbb{F}$-vector space $\mathbb{F}[\overline{x}]$. As such, as is well known, the dimension of these operators is sub-additive. In particular, we have the following lemma.

Lemma 4.3. *Let $\{\varphi_i : \mathbb{F}[\overline{x}] \to \mathbb{F}[\overline{x}]\}_i$ be a collection of linear operators. Let $\boldsymbol{\varphi} : \mathbb{F}[\overline{x}] \to 2^{\mathbb{F}[\overline{x}]}$ be defined by $\boldsymbol{\varphi}(f) := \{\varphi_i(f)\}_i$.*

Then for any polynomials $f, g \in \mathbb{F}[\overline{x}]$, $\operatorname{span} \boldsymbol{\varphi}(f+g) \subseteq \operatorname{span}(\boldsymbol{\varphi}(f) \cup \boldsymbol{\varphi}(g))$. In particular, $\dim(\operatorname{span} \boldsymbol{\varphi}(\sum_i \alpha_i f_i)) \leq \sum_i \dim \operatorname{span} \boldsymbol{\varphi}(f_i)$. □

We now quote results of Gupta, Kamath, Kayal, and Saptharishi [GKKS14] describing how the shifted derivative measure interacts with function composition. Note that their result is actually for shifted partial derivatives, while Forbes [For14, Corollary C.2.10] extended the result to shifted Hasse derivatives.

Corollary 4.4 (Gupta-Kamath-Kayal-Saptharishi [GKKS14])**.** *Let $f \in \mathbb{F}[y_1, \ldots, y_m]$ and $t \geq 1$. Suppose $\overline{g} \in (\mathbb{F}[x_1, \ldots, x_n])^m$, where each g_i is of degree $\leq t$. Then for any,$\ell, k \geq 0$,*

$$\boldsymbol{\overline{x}}^{\leq \ell} \boldsymbol{\partial}_{\overline{x}^{\leq k}} (f \circ \overline{g}) \subseteq \operatorname{span} \boldsymbol{\overline{x}}^{\leq (t-1)k+\ell} \left(\left[\boldsymbol{\partial}_{\overline{y}^{\leq k}}(f) \right] (\overline{g}) \right) .$$

In particular,

$$\begin{aligned} \dim \boldsymbol{\overline{x}}^{\leq \ell} \boldsymbol{\partial}_{\overline{x}^{\leq k}} (f \circ \overline{g}) &\leq \binom{n + (t-1)k + \ell}{(t-1)k + \ell} \cdot \dim \left[\boldsymbol{\partial}_{\overline{y}^{\leq k}}(f) \right] (\overline{g}) \\ &\leq \binom{n + (t-1)k + \ell}{(t-1)k + \ell} \cdot \binom{m+k}{k} . \end{aligned}$$ □

Note that the third part of the statement follows from the second along with Lemma 4.2.

We now arrive at the bound on the dimension of a shifted derivative space of $\sum \bigwedge \sum \prod^t$ formula due to Kayal [Kay12] and Gupta-Kamath-Kayal-Saptharishi [GKKS14] , which we state a slight generalization to $m > 1$ as this will be used for our divisibility algorithms. Specifically, we combine the upper bound of Corollary 4.4 with subadditivity (Lemma 4.3).

Corollary 4.5 (Kayal [Kay12], Gupta-Kamath-Kayal-Saptharishi [GKKS14])**.** *Let $t \geq 1$. Consider $f(\overline{x}) = \sum_{i=1}^{s} f_i(g_{i,1}(\overline{x}), \ldots, g_{i,m}(\overline{x}))$ where $\deg g_{i,j} \leq t$. Then*

$$\dim \boldsymbol{\overline{x}}^{\leq \ell} \boldsymbol{\partial}_{\overline{x}^{\leq k}} f \leq s \binom{k+m}{m} \binom{n + (t-1)k + \ell}{(t-1)k + \ell} .$$ □

Remark 4.6. In eventual estimates that we will use (Lemma A.3) we will take $k = \Theta({}^{n}\!/_{t})$ and $\ell = (t-1)(n + (t-1)k)$. As such, using that $({}^{a}\!/_{b})^b \leq \binom{a}{b}$, one can see that $\binom{n+(t-1)k+\ell}{(t-1)k+\ell} = \binom{n+(t-1)k+\ell}{n} \geq \left(\frac{t(n+(t-1)k)}{n}\right)^n \geq ({}^{t(n+0)}\!/_{n})^n \geq t^n$. Similarly, using $\binom{a}{b} \leq ({}^{\mathrm{e}a}\!/_{b})^b$, one sees that $\binom{n+(t-1)k+\ell}{n} \leq \mathrm{e}^n \left(\frac{t(n+(t-1)k)}{n}\right)^n \leq ({}^{t \cdot \mathcal{O}(n)}\!/_{n})^n \leq O(t)^n$. Thus, for $t \geq 2$ the above upper on the measure is exponential, but is only singly exponential in the number n of variables. ◊

Lemma 4.7. *Let $\overline{x}^{\overline{a}} \in \mathbb{F}[x_1, \ldots, x_n]$ be a monomial. Then $\boldsymbol{\overline{x}}^{\leq \ell} \boldsymbol{\partial}_{\overline{x}^{\leq k}} (\overline{x}^{\overline{a}})$ is a set of terms (or zero), and thus*

$$\dim \operatorname{span} \left(\boldsymbol{\overline{x}}^{\leq \ell} \boldsymbol{\partial}_{\overline{x}^{\leq k}} (\overline{x}^{\overline{a}}) \right) = \left| \mathrm{TM} \left(\boldsymbol{\overline{x}}^{\leq \ell} \boldsymbol{\partial}_{\overline{x}^{\leq k}} (\overline{x}^{\overline{a}}) \right) \right| .$$

Proof: Note that $\overline{x}^{\overline{c}} \partial_{\overline{x}^{\overline{b}}} \overline{x}^{\overline{a}} = \overline{x}^{\overline{c}} \cdot \binom{\overline{a}}{\overline{b}} \overline{x}^{\overline{a}-\overline{b}} = \binom{\overline{a}}{\overline{b}} \overline{x}^{\overline{a}-\overline{b}+\overline{c}}$, which is a term (or zero). The claim then follows from appealing to Corollary 3.11. □

4.2 Relating the Measure to Trailing Monomials

We now give the first main insight of this paper, which shows that the shifted partial derivative measure of a polynomial f is lower bounded by the measure of the trailing monomial of f. Forbes and Shpilka [FS13a] observed this for the *partial derivative measure* and the below is a generalization to shifted partial derivatives.

Lemma 4.8. *Let $f \in \mathbb{F}[\overline{x}]$. Then for any $\ell, k \geq 0$,*

$$\begin{aligned} \dim \operatorname{span} \boldsymbol{\overline{x}}^{\leq \ell} \boldsymbol{\partial}_{\overline{x}^{\leq k}}(f) &\geq \left| \mathrm{TM}\left(\boldsymbol{\overline{x}}^{\leq \ell} \boldsymbol{\partial}_{\overline{x}^{\leq k}}(f) \right) \right| && (4.9) \\ &\geq \left| \mathrm{TM}\left(\boldsymbol{\overline{x}}^{\leq \ell} \boldsymbol{\partial}_{\overline{x}^{\leq k}}\left(\mathrm{TM}(f)\right) \right) \right| && (4.10) \\ &= \dim \operatorname{span} \boldsymbol{\overline{x}}^{\leq \ell} \boldsymbol{\partial}_{\overline{x}^{\leq k}}(\mathrm{TM}(f))\ . && (4.11) \end{aligned}$$

Proof: Denote $S := \boldsymbol{\overline{x}}^{\leq \ell} \boldsymbol{\partial}_{\overline{x}^{\leq k}}(f)$ and $T := \boldsymbol{\overline{x}}^{\leq \ell} \boldsymbol{\partial}_{\overline{x}^{\leq k}}(\mathrm{TM}(f))$.

(4.9): This is Lemma 3.9 applied to S.

(4.10): When $\overline{x}^{\overline{c}} \partial_{\overline{x}^{\overline{b}}}(\mathrm{TM}(f)) \in T$ is zero there is no trailing monomial. When $\overline{x}^{\overline{c}} \partial_{\overline{x}^{\overline{b}}}(\mathrm{TM}(f)) \in T$ is non-zero we see that

$$\mathrm{TM}\left(\overline{x}^{\overline{c}} \partial_{\overline{x}^{\overline{b}}}(f) \right) = \mathrm{TM}\left(\overline{x}^{\overline{c}} \partial_{\overline{x}^{\overline{b}}}(\mathrm{TM}(f)) \right)$$

by Corollary 3.12. Thus, $\mathrm{TM}(S) \supseteq \mathrm{TM}(T)$ yielding the claim.

(4.11): This is because T is a set of terms (Lemma 4.7). □

While the above idea sufficed for the partial derivative measure in Forbes-Shpilka [FS13a], the shifted partial derivative measure is more difficult to handle. That is, for $t \geq 2$ the measure is exponentially large even when the $\sum \bigwedge \sum \prod^t$ formula has small size (Remark 4.6), which does not occur when studying the partial derivative measure of $\sum \bigwedge \sum \prod^1$ formulas. As discussed above, this largeness of the measure makes it seem difficult to develop PIT algorithms for $\sum \bigwedge \sum \prod^t$ when $t \geq 2$.

However, another insight of this paper is that, as shown in Remark 4.6, the shifted partial measure is only singly exponential in the number n of variables. Thus, if we can reduce the number of variables to logarithmic then the measure will become polynomially large. With this motivation, the next lemma shows that zeroing out some variables preserves the trailing monomial. In zeroing-out we will change from x-variables to y-variables to highlight the distinction, as this will be important in the usage in Proposition 4.18.

Lemma 4.12. *Let $f \in \mathbb{F}[x_1, \ldots, x_n]$. Let $S \subseteq [n]$ and let $\pi_S : \mathbb{F}[\overline{x}] \to \mathbb{F}[\overline{y}]$ be the substitution map that zeroes out any variable x_i outside S, that is $x_i \mapsto y_i$ for $i \in S$ and $x_i \mapsto 0$ for $i \notin S$. Identify $\mathbb{F}[\overline{y}]$ with the subring $\mathbb{F}[\overline{x}|_S]$ of $\mathbb{F}[\overline{x}]$ so that the monomial order '$\prec$' on $\mathbb{F}[\overline{x}]$ induces a monomial order on $\mathbb{F}[\overline{y}]$.*

Let $\overline{x}^{\overline{a}}$ be the trailing monomial of f, $\overline{x}^{\overline{a}} := \mathrm{TM}(f)$. Suppose that S contains the variables in $\overline{x}^{\overline{a}}$, so $S \supseteq \mathrm{Supp}(\overline{a})$. Then applying π_S commutes with taking the trailing monomial, that is, $\mathrm{TM}(\pi_S(f)) = \pi_S(\mathrm{TM}(f)) = \overline{y}^{\overline{a}}$.

Proof: By the hypothesis, we see that $\mathrm{TM}(f)$ is preserved by π_S, so that $\pi_S(\overline{x}^{\overline{a}}) = \overline{y}^{\overline{a}}$. Now consider the expansion of f,

$$f(\overline{x}) = \sum_{\overline{b}} \beta_{\overline{b}} \overline{x}^{\overline{b}}\ .$$

Then as π_S is a linear map,

$$\pi_S(f) = f(\pi_S(\overline{x})) = \sum_{\overline{b}} \beta_{\overline{b}} \pi_S(\overline{x}^{\overline{b}}) = \sum_{\mathrm{Supp}(\overline{b}) \subseteq S} \beta_{\overline{b}} \overline{y}^{\overline{b}}\ .$$

Viewing $\mathbb{F}[\overline{y}]$ as a subring of $\mathbb{F}[\overline{x}]$, those monomials in $\pi_S(f)$ also appear in f, so that $\operatorname{Supp}(\pi_S(f)) \subseteq \operatorname{Supp}(f)$. As $\overline{y}^{\overline{a}} \equiv \overline{x}^{\overline{a}}$ is in both sets and is a minimum under '$\prec$' in $\operatorname{Supp}(f)$ it follows that $\overline{y}^{\overline{a}}$ is also minimum in $\operatorname{Supp}(\pi_S(f))$. □

4.3 Lower Bounds for the Measure of Monomials

We now give lower bounds on the dimension of shifted partials of arbitrary monomials via a straightforward generalization of the argument of Kayal [Kay12], who did this for the multilinear monomial $\overline{x}^{\overline{1}}$. The lower bound will actually be for the number of distinct trailing monomials, which by Corollary 3.11 will equal the dimension of the span of the shifted partials.

Lemma 4.13 (implicit in Kayal [Kay12])**.** *Let $\overline{x}^{\overline{a}} \in \mathbb{F}[x_1, \ldots, x_n]$ be a monomial and suppose that $\operatorname{char}(\mathbb{F}) > \operatorname{ideg} \overline{x}^{\overline{a}}$. Then*

$$\dim \operatorname{span}\left(\boldsymbol{\overline{x}}^{\leq \ell} \boldsymbol{\partial}_{\overline{x}^{\leq k}}(\overline{x}^{\overline{a}})\right) = \left|\operatorname{TM}\left(\boldsymbol{\overline{x}}^{\leq \ell} \boldsymbol{\partial}_{\overline{x}^{\leq k}}(\overline{x}^{\overline{a}})\right)\right| \geq \binom{\|\overline{a}\|_0}{k}\binom{n-k+\ell}{\ell}.$$

Proof: $\underline{\dim = |\operatorname{TM}|}$: This is because the shifted partials are terms (or zero) (Lemma 4.7).

$\underline{\text{lower bound on } |\operatorname{TM}|}$: Let $S = \operatorname{Supp}(\overline{a})$ be the support of $\overline{a}$, which we identify with those variables appearing in $\overline{x}^{\overline{a}}$. We now identify a set of shifted partial derivatives $\overline{x}^{\overline{c}} \partial_{\overline{x}^{\overline{b}}}$ that produce terms with distinct monomials, and we use the same set as Kayal [Kay12]. That is, we first differentiate with respect to a multilinear monomial $\overline{x}^{\overline{b}}$ of degree k, only differentiating variables in the support of $\overline{x}^{\overline{a}}$ (so we do not annihilate $\overline{x}^{\overline{a}}$). Then we multiply by a monomial $\overline{x}^{\overline{c}}$ in variables we have not differentiated. More formally, we consider the pairs of exponents

$$E := \{(\overline{c}, \overline{b}) \mid \overline{0} \leq \overline{b} \leq \overline{1}, \deg \overline{x}^{\overline{b}} = k, \operatorname{Supp}(\overline{b}) \subseteq S,\ \operatorname{Supp}(\overline{c}) \subseteq [n] \setminus \operatorname{Supp}(\overline{b}), \deg \overline{x}^{\overline{c}} \leq \ell\}.$$

Notice that while the shifted partial derivative measure considers the set of differential operators $\{\overline{x}^{\overline{c}} \partial_{\overline{x}^{\overline{b}}}\}_{\overline{b},\overline{c}}$ where the set of exponents $(\overline{c}, \overline{b})$ is a *product set*, the set of exponents E is *not* such a product set. We now show that the shifted partials arising from E yield distinct monomials, and that there are many such monomials.

Subclaim 4.14. *The terms in $D := \left\{\overline{x}^{\overline{c}} \partial_{\overline{x}^{\overline{b}}} \overline{x}^{\overline{a}}\right\}_{(\overline{c},\overline{b}) \in E}$ have distinct underlying monomials, so that $|\operatorname{TM}(D)| = |E|$.*

Sub-Proof: As above, $\overline{x}^{\overline{c}} \partial_{\overline{x}^{\overline{b}}} \overline{x}^{\overline{a}} = \binom{\overline{a}}{\overline{b}} \overline{x}^{\overline{a}-\overline{b}+\overline{c}}$. Note that $\binom{\overline{a}}{\overline{b}} \neq 0$. That is, as $0 \leq b_i \leq \min\{1, a_i\}$, $\binom{a_i}{b_i} \in \{1, a_i\}$. As $\operatorname{char}(\mathbb{F}) > \operatorname{ideg} \overline{x}^{\overline{a}} \geq a_i$ we have that $\binom{a_i}{b_i} \neq 0$. Thus $\binom{\overline{a}}{\overline{b}} = \prod_{i \in [n]} \binom{a_i}{b_i} \neq 0$. Thus, the claim amounts to showing that the vectors $\overline{a} - \overline{b} + \overline{c}$ are distinct, in particular that we can recover $\overline{b}$ and $\overline{c}$ from $\overline{d} := \overline{a} - \overline{b} + \overline{c}$ (as $\overline{a}$ is fixed).

Consider $\overline{a} - \overline{d} = \overline{c} - \overline{b}$. Note that $\overline{b}$ and $\overline{c}$ have disjoint supports by definition of $\overline{c}$. Thus, it follows that there is no cancellation in the difference $\overline{c} - \overline{b}$ so that $\overline{c}$ is the positive part of $\overline{c} - \overline{b}$ and $\overline{b}$ the negative part. That is, for all $i \in [n]$, $c_i = \max\{(\overline{a} - \overline{d})_i, 0\}$ and $b_i = -\min\{(\overline{a} - \overline{d})_i, 0\}$. Thus, $\overline{b}$ and $\overline{c}$ are uniquely determined by $\overline{a} - (\overline{a} - \overline{b} + \overline{c})$ and thus are determined by $\overline{a} - \overline{b} + \overline{c}$. □

Subclaim 4.15. $|E| = \binom{\|\overline{a}\|_0}{k}\binom{n-k+\ell}{\ell}$.

Sub-Proof: The exponent $\overline{b}$ is chosen as k variables from the support S of $\overline{a}$, which has size $\|\overline{a}\|_0$, so that there are $\binom{\|\overline{a}\|_0}{k}$ such $\overline{b}$. Once $\overline{b}$ is chosen, $\overline{c}$ is chosen as a degree $\leq \ell$ monomial in those $n - k$ variables not in the support of $\overline{b}$, of which there are $\binom{n-k+\ell}{\ell}$. □

Putting the above together, we have formed a subset D of $\boldsymbol{x}^{\leq \ell}\boldsymbol{\partial}_{\overline{x}^{\leq k}}(\overline{x}^{\overline{a}})$, where D is a set of terms with distinct underlying monomials (and thus distinct trailing monomials). Further, the number of such monomials is as desired. □

Remark 4.16. The above proof requires that the field $\mathbb{F}$ has characteristic larger than the individual degree of the monomial $\overline{x}^{\overline{a}}$. For the multilinear monomial $\overline{x}^{\overline{1}}$ this is no restriction as all fields have characteristic ≥ 2 (if we think of characteristic zero fields as having infinite characteristic). In general, this restrict is needed, as in characteristic p the monomial x^p only has two non-zero Hasse derivatives $\partial_1 x^p = x^p$ and $\partial_{x^p} = 1$.

One can remove this restriction if one chooses the derivatives $\overline{b}$ from the set E to have $b_i \in \{0, a_i\}$, as then $\binom{a_i}{b_i} = 1$ so none of the derivatives annihilate $\overline{x}^{\overline{a}}$. While this will yield the above lower bound in small characteristic, this set of shifted partial derivatives will yield a much worse *upper* bound, and thus most of the results of this paper are restricted to polynomially large characteristic. ◊

Remark 4.17. As mentioned, Kayal [Kay12] used the above for $\overline{x}^{\overline{a}}$ where $\overline{a} = \overline{1}$, for which $\|\overline{a}\|_1 = n$. However, we separate these two parameters in the above to highlight the dependence of shifted partials on the number of variables. That is, the $\boldsymbol{x}^{\leq k}$ operator will multiply by monomials in *all* variables in $\overline{x}$. However, we will need to understand this measure on monomials $\overline{x}^{\overline{a}}$ with $\|\overline{a}\|_1 \ll n$. In such cases, the variables in $\overline{x} \setminus \mathrm{Supp}(\overline{x}^{\overline{a}})$ are non-essential. While multiplying by shifts in these variables will increase the resulting lower bound of the measure of $\overline{x}^{\overline{a}}$, it will also dramatically increase the measure of the computation we are trying to understand. Specifically, as mentioned in Remark 4.6, the bound on the shifted partial measure of $\sum\bigwedge\sum\prod^t$ computation grows as $\Theta(t)^n$ and thus extra variables are quite costly.

Put another way, computation of a polynomial $f(\overline{x})$ within the ring $\mathbb{F}[\overline{x}, \overline{y}]$ can (for reasonable models of computation) always be translated to computation of $f(\overline{x})$ within the ring $\mathbb{F}[\overline{x}]$ by setting $\overline{y}$ to zero. As such, introducing new dummy variables $\overline{y}$ to the computation of $f(\overline{x})$ within $\mathbb{F}[\overline{x}]$ is never strictly necessary to obtain good lower bounds. As such, we will eliminate such variables. ◊

4.4 Trailing Monomial Bounds for Computation

We now put together the above results to show that small $\sum\bigwedge\sum\prod^t$ computation must compute polynomials with trailing monomials involving few variables. That is, we consider a small $\sum\bigwedge\sum\prod^t$ formula with a trailing monomial $\overline{x}^{\overline{a}}$. We eliminate variables not in $\overline{x}^{\overline{a}}$, noting that this does not increase the $\sum\bigwedge\sum\prod^t$ complexity. We then see that $\overline{x}^{\overline{a}}$ is of "full-support" in $\overline{x}$ so that we are essentially in the setting of Kayal [Kay12]. If the $\sum\bigwedge\sum\prod^t$ formula only computed $\overline{x}^{\overline{a}}$ we would be directly done by scaling down his lower bound. However, by robustness we can pass from the entire polynomial to $\overline{x}^{\overline{a}}$ directly, giving the result.

Proposition 4.18. *Let $f(\overline{x}) \in \mathbb{F}[x_1, \ldots, x_n]$ be a non-zero polynomial of the form*

$$f(\overline{x}) = \sum_{i=1}^{s} f_i(g_{i,1}(\overline{x}), \ldots, g_{i,m}(\overline{x})) \ ,$$

where $\deg g_{i,j} \leq t$. Let $\overline{x}^{\overline{a}}$ be the trailing monomial of f, $\overline{x}^{\overline{a}} := \mathrm{TM}(f)$. If $\mathrm{char}(\mathbb{F}) > \mathrm{ideg}\,\overline{x}^{\overline{a}}$, then

$$\|\overline{a}\|_0 \leq 2\mathrm{e}^3 t(\ln s + m\ln(2m) + 1) \ .$$

In particular, for $m = O(1)$, $\|\overline{a}\|_0 \leq \mathcal{O}(t \ln s)$.

Proof: Let $m = \|\overline{a}\|_0$, and let $S \subseteq [n]$ be the support of $\overline{a}$, $S := \mathrm{Supp}(\overline{a})$. We now assume the setup of Lemma 4.12 to define the substitution homomorphism $\pi_S : \mathbb{F}[\overline{x}] \to \mathbb{F}[\overline{y}]$ which zeroes-out variables outside of S and produces an isomorphism $\pi_S : \mathbb{F}[\overline{x}|_S] \xrightarrow{\sim} \mathbb{F}[\overline{y}]$. We make the notational distinction between $\overline{x}|_S$ and $\overline{y}$ to make clear that when we consider shifted partial derivatives we only multiply by shifts in the variables $\overline{y} \equiv \overline{x}|_S$ as opposed to shifting by all of the variables in $\overline{x}$.

Now consider the computation of f under the zeroing out of all variables outside S, so that

$$\pi_S(f) = f(\pi_S(\overline{x})) = \sum_{i=1}^{s} f_i(g_{i,1}(\pi_S(\overline{x})), \ldots, g_{i,m}(\pi_S(\overline{x}))) \, .$$

Note that $\pi_S(\overline{x}^{\overline{a}}) = \overline{y}^{\overline{a}}$ so $\pi_S(f) \neq 0$ (where we slightly abuse notation, as $\overline{a} \in \mathbb{N}^n$ in the expression '$\overline{x}^{\overline{a}}$' but we treat $\overline{a} \in \mathbb{N}^{\|\overline{a}\|_0}$ in the expression '$\overline{y}^{\overline{a}}$').

Note that π_S has not increased the complexity of f, for example $\deg_{\overline{y}} \pi_S(g_{i,j}) \leq \deg_{\overline{x}} g_{i,j}$. Thus, we can apply the upper bound on the shifted partials of $\sum\bigwedge\sum\prod^t$ (Corollary 4.5) to $\pi_S(f) \in \mathbb{F}[\overline{y}]$, using that there are only $\|\overline{a}\|_0$ variables, to obtain

$$\dim \boldsymbol{\overline{y}}^{\leq \ell} \boldsymbol{\partial}_{\overline{y}^{\leq k}}(\pi_S(f)) \leq s \binom{k+m}{m} \binom{\|\overline{a}\|_0 + (t-1)k + \ell}{(t-1)k + \ell} \, .$$

We now turn to the lower bound. From Lemma 4.12, we see that $\overline{y}^{\overline{a}}$ is also the trailing monomial of $\pi_S(f) \in \mathbb{F}[\overline{y}]$. We now lower bound the dimension of the shifted partial derivatives of $\pi_S(f) \in \mathbb{F}[\overline{y}]$, using that $\overline{y}$ only has $\|\overline{a}\|_0$ variables.

$$\begin{aligned} \dim \operatorname{span} \boldsymbol{\overline{y}}^{\leq \ell} \boldsymbol{\partial}_{\overline{y}^{\leq k}}(\pi_S(f)) &\geq \dim \operatorname{span} \boldsymbol{\overline{y}}^{\leq \ell} \boldsymbol{\partial}_{\overline{y}^{\leq k}}(\mathrm{TM}(\pi_S(f))) \\ &= \dim \operatorname{span} \boldsymbol{\overline{y}}^{\leq \ell} \boldsymbol{\partial}_{\overline{y}^{\leq k}}(\overline{y}^{\overline{a}}) \\ &\geq \binom{\|\overline{a}\|_0}{k} \binom{\|\overline{a}\|_0 - k + \ell}{\ell} \, , \end{aligned}$$

where we have respectively applied Lemma 4.8, Lemma 4.12, and (as the characteristic is large enough) Lemma 4.13.

Putting the above together, we have that for any $k, \ell \in \mathbb{N}$ that

$$s \geq \frac{1}{\binom{k+m}{m}} \frac{\binom{\|\overline{a}\|_0}{k}\binom{\|\overline{a}\|_0 - k + \ell}{\ell}}{\binom{\|\overline{a}\|_0 + (t-1)k + \ell}{(t-1)k + \ell}} \, .$$

Setting the parameters and estimating appropriately (Lemma A.6), the bounds then follow. □

As a corollary, we recover the lower bound of Kayal [Kay12, GKKS14] by setting $m = 1$ in the above.

Corollary 4.19 (Kayal [Kay12], Gupta-Kamath-Kayal-Saptharishi [GKKS14])**.** *For any field $\mathbb{F}$, computing $x_1 \cdots x_n$ as a $\sum\bigwedge\sum\prod^t$ formula requires top-fan-in $\geq \exp(\Omega(n/t))$.* □

When combining the above structural result on the trailing monomial with the methods for obtaining hitting sets from such (Corollary 3.15), we obtain the following hitting sets for $\sum\bigwedge\sum\prod^t$ formulas.

Corollary 4.20. *Let $\mathbb{F}$ be a field with $\mathrm{char}(F) > d$. Then the class of n-variate, degree-($\leq d$) polynomials $f(\overline{x}) \in \mathbb{F}[x_1, \ldots, x_n]$ computed as $f(\overline{x}) = \sum_{i=1}^{s} f_i(\overline{x})^{d_i}$ where $\deg f_i \leq t$ ($\sum\bigwedge\sum\prod^t$ formula with top-fan-in s) has a $\mathsf{poly}(n, d, t \lg s)$-explicit hitting set of size $\mathsf{poly}(n, d)^{\mathcal{O}(t \lg s)}$.* □

5 Small-Support Monomials for Sparse Polynomials

In this section we introduce a new operator $((\overline{\boldsymbol{x}}+\overline{\boldsymbol{\alpha}})\circ\boldsymbol{\partial}_{\overline{x}})$ which is a variant of the partial derivative operator, defining the measure of f to be the dimension of this operator on f. We will show that while the (shifted) partial derivative measure is translation invariant, this new measure is not and thus can hope to distinguish between a polynomial sparse in the $\overline{x}$ basis versus a (sufficiently different) translated basis $\overline{x}+\overline{\alpha}$. After defining this measure, we show directly that that it is small on sparse polynomials in the correct basis. We then lower bound the number of trailing monomials of the $((\overline{\boldsymbol{x}}+\overline{\boldsymbol{\alpha}})\circ\boldsymbol{\partial}_{\overline{x}})$ operator on f in terms of the number of trailing monomials of the partial derivative space of f, as well as in terms of the number of such monomials in the partial derivative space of $\mathrm{TM}(f)$. In particular, a monomial in an incorrect basis will have a large $((\overline{\boldsymbol{x}}+\overline{\boldsymbol{\alpha}})\circ\boldsymbol{\partial}_{\overline{x}})$ measure. Combining these facts gives the desired upper bounds for the number of variables in the trailing monomial of a sparse polynomial in the incorrect basis.

5.1 Shifted Partial Derivatives are Translation Invariant

We begin by showing that the dimension of shifted partial derivatives is invariant under translation. While in most cases such invariance would be a positive feature of a measure (as many important features of polynomials are basis independent), its invariance makes this measure particularly unsuited to distinguishing the monomial $\overline{x}^{\overline{1}}$ from its translation $(\overline{x}+\overline{1})^{\overline{1}}$.

We begin with a basic fact about Hasse derivatives that is simple case of the chain rule in case of translation maps.

Lemma 5.1. *Let $f\in\mathbb{F}[x_1,\ldots,x_n]$ and $\overline{\alpha}\in\mathbb{F}^n$. Then*

$$\partial_{\overline{x}^{\overline{a}}}(f(\overline{x}+\overline{\alpha}))=(\partial_{\overline{x}^{\overline{a}}}f)(\overline{x}+\overline{\alpha})\ .$$

Proof: By the definition of Hasse derivatives, we have that

$$f(\overline{x}+\overline{z})=\sum_{\overline{a}}(\partial_{\overline{x}^{\overline{a}}}f)(\overline{x})\cdot\overline{z}^{\overline{a}}\ .$$

Similarly, for $g(\overline{x}):=f(\overline{x}+\overline{\alpha})$ we have

$$g(\overline{x}+\overline{z})=\sum_{\overline{a}}(\partial_{\overline{x}^{\overline{a}}}g)(\overline{x})\cdot\overline{z}^{\overline{a}}\ .$$

Applying the homomorphism induced by the substitution map $\overline{x}\mapsto\overline{x}+\overline{\alpha}$ we see that

$$\sum_{\overline{a}}(\partial_{\overline{x}^{\overline{a}}}g)(\overline{x})\overline{z}^{\overline{a}}=g(\overline{x}+\overline{z})=f(\overline{x}+\overline{\alpha}+\overline{z})=\sum_{\overline{a}}(\partial_{\overline{x}^{\overline{a}}}f)(\overline{x}+\overline{\alpha})\overline{z}^{\overline{a}}\ .$$

Taking coefficients with respect to $\overline{z}$ yields the claim. □

We now show how the dimension of shifted partial derivatives does not change under translation.

Lemma 5.2. *For any $f\in\mathbb{F}[x_1,\ldots,x_n]$, $\overline{\alpha}\in\mathbb{F}^n$ and $\ell,k\geq 0$,*

$$\dim\overline{\boldsymbol{x}}^{\leq\ell}\boldsymbol{\partial}_{\overline{x}^{\leq k}}(f(\overline{x}))=\dim\overline{\boldsymbol{x}}^{\leq\ell}\boldsymbol{\partial}_{\overline{x}^{\leq k}}(f(\overline{x}+\overline{\alpha}))\ .$$

Proof: $\underline{\leq}$: Consider the homomorphism $\varphi : \mathbb{F}[\overline{x}] \to \mathbb{F}[\overline{x}]$ induced by the translation $\overline{x} \mapsto \overline{x} + \overline{\alpha}$. Then,

$$\dim \operatorname{span} \boldsymbol{\overline{x}}^{\leq \ell} \boldsymbol{\partial}_{\overline{x}^{\leq k}} f(\overline{x}) = \dim \operatorname{span} \left\{ \overline{x}^{\overline{c}} \partial_{\overline{x}^{\overline{b}}} f(\overline{x}) \right\}_{\deg \overline{x}^{\overline{c}} \leq \ell, \deg \overline{x}^{\overline{b}} \leq k}$$

as φ is an invertible linear map,

$$= \dim \operatorname{span} \left\{ \varphi \left(\overline{x}^{\overline{c}} \partial_{\overline{x}^{\overline{b}}} f(\overline{x}) \right) \right\}_{\deg \overline{x}^{\overline{c}} \leq \ell, \deg \overline{x}^{\overline{b}} \leq k}$$
$$= \dim \operatorname{span} \left\{ (\overline{x} + \overline{\alpha})^{\overline{c}} \left(\partial_{\overline{x}^{\overline{b}}} f \right) (\overline{x} + \overline{\alpha}) \right\}_{\overline{c}, \overline{b}}$$

appealing to (a case of) the chain rule (Lemma 5.1),

$$= \dim \operatorname{span} \left\{ (\overline{x} + \overline{\alpha})^{\overline{c}} \partial_{\overline{x}^{\overline{b}}} (f(\overline{x} + \overline{\alpha})) \right\}_{\overline{c}, \overline{b}}$$

using that $\deg(\overline{x} + \overline{\alpha})^{\overline{c}} = \deg \overline{x}^{\overline{c}}$,

$$\leq \dim \operatorname{span} \left\{ \overline{x}^{\overline{c}} \partial_{\overline{x}^{\overline{b}}} (f(\overline{x} + \overline{\alpha})) \right\}_{\overline{c}, \overline{b}}$$
$$= \dim \operatorname{span} \boldsymbol{\overline{x}}^{\leq \ell} \boldsymbol{\partial}_{\overline{x}^{\leq k}} (f(\overline{x} + \overline{\alpha})) \ . \qquad \square$$

$\underline{\geq}$: Apply the above argument with $g(\overline{x}) := f(\overline{x} + \overline{\alpha})$ and $g(\overline{x} + (-\overline{\alpha})) = f(\overline{x})$.

Note that this result is also true for the $\boldsymbol{\overline{x}}^{\leq \ell} \boldsymbol{\partial}_{\overline{x}^{<\infty}}$ measure, as for any polynomial f and $k \geq \deg f$ we have $\boldsymbol{\overline{x}}^{\leq \ell} \boldsymbol{\partial}_{\overline{x}^{<\infty}}(f) = \boldsymbol{\overline{x}}^{\leq \ell} \boldsymbol{\partial}_{\overline{x}^{\leq k}}(f)$ (and degree is invariant under $\overline{x} \mapsto \overline{x} + \overline{\alpha}$). This fact will also be used below without mention.

5.2 The Measure and Upper Bounds for Computation

We now give the new measure that will allow us to distinguish between sparse polynomials in different bases.

Definition 5.3. *Let $\overline{\alpha} \in \mathbb{F}^n$. The operator $((\boldsymbol{\overline{x}} + \boldsymbol{\overline{\alpha}}) \circ \boldsymbol{\partial}_{\overline{x}})^{\leq k} : \mathbb{F}[x_1, \ldots, x_n] \to 2^{\mathbb{F}[\overline{x}]}$ is defined by*

$$((\boldsymbol{\overline{x}} + \boldsymbol{\overline{\alpha}}) \circ \boldsymbol{\partial}_{\overline{x}})^{\leq k} (f) := \left\{ (\overline{x} + \overline{\alpha})^{\overline{b}} \partial_{\overline{x}^{\overline{b}}} f \right\}_{\deg \overline{x}^{\overline{b}} \leq k} \ .$$

Define $((\boldsymbol{\overline{x}} + \boldsymbol{\overline{\alpha}}) \circ \boldsymbol{\partial}_{\overline{x}})^{<\infty} (f)$ to be the union of $((\boldsymbol{\overline{x}} + \boldsymbol{\overline{\alpha}}) \circ \boldsymbol{\partial}_{\overline{x}})^{\leq k} (f)$ over all $k \in \mathbb{N}$. ◊

Remark 5.4. We briefly explain the choice of notation. In the above '$\partial_{\overline{x}}$' will actually represent the vector $(\partial_{x_1}, \ldots, \partial_{x_n})$, which is typically known as the gradient and denoted '∇'. However, to preserve similarity with the notation of shifted partial derivatives we have chosen to use '$\partial_{\overline{x}}$' as this notation is not meant to read literally. The '$\circ$' is intended here to be a component-wise product of the vectors $\overline{x} + \overline{\alpha}$ and $\partial_{\overline{x}}$, so that $(\overline{x} + \overline{\alpha}) \circ \partial_{\overline{x}}$ is intended to be the vector of differential operators $((x_1 + \alpha_1)\partial_{x_1}, \ldots, (x_n + \alpha_n)\partial_{x_n})$. That we then take this entire vector to the power '$\leq k$' means that we should consider all degree $\leq k$ monomials in this vector. Read literally, this would yield $((\overline{x} + \overline{\alpha}) \circ \partial_{\overline{x}})^{\overline{b}} = \prod_{i=1}^{n} ((x_i + \alpha_i)\partial_{x_i})^{b_i}$. Using commutativity of variable disjoint differential operators (as differential operators, $y\partial_x = \partial_x y$ for $x \neq y$), one can see that this equals $(\overline{x} + \overline{\alpha})^{\overline{b}} \partial_{\overline{x}}^{\overline{b}}$. Note that (following our notational conventions (Section 2)) $\partial_{\overline{x}}^{\overline{b}}$ is actually a partial derivative and not the intended Hasse derivative of $\partial_{\overline{x}^{\overline{b}}}$, as iterated Hasse derivatives are not Hasse derivatives. As such, the notation above is merely meant to be suggestive and not interpreted literally as a composition of maps (in contrast, the shift and derivative operators of Definition 4.1 can be interpreted literally as maps). ◊

It is illuminating to briefly contrast this measure with shifted partial derivatives, and setting $\overline{\alpha} = \overline{0}$ makes this comparison easier. Shifted partial derivatives considers sets of differential operators $\{\overline{x}^{\overline{c}}\partial_{\overline{x}^{\overline{b}}}\}_{\overline{c},\overline{b}}$ where the exponents $\overline{c}$ and $\overline{b}$ are *uncorrelated*. In contrast, this $((\overline{\boldsymbol{x}} + \overline{\boldsymbol{\alpha}}) \circ \boldsymbol{\partial}_{\overline{x}})$ measure considers differential operators $\{\overline{x}^{\overline{c}}\partial_{\overline{x}^{\overline{b}}}\}_{\overline{c}=\overline{b}}$, so that the exponents $\overline{c}$ and $\overline{b}$ are *strongly correlated* (and in particular are equal). The use of correlation seems novel to this paper and drives the results of this section.

We now show that the measure is small for monomials in the correct basis.

Lemma 5.5. *Let $\overline{\alpha} \in \mathbb{F}^n$ and $(\overline{x} + \overline{\alpha})^{\overline{a}} \in \mathbb{F}[x_1, \ldots, x_n]$. Then*

$$\left|\mathrm{TM}\left(((\overline{\boldsymbol{x}} + \overline{\boldsymbol{\alpha}}) \circ \boldsymbol{\partial}_{\overline{x}})^{<\infty}\left((\overline{x} + \overline{\alpha})^{\overline{a}}\right)\right)\right| = \dim\operatorname{span}((\overline{\boldsymbol{x}} + \overline{\boldsymbol{\alpha}}) \circ \boldsymbol{\partial}_{\overline{x}})^{<\infty}\left((\overline{x} + \overline{\alpha})^{\overline{a}}\right) = 1\ .$$

Proof: $\underline{\dim = 1:}$ Observe that $(\overline{x} + \overline{\alpha})^{\overline{b}}\partial_{\overline{x}^{\overline{b}}}(\overline{x} + \overline{\alpha})^{\overline{a}} = (\overline{x} + \overline{\alpha})^{\overline{b}} \cdot \binom{\overline{a}}{\overline{b}}(\overline{x} + \overline{\alpha})^{\overline{a}-\overline{b}} = \binom{\overline{a}}{\overline{b}}(\overline{x} + \overline{\alpha})^{\overline{a}}$, where we can compute the Hasse derivative of $(\overline{x} + \overline{\alpha})^{\overline{a}}$ via the chain rule (Lemma 5.1). Thus, $\operatorname{span}((\overline{\boldsymbol{x}} + \overline{\boldsymbol{\alpha}}) \circ \boldsymbol{\partial}_{\overline{x}})^{<\infty}\left((\overline{x} + \overline{\alpha})^{\overline{a}}\right) \subseteq \operatorname{span}(\overline{x} + \overline{\alpha})^{\overline{a}}$. Taking $\overline{b} = \overline{0}$ shows that these spans are equal as $\binom{\overline{a}}{\overline{0}} = 1$, giving the claim.

$\underline{\mathrm{TM} = 1:}$ This follows directly from the above analysis. □

Note that the space $((\overline{\boldsymbol{x}} + \overline{\boldsymbol{\alpha}}) \circ \boldsymbol{\partial}_{\overline{x}})^{<\infty}((\overline{x} + \overline{\alpha})^{\overline{a}})$ is not strictly speaking a set of terms. That is, we defined 'terms' to be in the $\overline{x}$-basis, but the polynomials in $((\overline{\boldsymbol{x}} + \overline{\boldsymbol{\alpha}}) \circ \boldsymbol{\partial}_{\overline{x}})^{<\infty}((\overline{x} + \overline{\alpha})^{\overline{a}})$ are terms in the $(\overline{x} + \overline{\alpha})$-basis. Thus, the equality of the number of trailing monomials and the dimension (Corollary 3.11) does not strictly apply, but does in fact hold by applying in the translated basis (but note that trailing monomials in the $\overline{x}$- and $(\overline{x} + \overline{\alpha})$-basis are vastly different; we will only consider trailing monomials in the $\overline{x}$-basis in this work).

We now extend this bound to sparse polynomials using subadditivity (Lemma 4.3).

Corollary 5.6. *Let $f(\overline{x}) \in \mathbb{F}[x_1, \ldots, x_n]$ be an $(\leq s)$-sparse polynomial. For any $\overline{\alpha} \in \mathbb{F}^n$,*

$$\dim\operatorname{span}((\overline{\boldsymbol{x}} + \overline{\boldsymbol{\alpha}}) \circ \boldsymbol{\partial}_{\overline{x}})^{<\infty}(f(\overline{x} + \overline{\alpha})) \leq s\ .$$

□

5.3 Relating the Measure to Trailing Monomials

We now demonstrate how the $((\overline{\boldsymbol{x}} + \overline{\boldsymbol{\alpha}}) \circ \boldsymbol{\partial}_{\overline{x}})^{\leq k}$ measure interacts with trailing monomials. Note that these lower bound results will only hold when the translation $\overline{\alpha}$ has full support, in contrast to the upper bound of Lemma 5.5 which holds for any $\overline{\alpha}$. We will actually consider the more general operator $\overline{\boldsymbol{x}}^{\leq \ell}((\overline{\boldsymbol{x}} + \overline{\boldsymbol{\alpha}}) \circ \boldsymbol{\partial}_{\overline{x}})^{\leq k}$ as the proof is not more difficult and we will use this generalization in Section 6.

Lemma 5.7. *Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ and let $\ell, k \geq 0$. Suppose $\overline{\alpha} \in (\mathbb{F} \setminus \{0\})^n$, so that $\overline{\alpha}$ has full support. Then*

$$\left|\mathrm{TM}\left(\overline{\boldsymbol{x}}^{\leq \ell}((\overline{\boldsymbol{x}} + \overline{\boldsymbol{\alpha}}) \circ \boldsymbol{\partial}_{\overline{x}})^{\leq k}(f)\right)\right| = \left|\mathrm{TM}\left(\overline{\boldsymbol{x}}^{\leq \ell}\boldsymbol{\partial}_{\overline{x}^{\leq k}}(f)\right)\right|\ .$$

In particular, with $\ell = 0$,

$$\left|\mathrm{TM}\left(((\overline{\boldsymbol{x}} + \overline{\boldsymbol{\alpha}}) \circ \boldsymbol{\partial}_{\overline{x}})^{\leq k}(f)\right)\right| = \left|\mathrm{TM}\left(\boldsymbol{\partial}_{\overline{x}^{\leq k}}(f)\right)\right|\ .$$

Proof: Let $S := \overline{\boldsymbol{x}}^{\leq \ell}((\overline{\boldsymbol{x}} + \overline{\boldsymbol{\alpha}}) \circ \boldsymbol{\partial}_{\overline{x}})^{\leq k}(f)$ and $T = \overline{\boldsymbol{x}}^{\leq \ell}\boldsymbol{\partial}_{\overline{x}^{\leq k}}(f)$. We prove the claim by showing that $\mathrm{TM}(S) = \mathrm{TM}(T)$. Consider some exponent vectors $\overline{c}, \overline{b}$, so that $\overline{x}^{\overline{c}}\partial_{\overline{x}^{\overline{b}}}(f) \in T$ and $\overline{x}^{\overline{c}}(\overline{x} + \overline{\alpha})^{\overline{b}}\partial_{\overline{x}^{\overline{b}}}(f) \in S$. Then $\partial_{\overline{x}^{\overline{b}}}(f) = 0$ iff $\overline{x}^{\overline{c}}(\overline{x} + \overline{\alpha})^{\overline{b}}\partial_{\overline{x}^{\overline{b}}}(f) = 0$, so that these polynomials either both have a trailing monomial (if they are non-zero) or they both do not have a trailing monomial (if they are both zero). If they are both non-zero, then $\partial_{\overline{x}^{\overline{b}}}(f) \neq 0$ in particular, and

$$\mathrm{TM}\left(\overline{x}^{\overline{c}}(\overline{x}+\overline{\alpha})^{\overline{b}}\partial_{\overline{x}^{\overline{b}}}(f)\right) = \mathrm{TM}\left(\overline{x}^{\overline{c}}\right)\mathrm{TM}\left((\overline{x}+\overline{\alpha})^{\overline{b}}\right)\cdot\mathrm{TM}\left(\partial_{\overline{x}^{\overline{b}}}(f)\right) = \mathrm{TM}\left(\overline{x}^{\overline{c}}\partial_{\overline{x}^{\overline{b}}}(f)\right) ,$$

where this equality is the content of Lemma 3.6, where we use that $h(\overline{x}) := (\overline{x}+\overline{\alpha})^{\overline{b}}$ has $h(\overline{0}) = \overline{\alpha}^{\overline{b}} \neq 0$ so that $\mathrm{TM}(h) = 1$, as $\overline{\alpha}$ has full support. □

Remark 5.8. As mentioned before, we use trailing monomials in this work as opposed to leading monomials in this work. The above result is the reason for this choice. In particular, the above is false for leading monomials as seen by considering $((\overline{\boldsymbol{x}}+\overline{\boldsymbol{1}})\circ\boldsymbol{\partial}_{\overline{x}})^{<\infty}\overline{x}^{\overline{1}}$ as compared to $\boldsymbol{\partial}_{\overline{x}^{<\infty}}(\overline{x}^{\overline{1}})$. For any $\overline{b}$, $(\overline{x}+\overline{1})^{\overline{b}}\partial_{\overline{x}^{\overline{b}}}\overline{x}^{\overline{1}} = (\overline{x}+\overline{1})^{\overline{b}}\binom{\overline{1}}{\overline{b}}\overline{x}^{\overline{1}-\overline{b}}$. It is not hard to see that this polynomial is either zero, or the leading monomial is $\overline{x}^{\overline{1}}$. Thus, $|\,\mathrm{TM}(((\overline{\boldsymbol{x}}+\overline{\boldsymbol{1}})\circ\boldsymbol{\partial}_{\overline{x}})^{<\infty}\overline{x}^{\overline{1}})| \le 1$. In contrast, by the below Lemma 5.12, we see that $|\,\mathrm{TM}(\boldsymbol{\partial}_{\overline{x}^{<\infty}}(\overline{x}^{\overline{1}}))| \ge 2^n$, but $2^n \not\le 1$ for $n \ge 1$.

Put another way, the reason for the trailing monomial is that it can be changed by translation, while the leading monomial cannot. That is, $\mathrm{LM}(f(\overline{x})) = \mathrm{LM}(f(\overline{x}+\overline{\alpha}))$ for any $\overline{\alpha}$, but $\mathrm{TM}(f(\overline{x})) \neq \mathrm{TM}(f(\overline{x}+\overline{\alpha}))$ in general. ♢

Remark 5.9. Note that in the end one would like to relate the dimension of these spaces, as the dimension of these spaces is the actual measure we will use in the end (in particular because dimension is sub-additive). However, the above lemma is *false* when we replace "$|\,\mathrm{TM}\,|$" with "dim". That is, consider again $(\overline{x}+\overline{1})^{\overline{1}}$. By Lemma 5.5 we see that $\dim\operatorname{span}((\overline{\boldsymbol{x}}+\overline{\boldsymbol{1}})\circ\boldsymbol{\partial}_{\overline{x}})^{<\infty}((\overline{x}+\overline{1})^{\overline{1}}) = 1$. However, by the below Lemma 5.12 we see that $\dim\operatorname{span}\boldsymbol{\partial}_{\overline{x}^{<\infty}}\overline{x}^{\overline{1}} \ge 2^n$ and as the partial derivative measure is translation invariant (Lemma 5.2) we have that $\dim\operatorname{span}\boldsymbol{\partial}_{\overline{x}^{<\infty}}(\overline{x}+\overline{1})^{\overline{1}} \ge 2^n$. However, $2^n \not\le 1$ for $n \ge 1$.

One way to interpret this is that as $\dim\operatorname{span} S \ge |\,\mathrm{TM}(S)|$ for any set of polynomials S (Lemma 3.9), getting a lower bound on $|\,\mathrm{TM}(S)|$ can be a stronger result than just getting a lower bound on $\dim\operatorname{span} S$. While in many cases we have that $\dim\operatorname{span} S \gtrsim |\,\mathrm{TM}(S)|$ (such as Corollary 3.11), so that these two measures are roughly comparable, this is not always true. That is, as seen above $\dim\operatorname{span}\boldsymbol{\partial}_{\overline{x}^{<\infty}}(\overline{x}+\overline{1})^{\overline{1}} \ge 2^n$ but as $\boldsymbol{\partial}_{\overline{x}^{<\infty}}(\overline{x}+\overline{1})^{\overline{1}} = \{\binom{\overline{1}}{\overline{b}}(\overline{x}+\overline{1})^{\overline{1}-\overline{b}}\}_{\overline{b}}$ we see that each polynomial in this space is zero or has a trailing monomial of 1, so that $|\,\mathrm{TM}(\boldsymbol{\partial}_{\overline{x}^{<\infty}}(\overline{x}+\overline{1})^{\overline{1}})| = 1$. As such, in getting the stronger lower bound on $|\,\mathrm{TM}(S)|$ one can expect to "do more" with such a lower bound, and the above Lemma 5.7 is an example. ♢

Remark 5.10. Note that $\overline{\alpha} \in (\mathbb{F}\setminus\{0\})^n$ is very much a necessary condition as seen via the proof. More abstractly, the goal of this section is to show that monomials in the $\overline{x}$ basis become highly non-sparse in a translated basis. However, the sparsity of $(\overline{x}+\overline{\alpha})^{\overline{1}}$ is $2^{\|\overline{\alpha}\|_0}$. Thus, for this to be maximally non-sparse we require that $\|\overline{\alpha}\|_0$ is maximal. ♢

We now relate the $((\overline{\boldsymbol{x}}+\overline{\boldsymbol{\alpha}})\circ\boldsymbol{\partial}_{\overline{x}})^{\le k}$ measure of f to *some* measure on its trailing monomial. While in the basic case of shifted partial derivatives (Lemma 4.8) the measure on f was related to the same measure of $\mathrm{TM}(f)$, we will now not consider $((\overline{\boldsymbol{x}}+\overline{\boldsymbol{\alpha}})\circ\boldsymbol{\partial}_{\overline{x}})^{\le k}$ on $\mathrm{TM}(f)$ but rather $\boldsymbol{\partial}_{\overline{x}^{\le k}}$. In fact, as the proof is no harder, we consider the shifted versions of these operators.

Lemma 5.11. *Let $f \in \mathbb{F}[x_1,\ldots,x_n]$ and $\ell, k \ge 0$. Suppose $\overline{\alpha} \in (\mathbb{F}\setminus\{0\})^n$ so that $\overline{\alpha}$ has full support. Then*

$$\dim\operatorname{span}\left(\overline{\boldsymbol{x}}^{\le\ell}\left((\overline{\boldsymbol{x}}+\overline{\boldsymbol{\alpha}})\circ\boldsymbol{\partial}_{\overline{x}}\right)^{\le k}(f)\right) \ge \dim\operatorname{span}\left(\overline{\boldsymbol{x}}^{\le\ell}\boldsymbol{\partial}_{\overline{x}^{\le k}}(\mathrm{TM}(f))\right) .$$

In particular, with $\ell = 0$,

$$\dim\operatorname{span}\left(\left((\overline{\boldsymbol{x}}+\overline{\boldsymbol{\alpha}})\circ\boldsymbol{\partial}_{\overline{x}}\right)^{\le k}(f)\right) \ge \dim\operatorname{span}\left(\boldsymbol{\partial}_{\overline{x}^{\le k}}(\mathrm{TM}(f))\right) .$$

Proof:

$$\begin{aligned}\dim \operatorname{span}\left(\overline{\boldsymbol{x}}^{\le \ell}\left((\overline{\boldsymbol{x}}+\overline{\boldsymbol{\alpha}})\circ \boldsymbol{\partial}_{\overline{x}}\right)^{\le k}(f)\right) &\ge \left|\mathrm{TM}\left(\overline{\boldsymbol{x}}^{\le \ell}\left((\overline{\boldsymbol{x}}+\overline{\boldsymbol{\alpha}})\circ \boldsymbol{\partial}_{\overline{x}}\right)^{\le k}(f)\right)\right| \\ &= \left|\mathrm{TM}\left(\overline{\boldsymbol{x}}^{\le \ell}\boldsymbol{\partial}_{\overline{x}^{\le k}}(f)\right)\right| \\ &\ge \dim \operatorname{span}\left(\overline{\boldsymbol{x}}^{\le \ell}\boldsymbol{\partial}_{\overline{x}^{\le k}}(\mathrm{TM}(f))\right)\ ,\end{aligned}$$

where the first line uses that dimension is lower bounded by the number of trailing monomials (Lemma 3.9), the second uses the relation between the trailing monomials in $((\overline{\boldsymbol{x}}+\overline{\boldsymbol{\alpha}})\circ \boldsymbol{\partial}_{\overline{x}})$ and the partial derivative space (Lemma 5.7), the third relates the trailing monomials of the shifted partial derivative space of f to that of the dimension of shifted partial derivatives of its trailing monomial (Lemma 4.8). □

5.4 Lower Bounds for the Measure of Monomials

We now proceed to give lower bounds on the measure of monomials. We will review known lower bounds for the partial derivative space of the monomial, and then use this to get lower bounds for our new measure. We begin with the space of (Hasse) derivatives.

Lemma 5.12 (see for example Kayal (in Saxena [Sax08]), or Forbes-Shpilka [FS13a])**.** *Let $\mathbb{F}$ be any field. Let $\overline{x}^{\overline{a}} \in \mathbb{F}[\overline{x}]$. Then*

$$\dim \operatorname{span} \boldsymbol{\partial}_{\overline{x}^{<\infty}}(\overline{x}^{\overline{a}}) = \left|\mathrm{TM}\left(\boldsymbol{\partial}_{\overline{x}^{<\infty}}\left(\overline{x}^{\overline{a}}\right)\right)\right| \ge 2^{\|\overline{a}\|_0}\ .$$

Proof: $\underline{\dim = |\mathrm{TM}|}$: Note that $\partial_{\overline{x}^{\overline{b}}}\overline{x}^{\overline{a}} = \binom{\overline{a}}{\overline{b}}\overline{x}^{\overline{a}-\overline{b}}$, which is a term (or zero). Thus, this part follows from the fact that dimension equals the number of distinct trailing monomials for such sets of polynomials (Corollary 3.11).

$\underline{\text{lower bound on } |\mathrm{TM}|}$: Consider the set of exponents

$$E := \{\overline{b} : b_i \in \{0, a_i\}\}\ .$$

Then for each i, $\binom{a_i}{b_i} = 1$ so that $\binom{\overline{a}}{\overline{b}} = \prod_i \binom{a_i}{b_i} = \prod_i 1 = 1$, so that $\partial_{\overline{x}^{\overline{b}}}\overline{x}^{\overline{a}} = \binom{\overline{a}}{\overline{b}}\overline{x}^{\overline{a}-\overline{b}} = \overline{x}^{\overline{a}-\overline{b}}$. These terms are in fact monomials and are all distinct. As clearly $|E| = 2^{\|\overline{a}\|_0}$ this establishes the claim. □

Note that one can say something stronger if $\operatorname{char}(\mathbb{F}) > \operatorname{ideg} \overline{x}^{\overline{a}}$, in that $\dim \operatorname{span} \boldsymbol{\partial}_{\overline{x}^{<\infty}}(\overline{x}^{\overline{a}}) = \prod_i (a_i + 1)$, called $\|\overline{a}\|_\times$ in [FS13a, For14].

5.5 Trailing Monomial Bounds for Computation

We now obtain upper bounds for the number of variables in the trailing monomials of sparse polynomials when they are in the incorrect basis.

Theorem 5.13. *Let $f(\overline{x}) \in \mathbb{F}[x_1, \ldots, x_n]$ be $(\le s)$-sparse, and let $\overline{\alpha} \in (\mathbb{F} \setminus \{0\})^n$ so that $\overline{\alpha}$ has full-support. Let $\overline{x}^{\overline{a}}$ be the trailing monomial of $f(\overline{x}+\overline{\alpha})$. Then*

$$\|\overline{a}\|_0 \le \lg s\ .$$

Proof: By the upper bound on our measure for sparse polynomials (Corollary 5.6),

$$s \ge \dim \operatorname{span}\left((\overline{\boldsymbol{x}}+\overline{\boldsymbol{\alpha}})\circ \boldsymbol{\partial}_{\overline{x}}\right)^{<\infty}(f(\overline{x}+\overline{\alpha}))$$

relating $((\boldsymbol{\overline{x}} + \boldsymbol{\overline{\alpha}}) \circ \boldsymbol{\partial}_{\overline{x}})$ on $f(\overline{x}+\overline{\alpha})$ to $\boldsymbol{\partial}_{\overline{x}}$ on $\mathrm{TM}(f(\overline{x}+\overline{\alpha}))$ (Lemma 5.11),

$$\geq \dim \operatorname{span}\left(\boldsymbol{\partial}_{\overline{x}^{<\infty}}(\overline{x}^{\overline{a}})\right)$$

using our lower bound on this measure of the monomial (Lemma 5.12),

$$\geq \|\overline{a}\|_0 \ . \qquad \square$$

In particular, we can take $\overline{\alpha} = \overline{1}$ to induce a small-support monomial. That is, we only need a *single* translation for sparse polynomials to induce such a monomial, which improves upon the results of Agrawal-Saha-Saxena [ASS13] which would use $\mathsf{poly}(s)^{\Omega(\lg s)}$ such translations for s-sparse polynomials.

Corollary 5.14. *Let $f(\overline{x}) \in \mathbb{F}[\overline{x}]$ be $(\leq s)$-sparse. Then the trailing monomial of $f(\overline{x}+\overline{1})$ involves $\leq \lg s$ variables.* □

Aside from methods that treat sparse polynomials as a sub-model of more complicated computation such as roABPs, methods for directly understanding sparse polynomials (e.g.,BenOr-Tiwari [BOT88], Klivans-Spielman [KS01], etc.) do not handle translations of sparse polynomials as these methods treat sparsity as a combinatorial parameter, and as such the combinatorics of the situation are not preserved under translation. Our method is more algebraic in flavor and as such yields (slightly worse) results for translations of sparse polynomials, as we now show.

Corollary 5.15. *Let $f(\overline{x}) \in \mathbb{F}[x_1, \ldots, x_n]$ be a translation of a $(\leq s)$-sparse polynomial. That is, there is some $\overline{\alpha} \in \mathbb{F}^n$ and $(\leq s)$-sparse $g(\overline{x}) \in \mathbb{F}[\overline{x}]$ where $f(\overline{x}) = g(\overline{x}+\overline{\alpha})$. Then except for $\leq n$ values of β, the trailing monomial of $f(\overline{x}+\beta\cdot\overline{1})$ involves $\leq \lg s$ variables.*

Proof: Observe that $f(\overline{x}+\beta\cdot\overline{1}) = g(\overline{x}+\overline{\alpha}+\beta\cdot\overline{1})$, which is the translation of $g(\overline{x})$ under the map $\overline{x} \mapsto \overline{x} + (\overline{\alpha}+\beta\cdot\overline{1})$. By Theorem 5.13, whenever $\overline{\alpha}+\beta\cdot\overline{1} \in (\mathbb{F}\setminus\{0\})^n$, so that it has full support, it follows that the trailing monomial of $f(\overline{x}+\beta\cdot\overline{1})$ involves $\leq \lg s$ variables. Now note that for any i there is exactly 1 value of β so that $(\overline{\alpha}+\beta\cdot\overline{1})_i = \alpha_i + \beta\cdot 1 = 0$ (that is, $\beta = -\alpha_i$). Thus, it follows that there at most most n values where $\overline{\alpha}+\beta\cdot\overline{1} \notin (\mathbb{F}\setminus\{0\})^n$, yielding the claim. □

The above shows that by trying several values of β we can induce a small-support monomial in translations of sparse polynomials. As translations of sparse polynomials are computable by small commutative roABPs (Lemma 3.18) we can apply the hitting sets of Forbes, Shpilka, and Saptharishi [FSS14] (see Theorem 3.16) to obtain the following hitting sets for translations of sparse polynomials.

Corollary 5.16. *Let $|\mathbb{F}| \geq \mathsf{poly}(n,d,s)$. The class of n-variate, degree-$(\leq d)$ polynomials $f(\overline{x}) \in \mathbb{F}[x_1, \ldots, x_n]$ that are translations of some s-sparse polynomial has a$\mathsf{poly}(n,d,s)$-explicit hitting set of size $\mathsf{poly}(n,d,s)^{\mathcal{O}(\lg\lg s)}$.* □

6 Small-Support Monomials for $\sum \mathrm{m} \bigwedge \sum \prod^{\mathcal{O}(1)}$ Formulas

In the previous sections, we saw how the shifted partial derivative method $(\boldsymbol{\overline{x}}^{\leq \ell}\boldsymbol{\partial}_{\overline{x}^{\leq k}})$ can yield upper bounds on the trailing monomial of $\sum\bigwedge\sum\prod^t$ formulas (Section 4), and how another set of differential operators $(((\boldsymbol{\overline{x}}+\boldsymbol{\overline{\alpha}}) \circ \boldsymbol{\partial}_{\overline{x}})^{\leq k})$ can yield upper bounds on the trailing monomial of (translations of) sparse polynomials (Section 5). As such, we obtained PIT algorithms for these respective classes.

In this section, we consider how to get a PIT algorithm that handles both of these computational models simultaneously. In particular, we consider a computational model that subsumes these models, that of sum of products of $\sum\bigwedge\sum\prod^t$ formulas and sparse polynomials, which we defined as $\sum \mathrm{m} \bigwedge\sum\prod^t$ formulas above. Polynomial identity testing for this class is delicate as it requires a unified understanding of sparse polynomials (where $t = 0$) and $\sum\bigwedge\sum\prod^t$ formulas (which as computational models are mutually incomparable).

The previous work of Saha, Saptharishi and Saxena [SSS13] understood the $t = 1$ case by (when reinterpreted via the results of Forbes and Shpilka [FS13b]) converting the $\sum \mathrm{m} \bigwedge\sum$ formula to a (commutative) roABP and applying the previously mentioned PIT algorithms. While roABPs are a robust enough model to subsume sparse polynomials, $\sum\bigwedge\sum$, and $\sum \mathrm{m} \bigwedge\sum$ formulas, they provably cannot compute $\sum\bigwedge\sum\prod^2$ formulas (Section 8). As such, a new approach is needed for PIT of these formulas.

In this section, we take a different approach and build on the previous two sections. In particular, we observe that the measures considered ($\boldsymbol{\overline{x}}^{\le \ell}\boldsymbol{\partial}_{\overline{x}^{\le k}}$ and $((\boldsymbol{\overline{x}}+\boldsymbol{\overline{\alpha}})\circ\boldsymbol{\partial}_{\overline{x}})^{\le k}$) can be combined to handle both sparse polynomials and $\sum\bigwedge\sum\prod^t$ formula simultaneously, and further this measure can handle the larger class $\sum \mathrm{m} \bigwedge\sum\prod^t$. This model of computation, aside from being challenging as it requires techniques beyond those used for roABPs, naturally captures the complexity of certain divisibility testing questions as we will see in Section 7.

The results of this section will be relatively straightforward combinations of the techniques of the previous two sections. In particular, setting $t = 0$ in our results for $\sum \mathrm{m} \bigwedge\sum\prod^t$ will recover the results for sparse polynomials. While we could thus recover the main results of Section 5 on sparse polynomials via the results of this section, we covered the sparse polynomial case separately because it more clearly identifies the logical progression of techniques. Further, the results of Section 5 will work over characteristic, while the results here will need large characteristic.

We begin this section by showing that the two measures ($\boldsymbol{\overline{x}}^{\le \ell}\boldsymbol{\partial}_{\overline{x}^{\le k}}$ and $((\boldsymbol{\overline{x}}+\boldsymbol{\overline{\alpha}})\circ\boldsymbol{\partial}_{\overline{x}})^{\le k}$) considered so far in this work can be related.

Lemma 6.1. *Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ and $\overline{\alpha} \in \mathbb{F}^n$. Then*

$$\operatorname{span}((\boldsymbol{\overline{x}}+\boldsymbol{\overline{\alpha}})\circ\boldsymbol{\partial}_{\overline{x}})^{\le k}(f) \subseteq \operatorname{span}\boldsymbol{\overline{x}}^{\le k}\boldsymbol{\partial}_{\overline{x}^{\le k}}(f)\ .$$

In particular,

$$\dim\operatorname{span}((\boldsymbol{\overline{x}}+\boldsymbol{\overline{\alpha}})\circ\boldsymbol{\partial}_{\overline{x}})^{\le k}(f) \le \dim\operatorname{span}\boldsymbol{\overline{x}}^{\le k}\boldsymbol{\partial}_{\overline{x}^{\le k}}(f)\ .$$

Proof: For $(\overline{x}+\overline{\alpha})^{\overline{b}}\partial_{\overline{x}^{\overline{b}}}(f)$ with $\deg\overline{x}^{\overline{b}} \le k$, we see that $\deg(\overline{x}+\overline{\alpha})^{\overline{b}} = \deg\overline{x}^{\overline{b}} \le k$, and thus

$$(\overline{x}+\overline{\alpha})^{\overline{b}}\partial_{\overline{x}^{\overline{b}}}(f) \in \operatorname{span}\{\overline{x}^{\overline{c}}\partial_{\overline{x}^{\overline{b}}}(f)\}_{\deg\overline{x}^{\overline{c}}\le k} \subseteq \operatorname{span}\{\overline{x}^{\overline{c}}\partial_{\overline{x}^{\overline{b}'}}(f)\}_{\deg\overline{x}^{\overline{c}},\deg\overline{x}^{\overline{b}'}\le k}\ .$$

Taking this inclusion over all $\overline{b}$ gives the claim about the spans. The claim on dimension then follows immediately. □

One way to interpret this statement is that in studying the trailing monomials of (translations of) sparse polynomials we use a *sub-measure* of the shifted partial derivative measure.

In the parameters of interest for the shifted partial derivatives $\boldsymbol{\overline{x}}^{\le \ell}\boldsymbol{\partial}_{\overline{x}^{\le k}}$ of $\sum\bigwedge\sum\prod^{\mathcal{O}(1)}$, we take $\ell \approx tn$ and $k \approx n/t$ (Lemma A.3). As such, $k \ll \ell$ so that the additional shifts in $((\boldsymbol{\overline{x}}+\boldsymbol{\overline{\alpha}})\circ\boldsymbol{\partial}_{\overline{x}})^{\le k}$ resulting from the $(\overline{x}+\overline{\alpha})$ can be seen as negligible when compared to the shifts in $\boldsymbol{\overline{x}}^{\le \ell}$. As such, it is reasonable to expect the measure $\boldsymbol{\overline{x}}^{\ell}((\boldsymbol{\overline{x}}+\boldsymbol{\overline{\alpha}})\circ\boldsymbol{\partial}_{\overline{x}})^{\le k}$ on $\sum\bigwedge\sum\prod^{\mathcal{O}(1)}$ to be the roughly the same as the measure $\boldsymbol{\overline{x}}^{\ell}\boldsymbol{\partial}_{\overline{x}^{\le k}}$ on $\sum\bigwedge\sum\prod^{\mathcal{O}(1)}$. Similarly, as the $((\boldsymbol{\overline{x}}+\boldsymbol{\overline{\alpha}})\circ\boldsymbol{\partial}_{\overline{x}})^{\le k}$ measure is small on (translations of) sparse polynomials, it is reasonable to expect this measure to expect the $\boldsymbol{\overline{x}}^{\ell}((\boldsymbol{\overline{x}}+\boldsymbol{\overline{\alpha}})\circ\boldsymbol{\partial}_{\overline{x}})^{\leq k}$ measure to be somewhat small on (translations of) sparse polynomials. In particular, one could expect that this measure would prove a lower bound for the monomial $\overline{x}^{\overline{1}}$ for both $\sum\bigwedge\sum\prod^{\mathcal{O}(1)}$ and (translations of) sparse polynomials, as well as their common generalization of $\sum\mathrm{m}\bigwedge\sum\prod^{\mathcal{O}(1)}$.

By the above, we immediately see that the dimension of $\boldsymbol{\overline{x}}^{\ell}((\boldsymbol{\overline{x}}+\boldsymbol{\overline{\alpha}})\circ\boldsymbol{\partial}_{\overline{x}})^{\leq k}$ is at most that of $\boldsymbol{\overline{x}}^{\ell+k}\boldsymbol{\partial}_{\overline{x}^{\leq k}}$. However, this bound alone is not enough for our purposes as the dimension of $\boldsymbol{\overline{x}}^{\ell+k}\boldsymbol{\partial}_{\overline{x}^{\leq k}}$ is translation invariant (Lemma 5.2) which means it cannot recover the basis-dependent sparsity lower bounds of Section 5 much less generalize to $\sum\mathrm{m}\bigwedge\sum\prod^{t}$. As such, we instead generalize the previous upper bound for the measure of the monomial (Lemma 5.5), where we now allow a multiplication by an arbitrary polynomial f.

Lemma 6.2. *Let $\overline{x}^{\overline{a}}, f\in\mathbb{F}[x_1,\ldots,x_n]$ and $\overline{\alpha}\in\mathbb{F}^n$. Then*

$$\begin{aligned}\boldsymbol{\overline{x}}^{\leq\ell}((\boldsymbol{\overline{x}}+\boldsymbol{\overline{\alpha}})\circ\boldsymbol{\partial}_{\overline{x}})^{\leq k}\left((\overline{x}+\overline{\alpha})^{\overline{a}}f\right) &\subseteq \operatorname{span}\left((\overline{x}+\overline{\alpha})^{\overline{a}}\cdot\boldsymbol{\overline{x}}^{\leq\ell+k}\boldsymbol{\partial}_{\overline{x}^{\leq k}}(f)\right)\\ &:= \operatorname{span}\{(\overline{x}+\overline{\alpha})^{\overline{a}}\overline{x}^{\overline{c}}\partial_{\overline{x}^{\overline{b}}}(f)\}_{\deg\overline{x}^{\overline{c}}\leq\ell+k,\deg\overline{x}^{\overline{b}}\leq k}\ .\end{aligned}$$

In particular,

$$\dim\operatorname{span}\boldsymbol{\overline{x}}^{\leq\ell}((\boldsymbol{\overline{x}}+\boldsymbol{\overline{\alpha}})\circ\boldsymbol{\partial}_{\overline{x}})^{\leq k}\left((\overline{x}+\overline{\alpha})^{\overline{a}}f\right)\leq\dim\operatorname{span}\boldsymbol{\overline{x}}^{\leq\ell+k}\boldsymbol{\partial}_{\overline{x}^{\leq k}}(f)\ .$$

Proof: Consider $\overline{x}^{\overline{c}}(\overline{x}+\overline{\alpha})^{\overline{b}}\partial_{\overline{x}^{\overline{b}}}$ with $\deg\overline{x}^{\overline{c}}\leq\ell$ and $\deg\overline{x}^{\overline{b}}\leq k$. Then by the product rule of Hasse derivatives (Lemma 3.3),

$$\overline{x}^{\overline{c}}(\overline{x}+\overline{\alpha})^{\overline{b}}\partial_{\overline{x}^{\overline{b}}}\left((\overline{x}+\overline{\alpha})^{\overline{a}}f\right)=\overline{x}^{\overline{c}}(\overline{x}+\overline{\alpha})^{\overline{b}}\sum_{\overline{i}+\overline{j}=\overline{b}}\partial_{\overline{x}^{\overline{i}}}\left((\overline{x}+\overline{\alpha})^{\overline{a}}\right)\cdot\partial_{\overline{x}^{\overline{j}}}f$$

appealing to the chain rule (e.g., Lemma 5.1),

$$\begin{aligned}&=\overline{x}^{\overline{c}}(\overline{x}+\overline{\alpha})^{\overline{b}}\sum_{\overline{i}+\overline{j}=\overline{b}}\binom{\overline{a}}{\overline{i}}(\overline{x}+\overline{\alpha})^{\overline{a}-\overline{i}}\cdot\partial_{\overline{x}^{\overline{j}}}f\\ &=\overline{x}^{\overline{c}}\sum_{\overline{i}+\overline{j}=\overline{b}}\binom{\overline{a}}{\overline{i}}(\overline{x}+\overline{\alpha})^{\overline{a}}\cdot(\overline{x}+\overline{\alpha})^{\overline{b}-\overline{i}}\cdot\partial_{\overline{x}^{\overline{j}}}f\end{aligned}$$

using that $\overline{0}\leq\overline{b}-\overline{i}=\overline{j}\leq\overline{b}$ so that $\deg\overline{x}^{\overline{c}}(\overline{x}+\overline{\alpha})^{\overline{b}-\overline{i}}\leq\ell+k$,

$$\in\operatorname{span}\{(\overline{x}+\overline{\alpha})^{\overline{a}}\overline{x}^{\overline{c}'}\partial_{\overline{x}^{\overline{b}'}}(f)\}_{\deg\overline{x}^{\overline{c}'}\leq\ell+k,\deg\overline{x}^{\overline{b}'}\leq k}$$

The statement about dimension follows immediately as the polynomial $(\overline{x}+\overline{\alpha})^{\overline{a}}$ is a common factor and thus does not contribute to the dimension. □

Combining this result with the bound on the shifted partial derivatives for $\sum\bigwedge\sum\prod^{t}$ formulas (Corollary 4.4), we obtain the following corollary.

Corollary 6.3. *Let $f\in\mathbb{F}[y_1,\ldots,y_m]$ and $t\geq 1$. Suppose $\overline{g}\in(\mathbb{F}[x_1,\ldots,x_n])^m$, where each g_i is of degree $\leq t$. Then for any $\overline{\alpha}\in\mathbb{F}^n$ and $\ell,k\geq 0$,*

$$\boldsymbol{\overline{x}}^{\leq\ell}((\boldsymbol{\overline{x}}+\boldsymbol{\overline{\alpha}})\circ\boldsymbol{\partial}_{\overline{x}})^{\leq k}\left((\overline{x}+\overline{\alpha})^{\overline{a}}(f\circ\overline{g})\right)\subseteq\operatorname{span}\boldsymbol{\overline{x}}^{\leq tk+\ell}\left(\left[\boldsymbol{\partial}_{\overline{y}^{\leq k}}(f)\right](\overline{g})\right)\ .$$

In particular,

$$\dim \boldsymbol{\overline{x}}^{\le \ell}((\boldsymbol{\overline{x}}+\boldsymbol{\overline{\alpha}})\circ \boldsymbol{\partial}_{\overline{x}})^{\le k}\left((\overline{x}+\overline{\alpha})^{\overline{a}}(f\circ \overline{g})\right) \le \binom{n+tk+\ell}{tk+\ell}\cdot\binom{m+k}{k}\,. \qquad \square$$

Appealing to subadditivity (Lemma 4.3) yields an upper bound on our model $\sum \mathrm{m} \bigwedge \sum \prod^t$ (using that $\overline{x} \mapsto \overline{x}+\overline{\alpha}$ preserves degree).

Corollary 6.4. *Let $t \ge 1$. Consider $f(\overline{x}) = \sum_{i=1}^s (\overline{x}+\overline{\alpha})^{\overline{a}_i} f_i(g_{i,1}(\overline{x}+\overline{\alpha}),\dots,g_{i,m}(\overline{x}+\overline{\alpha}))$ where $g_{i,j} \in \mathbb{F}[x_1,\dots,x_n]$ with $\deg g_{i,j} \le t$, and $\overline{\alpha} \in \mathbb{F}^n$. Then*

$$\dim \boldsymbol{\overline{x}}^{\le \ell}((\boldsymbol{\overline{x}}+\boldsymbol{\overline{\alpha}})\circ \boldsymbol{\partial}_{\overline{x}})^{\le k}(f) \le s\binom{k+m}{m}\binom{n+tk+\ell}{tk+\ell}\,. \qquad \square$$

Thus, this corollary shows that our new measure $(\dim \boldsymbol{\overline{x}}^{\le \ell}((\boldsymbol{\overline{x}}+\boldsymbol{\overline{\alpha}})\circ \boldsymbol{\partial}_{\overline{x}})^{\le k})$ on $\sum \mathrm{m} \bigwedge \sum \prod^t$ is at most the old measure $(\dim \boldsymbol{\overline{x}}^{\le \ell}\boldsymbol{\partial}_{\overline{x}^{\le k}})$ on $\sum \mathrm{m} \bigwedge \sum \prod^{t+1}$ (that is, compare the above to Corollary 4.5). This is somewhat surprising in that both sparse polynomials and $\sum \bigwedge \sum$ can be essentially handled by the partial derivative method (Section 5 for sparse polynomials, Forbes-Shpilka [FS13a] for $\sum \bigwedge \sum$). However, to combine these two classes into $\sum \mathrm{m} \bigwedge \sum$ we need the shifted partial derivative method, despite the fact that this class reduces to roABPs and thus can also be understood via those techniques.

We now combine these results to obtain our structural theorem, which largely mirrors the proof of Proposition 4.18 except that we must inspect the interaction between the projection to few variables and the translation $\overline{x} \mapsto \overline{x}+\overline{\alpha}$.

Proposition 6.5. *Let $f(\overline{x}) \in \mathbb{F}[x_1,\dots,x_n]$, $t \ge 1$, and $\overline{\alpha} \in (\mathbb{F}\setminus\{0\})^n$ be of full-support. Suppose that f is a non-zero polynomial of the form $f(\overline{x}) = \sum_{i=1}^s \overline{x}^{\overline{b}_i} f_i(g_{i,1}(\overline{x}),\dots,g_{i,m}(\overline{x}))$ where $\deg g_{i,j} \le t$. Let $\overline{x}^{\overline{a}}$ be the trailing monomial of $f(\overline{x}+\overline{\alpha})$, $\overline{x}^{\overline{a}} := \mathrm{TM}(f(\overline{x}+\overline{\alpha}))$. If $\mathrm{char}(\mathbb{F}) > \mathrm{ideg}\,\overline{x}^{\overline{a}}$, then*

$$\|\overline{a}\|_0 \le 2\mathrm{e}^3(t+1)(\ln s + m\ln(2m)+1)\,.$$

In particular, for $m = O(1)$, $\|\overline{a}\|_0 \le \mathcal{O}(t\ln s)$.

Proof: We proceed as Proposition 4.18. That is, let $m = \|\overline{a}\|_0$, and let $S \subseteq [n]$ be the support of $\overline{a}$, $S := \mathrm{Supp}(\overline{a})$. We define the substitution homomorphism $\pi_S : \mathbb{F}[\overline{x}] \to \mathbb{F}[\overline{y}]$ as from Lemma 4.12 which zeroes out variables outside S and yields the isomorphism $\mathbb{F}[\overline{x}|_S] \xrightarrow{\sim} \mathbb{F}[\overline{y}]$.

Define $\hat{f}(\overline{y}+\overline{\alpha}|_S) := \pi_S(f(\overline{x}+\overline{\alpha}))$, and we now relate the complexity of $\hat{f}$ to f. First, note that as π_S is a homomorphism we have that π_S acts on $(\overline{x}+\overline{\alpha})^{\overline{b}}$ as follows.

$$\pi_S\left((\overline{x}+\overline{\alpha})^{\overline{b}}\right) = \prod_i \pi_S\left(x_i+\alpha_i\right)^{b_i} = \prod_{i\notin S}\alpha_i^{b_i}\cdot\prod_{i\in S}(y_i+\alpha_i)^{b_i} =: (\overline{\alpha}|_{[n]\setminus S})^{\overline{b}|_{[n]\setminus S}}\cdot(\overline{y}+\overline{\alpha}|_S)^{\overline{b}|_S}\,.$$

Extending this by linearity, for a polynomial $g(\overline{x}) = \sum_{\overline{b}} \beta_{\overline{b}}\overline{x}^{\overline{b}}$, π_S acts on $g(\overline{x}+\overline{\alpha})$ as

$$\begin{aligned}\pi_S(g(\overline{x}+\overline{\alpha})) &= \pi_S\left(\sum_{\overline{b}}\beta_{\overline{b}}(\overline{x}+\overline{\alpha})^{\overline{b}}\right)\\ &= \sum_{\overline{b}}\beta_{\overline{b}}\cdot(\overline{\alpha}|_{[n]\setminus S})^{\overline{b}|_{[n]\setminus S}}\cdot(\overline{y}+\overline{\alpha}|_S)^{\overline{b}|_S}\\ &= \hat{g}(\overline{y}+\overline{\alpha}|_S)\,,\end{aligned}$$

so that $\hat{g}(\overline{y})$ is given via

$$\hat{g}(\overline{y}) = \sum_{\overline{b}} \beta_{\overline{b}} \cdot (\overline{\alpha}|_{[n]\setminus S})^{\overline{b}|_{[n]\setminus S}} \cdot \overline{y}^{\overline{b}|_S} \ .$$

Thus, applying π_S to f, we get

$$\begin{aligned}
\hat{f}(\overline{y}+\overline{\alpha}|_S) &= \sum_{i=1}^{s} \pi_S\left(\overline{x}^{\overline{b}_i}\right) f_i(\pi_S(g_{i,1}(\overline{x}+\overline{\alpha})), \ldots, \pi_S(g_{i,m}(\overline{x}+\overline{\alpha}))) \\
&= \sum_{i=1}^{s} (\overline{\alpha}|_{[n]\setminus S})^{\overline{b}_i|_{[n]\setminus S}} \cdot (\overline{y}+\overline{\alpha}|_S)^{\overline{b}_i|_S} \cdot f_i(\hat{g}_{i,1}(\overline{y}+\overline{\alpha}|_S), \ldots, \hat{g}_{i,m}(\overline{y}+\overline{\alpha}|_S)) \\
&=: \sum_{i=1}^{s} (\overline{y}+\overline{\alpha}|_S)^{\overline{b}_i|_S} \cdot \tilde{f}_i(\hat{g}_{i,1}(\overline{y}+\overline{\alpha}|_S), \ldots, \hat{g}_{i,m}(\overline{y}+\overline{\alpha}|_S)) \ ,
\end{aligned}$$

where $\tilde{f}_i$ simply absorbs the constant $(\overline{\alpha}|_{[n]\setminus S})^{\overline{b}_i|_{[n]\setminus S}}$.

Thus, $\hat{f}(\overline{y}+\overline{\alpha}|_S)$ is of the same form and complexity as $f(\overline{x}+\overline{\alpha})$, except it is now a function in the fewer variables in the vector $\overline{y}+\overline{\alpha}|_S$. In particular, $\deg_{\overline{y}} \hat{g}_{i,j} \le \deg_{\overline{x}} g_{i,j}$. Thus, we can apply the upper bound on the $\overline{\boldsymbol{y}}^{\le \ell}((\overline{\boldsymbol{y}}+\overline{\boldsymbol{\alpha}}|_S) \circ \boldsymbol{\partial}_{\overline{y}})^{\le k}$ measure of $\sum \mathrm{m} \bigwedge \sum \prod^t$ formulas in the $\overline{y}+\overline{\alpha}|_S$ basis (Corollary 6.4) to $\hat{f}(\overline{y}+\overline{\alpha}|_S)$, using that there are only $\|\overline{a}\|_0$ variables, to obtain

$$\dim \overline{\boldsymbol{y}}^{\le \ell}((\overline{\boldsymbol{y}}+\overline{\boldsymbol{\alpha}}|_S) \circ \boldsymbol{\partial}_{\overline{y}})^{\le k}(\hat{f}(\overline{y}+\overline{\alpha}|_S)) \le s\binom{k+m}{m}\binom{\|\overline{a}\|_0 + tk + \ell}{tk+\ell} \ .$$

We now turn to the lower bound. As $\pi_S(\overline{x}^{\overline{a}}) = \overline{y}^{\overline{a}}$ so that $\hat{f}(\overline{y}+\overline{\alpha}|_S) \ne 0$, Lemma 4.12 implies that $\overline{y}^{\overline{a}}$ is also the trailing monomial of $\pi_S(f(\overline{x}+\overline{\alpha}))$. As $\overline{y}$ only has $\|\overline{a}\|_0$ variables, we obtain

$$\begin{aligned}
\dim \operatorname{span} \overline{\boldsymbol{y}}^{\le \ell}((\overline{\boldsymbol{y}}+\overline{\boldsymbol{\alpha}}|_S) \circ \boldsymbol{\partial}_{\overline{y}})^{\le k}(\hat{f}(\overline{y}+\overline{\alpha}|_S)) &\ge \dim \operatorname{span}\left(\overline{\boldsymbol{y}}^{\le \ell} \boldsymbol{\partial}_{\overline{y}^{\le k}}(\mathrm{TM}(\hat{f}(\overline{y}+\overline{\alpha}|_S)))\right) \\
&= \dim \operatorname{span}\left(\overline{\boldsymbol{y}}^{\le \ell} \boldsymbol{\partial}_{\overline{y}^{\le k}}(\overline{y}^{\overline{a}})\right) \\
&\ge \binom{\|\overline{a}\|_0}{k}\binom{\|\overline{a}\|_0 - k + \ell}{\ell} \ ,
\end{aligned}$$

where we have respectively applied Lemma 5.11, the definition of $\overline{y}^{\overline{a}}$, and (as the characteristic is large enough) Lemma 4.13.

Putting the above together, we have that for any $k, \ell \in \mathbb{N}$ that

$$s \ge \frac{1}{\binom{k+m}{m}} \frac{\binom{\|\overline{a}\|_0}{k}\binom{\|\overline{a}\|_0 - k + \ell}{\ell}}{\binom{\|\overline{a}\|_0 + tk + \ell}{tk+\ell}} \ .$$

Setting the parameters and estimating appropriately (Lemma A.6), the bounds then follow. □

Using this we can easily get a lower bound for $\sum \mathrm{m} \bigwedge \sum \prod^t$ formulas.

Corollary 6.6. *For any field $\mathbb{F}$ and $t \ge 1$, computing $(\overline{x}+\overline{1})^{\overline{1}} \in \mathbb{F}[x_1, \ldots, x_n]$ as a $\sum \mathrm{m} \bigwedge \sum \prod^t$ formula requires top-fan-in $\ge \exp(\Omega(n/t))$.* □

When combining the above structural result on the trailing monomial with the methods for obtaining hitting sets from such (Corollary 3.15), we obtain the following hitting sets for $\sum \mathrm{m} \bigwedge \sum \prod^t$ formulas.

Corollary 6.7. *Let $\mathbb{F}$ be a field with $\mathrm{char}(F) > d$ and $t \geq 1$. Then the class of n-variate, degree-($\leq d$) polynomials $f(\overline{x}) \in \mathbb{F}[x_1, \ldots, x_n]$ computed as $f(\overline{x}) = \sum_{i=1}^{s} \overline{x}^{\overline{a}_i} f_i(\overline{x})^{d_i}$ where $\deg f_i \leq t$ ($\sum \mathrm{m} \bigwedge \sum \prod^t$ formula with top-fan-in s) has a $\mathsf{poly}(n, d, t \lg s)$-explicit hitting set of size $\mathsf{poly}(n, d)^{\mathcal{O}(t \lg s)}$.* □

As $\sum \mathrm{m} \bigwedge \sum$ formula are computable by small commutative roABPs (Lemma 3.18) we can apply the hitting sets of Forbes, Shpilka, and Saptharishi [FSS14] (Theorem 3.16) to obtain the following hitting sets for this model.

Corollary 6.8. *Let $|\mathbb{F}| \geq \mathsf{poly}(n, d, s)$. The class of n-variate, degree-($\leq d$) polynomials $f(\overline{x}) \in \mathbb{F}[x_1, \ldots, x_n]$ computed as $f(\overline{x}) = \sum_{i=1}^{s} \overline{x}^{\overline{a}_i} f_i(\overline{x})^{d_i}$ where $\deg f_i \leq 1$ ($\sum \mathrm{m} \bigwedge \sum$ formula with top-fan-in s) has a $\mathsf{poly}(n, d, s)$-explicit hitting set of size $\mathsf{poly}(n, d, s)^{\mathcal{O}(\lg \lg s)}$.* □

7 Reducing Divisibility Testing to PIT

In this section we give our reduction from divisibility testing to polynomial identity testing. We begin by defining formal power series $\mathbb{F}[\![\overline{x}]\!]$ and review basic facts about them. We then give Strassen's [Str73] method of removing divisions from computations over the polynomial ring $\mathbb{F}[\overline{x}]$, where this method goes through the ring of formal power series $\mathbb{F}[\![\overline{x}]\!]$. While this method is well-known to be efficiently computable on general algebraic circuits, we review this method carefully so as to understand how much complexity this reduction adds when one applies it to restricted algebraic computation. We then observe that this method of removing division gives a simple way to check divisibility through polynomial identity testing and apply this to testing divisibility of a low-degree polynomial into a sparse polynomial.

7.1 Formal Power Series

We recall here the basics of formal power series. For more details see, for example, Shoup [Sho09, Section 16.8]

Definition 7.1. *The* ***ring of (formal) power series in variables*** $x_1, \ldots, x_n$ ***with coefficients in*** $\mathbb{F}$*, denoted $\mathbb{F}[\![x_1, \ldots, x_n]\!]$, is the ring of expressions*

$$\sum_{\overline{a} \in \mathbb{N}^n} \alpha_{\overline{a}} \overline{x}^{\overline{a}} \ ,$$

where we allow infinitely many $\overline{a}$ with $\alpha_{\overline{a}} \neq 0$. Addition is defined coordinate-wise, and multiplication is defined by $\overline{x}^{\overline{a}} \cdot \overline{x}^{\overline{b}} = \overline{x}^{\overline{a}+\overline{b}}$ and extended linearly.

The ***i-th homogeneous part*** *of the power series $f(\overline{x}) = \sum_{\overline{a}} \alpha_{\overline{a}} \overline{x}^{\overline{a}}$ is defined by*

$$\mathrm{H}_i(f) := \sum_{\deg \overline{x}^{\overline{a}} = i} \alpha_{\overline{a}} \overline{x}^{\overline{a}} \ . \qquad \Diamond$$

Note that the polynomial ring $\mathbb{F}[\overline{x}]$ is contained as a subring of $\mathbb{F}[\![\overline{x}]\!]$. However, while n-variate polynomials naturally define functions $\mathbb{F}^n \to \mathbb{F}$, power series cannot define functions unless the field $\mathbb{F}$ has a topology which allows a definition of convergence of infinite series.

We briefly recall some facts about taking homogeneous parts.

Lemma 7.2. *For $i \geq 0$, the taking the i-th homogeneous component $\mathrm{H}_i : \mathbb{F}[\![\overline{x}]\!] \to \mathbb{F}[\overline{x}]$ is a linear map.* □

Taking homogeneous parts of a multiplication can be decomposed into taking homogeneous parts of the multiplicands, a process known as *homogenization*.

Lemma 7.3. *Let $f, g \in \mathbb{F}[\![\overline{x}]\!]$. Then*

$$\mathrm{H}_i(fg) = \sum_{j+k=i} \mathrm{H}_j(f)\,\mathrm{H}_k(g)\ . \qquad \square$$

We now recall a basic fact about the invertibility of power series. This fact is the formal analogue of the Taylor series from calculus that shows that $\frac{1}{1-x} = 1 + x + x^2 + \cdots$ for $|x| < 1$.

Lemma 7.4. *Let $f(\overline{x}) \in \mathbb{F}[\![\overline{x}]\!]$ be a power series with no constant term, that is $\mathrm{H}_0(f) = 0$. Then $\mathrm{H}_j(f^i) = 0$ for $j > i$, and $1 - f$ is invertible in $\mathbb{F}[\![\overline{x}]\!]$ where its multiplicative inverse is*

$$(1-f)^{-1} = \frac{1}{1-f} = \sum_{i\geq 0} f^i = 1 + f + f^2 + \cdots\ ,$$

where $\sum_{i\geq 0} f^i$ is the power series defined by $\mathrm{H}_j(\sum_{i\geq 0} f^i) := \mathrm{H}_j(\sum_{0\leq i\leq j} f^i)$. In particular, we have that $\mathrm{H}_j(\sum_{i\geq 0} f^i) = \mathrm{H}_j(\sum_{0\leq i\leq k} f^i)$ for any $k \geq j$. □

7.2 Strassen's [Str73] Elimination of Division

We now arrive at Strassen's [Str73] method for removing a division gate (see also Shpilka and Yehudayoff [SY10]). We wish to compute a polynomial h within the polynomial ring $\mathbb{F}[\overline{x}]$, but that we receive a computation $h = g/f$ in the field of rational functions $\mathbb{F}(\overline{x})$. The field of rational functions $\mathbb{F}(\overline{x})$ cannot be embedded into the ring of power series $\mathbb{F}[\![\overline{x}]\!]$ in general (for example, x has a multiplicative inverse in $\mathbb{F}(\overline{x})$ but not in $\mathbb{F}[\![\overline{x}]\!]$) but one can still hope to embed some normalized version of computation $h = g/f$ into $\mathbb{F}[\![\overline{x}]\!]$. In particular, this embedding can succeed if f is invertible in $\mathbb{F}[\![\overline{x}]\!]$. By the above lemma, $f(\overline{x})$ is invertible if it has a constant term 1 and the inverse is given by $\sum_i (1-f)^i$, in which case $h = g/f = g\sum_i(1-f)^i$. While this expression is an infinite power series, we can appropriately truncate it as h is a polynomial.

Lemma 7.5 (Strassen [Str73])**.** *Let $f, g, h \in \mathbb{F}[\overline{x}] \subsetneq \mathbb{F}[\![\overline{x}]\!]$ are polynomials of degree $\leq d$, and suppose that $hf = g$ or equivalently $h = g/f$. Further, suppose that $f(\overline{0}) = 1$. Then,*

$$h(\overline{x}) = \sum_{i=0}^{d} \mathrm{H}_i\left(g(\overline{x}) \cdot \textstyle\sum_{j=0}^{d}(1 - f(\overline{x}))^j\right)\ . \qquad \square$$

Proof: Observe that $f = 1 - (1-f)$ and as $f(\overline{0}) = 1$ this means that $1 - f$ has no constant term, so that $\mathrm{H}_0(1-f) = 0$. Thus, by Lemma 7.4 we have that

$$f^{-1} = (1 - (1-f))^{-1} = \sum_{j\geq 0}(1-f)^j\ .$$

Applying this for $i \leq d$,

$$\begin{aligned}
\mathrm{H}_i(h) &= \mathrm{H}_i(g/f) \\
&= \mathrm{H}_i(g \cdot f^{-1}) \\
&= \mathrm{H}_i(g \cdot \textstyle\sum_{j\geq 0}(1-f)^j) \\
&= \mathrm{H}_i(g \cdot \textstyle\sum_{0\leq j\leq i}(1-f)^j) \\
&= \mathrm{H}_i(g \cdot \textstyle\sum_{0\leq j\leq d}(1-f)^j)\ .
\end{aligned}$$

As $h(\overline{x})$ is of degree $\leq d$ we have that $h(\overline{x}) = \sum_{0\leq i\leq d} \mathrm{H}_i(h)$, giving the result. □

The above showed how to compute $h = g/f$ without division when f is appropriately normalized. We now reduce the general case to this normalized case. While this reduction is standard and efficiently computable when f, g are given as general algebraic circuits, the complexity of this reduction when f, g are given as restricted circuits is more delicate. That is, the normalization will occur by applying the translation map $\varphi : \overline{x} \mapsto \overline{x} + \overline{\alpha}$. After this normalization, we will extract homogeneous parts as seen above. In our main application (Corollary 7.16) we will want $g(\overline{x})$ to be a sparse polynomial, and as such we will need to compute homogeneous parts of translations of $g(\overline{x})$. While homogeneous parts of sparse polynomials are sparse, and the methods of Section 5 give hitting sets for translations of sparse polynomials (which may be very non-sparse), the methods of this paper do not seem to in general give hitting sets for homogeneous parts of translations of sparse polynomials [12]. As such, to obtain our deterministic divisibility test Corollary 7.17 we need to carefully detail the exact form of the division removal.

We now work toward this normalization. The first observation is that divisibility is preserved under translation.

Lemma 7.6. *Let $f, g \in \mathbb{F}[\overline{x}]$. Then for any $\overline{\alpha}$, $f(\overline{x})|g(\overline{x})$ iff $f(\overline{x}+\overline{\alpha})|g(\overline{x}+\overline{\alpha})$.*

Proof: $\underline{\implies:}$ That $f(\overline{x})|g(\overline{x})$ means that there is an $h \in \mathbb{F}[\overline{x}]$ such that $f(\overline{x})h(\overline{x}) = g(\overline{x})$. Consider the substitution homomorphism $\varphi : \mathbb{F}[\overline{x}] \to \mathbb{F}[\overline{x}]$ induced by $\overline{x} \mapsto \overline{x} + \overline{\alpha}$. Thus $f(\overline{x}+\overline{\alpha})h(\overline{x}+\overline{\alpha}) = \varphi(f(\overline{x})h(\overline{x})) = \varphi(g(\overline{x})) = g(\overline{x}+\overline{\alpha})$, so $f(\overline{x}+\overline{\alpha})|g(\overline{x}+\overline{\alpha})$.

$\underline{\impliedby:}$ This follows from applying the first case to $\hat{f}(\overline{x}) := f(\overline{x}+\overline{\alpha})$ and $\hat{g}(\overline{x}) := g(\overline{x}+\overline{\alpha})$ along with the translation $-\overline{\alpha}$. □

In this reduction we will also need to compute homogeneous parts of polynomials. While this can be done for general algebraic circuits by the process of homogenizing the circuit (splitting each gate in the circuit into many gates, each of which computes a homogeneous part of the original gate), this process increases the complexity of this computation (in particular, the depth increases). An alternate well-known method for computing homogeneous parts is via interpolation, as we now state.

Lemma 7.7. *Let $f(\overline{x}) \in \mathbb{F}[\overline{x}]$ be of degree $\le d$. Let $S \subseteq \mathbb{F}$ a $\mathsf{poly}(d)$-explicit set with $|S| = d+1$. Then there are $\mathsf{poly}(d)$-explicit constants $\{\beta_{i,\alpha}\}_{0\le i\le d, \alpha\in S}$ which only depend on d and S such that*

$$\mathrm{H}_i(f(\overline{x})) = \sum_{\alpha\in S} \beta_{i,\alpha} f(\alpha\cdot\overline{x}) \, ,$$

where $\alpha\cdot\overline{x}$ is scalar multiplication of α on the vector $\overline{x}$. In particular, for a set $T \subseteq \{0, \ldots, d\}$,

$$\sum_{i\in T} \mathrm{H}_i(f(\overline{x})) = \sum_{\alpha\in S} \left(\textstyle\sum_{i\in T}\beta_{i,\alpha}\right) f(\overline{x}\cdot\alpha) \, . \qquad \square$$

The following technical lemma will also be helpful.

Lemma 7.8. *For $0 \le k \le d$, over $\mathbb{F}[x]$,*

$$\textstyle\sum_{0\le k\le d}(1+x)^i = \sum_{0\le k\le d}\binom{d+1}{k+1}x^k \, .$$

[12] Note that translation does not commute with taking homogeneous parts. For example $\mathrm{H}_1(x_1^2) = 0$ and thus $\varphi(\mathrm{H}_1(x_1^2)) = 0$. But $\mathrm{H}_1(\varphi(x_1^2)) = \mathrm{H}_1(x_1^2 + 2\alpha_1 x_1 + \alpha_1^2) = 2\alpha_1 x_1 \neq 0$.

Proof:

$$\begin{aligned}
\textstyle\sum_{0\le k\le d}(1+x)^i &= \frac{(1+x)^{d+1}-1}{(1+x)-1} \\
&= {}^1\!/_x \cdot \left(-1 + \textstyle\sum_{0\le k\le d+1}\binom{d+1}{k}x^k\right) \\
&= {}^1\!/_x \cdot \left(\textstyle\sum_{1\le k\le d+1}\binom{d+1}{k}x^k\right) \\
&= {}^1\!/_x \cdot \left(\textstyle\sum_{0\le k\le d}\binom{d+1}{k+1}x^{k+1}\right) \\
&= \textstyle\sum_{0\le k\le d}\binom{d+1}{k+1}x^k\ . \qquad \square
\end{aligned}$$

We now combine the above to show how to compute $g = h/f$ without division in general, paying careful attention to the complexity of the resulting computation.

Corollary 7.9. *Let $f, g \in \mathbb{F}[\overline{x}]$ be of degree $\le d$ and suppose that $f|g$. Suppose $f(\overline{\alpha}) \neq 0$. Let $S \subseteq \mathbb{F}$ be a $\mathsf{poly}(d)$-explicit set with $|S| = 2d^2+1$. Then there are $\mathsf{poly}(d)$-explicit constants $\{\eta_{\beta,k}\}_{\beta\in S, 0\le k\le d}$ computable from S and $f(\overline{\alpha})$, such that*

$$\frac{g(\overline{x})}{f(\overline{x})} = \sum_{\beta\in S} g(\beta\cdot\overline{x} + (1-\beta)\cdot\overline{\alpha}) \sum_{0\le k\le d} \eta_{\beta,k}\cdot f(\beta\cdot\overline{x} + (1-\beta)\cdot\overline{\alpha})^k\ ,$$

where $\beta\cdot\overline{x}$ scalar multiplication of β on the vector $\overline{x}$ and likewise for $(1-\beta)\cdot\overline{\alpha}$.

Proof: Let $h(\overline{x}) := g(\overline{x})/f(\overline{x}) \in \mathbb{F}[\overline{x}]$ so that $f(\overline{x})h(\overline{x}) = g(\overline{x})$. Then by considering the substitution homomorphism $\varphi : \mathbb{F}[\overline{x}] \to \mathbb{F}[\overline{x}]$ induced by $\overline{x} \mapsto \overline{x} + \overline{\alpha}$ we see that $f(\overline{x}+\overline{\alpha})h(\overline{x}+\overline{\alpha}) = g(\overline{x}+\overline{\alpha})$ so that

$$h(\overline{x}+\overline{\alpha}) = \frac{g(\overline{x}+\overline{\alpha})f(\overline{\alpha})^{-1}}{f(\overline{x}+\overline{\alpha})f(\overline{\alpha})^{-1}}\ .$$

Define $\hat{g}(\overline{x}) := g(\overline{x}+\overline{\alpha})f(\overline{\alpha})^{-1}$ and $\hat{f}(\overline{x}) := f(\overline{x}+\overline{\alpha})f(\overline{\alpha})^{-1}$, so we see that $\hat{f}(\overline{0}) = f(\overline{\alpha})f(\overline{\alpha})^{-1} = 1$ and $h(\overline{x}+\overline{\alpha}) = {}^{\hat{g}(\overline{x})}\!/_{\hat{f}(\overline{x})}$. Thus, appealing to removal of divisions for normalized polynomials (Lemma 7.5),

$$\begin{aligned}
h(\overline{x}+\overline{\alpha}) &= \sum_{i=0}^{d} \mathrm{H}_i\left(\hat{g}(\overline{x})\cdot \textstyle\sum_{j=0}^{d}(1-\hat{f}(\overline{x}))^j\right) \\
&= \sum_i \mathrm{H}_i\left(g(\overline{x}+\overline{\alpha})f(\overline{\alpha})^{-1}\cdot \textstyle\sum_j(1-f(\overline{x}+\overline{\alpha})f(\overline{\alpha})^{-1})^j\right)
\end{aligned}$$

appealing to the above technical lemma (Lemma 7.8),

$$\begin{aligned}
&= \sum_i \mathrm{H}_i\left(g(\overline{x}+\overline{\alpha})f(\overline{\alpha})^{-1}\cdot \textstyle\sum_{0\le k\le d}\dbinom{d+1}{k+1}(-1)^k f(\overline{\alpha})^{-k} f(\overline{x}+\overline{\alpha})^k\right) \\
&= \sum_i \mathrm{H}_i\left(\textstyle\sum_{0\le k\le d}\dbinom{d+1}{k+1}(-1)^k f(\overline{\alpha})^{-k-1}\cdot g(\overline{x}+\overline{\alpha})f(\overline{x}+\overline{\alpha})^k\right)
\end{aligned}$$

as $\deg g, \deg f \le d$, we are taking homogeneous parts of a degree $\le d + d^2 \le 2d^2$ polynomial, so by interpolation (Lemma 7.7) there are $\mathsf{poly}(d)$ explicit constants $\gamma_{i,\beta}$,

$$= \sum_{\beta \in S} (\textstyle\sum_i \gamma_{i,\beta}) \left(\textstyle\sum_{0 \le k \le d} \dbinom{d+1}{k+1} (-1)^k f(\overline{\alpha})^{-k-1} \cdot g(\beta \cdot \overline{x} + \overline{\alpha}) f(\beta \cdot \overline{x} + \overline{\alpha})^k \right)$$

$$= \sum_{\beta \in S} g(\beta \cdot \overline{x} + \overline{\alpha}) \sum_{0 \le k \le d} \left((\textstyle\sum_i \gamma_{i,\beta}) \dbinom{d+1}{k+1} (-1)^k f(\overline{\alpha})^{-k-1} \right) \cdot f(\beta \cdot \overline{x} + \overline{\alpha})^k$$

$$= \sum_{\beta \in S} g(\beta \cdot \overline{x} + \overline{\alpha}) \sum_{0 \le k \le d} \eta_{\beta,k} \cdot f(\beta \cdot \overline{x} + \overline{\alpha})^k \ ,$$

where we define $\eta_{\beta,k} := (\sum_{0 \le i \le d} \gamma_{i,\beta}) \binom{d+1}{k+1} (-1)^k f(\overline{\alpha})^{-k-1}$. It is straightforward to see that the $\{\eta_{\beta,k}\}$ have the desired explicitness given S and $f(\overline{\alpha})$, given the explicitness of the $\{\gamma_{i,\beta}\}$. Thus, the above computes $h(\overline{x} + \overline{\alpha})$. To compute $h(\overline{x})$ we then translate by $-\overline{\alpha}$ obtaining the result. □

We now use this removal of divisions to reduce divisibility testing to polynomial identity testing. The intuition of this result is that Strassen's [Str73] removal of divisions gives an algebraic circuit for g/f if $f|g$. The main insight is that we can still run the Strassen's [Str73] procedure even if $f \nmid g$. In this case, the procedure will produce *some* polynomial $\widetilde{g/f}$ computable by a small circuit. We can then check that this candidate polynomial is indeed the division, that is, to check that $g = \widetilde{g/f} \cdot f$. This check is exactly something solvable by polynomial identity testing.

Put another way, for polynomials f, g computable by small algebraic circuits, Strassen's [Str73] results show that if $f|g$ then there is a small witness for this, namely the (slightly larger) algebraic circuit for g/f. We observe here that this witness is efficiently checkable given an algorithm for polynomial identity testing, effectively putting divisibility testing in $\mathsf{NP}^{\mathsf{PIT}}$. As Strassen [Str73] showed that this witness is also efficiently constructible, then this essentially puts divisibility testing in $\mathsf{P}^{\mathsf{PIT}} \subseteq \mathsf{BPP}$. Note that while PIT has randomized algorithms with one-sided error, this reduction will only yield two-sided error as discussed below in Remark 7.12.

Corollary 7.10. *Let $f, g \in \mathbb{F}[\overline{x}]$ be of degree $\le d$. Suppose $f(\overline{\alpha}) \ne 0$. Let $S \subseteq \mathbb{F}$ be a $\mathsf{poly}(d)$-explicit set with $|S| = 2d^2 + 1$. Then there are $\mathsf{poly}(d)$-explicit constants $\{\eta_{\beta,k}\}_{\beta \in S, 0 \le k \le d}$ computable from S and $f(\overline{\alpha})$ such that*

$$f(\overline{x}) \text{ divides } g(\overline{x})$$

iff

$$g(\overline{x} + \overline{\alpha}) - f(\overline{x} + \overline{\alpha}) \sum_{\beta \in S} g(\beta \cdot \overline{x} + \overline{\alpha}) \sum_{0 \le k \le d} \eta_{\beta,k} \cdot f(\beta \cdot \overline{x} + \overline{\alpha})^k = 0 \ .$$

Proof: Let $h(\overline{x}) := \sum_{\beta \in S} g(\beta \cdot \overline{x} + \overline{\alpha}) \sum_{0 \le k \le d} \eta_{\beta,k} \cdot f(\beta \cdot \overline{x} + \overline{\alpha})^k$. While Corollary 7.9 works with $h(\overline{x} - \overline{\alpha})$, we will work with $h(\overline{x})$ here as divisibility is invariant under translation (Lemma 7.6).

$\underline{\Longrightarrow:}$ If $f(\overline{x})|g(\overline{x})$ then Corollary 7.9 shows that $g(\overline{x}+\overline{\alpha})/f(\overline{x}+\overline{\alpha}) = h(\overline{x})$, so that $g(\overline{x} + \overline{\alpha}) - f(\overline{x} + \overline{\alpha})h(\overline{x}) = 0$ as desired.

$\underline{\Longleftarrow:}$ If $g(\overline{x} + \overline{\alpha}) - f(\overline{x} + \overline{\alpha})h(\overline{x}) = 0$ then $f(\overline{x} + \overline{\alpha})h(\overline{x}) = g(\overline{x} + \overline{\alpha})$ so then $f(\overline{x} + \overline{\alpha})|g(\overline{x} + \overline{\alpha})$, and thus $f(\overline{x})|g(\overline{x})$. □

We now state our reduction in more general terms.

Corollary 7.11. *Let $\mathbb{F}$ be a field with $|\mathbb{F}| \ge 2d^2 + 1$. Let $\mathcal{C}_1, \mathcal{C}_2 \subseteq \mathbb{F}[x_1, \ldots, x_n]$ be two classes of n-variate degree-($\le d$) polynomials, both of which contain the scalars 0 and 1. For $i \in [2]$, define $\mathcal{D}_i \subseteq \mathbb{F}[\overline{x}]$ to the class of polynomials formed from $\mathcal{C}_i$ by re-weighting and translation, that is,*

$$\mathcal{D}_i := \{ f(\beta \cdot \overline{x} + \overline{\alpha}) \mid \beta \in \mathbb{F}, \overline{\alpha} \in \mathbb{F}^n, f \in \mathcal{C}_i \} \ .$$

Then testing divisibility of $\mathcal{C}_1$ polynomials into $\mathcal{C}_2$ polynomials is efficiently reducible to polynomial identity testing of $\sum^{\mathsf{poly}(d)} \mathcal{D}_1 \cdot \mathcal{D}_2 \cdot \bigwedge^{\mathsf{poly}(d)} \mathcal{D}_1$ polynomials, where

$$\sum^{\mathsf{poly}(d)} \mathcal{D}_1 \cdot \mathcal{D}_2 \cdot \bigwedge^{\mathsf{poly}(d)} \mathcal{D}_1 := \left\{ \sum_{i=1}^{s} \alpha_i \cdot f_i(\overline{x}) \cdot g_i(\overline{x}) \cdot h_i(\overline{x})^{d_i} \mid f_i, h_i \in \mathcal{D}_1, g_i \in \mathcal{D}_2, s, d_i \le \mathsf{poly}(d) \right\} .$$

That is, we have the following three versions of the above reduction.

1. ***Deterministic Black-Box Divisibility Testing:** Suppose $f(\overline{x}) \in \mathcal{C}_1$ and $g(\overline{x}) \in \mathcal{C}_2$ are given via black-box evaluation access (that is, a polynomial $p(\overline{x})$ is given by access to the evaluation function $\overline{\gamma} \mapsto p(\overline{\gamma})$). Then given a hitting set $\mathcal{H}$ for $\sum^{\mathsf{poly}(d)} \mathcal{D}_1 \cdot \mathcal{D}_2 \cdot \bigwedge^{\mathsf{poly}(d)} \mathcal{D}_1$, one can decide whether $f|g$ in $\mathsf{poly}(|\mathcal{H}|, n, d)$ steps.*

2. ***Randomized Black-Box Divisibility Testing:** Suppose $|\mathbb{F}| \ge 18d^2$. Suppose $f(\overline{x}) \in \mathbb{F}[\overline{x}]$ and $g(\overline{x}) \in \mathbb{F}[\overline{x}]$ are arbitrary n-variate degree-($\le d$) polynomials given via black-box evaluation access. Then one can decide whether $f|g$ in $\mathsf{poly}(n, d)$ steps with two-sided error $\le 1/2$.*

3. ***Deterministic White-Box Divisibility Testing:** Suppose $\mathcal{C}_1$ is a class of polynomials arising from a (possible restricted) class of algebraic circuits with size $\le s$, and likewise for $\mathcal{C}_2$, where the size is measured with respect to general algebraic circuits. Then let $\mathcal{D}_1$, $\mathcal{D}_2$ and $\sum^{\mathsf{poly}(d)} \mathcal{D}_1 \cdot \mathcal{D}_2 \cdot \bigwedge^{\mathsf{poly}(d)} \mathcal{D}_1$ denote not just the polynomials but also the class of (possible restricted) algebraic circuits arising from the appropriate operations. Then $\sum^{\mathsf{poly}(d)} \mathcal{D}_1 \cdot \mathcal{D}_2 \cdot \bigwedge^{\mathsf{poly}(d)} \mathcal{D}_1$ has size $\le \mathsf{poly}(s, n, d)$ algebraic circuits.*

   Further, now suppose $f(\overline{x}) \in \mathcal{C}_1$ and $g(\overline{x}) \in \mathcal{C}_2$ are given by their (possibly restricted) algebraic circuits of size $\le s$. Then given a $t(s, n, d)$-step white-box polynomial identity testing for the algebraic circuits from $\sum^{\mathsf{poly}(d)} \mathcal{D}_1 \cdot \mathcal{D}_2 \cdot \bigwedge^{\mathsf{poly}(d)} \mathcal{D}_1$, one can decide whether $f|g$ in $\mathsf{poly}(t(s, n, d), s, n, d)$-steps.

Proof: Consider Algorithm 1. First note that it correctly decides whether $f|g$. That is, when $f = 0$, a polynomial g, $0|g$ iff $g = 0$. When $f \ne 0$, the correctness then follows from Corollary 7.10. It remains to show that this algorithm can be efficiently implemented in the black- and white-box models. For notational simplicity, let $\mathcal{D}$ denote the class $\sum^{\mathsf{poly}(d)} \mathcal{D}_1 \cdot \mathcal{D}_2 \cdot \bigwedge^{\mathsf{poly}(d)} \mathcal{D}_1$. Note that by our hypothesis on $\mathcal{C}_1, \mathcal{C}_2$, we see that $\mathcal{C}_1, \mathcal{C}_2 \subseteq \mathcal{D}$.

deterministic black-box: We show here that Algorithm 1 can be deterministically implemented given a hitting set $\mathcal{H}$ for $\mathcal{D}$. First observe that the corner case of when $f = 0$ (Line (2)) can be decided using black-box access to f, g in $\mathsf{poly}(|\mathcal{H}|)$ steps, as they both belong to $\mathcal{D}$, so that $f = 0$ iff $f|_{\mathcal{H}} = 0$ (and likewise for g). Thus, now consider $f \ne 0$. In this case, $f|_{\mathcal{H}} \not\equiv 0$ so that there is some $\overline{\alpha} \in \mathcal{H}$ where $f(\overline{\alpha}) \ne 0$, giving the desired $\overline{\alpha}$ for Line (9), from which one can compute $f(\overline{\alpha})$ in constant time using access to f. Using an enumeration of the field $\mathbb{F}$ one can in $\mathsf{poly}(d)$ steps find an explicit set $S \subseteq \mathbb{F}$ with $|S| = 2d^2 + 1$. From this, one can in $\mathsf{poly}(d)$ steps construct the needed constants $\{\eta_{\beta,k}\}$ needed to form the polynomial $h(\overline{x})$ in Line (10) from Algorithm 1. Note that by definition of h, for any $\overline{\gamma} \in \mathbb{F}^n$ we can compute $h(\overline{\gamma})$ in $\mathsf{poly}(n, d)$-steps using black-box evaluations to f and g, so that we have efficient black-box access to h. In particular, we can compute $h|_{\mathcal{H}}$ in $\mathsf{poly}(|\mathcal{H}|, n, d)$-steps. As h is in $\mathcal{D}$ by construction, $h = 0$ iff $h|_{\mathcal{H}} = 0$ and thus we can decide whether $h = 0$. Thus, each step of this algorithm can be implemented in $\mathsf{poly}(|\mathcal{H}|, n, d)$ steps given black-box access to f and g.

randomized black-box: The efficiency of this algorithm is clear from the above analysis in the deterministic black-box case. That is, one replaces the hitting set $\mathcal{H}$ with a black-box randomized polynomial time algorithm such as the one based on the Schwartz-Zippel [Sch80, Zip79, DL78] lemma (which shows that for a degree-($\le d$) polynomial $p \neq 0$ that $\Pr_{\overline{\alpha}\in T^n}[p(\overline{\alpha}) = 0] \le {}^{d}/_{|T|}$). This algorithm has the benefit of finding a non-root $f(\overline{\alpha}) \neq 0$ whenever it declares $f \neq 0$, so this satisfies the needs of Line (9).

To analyze the error, note that the polynomials manipulating in Algorithm 1 (that is, f, g and h) are all of degree $\le d^2 + 2d \le 3d^2$, and that at most 3 such identity tests are ever performed. Thus, in $\mathsf{poly}(d)$-steps one can construct an explicit set $T \subseteq \mathbb{F}$ of size $18d^2$ and the probability of failure of any of these tests is at most $3 \cdot {}^{3d^2}/_{18d^2} \le {}^{1}/_{2}$. Thus, one sees that Algorithm 1 has two-sided error at most ${}^{1}/_{2}$.

deterministic white-box: The algebraic circuit size of polynomials in $\mathcal{D}$ follows immediately from their definition as they are formed from small computations arising from (re-weightings and translations of) circuits from $\mathcal{C}_1$ and $\mathcal{C}_2$.

Now observe that given such a white-box identity testing algorithm for $\mathcal{D}$, for $0 \neq f \in \mathcal{D}$ one can find a non-root $f(\overline{\alpha}) \neq 0$ in $\mathsf{poly}(n, d)$ steps. That is, take $T \subseteq \mathbb{F}$ to be an explicit set of size $d + 1$, which is computable in $\mathsf{poly}(d)$ steps. Interpolation then shows that $f \neq 0$ is equivalent to $f|_{T^n} \neq 0$, so that we can search for $\overline{\alpha}$ in T^n. Now note that this is possible via a standard search to decision reduction. That is, as $\deg f \le d$ it follows there is some $\alpha \in T$ such that $f(\alpha, \overline{x} \setminus \{x_1\}) \neq 0$, and this α can be found using $|T|$ calls to the white-box identity test (as substituting constants for variables does not increase the circuit size to more than $\mathsf{poly}(s, n)$). One can then set $(\overline{\alpha})_1 := \alpha$ to obtain a non-zero polynomial $f_1(\overline{x} \setminus \{x_1\} := f(\alpha_1, \overline{x} \setminus \{x_1\})$ on one fewer variable, and then one can recurse. In total, this takes $\mathsf{poly}(n, d)$ steps to find the desired $\overline{\alpha}$ with $f(\overline{\alpha}) \neq 0$.

Now consider Algorithm 1. As $\mathcal{C}_1, \mathcal{C}_2 \subseteq \mathcal{D}$, the identity tests for f and g can be implemented as desired. The above discussion allows Line (9) allows to be implemented efficiently. Finally, as $h(\overline{x}) \in \mathcal{D}$ has a small circuit that can be efficiently constructed from f and g, we can decide whether $h = 0$ using the provided white-box algorithm. Thus, Algorithm 1 can be implemented as desired yielding the decision procedure. □

Remark 7.12. Note that it is not clear that one can decide divisibility testing in the black-box model with an efficient randomized algorithm with one-sided error (in contrast to polynomial identity testing, which has such an algorithm). The main difficulty comes from the fact that natural algorithms for deciding whether $f|g$ (such as Algorithm 1) must first decide whether $f = 0$. While this question can be done with "one-sided error" using the Schwartz-Zippel [Sch80, Zip79, DL78] lemma, the "side" of this error is not a function of whether $f|g$ alone. That is, consider the following two error cases. Suppose that $f|g$ and $f, g \neq 0$, so that the one-sided identity test yields "$f = 0$, $g \neq 0$", and thus "$f \nmid g$", with some non-zero probability. In contrast, suppose $f \nmid g$ where $f = 0$ and $g \neq 0$. Then the one-sided identity test yields "$f = 0$, $g = 0$", and thus "$f|g$", with some non-zero probability. As these two cases show, the "trivial" corner case of when $f = 0$ causes a two-sided error to arise from the one-sided error of the identity test. If we forbid $f = 0$ then one can see that Algorithm 1 is an algorithm with one-sided error. ♢

7.3 Testing Divisibility of $\sum\prod^{\mathcal{O}(1)}$ into $\sum \mathrm{m} \bigwedge \sum\prod^{\mathcal{O}(1)}$

In the above we gave a general reduction from divisibility testing to polynomial identity testing. In this section we instantiate this reduction with the concrete problem of deciding whether a constant degree polynomial divides a sparse polynomial or more generally a $\sum \mathrm{m} \bigwedge \sum\prod^{\mathcal{O}(1)}$ polynomial. Applying this reduction carefully we see that PIT of $\sum \mathrm{m} \bigwedge \sum\prod^{\mathcal{O}(1)}$ naturally suffices for this application. Plugging in our hitting sets for this class yields the desired divisibility testing algorithms.

We begin with attempting to understand why a direct application of the reduction of Corollary 7.10 does not suffice. That is, suppose we have f and g where $\deg f \le 2$ and g is s-sparse

Algorithm 1 Reducing Divisibility Testing to PIT

```
procedure DIVISIBILITY($f \in \mathcal{C}_1$, $g \in \mathcal{C}_2$)
    if $f = 0$ then
        if $g = 0$ then
            return “$f|g$”
        else
            return “$f \nmid g$”
        end if
    else
        Find $\overline{\alpha}$ with $f(\overline{\alpha}) \neq 0$.
        Construct $h(\overline{x}) := g(\overline{x}+\overline{\alpha}) - f(\overline{x}+\overline{\alpha}) \sum_{\beta \in S} g(\beta \cdot \overline{x}+\overline{\alpha}) \sum_{0 \leq k \leq d} \eta_{\beta,k} \cdot f(\beta \cdot \overline{x}+\overline{\alpha})^k$ as in Corollary 7.10.
        if $h = 0$ then
            return “$f|g$”
        else
            return “$f \nmid g$”
        end if
    end if
end procedure
```

and we wish to decide whether $f|g$. The reduction of Corollary 7.10 would (at least) require PIT of expressions such as $\sum_i g(\beta_i \cdot \overline{x} + \overline{\alpha}) f(\beta_i \cdot \overline{x} + \overline{\alpha})^{d_i}$. If the β_i were not present (or were even independent of i), then this would be (up to a translation by $\overline{\alpha}$) a single $\sum \mathrm{m} \bigwedge \sum \prod^2$ formula. The difficulty is that the β_i *depends* on i, so that we have a large sum of translated $\sum \mathrm{m} \bigwedge \sum \prod^2$ formula. As there are many different translations, the methods of Section 6 do not yield good PIT algorithms[13]. As such, we cannot naively apply Corollary 7.10 to our PIT results to obtain divisibility algorithms.

Instead, we further study the situation at hand. In particular, in Corollary 7.10 we are free to choose $\overline{\alpha}$ subject to the condition $f(\overline{\alpha}) \neq 0$. As such, we will choose the point $\overline{\alpha}$ where $f(\overline{\alpha}) \neq 0$ to be a *simple* point in that it is sparse. That is, appealing to Lemma 3.14 we have the following lemma.

Lemma 7.13. *Let $f(\overline{x}) \in \mathbb{F}[x_1, \ldots, x_n]$ have $\deg f \leq t$. If $f \neq 0$ then there is an $\overline{\alpha} \in \mathbb{F}^n$ with $f(\overline{\alpha}) \neq 0$ and $\|\overline{\alpha}\|_0 \leq t$.* □

For such sparse non-roots of f, we can see that their effect on sparse polynomials g is somewhat benign.

Lemma 7.14. *Let $\overline{x}^{\overline{a}} \in \mathbb{F}[x_1, \ldots, x_n]$ be degree-($\leq d$). Then for $\overline{\alpha} \in \mathbb{F}^n$, $(\overline{x}+\overline{\alpha})^{\overline{a}}$ has sparsity $\leq \binom{\|\overline{\alpha}\|_0+d}{d}$.*

[13] As we will describe in a future version of this work, these methods can handle the sum of a constant number of different translations, but this constant goes into the exponent of the runtime of the algorithms.

Proof:

$$\begin{aligned}
(\overline{x}+\overline{\alpha})^{\overline{a}} &= \prod_{i\in[n]} (x_i+\alpha_i)^{a_i} \\
&= \prod_{i\in\mathrm{Supp}(\overline{\alpha})} (x_i+\alpha_i)^{a_i} \prod_{i\notin\mathrm{Supp}(\overline{\alpha})} (x_i+\alpha_i)^{a_i} \\
&= \prod_{i\in\mathrm{Supp}(\overline{\alpha})} (x_i+\alpha_i)^{a_i} \prod_{i\notin\mathrm{Supp}(\overline{\alpha})} x_i)^{a_i} \ .
\end{aligned}$$

Now note that $\prod_{i\in\mathrm{Supp}(\overline{\alpha})}(x_i+\alpha_i)^{a_i}$ is a degree $\le d$ polynomial in $\|\overline{\alpha}\|_0$ variables, and thus has sparsity $\le \binom{\|\overline{\alpha}\|_0+d}{d}$. Multiplication by the monomial $\prod_{i\notin\mathrm{Supp}(\overline{\alpha})} x_i)^{a_i}$ does not increase the sparsity, giving the claim. □

We now show how to apply such sparse non-roots to our reduction of divisibility to PIT.

Lemma 7.15. *Let $\mathbb{F}$ be a field with $S\subseteq\mathbb{F}$ and $|S|=2d^2+1$. Let $f(\overline{x})\in\mathbb{F}[x_1,\ldots,x_n]$ and $a\ge 1$ where* $\deg f\le t$ *and* $\deg f^a\le d$ *. Let $g(\overline{x})\in\mathbb{F}[\overline{x}]$, where $g(\overline{x})=\sum_{i=1}^s \overline{x}^{\overline{b}_i} g_i(\overline{x})^{d_i}$ for $\deg g_i\le r$ and $\deg \overline{x}^{\overline{b}_i} g_i(\overline{x})^{d_i}\le d$ for all i. Suppose $\overline{\alpha}\in\mathbb{F}^n$ has $f(\overline{\alpha})\neq 0$, so that $f(\overline{\alpha})^a\neq 0$, and define*

$$h(\overline{x}+\overline{\alpha}) := g(\overline{x}+\overline{\alpha}) - f(\overline{x}+\overline{\alpha})^a \sum_{\beta\in S} g(\beta\cdot\overline{x}+\overline{\alpha}) \sum_{0\le k\le d} \eta_{\beta,k}\cdot f(\beta\cdot\overline{x}+\overline{\alpha})^{a\cdot k} \ ,$$

as given by applying Corollary 7.9 to the divisibility of $f(\overline{x})^a$ into $g(\overline{x})$.

If $\|\overline{\alpha}\|_0\le t$ then $h(\overline{x}+\overline{\alpha})$ can be expressed as a $\mathsf{poly}(s,n,d,\binom{n+r}{r},\binom{n+t}{t})$*-explicit expression of the form*

$$h(\overline{x}) = \sum_{i=1}^{s'} \overline{x}^{\overline{a}_i} h_i(h_{i,1}(\overline{x}), h_{i,2}(\overline{x}), h_{i,3}(\overline{x})) \ ,$$

where $s'\le s(2d^2+2)\binom{n+t}{t}$, $\deg h_{i,1}\le r$ *and* $\deg h_{i,2},\deg h_{i,3}\le t$*.*

Proof: <u>form of h:</u>

$$\begin{aligned}
f(\overline{x}+\overline{\alpha})^a &\sum_{\beta\in S} g(\beta\cdot\overline{x}+\overline{\alpha}) \sum_{0\le k\le d} \eta_{\beta,k}\cdot f(\beta\cdot\overline{x}+\overline{\alpha})^{a\cdot k} \\
&= f(\overline{x}+\overline{\alpha})^a \sum_{\beta\in S}\left(\sum_{i=1}^{s} (\beta\cdot\overline{x}+\overline{\alpha})^{\overline{b}_i} g_i(\beta\cdot\overline{x}+\overline{\alpha})^{d_i}\right) \sum_{0\le k\le d} \eta_{\beta,k}\cdot f(\beta\cdot\overline{x}+\overline{\alpha})^{ak} \\
&= \sum_{\beta\in S}\sum_{i=1}^{s} (\beta\cdot\overline{x}+\overline{\alpha})^{\overline{b}_i} g_i(\beta\cdot\overline{x}+\overline{\alpha})^{d_i} f(\overline{x}+\overline{\alpha})^a \sum_{0\le k\le d} \eta_{\beta,k}\cdot f(\beta\cdot\overline{x}+\overline{\alpha})^{ak}
\end{aligned}$$

As $\|\overline{\alpha}\|_0\le t$, Lemma 7.14 shows that $(\overline{x}+\overline{\alpha})^{\overline{b}_i}$ is $(\le\binom{d+t}{t})$-sparse, from which it follows that $(\beta\cdot\overline{x}+\overline{\alpha})^{\overline{b}_i}$ is similarly sparse, so that $(\beta\cdot\overline{x}+\overline{\alpha})^{\overline{b}_i} = \sum_{j=1}^{\binom{n+t}{t}} \gamma_{\beta,i,j}\overline{x}^{\overline{c}_{\beta,i,j}}$ where $\deg \overline{x}^{\overline{c}_{\beta,i,j}}\le d$ and we take some $\gamma_{\beta,i,j}$ so we pad the sparsity to exactly $\binom{n+t}{t}$. As such we can rewrite the above as

$$= \sum_{\beta\in S}\sum_{i=1}^{s}\left(\sum_{j=1}^{\binom{n+t}{t}} \gamma_{\beta,i,j}\overline{x}^{\overline{c}_{\beta,i,j}}\right)\cdot g_i(\beta\cdot\overline{x}+\overline{\alpha})^{d_i} f(\overline{x}+\overline{\alpha})^a \sum_{0\le k\le d} \eta_{\beta,k}\cdot f(\beta\cdot\overline{x}+\overline{\alpha})^{ak}$$

Define $p_{\beta,i,j}$ by

$$p_{\beta,i,j}(y_1,y_2,y_3) := \gamma_{\beta,i,j} \cdot y_1^{d_i} \cdot y_2^{a} \cdot \sum_{0\le k\le d} \eta_{\beta,k} \cdot y_3^{ak} \ .$$

so that we can rewrite the above as

$$= \sum_{\beta\in S} \sum_{i=1}^{s} \sum_{j=1}^{\binom{n+t}{t}} \overline{x}^{\overline{c}_{\beta,i,j}} p_{\beta,i,j}\Big(g_i(\beta\cdot\overline{x}+\overline{\alpha}), f(\overline{x}+\overline{\alpha}), f(\beta\cdot\overline{x}+\overline{\alpha})\Big) \ .$$

Thus, this expression is of the form we wish for h, where in this expression we sum over $s(2d^2+1)\binom{n+t}{t}$ polynomials. Note now that the expression computed is equal to the difference of $h(\overline{x}+\overline{\alpha})$ and $g(\overline{x}+\overline{\alpha})$. As $g(\overline{x}+\overline{\alpha}) = \sum_{i=1}^{s}(\overline{x}+\overline{\alpha})^{\overline{b}_i} g_i(\overline{x}+\overline{\alpha})^{d_i}$, the above argument shows that $g(\overline{x}+\overline{\alpha})$ also has same form that we desire for h, but with top-fan-in $s\binom{n+t}{t}$. Thus, combining these expressions we obtain the desired form for h with top-fan-in $s(2d^2+1)\binom{n+t}{t} + s\binom{n+t}{t} = s(2d^2+2)\binom{n+t}{t}$.

explicitness: Note that size of the g_i is $\binom{n+r}{r}$, and likewise the size of f is $\binom{n+t}{t}$. That the above expression is explicit with the above time bound is then clear from construction. □

We now arrive at our specialized reduction from testing if a low-degree polynomial divides a $\sum \mathrm{m} \bigwedge \sum \prod^{\mathcal{O}(1)}$ formula to PIT.

Corollary 7.16. *Let $\mathbb{F}$ be a field with $S \subseteq \mathbb{F}$ and $|S| = 2d^2+1$. Let $f(\overline{x}) \in \mathbb{F}[x_1,\ldots,x_n]$ where $\deg f \le t \le d$. Let $g(\overline{x}) \in \mathbb{F}[\overline{x}]$, where $g(\overline{x}) = \sum_{i=1}^{s} \overline{x}^{\overline{b}_i} g_i(\overline{x})^{d_i}$ for $\deg g_i \le r$ and $\deg g(\overline{x}) \le d$. Then computing the maximum a such that $f^a | g$ is deterministically reducible to $\mathcal{O}(d)$ instances of PIT of formulas of the form*

$$h(\overline{x}) = \sum_{i=1}^{s'} \overline{x}^{\overline{a}_i} h_i(h_{i,1}(\overline{x}), h_{i,2}(\overline{x}), h_{i,3}(\overline{x})) \ ,$$

where $s' \le s(2d^2+2)\binom{n+t}{t}$, $\deg h_{i,1} \le r$ and $\deg h_{i,2}, \deg h_{i,3} \le t$. Further, this reduction is computable in $\mathsf{poly}(s,n,d,(n(r+1))^r,(n(t+1))^t)$-time in both the black-box and white-box model.

Proof: We implement Algorithm 1 with certain implementation optimizations. If $f = 0$ then $f^a|g$ iff $g = 0$. These identity tests are of the above form.

Now consider $f \ne 0$. We can find a t-sparse non-root of f in $(n(t+1))^t$ back-box evaluations to f as $\deg f \le t$ (Corollary 3.15). Now consider some a such that we wish to test whether $f^a|g$. As $f \ne 0$, $\deg f \ge 0$. We can in $\mathsf{poly}(n(t+1)^t)$ black-box evaluations determine $\deg f$. If $\deg f = 0$ then $f^a|g$ always. If $\deg f \ge 1$ then $f^a \nmid g$ for a where $\deg f^a > d$. In particular this means we only need to consider $a \le d$.

Now consider whether $f^a|g$. By Corollary 7.10 and Lemma 7.15 we see this reduces to a PIT of a polynomial of the desired form in the desired runtime.

Thus, computing the divisibility for each a we then return the maximum. □

Plugging in our PIT for $\sum \mathrm{m} \bigwedge \sum \prod^{\mathcal{O}(1)}$ (taking $m = 3$ in Proposition 6.5 and applying Corollary 3.15), we obtain the desired deterministic divisibility algorithm.

Corollary 7.17. *Let $\mathbb{F}$ be a field $\mathrm{char}(\mathbb{F}) > d$ and $|\mathbb{F}| \ge 2d^2+1$. Let $f(\overline{x}) \in \mathbb{F}[x_1,\ldots,x_n]$ where $\deg f \le t \le d$. Let $g(\overline{x}) \in \mathbb{F}[\overline{x}]$, where $g(\overline{x}) = \sum_{i=1}^{s} \overline{x}^{\overline{b}_i} g_i(\overline{x})^{d_i}$ for $\deg g_i \le r$ and $\deg g(\overline{x}) \le d$. Then computing the multiplicity of f as a factor of g can be deterministically computed in $\mathsf{poly}(s,n,d)^{\mathcal{O}(\max\{r,t\} \lg s)}$-time in the black-box model.* □

8 Comparing $\sum\bigwedge\sum\prod^2$ Formulas to roABPs

The previous sections of this paper have considered, for $t = \mathcal{O}(1)$, $\sum\bigwedge\sum\prod^t$ formulas (Section 4), sparse polynomials (Section 5), and $\sum \mathrm{m}\bigwedge\sum\prod^t$ formulas (Section 6). For $t = 1$ all of these models are computable by small roABPs (Lemma 3.18). It is natural to ask for $t > 1$ how this relation degrades.

In this section we address this question. As in previous sections we showed that $\sum \mathrm{m}\bigwedge\sum\prod^{\mathcal{O}(1)}$ formulas cannot compute large monomials in the $\overline{x} + \overline{1}$ basis, and this monomials have width-1 roABPs (Lemma 3.17), this shows $\sum \mathrm{m}\bigwedge\sum\prod^{\mathcal{O}(1)}$ formulas cannot simulate roABPs. More interestingly, after reviewing the basics of lower bounds for roABPs, we show that even $\sum\bigwedge\sum\prod^2$ formulas compute polynomials which require essentially have maximal complexity as roABPs (in any order).

This section has several parts. First, we review methods for proving lower bounds for roABPs. We then give a folklore $\sum\bigwedge\sum\prod^2$ formula that requires large roABPs in *some* variable order, but show that it has *small* roABPs in *another* variable order. We then use indicator variables to embed this example into a $\sum\bigwedge\sum\prod^3$ formula that requires large roABPs in *every* variable order. Finally, we consider a different embedding strategy to obtain a $\sum\bigwedge\sum\prod^2$ formula that requires large roABPs in every variable order.

8.1 Coefficient Dimension and roABPs

In this section we briefly review methods for proving lower bounds for roABPs. These methods are due to Nisan [Nis91] who phrased them in terms of non-commutative ABPs. For the explicit phrasing of these techniques in terms of roABPs see the thesis of Forbes [For14, Chapter 4].

We begin by defining the relevant complexity measure which intuitively measures the amount of "correlation" between the variables $\overline{x}$ and $\overline{y}$ in a polynomial $f(\overline{x}, \overline{y})$, which we refer to as *coefficient dimension* (which is equivalent to the notion of *evaluation dimension* of Saptharishi, see Forbes-Shpilka [FS13b]). This measure has sometimes been called the rank of the *partial derivative matrix*, but we avoid this name to prevent conflict with the space of partial derivatives operator ($\boldsymbol{\partial}_{\overline{x}^{\leq k}}$).

Definition 8.1 (Coefficient Dimension). *Let* $\mathbf{Coeff}_{\overline{y}} : \mathbb{F}[\overline{x}, \overline{y}] \to 2^{\mathbb{F}[\overline{x}]}$ *be the* ***space of*** $\mathbb{F}[\overline{x}][\overline{y}]$ ***coefficients (operator)****, defined by*

$$\mathbf{Coeff}_{\overline{y}}(f) := \left\{\mathrm{Coeff}_{\overline{y}^{\overline{b}}}(f)\right\}_{\overline{b}\in\mathbb{N}^n} ,$$

where coefficients of f are taken in $\mathbb{F}[\overline{x}][\overline{y}]$. Similarly, define $\mathbf{Coeff}_{\overline{x}} : \mathbb{F}[\overline{x}, \overline{y}] \to 2^{\mathbb{F}[\overline{y}]}$ by taking coefficients in $\mathbb{F}[\overline{y}][\overline{x}]$.

Slightly abusing notation, for $f \in \mathbb{F}[\overline{x}, \overline{y}]$ define

$$\dim \mathbf{Coeff}_{\overline{x}|\overline{y}}(f) := \max\{\dim \operatorname{span} \mathbf{Coeff}_{\overline{x}}(f), \dim \operatorname{span} \mathbf{Coeff}_{\overline{y}}(f)\} . \qquad \Diamond$$

In this definition we need not actually take the maximum between the $\overline{x}$-dimension and the $\overline{y}$-dimension, as they are equal.

Lemma 8.2. *For $f(\overline{x}, \overline{y}) \in \mathbb{F}[\overline{x}, \overline{y}]$,*

$$\dim \mathbf{Coeff}_{\overline{x}}(f) = \dim \mathbf{Coeff}_{\overline{y}}(f) = \dim \mathbf{Coeff}_{\overline{x}|\overline{y}}(f) . \qquad \square$$

The following lemma gives the trivial upper bound on coefficient dimension.

Lemma 8.3. *Let $f \in \mathbb{F}[\overline{x}, \overline{y}]$ have degree $\le d$, where $\overline{x}$ has n variables and $\overline{y}$ has m variables. Then*

$$\dim \mathbf{Coeff}_{\overline{x}|\overline{y}}(f) \le \min \left\{ \binom{n+d}{d}, \binom{m+d}{d} \right\} . \qquad \square$$

This next lemma shows that the coefficient dimension characterizes the complexity of roABPs (in a fixed variable order).

Lemma 8.4 (Nisan [Nis91], see also Forbes [For14, Lemma 4.5.8])**.** *Let $f \in \mathbb{F}[x_1, \ldots, x_n]$. For any width-$w$ roABP in variable order $x_1 < \cdots < x_n$,*

$$w \ge \max_i \dim \mathbf{Coeff}_{\overline{x}_{\le i}|\overline{x}_{>i}}(f) .$$

Further, f is computable by a width-w roABP in variable order $x_1 < \cdots < x_n$ where $w = \max_i \dim \mathbf{Coeff}_{\overline{x}_{\le i}|\overline{x}_{>i}}(f)$. □

8.2 Lower Bounds for Computing $\sum\bigwedge\sum\prod^2$ Formulas by roABPs, in Some Order

We now give a polynomial computable as a $\sum\bigwedge\sum\prod^2$ formula (in fact, a $\bigwedge\sum\prod^2$ formula) that has maximal coefficient dimension, a result that seems folklore. The lower bounds for roABPs then follows from Lemma 8.4.

Lemma 8.5 (Folklore)**.** *Let $d \ge 0$ and $\mathbb{F}$ be a field with $\mathrm{char}(\mathbb{F}) > d$. Let $\overline{x}$ and $\overline{y}$ be disjoint sets of n variables. Define $f(\overline{x}, \overline{y}) := (1 + \sum_{i \in [n]} x_i y_i)^d$. Then*

$$\dim \mathbf{Coeff}_{\overline{x}|\overline{y}}(f) = \binom{n+d}{d} .$$

In particular, $f(\overline{x}, \overline{y})$ requires width $\ge \binom{n+d}{d}$ to be computed by a roABP in any variable order where $\overline{x}$ precedes $\overline{y}$.

Proof: $\underline{\le}$: This follows from the trivial upper bound (Lemma 8.3).

$\underline{\ge}$: Consider $\overline{b}$ so $d' := \deg \overline{y}^{\overline{b}} \le d$. Then

$$\mathrm{Coeff}_{\overline{y}^{\overline{b}}} \left(1 + \sum_{i \in [n]} x_i y_i \right)^d = \binom{d}{b_1, \ldots, b_n, d-d'} \overline{x}^{\overline{b}} .$$

where this equality follows from the multinomial theorem, using that

$$\binom{d}{b_1, \ldots, b_n, d-d'} := \frac{d!}{b_1! \cdots b_n! (d-d')!} .$$

As $\mathrm{char}(\mathbb{F}) > d$ this multinomial coefficient is non-zero, so that $\mathrm{span}\, \mathrm{Coeff}_{\overline{y}^{\overline{b}}}(f)$ contains all degree $\le d$ monomials in $\overline{x}$, of which there are $\binom{n+d}{d}$. Thus, $\dim \mathbf{Coeff}_{\overline{x}|\overline{y}} \ge \binom{n+d}{d}$.

<u>roABP lower bound:</u> This follows immediately from applying Lemma 8.4. □

Note that if we removes the "1" in the formula $(1 + \sum_i x_i y_i)^d$ and instead considered $(\sum_i x_i y_i)^d$ we would get qualitatively the same result but with slightly weaker parameters. In particular, we would not match the upper bound of Lemma 8.3.

This result shows that there is a small $\bigwedge\sum\prod^2$ formula that requires a maximal width roABP in a certain order. However, this is somewhat unsatisfying as this formula has a small roABP in a *different* variable order, as we now show using the fact that $\sum\bigwedge\sum$ formulas reduce to roABPs (Lemma 1.3). We also need a lemma about breaking up a matrix $M(xy)$ into the product of a x-matrix and a y-matrix, essentially via brute force.

Lemma 8.6. *Let $C(z) \in \mathbb{F}[z]^{w\times w}$ be of individual degree $< d$. Then there exist matrices $A(x) \in \mathbb{F}[x]^{w\times wd}$ and $B(y) \in \mathbb{F}[y]^{wd\times w}$ of individual degree-($< d$) such that $C(xy) = A(x)B(y)$.*

Proof: Let $C(z) = \sum_{i\in[\![d]\!]} C_i z^i$. Define $A(x) := \sum_{i\in[\![d]\!]} A_i x^i$, where we define $A_i \in \mathbb{F}^{w\times wd}$ via block notation. That is, treating $\mathbb{F}^{w\times wd} = (\mathbb{F}^{w\times w})^{1\times[\![d]\!]}$, we define $A_i \in (\mathbb{F}^{w\times w})^{1\times[\![d]\!]}$ for $k \in [\![d]\!]$ by

$$(A_i)_{1,k} := \begin{cases} C_i & k = i \\ \mathbf{0}_w & \text{else} \end{cases} ,$$

where $\mathbf{0}_w$ is the $w \times w$ zero matrix.

Similarly, define $B(y) := \sum_{j\in[\![d]\!]} B_j y^j$ where we define $B_j \in \mathbb{F}^{wd\times w}$ via block notation. That is treating $\mathbb{F}^{wd\times w} = (\mathbb{F}^{w\times w})^{[\![d]\!]\times 1}$, we define $B_j \in (\mathbb{F}^{w\times w})^{[\![d]\!]\times 1}$ for $k \in [\![d]\!]$ by

$$(B_j)_{k,1} := \begin{cases} \mathrm{I}_w & k = j \\ \mathbf{0}_w & \text{else} \end{cases} ,$$

where I_w is the $w \times w$ identity matrix.

Thus, we see that

$$A_i B_j = \sum_k (A_i)_{1,k}(B_j)_{k,1} = \begin{cases} C_i & i = j \\ \mathbf{0}_w & \text{else} \end{cases} .$$

Thus,

$$\begin{aligned} A(x)B(y) &= \left(\textstyle\sum_{i\in[\![d]\!]} A_i x^i\right)\left(\textstyle\sum_{j\in[\![d]\!]} B_j y^j\right) \\ &= \textstyle\sum_{i,j\in[\![d]\!]} A_i B_j x^i y^j \\ &= \textstyle\sum_{i\in[\![d]\!]} C_i x^i y^i \\ &= C(xy) . \end{aligned}$$

□

We now use these two facts to give an upper bound for the roABP complexity of $(1 + \sum_i x_i y_i)^d$ in a variable order that interleaves $\overline{x}$ and $\overline{y}$.

Lemma 8.7. *Let $d \geq 0$ and $\mathbb{F}$ be an arbitrary field. Let $\overline{x}$ and $\overline{y}$ be disjoint sets of n variables. Define $f(\overline{x}, \overline{y}) := (1 + \sum_{i\in[n]} x_i y_i)^d$. Then in the variable order $x_1 < y_1 < \cdots < x_n < y_n$, $f(\overline{x}, \overline{y})$ can be computed by a roABP with width $(d + 1)^2$.*

Proof: Consider $g(\overline{z}) = (1 + \sum z_i)^d$. By Lemma 1.3, $g(\overline{z})$ has a width-$(d + 1)$ roABP in the variable order $z_1 < \cdots < z_n$. Thus, $g(\overline{z}) = \left(\prod_{i\in[n]} M_i(z_i)\right)_{1,1}$ for matrices $M_i(z_i) \in \mathbb{F}[z_i]^{(d+1)\times(d+1)}$ of individual degree $\leq d$. Now take $\overline{z} = \overline{y} \circ \overline{z}$ where '$\circ$' is the coordinate-wise product so that $z_i = x_i y_i$. Thus $f(\overline{x}, \overline{y}) = g(\overline{x} \circ \overline{y}) = \left(\prod_{i\in[n]} M_i(x_i y_i)\right)_{1,1}$. Breaking up these $M(x_i y_i)$ using brute force (Lemma 8.6) one can find $N_i(x_i) \in \mathbb{F}[x_i]^{(d+1)\times(d+1)^2}$ and $P_i(y_i) \in \mathbb{F}[y_i]^{(d+1)^2\times(d+1)}$ of individual degree-($\leq d$) such that $M_i(x_i y_i) = N_i(x_i)P_i(y_i)$. Thus

$$f(\overline{x}, \overline{y}) = \left(\prod_{i\in[n]} (N_i(x_i)P_i(y_i))\right)_{1,1} = \Big(N_1(x_1)P_1(y_1) \cdots N_n(x_n)P_n(y_n)\Big)_{1,1} .$$

As each matrix in this decomposition has at most $(d + 1)^2$ rows or columns it follows that this is the desired roABP. □

8.3 Lower Bounds for Computing $\sum\bigwedge\sum\prod^3$ Formulas by roABPs, in Any Order

In the previous section we saw a polynomial computable by a $\sum\bigwedge\sum\prod^2$ formula that is maximally hard for roABPs in a certain variable order. However, as we saw there is *another* variable order where this formula is computable by a small roABP. As there are now hitting sets for roABPs that work in an *unknown* order ([FSS14, AGKS14, GKST15]), these hitting sets would work for this $\sum\bigwedge\sum\prod^2$ formula. Thus, to show that PIT for roABPs does not suffice for $\sum\bigwedge\sum\prod^2$ formulas one would need such a formula that is hard in *every* variable order.

It is reasonable to expect such a hard-in-every-order $\sum\bigwedge\sum\prod^2$ formula as these formulas are inherently unordered (while roABPs are inherently ordered) so one should expect a "symmetric" $\sum\bigwedge\sum\prod^2$ formula that is hard in every variable order. A reasonable attempt to symmetrize the previously studied $(1+\sum_i x_iy_i)^d$ would be to consider $(\sum_{i<j} x_ix_j)^d$, however this formula has small roABPs in *every* order [14]. Intuitively, this is because if you partition $\overline{x} = (\overline{y}, \overline{z})$ then $\sum_{i<j} x_ix_j = (\sum_i y_i)(\sum_j z_j) + f(\overline{y}) + g(\overline{z})$ for some polynomials f, g. As such, we see that *no correlation* is induced between $\overline{y}$ and $\overline{z}$ as their only interaction is via a variable-disjoint product. This lack of correlation then extends under d-th powers.

In this section we thus take a different approach and seek to embed the $(1+\sum_i x_iy_i)^d$ example into *every* variable order of some polynomial f. We do this via introducing indicator z-variables which we can then set appropriately to reveal the embedding of $(1+\sum_i x_iy_i)^d$, which is reminiscent of Raz's [Raz06] full-rank polynomial. That is, the polynomial $(1+\sum_i x_iy_i)^d$ is hard because we have a "matching" between $\overline{x}$ and $\overline{y}$. Given only variables $\overline{z}$ we wish to support such matchings between all possible partitions $\overline{z} = (\overline{x}, \overline{y})$. To do so, we identity the variables $\overline{z}$ with nodes in a complete graph and weight each edge (z_i, z_j) with a new variable $t_{i,j}$. This thus leads to the polynomial $(1+\sum_{i<j} x_ix_jt_{i,j})^d$. Any partition $\overline{z} = (\overline{x}, \overline{y})$ then induces a cut of this graph, and setting the variables $t_{i,j}$ to $\{0,1\}$ values appropriately we can recover a large matching across this cut, yielding the polynomial $(1+\sum_i x_iy_i)^d$, giving the desired lower bound.

Lemma 8.8. *Let $d \ge 0$ and $\mathbb{F}$ be a field with $\operatorname{char}(\mathbb{F}) > d$. Let $\overline{x}$ be a set of n variables and $\overline{t}$ a disjoint set of variables indexed by $\binom{[n]}{2}$. Define $f(\overline{x}, \overline{t}) := (1+\sum_{i<j} x_ix_jt_{i,j})^d$. Then for any $S \subseteq [n]$ there is some $\overline{\gamma} \in \mathbb{F}^{\binom{[n]}{2}}$ such that*

$$\dim \mathbf{Coeff}_{\overline{x}_S|\overline{x}_{\overline{S}}}(f(\overline{x}, \overline{\gamma})) \ge \binom{\min\{|S|, n-|S|\}+d}{d},$$

where $\overline{S} := [n]\setminus S$ and where '$\overline{x}_S$' denotes in our usual notation $\overline{x}|_S$. In particular, computing $f(\overline{x}, \overline{t})$ as a roABP requires width $\ge \binom{\lfloor n/2\rfloor+d}{d}$ in every variable order.

Proof: $\underline{\dim \mathbf{Coeff}}$: As $\dim \mathbf{Coeff}_{\overline{x}_S|\overline{x}_{\overline{S}}} = \dim \mathbf{Coeff}_{\overline{x}_{\overline{S}}|\overline{x}_S}$ (Lemma 8.2) it suffices to study when $n-|S| \ge |S|$, as the opposite case is symmetric. In this case, we can then take $T \subseteq \overline{S}$ with $|T| = |S|$ and $\sigma : S \to T$ be a bijection/matching. Now define for $i < j$,

$$\gamma_{i,j} := \begin{cases} 1 & i \in S, j \in T, \sigma(i) = j \\ 0 & \text{else} \end{cases}.$$

[14] Decompose $2\sum_{i<j} x_ix_j = \left(\sum_i x_i\right)^2 - \sum_{i<j} x_i^2$. Powers of both $\left(\sum_i x_i\right)^2$ and $\sum_{i<j} x_i^2$ both have small roABPs in every order by Lemma 1.3. One can then use the binomial theorem to represent $\sum_{i<j} x_ix_j$ as a small sum of products of two roABPs (in the same variable order), which will have a small roABP by Lemma 3.17.

Thus,

$$\begin{aligned} f(\overline{x},\overline{\gamma}) &= (1+\sum_{i<j} x_i x_j \gamma_{i,j})^d \\ &= (1+\sum_{i\in S} x_i x_{\sigma(i)})^d \\ &= g(\overline{x}_S,\overline{x}_T)\ , \end{aligned}$$

where $g(\overline{y},\overline{z}) = (1+\sum_i y_i z_i)^d$. Thus, $\dim \mathbf{Coeff}_{\overline{x}_S}(f(\overline{x},\overline{\gamma})) = \dim \mathbf{Coeff}_{\overline{x}_S} g(\overline{x}_S,\overline{x}_T) = \binom{|S|+d}{d}$ (Lemma 8.5).

roABP lower bound: Suppose that $f(\overline{x},\overline{z})$ was computed by a width-w in some order roABP. Then as roABPs are closed under partial substitutions (Lemma 3.17), we have that for every $\overline{\gamma}\in\mathbb{F}^{\binom{[n]}{2}}$ that $f(\overline{x},\overline{\gamma})$ is computable by a width-w in some order $\pi:[n]\to[n]$. Define $S := \pi^{-1}(\{1,\ldots,\lfloor n/2\rfloor\})$, so then by Nisan's [Nis91] lower bound method (Lemma 8.4) we have that $w\geq \dim \mathbf{Coeff}_{\overline{x}_S|\overline{x}_{\overline{S}}}(f(\overline{x},\overline{\gamma}))$. Choosing $\overline{\gamma}$ as above, we get $w\geq\binom{\lfloor n/2\rfloor+d}{d}$ as desired. □

While this shows that $\bigwedge\sum\prod^3$ formulas compute polynomials that exponentially hard for roABPs in every order, there are three drawbacks to this result. The first is that this still does not address $\bigwedge\sum\prod^2$ formulas where one still expects this result to hold. The second drawback is for certain partial substitution of the variables the polynomial becomes easy (setting $\overline{z}=\overline{1}$ essentially yields the formula $(\sum_{i<j} x_i x_j)^d$ which as mentioned above has a small roABP in every order). The third drawback is that the lower bound is $\binom{\Theta(n)+d}{d}$ for a polynomial in $\Theta(n^2)$-variables, which in some regimes is approximately the square root of the maximal complexity $\binom{\Theta(n^2)+d}{d}$.

While the first and second drawbacks seem more inherent to this approach, the third is fixable [15]. However, we present an ever better construction in the next section that simultaneously addresses all of these concerns.

8.4 Lower Bounds for Computing $\sum\bigwedge\sum\prod^2$ Formulas by roABPs, in Any Order

In the previous subsections we showed that $(1+\sum_i x_i y_i)^d$ requires large roABPs in any variable ordering that partitions $\overline{x}$ and $\overline{y}$ (Subsection 8.2). This formula has such a lower bound because it induces a matching between $\overline{x}$ and $\overline{y}$ which induces a lot of "correlation", while roABPs cannot produce large correlation between partitions that respect their variable order. To extend this to a formula with a lower bound in any variable ordering we used indicator variables to embed large matchings across any possible partition, which resulted in a $\bigwedge\sum\prod^3$ formula (Subsection 8.4).

In this section we derive a lower bound inheriting the best of the above two results, a $\bigwedge\sum\prod^2$ formula that is essentially maximally hard for every partition. To do this, observe that our embedding of the matching $(1+\sum_i x_i y_i)^d$ into every partition in Subsection 8.4 was essentially *combinatorial* in that our indicator variables only used $\{0,1\}$ values. But as we have algebraic computation we are free to choose our constants from the entire field. In particular, we can consider *algebraic* notions of information flow. That is, $(1+\sum_i x_i y_i)^d = (1+\overline{x}^t A\overline{y})^d$ where A is the $n\times n$ identity matrix I_n. One could allow other full rank matrices A and the same lower bound would follow (as coefficient dimension is invariant to basis change (Lemma 8.18)). Thus, to ensure that we can embed the above

[15] One can address this drawback by the usage of constant-degree expander graphs. That is, we defined our hard formula as $(1+\sum_{(i,j)\in E} x_i x_j t_{i,j})^d$ where E was the edge set of the complete graph. The property of the complete graph that we needed was that for every balanced cut there is a large matching across this cut. By using the appropriate type of expander graph one could reduce the number of edges in E while roughly preserving the size of the resulting matchings, which should give a lower bound of $\binom{\Theta(n)+d}{d}$ in $\Theta(n)$ variables as desired.

$(1+\sum_i x_i y_i)^d$ into any partition of $\overline{z}$ into $\overline{z} = (\overline{x}, \overline{y})$ it seems reasonable to consider the polynomial $(1+\overline{z}^t A \overline{z})^d$ where A is *totally non-singular*, which we now define.

Definition 8.9. *A matrix $A \in \mathbb{F}^{n\times n}$ is **totally non-singular** if all square submatrices are non-singular, that is for every $S \subseteq [n]$ and $T \subseteq [n]$ with $|S| = |T|$ that $\det A|_{S\times T} \neq 0$.* ◊

We now define a well-known matrix called the *Cauchy matrix*.

Definition 8.10. *The $n \times m$ **Cauchy matrix** $C(\overline{\alpha}, \overline{\beta})$ over elements $\alpha_1, \ldots, \alpha_n$ and $\beta_1, \ldots, \beta_m$ in a field $\mathbb{F}$ is defined by*

$$(C(\overline{\alpha}, \overline{\beta}))_{i,j} := \frac{1}{\alpha_i + \beta_j} \ . \qquad \Diamond$$

Note that sub-matrices of Cauchy matrices are Cauchy. The determinant of square Cauchy matrices is well-known.

Lemma 8.11. *Let $\mathbb{F}$ be a field. Let $C(\overline{\alpha}, \overline{\beta})$ be a $n \times n$ Cauchy matrix. Then*

$$\det C(\overline{\alpha}, \overline{\beta}) = \frac{\prod_{i<j}(\alpha_i - \alpha_j) \cdot \prod_{i<j}(\beta_i - \beta_j)}{\prod_{i,j}(\alpha_i + \beta_j)} \ ,$$

if $\alpha_i + \beta_j \neq 0$ for all i, j. □

In particular, this is non-zero if all $\overline{\alpha}$ are distinct and all $\overline{\beta}$ are distinct. Note that this distinctness is also held by the sub-matrices, so Cauchy matrices are totally non-singular. In particular, if we take $\overline{\alpha} = \overline{\beta}$ (and pick $\overline{\alpha}$ so that it has no common elements with $-\overline{\alpha}$) we get a symmetric totally non-singular matrix.

Corollary 8.12. *Let $\mathbb{F}$ be a field with $|\mathbb{F}| \geq 2n$. Then one can find in $\mathsf{poly}(n)$ steps a symmetric $n \times n$ totally non-singular matrix in $\mathbb{F}^{n\times n}$.* □

Given the above matrix, we now work toward showing it can be used to construct the desired hard $\bigwedge\sum\prod^2$ formula. We need the following fact relating coefficients and derivatives.

Lemma 8.13. *Let $f \in \mathbb{F}[\overline{x}]$. Then*

$$\mathrm{Coeff}_{\overline{x}^{\overline{a}}}(f(\overline{x})) = (\partial_{\overline{x}^{\overline{a}}} f)(\overline{0}) \ .$$

In particular, for $f \in \mathbb{F}[\overline{x}][\overline{y}]$,

$$\mathrm{Coeff}_{\overline{x}^{\overline{a}}}(f(\overline{x}, \overline{y})) = (\partial_{\overline{x}^{\overline{a}}} f)(\overline{0}, \overline{y}) \ .$$

Proof: By definition $f(\overline{x} + \overline{z}) = \sum_{\overline{a}} \left(\partial_{\overline{x}^{\overline{a}}} f\right)(\overline{x}) \cdot \overline{z}^{\overline{a}}$, so that setting $\overline{x} = 0$ we obtain $f(\overline{z}) = \sum_{\overline{a}} \left(\partial_{\overline{x}^{\overline{a}}} f\right)(\overline{0}) \cdot \overline{z}^{\overline{a}}$. Setting $\overline{z} = \overline{x}$ recovers the result. □

We need a claim about the structure of Hasse derivatives of a power of a polynomial.

Lemma 8.14. *Let $f(\overline{x}) \in \mathbb{F}[x_1, \ldots, x_n]$ and $\overline{a}$ be so $a := \deg \overline{x}^{\overline{a}}$ and $d \geq a$. Then there is some $g(\overline{x}) \in \mathbb{F}[\overline{x}]$ so that*

$$(\partial_{\overline{x}^{\overline{a}}} f^d)(\overline{x}) = g(\overline{x}) f^{d-a+1} + \tbinom{d}{d-a, a_1, \ldots, a_n} \cdot f^{d-a} \cdot (\partial_{\overline{x}} f)^{\overline{a}} \ ,$$

where $\partial_{\overline{x}} f := (\partial_{x_1} f, \ldots, \partial_{x_n} f)$ so that $(\partial_{\overline{x}} f)^{\overline{a}} = \prod_i (\partial_{x_i} f)^{\overline{a}}$.

Proof: By the product rule (Lemma 3.3)

$$\left(\partial_{\overline{x}^{\overline{a}}} f^d\right) = \sum_{\bar{b}_1 + \cdots + \bar{b}_d = \overline{a}} \left(\partial_{\overline{x}^{\bar{b}_1}} f\right) \cdots \left(\partial_{\overline{x}^{\bar{b}_d}} f\right)$$

Now define $b_i := \|\overline{b}_i\|_1$ and collect these numbers into the single vector $\overline{b} \in \mathbb{N}^d$. As $\|\overline{b}\|_1 = a \le d$ it follows that $\|\overline{b}\|_\infty \ge 1$ for every $\overline{b}$ and that $\|\overline{b}\|_\infty = 1$ occurs for some $\overline{b}$. Note that if $\|\overline{b}\|_\infty = 1$ then $d-a$ of the $\overline{b}_i$ are $\overline{0}$, in which case $\partial_{\overline{x}^{\overline{b}_i}} f = f$, and otherwise $\overline{b}_i$ is some standard basis vector $\overline{e}_{j_i}$, in which case $\partial_{\overline{x}^{\overline{b}_i}} f = \partial_{x_{j_i}} f$. If $\|\overline{b}\|_\infty > 1$ then $\ge d-a+1$ of the $\overline{b}_i$ are $\overline{0}$. Thus,

$$
\begin{aligned}
&= \sum_{\overline{b}_1+\cdots+\overline{b}_d=\overline{a}, \|\overline{b}\|_\infty=1} \left(\partial_{\overline{x}^{\overline{b}_1}} f\right) \cdots \left(\partial_{\overline{x}^{\overline{b}_d}} f\right) + \sum_{\overline{b}_1+\cdots+\overline{b}_d=\overline{a}, \|\overline{b}\|_\infty>1} \left(\partial_{\overline{x}^{\overline{b}_1}} f\right) \cdots \left(\partial_{\overline{x}^{\overline{b}_d}} f\right) \\
&= \sum_{\overline{b}_1+\cdots+\overline{b}_d=\overline{a}, \|\overline{b}\|_\infty=1} f^{d-a} \prod_{i \mid \overline{b}_i \neq \overline{0}} \left(\partial_{\overline{x}^{\overline{b}_i}} f\right) + \sum_{\overline{b}_1+\cdots+\overline{b}_d=\overline{a}, \|\overline{b}\|_\infty>1} f^{d-a+1} \prod_{i \mid \overline{b}_i \neq \overline{0}} \left(\partial_{\overline{x}^{\overline{b}_i}} f\right)
\end{aligned}
$$

defining $g(\overline{x}) := \sum_{\overline{b}_1+\cdots+\overline{b}_d=\overline{a}, \|\overline{b}\|_\infty>1} \prod_{i \mid \overline{b}_i \neq \overline{0}} \left(\partial_{\overline{x}^{\overline{b}_i}} f\right)$.

$$
\begin{aligned}
&= g(\overline{x}) f^{d-a+1} + f^{d-a} \cdot \sum_{\overline{b}_1+\cdots+\overline{b}_d=\overline{a}, \|\overline{b}\|_\infty=1} \prod_{j \in [n]} \left(\partial_{x_j} f\right)^{|\{i \mid \overline{b}_i = \overline{e}_j\}|} \\
&= g(\overline{x}) f^{d-a+1} + f^{d-a} \cdot \sum_{\overline{b}_1+\cdots+\overline{b}_d=\overline{a}, \|\overline{b}\|_\infty=1} (\partial_{\overline{x}} f)^{\overline{a}}
\end{aligned}
$$

The sequence $\overline{b}_1, \ldots, \overline{b}_d$ is just a sequence of $\overline{0}, \overline{e}_1, \ldots, \overline{e}_n$, where $\overline{0}$ occurs $d-a$ times, and $\overline{e}_i$ occurs a_i times, so thus the number of such sequences is $\frac{d!}{(d-a)!a_1!\cdots a_n!} = \binom{d}{d-a, a_1, \ldots, a_n}$,

$$
= g(\overline{x}) f^{d-a+1} + \binom{d}{d-a, a_1, \ldots, a_n} \cdot f^{d-a} \cdot (\partial_{\overline{x}} f)^{\overline{a}} . \qquad \square
$$

This lemma shows that if we work modulo f^{d-a+1} then we just have $f^{d-a} \cdot (\partial_{\overline{x}} f)^{\overline{a}}$ (ignoring constants), which is much easier to work with than the entire derivative.

We now give a robust generalization of the coefficient dimension of the formula $(1 + \overline{x}^t \overline{y})^d$ from Subsection 8.2. That is, in this polynomial it is the $\overline{x}^t \overline{y}$ term that introduces the correlation between $\overline{x}$ and $\overline{y}$ (and the powering operation amplifies this). One could then add in additional terms such as $\overline{x}^t B \overline{x}$ which by themselves do not introduce any correlation between $\overline{x}$ and $\overline{y}$ and hope that correlation is not damaged significantly. Indeed, we now show this using the above lemma.

Lemma 8.15. *Let $d \ge 0$ and* char$(\mathbb{F}) > d$. *Let $\overline{x}$ and $\overline{y}$ be disjoint sets of n variables. Let $\delta \in \mathbb{F}$, $\overline{\beta}, \overline{\gamma} \in \mathbb{F}^n$ and $B, C \in \mathbb{F}^{n \times n}$. Define $f(\overline{x}, \overline{y}) := (\overline{x}^t \overline{y} + \overline{x}^t B \overline{x} + \overline{y}^t C \overline{y} + \overline{\beta}^t \overline{x} + \overline{\gamma}^t \overline{y} + \delta)$. If $\deg_{\overline{y}} f(\overline{0}, \overline{y}) \ge 1$, then*

$$
\dim \mathbf{Coeff}_{\overline{x}|\overline{y}} f(\overline{x}, \overline{y})^d \ge \binom{n-1+d}{d}
$$

Proof: To begin we state what a first derivative of f looks like.

Subclaim 8.16. *For $k \in [n]$ and $d \ge 0$,*

$$
\partial_{x_k} f(\overline{x}, \overline{y}) = y_k + \beta_k + \textstyle\sum_j (B_{j,k} + B_{k,j}) x_j .
$$

Sub-Proof:

$$
\begin{aligned}
\partial_{x_k} f(\overline{x}, \overline{y}) &= \partial_{x_k} (\overline{x}^t \overline{y} + \overline{x}^t B \overline{x} + \overline{y}^t C \overline{y} + \overline{\beta}^t \overline{x} + \overline{\gamma}^t \overline{y} + \delta) \\
&= \partial_{x_k} \left(\textstyle\sum_i x_i y_i + \sum_{i,j} x_i B_{i,j} x_j + \sum_{i,j} y_i C_{i,j} y_j + \sum_i \beta_i x_i + \sum_i \gamma_i y_i + \delta \right)
\end{aligned}
$$

note that $\sum_{i,j} x_i B_{i,j} x_j = B_{k,k}x_k^2 + \sum_{j\neq k} x_k B_{k,j} x_j + \sum_{i \neq k} x_i B_{i,k} x_k + \sum_{i,j\neq k} x_i B_{i,j} x_j$, so that $\partial_{x_k} \sum_{i,j} x_i B_{i,j} x_j = 2B_{k,k}x_k + \sum_{j\neq k} B_{k,j}x_j + \sum_{i\neq k} B_{i,k}x_i = \sum_j (B_{j,k} + B_{k,j})x_j$,

$$= y_k + \sum_i (B_{j,k} + B_{k,j})x_j + \beta_k \; . \qquad \square$$

We now seek to understand the space $\mathbf{Coeff}_{\overline{x}} f^d$ by looking at $\mathrm{Coeff}_{\overline{x}^{\overline{a}}}$ for $a := \deg \overline{x}^{\overline{a}} \leq d$. We proceed to extract coefficients via taking derivatives, appealing to Lemma 8.13.

$$\mathrm{Coeff}_{\overline{x}^{\overline{a}}} f(\overline{x}, \overline{y})^d = (\partial_{\overline{x}^{\overline{a}}} f^d)(\overline{0}, \overline{y})$$

using our lemma on the structure of derivatives of a power (Lemma 8.14) there is some $g(\overline{x}, \overline{y})$ so that

$$= \left[g \cdot f^{d-a+1} + \tbinom{d}{d-a, a_1, \ldots, a_n} \cdot f^{d-a} \cdot (\partial_{\overline{x}} f)^{\overline{a}} \right] (\overline{0}, \overline{y})$$

appealing to the above subclaim,

$$= \left[g \cdot f^{d-a+1} + \tbinom{d}{d-a, a_1, \ldots, a_n} \cdot f^{d-a} \cdot \prod_{i\in[n]} \left(y_i + \beta_i + \textstyle\sum_j (B_{i,j} + B_{j,i})x_j \right)^{a_i} \right] (\overline{0}, \overline{y})$$
$$= g(\overline{0}, \overline{y}) \cdot f(\overline{0}, \overline{y})^{d-a+1} + \tbinom{d}{d-a, a_1, \ldots, a_n} \cdot f(\overline{0}, \overline{y})^{d-a} \cdot (\overline{y} + \overline{\beta})^{\overline{a}} \; .$$

If the above expression was just the monomial $(\overline{y} + \overline{\beta})^{\overline{a}}$ then we would be done by following the proof of Lemma 8.5. Instead, we will need to kill off the $g(\overline{0}, \overline{y}) \cdot f(\overline{0}, \overline{y})^{d-a+1}$ term. To do so, we will work modulo $f(\overline{0}, \overline{y})^{d-a+1}$. To ensure that the $f(\overline{0}, \overline{y})^{d-a} \cdot (\overline{y} + \overline{\beta})^{\overline{a}}$ terms remain linearly independent we take only a subset of the possible $\overline{a}$. That is, as $\deg f(\overline{0}, \overline{y}) \geq 1$ there is some $\ell \in [n]$ so that $\deg_{y_\ell} f(\overline{0}, \overline{y}) \geq 1$. We will take those $\overline{a}$ where $a_\ell = 0$ and show that these monomials are linearly independent.

Subclaim 8.17. *Taking coefficients in $\mathbb{F}[\overline{y}][\overline{x}]$, the set of polynomials*

$$\{\mathrm{Coeff}_{\overline{x}^{\overline{a}}} f(\overline{x}, \overline{y})^d \mid \deg \overline{x}^{\overline{a}} \leq d, a_\ell = 0\}$$

are linearly independent.

Sub-Proof: For contradiction, suppose there was a non-trivial linear dependence

$$\sum_{\overline{a} \mid \deg \overline{x}^{\overline{a}} \leq d, a_\ell = 0} \alpha_{\overline{a}} \mathrm{Coeff}_{\overline{x}^{\overline{a}}} f(\overline{x}, \overline{y})^d = 0 \; .$$

Then take $b = \max\{\deg \overline{x}^{\overline{a}} \mid \alpha_{\overline{a}} \neq 0\}$, which exists as not all $\alpha_{\overline{a}}$ are zero. Using the above computation we obtain

$$\sum_{0\leq a\leq b} \sum_{\substack{\overline{a} \\ \deg \overline{x}^{\overline{a}} = a, a_\ell = 0}} \alpha_{\overline{a}} \left(g(\overline{0}, \overline{y}) \cdot f(\overline{0}, \overline{y})^{d-a+1} + \tbinom{d}{d-a, a_1, \ldots, a_n} \cdot f(\overline{0}, \overline{y})^{d-a} \cdot (\overline{y} + \overline{\beta})^{\overline{a}} \right) = 0 \; .$$

Taking this equation modulo $f(\overline{0}, \overline{y})^{d-b+1}$ we get

$$\sum_{\substack{\overline{a} \\ \deg \overline{x}^{\overline{a}} = b, a_\ell = 0}} \alpha_{\overline{a}} \cdot \tbinom{d}{d-b, a_1, \ldots, a_n} \cdot f(\overline{0}, \overline{y})^{d-b} \cdot (\overline{y} + \overline{\beta})^{\overline{a}} = 0 \pmod{f(\overline{0}, \overline{y})^{d-b+1}} \; .$$

In particular, dividing this equation by $f(\overline{0},\overline{y})^{d-b}$,

$$\sum_{\substack{\overline{a} \\ \deg \overline{x}^{\overline{a}}=b, a_\ell=0}} \alpha_{\overline{a}} \cdot \binom{d}{d-b,a_1,\ldots,a_n} \cdot (\overline{y}+\overline{\beta})^{\overline{a}} = 0 \pmod{f(\overline{0},\overline{y})} \ .$$

Rephrasing, this means that $f(\overline{0},\overline{y})$ divides this summation. However, as $\deg_{y_\ell} f(\overline{0},\overline{y}) \geq 1$ by choice of ℓ and $\deg_{y_\ell}(\overline{y}+\overline{\beta})^{\overline{a}} = 0$ by our choice of $\overline{a}$, this must mean that this summation is actually zero, that is,

$$\sum_{\substack{\overline{a} \\ \deg \overline{x}^{\overline{a}}=b, a_\ell=0}} \alpha_{\overline{a}} \cdot \binom{d}{d-b,a_1,\ldots,a_n} \cdot (\overline{y}+\overline{\beta})^{\overline{a}} = 0 \ .$$

However, the $(\overline{y}+\overline{\beta})^{\overline{a}}$ are all linearly independent (as being distinct monomials in the $\overline{y}+\overline{\beta}$ basis) and the $\binom{d}{d-b,a_1,\ldots,a_n}$ are all nonzero (as $\mathrm{char}(\mathbb{F}) > d$), so this implies the $\alpha_{\overline{a}}$ are all zero. This yields the desired contradiction of our choice of b. □

Thus, $\mathbf{Coeff}_{\overline{x}} f(\overline{x},\overline{y})^d$ contains $\binom{n-1+d}{d}$ linearly independent polynomials as desired. □

Note that this quantity is very close to the trivial upper bound of $\binom{n-1+d}{d}$ of Lemma 8.3.

We now seek to embed this example into every partition of a single formula. To do so, we need the following lemma showing that coefficient dimension is invariant under cetain changes of basis.

Lemma 8.18. *Let $f(\overline{x},\overline{y}) \in \mathbb{F}[\overline{x},\overline{y}]$ be a polynomial and let $\overline{x}$ be a vector of n variables. Let $A \in \mathbb{F}^{n\times n}$ be invertible. Then*

$$\dim \mathbf{Coeff}_{\overline{x}|\overline{y}}(f(A\overline{x},\overline{y})) = \dim \mathbf{Coeff}_{\overline{x}|\overline{y}}(f(\overline{x},\overline{y})) = \dim \mathbf{Coeff}_{\overline{x}|\overline{y}}(f(\overline{x},A\overline{y})) \ .$$

Proof: The second inequality follows from the first by symmetry of $\overline{x}$ and $\overline{y}$, as $\dim \mathbf{Coeff}_{\overline{y}|\overline{x}} f(\overline{y},\overline{x}) = \dim \mathbf{Coeff}_{\overline{y}} f(\overline{y},\overline{x}) = \dim \mathbf{Coeff}_{\overline{x}} f(\overline{x},\overline{y}) = \dim \mathbf{Coeff}_{\overline{x}|\overline{y}} f(\overline{x},\overline{y})$ (Lemma 8.2).

Let $S \subseteq \mathbb{F}[\overline{x}]$. Consider the map $\varphi : \mathbb{F}[\overline{x},\overline{y}] \to \mathbb{F}[\overline{x},\overline{y}]$ defined $(\overline{x},\overline{y}) \mapsto (A\overline{x},\overline{y})$. As A is invertible, φ is an isomorphism of vector spaces. Thus, $\dim \operatorname{span} S = \dim \operatorname{span} \varphi(S)$.

Note that $f(\overline{x},\overline{y}) = \sum_{\overline{b}} f_{\overline{b}}(\overline{x})\overline{y}^{\overline{b}}$ where $f_{\overline{b}} := \mathrm{Coeff}_{\overline{y}^{\overline{b}}}(f(\overline{x},\overline{y}))$. Thus, $f(A\overline{x},\overline{y}) = \varphi(f(\overline{x},\overline{y})) = \sum_{\overline{b}} f_{\overline{b}}(A\overline{x})\overline{y}^{\overline{b}}$, so $\mathrm{Coeff}_{\overline{y}^{\overline{b}}}(f(A\overline{x},\overline{y})) = f_{\overline{b}}(A\overline{x}) = \varphi(f_{\overline{b}}(\overline{x})) = \varphi(\mathrm{Coeff}_{\overline{y}^{\overline{b}}}(f(\overline{x},\overline{y})))$. Thus

$$\varphi(\mathbf{Coeff}_{\overline{y}}(f(\overline{x},\overline{y}))) = \mathbf{Coeff}_{\overline{y}}(f(A\overline{x},\overline{y})) \ ,$$

In particular $\dim \operatorname{span} \mathbf{Coeff}_{\overline{y}}(f(\overline{x},\overline{y})) = \dim \operatorname{span} \mathbf{Coeff}_{\overline{y}}(f(A\overline{x},\overline{y}))$ as desired. □

We also show that coefficient dimension does not increase under partial substitution.

Lemma 8.19. *Let $f(\overline{x},\overline{y},\overline{y}) \in \mathbb{F}[\overline{x},\overline{y},\overline{y}]$ be a polynomial. Then, for any $\overline{\gamma} \in \mathbb{F}^{|\overline{z}|}$ where $|\overline{z}|$ is the number of variables in $\overline{z}$,*

$$\dim \mathbf{Coeff}_{\overline{x}|\overline{y}\overline{z}} f(\overline{x},\overline{y},\overline{z}) \geq \dim \mathbf{Coeff}_{\overline{x}|\overline{y}} f(\overline{x},\overline{y},\overline{\gamma}) \ .$$

Proof: Taking coefficients in $\mathbb{F}[\overline{y},\overline{z}][\overline{x}]$,

$$\begin{aligned}
\dim \mathbf{Coeff}_{\overline{x}|\overline{y}\overline{z}} f(\overline{x},\overline{y},\overline{z}) &= \dim \operatorname{span} \mathbf{Coeff}_{\overline{x}} f(\overline{x},\overline{y},\overline{z}) \\
&= \dim \operatorname{span}\{\mathrm{Coeff}_{\overline{x}^{\overline{a}}} f(\overline{x},\overline{y},\overline{z})\} \\
&\geq \dim \operatorname{span}\{\mathrm{Coeff}_{\overline{x}^{\overline{a}}} f(\overline{x},\overline{y},\overline{\gamma})\} \\
&= \dim \operatorname{span} \mathbf{Coeff}_{\overline{x}} f(\overline{x},\overline{y},\overline{\gamma}) \ .
\end{aligned}$$

where we use that the substitution homormorphism $\overline{z} \mapsto \overline{\gamma}$ is a linear map so cannot increase dimension. □

We now consider $(\overline{z}^t M \overline{z})^d$ where M is totally non-singular, and show that its coefficient dimension under any partial substitution embeds (under a change of basis) the above robust generalization of $(1+\overline{x}^t\overline{y})^d$.

Lemma 8.20. *Let $d \geq 0$ and $\operatorname{char}(\mathbb{F}) > d, 2$. Define $f(\overline{v}) \in \mathbb{F}[\overline{v}]$ by $f(\overline{x}) := \overline{v}^t M \overline{v}$ where $M \in \mathbb{F}^{|\overline{v}|\times|\overline{v}|}$ is symmetric and totally non-singular, and where we define $|\overline{v}|$ to be the number of variables in $\overline{v}$. Then for any partition of $\overline{v}$ into $\overline{v} = (\overline{x}, \overline{y}, \overline{z})$ and $\overline{\gamma} \in \mathbb{F}^{|\overline{z}|}$,*

$$\dim \mathbf{Coeff}_{\overline{x}|\overline{y}} f(\overline{x}, \overline{y}, \overline{\gamma})^d \geq \min\left\{ \tbinom{|\overline{x}|-1+d}{d}, \tbinom{|\overline{y}|-1+d}{d} \right\} .$$

Proof: As $\dim \mathbf{Coeff}_{\overline{x}|\overline{y}}(f) = \dim \mathbf{Coeff}_{\overline{y}|\overline{x}}(f) = \dim \operatorname{span} \mathbf{Coeff}_{\overline{x}}(f)$, it suffices by symmetry to prove the claim when $|\overline{x}| \geq |\overline{y}|$. Further, as coefficient dimension does not increase under partial substitutions (Lemma 8.19) we can assume $|\overline{x}| = |\overline{y}|$. That is, if we partition $\overline{x} = (\overline{x}', \overline{z}')$ where $|\overline{x}'| = |\overline{y}|$ then $\dim \mathbf{Coeff}_{\overline{x}'\overline{z}'|\overline{y}} f(\overline{x}', \overline{z}', \overline{y}, \overline{\gamma}) \geq \dim \mathbf{Coeff}_{\overline{x}'|\overline{y}} f(\overline{x}', \overline{\gamma}', \overline{y}, \overline{\gamma})$ for any $\overline{\gamma}' \in \mathbb{F}^{|\overline{x}''|}$. Applying the claim in this case (so that we evaluate $\overline{z}$ *and* $\overline{z}'$) yields the desired lower bound of $\binom{|\overline{x}|-1+d}{d}$. Thus, we now prove the claim when $|\overline{x}| = |\overline{y}|$.

Set $\overline{z} = \overline{\gamma}$. In block notation we have

$$\overline{v} = \begin{bmatrix} \overline{x} \\ \overline{y} \\ \overline{\gamma} \end{bmatrix}, \qquad\qquad M = \begin{bmatrix} A & B & C \\ D & E & F \\ G & H & I \end{bmatrix} .$$

Note that 'I' here is not to be confused with the identity matrix I. By assumption that $|\overline{x}| = |\overline{y}|$, we have (in particular) that B and E are square. As M is symmetric, A, E and I are symmetric, $B^t = D$, $C^t = G$ and $F^t = H$. Thus,

$$\begin{aligned} f(\overline{x}, \overline{y}, \overline{\gamma}) = \overline{v}^t M \overline{v} &= \Big((\overline{x}^t A \overline{x} + \overline{x}^t B \overline{y} + \overline{x}^t C \overline{\gamma}) + (\overline{y}^t D \overline{x} + \overline{y}^t E \overline{y} + \overline{y}^t F \overline{\gamma}) + (\overline{\gamma}^t G \overline{x} + \overline{\gamma}^t H \overline{y} + \overline{\gamma}^t I \overline{\gamma}) \Big) \\ &= \overline{x}^t (B + D^t) \overline{y} + \overline{x}^t A \overline{x} + \overline{y}^t E \overline{y} + \overline{\gamma}^t (C + G^t) \overline{x} + \overline{\gamma}^t (F^t + H) \overline{y} + \overline{\gamma}^t I \overline{\gamma} \end{aligned}$$

appealing to symmetry,

$$= 2\overline{x}^t B \overline{y} + \overline{x}^t A \overline{x} + \overline{y}^t E \overline{y} + 2\overline{\gamma}^t C \overline{x} + 2\overline{\gamma}^t H \overline{y} + \overline{\gamma}^t I \overline{\gamma} .$$

As M is totally non-singular, $2B$ is invertible (as $\operatorname{char}(\mathbb{F}) > 2$) and that E is as well. Now consider the change of coordinates $\overline{y} \mapsto (2B)^{-1}\overline{y}$, in which case

$$f(\overline{x}, (2B)^{-1}\overline{y}, \overline{\gamma}) = \overline{x}^t \overline{y} + \overline{x}^t A \overline{x} + \overline{y}^t ((2B)^{-1})^t E (2B)^{-1} \overline{y} + 2\overline{\gamma}^t C \overline{x} + 2\overline{\gamma}^t H (2B)^{-1} \overline{y} + \overline{\gamma}^t I \overline{\gamma} .$$

In particular,

$$f(\overline{0}, (2B)^{-1}\overline{y}, \overline{\gamma}) = \overline{y}^t ((2B)^{-1})^t E (2B)^{-1} \overline{y} + 2\overline{\gamma}^t H (2B)^{-1} \overline{y} + \overline{\gamma}^t I \overline{\gamma} .$$

As E and $2B$ are invertible, it implies that $((2B)^{-1})^t E (2B)^{-1}$ is invertible and in particular non-zero. Thus $\deg_{\overline{y}} f(\overline{0}, (2B)^{-1}\overline{y}, \overline{\gamma}) \geq 2$.

Thus, appealing to the invariance of coefficient dimension under change of basis (Lemma 8.18),

$$\dim \mathbf{Coeff}_{\overline{x}|\overline{y}} f(\overline{x}, \overline{y}, \overline{\gamma})^d = \dim \mathbf{Coeff}_{\overline{x}|\overline{y}} f(\overline{x}, (2B)^{-1}\overline{y}, \overline{\gamma})^d$$

Now note that $f(\overline{x}, (2B)^{-1}\overline{y}, \overline{\gamma})^d$ is exactly of the form so that Lemma 8.15 applies, yielding the lower bound

$$\geq \binom{|\overline{x}| - 1 + d}{d} .$$

□

Note that the trivial upper bound in this situation is $\binom{n+2d}{2d}$ as $(\overline{v}^t M \overline{v})^d$ is of degree $2d$. Thus, there is still some distance to maximality in the above example.

Putting the above together with our explicit construction of symmetric totally non-singular matrices (Corollary 8.12) we obtain the following corollary.

Corollary 8.21. *Let $\mathbb{F}$ be a field with $|\mathbb{F}| \geq 2n$ and $\operatorname{char}(\mathbb{F}) > d, 2$. Then one can compute in* $\mathsf{poly}(n)$ *steps a matrix $M \in \mathbb{F}^{n \times n}$ and define $f(\overline{v}) \in \mathbb{F}[v_1, \ldots, v_n]$ by $f(\overline{v}) := \overline{v}^t M \overline{v}$, such that the $\bigwedge\sum\prod^2$ formula $f(\overline{v})^d$ has the property that for any partition of $\overline{v}$ into $\overline{v} = (\overline{x}, \overline{y}, \overline{z})$ and $\overline{\gamma} \in \mathbb{F}^{|\overline{z}|}$,*

$$\dim \mathbf{Coeff}_{\overline{x}|\overline{y}} f(\overline{x}, \overline{y}, \overline{\gamma})^d \geq \min\left\{\binom{|\overline{x}|-1+d}{d}, \binom{|\overline{y}|-1+d}{d}\right\} .$$

In particular, any roABP computing $f(\overline{v})$ in any order requires width $\geq \binom{\lfloor n/2 \rfloor - 1 + d}{d}$. □

Note that this result is fairly strong in that it separates $\bigwedge\sum\prod^2$ from roABPs in *every* variable order, even allowing *arbitrary* partial substitutions (that leave $\Omega(n)$ many variables untouched). This latter property is stronger than that of the separation of $\bigwedge\sum\prod^3$ and roABPs of Subsection 8.3 as that separation only worked for certain evaluations of the indicator functions.

Acknowledgments

We would like to thank Pritish Kamath, Ramprasad Saptharishi and Amir Shpilka for many discussions about shifted partial derivatives. We would also like to thank Amir Shpilka in particular for the question of how to deterministically test whether a quadratic polynomial divides a sparse polynomial, Chandan Saha for questions that led to the results of Section 8, and Jakob Nordström for inviting him to KTH to give lectures on algebraic complexity, where the ideas of Section 5 were then conceived. We would also like to thank Peter Bürgisser, Yuval Filmus, Neeraj Kayal, Pascal Koiran, Swastik Kopparty, Mrinal Kumar, Rafael Oliveira, Shubhangi Saraf, and Avi Wigderson for various comments and questions.

References

[AGKS14] Manindra Agrawal, Rohit Gurjar, Arpita Korwar, and Nitin Saxena. Hitting-sets for ROABP and sum of set-multilinear circuits. *arXiv*, 1406.7535, 2014.

[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.

[AJS09] Vikraman Arvind, Pushkar S. Joglekar, and Srikanth Srinivasan. Arithmetic circuits and the hadamard product of polynomials. In *Proceedings of the 29th International Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2009)*, pages 25–36, 2009. Full version at arXiv:0907.4006.

[AMS10] Vikraman Arvind, Partha Mukhopadhyay, and Srikanth Srinivasan. New results on noncommutative and commutative polynomial identity testing. *Computational Complexity*, 19(4):521–558, 2010. Preliminary version in the *23rd Annual IEEE Conference on Computational Complexity (CCC 2008)*.

[ASS13] Manindra Agrawal, Chandan Saha, and Nitin Saxena. Quasi-polynomial hitting-set for set-depth-Δ formulas. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC 2013)*, pages 321–330, 2013. Full version at arXiv:1209.2333.

[ASSS12] Manindra Agrawal, Chandan Saha, Ramprasad Saptharishi, and Nitin Saxena. Jacobian hits circuits: hitting-sets, lower bounds for depth-D occur-k formulas & depth-3 transcendence degree-k circuits. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC 2012)*, pages 599–614, 2012. Full version at arXiv:1111.0582.

[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. Full version in the Electronic Colloquium on Computational Complexity (ECCC), Technical Report TR08-062.

[BOT88] Michael Ben-Or and Prasoon Tiwari. A deterministic algorithm for sparse multivariate polynomial interpolation (extended abstract). In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing (STOC 1998)*, pages 301–309, 1988.

[CGK$^+$13a] Arkadev Chattopadhyay, Bruno Grenet, Pascal Koiran, Natacha Portier, and Yann Strozecki. Factoring bivariate lacunary polynomials without heights. In *Proceedings of the 40th International Colloquium on Automata, Languages and Programming (ICALP 2013)*, pages 141–148, 2013. The full version of this work ([CGK$^+$13b]) contains multivariate generalizations of the original (bivariate) full version (arXiv:1206.4224).

[CGK$^+$13b] Arkadev Chattopadhyay, Bruno Grenet, Pascal Koiran, Natacha Portier, and Yann Strozecki. Computing the multilinear factors of lacunary polynomials without heights. *arXiv*, 1311.5694, 2013.

[CLO07] David Cox, John Little, and Donal O'Shea. *Ideals, varieties, and algorithms*. Undergraduate Texts in Mathematics. Springer, New York, third edition, 2007. An introduction to computational algebraic geometry and commutative algebra.

[DKSS13] Zeev Dvir, Swastik Kopparty, Shubhangi Saraf, and Madhu Sudan. Extensions to the method of multiplicities, with applications to kakeya sets and mergers. *SIAM J. Comput.*, 42(6):2305–2328, 2013. Preliminary version in the *50th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2009)*.

[DL78] Richard A. DeMillo and Richard J. Lipton. A probabilistic remark on algebraic program testing. *Inf. Process. Lett.*, 7(4):193–195, 1978.

[DO14] Zeev Dvir and Rafael Oliveira. Factors of sparse polynomials are sparse. *arXiv*, 1404.4834, 2014. This manuscript has been withdrawn.

[DOS14] Zeev Dvir, Rafael Oliveira, and Amir Shpilka. Testing equivalence of polynomials under shifts. In *Proceedings of the 41st International Colloquium on Automata, Languages and Programming (ICALP 2014)*, pages 417–428, 2014. Full version at arXiv:1401.3714.

[DS07] Zeev Dvir and Amir Shpilka. Locally decodable codes with two queries and polynomial identity testing for depth 3 circuits. *SIAM J. Comput.*, 36(5):1404–1434, 2007. Preliminary version in the *37th Annual ACM Symposium on Theory of Computing (STOC 2005)*.

[DSY09] Zeev Dvir, Amir Shpilka, and Amir Yehudayoff. Hardness-randomness tradeoffs for bounded depth arithmetic circuits. *SIAM J. Comput.*, 39(4):1279–1293, 2009. Preliminary version in the *40th Annual ACM Symposium on Theory of Computing (STOC 2008)*.

[Fis94] Ismor Fischer. Sums of like powers of multivariate linear forms. *Mathematics Magazine*, 67(1):59–61, 1994.

[FLMS14] Hervé Fournier, Nutan Limaye, Guillaume Malod, and Srikanth Srinivasan. Lower bounds for depth 4 formulas computing iterated matrix multiplication. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC 2014)*, pages 128–135, 2014. Full version in the Electronic Colloquium on Computational Complexity (ECCC), Technical Report TR13-100.

[For14] Michael Forbes. *Polynomial Identity Testing of Read-Once Oblivious Algebraic Branching Programs*. PhD thesis, Massachusetts Institute of Technology, June 2014.

[FS12] Michael A. Forbes and Amir Shpilka. On identity testing of tensors, low-rank recovery and compressed sensing. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC 2012)*, pages 163–172, 2012. Full version at arXiv:1111.0663.

[FS13a] Michael A. Forbes and Amir Shpilka. Explicit Noether Normalization for simultaneous conjugation via polynomial identity testing. In *Proceedings of the 17th International Workshop on Randomization and Computation (RANDOM 2013)*, pages 527–542, 2013. Full version at arXiv:1303.0084.

[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. Full version at arXiv:1209.2408.

[FSS14] Michael A. Forbes, Ramprasad Saptharishi, and Amir Shpilka. Hitting sets for multilinear read-once algebraic branching programs, in any order. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC 2014)*, pages 867–875, 2014. Full version at arXiv:1309.5668.

[GKKS13] Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Arithmetic circuits: A chasm at depth three. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2013)*, pages 578–587, 2013. Full version in the Electronic Colloquium on Computational Complexity (ECCC), Technical Report TR13-026.

[GKKS14] Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Approaching the chasm at depth four. *J. ACM*, 61(6):33:1–33:16, December 2014. Preliminary version in the *28th Annual IEEE Conference on Computational Complexity (CCC 2013)*.

[GKST15] Rohit Gurjar, Arpita Korwar, Nitin Saxena, and Thomas Thierauf. Deterministic identity testing for sum of read once ABPs. In *Proceedings of the 30th Annual IEEE Conference on Computational Complexity (CCC 2015)*, 2015. Full version at arXiv:1411.7341.

[Gre14a] Bruno Grenet. Computing low-degree factors of lacunary polynomials: a Newton-Puiseux approach. In *Proceedings of the 2014 International Symposium on Symbolic and Algebraic Computation (ISSAC 2014)*, pages 224–231, 2014. The full version of this work ([Gre14b]) contains simplified proofs of the original full version (arXiv:1401.4720).

[Gre14b] Bruno Grenet. Bounded-degree factors of lacunary multivariate polynomials. *arXiv*, 1412.3570, 2014.

[Gup14] Ankit Gupta. Algebraic geometric techniques for depth-4 PIT & Sylvester-Gallai conjectures for varieties. *Electronic Colloquium on Computational Complexity (ECCC)*, 21:130, 2014.

[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.

[Kal89] Erich L. Kaltofen. Factorization of polynomials given by straight-line programs. In Silvio Micali, editor, *Randomness and Computation*, volume 5 of *Advances in Computing Research*, pages 375–412. JAI Press, Inc., Greenwich, CT, USA, 1989.

[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.

[KI04] Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Computational Complexity*, 13(1-2):1–46, 2004. Preliminary version in the *35th Annual ACM Symposium on Theory of Computing (STOC 2003)*.

[KK05] Erich L. Kaltofen and Pascal Koiran. On the complexity of factoring bivariate supersparse (lacunary) polynomials. In *Proceedings of the 2005 International Symposium on Symbolic and Algebraic Computation (ISSAC 2005)*, pages 208–215, 2005.

[KK06] Erich L. Kaltofen and Pascal Koiran. Finding small degree factors of multivariate supersparse (lacunary) polynomials over algebraic number fields. In *Proceedings of the 2006 International Symposium on Symbolic and Algebraic Computation (ISSAC 2006)*, pages 162–168, 2006.

[KLSS14a] Neeraj Kayal, Nutan Limaye, Chandan Saha, and Srikanth Srinivasan. Superpolynomial lower bounds for depth-4 homogeneous arithmetic formulas. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC 2014)*, pages 119–127, 2014. Full version in the Electronic Colloquium on Computational Complexity (ECCC), Technical Report TR14-005.

[KLSS14b] Neeraj Kayal, Nutan Limaye, Chandan Saha, and Srikanth Srinivasan. An exponential lower bound for homogeneous depth four arithmetic formulas. In *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2014)*, pages 61–70, 2014. Full version in the Electronic Colloquium on Computational Complexity (ECCC), Technical Report TR14-005.

[KMSV13] Zohar Shay Karnin, Partha Mukhopadhyay, Amir Shpilka, and Ilya Volkovich. Deterministic identity testing of depth-4 multilinear circuits with bounded top fan-in. *SIAM J. Comput.*, 42(6):2114–2131, 2013. Preliminary version in the *42nd Annual ACM Symposium on Theory of Computing (STOC 2010)*.

[Koi12] Pascal Koiran. Arithmetic circuits: The chasm at depth four gets wider. *Theor. Comput. Sci.*, 448:56–65, 2012. Preliminary version at arXiv:1006.4700.

[KS01] Adam Klivans and Daniel A. Spielman. Randomness efficient identity testing of multivariate polynomials. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC 2001)*, pages 216–223, 2001.

[KS07] Neeraj Kayal and Nitin Saxena. Polynomial identity testing for depth 3 circuits. *Computational Complexity*, 16(2):115–138, 2007. Preliminary version in the *21st Annual IEEE Conference on Computational Complexity (CCC 2006)*.

[KS09] Neeraj Kayal and Shubhangi Saraf. Blackbox polynomial identity testing for depth 3 circuits. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2009)*, pages 198–207, 2009. Full version in the Electronic Colloquium on Computational Complexity (ECCC), Technical Report TR09-032.

[KS11] Zohar Shay Karnin and Amir Shpilka. Black box polynomial identity testing of generalized depth-3 arithmetic circuits with bounded top fan-in. *Combinatorica*, 31(3):333–364, 2011. Preliminary version in the *23rd Annual IEEE Conference on Computational Complexity (CCC 2008)*.

[KS14a] Mrinal Kumar and Shubhangi Saraf. The limits of depth reduction for arithmetic formulas: it's all about the top fan-in. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC 2014)*, pages 136–145, 2014. Full version at arXiv:1311.6716.

[KS14b] Mrinal Kumar and Shubhangi Saraf. Superpolynomial lower bounds for general homogeneous depth 4 arithmetic circuits. In *Proceedings of the 41st International Colloquium on Automata, Languages and Programming (ICALP 2014)*, pages 751–762, 2014. Full version at arXiv:1312.5978.

[KS14c] Mrinal Kumar and Shubhangi Saraf. On the power of homogeneous depth 4 arithmetic circuits. In *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2014)*, pages 364–373, 2014. Full version at arXiv:1404.1950.

[KS15] Neeraj Kayal and Chandan Saha. Lower bounds for depth three arithmetic circuits with small bottom fanin. In *Proceedings of the 30th Annual IEEE Conference on Computational Complexity (CCC 2015)*, 2015. Full version in the Electronic Colloquium on Computational Complexity (ECCC), Technical Report TR14-089.

[KSS14a] Neeraj Kayal, Chandan Saha, and Ramprasad Saptharishi. A super-polynomial lower bound for regular arithmetic formulas. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC 2014)*, pages 146–153, 2014. Full version in the Electronic Colloquium on Computational Complexity (ECCC), Technical Report TR13-091.

[KSS14b] 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)*, 2014. Full version in the Electronic Colloquium on Computational Complexity (ECCC), Technical Report TR14-001.

[KT90] Erich L. Kaltofen and Barry M. Trager. Computing with polynomials given by black boxes for their evaluations: greatest common divisors, factorization, separation of numerators and denominators. *J. Symb. Comput.*, 9(3):301–320, 1990. Preliminary version in the *29th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1988)*.

[KUW86] Richard M. Karp, Eli Upfal, and Avi Wigderson. Constructing a perfect matching is in random NC. *Combinatorica*, 6(1):35–48, 1986. Preliminary version in the *17th Annual ACM Symposium on Theory of Computing (STOC 1985)*.

[Lan12] Joseph M. Landsberg. *Tensors: geometry and applications*, volume 128 of *Graduate Studies in Mathematics*. American Mathematical Society, Providence, RI, 2012.

[Lov79] László Lovász. On determinants, matchings, and random algorithms. In *Fundamentals of computation theory (Proc. Conf. Algebraic, Arith. and Categorical Methods in Comput. Theory, Berlin/Wendisch-Rietz, 1979)*, volume 2 of *Math. Res.*, pages 565–574. Akademie-Verlag, Berlin, 1979.

[Mul12] 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. Full version at arXiv:1209.5993.

[MVV87] Ketan Mulmuley, Umesh V. Vazirani, and Vijay V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, 1987. Preliminary version in the *19th Annual ACM Symposium on Theory of Computing (STOC 1987)*.

[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.

[NW96] Noam Nisan and Avi Wigderson. Lower bounds on arithmetic circuits via partial derivatives. *Computational Complexity*, 6(3):217–234, 1996. Preliminary version in the *36th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1995)*.

[OSV15] Rafael Oliveira, Amir Shpilka, and Ben Lee Volk. Subexponential size hitting sets for bounded depth multilinear formulas. In *Proceedings of the 30th Annual IEEE Conference on Computational Complexity (CCC 2015)*, 2015. Full version at arXiv:1411.7492.

[Raz06] Ran Raz. Separation of multilinear circuit and formula size. *Theory of Computing*, 2(6):121–135, 2006. Preliminary version in the *45th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2004)*.

[Raz09] Ran Raz. Multi-linear formulas for permanent and determinant are of super-polynomial size. *J. ACM*, 56(2), 2009. Preliminary version in the *36th Annual ACM Symposium on Theory of Computing (STOC 2004)*.

[RS05] Ran Raz and Amir Shpilka. Deterministic polynomial identity testing in non-commutative models. *Comput. Complex.*, 14(1):1–19, April 2005. Preliminary version in the *19th Annual IEEE Conference on Computational Complexity (CCC 2004)*.

[RY09] Ran Raz and Amir Yehudayoff. Lower bounds and separations for constant depth multilinear circuits. *Computational Complexity*, 18(2):171–207, 2009. Preliminary version in the *23rd Annual IEEE Conference on Computational Complexity (CCC 2008)*.

[Sap14] Ramprasad Saptharishi. Recent progress on arithmetic circuit lower bounds. *Bulletin of the EATCS*, 114:76–118, 2014.

[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. Preliminary version in the Electronic Colloquium on Computational Complexity (ECCC), Technical Report TR07-124.

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

[Sax14] Nitin Saxena. Progress on polynomial identity testing - II. *arXiv*, 1401.0976, 2014.

[Sch80] J. T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, October 1980. Preliminary version in the *International Symposium on Symbolic and Algebraic Computation (EUROSAM 1979)*.

[Sho09] Victor Shoup. *A computational introduction to number theory and algebra*. Cambridge University Press, Cambridge, second edition, 2009.

[SS11] Nitin Saxena and C. Seshadhri. An almost optimal rank bound for depth-3 identities. *SIAM J. Comput.*, 40(1):200–224, 2011. Preliminary version in the *24th Annual IEEE Conference on Computational Complexity (CCC 2009)*.

[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. Preliminary version in the *43rd Annual ACM Symposium on Theory of Computing (STOC 2011)*.

[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, 2013. Preliminary version in the *51st Annual IEEE Symposium on Foundations of Computer Science (FOCS 2010)*.

[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. Preliminary version in the Electronic Colloquium on Computational Complexity (ECCC), Technical Report TR11-021.

[Str73] Volker Strassen. Vermeidung von divisionen. *J. Reine Angew. Math.*, 264:184–202, 1973.

[SV09] Amir Shpilka and Ilya Volkovich. Improved polynomial identity testing for read-once formulas. In *Proceedings of the 13th International Workshop on Randomization and Computation (RANDOM 2009)*, pages 700–713, 2009. Full version in the Electronic Colloquium on Computational Complexity (ECCC), Technical Report TR10-011.

[SV10] Amir Shpilka and Ilya Volkovich. On the relation between polynomial identity testing and finding variable disjoint factors. In *Proceedings of the 37th International Colloquium* *on Automata, Languages and Programming (ICALP 2010)*, pages 408–419, 2010. Full version in the Electronic Colloquium on Computational Complexity (ECCC), Technical Report TR10-036.

[SV11] Shubhangi Saraf and Ilya Volkovich. Black-box identity testing of depth-4 multilinear circuits. In *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing (STOC 2011)*, pages 421–430, 2011. Full version in the Electronic Colloquium on Computational Complexity (ECCC), Technical Report TR11-046.

[SY10] Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science*, 5(3-4):2070–388, 2010.

[Tav13] Sébastien Tavenas. Improved bounds for reduction to depth 4 and depth 3. In *Proceedings of the 38th Internationl Symposium on the Mathematical Foundations of Computer Science (MFCS 2013)*, pages 813–824, 2013. Full version at arXiv:1304.5777.

[Vol14] Ilya Volkovich. Deterministically factoring sparse polynomials into multilinear factors. *Electronic Colloquium on Computational Complexity (ECCC)*, 21:168, 2014.

[VSBR83] Leslie G. Valiant, Sven Skyum, Stuart J. Berkowitz, and Charles Rackoff. Fast parallel computation of polynomials using few processors. *SIAM J. Comput.*, 12(4):641–644, 1983. Preliminary version in the *6th Internationl Symposium on the Mathematical Foundations of Computer Science (MFCS 1981)*.

[vzGK85] Joachim von zur Gathen and Erich L. Kaltofen. Factoring sparse multivariate polynomials. *J. Comput. Syst. Sci.*, 31(2):265–287, 1985. Preliminary version in the *24th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1983)*.

[Zip79] Richard Zippel. Probabilistic algorithms for sparse polynomials. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation (EUROSAM 1979)*, pages 216–226. Springer-Verlag, 1979.

A Binomial Estimates

We now proceed to give the needed estimates for ratios of binomial coefficients in the above results. Kayal [Kay12] gave such estimates using some methods of Gupta, Kayal, Kamath, and Saptharishi [GKKS14]. For completeness we give an analysis using a slight refinement of the methods of Gupta, Kayal, Kamath and Saptharishi [GKKS14]. In particular, they gave a way to split this ratio of binomial coefficients into a product of ratios of similar factorials. They then bounded these ratios of factorials via the following lemma (for which we quote a non-asymptotic version).

Lemma A.1 (Gupta, Kayal, Kamath and Saptharishi [GKKS14, Lemma 6.2])**.** *Let $n \geq 1$ and $i, j \geq 0$ with $j \leq n/2$,*

$$\left|\ln \frac{(n+i)!}{(n-j)!} - (i+j)\ln n\right| \leq 2\frac{(i+j)^2}{n} . \qquad \square$$

While the above lemma would work in our situation, we find it more convenient to use the following weaker version as it does not have any error term.

Lemma A.2. *For $n \geq 1$ and $i, j \geq 0$ with $n - j \geq 1$,*

1. $\ln \frac{(n+i)!}{(n-j)!} \leq (i+j)\ln(n+i)$.
2. $\ln \frac{(n+i)!}{(n-j)!} \geq (i+j)\ln(n-j)$.

Proof: (1):

$$\begin{aligned}\frac{(n+i)!}{(n-j)!} &= \prod_{k=n-j+1}^{n+i} k \\ &\leq \prod_{k=n-j+1}^{n+i} n+i \\ &= (n+i)^{(n+i)-(n-j+1)+1} = (n+i)^{i+j} ,\end{aligned}$$

and taking logarithms yields the claim.

(2): As above, but now use that $k \geq n - j$. $\square$

We remark that we could use that $k \geq n - j + 1$ here to obtain a slightly sharper estimate. We now turn to the desired binomial estimate.

Lemma A.3 (Kayal [Kay12], Gupta-Kamath-Kayal-Saptharishi [GKKS14])**.** *Let $n, t \geq 1$. Then*

$$\ln \frac{\binom{n}{k}\binom{n-k+\ell}{\ell}}{\binom{n+\ell+(t-1)k}{\ell+(t-1)k}} \geq k ,$$

for $\ell = (t-1)(n + (t-1)k) \in \mathbb{N}$ and $k = \epsilon n/t \in \mathbb{N}$, whenever $0 \leq \epsilon \leq \frac{1}{e^2+1} < 1$. In particular, $0 \leq \epsilon \leq 1/e^3$ suffices.

Proof: $k = 0$: Then the inequality reduces to "$\ln \binom{n+\ell}{\ell}/\binom{n+\ell}{\ell} = \ln 1 \geq 0$", which holds.

$t = 1$: Then $\ell = 0$, and the inequality reduces to "$\ln \binom{n}{k} \geq k$", which holds as $\ln \binom{n}{k} \geq k \ln n/k$ and $k = \epsilon n/t \leq \frac{n}{e^2+1} \leq n/e$ so that $\ln n/k \geq 1$.

<u>$k \geq 1$, $t \geq 2$:</u>

$$\ln \frac{\binom{n}{k}\binom{n-k+\ell}{\ell}}{\binom{n+\ell+(t-1)k}{\ell+(t-1)k}} = \ln \left[\binom{n}{k} \cdot \frac{(n+\ell-k)!}{\ell!(n-k)!} \cdot \frac{n!(\ell+(t-1)k)!}{(n+\ell+(t-1)k)!} \right]$$

regrouping terms as in Gupta, Kayal, Kamath and Saptharishi [GKKS14], noting that all expressions here are positive integers,

$$= \ln \binom{n}{k} + \ln \frac{n!}{(n-k)!} + \ln \frac{(\ell+(t-1)k)!}{\ell!} - \ln \frac{(n+\ell+(t-1)k)!}{(n+\ell-k)!}$$

using that $\ln \binom{n}{k} \geq k \ln {}^{n}\!/_{k}$, and then using Lemma A.2 to approximate the other three terms (as we have positive integers), the first around n, the second around ℓ and the third around $n+\ell$,

$$\begin{aligned} &\geq k \ln \frac{n}{k} + k \ln(n-k) + (t-1)k \ln \ell - tk \ln(n+\ell+(t-1)k) \\ &= k \ln \left(\frac{n}{k} \cdot (n-k) \cdot \frac{\ell^{t-1}}{(n+\ell+(t-1)k)^t} \right) \end{aligned}$$

By differentiation we see that $\ell = (t-1)(n+(t-1)k)$ maximizes this expression, justifying this choice. Thus, taking ℓ with this value we see that $n+\ell+(t-1)k = t(n+(t-1)k)$, and thus,

$$= k \ln \left(\frac{n(n-k)}{k \cdot t(n+(t-1)k)} \cdot \left(\frac{(t-1)(n+(t-1)k)}{t(n+(t-1)k)} \right)^{t-1} \right)$$

using that $1+x \leq \mathrm{e}^x$ for $x \in \mathbb{R}$ yields that $\left(1+\frac{1}{t-1}\right)^{t-1} \leq \mathrm{e}$, and taking the reciprocal yields $\left(\frac{t-1}{t}\right)^{t-1} \geq {}^{1}\!/_{\mathrm{e}}$,

$$= k \ln \left(\frac{n(n-k)}{k \cdot t(n+(t-1)k)} \cdot \frac{1}{\mathrm{e}} \right)$$

taking that $k = {}^{\epsilon n}\!/_{t}$,

$$\begin{aligned} &= k \ln \left(\frac{1}{\epsilon \mathrm{e}} \cdot \left(1 - \frac{tk}{n+(t-1)k} \right) \right) = k \ln \left(\frac{1}{\epsilon \mathrm{e}} \cdot \left(1 - \frac{t}{{}^{t}\!/_{\epsilon} + (t-1)} \right) \right) \\ &\geq k \ln \left(\frac{1}{\epsilon \mathrm{e}} \cdot (1-\epsilon) \right) \end{aligned}$$

using that $\frac{1}{\epsilon \mathrm{e}}(1-\epsilon) \geq \mathrm{e}$ for $\epsilon \leq \frac{1}{\mathrm{e}^2+1}$,

$$\geq k \,. \qquad \square$$

We remark that the $t = 1$ analysis need not actually be separate. That is, the only obstruction in the above is that $t = 1$ forces $\ell = 0$, which formally prohibits the application of Lemma A.2. However, there is a sharper version of this result, as remarked after Lemma A.2 indicates, that would allow one to make the analysis go through but this makes the proof more technical.

We now proceed to combine this bound with others to apply it to the ratios at hand. First, we will use a simple bound of binomial coefficients.

Lemma A.4. *For $k, m \in \mathbb{N}$,*

$$\binom{k+m}{m} \le (k+1)^m \ .$$

Proof:

$$\binom{k+m}{m} = \frac{(k+m)!}{m!k!} = \prod_{i=1}^{m} \frac{k+i}{i} = \prod_i \left(\frac{k}{i}+1\right) \le \prod_i (k+i) = (k+1)^m \ . \qquad \square$$

Note that one can also see this inequality by noting that $\binom{k+m}{m}$ counts the number monomials of degree $\le k$ in m variables, and this is less than the number of monomials in m variables of individual degree $\le k$, of which there are $(k+1)^m$.

We now turn to an inequality bounding polynomials by exponentials.

Lemma A.5. *Let $x \ge 0$ and $m \in \mathbb{N}$. Then*

$$\mathrm{e}^{x/2} \ge \frac{(1+x)^m}{\sqrt{\mathrm{e}}(2m)^m} \ .$$

Proof: For $y \ge 0$,

$$\mathrm{e}^y = \sum_{i=0}^{\infty} \frac{y^i}{i!} \ge \frac{y^{2m}}{(2m)!} \ge \left(\frac{y}{2m}\right)^{2m} \ ,$$

where the first statement is by definition, the second uses that $y \ge 0$, and the third uses that $k! \le k^k$. Taking square roots thus yields that $\mathrm{e}^{y/2} \ge ({y}/{2m})^m$. Replacing $y = 1+x$, so that $x \ge 0$ means $y \ge 0$ so the above applies, we see that $\mathrm{e}^{x/2}\sqrt{\mathrm{e}} \ge ({(1+x)}/{2m})^m$, which yields the result. $\square$

We now put the above lemmas together to obtain the main estimate that we need.

Lemma A.6. *Let $n, t, m \ge 1$. Suppose that*

$$s \ge \frac{1}{\binom{k+m}{m}} \cdot \frac{\binom{n}{k}\binom{n-k+\ell}{\ell}}{\binom{n+\ell+(t-1)k}{\ell+(t-1)k}} \ ,$$

for all $\ell, k \in \mathbb{N}$. Then

1. *$s \ge \frac{\exp({n}/{(2\mathrm{e}^3 t)})}{\mathrm{e}(2m)^m}$. In particular, for $m = O(1)$, $s \ge \exp(\Omega({n}/{t}))$.*
2. *$n \le 2\mathrm{e}^3 t(\ln s + m\ln(2m) + 1)$. In particular, for $m = O(1)$, $n \le \mathcal{O}(t \ln s)$.*

Proof: (1): Pick $k = \left\lfloor \frac{n}{\mathrm{e}^3 t} \right\rfloor$ and $\ell = (t-1)(n+(t-1)k)$. Thus, respectively appealing to Lemma A.4, Lemma A.3, Lemma A.5, and $\lfloor x \rfloor \geq x - 1$,

$$\begin{aligned}
s &\geq \frac{1}{\binom{k+m}{m}} \cdot \frac{\binom{n}{k}\binom{n-k+\ell}{\ell}}{\binom{n+\ell+(t-1)k}{\ell+(t-1)k}} \\
&\geq \frac{1}{(k+1)^m} \cdot \frac{\binom{n}{k}\binom{n-k+\ell}{\ell}}{\binom{n+\ell+(t-1)k}{\ell+(t-1)k}} \\
&\geq \frac{1}{(k+1)^m} \cdot \mathrm{e}^k \\
&\geq \frac{1}{\sqrt{\mathrm{e}}(2m)^m}\mathrm{e}^{-k/2} \cdot \mathrm{e}^k \\
&\geq \frac{1}{\sqrt{\mathrm{e}}(2m)^m} \exp\left(\frac{\frac{n}{\mathrm{e}^3 t} - 1}{2}\right) \\
&= \frac{\exp(n/(2\mathrm{e}^3 t))}{\mathrm{e}(2m)^m}
\end{aligned}$$

(2): This follows from (1) immediately, as follows.

$$\begin{aligned}
\exp(n/(2\mathrm{e}^3 t)) &\leq \mathrm{e}(2m)^m s \\
n/(2\mathrm{e}^3 t) &\leq \ln s + m \ln(2m) + 1 \\
n &\leq 2\mathrm{e}^3 t(\ln s + m\ln(2m) + 1)\ .
\end{aligned}$$

□