Small hitting-sets for tiny algebraic circuits or: How to turn bad designs into good

Manindra Agrawal * Michael A. Forbes† Sumanta Ghosh ‡ Nitin Saxena §

Abstract

Research in the last decade has shown that to prove lower bounds or to derandomize polynomial identity testing (PIT) it suffices to solve these questions for restricted circuits. In this work, we study the possibly most restricted class of circuits, within depth-4, which would yield such results for general circuits (that is, the complexity class VP). We show that if, for some $\mu$, there is a poly$(s, 2^{\mu n}, \mu(a))$-time blackbox PIT for size-$s$ $\Sigma \land^a \Sigma \Pi^b$, where the number of variables, or arity, is $n$, then blackbox PIT for VP is in QuasiP. Further, in a strengthening of (Kabanets & Impagliazzo, STOC’03), the former algorithm implies that either $E \not\subseteq \#P/poly$ or that VNP requires size-$2^{\Omega(n)}$ VP circuits. Parameters $a, b$ & $n$, for which the time-complexity above is poly$(s)$, define the model of tiny diagonal depth-4. Note that these are merely polynomials with arity $O(\log s)$ whose degree $ab$ is arbitrary close to $\log s$. In fact, we show that one only needs, for some $\mu'$, a poly$(s, 2^{\mu}(a'))$-time blackbox PIT for individual-degree-$a'$ arity-$n$ homogeneous polynomials computed by a size-$s$ depth-3 circuit. Alternatively, we claim that, to understand VP we barely need a fixed-parameter tractable (FPT) blackbox PIT for depth-3 (parameters being arity & individual-degree).

Almost any FPT PIT is welcome: We show that if, for some $\mu$, there is a poly$(s, \mu(n))$-time blackbox PIT for size-$s$ arity-$\Sigma \Pi \Sigma \land$ circuits (resp. $\Sigma \land^n \Sigma \Pi$), then blackbox PIT for VP is in QuasiP, and one proves the claimed lower bound. (Fig.1 lists more results.)

Finally, our methods prove a stunning arity reduction for PIT: to solve the general problem in poly$(sd)$-time it suffices to find a blackbox PIT with poly$(sd, \exp^{c2}(n))$-time, where $\exp^{c2}$ refers to a $c = O(1)$ fold composition of $\exp$ (eg. $\exp^{5}(n) := 2^{2^{2^n}}$). This suggests that, in algebraic-geometry terms, PIT is an ‘extremely low’ dimensional problem.

One expects that with this severe restriction (or tinyness) on $n, a, b$ and the semantic individual-degree, it should be “super-exponentially” easier to design hitting-sets. Indeed, we give several examples of $(\log s)$-variate circuits where a new measure (called cone-size) helps in devising poly$(s)$-time hitting-sets, but the same question for their $s$-variate versions is open till date: For eg., diagonal depth-3 circuits, and in general, models that have a small partial derivative space. The latter models are very well studied, following (Nisan & Wigderson, FOCS’95), but no $sd^{2\Omega(n)}$-time PIT algorithm was known before us.

We introduce a novel concept, called cone-closed basis isolation, and provide example models where it occurs, or can be achieved by a small shift. This refines the previously studied notions of low-support (resp. low-cone) rank concentration and least basis isolation in certain ABP models. Cone-closure holds special relevance in the low-arity regime.


Keywords: hitting-set, tiny, arity, depth-3, depth-4, derandomization, identity testing, lower bound, VP, VNP, E, #P/poly, SUBEXP, circuit, concentration, NW design, circuit factoring.

*Department of Computer Science & Engineering, IIT Kanpur, India, manindra@cse.iitk.ac.in
†Simons Institute for the Theory of Computing, University of California, Berkeley, USA, miforbes@csail.mit.edu
‡CSE, IITK, sumghosh@cse.iitk.ac.in
§CSE, IITK, nitin@cse.iitk.ac.in
1 Introduction

The Polynomial Identity Testing (PIT) problem is to decide whether a multivariate polynomial is zero, where the input is given as an algebraic circuit. An algebraic circuit over a field \( \mathbb{F} \) is a layered acyclic directed graph with one sink node called output node; source nodes are called input nodes and are labeled by variables or field constants; non-input nodes are labeled by \( \times \) (multiplication gate) and \( + \) (addition gate) in alternate layers. Sometimes edges may be labeled by field constants. The computation is defined in a natural way. The complexity parameters of a circuit are: 1) size- maximum number of edges and vertices, 2) depth- maximum number of layers, and 3) degree- maximum degree among all polynomials computed at each node. This is sometimes called the syntactic degree, to distinguish from the (semantic) degree of the final polynomial computed. The families of circuits, that are \( n \)-variate \( \text{poly}(n) \)-size and \( \text{poly}(n) \)-degree, define the class VP [Val79]; also see [Bir13] for interesting variants of this algebraic computing model.

In this work we study \( n \)-variate polynomials computable by circuits of size \( \leq s \) of individual degree \( \text{ideg} \leq a \), where one of the parameters is tiny as compared to the others. For example, we study such polynomials where the number of variables \( n \) is very small, such as \( n \leq O(\log s) \). Alternatively, to be consistent with VP definition, one could think of a circuit where the output depends only on the first \( \leq O(\log s) \) variables out of the possible \( \{x_1, \ldots, x_s\} \). When \( a \leq O(1) \) this model is equivalent to that of \( \text{poly}(s) \)-size depth-2 circuits, which are well-understood even in the blackbox model [BOT88], and thus we probe these polynomials when \( a \geq \omega(1) \). We can even approach this question when \( n \) is \( \omega(1) \) but arbitrarily small, in which case we now consider polynomials of individual degree \( O(s) \) to again avoid collapsing to \( \text{poly}(s) \)-size depth-2 circuits. Basically, we need to allow the number of monomials \( a^n \) to grow as \( s^{\omega(1)} \) for the model to be nontrivial; and we demonstrate that this is precisely the chasm to be crossed to get a fundamentally new, & almost complete, understanding of VP. See Fig.1 for the list of models.

The polynomial computed by a circuit may have, in the worst-case, an exponential number of monomials compared to its size. So, by computing the explicit polynomial from input circuit, we cannot solve PIT problem in polynomial time. However, evaluation of the polynomial at a point can be done, in time polynomial in the circuit size, by assigning the values at input nodes. This helps us to get a polynomial time randomized algorithm for PIT by evaluating the circuit at a random point, since any nonzero polynomial evaluated at a random point gives a nonzero value with high probability [DL78] [Zip79] [Sch80]. However, finding a deterministic polynomial time algorithm for PIT is a long-standing open question in algebraic complexity theory. It naturally appears in the algebraic approaches to the \( P \neq \text{NP} \) question, eg. [GMQ16] [Gro15] [Mul12b] [Mul12a]. The famous algebraic analog is the \( \text{VP} \neq \text{VNP} \) question [Val79]. The PIT problem has applications both in proving circuit lower bounds [KI03] [Agr05] and in algorithm design [MVV87] [AKS04] [KSS14] [DdOS14]. For more details on PIT, see the surveys [Sax09] [Sax13] [SY10].

PIT algorithms are of two kinds: 1) whitebox- allowed to see the internal structure of the circuit, and 2) blackbox- only evaluation of the circuit is allowed at points in a small field extension. Blackbox PIT is equivalent to efficiently finding a set of points, called a hitting-set \( \mathcal{H} \), such that for any circuit \( C \), in a family \( \mathcal{C} \), computing a nonzero polynomial, the set \( \mathcal{H} \) must contain a point where \( C \neq 0 \). For us a more functional approach would be convenient. We think in terms of an \( n \)-tuple of univariates \( f(y) = (f_1(y), \ldots, f_n(y)) \) whose set of evaluations contain \( \mathcal{H} \). Such an \( f(y) \) can be efficiently obtained from a given \( \mathcal{H} \) (using interpolation) and vice-versa. Clearly, if \( \mathcal{H} \) is a hitting-set for \( \mathcal{C} \) then \( C(f(y)) \neq 0 \), for any nonzero \( C \in \mathcal{C} \). This tuple of univariates is called a hitting-set generator (Sec.2) and we will freely exploit this connection in all the proofs or theorem statements. A takeaway from our work is that even extremely tiny ‘local patches’ of a hitting-set generator bear tremendous ‘information’ (Thm.2 Fig.2 Meta-Thm.1):
a property that reminds one of fractals in physical sciences [Man77].

Existing deterministic algorithms solving PIT for restricted classes have been developed by leveraging insight into the weaknesses of these models. For example, deterministic PIT algorithms are known for subclasses of depth-3 circuits [KS07, Sax08, SS12], subclasses of depth-4 circuits [ASSS12, BMS13, SSS13, For15, KS16a, KS16b, PSS16], read-once algebraic branching programs (ROABP) and related models [FS12, ASS13, SSS14, AGKS15, GKS16, GKS16], certain types of symbolic determinants [FGT16, GT16], as well as non-commutative models [LMP16, GGOW16]. An equally large number of special models have been used to prove lower bounds, see for example the ongoing online survey of Saptharishi [Sap16], and there are barriers conjectured [FSV17].

While studying such restricted models may at first seem to give limited insight into general circuits, various works (discussed below) have shown this not to be the case as full derandomization of PIT for depth-4 (resp. depth-3) circuits would imply derandomization of PIT for general circuits. The goal of this work is to sharpen this connection by additionally limiting the number of variables (resp. semantic individual-degree) in the depth-4 circuit, and showing that such a connection still holds. In doing so we establish new concepts for studying this small-variable regime, and show how to derive polynomial-size hitting sets for some small-variable circuit classes where only quasipolynomial-size, but not poly-sized, hitting-sets were previously known.

1.1 Main results

Algebraic circuits were defined with the hope that they would have better structure than boolean circuits. Indeed, unlike boolean circuits, any VP circuit of arbitrary depth can be reduced nontrivially to depth-4 [AV08, Koi12, Hay15, CKSV16] or depth-3 [GKKS13]. As a consequence, the lower bound questions against VP reduce to the lower bound questions for depth-4 (or to depth-3 for selected fields). In circuit complexity the base field \( F \) of interest is either \( \mathbb{Q} \) or \( F_q \) (for a prime-\( p \)-power \( q \)). Other popular fields, eg. number field, function field or \( p \)-adic field, are dealt with using similar computational methods. In this paper, unless stated otherwise, we assume \( F = \mathbb{Q} \). (Though many of our ideas would generalize to other base rings.)

The PIT question for VP circuits reduces even more drastically. The reason is that now one invokes circuit factorization results [Kal89] that use algebra in a way heavier than the depth-reduction results. So we will invoke that VP is closed under factorization, in addition to the fact that it affords depth-reduction. Recall the \( \Sigma \Pi \Sigma \Pi \) (resp. \( \Sigma \land \Sigma \Pi \)) model that computes a polynomial by summing products (resp. powers) of sparse polynomials (see Defn. 12). In 2008, Agrawal and Vinay [AV08, Thm.3.2] showed that solving blackbox PIT in poly\((s)\)-time for size-\( s \)-s-variate depth-4 circuits of the form \( \Sigma_1 \Pi_1 \Sigma_2 \Pi_2 \Sigma_2 \Pi_2 \Sigma_3 \Pi_3 \) (Defn. 12), where \( a \) is some small unbounded function, gives an \( (sd)^{O(\log s)} \)-time hitting-set for VP (size-\( s \)-degree \( d \)).

Here, we weaken the hypothesis further. We show that solving blackbox PIT in poly\((s)\)-time for size-\( s \) \( O(\log s) \)-variate \( \Sigma \land \Sigma \Pi \Pi O(\log s) \) circuits, where \( a = a(s) \) and semantic individual-degree \( a'(s) \) are some arbitrarily small unbounded functions, is sufficient to get an \( (sd)^{O(\log sd)} \)-time hitting-set for VP. (We could handle a smaller bottom-fanin than \( \log s \).) We note that the brute-force deterministic algorithm would run here in time \( a'^{O(\log s)} = s^{O(\log a')} \), and thus we show that reducing this runtime to polynomial would have dramatic consequences. We call such depth-4 circuits as tiny diagonal depth-4 (a better definition is Defn. 12). Compared to the previous result, one of the many advantages in our model is that an exponential running time, wrt the number of variables (arity \( n \)), is allowed. Formally, we design an efficient arity-reducing polynomial-map (the polynomials designed have individual-degree \( O(1) \)). Clearly, the map can be used to also deduce about the quasipoly-time blackbox PIT for VP.

**Theorem 1.** Suppose we have poly-time hitting-set for some tiny diagonal depth-4 model. Then, we design a poly\((sd)\)-time arity reducing \( (n \mapsto O(\log sd)) \) polynomial-map of constant
individual-degree that preserves the nonzeroness of any \(n\)-variate size-\(s\) degree-\(d\) algebraic circuit. Furthermore, we get an \(E\)-computable homogeneous polynomial with algebraic circuit size \(2^{O(n)}\).

By the known depth-3 chasm \[\text{GKKS13}\], the hypothesis in Thm.\[1\] can be weakened to: if, for some \(\mu\), we have a \(\text{poly}(s, 2^n, \mu(a'))\)-time hitting-set for size-\(s\) arity-\(n\) depth-3 circuits that compute polynomials of semantic individual-degree \(\leq a'\), then \(\cdots\). For this the proof sketch is given in Sec.\[A\] (Thm.\[14\]), where also the case of a tiny width-2 ABP (Thm.\[15\]) and a ‘multilinear tiny’ depth-3 variant (Thm.\[16\]) are discussed. Also, see Fig.\[1\] and the discussion in Sec.\[B\] to get a quick overview of the gamut of weak hypotheses that we can deal with.

Note that the sparsity of a polynomial computed by tiny diagonal depth-4 is \(a'^n = s^{O(\log a')}\) which gives us a brute-force hitting-set of similar complexity \[\text{BOT88}\]. We want to bring it down to \(s^{O(1)}\); leaving us with an arbitrarily small gap to close algorithmically. Our methods show that any of these hitting-set designs will establish: Either \(E\not\subseteq \#P/poly\) or \(VNP\) has polynomials of algebraic circuit complexity \(2^{O(n)}\) (Lem.\[13\], Cor.\[7\]). Note that these are long-standing open questions \[\text{NW94, Val79}\]. Their connection with PIT, in our results, is a significant strengthening of the conclusion of \[\text{K103, Thm.18}\] who had first proved the flip: PIT in \(\text{NSUBEXP}\) implies that either \(\text{NEXP} \not\subseteq \#P/poly\) or \(VNP \not\subseteq \text{VP}\). Our methods yield stronger lower bound conclusions because we use the poly-time hitting-set to \textit{directly} get a hard polynomial. (We discuss our conclusions, if merely a subexp-time hitting-set is known, in Sec.\[B\].)

Moreover, we get the following stunning property of blackbox PIT. (In some sense, it signifies that “extremely” tiny-VP PIT implies VP PIT.) Let \(\text{exp}^{oc}(n)\) denote the \(c\)-fold composition of exponentiation over base \(2\). Eg. \(\text{exp}^{o0}(n) = n\), \(\text{exp}^{o1}(n) = 2^n\) and \(\text{exp}^{o2}(n) = 2^{2^n}\). Similarly, define its inverse function \(\log^{oc}(n)\), so that \(\text{exp}^{oc}(\log^{oc}(n)) = n\).

**Theorem 2 (PIT arity reduction).** If, for some \(c \in \mathbb{N}\), we have \(\text{poly}(sd, \text{exp}^{oc}(n))\)-time hitting-set for size-\(s\) degree-\(d\) arity-\(n\) circuits, then for general circuits we have a \(\text{poly}(sd)\)-time hitting-set (and we get an \(E\)-computable homogeneous polynomial with exponential algebraic complexity).

A proof outline is in Fig.\[2\]. As an aside, if one only cares about a \textit{super-polynomial} lower bound consequence, then we suggest in Sec.\[B\] that one only needs to find a hitting-set for VP such that the time-complexity is \textit{subexp} in \(sd\) and exponential in \(n\). See Meta-Thm.\[1\]

One now wonders whether the hypothesis, in the theorem above, can be further weakened in terms of the arity \(n\)? We give a partial answer by studying the model \(\Sigma \Pi \Sigma \land\) (i.e. a sum of products, where each factor is a sum of univariate polynomials).

**Theorem 3 (Tinier arity).** If, for some \(\mu\), we have a \(\text{poly}(s, \mu(n))\)-time hitting-set for size-\(s\) arity-\(n\) \(\Sigma \Pi \Sigma \land\) circuits, then for VP circuits we have a \(\text{poly}(sd)\)-time arity reduction \((n \mapsto O(\log sd))\) that preserves nonzeroness (and proves an exponential lower bound).

The PIT algorithms in current literature always try to achieve a subexponential dependence on \(n\), the number of variables. Our results demonstrate that all we need is a \(\text{poly}(sd, 2 \cdot 2^n)\)-time algorithm (with constant length 2-tower) for size-\(s\) degree-\(d\) circuits, to \textit{completely} solve VP PIT. Or, a \(\text{poly}(s\mu(n))\)-time algorithm (for \(\Sigma \Pi \Sigma \land\)) to “partially” solve VP PIT and to prove “either \(E\not\subseteq \#P/poly\) or \(VNP \not\subseteq \text{VP}\).” For example, even a \(\text{poly}(s, A(n))\)-time hitting-set for \(\Sigma \Pi \Sigma \land\), where \(A\) is an Ackermann function \[\text{Ack28}\], would be tremendous progress. A similar case can be made for \(\Sigma \land^n \Sigma \Pi(n)\) circuits, where both \(a\) and \(n\) are arbitrarily small unbounded functions, see Thm.\[18\] (i.e. time-complexity may be arbitrary in terms of both \(a\) and \(n\)). Also, see Fig.\[1\].

Obviously, we should now discover techniques and measures that are specialized to this tiny regime. Many previous works use support size of a monomial as a measure to achieve rank concentration \[\text{ASS13, FSS14, GKST16}\]. For a monomial \(m\), its support is the set of variables whose exponents are positive. We introduce a different measure: \textit{cone-size} (see Defn.\[19\]) which
is the number of monomials that divide \( m \) (also see [ASS13, For14]). It has two advantages in the tiny regime. First, the number of monomials with cone-size at most \( s \) is poly\((s)\) (Lem.21). Second, for any circuit \( C \) and a monomial \( m \), we devise (in blackbox) a circuit \( C' \) which computes the coefficient of \( m \) in \( C \) and has size polynomial in that of \( C \) and the cone-size of \( m \) (Lem.20).

Using this measure we can define a new concept of rank concentration [ASS13] – ‘low’-cone concentration – and we are able to give poly-time hitting-sets for a large class of tiny circuits (i.e. \( n \) is logarithmic wrt size). We prove our result in a general form (Thm.4) and as a corollary (Cor.22) we get our claim. This gives us a poly-time hitting-set for depth-3 diagonal circuits where the rank of the linear forms is logarithmic wrt the size (Thm.23).

**Theorem 4.** Let \( C \) be a set of arity-\( n \) degree-\( d \) circuits with size-\( s \) s.t. for all \( C \in C \), the dimension of the partial derivative space of \( C \) is at most \( k \). Then, blackbox PIT for \( C \) can be solved in \((sdk)^{O(1)} \cdot (3n/\log k)^{O(\log k)}\) time.

Note that for \( n = O(\log k) = O(\log sd) \), the above bound is poly-time and such a PIT result was not known before. For instance, general diagonal depth-3 is a prominent model with low partial derivative space; it has a whitebox poly-time PIT [Sax08] but no poly-time hitting-set is known (though [FSS14] gave an \( s^{O(\log \log s)} \)-time hitting-set.). Even for \( O(\log k) \)-variate diagonal depth-3 no poly-time hitting-sets were known before our work.

We investigate another structural property useful in the tiny regime. Consider a polynomial \( f(x) \) with coefficients over \( \mathbb{F}_k \). Let \( sp(f) \) be the subspace spanned by its coefficients. We say that \( f \) has a cone-closed basis if there is a set of monomials \( B \) whose coefficients form a basis of \( sp(f) \) and if \( B \) is closed under submonomials. We prove that this notion is a strengthening of both low-support [ASS13] and low-cone concentration ideas [For15] (see Lem.21). Thus, if we can achieve it efficiently, then we will get hitting-sets with the desirable time-complexity \( sd^{O(n)} \). Recently, this notion of closure has also appeared as an abstract simplicial complex in [GKPT16].

Interestingly, we show that a general polynomial \( f \), when shifted by a ‘random’ amount (or by ‘formal’ variables), becomes cone-closed. More generally, we prove the following theorem relating this concept to that of basis isolating weight assignment [AGKS15].

**Theorem 5.** Let \( f(x) \in \mathbb{F}[x]^k \) be an arity-\( n \) degree-\( d \) polynomial over \( \mathbb{F}_k \). Let \( w \) be a basis isolating weight assignment of \( f(x) \). Then \( f(x + tw) \) has a cone-closed basis over \( \mathbb{F}(t) \).

### 1.2 Proof ideas

The proof of Thm.4 is a technical refinement of the strategy of [AV08, Thm.3.2] in at least four ways. Namely, making the parameters \( n, a', b \) smaller and use of powering gates. First, we get a homogeneous polynomial \( g \) that is hard for tiny diagonal depth-4 circuits \( C \) from the hitting-set for \( C \) (Lem.6). Essentially, \( g \) will be an annihilating polynomial of the hitting-set generator. The novelty here is that we have to allow \( g \) to be non-multilinear (unlike [AV08, Lem.3.3]), for it to exist, as the arity of \( C \) in our case is logarithmically small. We show that a multi-\( \delta \)-ic \( g \) suffices (i.e. \( g \) has individual-degree bounded by a constant \( \delta \)). By VP depth-reduction to depth-4, and Fischer’s trick (that is ‘cheap’ to apply in the tiny regime), this polynomial remains hard for VP. Next by Lem.10 where we use Nisan-Wigderson design [NW94] and Kaltofen’s factorization [Kal89], we get a poly-time arity-reducing polynomial-map based on \( g \) that keeps any nonzero VP circuit nonzero and reduces its arity from \( n \) to \( O(\log sd) \). This gives the theorem.

The ‘log’ function in our method comes from the use of Nisan-Wigderson design and because we want the \( \delta \) to be constant in depth-reduction to depth-4; reducing the arity further would need a new idea. Moreover, we observe that the annihilator \( g = g_n \) is an E-computable (i.e. \( 2^{O(n)} \)-time) polynomial with exponential algebraic circuit complexity. This means that: Either \( \mathbf{E} \neq \mathbf{P}/\text{poly} \) or VNP has a polynomial with exponential algebraic complexity (Lem.13). It is not clear whether
we can strengthen the connection all the way to the (conjectured) VNP \( \not\equiv \) VP. Perhaps, this will require starting with a more structured hitting-set generator for \( C \), so that its annihilator \( g \) is a polynomial whose coefficient bits are \((\#P/poly)\)-computable (see Valiant’s criterion \cite{B13}, Prop.2.20). Numerous examples of such polynomials, arising from basic hitting-set designs, can be found in \cite{A11,KP11,K08,Sec.4}. 

Thm.2 is inspired from the ideas above; one crucial difference being that the annihilating polynomial \( g \), from the hitting-set, will now have nonconstant individual-degree. This makes the depth-reduction (to depth-4) \cite{AV08} prohibitively expensive. But, we can still invoke the other arguments, related to the Nisan-Wigderson design and Kaltofen’s factorization, to “stretch” the arity from \( \log c (sd) \) to \( n \), in \( c + 1 = O(1) \) many phases. This requires a new method to stretch in very tiny regimes, given “super”-hard polynomials (see Lem.\( [17] \)). A proof outline is in Fig.2. Such a result is possible in algebraic circuits, in contrast to boolean circuits, because we exploit the arity vs. degree tradeoff.

This proof method is robust enough to work, and gives most of the desired conclusions, even if the time-complexity of the hitting-set depends \( super \)- polynomially on \( sd \). This extension is discussed briefly in Sec.B (together with the obstacles in the analogous extension of Thm.1).

The proof of Thm.3 requires one to move in a different regime where the arity is \( n = a \log s = \omega(\log s) \), the semantic individual-degree is one and the circuit is depth-3 (Thm.16). We reach there in a sequence of steps, and then apply the Kronecker map \( (x_i \mapsto y^2) \) locally in blocks of size \( \log s \). This leads to an arity reduction \( a \log s \mapsto a \) in the Hitting-set problem. Refer Fig.1.

The proof of Thm.4 has two steps. In the first step, we show that with respect to any monomial ordering, the dimension \( k \) of the partial derivative space of a polynomial is lower bounded by the cone-size of its leading monomial. So, for every nonzero \( C \in C \) there is a monomial with nonzero coefficient and cone-size \( \leq k \). The second step is to check whether the coefficients of all the monomials in \( C \), with cone-size \( \leq k \), are zero. Lem.\( [20] \) describes the time required to check whether the coefficient of a monomial is zero. Lem.\( [21] \) gives us an optimal upper bound on the number of monomials with cone-size \( \leq k \).

Thm.5 unfolds an interesting combinatorial interaction between variable shift and cones of resulting monomials. Let a set of monomials \( B \) be the least basis, wrt to the basis isolating weight assignment, of \( f \). We consider the set of all submonomials of those in \( B \) and identify a subset \( A \) that is cone-closed. We define \( A \) in an algorithmic way, as described in Algo.\( [11] \). The fact that \( A \) is exactly a basis of the shifted \( f \) is proved in Lem.\( [29] \) by studying the action of the shift on the coefficient vectors. This has an immediate (nontrivial) consequence that any polynomial \( f \) over \( F_k \), when shifted by formal variables, becomes cone-closed.

2 Tiny diagonal depth-4 circuits—Proof of Theorems 1-3

In this section we will revisit the techniques that have appeared in some form in \cite{NW94,K08,A05,AV08,GKKS13} and strengthen each of them to derive our results. First, we show how to get a hard polynomial from a hitting-set in a general setting.

**Hitting-set generator.** Let \( C \) be a set of arity-\( n \) circuits. We call an \( n \)-tuple of univariates \( f(y) = (f_1(y), \ldots, f_n(y)) \) a \((t,d)\)-hsg (hitting-set generator) for \( C \) if: (1) for any nonzero \( C \in C \), \( C(f(y)) \neq 0 \), and (2) \( f \) has time-complexity \( t \) and the degree of each \( f_i \) is at most \( d \).

From this, we get a hard polynomial simply by looking at an annihilating polynomial of \( f(y) \).

**Lemma 6 (Hitting-set to hardness).** Let \( f(y) = (f_1(y), \ldots, f_n(y)) \) be a \((t,d)\)-hsg for \( C \). Then, there exists an arity-\( n \) polynomial \( g(x) \) that is not in \( C \), is computable in \( \text{poly}(tdn) \)-time, has individual degree less than \( \delta := [(d + 1)^{d/n}] \), and is homogeneous of degree \( (\delta - 1)n/2 \).
Proof. A natural candidate for \( g(x) \) is any annihilating polynomial of the \( n \) polynomials \( f(y) = (f_1(y), \ldots, f_n(y)) \), since for every nonzero \( h \in C \), \( h(f) \) is nonzero. Define \( \delta \geq 2 \) as the smallest integer such that \( \delta^{n/\delta} > d \). Consider \( g(x) \) as an arity-\( n \) polynomial with individual degree less than \( \delta \) and homogeneous of degree \((\delta - 1)n/2\). Then, \( g(x) \) can be written as:

\[
g(x) = \sum_{|c|=(\delta-1)n/2, \ 0 \leq c, c < \delta} c_x^e
\]

where, \( c_x^e \)'s are unknown to us. Note that the number of summands is at least \((\delta/2)^{n/2} \cdot \binom{n}{\delta/2} > \delta^{n/2} \) (for \( n \geq 4 \)). The former estimate can be obtained by picking a subset \( S \in \binom{[n]}{\delta/2} \) and considering all monomials in \( x_S \) of individual-degree < \( \delta/2 \). For every such monomial in \( x_S \) we can pick a (complementary) monomial \( x_{[n]\setminus S} \) with exponents from \([\delta/2, \ldots, \delta - 1]\) such that the product of these monomials has degree exactly \((\delta - 1)n/2\).

We can set all these \( c_x^e \)'s to zero except the ones corresponding to an index-set \( I \) of size \( \delta_0 := d n(\delta - 1)/2 + 2 < \delta^{n/3} n (\delta - 1)/2 + 2 \leq \delta^{n/2} \). This way we have exactly \( \delta_0 \) unknowns. To be an annihilating polynomial of \( f(y) \), we need \( g(f) = 0 \). By comparing the coefficients of the monomials in \( y \), both sides of Eqn[1] we get a linear system in the unknowns.

Suppose that \( \delta_1 \) is the degree of \( y \) in \( g(f) \). Then, \( g(f) \) can be written as \( g(f) = \sum_{i=0}^{\delta_1} p_i \cdot y_i \), where \( p_i \)'s are linear polynomials in \( c_x^e \)'s. The constraint \( g(f) = 0 \) gives us a system of linear equations with the number of unknowns \( \delta_0 \) and the number of equations \( \delta_1 + 1 \). The value of \( \delta_1 \) can be at most \( d \cdot n \cdot (\delta - 1)/2 \), which means that the number of unknowns \( \delta_0 \) is greater than the number of equations \( \delta_1 \). So, our system of homogeneous linear equations always has a nontrivial solution, which gives us a nonzero \( g \) as promised.

Computing \( f(y) \) takes \( t \) time and a solution of the linear equations can be computed in \( \text{poly}(t d n) \)-time. So, \( g(x) \) can be computed in \( \text{poly}(t d n) \)-time.

**Corollary 7** (E-computable). In the proof of Lem[6], if \( t d = 2^{O(n)} \) then the polynomial family \( g_n := g \), indexed by the arity, is E-computable (i.e. all the formal monomials and their coefficients can be produced in \( \text{poly}(2^n) \)-time given unary \( n \)).

**Proof.** The linear system that we got can be solved in \( \text{poly}(t d n) \)-time. As it is homogeneous we can even get an integral solution in the same time-complexity. Thus, assuming \( t d = 2^{O(n)} \), the time-complexity of computing a bit of coefficient \( g \) is \( \text{poly}(t d n) = \text{poly}(2^n) \), the coefficient bitsize is \( \text{poly}(2^n) \) and so is the number of monomials in \( g \) (\( \delta = [(d + 1)^{n/2}] = O(1) \)). In other words, if we consider the polynomials \( g_n := g \), indexed by the arity, then the family \( \{g_n\}_n \) is E-computable.

Towards a converse of the above lemma, a crucial ingredient is the Nisan-Wigderson design [NW94]. To describe it simply, the design stretches a seed from \( \ell \) to \( m \geq 2^\ell \) as follows,

**Definition 8.** Let \( \ell > n > d \). A family of subsets \( \mathcal{F} = \{I_1, \ldots, I_m\} \) on \( [\ell] \) is called an \((\ell, n, d)\)-design, if \( |I_i| = n \) and for all \( i \neq j \in [m] \), \( |I_i \cap I_j| \leq d \).

**Lemma 9** (Nisan-Wigderson design, Chap.16 [AB09]). There exists an algorithm which takes \((\ell, n, d)\) and a base set \( S \) of size \( \ell > \frac{10n^2}{d} \) as input, and outputs an \((\ell, n, d)\)-design \( \mathcal{F} \) having \( \geq 2^d \) subsets, in time \( 2^{O(\ell)} \).

Next, we use a hard polynomial \( q_m \) on a small design to get a poly-time computable arity-reducing polynomial-map for VP that preserves nonzeroness. Below we consider the individual-degree \( \delta \) to be a constant. A much more general version of this statement is postponed to Lem[17] where the notation would become more involved, the constant parameters would get slightly worse, and the Nisan-Wigderson stretch would be composed many times with itself.
Lemma 10 (Hardness to VP reduction). Let \( \{q_m\}_{m \geq 1} \) be a family of multi-(\( \delta - 1 \))-ic polynomials such that it can be computed in \( \delta^{O(m)} \) time, but has no \( \delta^m \)-size algebraic circuit. Then there is a \( \delta^{O(\log sd)} \)-time arity reduction, from \( n \) to \( O(\log sd) \), for VP circuits.

Proof. Note that there is a constant \( c_0 > 0 \) such that \( q_m \) requires \( \Omega(\delta^{\alpha m}) \)-size algebraic circuits. Otherwise \( \{q_m\}_{m \geq 1} \) will be in \( \cap_{c > 0} \text{Size}(\delta^{\alpha c}) \), and hence in \( \text{Size}(\delta^{\alpha m}) \).

Let \( C \) be a set of arity-\( n \) degree-\( d \) VP circuits with size \( \leq s \). Let \( n' := sd \geq n \). Let \( F = \{S_1, \ldots, S_{n'}\} \) be a \( (c_2 \log n', c_1 \log n', 10 \log n') \)-design on the variable set \( Z = \{z_1, \ldots, z_{c_2 \log n'}\} \). Constants \( c_2 > c_1 > 10 \) will be fixed later. Our hitting-set generator for \( C \) is defined as: for all \( i \in [n] \), \( x_i = q_{c_1 \log n'}(S_i) := p_i \). Then, we show that for any nonzero polynomial \( C(x) \in C \), \( C(p_1, \ldots, p_n) \) is also nonzero.

For the sake of contradiction, assume that \( C(p_1, \ldots, p_n) \) is zero. Since \( C(x) \) is nonzero, we can find the smallest \( j \in [n] \) such that \( C(p_1, \ldots, p_{j-1}, x_j, \ldots, x_n) =: C_1 \) is nonzero, but \( C_1(x_j = p_j) \) is zero. Thus, \( (x_j - p_j) \) divides \( C_1 \). Let \( a \) be an assignment on all the variables in \( C_1 \), except \( x_j \) and the variables \( S_j \) in \( p_j \), with the property: \( C_1 \) at \( a \) is nonzero. Since \( C_1 \) is nonzero, we can find such an assignment. Now our new polynomial \( C_2 \) on the variables \( S_j \) is of the form:

\[
C_2(S_j) = C(p'_1, \ldots, p'_{j-1}, x_j, a_{j+1}, \ldots, a_n)
\]

where, for each \( i \in [j-1] \), \( p'_i \) is the polynomial on the variables \( S_i \cap S_j \), and \( a_i \)'s are field constants decided by our assignment \( a \). By the design, for each \( i \in [j-1] \), \( |S_i \cap S_j| \leq 10 \log n' \). Since \( p'_i \) are polynomials on variables \( S_i \cap S_j \) of individual degree \( \leq \delta \), each \( p'_i \) has a circuit of size at most \( n_0^{10 \log n'} \leq \delta^{11 \log n'} \). Then we have a circuit for \( C_2 \) of size at most \( s_1 := s + n \cdot \delta^{11 \log n'} \), and degree at most \( d_1 := d \cdot \delta c_1 \log n' \). Since \( (x_j - p_j) \) divides \( C_2 \), we can invoke the VP factorization algorithm [Kal89] (see [Bir13] Thm.2.21) for the algebraic circuit complexity of factors) and get an algebraic circuit for \( p_j \) of size \( (s_1 d_1)^c_3 \), for some absolute constant \( c_3 \) (independent of \( c_1, c_2 \)).

Now we fix constants \( c_1, c_2 \). Pick \( c_1 \) such that \( \delta^{c_1 \log n'} \) is asymptotically larger than \( (2sn_0^{11 \log n'} \cdot d \delta c_1 \log n')^{c_3} > (s_1 d_1)^{c_3} \). Since \( sd = n' \) and \( \delta \geq 2 \), the absolute constant \( c_1 := 15c_3/c_0 \) (independent of \( c_2 \)) satisfies the above condition.

Pick \( c_2 \), following Lemmata, such that \( c_2 \log n' > 10 \cdot (c_1 \log n')^2/(10 \log n') \). So, \( c_2 := 1 + c_1^3 \) works. With these values of \( c_1, c_2 \), we have a design that ‘stretches’ \( c_2 \log n' \) variables to \( n \) subsets with the required ‘low’ intersection property. It is computable in \( \text{poly}(n') \)-time.

Moreover, if \( C(p_1, \ldots, p_n) \) is zero then, by the above discussion, \( p_j = q_{c_1 \log n'}(S_j) \) has a circuit of size \( (s_1 d_1)^{c_3} = o(\delta^{c_1 \log n'}) \). This violates the lower bound hypothesis. Thus, \( C(p_1, \ldots, p_n) \) is nonzero.

The time for computing \( (p_1, \ldots, p_n) \) depends on: (1) computing the design (i.e. \( \text{poly}(n') \)-time), and (2) computing \( q_{c_1 \log n'} \) (i.e. \( \delta^{O(\log n')} \)-time). Thus, the arity reduction map for VP is computable in \( \delta^{O(\log n')} \) time. \( \square \)

Once we have a polynomial that is hard for a tiny model, to apply the above lemma, we need to show that it is also hard for VP. This is done by depth-reduction results. First, we need a lemma that converts a monomial into a sum of powers. This was used in [GKKS13]. (It requires char \( F = 0 \) or large.)

Lemma 11 (Fischer’s Trick [Fis94]). Over a field \( F \) of char(\( F \)) = 0 or \( > r \), any expression of the form \( g = \sum_{i \in [k]} \prod_{j \in [l]} g_{ij} \) with deg(\( g_{ij} \)) \( \leq d \), can be rewritten as \( g = \sum_{i \in [k']} c_i g_{i}^{k'} \) where \( k' := k2^r \), deg(\( g_i \)) \( \leq d \) and \( c_i \in F \). In fact, each \( g_i \) is a linear combination of \( \{g_{i,j} \} \) for some \( i' \).

Motivated by this transformation (when \( 2^r \) is ‘small’), we define a tiny subclass of VP.

Definition 12. The diagonal depth-4 circuits compute polynomials of the form \( \sum_{i \in [k]} c_i f_i^{k_i} \) where \( f_i \)'s are sparse polynomials in \( \mathbb{F}[x_1, \ldots, x_n] \) of degree \( \leq b \), \( a_i \leq a \) and \( c_i \)'s in \( \mathbb{F} \). A standard
notation to denote this class is $\Sigma \land^a \Sigma \Pi^b(n)$. This is a special case of the depth-$4$ $\Sigma \Pi^a \Sigma \Pi^b(n)$ model that computes polynomials of the form $\sum_{i \in [k]} \prod_{j \in [a]} f_{i,j}$ where $f_{i,j}$’s are sparse polynomials in $\mathbb{F}[x_1, \ldots, x_n]$ of degree $\leq b$.

The superscripts $a, b$ on the product gates denote an upper bound on the fanin. We denote $\Sigma \Pi \Sigma \Pi^1$ circuits by $\Sigma \Pi \Sigma$ and call them depth-3.

Given a constant $c > 1$ and functions $\mu(a), \mu'(a) \geq \Omega(a)$, we define the class $\mathcal{T}_{\mu, \mu', c}$, called tiny diagonal depth-$4$, containing $\Sigma \land^a \Sigma \Pi^b(n)$ circuits of size $\leq s$ that compute homogeneous polynomials of semantic individual-degree $\leq a'$ and $2^a + 2^b + \mu(a) + \mu'(a') < s^c$. (Our proofs can handle an additional constraint on the bottom-fanin: $b \leq a'n/a^{\log_2(b)}$.)

Analogously, we define the class $\mathcal{T}_{\mu', c}$, called tiny depth-3, containing $\Sigma \Pi \Sigma(n)$ circuits of size $\leq s$ that compute homogeneous polynomials of semantic individual-degree $\leq a'$, and $2^n + \mu'(a') < s^c$.

**Remark.** Note that $n, b = O(\log s)$ and by picking the function $\mu(\cdot)$ (resp. $\mu'$) arbitrarily large we can make $a = a(s) = \omega(1)$ (resp. $a'$) an arbitrarily small computable function. Also, in this regime the number of monomials in the bottom $\Sigma \Pi^b$ layer is $(n^b + b)^c < 2^{n+b} < s^{2c}$, so the $f_j$’s can be thought of as given in the dense representation. Analogously, in tiny depth-3, $n = O(\log s)$ and the semantic individual-degree bound $a' = \omega(1)$ can be picked arbitrarily small.

An alternative interpretation of the tiny models can be given using the parameterized complexity $[\text{DF13}]$ of PIT. Essentially, we are interested in hitting-sets for the diagonal depth-$4$ model that are fixed-parameter tractable (FPT) wrt $n, b, a, b'$ (input size is $s$). Analogously, we are interested in hitting-sets for the depth-$3$ model that are fixed-parameter tractable wrt $n$ and $a'$, where $a'$ is the semantic individual-degree bound (input size is $s$). Our tiny models are essentially the smallest input instances in the kernel of the corresponding FPT algorithms.

Now we invoke VP depth-reduction to get to the tiny model, and finish our proof.

**Proof of Thm.**[7] The proof is along the lines of [AV08 Thm.3.2]. Using Lem.[6] from poly-time hitting-set generator for tiny diagonal depth-$4$, we get a hard polynomial for this model. Then we show that it is also hard for VP and invoke Lem.[10] to get the VP arity reduction.

Now we provide the details. Let constant $c > 1$, functions $\mu(a) = \Omega(a)$ and $\mu'$ be given in the hypothesis. Let $\mathcal{C} \subset \mathcal{T}_{\mu, \mu', c}$ be the set of tiny diagonal depth-$4$ circuits of size $\leq s$ and arity $m := \log s$. Assume that $\mathcal{C}$ has a $(s^c, s^c)$-hsig $f(y)$ for some constant $c > 1$. Then using Lem.[6] we have an $m$-variate homogeneous polynomial $q_m$ with individual degree less than some constant $\delta$, such that $\delta^{m/3} = \delta(\log n)^{1/3} > s^c$, and $q_m$ is computable in $s^{O(1)} = \delta^{O(m)}$ time. It has degree $= (\delta - 1)m/2$. Importantly, $q_m \notin \mathcal{T}_{\mu, \mu', c}$, thus, no tiny diagonal depth-$4$ circuit of size $\leq s = \delta^{\Theta(m)}$ can compute it (otherwise, $q_m(f(y)) \neq 0$ which contradicts its definition as an annihilator). Next we show that it is also not computable by any $\delta^{o(m)}$-size algebraic circuit.

For the sake of contradiction, assume that $q_m$ has a $\delta^{o(m)}$-size circuit. From depth-reduction results [Sap16 Chap.5] we get a homogeneous circuit $C$, of $\Theta(\log \delta m)$-depth and $s_m = \delta^{o(m)}$ size, with the additional properties:

1. alternative layers of addition/multiplication gates with the top-gate (root) being addition.
2. below each multiplication layer the polynomial degree at least halves.
3. fan-in of each multiplication gate is at most 5.

Now we cut the circuit $C$ at the $t$-th layer of multiplication gates from the top, where $t = t(s_m)$ will be fixed later, to get the following two parts:

**Top part:** the top part computes a polynomial of degree at most $5t$ and the number of variables is at most $s_m$. So it can be reduced to a $\Sigma \Pi$ circuit of size $(s_m + 5^t) = s_m^{O(5)}$ (Stirling’s approximation, see [Sap16 Prop.4.4]).
**Bottom part:** In the bottom part, we can have at most \( s_m \) many top-multiplication gates that feed into the top part as input. Each multiplication gate computes a homogeneous polynomial of degree at most \((\delta - 1)m/2 \cdot 2^{-t}\) and the number of variables is at most \( m \). So each multiplication gate can be reduced to a \( \Sigma \Pi \) circuit of size \( (\delta - 1)m/2^{t+1} = 2^{O(\delta m t/2^{t})} \).

From the above discussion, we have a homogeneous \( \Sigma \Pi^d \Sigma \Pi^{(\delta - 1)m/2^{t+1}} \) circuit \( C' \), computing \( q_m \), that has size \( s_m^{O(\delta^2)} + s_m \cdot 2^{O(\delta m t/2^t)} \).

The second summand becomes \( 2^m \) if we pick \( t = \omega(1) \) (recall that \( s_m = \delta o(m) \) and \( \delta = O(1) \)). To get a similar upper bound on the first summand we need to pick \( 5^t \log s_m = o(m \log \delta) \). Finally, we also want \( \mu(5^t) = o(m) \). A function \( t = t(s_m) = t(s) \), satisfying the three conditions, exists as \( \log s_m = o(m \log \delta) \) and \( \mu(\cdot) \) is an increasing function. Let us fix such a function \( t \). (As \( C \) has super-constant depth, we can also assume that the cut at depth \( t \) will be possible.) Thus the circuit \( C' \), computing \( q_m \), has size \( s'_m = \delta o(m) \).

Let \( a := 5^t \) resp. \( b := (\delta - 1)m/2^{t+1} \) be the bounds for the top resp. bottom fanins of the product gates. Note that \( b \leq \delta m/a \log_2 2 \). Consider the measure \( E := 2^m + 2^b + \mu(a) + \mu'(\delta) \). We have the estimate \( E \leq s + 2^{\delta m(a) + o(m)} + O(1) = s + O(s) = o(s^5) \). So now we have a shallow circuit for \( q_m \) of the form homogeneous \( \Sigma \Pi^5 \Sigma \Pi^b \). Applying Lem.11, we get a tiny diagonal depth-4 circuit, in \( T_{\mu, \mu'} \), computing \( q_m \) of the form \( \Sigma \land^a \Sigma \Pi^b \) and size \( s'_m \cdot 2^{t} = \delta o(m) \cdot 2^{O(\mu(a))} = \delta o(m) \) which is \( < s \). This contradicts the hardness of \( q_m \). Thus, there is no algebraic circuit for \( q_m \) of size \( \delta o(m) \).

Now invoking Lem.10 (resp. Lem.13) on the hard family \( \{q_m\}_{m \geq 1} \), we get our PIT (resp. lower bound) claim. \( \square \)

The existence of such a family \( \{q_m\}_m \) has interesting complexity consequences.

**Lemma 13** (Class separation). If we have an E-computable polynomial family \( \{f_n\}_n \) with individual-degree \( O(1) \) and algebraic circuit complexity \( 2^{\Omega(n)} \), then either \( E \subseteq \#P/poly \) or \( VNP \) has polynomials of algebraic circuit complexity \( 2^{\Omega(n)} \).

**Proof.** Say, for a constant \( \delta \geq 1 \), we have an E-computable multi-\( \delta \)-ic polynomial family \( \{f_n\}_n \) with algebraic circuit complexity \( 2^{\Omega(n)} \). Clearly, the coefficients in \( f_n \) have bitsize \( 2^{O(n)} \). By using a simple transformation, given in [KP09, Lem.3.9], we get a multi-\( \delta \)-ic polynomial family \( \{h_n\}_n \), that is E-computable and has algebraic complexity \( 2^{\Omega(n)} \), such that its coefficients are \( \{0, \pm 1\} \).

Assume \( E \subseteq \#P/poly \). Since each coefficient of \( h_n \) is a signed-bit that is computable in \( E \), we deduce that the coefficient-function of \( h_n \) is in \( \#P/poly \). Thus, by [Bir13, Prop.2.20], \( \{h_n\}_n \) is in \( VNP \) and has algebraic circuit complexity \( 2^{\Omega(n)} \). \( \square \)

Our techniques could handle many other ‘tiny’ models. The proofs are given in Sec.A

**Theorem 14** (Tiny depth-3). If we have poly-time hitting-set for a tiny depth-3 model, then for VP circuits we have a poly(sd)-time arity reduction \( (n \mapsto O(\log sd)) \) that preserves nonzeroness (and proves an exponential lower bound).

**Theorem 15** (Width-2 ABP). If we have poly(s2^n)-time hitting-set for size-s width-2 upper-triangular ABP, then for VP circuits we have a poly(sd)-time arity reduction \( (n \mapsto O(\log sd)) \) that preserves nonzeroness (and proves an exponential lower bound).

Our method could also handle individual-degree \( a' = O(1) \) (eg. multilinear polynomials), but then we have to allow arity \( \omega(\log s) \) (clearly, arity \( O(\log s) \) trivializes the model [BOT88]). We state our result below in parameterized complexity terms. (Proof in Sec.A)
Multilinear tiny depth-3. Given a constant $c > 1$ and an arbitrary function $\mu'(a') = \Omega(a')$, we define the class $\mathcal{M}_{\mu', c}$, called multilinear tiny depth-3, containing $\Sigma_2 \Pi_2 \Sigma(n)$ circuits of size $\leq s$ that compute multilinear homogeneous polynomials, and $\mu'(n/\log s) < s^c$.

**Theorem 16** (Multilinear tiny depth-3). If we have poly-time hitting-set for a multilinear tiny depth-3 model, then for VP circuits we have a poly$(sd)$-time arity reduction ($n \mapsto O(\log sd)$) that preserves nonzeroness (and proves an exponential lower bound).

### 2.1 Many-fold composition of Nisan-Wigderson design—Proof of Thm. 2

In the following theorem statement one should think of $\varepsilon(\cdot)$ to be a “very” slowly growing function, eg. $\log^\omega(s)$ for $c \geq 1$. This would determine the arity of the hard polynomial and $\delta(\cdot)$ would determine its individual-degree. Also, the proof requires two constants ($c_0, c_1$) that we have fixed a priori in the statement without providing a motivation! They are there to play opposite roles: $c_0$ determines the ‘hardness’ parameter of $q_m'$, and $c_1$ determines the “easiness” of $q_m'$ when a lot of its variables are fixed non-uniformly. There is a second subscript $s$ in $q_m$ that we will suppress when it is clear from the context.

**Lemma 17** (Hardness to tinier arity). Let $c_3 \geq 1$ be the exponent in Kaltofen’s factoring algorithm [Bir13 Thm. 2.21]. Let $c \geq 1$ be a constant, $c_0 := [9\sqrt{e} + 3]c_3$, $c_1 := [30e + 10\sqrt{e} + 1]c_3$ and $c_2 := 1 + c_0^2$. Let $\varepsilon : N \to N_{\geq 0}$ be an unbounded function such that $\varepsilon(s) \leq O(\log s)$.

Suppose we have a family $\{q_{m,s} \mid m \in \mathbb{N}, s \in \varepsilon^{-1}(\{m/c_1\})\}$ of multi-$(\delta(m) - 1)$-ic arity-$m$ polynomials that can be computed in $s^{O(1)}$ time, but has no size-$s$ algebraic circuit, where $\delta(m) := s^{3\varepsilon/m} \leq s^{3c_1/(c_0s)}$. Then, there is a poly$(sd)$-time arity reduction map, reducing $n \mapsto \min\{2^{\varepsilon((sd)^{c_0})}, s\}$ to $c_2\varepsilon((sd)^{c_0})$ and preserving nonzeroness, for size-$s$ degree-$d$ arity-$n$ circuits.

**Proof.** Let $s' := sd$. Let $\mathcal{C}$ be a set of arity $n \leq s'$ degree-$d$ algebraic circuits with size $\leq s$. We intend to stretch $c_2\varepsilon((s)^{c_0})$ arity to $n$. Define $m' := c_1\varepsilon((sd)^{c_0})$. Note that $q_{m'} := q_{m',c_0}$ has no algebraic circuit of size $s^{c_0}$. Its individual-degree is $\leq \delta(m') = s^{3\varepsilon/m'} = s^{c_0(1)}$.

Let $\mathcal{F} = \{S_1, \ldots, S_n\}$ be a $(c_2\varepsilon((s)^{c_0}), m', 10\varepsilon((s)^{c_0}))$-design on the variable set $Z = \{z_1, \ldots, z_S\}$. Constants $c_2 > c_1 > 10$ will ensure the existence of the design. Our hitting-set generator for $\mathcal{C}$ is defined as: for all $i \in [n], x_i \mapsto q_{m'}(S_i) := p_i$. Then, we show that for any nonzero polynomial $C(x) \in \mathcal{C}, C(p_1, \ldots, p_n)$ is also nonzero.

For the sake of contradiction, assume that $C(p_1, \ldots, p_n)$ is zero. Since $C(x)$ is nonzero, we can find the smallest $j \in [n]$ such that $C(p_1, \ldots, p_{j-1}, x_j, \ldots, x_n) =: C_1$ is nonzero, but $C_1(x_j = p_j)$ is zero. Thus, $(x_j - p_j)$ divides $C_1$. Let $a$ be an assignment on all the variables in $C_1$, except $x_j$ and the variables $S_j$ in $p_j$, with the property: $C_1$ at $a$ is nonzero. Since $C_1$ is nonzero, we can find such an assignment. Now our new polynomial $C_2$ on the variables $S_j$ is of the form:

$$C_2(S_j) = C(p'_1, \ldots, p'_{j-1}, x_j, a_{j+1}, \ldots, a_n)$$

where, for each $i \in [j - 1], p'_i$ is the polynomial on the variables $S_i \cap S_j$, and $a_i$’s are field constants decided by our assignment $a$. By the design, for each $i \in [j - 1], |S_i \cap S_j| \leq 10\varepsilon((s)^{c_0})$. Since $p'_i$ are polynomials on variables $S_i \cap S_j$ of individual degree $< \delta(m')$, each $p'_i$ has a circuit of size at most $m'\delta(m') \cdot \delta(m')^{10\varepsilon((s)^{c_0})} = m'\delta(m') \cdot \delta(m')^{10m'c_1}$. Then we have a circuit for $C_2$ of size at most $s_1 := s + nm'\delta(m') \cdot \delta(m')^{10m'c_1}$, and degree at most $d_1 := dm'\delta(m')$. Since $(x_j - p_j)$ divides $C_2$, we can invoke the VP factorization algorithm [Kal89] (see [Bir13 Thm. 2.21]) for the algebraic circuit complexity of factors) and get an algebraic circuit for $p_j$ of size $(s_1d_1)^{c_3}$

$$\leq (snm'\delta(m') \cdot \delta(m')^{10m'c_1} \cdot dm'\delta(m'))^{c_3} = \left(s' \cdot n^2 \cdot \delta(m')^{2m + \frac{10m'c_1}{c_1}}\right)^{c_3}$$

$$< (s^2 \cdot s' \cdot \delta(m')^{10m'c_1})^{c_3}$$
\[
q_j = q_{m'}(S_j) \text{ has size smaller than } s^{\epsilon c_0}, \text{ which contradicts the hardness of } q_{m'} \text{. Thus, } C(p_1, \ldots, p_n) \text{ is nonzero.}
\]

The time for computing \((p_1, \ldots, p_n)\) depends on: (1) computing the design (i.e. \(\text{poly}(2^m)\))-

time, and (2) computing \(q_{m'}\) (i.e. \(\text{poly}(sd)\))-time. Thus, the arity reduction map for VP is

computable in \(\delta(m')^{O(m')} = \text{poly}(sd)\)-time.

Now we will use the above lemma iteratively, \(\leq c + 1\) times, to get Thm\(^2\) See a proof

outline in Fig\(^2\).

\[\text{Proof of Thm}\(^2\)\] Note that, because of general depth-reduction results \([\text{Sap16, Thm.5.15}]\), we can

assume wlg that the syntactic and semantic degrees of an algebraic circuit are equal. Suppose, for a constant \(c \in \mathbb{N}_{\geq 1}\), we have a \(\text{poly}(s, \exp^c(n))\)-time hitting-set \(\mathcal{H}_{c,s,n}\) for size-\(s\) (semantic)

degree-\(s\) \(c\)-parity-\(n\) circuit family \(c_{s,s,m}\). Let the time-complexity of the hitting-set

be less than \((s \exp^c(n))^{c/2}\), for a constant \(c\), and define constants \(c_1, c_0\) as in Lem\(^17\) Note

that this time-complexity is \(< s^c\) for arity \([\log^{c+1}(s)]\). (Note: We will assume the boundary

condition \(s > \exp^c(c+1)(2)\).) We will now iteratively find hitting-sets for larger arities in \(c + 1\) phases

(Caution: The constants \(c_0, c_1, c_2\) will change in every phase, as they depend on the

time-complexity of the hitting-set obtained from the previous phase.)

\[\text{Phase } 1- \text{ Solving for arity } \omega(\log^c(sd)): \text{ Consider the functions } \epsilon_{c+1} : s \mapsto 2[\log^o(s)] \text{ ,}

\]

and \(m := c_1 \epsilon_{c+1}(s) = o(\log^c s)\). In particular, \(\exp^o(m) < s\). Thus, the hitting-set \(\mathcal{H}_{c,s,m}\),

for the tiny models \(c_{s,s,m}\), has time-complexity less than \(s^c\). By Lem\(^6\) we get an \(m\)-variate

\(q_m := q_{m,s} \notin c_{s,s,m}\) such that: (1) it is computable in \(\text{poly}(s)\)-time, (2) its individual-degree is

at most \(\delta(m) = s^3e/m\), and (3) total semantic degree is \(\leq s^{3e/m}m = o(s)\). Altogether, we can

deduce that \(q_m\) has no algebraic circuit of size \(< s\).

Thus, by Lem\(^17\) we have a \(\text{poly}(sd)\)-time arity reduction, reducing \(n := 2^\epsilon_{c+1}(sd)^o\) to

c\(2^\epsilon_{c+1}(sd)^o\), for size-\(s\) degree-\(d\) \(c\)-arity-\(n\) circuits. In other words, this gives a \(\text{poly}(sd)\)-time

hitting-set \(\mathcal{H}'_{c,s,d,n}\) for arity \(n = 2^\epsilon_{c+1}(sd)^o\geq (\log^c(sd)^o)^2 \geq \omega(\epsilon_{c}(sd))\) circuits. Because

we first reduce the arity from \(n\) to \(c_2^\epsilon_{c+1}(sd)^o = o(\log^c(sd)^o)\), preserving nonzeroness, and then

use the hitting-set \(\mathcal{H}_{c,s,d}^{O(1)}\) of \(c_2^\epsilon_{c+1}(sd)^o\). This hitting-set has time-complexity \((sd)^{O(1)}\), and

the former arity reduction using \(q_m\) \((m' := c_1 \epsilon_{c+1}(sd)^o)\) is computable in \(\text{poly}(sd)\)-time.

\[\text{Phase } 2- \text{ Solving for arity } \omega(\log^{c-1}(sd)): \text{ In this phase we have a } \text{poly}(s)\text{-time hitting-set}

\mathcal{H}'_{c,s,m,n}\text{, where } m = c_1 \epsilon_{c}(s) := 2c_1[\log^c s] = o((\log^c s)^2), \text{ for size-} s\text{-degree-} s\text{-arity-} n\text{ circuit.}

As in the previous phase, we use Lem\(^6\) on \(\mathcal{H}'_{c,s,m}\) to get a hard polynomial. Then, using

Lem\(^17\) we get a \(\text{poly}(sd)\)-time hitting-set \(\mathcal{H}'_{c,1,s,d,n}\) for arity \(n = 2^\epsilon_{c-1}(sd)^o\geq (\log^{c-1}(sd)^o)^2 \geq \omega(\epsilon_{c-1}(sd))\) circuits, by first reducing the arity to \(c_2^\epsilon_{c-1}(sd)^o = o((\log^c(sd)^o)^2)\) and then

using \(\mathcal{H}_{c,s,d}^{O(1)}\) of \(c_2^\epsilon_{c-1}(sd)^o\).

\[\text{Phase } c + 2- \text{ Solving for general arity } n: \text{ After } \leq c + 1 \text{ such iterations we would have a}

\text{poly}(sd)\text{-time hitting-set } \mathcal{H}_{c,s,d,n}\text{, for size-} s\text{-degree-} d\text{-arity-} n\text{ general circuits. Note that each}

arity-growth phase gives a polynomially-bigger hitting-set. Thus, we need } c\text{ to be a constant so}

that the final hitting-set has time-complexity \(\text{poly}(sd)\).

\footnote{The first phase seems extraneous, but we need it because Nisan-Wigderson design introduces constants
\((c_2, c_1, 10)\). The latter force us to start with a hitting-set for arity slightly more than \([\log^c(sd)]\), i.e. \(\omega(\log^c(sd))\).}
2.2 Arbitrarily small arity suffices– Proof of Thm.3

Using the multilinear tiny depth-3 model (Thm.16) we will now reduce the arity, for blackbox PIT purposes, arbitrarily.

Proof of Thm.3 Suppose we have a poly(s,μ(n))-time hitting-set Hs,n for size-s arity-n ΣΠΣ∧ circuits. Wlog we can assume that μ(n) = Ω(n). Let a = a(s) be a function satisfying μ(a) ≤ s, and define n = n(s) := a log s. Consider a size-s ΣΠΣ(n) circuit C ≠ 0 computing a multilinear polynomial. We intend to design a hitting-set for C.

Partition the variable set \{x_1, \ldots, x_n\} into a blocks B_j, j \in [a], each of size log s. Let B_j = \{x_{u(j)+1}, x_{u(j)+2}, \ldots, x_{u(j)+\log s}\}, for all j \in [a] (pick u to be the appropriate function). Consider the arity-reducing “local Kronecker” map \(\varphi: x_{u(j)+i} \mapsto y_{j}^{2^i}\). Note that \(\varphi(C) \in \mathbb{F}[y_1, \ldots, y_a]\), and its semantic individual-degree is at most 2s.

It is easy to see that \(\varphi(C) \neq 0\) (basically, use the fact that C computes a nonzero multilinear polynomial and \(\varphi\) keeps the multilinear monomials distinct). Finally, \(\varphi(C)\) becomes an arity-a ΣΠΣ∧ circuit of size at most \(s + s \cdot 2^{\log s} = O(s^2)\). Thus, using \(H_{O(s^2),a}\) we get a hitting-set for \(\varphi(C)\) of time-complexity \(poly(s^2,\mu(a)) = poly(s)\). In turn, we get a poly-time hitting-set for multilinear tiny depth-3 model \(\mathcal{M}_{\mu,2}\). By invoking Thm.16 we finish the argument. □

We can also work with a version of diagonal depth-4 with arbitrarily small n & a.

Theorem 18 (Tinier n, a). If, for some \(\mu, \mu'\), we have \(poly(s,\mu(a),\mu'(n))\)-time hitting-set for size-s Σ∧a ΣΠΣ(n) circuits, then for VP circuits we have a \(poly(sd)\)-time arity reduction (n \(\mapsto O(\log sd)\)) that preserves nonzeroness (and proves an exponential lower bound).

(Proved in Sec.A)

3 Low-cone concentration and hitting-sets– Proof of Thm.4

In this section we initiate a study of properties that are relevant for tiny circuits (or the log-arity regime).

Definition 19 (Cone of a monomial). A monomial \(x^e\) is called a submonomial of \(x^f\), if \(e \leq f\) (i.e. coordinate-wise). We say that \(x^e\) is a proper submonomial of \(x^f\), if \(e \leq f\) and \(e \neq f\).

For a monomial \(x^e\), the cone of \(x^e\) is the set of all submonomials of \(x^e\). The cardinality of this set is called cone-size of \(x^e\). It equals \([\{e+1\}] = \prod_{i \in [n]} (e_i + 1)\), where \(e = (e_1, \ldots, e_n)\).

A set \(S\) of monomials is called cone-closed if for every monomial in \(S\) all its submonomials are also in \(S\).

Lemma 20 (Coeff. extraction). Let \(C\) be a circuit which computes an arity-\(n\) degree-\(d\) polynomial. Then for any monomial \(m = \prod_{i \in [n]} x_i^{e_i}\), we have blackbox access to a poly((C|d,cs(m)))-size circuit computing the coefficient of \(m\) in \(C\), where cs(m) denotes the cone-size of \(m\).

Proof. Our proof is in two steps. First, we inductively build a circuit computing a polynomial which has two parts; one is coef\(_m\)(C) \cdot m and the other one is a “junk” polynomial where every monomial is a proper super-monomial of \(m\). Second, we construct a circuit which extracts the coefficient of \(m\). In both these steps the key is a classic interpolation trick.

We induct on the variables. For each \(i \in [n]\), let \(m_{[i]}\) denote \(\prod_{j \in [i]} x_j^{e_j}\). We will construct a circuit \(C^{(i)}\) which computes a polynomial of the form,

\[
C^{(i)}(x) = \text{coef}_{m_{[i]}}(C) \cdot m_{[i]} + C^{(i)}_{\text{junk}}
\]
where, for every monomial $m'$ in the support of $C_{junk}^{(i)}$, $m_{[i]}$ is a proper submonomial of $m'_{[i]}$.

**Base case:** Since $C := C^{(0)}$ computes an arity-$n$ degree-$d$ polynomial, $C(x)$ can be written as $C(x) = \sum_{j=0}^{d} c_j x_1^j$ where, $c_j \in \mathbb{F}[x_2, \ldots, x_n]$. Let $\alpha_0, \ldots, \alpha_e$ be some $e_i + 1$ distinct elements in $\mathbb{F}$. For every $\alpha_j$, let $C_{\alpha_j x_1}$ denote the circuit $C(\alpha_j x_1, x_2, \ldots, x_n)$ which computes $c_0 + c_1 \alpha_j x_1 + \ldots + c_e \alpha_j^e x_1^e + \cdots + c_d \alpha_j^d x_1^d$. Since

$$M = \begin{bmatrix} 1 & \alpha_0 & \ldots & \alpha_0^e \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha_e & \ldots & \alpha_e^e \end{bmatrix}$$

is an invertible Vandermonde matrix, one can find an $a = [a_0, \ldots, a_e] \in \mathbb{F}^{e+1}$, $a \cdot M = [0, 0, \ldots, 1]$. Using this $a$, we get the circuit $C^{(1)} := \sum_{j=0}^{e} a_j C^{(0)}_{\alpha_j x_1}$. Its least monomial wrt $x_1$ has degree $\deg_{x_1} \geq e_1$, which is the property that we wanted.

**Induction step** ($i \rightarrow i + 1$): From induction hypothesis, we have the circuit $C^{(i)}$ with the properties mentioned in Eqn. $[\mathbb{F}]$. The polynomial can also be written as $b_0 + b_1 x_{i+1} + \ldots + b_{e_i+1} x_{i+1}^{e_i+1} + \cdots + b_d x_{i+1}^d$, where every $b_j$ is in $\mathbb{F}[x_1, \ldots, x_i, x_{i+2}, \ldots, x_n]$. Like the proof of the base case, for $e_i+1 + 1$ distinct elements $\alpha_0, \ldots, \alpha_{e_i+1} \in \mathbb{F}$, we get $C^{(i+1)} = \sum_{j=0}^{e_i+1} a_j C^{(i)}_{\alpha_j x_{i+1}}$, for some $a = [a_0, \ldots, a_{e_i+1}] \in \mathbb{F}^{e_i+1}$ and the structural constraint of $C^{(i+1)}$ is easy to verify, completing the induction.

Now we describe the second step of the proof. After first step, we get

$$C^{(n)}(x) = \text{coef}_m(C) \cdot m + C_{junk}^{(n)}$$

where for every monomial $m'$ in the support of $C_{junk}^{(n)}$, $m$ is a proper submonomial of $m'$. Consider the polynomial $C^{(n)}(x_1 t, \ldots, x_n t)$ for a fresh variable $t$. Then, using interpolation wrt $t$ we can construct a $O(|C^{(n)}| \cdot d)$-size circuit for $\text{coef}_m(C) \cdot m$, by extracting the coefficient of $t^{\deg(m)}$, since the degree of every monomial appearing in $C_{junk}^{(n)}$ is $\deg(m)$. Now evaluating at $1$, we get $\text{coef}_m(C)$. The size, or time, constraint of the final circuit clearly depends polynomially on $|C|$, $d$ and $\text{cs}(m)$.

But, how many low-cone monomials can there be? Fortunately, in the log-arity regime they are not too many [Sap13]. Though, in general, they are quasipolynomially many.

**Lemma 21** (Counting low-cones). The number of arity-$n$ monomials with cone-size at most $k$ is $O(nk^2)$, where $s := (3n/\log k)^{\log k}$.

**Proof.** First, we prove that for any fixed support set, the number of cone-size $\leq k$ monomials is less than $k^2$. Next, we multiply by the number of possible support sets to get the estimate.

Let $T(k, \ell)$ denote the number of cone-size $\leq k$ monomials $m$ with support set, say, exactly $\{x_1, \ldots, x_\ell\}$. Since the exponent of $x_\ell$ in such an $m$ is at least 1 and at most $k - 1$, we have the following by the disjoint-sum rule: $T(k, \ell) \leq \sum_{i=2}^{k} T(k/i, \ell - 1)$. This recurrence affords an easy inductive proof as, $T(k, \ell) \leq \sum_{i=2}^{k} (k/i)^2 < k^2 \cdot \sum_{i=2}^{k} \left(\frac{1}{i-1} - \frac{1}{i}\right) < k^2$.

From the definition of cone, a cone-size $\leq k$ monomial can have support size at most $\ell := \lfloor \log k \rfloor$. The number of possible support sets, thus, is $\sum_{i=0}^{\ell} \binom{n}{i}$. Using the binomial estimates [Juk10, Chap.1], we get $\sum_{i=0}^{\ell} \binom{n}{i} \leq (3n/\ell)^{\ell}$.

The partial derivative space of algebraic circuits has been defined, and mined, in various works [CKW11]. Even when this space is small we do not have efficient hitting-sets known (though [FSS14] gave an $O(\log \log s \cdot \text{time hitting-set})$. Below we give a poly-time solution in the log-arity regime. (It requires that $\mathbb{F} = 0$ or large.)
**Proof of Thm[4]** The proof has two steps. First, we show that with respect to any monomial ordering $\prec$, for all nonzero $C \in C$, the dimension of the partial derivative space of $C$ is lower bounded by the cone-size of the leading monomial (that nontrivially occurs) in $C$. Using this, we can get a blackbox PIT algorithm for $C$ by testing the coefficients of all the monomials of $C$ of cone-size $\leq k$ for zeroness. Next, we estimate the time complexity to do this.

The first part is the same as the proof of [For14 Cor.8.4.14] (with origins in [FSS13]). Here, we give a brief outline. Let $LM(\cdot)$ be the leading monomial operator wrt the monomial ordering $\prec$. It can be shown that for any polynomial $f(x)$, the dimension of its partial derivative space $\partial_{x^{\prec}}(f)$ is the same as $D := \# \{LM(g) \mid g \in \partial_{x^{\prec}}(f)\}$ (see [For14 Lem.8.4.12]). This means that $\dim\partial_{x^{\prec}}(f)$ is lower-bounded by the cone-size of $LM(f)$ [For14 Cor.8.4.13], which completes the proof of our first part.

Next, we apply Lemma 20 on the circuit $C$ and a monomial $m$ of cone-size $\leq k$, to get the coefficient of $m$ in $C$ in poly$(sdk)$-time. Finally, Lemma 21 tells that we have to access at most $k^2 \cdot (3n/\log k)^{\log k}$ many monomials $m$. Multiplying these two expressions gives us the time bound.

This gives us immediately,

**Corollary 22.** Let $C$ be a set of arity-$n$ degree-$d$ size-$s$ circuits with $n = O(\log sd)$. Suppose that, for all $C \in C$, the dimension of the partial derivative space of $C$ is poly$(sd)$. Then, the blackbox PIT for $C$ can be solved in poly$(sd)$-time.

A depth-3 diagonal circuit [Sax08] is of the form $C(x) = \sum_{i \in [k]} c_i \ell_i^d$, where $\ell_i$’s are linear polynomials over $F$ and $c_i$’s in $F$. We use $rk(C)$ to denote the linear rank of the polynomials $\{\ell_i\}_i$.

**Theorem 23.** Let $C$ be the set of all arity-$n$ degree-$d$ size-$s$ depth-3 diagonal circuits. Suppose that, for all $C \in C$, $rk(C) = O(\log sd)$. Then, the blackbox PIT for $C$ can be solved in poly$(sd)$-time.

(Proved in Sec[C])

4 Cone-closed basis after shifting— Proof of Thm[5]

In this section we will consider polynomials over a vector space, say $F^k$. This viewpoint has been useful in studying algebraic branching programs (ABP), eg. [ASS13 FSS14 AGKS15 GKST16]. Let $D \in F^k[x]$ and let $sp(D)$ be the span of its coefficients. We say that $D$ has a cone-closed basis if there is a cone-closed set of monomials $B$ whose coefficients in $D$ form a basis of $sp(D)$.

This definition is motivated by the fact that there are some models which have this property naturally, for eg. see Lemma 33. In general, this concept subsumes some of the well-known notions of rank concentration [ASS13 FSS14 For15 For14], i.e. ensuring a basis of $sp(D)$ in a set of monomials that have a small measure in some sense (eg. cone-size or support-size.).

**Lemma 24.** Let $D(x)$ be a polynomial in $F^k[x]$. Suppose that $D(x)$ has a cone-closed basis. Then, $D(x)$ has $(k+1)$-cone concentration and $(\log 2k)$-support concentration.

**Proof.** Let $B$ be a cone-closed set of monomials forming the basis of $sp(D)$. Clearly, $|B| \leq k$. Thus, each $m \in B$ has cone-size $\leq k$. In other words, $D$ is $(k+1)$-cone concentrated.

Moreover, each $m \in B$ has support-size $\leq \lg k$. In other words, $D$ is $(\log 2k)$-support concentrated.

}$
Ideally, we would want to modify a given tiny circuit to get a cone-closed basis. This would solve the PIT problem as shown in the previous section. What are the possible ways to get this? We will show that the concept of basis isolating weight assignment, introduced in [AGKS15], leads to a cone-closed basis.

**Basis & weights.** Consider a weight assignment $w$ on the variables $x$. It extends to monomials $m = x^e$ as $w(m) := (e, w) = \sum_{i=1}^{\ell} e_i w_i$. Sometimes, we also use $w(e)$ to denote $w(m)$. Similarly, for a set of monomials $B$, the weight of $B$ is $w(B) := \sum_{m \in B} w(m)$.

Let $B = \{m_1, \ldots, m_{\ell}\}$ resp. $B' = \{m'_1, \ldots, m'_{\ell}\}$ be an ordered set of monomials (non-decreasing wrt $w$) that forms a basis of the span of coefficients of $f \in \mathbb{F}^k[x]$. Wrt $w$, we say that $B < B'$ if there exists $i \in [\ell]$ such that $\forall j < i$, $w(m_j) = w(m'_j)$ but $w(m_i) < w(m'_i)$. We say that $B \leq B'$ if either $B < B'$ or if $\forall i \in [\ell]$, $w(m_i) \leq w(m'_i)$. A basis $B$ is called a least basis, if for any other basis $B'$, $B \leq B'$. When is it unique?

A weight assignment $w$ is called a basis isolating weight assignment for a polynomial $f(x) \in \mathbb{F}^k[x]$ if there exists a basis $B$ such that:

1. weights of all monomials in $B$ are distinct, and
2. the coefficient of every $m \in \text{supp}(f) \setminus B$ is in the linear span of $\{\text{coef}_{m'}(f) \mid m' \in B, w(m') < w(m)\}$.

**Lemma 25.** If $w$ is a basis isolating weight assignment for $f$, then $f$ has a unique least basis $B$ wrt $w$. In particular, for any other basis $B'$ of $f$, we have $w(B) < w(B')$.

**Proof.** Let $\ell$ be the dimension of $\text{sp}(f)$. Since $w$ is a basis isolating weight assignment, we get a basis $B$ that satisfies the two conditions in the definition of $w$. We will show that $B$ is the unique least basis. Let $B = \{m_1, \ldots, m_{\ell}\}$ with $w(m_1) < \ldots < w(m_{\ell})$.

Consider any other basis $B' = \{m'_1, \ldots, m'_{\ell}\}$, with $w(m'_1) \leq \ldots \leq w(m'_{\ell})$. Let $j$ be the minimum number such that $m_j \neq m'_j$ (it exists as $B \neq B'$). Suppose $w(m_j) \geq w(m'_j)$. Since $m'_j \notin B$, the coefficient of $m'_j$ can be written as a linear combination of the coefficients of $m_i$'s for $i < j$. From the definition of $j$, for all $i < j$, $m_i = m'_i$. So the coefficient of $m'_j$ can also be written as a linear combination of the coefficients of $m_i$'s for $i < j$. This contradicts that $B'$ is a basis and proves that $w(m_j) < w(m'_j)$.

Now we move beyond $j$. First, we prove that for all $i \in [\ell]$, $w(m_i) \leq w(m'_i)$. For the sake of contradiction assume that there exists a number $a$ such that $w(m_a) > w(m'_a)$. Pick the least such $a$. Let $V$ be the span of the coefficients of monomials in $f$ whose weights are $\leq w(m'_a)$. Since, for all $i \in [a]$, the coefficient of $m'_i$ is in $V$ and all of them are linearly independent, we know that $\dim(V) \geq a$. On the other hand, for every monomial $m$ in $f$ of $w(m) \leq w(m'_a) < w(m_a)$, the coefficient of $m$ can be written as a linear combination of the coefficients of $m_i$'s where $i < a$. This implies that $\dim(V) < a$, which yields a contradiction. Thus, for all $i \in [\ell]$, $w(m_i) \leq w(m'_i)$.

In other words, $B \leq B'$. Together with $w(m_j) < w(m'_j)$, we get that $B < B'$ and $w(B) < w(B')$.

Next we want to study the effect of shifting $f$ by a basis isolating weight assignment. To do that we require an elaborate notation. As before $f(x)$ is an arity-$n$ degree-$d$ polynomial over $\mathbb{F}^k$.

For a weight assignment $w$, by $f(x + t^w)$ we denote the polynomial $f(x_1 + t^{w_1}, \ldots, x_n + t^{w_n})$. Let $M = \{a \in \mathbb{N}^n : |a|_1 \leq d\}$ correspond to the relevant monomials. For every $a \in M$, $\text{coef}_{x^a}(f(x + t^w))$ can be expanded using the binomial expansion, and we get:

$$\sum_{b \in M} \binom{b}{a} \cdot t^{w(b) - w(a)} \cdot \text{coef}_{x^a}(f(x)).$$

(3)

We express this data in matrix form as $F' = D^{-1}TD \cdot F$, where the matrices involved are,
1. $F$ and $F'$: rows are indexed by the elements of $M$ and columns are indexed by $[k]$. In $F$ resp. $F'$ the $a$-th row is $\text{coef}_x(a(f(x)))$ resp. $\text{coef}_x(a(f(x) + i^{w}))$.

2. $D$: is a diagonal matrix with both the rows and columns indexed by $M$. For $a \in M$, $D_{a,a} := i^{w(x^a)}$.

3. $T$: both the rows and columns are indexed by $M$. For $a, b \in M$, $T_{a,b} := \binom{b}{a}$.

We will prove the following combinatorial property of $T$: For any $B \subseteq M$, there is a cone-closed $A \subseteq M$ such that the submatrix $T_{A,B}$ has full rank. Our proof is an involved double-induction, so we describe the construction of $A$ as Algorithm 1.

<table>
<thead>
<tr>
<th>Algorithm 1 Finding cone-closed set</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input</strong>: A subset $B$ of the $n$-tuples $M$.</td>
</tr>
<tr>
<td><strong>Output</strong>: A cone-closed $A \subseteq M$ with full rank $T_{A,B}$.</td>
</tr>
<tr>
<td><strong>function</strong> $\text{FIND-CONE-CLOSED}(B, n)$</td>
</tr>
<tr>
<td>if $n = 1$ then</td>
</tr>
<tr>
<td>$s \leftarrow</td>
</tr>
<tr>
<td>return ${0, \ldots, s - 1};$</td>
</tr>
<tr>
<td>else</td>
</tr>
<tr>
<td>Let $\pi_n$ be the map which projects the set of monomials $B$ on the first $n - 1$ variables;</td>
</tr>
<tr>
<td>Let $\ell$ be the maximum number of preimages under $\pi_n$;</td>
</tr>
<tr>
<td>$\forall i \in [\ell]$, $F_i$ collects those elements in $\text{Img}(\pi_n)$ whose preimage size $\ge i$;</td>
</tr>
<tr>
<td>$A_0 \leftarrow \emptyset;$</td>
</tr>
<tr>
<td>for $i \leftarrow 1$ to $\ell$ do</td>
</tr>
<tr>
<td>$S_i \leftarrow \text{FIND-CONE-CLOSED}(F_i, n - 1);$</td>
</tr>
<tr>
<td>$A_i \leftarrow A_{i-1} \cup S_i \times {i - 1};$</td>
</tr>
<tr>
<td>end for</td>
</tr>
<tr>
<td>return $A;$</td>
</tr>
<tr>
<td>end if</td>
</tr>
<tr>
<td>end function</td>
</tr>
</tbody>
</table>

**Lemma 26** (Comparison). Let $B$ and $B'$ be two nonempty subsets of $M$ such that $B \subseteq B'$. Let $A = \text{FIND-CONE-CLOSED}(B, n)$ and $A' = \text{FIND-CONE-CLOSED}(B', n)$ in Algo. Then $A \subseteq A'$. Moreover, $|A| = |B|$.  

**Proof.** We prove the lemma using induction on $n$.

**Base case** ($n = 1$): For $n = 1$, the set $A$ is $\{0, \ldots, |B| - 1\}$ and the other one $A'$ is $\{0, \ldots, |B'| - 1\}$. Since $B$ is a subset of $B'$, $|B| \leq |B'|$. So $A$ is also a subset of $A'$.

**Induction step** ($n - 1 \rightarrow n$): Let $\ell$ resp. $\ell'$ be the bounds on the size of preimages of $\pi_n$ in $B$ resp. $B'$. To denote the set of all elements in $\text{Img}(\pi_n)$ whose preimage size $\ge i$, we use $F_i$ resp. $F'_i$. Since $B \subseteq B'$ we have $\ell \leq \ell'$, and for all $i \in [\ell']$, $F_i \subseteq F'_i$. So from induction hypothesis, $S_i \subseteq S'_i$. Since $A = \bigcup_{i=1}^{\ell'} S_i \times \{i - 1\}$ and $A' = \bigcup_{i=1}^{\ell'} S'_i \times \{i - 1\}$, we deduce that $A \subseteq A'$.

Note that $|A| = |B|$ is true when $n = 1$. Let us prove the induction step from $n - 1$ to $n$. Since $|A| = \sum_{i\in[\ell]} |S_i|$, and by induction hypothesis $|S_i| = |F_i|$, we deduce that $|A| = \sum_{i\in[\ell]} |F_i|$. From the definition of $F_i$'s we get that $\text{Img}(\pi_n) = F_1 \supseteq F_2 \supseteq \cdots \supseteq F_{\ell}$. A monomial $m \in \pi_n(B)$ that has preimage size $j$, is counted exactly $j$ times in $\sum_{i\in[\ell]} |F_i|$. Thus, $|A| = \sum_{i\in[\ell]} |F_i| = |B|$. 

**Lemma 27** (Closure). Let $B$ be a nonempty subset of $M$. If $A = \text{FIND-CONE-CLOSED}(B, n)$ in Algo, then $A$ is cone-closed.  
(Proved in Sec.D)
We recall a fact that has been used for ROABP PIT. (It requires char $\mathbb{F} = 0$ or large.)

**Lemma 28.** [GKST08, Clm.3.3] Let $a_1, \ldots, a_n$ be distinct non-negative integers. Let $A$ be an $n \times n$ matrix with, $i, j \in [n]$, $A_{i,j} := \binom{a_i}{a_j}$. Then, $A$ is full rank.

**Lemma 29** (Full rank). If $A = \text{Find-Cone-Closed}(B, n)$ then $T_{A,B}$ has full rank.

(Proved in Sec[D])

Now we are ready to prove our main theorem using the transfer matrix equation.

**Proof of Thm**. As we mentioned in Eqn[3], the shifted polynomial $f(x + t^w)$ yields a matrix equation $F' = D^{-1}TD \cdot F$. Let $k'$ be the rank of $F$. We consider the following two cases.

Case 1 ($k' < k$): We reduce this case to the other one where $k' = k$. Let $S$ be a subset of $k'$ columns such that $F_{M,S}$ has rank $k'$. The matrix $F_{M,S}$ denotes the polynomial $f_S(x) \in \mathbb{F}[x]^{k'}$, where $f_S(x)$ is the projection of the ‘vector’ $f(x)$ on the coordinates indexed by $S$. So, any linear dependence relation among the coefficients of $f(x)$ is also valid for $f_S(x)$. So $w$ is also a basis isolating weight assignment for $f_S(x)$. Now from our Case 2, we can claim that $f_S(x + t^w)$ has a cone-closed basis $A$. Thus, coefficients of the monomials, corresponding to $A$, in $f(x)$ form a basis of $\text{sp}(f)$. This implies that $f(x + t^w)$ has a cone-closed basis $A$.

Case 2 ($k' = k$): Let $B$ be the least basis of $f(x)$ wrt $w$ and $A = \text{Find-Cone-Closed}(B, n)$. We prove that the coefficients of monomials in $A$ form a basis of the coefficient space of $f(x + t^w)$. To prove this, we show that $\det(F_{A,[k]}'') \neq 0$. Define $T' := TDF$ so that $F' = D^{-1}T'$. Using Cauchy-Binet formula [Zen93], we get that

$$\det(F_{A,[k]}'') = \sum_{C \in \binom{M}{k} \setminus \{A\}} \det(D_{A,C}^{-1}) \cdot \det(T_{C,[k]}').$$

Since for all $C \in \binom{M}{k} \setminus \{A\}$, the matrix $D_{A,C}^{-1}$ is singular, we have $\det(F_{A,[k]}'') = \det(D_{A,A}) \cdot \det(T_{A,[k]}')$. Again applying Cauchy-Binet formula for $\det(T_{A,[k]}')$, we get

$$\det(F_{A,[k]}'') = \det(D_{A,A}) \cdot \sum_{C \in \binom{M}{k} \setminus \{A\}} t^{w(C)} \det(T_{A,C}) \cdot \det(F_{C,[k]}'').$$

From Lem[25], we have that for all basis $C \in \binom{M}{k} \setminus \{B\}$, $w(C) > w(B)$. The matrix $T_{A,B}$ is nonsingular by Lem[29] and the other one $F_{B,[k]}$ is nonsingular since $B$ is a basis. Hence, the sum is a nonzero polynomial in $t$. In particular, $\det(F_{A,[k]}'') \neq 0$, which ensures that the coefficients of the monomials corresponding to $A$ form a basis of $\text{sp}_F(f(x + t^w))$. Since Lem[27] says that $A$ is also cone-closed, we get that $f(x + t^w)$ has a cone-closed basis.

**5 Conclusion**

We introduce the tiny diagonal depth-4 (resp. tiny variants of depth-3, width-2 ABP and extremely low-arity $\Sigma \Pi \Sigma \Lambda$ or $\Sigma \Lambda^a \Sigma \Pi$) model with the motivation that its poly-time hitting-set would: (1) solve VP PIT (in quasipoly-time) via a poly-time arity reduction ($n \mapsto \log sd$), and (2) prove that either $E \subseteq \#P/poly$ or VNP has polynomials of algebraic complexity $2^{\Omega(n)}$. Since now we could focus solely on the PIT of sub-log-arity VP circuits, we initiate a study of properties that are useful in that regime. These are low-cone concentration and cone-closed basis. Using these concepts we solve a special case of diagonal depth-3 circuits. This work throws up a host of tantalizing models and poses several interesting questions.
Could the arity reduction phenomenon in Thm. 2 be improved (say, to the Ackermann function $A(n)$)?

Could we show that the $g$ in Cor. 7 is in VNP and not merely E-computable? This would tie blackbox PIT tightly with VNP $\neq$ VP.

Could we prove nontrivial lower bounds against the tiny models?

Could we solve PIT for size-$s$ $\Sigma \Pi \Sigma \land (n)$ in $\text{poly}(s, \mu(n))$-time, for some $\mu$?

Could we solve PIT for size-$s$ semantic individual-degree-$a'$ $\Sigma \Pi \Sigma (n)$ circuits in $\text{poly}(s2^n, \mu'(a'))$-time, for some $\mu'$?

Could we solve PIT for size-$s$ $\Sigma \Pi \Sigma (n)$ in $\text{poly}(s, \mu(n))$-time, for some $\mu$?

Could we do blackbox PIT for ROABP when $n = \omega(1)$? For instance, given oracle $C = \sum_{i \in [k]} \prod_{j \in [n]} f_{i,j}(x_j)$ of size $s$, we want a hitting-set in $\text{poly}(s, \mu(n))$-time, for some $\mu$. It is known that diagonal depth-3 blackbox PIT reduces to this problem if we demand $\mu(n) = 2^{O(n)}$ [FSS14].

Could we do blackbox PIT for size-$s$ arity-$(\log s)$ individual-degree-$(\log \log s)$ ROABPs?

Could we do blackbox PIT for size-$s$ arity $\omega(\log s)$ multilinear ROABPs?

In this work we have focused on PIT algorithms whose time-complexity depends polynomially on $s$ (circuit-size), while it could have super-polynomial dependence on $n$. It is a natural question (asked by Ramprasad Saptharishi): What can we conclude if we assume a mere $\exp(s^{o(1)})$-time blackbox PIT for one of these tiny models? Eg. can our methods be extended to get an arity-reducing $(n \rightarrow (sd)^{o(1)})$ polynomial-map for VP PIT? We do discuss this briefly in Sec. B suggesting that an $\exp((sd)^{o(1)} + O(n))$-time blackbox PIT for VP would already get “close” to separating VNP from VP (Meta-Thm. 1). An analogous connection with tiny models is left as an open question.

Acknowledgements

We thank Ramprasad Saptharishi for many useful discussions. M.F. & N.S. thank the organizers of algebraic complexity workshops in 2014 (MPI Saarbrücken & TIFR Mumbai) that initiated the early discussions. S.G. & N.S. thank the organizers, and participants, of the IMSc Chennai workshop in 2017 for discussing the preliminary results. N.S. thanks the funding support from DST (DST/SJF/MSA-01/2013-14).

References


Figure 1: The myriad tiny models and the implications (in bold arrows). See Meta-Thm.1 for weaker hypotheses.
hitting-set for size-s, deg-d, arity-log^c sd

Lem.6

hitting-set for size-s, deg-d, arity-ω(log^c+1 sd)

Lem.17 with ε = ε_c+1

(s^c, s^ε)-hsg for C_{s,s,n}

n = c_1ε_c+1(s)

Lem.6

{q_m,s} of ideg s^{3ε/m}, it’s computable in s^{Ω(1)}-time but not by size < s = ε_{c+1}^{-1}(Θ(m)) circuits

reduces arity

Lem.17 with ε = ε_c

hitting-set for size-s, deg-d, arity-ω(log^c sd)

Lem.6

(s^c, s^ε)-hsg for C_{s,s,n}

n = c_1ε_c(s)

{q_m,s} is hard with appropriate parameters

reduces arity

Repeatedly use Lem.17

hitting-set for size-s, deg-d, arity-ω(log^c-1 sd)

Lem.6

(s^c, s^ε)-hsg for C_{s,s,n}

n = c_1ε_{c-1}(s)

{q_m,s} is hard with appropriate parameters

reduces arity

Lem.17 or 10

hitting-set for size-s, deg-d, arity-ω(log sd)

Lem.6

(s^c, s^ε)-hsg for C_{s,s,n}

n = c_1ε_1(s)

{q_m,s} is hard with appropriate parameters

reduces arity

poly-time hitting-set for size-s, deg-d, var-n circuits

Figure 2: Proof outline of Thm.2 (i.e. c + 1 invocations of NW design).


A Proofs from Sec.2: Other tiny models

If a circuit $C$ computes a polynomial of individual-degree $\leq d_0$ then we say that the 
\textit{semantic individual-degree} bound of $C$ is $d_0$. Recall the definition of the tiny depth-3 model (Defn.12).

\textsc{Theorem 14 (restated). [Tiny depth-3]} If we have poly-time hitting-set for a tiny depth-3 model, then for VP circuits we have a $\poly(sd)$-time arity reduction ($n \mapsto O(\log sd)$) that preserves nonzeroness (and proves an exponential lower bound).

\textit{Proof.} The proof strategy is identical to that of Thm.1. So, we will only sketch the main points here.

Let constant $c > 1$, and function $\mu'$ be given in the hypothesis. Let $C \subset T_{\mu',e}$ be the set of tiny depth-3 circuits of size $\leq s$ and arity $m := \log s$. Assume that $C$ has a $(s^c,s^e)$-hsg $f(y)$ for some constant $c \geq 1$. Then using Lem.6, we have an $m$-variate homogeneous polynomial $q_m$ with individual-degree less than some constant $\delta$, such that $\delta^{m/3} = \delta^{O(s)/3} > s^{e}$, and $q_m$ is computable in $s^{O(1)} = \delta^{O(m)}$ time. It has degree $= (\delta - 1)m/2$. Importantly, $q_m \notin T_{\mu',e}$, thus, no tiny depth-3 circuit of size $\leq s = \delta^{O(m)}$ can compute it (otherwise, $q_m(f(y)) \neq 0$ which contradicts its definition as an annihilator). Next we show that it is also not computable by any $\delta^{o(m)}$-size algebraic circuit.

For the sake of contradiction, assume that $q_m$ has a $\delta^{o(m)}$-size circuit. Repeat the depth-reduction arguments, as in the proof of Thm.1. Let $a := 5'$ and $b := (\delta - 1)m/2^e+1$. Note that we can ensure $a,b = o(\log s)$, $a = \omega(1)$, and we have a shallow circuit for $q_m$ of the form homogeneous $\Sigma \mu \Sigma \Pi^b$.

It was shown in [OKKS13] that any size-$s'$ $\Sigma \Pi^a \Sigma \Pi^b(n)$ circuit can be transformed to a $\poly(s^2s'^b)$-size $\Sigma \Pi \Sigma^b(n)$ circuit. Applying it here, we get a depth-3 circuit $C'$, computing $q_m$, of the form $\Sigma \Pi \Sigma$ and size $\delta^{o(m)} \cdot 2^{a+b} = \delta^{o(m)}$. Moreover, the measure $2^m + \mu'(\delta) = s + O(1) = o(s^e)$. Thus, $C'$ is a tiny depth-3 circuit in $T_{\mu',e}$, of size $\delta^{o(m)}$ which is $< s$. This contradicts the hardness of $q_m$. Thus, there is no algebraic circuit for $q_m$ of size $\delta^{o(m)}$.

Now invoking Lem.10 on the hard family $\{q_m\}_{m \geq 1}$, we get our claim. Moreover, by Cor.7, $\{q_m\}_m$ is an E-computable homogeneous polynomial family that has algebraic circuit complexity $2^{\Omega(m)}$.

We will now consider the polynomials that can be computed by upper-triangular width-2 algebraic branching programs (ABP).

\textsc{Theorem 15 (restated). [Width-2 ABP]} If we have $\poly(s2^n)$-time hitting-set for size-$s$ arity-$n$ width-2 upper-triangular ABP, then for VP circuits we have a $\poly(sd)$-time arity reduction ($n \mapsto O(\log sd)$) that preserves nonzeroness (and proves an exponential lower bound).
Proof. In [SSS09, Thm.3] an efficient transformation was given that rewrites a size-$s$ arity-$n$ depth-$3$ circuit, times a special product of a linear polynomials, as a poly($s$)-size arity-$n$ width-$2$ upper-triangular ABP. Thus, a poly($s^{2n}$)-time hitting-set for the latter model gives a poly($s^{2n}$)-time hitting-set for the former. This by Thm[14] gives us the exponentially hard polynomial family $\{q_m\}_{m \geq 1}$ that is E-computable. Now invoking Lem[10] on the hard family, we get the arity reduction.

THEOREM 16 (repeated). [Multilinear tiny depth-3] If we have poly-time hitting-set for a multilinear tiny depth-3 model, then for VP circuits we have a poly($sd$)-time arity reduction $(n \mapsto O(\log sd))$ that preserves nonzeroness (and proves an exponential lower bound).

Proof. The proof strategy is identical to that of Thm[1]. So, we will only sketch the main points here. Let constant $c > 1$, and function $\mu$ be given in the hypothesis. Let $a' = a'(s)$ be a 'small' unbounded function determined by the constraint $\mu'(a') \leq s$. Assume that size-$s$ arity-$a'(\log s)$ polynomials in $M_{\mu',c}$ have a $(s^e,s^e)$-hsg $f(y)$ for some constant $e \geq 1$.

Let $C \subset M_{\mu',c}$ be the set of multilinear tiny depth-3 circuits of size $\leq s$ and arity $m := 1 + 3e \log s < a'(\log s)$. Since $C$ has the $(s^e,s^e)$-hsg $f(y)$, so using Lem[6] we have an arity $m$ multilinear homogeneous annihilating polynomial $q_m(x_1,\ldots,x_m)$, which is computable in $s^{O(1)}$ time (note: $2^m/3 > 2^{e \log s} = s^e$). It has degree $m/2$. Importantly, $q_m \notin M_{\mu',c}$, thus, no multilinear tiny depth-3 circuit of size $\leq s = 2^{\Theta(m)}$ can compute it (otherwise, $q_m(f(y)) \neq 0$ which contradicts its definition as an annihilator). Next we show that it is also not computable by any $2^{o(m)}$-size algebraic circuit.

For the sake of contradiction, assume that $q_m$ has an $2^{o(m)}$-size circuit. Repeat the depth-reduction arguments, as in the proof of Thm[1] after cutting at depth $t = \omega(1)$. Let $a := 5t$ and $b := m/2^t$. Note that we can ensure $a,b = o(m) = o(\log s)$, $a = o(1)$, and we have a shallow circuit for $q_m$ of the form homogeneous $\Sigma^e \Pi^b$.

It was shown in [GKKS13] that any size-$s'$ $\Sigma^e \Pi^b(n)$ circuit can be transformed to a poly($s'2^a+b$)-size $\Sigma^e \Pi^b(n)$ circuit. Applying it here, we get a depth-3 circuit $C'$, computing $q_m$, of the form $\Sigma^e \Pi^b$ and size $2^{o(m)} \cdot 2^a+b = 2^{o(m)}$. Moreover, the measure $\mu'(m/\log s) \leq \mu'(3e+1) = O(1) = o(s^e)$. Thus, $C'$ is a multilinear tiny depth-3 circuit in $M_{\mu',c}$, of size $2^{o(m)}$ which is $< s$. This contradicts the hardness of $q_m$. Thus, there is no algebraic circuit for $q_m$ of size $2^{o(m)}$.

Now invoking Lem[10] on the hard family $\{q_m\}_{m \geq 1}$, we get our claim. Moreover, by Cor[7] $\{q_m\}_{m \geq 1}$ is an E-computable homogeneous polynomial family that has algebraic circuit complexity $2^{\Omega(m)}$.

THEOREM 18 (repeated). [Tinier n,a] If, for some $\mu, \mu'$, we have poly($s,\mu(a),\mu'(n)$)-time hitting-set for size-$s$ $\Sigma^{\wedge a} \Pi(n)$ circuits, then for VP circuits we have a poly($sd$)-time arity reduction $(n \mapsto O(\log sd))$ that preserves nonzeroness (and proves an exponential lower bound).

Proof. Suppose we have a poly($s,\mu(a),\mu'(n)$)-time hitting-set $H_{s,a,n}$ for size-$s$ $\Sigma^{\wedge a} \Pi(n)$ circuits. Wlog we can assume that $\mu'(n) = \Omega(n)$. Let $a' = a'(s) = \omega(1)$ be a function satisfying $\mu'(a') \leq s$, and define $n' = n(s) := a' \log s$. Consider a size-$s$ $\Sigma^{\wedge a} \Pi(n')$ circuit $C \neq 0$ computing a multilinear polynomial. We intend to design a hitting-set for $C$.

Partition the variable set $\{x_1,\ldots,x_n\}$ into $a'$ blocks $B_{ij}, j \in [a']$, each of size $\log s$. Recall the map $\varphi$ designed in the proof of Thm[3]. We have $\varphi(C) \neq 0$ (basically, use the fact that $C$ computes a nonzero multilinear polynomial and $\varphi$ keeps the multilinear monomials distinct). Finally, $\varphi(C)$ becomes an arity-$a'$ $\Sigma^{\wedge a} \Pi$ circuit of size at most $s + s \cdot 2^{\log s} = O(s^2)$. Thus, using $H_{O(s^2),a,a'}$ we get a hitting-set for $\varphi(C)$ of time-complexity $\text{poly}(s^2,\mu(a),\mu'(a')) = \text{poly}(s,\mu(a))$. In turn, we get a poly($s,\mu(a)$)-time hitting-set for $\Sigma^{\wedge a} \Pi$ circuits that compute multilinear polynomials of arity $a' \log s = \omega(\log s)$. Using an argument identical to that in the proof of
Moreover, in depth-reduction to depth-4 we incur a ‘huge’ cost (close to $2^{o(q)}$ that the function $f$ complexity is a sub-exp function.

In sub-exponential time, then $f(s) = \omega(1)$ if $s \in O(1)$ and the hardness is circuit size $s = f^{-1}(2^m)^{\Omega(1)}$. Since $f$ is sub-exp we can deduce the lower bound $s = m^\omega(1)$ for depth-4. The latter is clearly too weak to apply depth-reduction techniques and so we are unable to say anything about the (general) circuit complexity of $q_m$.

The proof of Thm.2 is better in this respect as it does not need depth-reduction. Though, it uses many-fold composition of the Nisan-Wigderson design. So, now we need the function $f$ to satisfy the self-composition property: The function $f(s^{O(1)})$ should not grow “too much” faster than $f(s)$. One such class of functions that we can think of is sub-exp:

**Definition 30.** A function $f(s)$ is in sub-exp if $f(s) = \exp(s^{\omega(1)})$.

One can recall the standard complexity class, $\text{SUBEXP} := \cap_{c>0} \text{DTIME}(\exp(n^c))$. Basically, these are decision problems whose time-complexity is a sub-exp function.

(2) In depth-reduction to depth-4 we need the individual-degree of $q_m$ to be $O(1)$, to avoid a blowup in the resulting circuit size. This means that we have to use arity $m = \log f(s^{O(1)})$, so that the annihilating polynomial $q_m$ exists (in Lem.4).

(3) Moreover, in depth-reduction to depth-4 we incur a ‘huge’ cost (close to $2^{o(m)}$). Thus, the depth-4 hardness of $q_m$ should be $2^{\Omega(m)}$ and not merely $m^{\omega(1)}$.

The annihilating polynomial idea gives us an E-computable polynomial family $\{q_m\}_m$ of constant individual-degree, where the arity $m$ is essentially $\log f(s^{O(1)})$ and the hardness is circuit size $s = f^{-1}(2^m)^{\Omega(1)}$. Since $f$ is sub-exp we can deduce the lower bound $s = m^{\omega(1)}$ for depth-4. The latter is clearly too weak to apply depth-reduction techniques and so we are unable to say anything about the (general) circuit complexity of $q_m$.

The proof of Thm.2 is better in this respect as it does not need depth-reduction. Though, it uses many-fold composition of the Nisan-Wigderson design. So, now we need the function $f$ to satisfy the self-composition property: The function $f(f(s^{O(1)}))$ should not grow “too fast”. One such class of functions that we can think of is sub-sub-exp $[AG94]$.

**Definition 31.** For a constant $c \in \mathbb{N}$, sub$_c$exp is the class of functions such that for any $f(s)$ in sub$_c$exp, the $c$-fold composition $f^{\circ c}$ is in sub-exp.

Note that sub$_1$exp is the same as sub-exp.

One can define a complexity class $\text{SUB}_c\text{EXP}$ that collects decision problems whose time-complexity is a sub$_c$exp function.

An instructive example of such functions is, for a constant $A \in \mathbb{N}$, consider $f(s) \in \exp^{\circ A}(O(\log^{oA}(s)))$. It can be seen that this function is in sub$_c$exp. For instance, $A = 1$ gives poly$(s)$, $A = 2$ gives quasipoly$(s)$, and $A = 3$ gives $\exp((\log s)^{\text{poly}(\log \log s)})$.

**Meta-Theorem 1 (sub-exp).** Let $f$ be a sub$_{c+1}$exp (resp. sub$_1$exp) function for a constant $c \geq 2$ (resp. $c = 1$). If one can do blackbox PIT for VP in $f(s)d + \exp^{\circ c}(O(n))$ time, then blackbox PIT for VP is in $\text{SUBEXP}$.

Moreover, one gets an E-computable polynomial family $\{q_m\}_m$ whose algebraic circuit size is $m^{\omega(1)}$. So, either $E^\Sigma P/poly$ or $\text{VNP} \neq \text{VP}$.

**Proof sketch.** Let us first see the ideas with $c = 1$.

The annihilating polynomial idea gives us an $f(s^{O(1)})$-time computable (in the end E-computable) polynomial family $\{q_m\}_m$ of constant individual-degree, where the arity $m$ is
essentially $\log f(s^{O(1)})$, and the hardness\footnote{Technically, we will require that $m \leq n$ and $\deg(q_m) \leq d$. So, we have to work with appropriate triples $(s,d,n)$.} is circuit size $s = f^{-1}(2^n)^{O(1)}$. Since $f$ is subexp we can deduce the lower bound $s = m^{\omega(1)}$. This already implies that: Either $E \not \subseteq \#P/poly$ or $\text{VNP} \not \subseteq \text{VP}$.

Now we use the Nisan-Wigderson design with parameters $(\log f(s^{O(1)}), \log f(s^{O(1)}), \log s')$, where $s' := sd$ corresponds to the general circuit family whose arity reduction we want. The design will take time $\exp((\log^2 f(s^{O(1)})) / \log s')$. The arity will get reduced from $n$ to $\log f(s^{O(1)})$.

Note that in the proof of Lem.\ref{lem:10} the circuit $C_2$, on which Kaltofen's factorization is applied, has size $s^{O(1)}$; so, we can prove the arity reduction claim by fixing the constant parameters appropriately.

This arity reduction implies a blackbox PIT algorithm, for algebraic circuits of size-$s$ degree-$d$ arity-$n$, with time-complexity

$$2(\log d) \log f((sd)^{O(1)}) + 2(\log^2 f((sd)^{O(1)})) / \log(sd)$$

which is again in subexp, and is hence nontrivial.

When $c > 1$, then we will need to further reduce the arity from $\log f(s^{O(1)})$ to $\log \log f(s^{O(1)})$, and so on. The proof of Thm.\ref{thm:2} requires $(c + 1)$ phases (see Fig.\ref{fig:2}). So, we require $f$ to be in sub$_{c+1}\text{exp}$ to get a nontrivial result.

We leave the detailed calculations for a later version of the paper. \hfill \Box

\section{Proofs from Sec.\ref{sec:3}}

An arity-$n$ depth-$3$ diagonal circuit over $\mathbb{F}$ can be written as $C(x) = \sum_{i=1}^{k} c_i \tilde{f}_i^d$, where $\tilde{f}_i$'s are linear polynomials. Let $f_i$ be the non-constant part of $\tilde{f}_i$ for all $i \in [k]$. Suppose that $\text{rk}_F\{f_1, \ldots, f_r\} =: r$. Wlog, we can assume that $f_1, \ldots, f_r$ is a basis of the space spanned by $f_i$'s. Then there exists an $r$-variate polynomial $A(z)$ such that $C(x) = A(f_1, \ldots, f_r)$. Let $L_\mathbb{F}[x]$, where $x = (x_1, \ldots, x_n)$, resp. $L_\mathbb{F}[y]$, where $y = (y_1, \ldots, y_r)$, be the vector space of linear polynomials over $\mathbb{F}$.

Using the construction of \cite[Sec.3.2]{SS12}, in poly$(kd)$-time, we can find a linear transformation $\Psi : L_\mathbb{F}[x] \rightarrow L_\mathbb{F}[y]$ such that $\text{rk}_F\{\Psi(f_1), \ldots, \Psi(f_r)\} = r$ and $g_i := \Psi(f_i)$ are linear forms (i.e. homogeneous and degree one). Now we prove the following fact which will ensure the non-zeroness of $C(\Psi(x))$.

\begin{lemma}
If $A(g_1, \ldots, g_r) = 0$ then $A$ is the zero polynomial.
\end{lemma}

\begin{proof}
Since $g_1, g_r$ are $r$ linearly independent linear forms on $(y_1, \ldots, y_r)$, we have an invertible linear map $\tau$ from $L_\mathbb{F}[y]$ to itself such that $\tau(g_i) = y_i$, equivalently, $\tau^{-1}(y_i) = g_i$. Thus, $\tau^{-1}$ induces an $\mathbb{F}$-automorphism $\tilde{\tau}$ on $\mathbb{F}[y]$.

Suppose that $A(g_1, \ldots, g_r) = 0$. Then, applying $\tilde{\tau}$ on $A(g_1, \ldots, g_r)$, we get $A(y_1, \ldots, y_r) = 0$, thus $A(z) = 0$. \hfill \Box
\end{proof}

\begin{theorem}[restated] Let $C$ be the set of all arity-$n$ degree-$d$ size-$s$ depth-$3$ diagonal circuits. Suppose that, for all $C \in C$, $\text{rk}(C) = O(\log sd)$. Then, the blackbox PIT for $C$ can be solved in $\text{poly}(sd)$-time.
\end{theorem}

\begin{proof}
The above description gives us a nonzeroness preserving arity reduction $(n \rightarrow \text{rk}(C))$ method that reduces $C$ to an $O(\log(sd))$-variate degree-$d$ poly$(s)$-size depth-$3$ diagonal circuit $C'$.

\end{proof}
Clearly the dimension of the partial derivative space of \( C' \) is \( \text{poly}(sd) \) [For'14, Lem.8.4.8]. Hence, Cor 22 gives us a \( \text{poly}(sd) \)-time hitting-set for \( C' \). \( \square \)

D Proofs from Sec.4

**Lemma 27 (repeated).** Let \( B \) be a nonempty subset of \( M \). If \( A = \text{Find-Cone-Closed}(B, n) \) in Algo 1 then \( A \) is cone-closed.

**Proof.** We prove it by induction on \( n \).

*Base case (\( n = 1 \)):* For \( n = 1 \), \( A = \{0, \ldots, |B| - 1\} \). So \( A \) is cone-closed.

*Induction step (\( n - 1 \to n \)):* Now \( A = \bigcup_{i=1}^{n} S_i \times \{i - 1\} \). Let \( f \) be an element in \( A \) and \( x^e \) be a submonomial of \( x^f \). We will show that \( e \in A \). Let \( f = (f', k) \) and \( e = (e', t) \), so that \( t \leq k \). We divide our proof into the following two cases.

*Case 1 (\( t = k \)):* We have \( f' \in S_{k+1} = \text{Find-Cone-Closed}(F_{k+1}, n - 1) \). By induction hypothesis, \( S_{k+1} \) is cone-closed. Since \( e' \leq f' \), we get \( e' \in S_k \). So, \( e = (e', k) \in S_{k+1} \times \{k\} \), which implies that it is also in \( A \).

*Case 2 (\( t < k \)):* We have \( F_{k+1} \subseteq F_{t+1} \). By Lem 26 we get \( S_{k+1} \subseteq S_{t+1} \). So \( f' \in S_{t+1} \). From induction hypothesis, \( S_{t+1} \) is a cone-closed set. This implies that \( e' \in S_{t+1} \) and \( e \in S_{t+1} \times \{t\} \). Thus, \( e \) is also in \( A \).

Since \( e \) was arbitrary, we deduce that \( A \) is cone-closed. \( \square \)

**Lemma 29 (repeated).** If \( A = \text{Find-Cone-Closed}(B, n) \) then \( T_{A,B} \) has full rank.

**Proof.** The proof will be by double-induction: outer induction on \( n \) and an inner induction on iteration \( i \) of the ‘for’ loop (Algo 1).

*Base case: * For \( n = 1 \), the claim is true due to Lem 28.

*Induction step (\( n - 1 \to n \)):* To show \( T_{A,B} \) full rank, we prove that for any vector \( b \in \mathbb{F}^{[R]} \): if \( T_{A,B} \cdot b = 0 \) then \( b = 0 \). For this we show that the following invariant holds at the end of each iteration \( i \) of the ‘for’ loop (Algo 1).

*Invariant (arity- \( n \) \& \( i \)-th iteration):* For each \( f \in B \) such that the preimage size of \( \pi_n(f) \) is at most \( i \), the product \( T_{A,B} \cdot b = 0 \) implies that \( b_{\pi_i} = 0 \).

At the end of iteration \( i \), we have the vector \( T_{A_i,B} \cdot b \). Recall that \( A_1 = S_i \times \{0\} \) and \( F_1 = \pi_n(B) \). So \( T_{A_1,B} \cdot b = T_{S_i,F_1} \cdot c \), where for \( e \in F_1 \), \( c_e := \sum_{(e,k) \in \pi_n^{-1}(e)^{(k)}} b(e,k) \). Thus, \( T_{A_1,B} \cdot b = 0 \) implies \( T_{S_i,F_1} \cdot c = 0 \). Since \( S_i = \text{Find-Cone-Closed}(F_1, n - 1) \), using induction hypothesis, we get that \( c = 0 \). This means that for \( e \in B \) such that the preimage size of \( \pi_n(e) \) is at most \( i \), we have \( c_e = 0 \). This proves our invariant at the end of the iteration \( i = 1 \).

(*i - 1 \to i*: Suppose that at the end of \((i-1)\)-th iteration, the invariant holds. We show that it also holds at the end of the \(i\)-th iteration. For each \( j \in [i] \), let \( v_j \) denote the projection of \( T_{A,B} \cdot b \) on the coordinates indexed by \( S_j \setminus \{j - 1\} \). By focusing on the latter rows of \( T_{A,B} \), we can see that \( v_j = T_{S_j,F_1} \cdot c_j \), where the vector \( c_j \) is defined as, for \( e \in F_1 \),

\[
(c_j)_e := \sum_{(e,k) \in \pi_n^{-1}(e)} \binom{k}{j-1} b(e,k).
\]

Suppose that \( T_{A,B} \cdot b = 0 \). All we have to argue is that for every \( f \in B \) such that the preimage size of \( e := \pi_n(f) \) is \( i \), the coordinate \( b_{\pi_i} = 0 \).

Since \( T_{A,B} \cdot b = 0 \), its projection \( v_j = T_{S_j,F_1} \cdot c_j \) is zero too. By induction hypothesis (on \( i - 1 \)), for each \( e \in F_1 \) with preimage size < \( i \), the coordinate \( (c_j)_e = 0 \). Thus, the vector
\[ T_{S_j,F_j} \cdot c_j = T_{S_j,F_j} \cdot c_j' \] where the vector \( c_j' \) is defined as, for \( e \in F_j \), \( (c_j')_e := c_{je} \). Consequently, \( T_{S_j,F_j} \cdot c_j' = 0 \), for \( j \in [i] \). By induction hypothesis (on \( n - 1 \)), we know that \( T_{S_j,F_j} \) is full rank. So \( c_j' = 0 \), which tells us that \( c_j = 0 \), for \( j \in [i] \).

Fix an \( e \in F_1 \), with preimage size = \( i \), and let the preimages be \( \{(e, k_1), \ldots, (e, k_i)\} \) where \( k_j \)'s are distinct nonnegative integers. Since \( c_j = 0 \), for \( j \in [i] \), we get from Eqn.4 and Lem.28 that: \( b_{(e, k_j)} = 0 \) for all \( j \in [i] \). In other words, for any \( f \in B \) such that the preimage size of \( \pi_n(f) \) is \( i \), the coordinate \( b_f = 0 \).

\( (i = \ell) \): Since \( A = A_\ell \), the output of \( \text{Find-Cone-closed}(B, n) \), using our invariant at the end of \( \ell \)-th iteration we deduce that \( T_{A,B} \cdot b = 0 \) implies \( b = 0 \). Thus, \( T_{A,B} \) has full rank.

### E Models with a cone-closed basis

We give a simple proof showing that a typical diagonal depth-3 circuit is already cone-closed. Consider the polynomial \( D(x) = (1 + a_1x_1 + \ldots + a_nx_n)^d \) in \( \mathbb{F}^k[x] \), where \( \mathbb{F}^k \) is seen as an \( \mathbb{F} \)-algebra with coordinate-wise multiplication.

**Lemma 33.** \( D(x) \) has a cone-closed basis.

**Proof.** Consider the \( n \)-tuple \( L := (a_1, \ldots, a_n) \). Then for every monomial \( x^e \), the coefficient of \( x^e \) in \( D \) is \( L^e := \prod_{i=1}^{n} a_i^{c_i} \), with some nonzero scalar factor (note: here we seem to need \( \text{char}(\mathbb{F}) \) zero or large). We ignore this constant factor, since it does not affect linear dependence relations.

Consider any proper monomial ordering \( \prec \) (eg. \( \text{deg-lex} \)). Now we prove that the ‘least basis’ of \( D(x) \) with respect to this monomial ordering is cone-closed.

We incrementally devise a monomial set \( B \) as follows: Arrange all the monomials in ascending order. Starting from least monomial, put a monomial in \( B \) if its coefficient can not be written as a linear combination of its previous (thus, smaller) monomials. From construction, the coefficients of monomials in \( B \) form the least basis for the coefficient space of \( D(x) \). Now we show that \( B \) is cone-closed. We prove it by contradiction.

Let \( x^f \in B \) and let \( x^e \) be its submonomial that is not in \( B \). Then we can write

\[
L^e = \sum_{x^b \prec x^e} c_b L^b \quad \text{with } c_b \text{'s in } \mathbb{F}.
\]

Multiplying by \( L^{f-e} \) on both sides, we get

\[
L^f = \sum_{x^b \prec x^e} c_b L^{b+f-e} = \sum_{x^{b'} \prec x^f} c'_{b'} L^{b'}.
\]

Note that \( x^{b'} \prec x^f \) holds true by the way a monomial ordering is defined. This equation contradicts the fact that \( x^f \in B \), and completes the proof.