A Largish Sum-of-Squares Implies Circuit Hardness and Derandomization

Pranjal Dutta * Nitin Saxena † Thomas Thierauf ‡

Abstract

For a polynomial f, we study the *sum of squares representation* (SOS), i.e. $f = \sum_{i\in[s]} c_i f_i^2$, where c_i are field elements and the f_i's are polynomials. The size of the representation is the number of monomials that appear across the f_i's. Its minimum is the *support-sum* $S(f)$ of f.

For simplicity of exposition, we consider univariate f. A trivial lower bound for the support-sum of, a full-support univariate polynomial, f of degree d is $S(f) \geq d^{0.5}$. We show that the existence of an explicit polynomial f with support-sum just slightly larger than the trivial bound, that is, $S(f) \geq d^{0.5+\varepsilon(d)}$, for a sub-constant function $\varepsilon(d) > \omega(\sqrt{\log\log d / \log d})$, implies that VP $\neq$ VNP. The latter is a major open problem in algebraic complexity. A further consequence is that blackbox-PIT is in SUBEXP. Note that a random polynomial fulfills the condition, as there we have $S(f) = \Theta(d)$.

We also consider the *sum-of-cubes representation* (SOC) of polynomials. In a similar way, we show that here, an explicit hard polynomial even implies that blackbox-PIT is in P.

2012 ACM CCS concept: Theory of computation - Algebraic complexity theory, Problems, reductions and completeness, Pseudorandomness and derandomization; Computing methodologies - Algebraic algorithms; Mathematics of computing - Combinatoric problems.
Keywords: VP, VNP, hitting set, circuit, univariate, polynomial, squares, cubes, SOS, SOC, PIT, lower bound, sparsity, monomials, support, explicit.

1 Introduction

The sum-of-squares representation (SOS) is one of the most fundamental in number theory and algebra. Lagrange's four-squares theorem inspired generations of mathematicians [Ram17]. Hilbert's *17th problem* asks whether a multivariate polynomial, that takes only non-negative values over the reals, can be represented as an SOS of rational functions [Pfi76]. In engineering, SOS has found many applications in approximation, optimization and control theory, see [Rez78, Las07, Lau09, BM16]. In this work, we show a connection to central complexity questions.

Consider the following basic problem on the size of SOS-representations.

Open Problem. *Exhibit an explicit univariate polynomial $f(x) \in \mathbb{C}[x]$ of degree d such that any SOS-representation $f(x) = \sum_i f_i(x)^2$ requires $\sum_i$ sparsity$(f_i) > \omega(\sqrt{d})$.*

Before delving into the meaning of *explicitness*, note that $\Omega(\sqrt{d})$ is a trivial lower bound, for a polynomial of degree d, with full support (by counting monomials). Moreover, for most polynomials f, a larger lower bound of $\Omega(d)$ holds, by a dimension argument. In other words, we ask for an explicit polynomial $f(x)$ that has a merely largish $\sum\bigwedge^2\sum\bigwedge$-formula. We show that one can bootstrap the seemingly weak hardness condition for SOS to general circuits (see Theorem 1) and to the infamous *determinant vs. permanent* question (see Corollary 4).

*Chennai Mathematical Institute, India (& CSE, IIT Kanpur), pranjal@cmi.ac.in
†CSE, Indian Institute of Technology, Kanpur, nitin@cse.iitk.ac.in
‡Aalen University, Germany, thomas.thierauf@uni-ulm.de

1.1 Algebraic circuits and univariate polynomials

Valiant defined the algebraic complexity classes VP and VNP based on algebraic circuits (for definitions see Section 2). They are considered as the algebraic analog of boolean classes P and NP. Separating VP from VNP is a long-standing open problem. One of the popular ways has been via depth-reduction results [AV08, Koi12, GKKS13, Tav15]. It seems that showing strong lower bounds require a deeper understanding of the algebraic-combinatorial structure of circuits, which may be easier to unfold for more analytic models that appear in wider mathematics.

It is known that 'most' of the polynomials of degree d are *hard*, i.e. they require $\Omega(d)$ size circuits; for a self-contained proof, see [CKW11, Theorem 4.2] [1]. In fact, for p_i being the i-th prime, $\sum_{i=0}^{d} \sqrt{p_i}\, x^i$ and $\sum_{i=0}^{d} 2^{2^i} x^i$, both require circuits of size $\Omega\left(d/\log d\right)$, see [BCS13, Cor.9.4] & [Str74]. Such polynomials can be converted to an *exponentially hard* multilinear polynomial $f_n(\boldsymbol{x})$. Unfortunately, this *strong* lower bound is insufficient to separate VP and VNP because the polynomial family is *non-explicit*– so f_n may not be in VNP. For details, see [HS80, Bür13].

Thus, the explicitness of the family plays a major role in its usefulness in algebraic complexity.

Definition 1 (Explicit functions)**.** *Let $(f_d)_d$ be a polynomial family, where $f_d(x)$ is of degree d. The family is* explicit*, if its coefficient-function is computable in time* $\mathsf{poly}\log(d)$ *and each coefficient can be at most* $\mathsf{poly}(d)$*-bits long. The coefficient-function gets input (j, i, d) and outputs the j-th bit of the coefficient of x^i in f_d.*

Alternative versions of explicitness define the coefficient-function to be computable in $\#\mathsf{P}/\mathsf{poly}$ or CH, which would be good enough for our purpose (see Theorem 3).

An *explicit* candidate for the hard family is the *Pochhammer-Wilkinson* polynomial, $f_d(x) := \prod_{i=1}^{d}(x-i)$.[2] Other explicit families, but *not* hard, are $(x+1)^d$ and the *Chebyshev* polynomial (that writes $\cos d\theta$ as a function of $\cos\theta$) [MH02]. These three are quite relevant to this work.

The interplay between proving lower bounds and derandomization is one of the central themes in complexity theory [NW94]. Blackbox Polynomial Identity Testing (PIT) asks for an algorithm to test the zeroness of a given algebraic circuit via mere query access. It is still an open question to design an efficient deterministic PIT algorithm. A circuit of size s can have $\exp(s)$ many monomials. However, since a non-zero polynomial evaluated at a random point is non-zero with high probability (by the *Polynomial Identity Lemma* [Ore22, DL78, Zip79, Sch80]), one gets a randomized poly-time algorithm for PIT. For PIT refer [Sax09, Sax14, SY10, Mul12, Wig17].

One important direction, from hardness to derandomization, is to design deterministic PIT algorithms for small circuits assuming access to *explicit hard polynomials* [NW94, KI04]. Most of the constructions use the concept of *hitting-set generator* (HSG), see Definition 8. Very recent work discovered that PIT is amenable to the phenomenon of *bootstrapping* (w.r.t. variables) [AGS19, KST19]. Finally, Guo et al. [GKSS19] showed: ample circuit-hardness of *constant*-variate polynomials (including univariate) implies blackbox-PIT in P.

1.2 Sum-of-squares model (SOS)

We want to relate variants of SOS to PIT and circuit lower bounds. Towards that, we show a connection between large SOS representation and hard polynomials; strong enough to imply $\mathsf{VP} \neq \mathsf{VNP}$ and subsequently $\mathsf{PIT} \in \mathsf{SUBEXP}$. This is mainly achieved by an SOS-decomposition result for circuits via Algebraic Branching Programs (ABP) (for definition see Section 2). It expresses any d-degree polynomial $f(\boldsymbol{x})$ of circuit size s as sum of squares of polynomials with degree at most $d/2$. We manage the top-fanin of SOS within a quasi-polynomial blow-up.

[1] The size-bound in the previous such proofs usually counted only #nodes in the circuit, achieving square-root in the bound; while we use #vertices (nodes) + #edges.

[2] One can show that the coefficients are computable in CH, see [Bür09]. To separate VP and VNP, starting from such CH-explicit polynomial families, generally our techniques require GRH (Generalized Riemann Hypothesis).

Finally, we apply a careful *multi-linearization* trick to convert the hardness from the univariate SOS-model to general circuits.

We say that an n-variate polynomial $f(\boldsymbol{x}) \in R[\boldsymbol{x}]$ over a ring R is computed as a *sum-of-squares* (SOS) [3] if

$$f \;=\; \sum_{i=1}^{s} c_i f_i^{\,2}\,, \tag{1}$$

for some *top-fanin* s, where $f_i(\boldsymbol{x}) \in R[\boldsymbol{x}]$ and $c_i \in R$.

Definition 2 (Support-sum size $S_R(f)$)**.** *The* size *of the representation of f in (1) is the* support-sum, *the sum of the support size (or sparsity) of the polynomials f_i. The* support-sum size *of f, denoted by $S_R(f)$, is defined as the minimum support-sum of f.*

If we consider the expression in (1) as a $\sum\bigwedge^2\sum\prod$-formula, then the support-sum is the number of $\prod$-operations directly above the input level.

For any N-variate polynomial f of degree d. Let $|f|_0$ denote the sparsity of f. For any field $R = \mathbb{F}$ of characteristic $\neq 2$, we have

$$|f|_0^{1/2} \;\leq\; S_{\mathbb{F}}(f) \;\leq\; 2\,|f|_0 + 2\,. \tag{2}$$

The lower bound can be shown by counting monomials. The upper bound is because

$$f \;=\; (f+1)^2/4 - (f-1)^2/4\,. \tag{3}$$

In particular, the SOS-model is *complete* for any field of characteristic $\neq 2$. It can be argued by a geometric-dimension argument that for most N-variate (constant $N \geq 1$) polynomials f of degree d, we have $S_{\mathbb{F}}(f(\boldsymbol{x})) \;=\; \Theta(d^N)$, as for random f, $|f|_0 \;=\; \Theta(d^N)$.

We want to explore how $S_{\mathbb{F}}(f_d)$ behaves w.r.t. d, for *explicit* families $(f_d)_d$, that is, the coefficient-function of the family is computable in time $\mathrm{poly}(\log d)$. We call a polynomial family SOS-hard, if its support-sum is just *slightly* larger than the trivial lower bound from (2).

Definition 3 (SOS-hardness)**.** *For constant $N \geq 1$, an explicit N-variate polynomial family $(f_d(\boldsymbol{x}))_d$ is* SOS-hard, *if $S_{\mathbb{F}}(f_d) = \Omega(d^{N(0.5+\varepsilon)})$, where $\varepsilon := \varepsilon(d) = \omega\left(\sqrt{\frac{\log\log d}{\log d}}\right)$ is a sub-constant function.*

Remarks.

1. For our purpose we could relax the explicitness condition such that the j-th bit of $\mathrm{coef}_{x^i}(f_d)$ is computable in $\mathrm{poly}(2^{1/\varepsilon})$ time. This makes the family *barely explicit* w.r.t. d. In fact, $\#P/\mathrm{poly}$ w.r.t. $2^{1/\varepsilon}$ works too. Eg. $f_d = \sum_{i\in[d]} 2^{i^2} x^i$ is an easy candidate for $N = 1$.
2. $\Omega(d^{N(0.5+\varepsilon)})$, instead of $\Omega(d^N)$, which is the expected bound for most f_d, is a much weaker requirement. In fact, the trivial lower bound is $S(f_d) \geq \Omega(d^{N/2})$. Thus, we demand just a tiny improvement over the trivial bound (namely, by a factor of $d^{N\varepsilon} = d^{o(1)}$) [4].

1.3 Our results for SOS

Algebraic circuits are quite well-structured, for instance, there is a famous depth-$O(\log d)$ reduction result [VSBR83, SY10, Sap19]. Its proof methods implicitly establish (for a proof sketch, see Lemma 16) that an n-variate, degree d polynomial $f(\boldsymbol{x})$, computed by a circuit of size s, can be rewritten as

$$f(\boldsymbol{x}) \;=\; \sum_{i=1}^{O(sd^2)} c_i f_i(\boldsymbol{x})^2\,, \tag{4}$$

for $c_i \in \mathbb{F}$ and $f_i \in \mathbb{F}[\boldsymbol{x}]$, where each f_i has circuit size at most $O(sd^2)$ and $\deg(f_i) \leq 2d/3$, for all i. Moreover, with a larger, *quasi*-polynomial blowup in the top-fanin, we bring down the degree really to $d/2$ (via Algebraic Branching Programs (ABP)); for the details, see Section 3.1.

[3] In the real analytic sense, this is a *weighted* SOS. However, over complex field $\mathbb{C}$ one can include the constants inside f_i. In this work one could also restrict to $c_i = \pm 1$; so any field $\mathbb{F}$ with $\sqrt{-1}$ gives exact SOS.

[4] For eg. $(\log d)^{\sqrt{\log d}}$ is such a function that works in d^{ε}.

Main Lemma (SOS Decomposition). *Let $\mathbb{F}$ be a field of characteristic $\neq 2$. Let $f(\boldsymbol{x})$ be an n-variate polynomial over $\mathbb{F}$ of degree d, computed by a circuit of size s. Then there exist $f_i \in \mathbb{F}[\boldsymbol{x}]$ and $c_i \in \mathbb{F}$ such that $f(\boldsymbol{x}) = \sum_{i=1}^{s'} c_i f_i(\boldsymbol{x})^2$, for $s' \leq (sd)^{O(\log d)}$ and $deg(f_i) \leq \lceil d/2 \rceil$, for all $i \in [s']$.*

The leitmotif of this paper is the interplay between SOS-hardness and derandomization/ hardness questions in algebraic complexity. Could a barely explicit and mildly hard polynomial in the SOS-model settle the VP vs. VNP question? We evince a positive answer.

Theorem 1 (Circuit hardness). *If there exists an SOS-hard polynomial family then* VP $\neq$ VNP.

Remarks.

1. Our proof-method from constant-N-variate SOS-hardness to VP $\neq$ VNP is essentially the same as the one for $N = 1$ (eg. replace d by d^N). So, for simplicity of exposition, from now on we will focus on univariate SOS-hardness.
2. In the *non-commutative* setting, lower bound on sum-of-squares (of multivariates) implies that Permanent is hard [HWY11]. Our theorem can be seen as its natural analog in the commutative setting; where potential cancellations could give smaller representations.
3. Another simple candidate for SOS-hardness is $f_d = (x+1)^d$ (though, by repeated squaring, it has circuit size $\Theta(\log d)$). However, its coefficients are not poly$\log(d)$-time explicit. Nevertheless, from its CH-explicitness, and GRH, the theorem does hold [5].
4. In the theorem and Equation (1), we could restrict the degrees of f_i to be $O(d\varepsilon \log d) = d \cdot o(\log d)$ and the top-fanin $s = d^{o(\varepsilon)} = d^{o(1)}$. (Also, Corollary 4 works with analogously *weaker* ε.) This might help in constructing polynomials with a weaker SOS-hardness notion. See Section 3.2 for more details.
5. A stronger SOS-hardness notion with *constant* ε, gives an *exponential* separation between VP and VNP. This proof has many technical differences; refer to Theorem 19 for the details.

Hardness of general circuits often leads to nontrivial *derandomization* [NW94, KI04, AGS19, GKSS19]. Our methods in Theorem 1 consequently put blackbox-PIT in SUBEXP [KI04, Thm. 7.7]. In fact, if ε is a constant, then it puts blackbox-PIT $\in$ QP (*Quasi*-polynomial-time) (Theorem 19).

1.4 Sum-of-cubes model (SOC)

We show that a strong lower bound in the sum-of-cubes model leads to a *complete* derandomization of blackbox-PIT. We say that an n-variate polynomial $f(\boldsymbol{x}) \in R[\boldsymbol{x}]$ over a ring R is computed as a *sum-of-cubes* (SOC), if

$$f = \sum_{i=1}^{s} c_i f_i^3 \,, \tag{5}$$

for some top-fanin s, where $f_i(\boldsymbol{x}) \in R[\boldsymbol{x}]$ and $c_i \in R$.

Definition 4 (Support-union size $U_R(f,s)$). *The* size *of the representation of f in (5) is the size of the* support-union, *namely the number of distinct monomials in the representation, $\left|\bigcup_{i=1}^{s} \text{supp}(f_i)\right|$, where* support supp$(f_i)$ *denotes the set of monomials with a nonzero coefficient in the polynomial $f_i(\boldsymbol{x})$. The* support-union size *of f with respect to s, denoted $U_R(f,s)$, is defined as the minimum support-union size when f is written as in (5).*

If we consider the expression in (5) as a $\sum\bigwedge^3\sum\prod$-circuit, then the support-union size is the number of $\prod$-operations directly above the input level (unlike $\sum\bigwedge^2\sum\prod$-*formula* in Defn. (2)).

The two measures– support-union and support-sum –are largely incomparable, since $U(\cdot)$ has the extra argument s. Still one can show: $S_{\mathbb{F}}(f) \geq \min_s (U_{\mathbb{F}}(f, 4s) - 1)$ (Lemma 14).

[5] Similarly, for the polynomial family, $f_d(x) = \prod_{i\in[d]}(x-i)$. Or, $f_d(x) = \sum_{0\leq i\leq d} x^i/i!$ and Chebyshev polynomials.

For any polynomial f of sparsity $|f|_0$, we have

$$|f|_0^{1/3} \;\le\; U_{\mathbb{F}}(f,s) \;\le\; |f|_0 + 1, \tag{6}$$

where the upper bound is for $s \ge 3$ and for fields $R = \mathbb{F}$ of characteristic $\neq 2,3$. The lower bound can be shown by counting monomials. The upper bound is because

$$f \;=\; (f+2)^3/24 + (f-2)^3/24 - f^3/12\,. \tag{7}$$

Hence, the SOC-model is *complete* for any field of characteristic $\neq 2,3$.[6]

For simplicity, fix #variables $N = 1$. Here are two more examples (that we know of) for the trade-off between s and the measure $U_{\mathbb{F}}(f,s)$, for any f.

Examples.

1. For small $s = \Theta(d^{1/2})$, we have $U_{\mathbb{F}}(f,s) = O(d^{1/2})$ (Corollary 10).
2. For large $s = \Omega(d^{2/3})$, we have $U_{\mathbb{F}}(f,s) = \Theta(d^{1/3})$ (Theorem 11).

However, it is unclear whether, over $\mathbb{F} = \mathbb{Q}$, for a very small fanin s, support-union $= o(d)$ exists. This trade-off between the measure U and the top-fanin s in the above examples, motivated us to define hardness in the SOC-model as follows.

Definition 5 (SOC-hardness)**.** *A* $\mathsf{poly}(d)$*-time explicit univariate polynomial family* $(f_d)_d$ *is* SOC-hard*, if there exists a positive constant* $\varepsilon' < 1/2$ *such that* $U_{\mathbb{F}}\left(f_d, d^{\varepsilon'}\right) = \Omega(d)$.

1.5 Our results for SOC

Though technically incomparable, the SOC-hardness feels stronger than SOS-hardness (for $N = 1$); indeed it can be used to prove a connection like Theorem 1. Now, we show an even stronger consequence— a complete derandomization of blackbox-PIT.

Theorem 2 (Derandomization)**.** *If there is an SOC-hard polynomial family then blackbox-PIT* $\in \mathsf{P}$.

Remarks.

1. Older results too lead to various conditional derandomizations. E.g. *multi*-variate hard polynomials lead to blackbox-PIT $\in \mathsf{QP}$ (*quasipoly-time*) [KI04, AGS19]. Recently, [GKSS19] showed that the *circuit* hardness of a constant-variate polynomial family yields blackbox-PIT $\in \mathsf{P}$ (Theorem 20). Our hardness assumption is merely in the SOC-model. In fact, SOC is the *first* restricted model where hardness implies *complete* derandomization.
2. For Theorem 2, we could restrict the degrees of f_i, to be $O(d)$. See Section 3.3, Remark 3.3.

1.6 Basic arguments

There have been a series of works that connect the hardness in restricted univariate (resp. constant-variate) models to VP vs. VNP and the PIT problem. This work is more about remodeling the major questions in the *simplest* format possible. We show how to transfer the hardness of a (univariate) polynomial family in the SOS, resp. SOC-model, to a hard (multivariate) polynomial family in the circuit-model. To do so, we adapt the existing powerful techniques to our setting. Intuitively, one would expect that the analytic nature of SOS and SOC (over $\mathbb{R}$ or $\mathbb{C}$) makes it easier to prove hardness in these models than for general circuits. In any case, we show that this would suffice to solve central questions in algebraic complexity.

The gap between the SOS-model and general circuits is mainly bridged by a decomposition lemma (Main Lemma) which emerges via ABPs. Frontiers based depth-reduction [VSBR83] implicitly shows that any polynomial $f(\boldsymbol{x})$ of degree d, computed by a homogeneous circuit of size s, can be decomposed as $f(\boldsymbol{x}) = \sum_{i=1}^{s} f_{i1} \cdot f_{i2}$, where $\deg(f_{ij}) \le 2d/3$ and $\mathsf{size}(f_{ij}) \le O(s)$; for a proof see Lemma 16. However, such proof strategies can never give intermediate

[6] In this work, we could restrict the defining Eqn.(5) to $c_i = 1$, as long as $3^{-1/3} \in \mathbb{F}$ (owing to Eqn.(9)).

polynomials of degree *exactly* $d/2$, simply because degree $\approx d/2$ polynomial may not even *exist* in the computation tree, and thus, frontiers at appropriate layers do not really help. However, in the case of *homogeneous* ABPs, the intermediate degrees increase gradually, as the labels are *linear* forms. In particular, a layer of vertices computing degree *exactly* $d/2$ exists. By cutting the ABP, say, of width w, at the $d/2$-th layer, we get $f = (f_1, \ldots, f_w)^T \cdot (f'_1, \ldots, f'_w) = \sum_{i=1}^{w} f_i \cdot f'_i$. This directly gives an SOS-form of top-fanin at most $2w$. The conversion from a homogeneous circuit to a homogeneous ABP is pretty straight-forward in the literature. Use log-depth-reduction [VSBR83] and induct on the depth to conclude that $s^{O(\log d)}$-size ABP exists. Finally, homogenize the ABP with a polynomial blowup in size. (See [Kum19, Lem.15] or [Sap19].)

The main idea in Theorem 1 is to lift the hardness of $f = f_d$ in the SOS-model to a multivariate polynomial, which we prove to be super-polynomially hard in the general circuit model (implying $\notin$ VP) and explicit (implying $\in$ VNP). Usually, to convert a univariate polynomial to multivariate, (inverse) Kronecker type substitution is used; here we *do not* use the Kronecker due to a technical barrier and the reason will be addressed in the next paragraph. Instead, we use a *multilinear* map ϕ that sends x^i to $\phi(x^i) := \prod_{j\in[n],\, \ell\in[0\ldots k-1]} y_{j,\ell}$, where $\ell \cdot k^{j-1}$ contributes to the $\mathrm{base}_k(i)$-*representation* in the j-th position; n and k are both functions of d to be fixed. Consider, by linear extension, $\phi(f) =: P_{n,k}$. By construction $P_{n,k}$ is a kn variate n degree multilinear polynomial. With appropriate parameter fixing, we show that $\mathrm{size}(P_{n,k}) = (kn)^{\omega(1)}$. The proof goes via contradiction. If the size is smaller, then using Main Lemma, we get $P_{n,k}$ as sum of $d^{o(\varepsilon)}$-many Q_i^2's; where the intermediate polynomial Q_i (kn-variate) has degree at most $n/2$. Thus, a naive upper bound on the support-sum (after proper parameter fixing) is $d^{o(\varepsilon)} \cdot \binom{kn+n/2}{n/2} < d^{o(\varepsilon)} \cdot d^{1/2+\varepsilon/2} = o(d^{1/2+\varepsilon})$, a contradiction to the SOS-hardness!

Here we remark that Kronecker type substitution *does not* give the desired result. It basically maps a monomial x^e to $\boldsymbol{x}^{\boldsymbol{e}}$, where $\boldsymbol{e} := \mathrm{base}_{(n+1)}(e)$ for some n; then n is the individual-degree in the image, and $(n+1)^k \geq d+1 > n^k$. However, this map converts f to be a k-variate, individual-degree n polynomial family and the naive binomial upper-bound on the number of terms would be $\binom{k+kn/2}{k} > (n+1)^k > d$; which is useless. (Here we use $kn/2$ as the degree of $P_{n,k}$ is kn while the degree of the intermediate polynomial halves.) Thus, the multi-linearization trick, along with the SOS decomposition lemma via ABPs, are indispensable in our proof.

The proof of Theorem 2 works very differently than that of Theorem 1. As, its goal is to devise an amply hard polynomial with a *constant* number of variables only; it limits our tricks quite a bit.[7] It uses (inverse) Kronecker map to construct $P_{n,k}$ from $f = f_d$, a constant-k-variate, individual-degree n polynomial. We show this polynomial to be $s = n^{\Omega(1)}$ hard. Recall that an explicit constant-variate *circuit*-hard polynomial can be used as an efficient hitting-set generator; showing blackbox-PIT $\in$ P [GKSS19]. The hardness result organically comes from a *SOC decomposition lemma* (Lemma 5); using a 'constant-boosting' of frontier-based Lemma 16 and a 'greedy clustering'. Basically, we show that any homogeneous polynomial $P(\boldsymbol{x})$ of degree d, computed by a homogeneous circuit of size s', can be written as $P(\boldsymbol{x}) = \sum_{i=1}^{\mathsf{poly}(s')} c_i \cdot Q_i(\boldsymbol{x})^3$, where $\deg(Q_i) \leq 4d/11$.[8] Applying this to each homogeneous part of $P_{n,k}$, and then Kronecker substitution would show (with proper parameter fixing) that $U_{\mathbb{F}}(f) \leq |\bigcup_{i=1}^{\mathsf{poly}(s,n)} \mathrm{supp}(Q_i)| \leq \binom{k+4kn/11}{k} < c \cdot d$, for *any* positive constant c. We use Eqn.(8) to bound the binomial and reach a contradiction. The constant $4/11$ is nothing special; any constant in $(1/3, 1/e)$ would work.

[7] Eg. the failure analysis above with $\binom{k+kn/2}{k}$ is also partly the reason why SOS can't give complete PIT.

[8] We cannot use such a decomposition lemma using ABPs, as the *super*-polynomial blowup in the fanin, owing to the larger degree ($\approx d^{1/k}$), would fail to prove the desired circuit-hardness of the resulting polynomial family.

1.7 Further comparison with prior related works

Technically, SOS-hardness is incomparable to earlier notions involving uni/multi-variate polynomials. Although the proofs are mostly combinations and careful application of existing methods, the connection to the SOS-model is new and perhaps unexpected. A handful of works do lift the lower bound results in very restricted models (of uni/multi-variate polynomials) to circuit generality. In this subsection, we compare our models to previous work in more detail. We focus on the SOS-model, but similar things can be said for SOC.

Former depth-4 circuits vs. SOS-model. Almost all of the previous works are concerned with the sum-of unbounded-powers $\sum \bigwedge^{\omega(1)} \sum \prod$ as they use the standard depth-reduction results [AV08, Koi12, GKKS13, AGS19]. Sufficiency of proving lower bound on restricted models of *uni*variate polynomials came when Koiran [Koi11] showed: if there exists an explicit univariate polynomial $f(x)$ of degree d s.t. any $\sum \bigwedge^{\omega(1)} \sum \bigwedge$ representation of the form $f(x) = \sum_{i=1}^{s} c_i \cdot Q_i^{e_i}$, where sparsity$(Q_i) \leq t$ with unbounded exponents e_i, requires top-fanin $s \geq (d/t)^{\Omega(1)}$, then VP $\neq$ VNP. In our work, the demand is different (or incomparable) mainly in two places– (1) the model is simpler, it is merely $\sum^{s_1} \bigwedge^2 \sum^{s_2} \bigwedge$; (2) we want to lower-bound the support-sum $\approx s_1 \cdot s_2$, which is *neither* the "size" of the univariate depth-4 circuit *nor* the "top-fanin".

τ-conjecture vs. SOS-hardness. The τ-conjecture was first introduced by Shub and Smale [SS95, Sma98], and then was modified by Koiran [Koi11] for *real* roots. The original Shub-Smale τ-conjecture is about the number of *distinct integer roots*, and thus $(x+1)^d$ is not a good candidate *unlike* for SOS-hardness (with $N = 1$). Koiran considered the real roots of a polynomial computed by a restricted $\sum^k \prod^m \sum^t \bigwedge$-circuit and conjectured that the number of real roots is at most poly(kmt). Their work and [KPT15] suggest that it is important to exclude multiplicities. In that sense, $(x+1)^d$ was still not a good candidate. However, Hrubeš generalized the conjecture; by proving that $(x+1)^d \in \sum^k \prod^m \sum^t \bigwedge$ necessarily implies that kmt is *large*, assuming the truth of Koiran's τ-conjecture [Hru13, Obs.3.4].

The other significant differences are the same as pointed out in the case of depth-4 circuits above, since m is unbounded in earlier works. Additionally, SOS-hardness (with $N = 1$) goes a step further and minimizes the desired $\varepsilon(d)$ to be vanishingly close to 0.

τ-conjecture for Newton Polygons vs. SOS-hardness. Koiran et al. [KPTT15] generalized the τ-conjecture to *bi*variate polynomials, where the measure now is #edges in its Newton polygon. The model of interest was still the same, $\sum^k \prod^m \sum^t \bigwedge$, with $m = \omega(1)$. The differences to SOS-hardness (with $N = 2$) are the same as pointed in the case of depth-4 circuits above, since m is unbounded. Also, these conjectures seem to be independent of SOS-hardness in the sense that we know of no implication either way.

Existence of $(r, 2)$-elusive function vs. SOS-hardness. Raz [Raz10] formalized a notion of elusive maps and established a connection between the existence of explicit elusive maps and VP vs. VNP. A polynomial map $L : \mathbb{F}^n \to \mathbb{F}^m$ is $(r, 2)$-*elusive* if for every degree-2 polynomial mapping $M : \mathbb{F}^r \to \mathbb{F}^m$, Image$(L) \not\subset$ Image(M). Formally, he showed that any explicit polynomial map which is $(r, 2)$-elusive with $m = n^{\omega(1)}$ and $r = n^{0.9}$ implies VP $\neq$ VNP. Observe that one can reinterpret the coefficients of the f_i^2's in Eqn.(1) as expressing coef(f) via quadratic forms (like M). However, the old elusive formulation is *too* general in the sense: the parameters r vs. m have a superpoly-large gap (and still M has to elude all L). On the other hand, SOS-hardness, say for $N = 1$, goes a step further and optimizes the gap to be vanishingly close to *square*. Further, SOS gives a rather specialized degree-2 polynomial mapping.

Other related works. We already mentioned the work of Koiran [Koi11], where he showed that proving $s \geq (d/t)^{\Omega(1)}$ for $f \in \sum^s \bigwedge^{\omega(1)} \sum^t \bigwedge$, suffices to show VP $\neq$ VNP. In the case of $\deg(Q_i) \leq t$, an unconditional lower bound of $s \geq \Omega(\sqrt{d/t})$ is indeed known [KKPS15]. For

$\deg(Q_i) \leq 1$, the bound $s \geq \Omega(d)$ has been established for certain polynomials; using the concept of *Birkhoff Interpolation* [GMK17, KPGM18].

In [Koi12], ideas of conversions between circuits (depth-4 circuits with unbounded fanin) and ABPs were used. We also remark that in the monotone circuit regime, the SOS decomposition type lemma has been known and used to show *exponential* hard polynomials [RY11]. Recently, a strong separation between Monotone VP and Monotone VNP has been settled [Yeh19, Sri19].

The number-theory version of SOS (resp. SOC) is related to the classical *Waring problem*. It asks for a number k whether there exists a number $g(k)$ such that every natural number can be written as the sum of $g(k)$-many k-th powers of numbers. Some celebrated examples are $g(2) = 4$ [Dix64] and $g(3) = 9$ [Kem12]. Later, many variants of Waring's problem for polynomials have been studied using real/complex analytic tools [FOS12, CCG12, BT15].

2 Preliminaries

Basic notation. Denote the underlying field as $\mathbb{F}$ and assume that it is $\mathbb{Q}, \mathbb{Q}_p$, or their fixed extensions. Our results hold also for finite fields of large characteristic.

Let $[n] = \{1, \ldots, n\}$. For $i \in \mathbb{N}$ and $b \geq 2$, we denote by $\text{base}_b(i)$ the unique k-tuple $(i_1, \ldots, i_k)$ such that $i =: \sum_{j=1}^{k} i_j \cdot b^{j-1}$.

For binomial coefficients, we use an easy bound based on the e^k-series [Wik], for $1 \leq k \leq n$,

$$\left(\frac{n}{k}\right)^k \leq \binom{n}{k} \leq \left(\frac{en}{k}\right)^k. \tag{8}$$

Polynomials. For $p \in \mathbb{F}[\boldsymbol{x}]$, where $\boldsymbol{x} = (x_1, \ldots, x_m)$, for some $m \geq 1$, the *support of* p, denoted by $\text{supp}(p)$, is the set of nonzero monomials in p. *Sparsity* or *support size of* p is $|p|_0 := |\text{supp}(p)|$. By $\text{coef}(p)$ we denote the *coefficient vector* of p (in some fixed order). For polynomials $p_1, \ldots, p_s \in \mathbb{F}[\boldsymbol{x}]$, their *span* is the vector space $\text{span}_{\mathbb{F}}(p_1, \ldots, p_s) := \{ \sum_i c_i p_i \mid c_i \in \mathbb{F} \}$.

For an exponent vector $\boldsymbol{e} = (e_1, \ldots, e_k)$, we use $\boldsymbol{x}^{\boldsymbol{e}}$ to denote the monomial $x_1^{e_1} \ldots x_k^{e_k}$.

Algebraic circuits. An *algebraic circuit* over a field $\mathbb{F}$ is a layered directed acyclic graph that uses field operations $\{+, \times\}$ and computes a polynomial. It can be thought of as an algebraic analog of boolean circuits. The leaf nodes are labeled with the input variables $x_1, \ldots, x_n$ and constants from $\mathbb{F}$. Other nodes are labeled as addition and multiplication *gates*. The root node outputs the polynomial computed by the circuit.

Complexity parameters of a circuit are: **1)** the *size*, i.e. number of edges and nodes, **2)** the *depth*, i.e. number of layers, **3)** the *fan-in*, i.e. maximum number of inputs to a node, (resp. the *fan-out*, i.e. maximum number of outputs of a node).

When the graph is in fact a tree, i.e., the fan-out is 1, we call the circuit an *algebraic formula*.

For a polynomial f, the size of the smallest circuit computing f is denoted by $\text{size}(f)$, it is the *algebraic circuit complexity* of f. By $\mathcal{C}(n, D, s)$, we denote the set of circuits C that compute n-variate polynomials of degree D such that $\text{size}(C) \leq s$.

In *complexity classes*, we specify an upper bound on these parameters. Valiant's class VP contains the families of n-variate polynomials of degree $\text{poly}(n)$ over $\mathbb{F}$, computed by circuits of $\text{poly}(n)$-size. The class VNP can be seen as a non-deterministic analog of the class VP. A family of n-variate polynomials $(f_n)_n$ over $\mathbb{F}$ is in VNP if there exists a family of polynomials $(g_n)_n$ in VP such that for every $\boldsymbol{x} = (x_1, \ldots, x_n)$ one can write $f_n(\boldsymbol{x}) = \sum_{w \in \{0,1\}^{t(n)}} g_n(\boldsymbol{x}, w)$, for some polynomial $t(n)$ which is called the *witness size*. It is straightforward to see that VP $\subseteq$ VNP and *conjectured* to be different (Valiant's Hypothesis [Val79]). For more details see [Mah14, SY10, BCS13]. Unless specified particularly, we consider the field $\mathbb{F} = \mathbb{Q}$ (resp. a finite field with large characteristic).

Valiant [Val79] showed a *sufficient* condition for a polynomial family $(f_n(\boldsymbol{x}))_n$ to be in VNP. We use a slightly modified version of the criterion, for a proof see Appendix D.

Theorem 3 (Valiant's VNP criterion, [Val79]). *Let function $\phi_n : \{0,1\}^n \to \{0,1\}^{2^n}$ be such that each bit of $\phi_n(\cdot)$, is computable in* #P/poly. *Then, the family of polynomials defined by $f_n(\boldsymbol{x}) := \sum_{\boldsymbol{e}\in\{0,1\}^n} \phi_n(\boldsymbol{e}) \cdot \boldsymbol{x}^{\boldsymbol{e}}$, is in* VNP.

Algebraic branching programs (ABP). An *algebraic branching program* (*ABP*) in variables $\boldsymbol{x}$ over a field $\mathbb{F}$ is a directed acyclic graph with a *starting vertex* s with in-degree zero, an *end vertex* t with out-degree zero. The edge between any two vertices is labeled by affine form $a_1x_1 + \ldots + a_nx_n + c \in \mathbb{F}[\boldsymbol{x}]$, where $a_i, c \in \mathbb{F}$.

The *weight of a path* in an ABP is the product of labels of the edges in the path. The *polynomial computed at a vertex* v is the sum of weights of all paths from the starting vertex s to v. The *polynomial computed by the ABP* is the polynomial computed at the end vertex t.

The polynomial computed by an ABP can be written as a matrix product $U^T(\prod_i M_i)V$, where $U, V \in \mathbb{F}^{w\times 1}$ and $M_i \in \mathbb{F}[\boldsymbol{x}]^{w\times w}$ with entries being affine linear forms. The parameter w is called the *width* of the ABP. The class VBP contains the families of polynomials computed by ABPs of size $\mathsf{poly}(n)$. This implies that the degree is $\mathsf{poly}(n)$ too.

An ABP is a very restricted circuit, but still being able to compute determinants [MV99].

We say that the ABP is *homogeneous*, if the polynomial computed at every vertex is a homogeneous polynomial. It is known that for an ABP S of size s computing a homogeneous polynomial f, there is an equivalent homogeneous ABP A' of size $\mathsf{poly}(s)$, where each edge-label is a *linear form* $a_1x_1 + \cdots + a_nx_n$. Moreover, when f has degree D, then A' has $D+1$ layers and each vertex in the i-th layer computes a homogeneous polynomial of degree $= i$ (see [Kum19, Lem.15] or [Sap19]).

Here, we also remark that each homogeneous part of a degree d polynomial $f(\boldsymbol{x})$, computed by s-size circuit, can also be computed by a *homogeneous* circuit of size $O(sd^2)$; see [SY10, Sap19].

3 Proof of the main results

3.1 SOS decomposition of circuits: Proof of Main Lemma

Proof of Main Lemma. Let C be a circuit of size s computing $f(\boldsymbol{x})$. W.l.o.g., $f(\boldsymbol{x})$ is a homogeneous polynomial (as later we will apply to every homogeneous component of f). Using the log-depth reduction of [VSBR83], there is a homogeneous circuit C' of depth $\log d$ and size $\mathsf{poly}(s)$ that computes F.

Now we convert the circuit C' to a *layered* ABP A as follows: first, convert the circuit C' to a formula F. By induction on the depth of the circuit one can show that F has size $s^{O(\log d)}$. Secondly, we convert F to an ABP A. It is well known that for any formula of size t, there exists an ABP of size at most $t+1$, computing the same polynomial, for details see [Sau12, Lemma 2.14]. Thus, the ABP A computing f has size at most $s^{O(\log d)}$.

Further, we *homogenize* the ABP A as explained at the end of the preliminary section. Let A' be the homogenized ABP computing f. Its size is $s' := \mathsf{poly}(s^{O(\log d)}) = s^{O(\log d)}$.

Finally, cut ABP A' in half, at the $\lceil d/2 \rceil$-th layer, to get: $f = (f_1, \ldots, f_{s'})^T \cdot (f'_1, \ldots, f'_{s'}) = \sum_{i=1}^{s'} f_i \cdot f'_i$, where, degree of each f_i, f'_i is at most $\lceil d/2 \rceil$. This can be easily rewritten as SOS by Equation (3). The top-fanin of SOS is at most $2s'$.

For a non-homogeneous polynomial $f(\boldsymbol{x})$, we can apply the above for each homogeneous part of $f(\boldsymbol{x})$. It is well known that each homogeneous part can be computed by a homogeneous circuit of size $O(sd^2)$. Thus, for non-homogeneous polynomials, s can be replaced by $O(sd^2)$; hence the top-fanin of SOS is $(sd^2)^{O(\log d)} = (sd)^{O(\log d)}$. □

3.2 SOS-hardness to VP $\neq$ VNP: Proof of Theorem 1

Proof of Theorem 1. We will construct an explicit (multivariate) polynomial family, using SOS-hard univariate f_d, which is not in VP, but is in VNP. This would imply that VP $\neq$ VNP.

Construction: We will construct $(P_{n,k})_k$ from f_d, where $P_{n,k}$ is a multilinear degree-n and kn-variate polynomial, for $n = n(d)$ and $k = k(d)$[9]. We will specify k and n in the course of the proof. The basic relation between d, n and k is that $k^n \geq d+1 > (k-1)^n$. Introduce kn many new variables $y_{j,\ell}$, where $1 \leq j \leq n$ and $0 \leq \ell \leq k-1$. Let $\phi_{n,k}$ be the map,

$$\phi_{n,k} \,:\, x^i \mapsto \prod_{j=1}^{n} y_{j,i_j} \;\text{, where } i =: \sum_{j=1}^{n} i_j \cdot k^{j-1},\;\; 0 \leq i_j \leq k-1\,.$$

Note: for $i \in [0,d]$, $\phi_{n,k}$ maps x^i uniquely to a multilinear monomial of degree n. By linear extension, define $\phi_{n,k}(f_d) =: P_{n,k}$. By construction, $P_{n,k}$ is n-degree, kn-variate multilinear polynomial. Let $\psi_{n,k}$ be the homomorphism that maps any n-degree multilinear monomial, defined on variables $y_{j,\ell}$, such that $y_{j,\ell} \mapsto x^{\ell \cdot k^{j-1}}$. Observe that, $\psi_{n,k} \circ \phi_{n,k}(f) = f$, for any degree $\leq d$ polynomial $f \in \mathbb{F}[x]$.

SOS-hardness $\implies$ hardness of $P_{n,k}$: Assume that family (f_d) is SOS-hard with parameter ε. We will show that $\text{size}(P_{n,k}) \geq d^{\mu(d)} = (kn)^{\omega(1)}$ for some function μ depending on $\varepsilon(d)$. We have $\varepsilon > \omega(\sqrt{\log\log d / \log d})$ and w.l.o.g. $\varepsilon < (\log\log d/\log d)^{1/3}$, for large d (Note: Proving for a small ε suffices; also $1/3$ is nothing special, any constant $< 1/2$ in the exponent works.).

Suppose, $\text{size}(P_{n,k}) \leq d^{\mu}$, for some $\mu(d)$. Then, from Main Lemma, we know that $\exists Q_i$'s such that $P_{n,k} = \sum_{i=1}^{s} c_i \cdot Q_i^2$, where $s \leq (d^{\mu} \cdot n)^{c \log n}$, for some constant c, with $\deg(Q_i) \leq \lceil n/2 \rceil$. Note: $f_d = \psi_{n,k} \circ \phi_{n,k}(f_d) = \sum_{i=1}^{s} c_i \cdot \psi_{n,k}(Q_i)^2$. As $\psi_{n,k}$ cannot increase the sparsity, $|\psi_{n,k}(Q_i)|_0 \leq |Q_i|_0 \leq \binom{kn+\lceil n/2 \rceil}{\lceil n/2 \rceil}$, [10] for each $i \in [s]$. Thus, by definition $S_{\mathbb{F}}(f_d) \leq s \cdot \binom{kn+\lceil n/2 \rceil}{\lceil n/2 \rceil}$. The idea is to fix parameters so that $S(f_d) < o(d^{1/2+\varepsilon})$. We will fix μ such that

1. $s \leq d^{\delta_1}$ for some function δ_1,
2. $\binom{kn+\lceil n/2 \rceil}{\lceil n/2 \rceil} \leq d^{\delta_2}$ for some function δ_2,
3. $d^{\delta_1+\delta_2} < o(d^{1/2+\varepsilon})$,
4. $d^{\mu} > (kn)^{\omega(1)}$.

Note: from conditions 1-3, if $\text{size}(P_{n,k}) \leq d^{\mu}$ then $S(f_d) < o(d^{1/2+\varepsilon})$, contradicting the SOS-hardness. Thus, condition 4 would give super-polynomial hardness result.

Parameter fixing: Let $\mu := 1/\sqrt{\log d \cdot \log\log d}$ and $\delta_1 := c' \cdot \mu \cdot \log n$ for some $c' > c$. Let $\delta_2 := 1/2 + \varepsilon/2$. Fix $k := \lceil 6^{1/\varepsilon} + 1 \rceil$. This fixing of k together with $k^n \geq d+1 > (k-1)^n$ implies that $n = \Theta(\varepsilon \cdot \log d)$. We also assume n to be even for simplicity, to avoid the ceiling function.

Bound on the binomial: Note that, it is enough to have the following chain of inequalities:

$$\binom{kn+n/2}{n/2} \leq (e+2ek)^{n/2} \leq (6(k-1))^{n/2} \leq (k-1)^{n\delta_2} \leq d^{\delta_2}\,.$$

First inequality is by Eqn.(8); the second one is by the fact that $2e < 6$, thus for large enough k, it holds; and the last inequality follows by the assumption that $d \geq (k-1)^n$. For the third one, it suffices to ensure that $(k-1)^{\delta_2 - 1/2} \geq \sqrt{6}$. This is where we used the fact that $\delta_2 - 1/2 = \varepsilon/2 > 0$ and thus it is enough to fix $k - 1 = \lceil 6^{1/\varepsilon} \rceil$.

Bound on top-fanin s: Note that $s \leq (d^{\mu} \cdot n)^{c \log n}$ from Main Lemma for some constant c. We want $d^{c' \cdot \mu \cdot \log n} = d^{\delta_1} \geq (d^{\mu} \cdot n)^{c \log n}$. It suffices to show that $d^{(c'-c)\cdot\mu} \geq n^c$. It is fairly

[9] In this section think of n as a *tiny* function of k. Thus indexing the family over k suffices.

[10] Any n variate degree d polynomial can have sparsity at most $\binom{n+d}{d}$.

straightforward to verify that with our parameters fixing of $\mu \log d = \sqrt{\log d / \log\log d}$, and $\log n \leq O(\log\log d)$, the above inequality holds for large enough d.

Checking $d^{\delta_1+\delta_2} = o(d^{1/2+\varepsilon})$: Note that, $\log n = O(\log\log d)$ and thus $\delta_1 = O(\sqrt{\log\log d / \log d})$ $= o(\varepsilon)$. Hence, $\delta_1 + \delta_2 = o(\varepsilon) + 1/2 + \varepsilon/2 < 1/2 + \varepsilon$; since $d^\varepsilon \to \infty$ as $d \to \infty$, the conclusion follows.

Checking $d^\mu = (kn)^{\omega(1)}$: Note that, $d^\mu = (kn)^{\omega(1)} \iff \mu = \omega(1) \cdot \log(kn)/\log d \iff \mu \cdot \log d = \omega(\log(kn))$. It is clear that, $\log(kn) = \log k + \log n \leq O(1/\varepsilon)$ for large enough n (or equivalently d), as $\log n = O(\log\log d) = o(1/\varepsilon)$ and $\log k = \log\lceil 6^{1/\varepsilon} + 1\rceil = O(1/\varepsilon)$.

Also, note that $\mu \cdot \log d = \sqrt{\log d / \log\log d} = \omega(1/\varepsilon) = \omega(\log(kn))$.

Finally, all the conditions 1-4 are met with the appropriate fixing of parameters as shown above. Thus, we deduce $\text{size}(P_{n,k}) \geq d^\mu = (kn)^{\omega(1)}$, i.e. $P_{n,k}$ requires *super*-polynomial size circuit. Therefore, $(P_{n,k})_k \notin \mathsf{VP}$.

Explicitness: We will show that $P_{n,k}$ is explicit, i.e. $(P_{n,k})_k \in \mathsf{VNP}$. By construction, $P_{n,k}$ is a kn variate, individual degree n multilinear polynomial, so we can write it as

$$P_{n,k} = \sum_{\boldsymbol{e} \in \{0,1\}^{kn}} \phi(\boldsymbol{e}) \cdot \boldsymbol{y}^{\boldsymbol{e}} \,.$$

Here $\boldsymbol{y}$ denotes the kn variables $y_{j,\ell}$ where $1 \leq j \leq n$ and $0 \leq \ell \leq k-1$ and $\boldsymbol{e}$ denotes the exponent-vector. As each x^e in $\text{supp}(f_d)$ maps to a monomial $\boldsymbol{y}^{\boldsymbol{e}}$ uniquely; given $\boldsymbol{e}$, one can easily compute $e := \sum_{j=1}^{n} e_j \cdot k^{j-1}$ and thus $\phi(\boldsymbol{e}) = \text{coef}_{x^e}(f_d)$. By the explicitness hypothesis, any bit of $\phi(\boldsymbol{e})$ is computable in $\mathsf{poly}(\log d) < \mathsf{poly}(2^{1/\varepsilon}) = \mathsf{poly}(kn)$ time. Using Theorem 3, it is clear that $(P_{n,k})_k \in \mathsf{VNP}$, by a wide margin.

So, $(P_{n,k})_k \in \mathsf{VNP}$ and SOS-hardness imply $(P_{n,k})_k \notin \mathsf{VP}$. This proves Theorem 1. □

Corollary 4 (Determinant vs Permanent)**.** *SOS-hardness weakened with* $\varepsilon > \omega(1/\sqrt{\log d})$ *(a smaller* ε *than the original) already implies* $\mathsf{VBP} \neq \mathsf{VNP}$.

Proof Sketch. The log-factor in the exponent is avoidable in the Main Lemma, if the initial polynomial is already an ABP of size s (instead of a circuit). In the above proof, we could then fix $\delta_1 := c'\mu$. This would remove the extra '$\log n = \log\log d$' factors from the calculations. □

Remarks.

1. We showed an explicit super-polynomially hard family $(P_{n,k})_k$. The result of [KI04, Theorem 7.7] then implies PIT $\in$ SUBEXP.

2. If the given ε was a constant, say 0.001; then a very different parameters setting ($k = O(1)$ and $n = O(\log d)$) gives a *sub-exponential* hard polynomial family $(P_{n,k})_n$ of size $> 2^{\Omega(\log d/\log\log d)}$. This happens because of the *super-polynomial* blowup in the size while converting a circuit to an ABP in Main Lemma. However, a repeated boosting of [VSBR83] type lemma (Lemma 18) gives a decomposition with intermediate polynomials having degree *close* to $d/2$. Finally this gives a truly *exponential* hard family $(P_{k,n})_n$; for details see Theorem 19. Thus, [KI04] gives PIT $\in$ QP, when ε is a constant.

   Here, we also remark that "halving" the degree with $\log d$ exponent in the top-fanin gives *better* result than "close" to halving because finally the contribution of the exponent is *quite small* in our application (and in fact absent in case of Corollary 4). However, for constant ε, the scenario changes as mentioned above.

3. As $\deg(Q_i) \leq n/2$, we have $\deg(\psi_{n,k}(Q_i)) \leq n/2 \cdot (k-1) \cdot k^{n-1} < n.k^n = O(nd) = o(d \log d)$. Here we used that $k^n/(k-1)^n < (1 + 1/(k-1))^n < e$, for large d. Thus, it is enough to consider the restricted-degree SOS representation and prove the conjecture.

4. One can further restrict (proof requirement-wise) the SOS top-fanin to a mere $d^{\delta_1} = \exp(O(\sqrt{\log d \cdot \log\log d}))$ which is extremely small compared to d (in fact, $d^{\delta_1} = d^{o(\varepsilon)}$).

3.3 SOC-hardness to blackbox-PIT $\in$ P: Proof of Theorem 2

Proof of Theorem 2. The idea is to convert the SOC-hard polynomial $f_d(x)$ to a *constant-k*-variate individual-degree-n polynomial family $(P_{n,k})_n$ which is 'mildly' *hard*. Later, using [GKSS19], we will conclude that blackbox-PIT $\in$ P. The following lemma is the *crucial* ingredient to connect general circuits to an SOC representation.

Lemma 5 (SOC decomposition)**.** *Let $\mathbb{F}$ be a field of characteristic $\neq 2,3$. Let $f(\boldsymbol{x}) \in \mathbb{F}[\boldsymbol{x}]$ be an n-variate, degree-d polynomial, computed by a circuit of size s. Then, there exist polynomials $f_i \in \mathbb{F}[x]$ and $c_i \in \mathbb{F}$ such that $f(\boldsymbol{x}) = \sum_{i=1}^{s'} c_i \cdot {f_i}^3$, for some top-fanin $s' \leq \mathsf{poly}(s,d)$; achieving $deg(f_i) < 4d/11$, for all $i \in [s']$.*

Proof of Lemma 5. We will first show this for homogeneous polynomials, and then apply it to each homogeneous part of a general $f(\boldsymbol{x})$. Assume that, circuit of $f(\boldsymbol{x})$ is homogeneous. Lemma 16 establishes that $f(\boldsymbol{x})$ can be decomposed as $\sum_{i=1}^{s} \tilde{f}_{i1} \cdot \tilde{f}_{i2}$, where $\tilde{f}_{ij}$ has circuits of size $O(s)$ and $\deg(\tilde{f}_{ij}) \leq 2d/3$, with $\deg(\tilde{f}_{i1}) + \deg(\tilde{f}_{i2}) = d$.

Choose a constant m such that $(2/3)^m < 4/11 - 1/3 = 1/33$ ($m := 9$ suffices). Apply Lemma 16 m times, recursively on each successive circuit $\tilde{f}_{ij}$. As m is constant, it is easy to conclude that $f(\boldsymbol{x})$ can be written as

$$f(\boldsymbol{x}) = \sum_{i=1}^{\mathsf{poly}(s)} g_{i,1} \cdot g_{i,2} \cdot \ldots \cdot g_{i,2^m},$$

where $\deg(g_{i,j}) \leq (2/3)^m \cdot d$, and $\mathrm{size}(g_{ij}) = O(s)$. For each product $g_{i,1} \cdot \ldots g_{i,2^m}$, pick a $j_1 \in [2^m]$ such that $d/3 \leq \sum_{k=1}^{j_1} \deg(g_{i,k}) < 4d/11$. As each $\deg(g_{i,k})$ is less than the gap between upper and lower bounds, namely $4d/11 - d/3$, such j_1 exists. Note that, $\sum_{k=j_1+1}^{2^m} \deg(g_{i,k}) > d - 4d/11 = 7d/11 > d/3$. Choose a $[2^m] \ni j_2 > j_1$ such that $d/3 \leq \sum_{k=j_1+1}^{j_2} \deg(g_{i,k}) < 4d/11$; such j_2 exists by a similar argument.

Define, $f_{i1} := g_{i,1} \cdot \ldots \cdot g_{i,j_1}$, $f_{i2} := g_{i,j_1+1} \cdot \ldots \cdot g_{i,j_2}$, and $f_{i3} := g_{i,j_2+1} \cdot \ldots g_{i,2^m}$. By definition, $\deg(f_{i1})$, $\deg(f_{i2}) \in [d/3, 4d/11]$. As, $\deg(f_{i1}) + \deg(f_{i2}) + \deg(f_{i3}) = \sum_{k \in [2^m]} \deg(g_{i,k}) = d \implies \deg(f_{i3}) \leq d/3 < 4d/11$. As each $g_{i,j}$ has a homogeneous circuit of size $O(s)$, so does f_{ij}. Hence, $f(\boldsymbol{x}) = \sum_{i=1}^{\mathsf{poly}(s)} f_{i1} \cdot f_{i2} \cdot f_{i3}$. Use the identity

$$24 \cdot a \cdot b \cdot c = (a+b+c)^3 - (a-b+c)^3 - (a+b-c)^3 + (a-b-c)^3, \tag{9}$$

to write each $f_{i1} \cdot f_{i2} \cdot f_{i3}$ as sum of four cubes. Relabeling yields $f(\boldsymbol{x}) = \sum_{i=1}^{\mathsf{poly}(s)} c_i \cdot f_i^3$. As each f_i is a linear combination of f_{jk}'s, the degree does not change and the size is still $O(s)$.

It is well known that each homogeneous part can be computed by a homogeneous circuit of size $O(sd^2)$. Thus, for non-homogeneous polynomials, s can be replaced by $O(sd^2)$ and the conclusion follows. □

Let k be a constant (to be fixed later) and $\boldsymbol{x} := (x_1, \ldots, x_k)$. For all large enough $n \in \mathbb{N}$, define $d := d(n) := (n+1)^k - 1$. Let $P_{n,k}$ be a k-variate polynomial of individual degree at most n such that after the Kronecker substitution, $P_{n,k}(x, x^{n+1}, \ldots, x^{(n+1)^{k-1}}) := f_d$. It is easy to construct $P_{n,k}$ from a given d; just convert every $x^e \in \mathrm{supp}(f_d)$ to $x_1^{e_1} \cdot \ldots \cdot x_k^{e_k}$, where $e =: \sum_{i=1}^{k} e_i \cdot (n+1)^{i-1}$ and $0 \leq e_i \leq n$.

By the explicitness of f_d, $(P_{n,k})_n$ is a very explicit polynomial family; its coefficient-vector $\mathrm{coef}(P_{n,k})$ can be computed in $\mathsf{poly}(d) = \mathsf{poly}(n)$ time.

Next, we will show the hardness of the polynomial family $(P_{n,k})_n$. The SOC-hardness implies that there exists a constant δ such that $U(f_d, d^{\varepsilon'}) \geq \delta \cdot d$, for all large enough d. Also, let c be the constant such that $s' =: (sd)^c$ in Lemma 5. Let $\mu := 2/(\varepsilon'/c - 1/k)$, and later we will choose $k > c/\varepsilon'$.

Claim 6 (Hardness of $P_{n,k}$)**.** $\text{size}(P_{n,k}) > d^{1/\mu}$, *for all large enough* n.

Assume to the contrary, that there exists an infinite subset $J \subset \mathbb{N}$ such that $\text{size}(P_{n,k}) \leq d^{1/\mu}$, for all $n \in J$. We will show that family (f_d) is not SOC-hard over an infinite subset $J' := \{d : n \in J\} \subseteq \mathbb{N}$, which is a contradiction.

Let C be a circuit of size $\leq d^{1/\mu}$ that computes $P_{n,k}$, for some n. Then, using Lemma 5, we know that there exist $Q_i \in \mathbb{F}[\boldsymbol{x}]$, of degree at most $4 \cdot \deg(P_{n,k})/11 \leq 4kn/11$, such that $P_{n,k} = \sum_{i=1}^{s_0} c_i \cdot Q_i^3$, where $s_0 \leq (d^{1/\mu} \cdot kn)^c$. Apply the Kronecker map $x_i \mapsto x^{(n+1)^{i-1}}$ on both sides yields $f_d = \sum_{i=1}^{s_0} c_i \cdot \tilde{Q}_i^3$, where $\tilde{Q}_i := Q_i(x, x^{n+1}, \ldots, x^{(n+1)^{k-1}})$. Since Kronecker substitution cannot increase the support size, $|\bigcup_i \text{supp}(\tilde{Q}_i)| \leq |\bigcup_i \text{supp}(Q_i)| \leq \binom{k+4kn/11}{k} =: s_1$. Thus, $U_{\mathbb{F}}(f_d, s_0) \leq s_1$.

We want to show that $s_0 < d^{\varepsilon'}$ and $s_1 < \delta \cdot d$, for all large enough n. Then, we have $U_{\mathbb{F}}(f_d, d^{\varepsilon'}) < \delta \cdot d$, for all large $d \in J' \subset \mathbb{N}$; which contradicts the SOC-hardness of f_d.

Bound on s_0**.** We have for large enough n (and thus d),

$$s_0 \leq (d^{1/\mu} \cdot k \cdot n)^c < d^{c/\mu} \cdot k^c \cdot d^{c/k} = k^c \cdot d^{c/\mu + c/k} < d^{\varepsilon'} .$$

We used that $d = (n+1)^k - 1 > n^k$ for large n, and $\mu > 1/(\varepsilon'/c - 1/k) \iff 1/\mu + 1/k < \varepsilon'/c$.

Bound on s_1**.** By Eqn.(8), we have

$$s_1 = \binom{k + 4nk/11}{k} \leq (e\,(1 + 4n/11))^k < (10.9n/11)^k < (10.9/11)^k \cdot d\,.$$

As $4e \approx 10.873$, we used that $e(1 + 4n/11) < (10.9/11) \cdot n$ and $d > n^k$, for large n .

Therefore, it suffices to show that $(10.9/11)^k < \delta$. Choose $k > \log_{11/10.9}(1/\delta)$. It suffices, from the above calculations, to pick $k > \max\left(c/\varepsilon', \log_{11/10.9}(1/\delta)\right)$. This proves Claim 6. □

From hardness to HSG. We show that from the hardness of $P_{n,k}$ in Claim 6, we can fulfil the assumption in Theorem 20: $\text{size}(P_{n,k}) > s^{10k+2} \deg(P_{n,k})^3$, for some 'growing' function $s = s(n)$. Recall that $\deg(P_{n,k}) \leq kn$. We define, $s(n) := n^{1/(10k+3)}$. Then we have

$$s^{10k+2} (kn)^3 = n^{(10k+2)/(10k+3)} (kn)^3 = k^3 n^{4-(1/(10k+3))} < n^4 , \tag{10}$$

for large enough n. Additionally, assume that $4 \leq k/\mu$. Recall the fact: $n^k < d$ for large n. So, we can continue Eqn.(10) as

$$n^4 \leq n^{k/\mu} < d^{1/\mu} < \text{size}(P_{n,k}) . \tag{11}$$

Equations (10) and (11) give the desired hardness of $P_{n,k}$. It remains to ensure the last requirement of $4 \leq k/\mu$. We show below that choosing $k \geq 9c/\varepsilon'$ suffices:

$$\mu = 2/(\varepsilon'/c - 1/k) \leq 2/\,(9/k - 1/k) = k/4\,.$$

Hence our final choice for k is: $k \geq \max(9c/\varepsilon', \log_{11/10.9}(1/\delta))$.

Thus, Theorem 20 gives a $\text{poly}(s)$-time HSG for $\mathcal{C}(s, s, s)$. Hence, blackbox-PIT $\in$ P. □

Remark. Recall the proof notation. As the degree of Q_i's is $< 4kn/11$, the degree of $\tilde{Q}_i$ is $\leq (n+1)^{k-1} \cdot 4kn/11 < 4k/11 \cdot (n+1)^k = 4k/11 \cdot (d+1) = O(d)$ ($\because k$ is a constant). Thus, it *suffices* to study the representation of f_d as sum-of-cubes $\tilde{Q}_i^3$, where $\deg(\tilde{Q}_i) \leq O(d)$.

4 Conclusion

This work established that studying the univariate sum-of-squares representation (resp. cubes) is fruitful. Proving a *vanishingly* better lower bound than the trivial one, suffices to both derandomize and prove hardness in algebraic complexity.

Here are some immediate questions which require rigorous investigation.

1. Does existence of a SOS-hard family solve PIT completely? The current proof technique fails to reduce from cubes to squares.

2. Prove existence of a SOS-hard family for the *sum of constantly* many squares.
3. Prove existence of a SOC-hard family for a 'generic' polynomial f with rational coefficients ($\mathbb{Q}$). Does it fail when we move to *complex* coefficients ($\mathbb{C}$)?
4. Can we optimize ε in the SOS-hardness condition (& Corollary 4)? In particular, does proving an SOS lower-bound of $\sqrt{d} \cdot \mathsf{poly}(\log d)$, suffice to deduce a separation between determinant and permanent (similarly VP and VNP)?

Acknowledgments. P. D. thanks CSE, IIT Kanpur for the hospitality, and acknowledges the support of Google PhD Fellowship. N. S. thanks the funding support from DST (DST/SJF/MSA-01/2013-14) and N. Rama Rao Chair. Thanks to Manindra Agrawal for many useful discussions to optimize the SOS representations; to J. Maurice Rojas for several comments; to Arkadev Chattopadhyay for organizing a TIFR Seminar on this work. T. T. thanks DFG for the funding (grant TH 472/5-1), and CSE, IIT Kanpur for the hospitality.

References

[Agr20] Manindra Agrawal. Private Communication, 2020. 19

[AGS19] Manindra Agrawal, Sumanta Ghosh, and Nitin Saxena. Bootstrapping variables in algebraic circuits. *Proceedings of the National Academy of Sciences*, 116(17):8107–8118, 2019. Earlier in Symposium on Theory of Computing, 2018 (STOC'18). 2, 4, 5, 7

[AJMV98] Eric Allender, Jia Jiao, Meena Mahajan, and V. Vinay. Non-Commutative Arithmetic Circuits: Depth Reduction and Size Lower Bounds. *Theoretical Computer Science*, 209(1-2):47–86, 1998. 20

[AV08] Manindra Agrawal and V Vinay. Arithmetic Circuits: A Chasm at Depth Four. In *Foundations of Computer Science, 2008. FOCS'08. IEEE 49th Annual IEEE Symposium on*, pages 67–75. IEEE, 2008. 2, 7

[BCS13] Peter Bürgisser, Michael Clausen, and Amin Shokrollahi. *Algebraic Complexity Theory*, volume 315. Springer Science & Business Media, 2013. 2, 8

[BM16] Boaz Barak and Ankur Moitra. Noisy tensor completion via the Sum-of-squares Hierarchy. In *Conference on Learning Theory*, pages 417–445, 2016. 1

[BT15] Grigoriy Blekherman and Zach Teitler. On maximum, typical and generic ranks. *Mathematische Annalen*, 362(3-4):1021–1031, 2015. 8

[Bür09] Peter Bürgisser. On Defining Integers and Proving Arithmetic Circuit Lower Bounds. *Computational Complexity*, 18(1):81–103, 2009. 2

[Bür13] Peter Bürgisser. *Completeness and Reduction in Algebraic Complexity Theory*, volume 7. Springer Science & Business Media, 2013. 2, 22

[CCG12] Enrico Carlini, Maria Virginia Catalisano, and Anthony V Geramita. The solution to the Waring problem for monomials and the sum of coprime monomials. *Journal of algebra*, 370:5–14, 2012. 8

[CKW11] Xi Chen, Neeraj Kayal, and Avi Wigderson. *Partial derivatives in arithmetic complexity and beyond*. Now Publishers Inc, 2011. 2

[Dix64] John D Dixon. Another Proof of Lagrange's Four Square Theorem. *The American Mathematical Monthly*, 71(3):286–288, 1964. 8

[DL78] Richard A. Demillo and Richard J. Lipton. A probabilistic remark on algebraic program testing. *Information Processing Letters*, 7(4):193 – 195, 1978. 2

[FOS12] Ralf Fröberg, Giorgio Ottaviani, and Boris Shapiro. On the Waring problem for polynomial rings. *Proceedings of the National Academy of Sciences*, 109(15):5600–5602, 2012. 8

[GKKS13] Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Arithmetic circuits: A chasm at depth three. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 578–587. IEEE, 2013. 2, 7

[GKSS19] Zeyu Guo, Mrinal Kumar, Ramprasad Saptharishi, and Noam Solomon. Derandomization from Algebraic Hardness: Treading the Borders. In *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019*, pages 147–157, 2019. Online version: https://mrinalkr. bitbucket. io/papers/newprg. pdf. 2, 4, 5, 6, 12, 23, 24

[GMK17] Ignacio Garcia-Marco and Pascal Koiran. Lower bounds by Birkhoff interpolation. *Journal of Complexity*, 39:38–50, 2017. 8

[Hru13] Pavel Hrubeš. On the Real τ-Conjecture and the Distribution of Complex Roots. *Theory of Computing*, 9(1):403–411, 2013. 7

[HS80] Joos Heintz and Malte Sieveking. Lower bounds for polynomials with algebraic coefficients. *Theoretical Computer Science*, 11(3):321–330, 1980. 2

[HWY11] Pavel Hrubeš, Avi Wigderson, and Amir Yehudayoff. Non-commutative circuits and the sum-of-squares problem. *Journal of the American Mathematical Society*, 24(3):871–898, 2011. 4

[Kem12] Aubrey Kempner. Bemerkungen zum Waringschen Problem. *Mathematische Annalen*, 72(3):387–399, 1912. 8

[KI04] Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Computational Complexity*, 13(1-2):1–46, 2004. 2, 4, 5, 11, 23

[KKPS15] Neeraj Kayal, Pascal Koiran, Timothée Pecatte, and Chandan Saha. Lower bounds for sums of powers of low degree univariates. In *International Colloquium on Automata, Languages, and Programming*, pages 810–821. Springer, 2015. 7

[Koi11] Pascal Koiran. Shallow circuits with high-powered inputs. In *Innovations in Computer Science - ICS 2010, Tsinghua University, Beijing, China, January 7-9, 2011. Proceedings*, pages 309–320, 2011. 7

[Koi12] Pascal Koiran. Arithmetic circuits: The chasm at depth four gets wider. *Theoretical Computer Science*, 448:56–65, 2012. 2, 7, 8

[KP11] Pascal Koiran and Sylvain Perifel. Interpolation in Valiant's theory. *Computational Complexity*, 20(1):1–20, 2011. 22

[KPGM18] Pascal Koiran, Timothée Pecatte, and Ignacio Garcia-Marco. On the linear independence of shifted powers. *Journal of Complexity*, 45:67–82, 2018. 8

[KPT15] Pascal Koiran, Natacha Portier, and Sébastien Tavenas. A Wronskian approach to the real τ-conjecture. *Journal of Symbolic Computation*, 68:195–214, 2015. 7

[KPTT15] Pascal Koiran, Natacha Portier, Sébastien Tavenas, and Stéphan Thomassé. A τ-Conjecture for Newton Polygons. *Foundations of computational mathematics*, 15(1):185–197, 2015. 7

[KST19] Mrinal Kumar, Ramprasad Saptharishi, and Anamay Tengse. Near-optimal Bootstrapping of Hitting Sets for Algebraic Circuits. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 639–646, 2019. 2

[Kum19] Mrinal Kumar. A quadratic lower bound for homogeneous algebraic branching programs. *computational complexity*, 28(3):409–435, 2019. 6, 9

[Las07] Jean B Lasserre. A sum of squares approximation of nonnegative polynomials. *SIAM review*, 49(4):651–669, 2007. 1

[Lau09] Monique Laurent. Sums of squares, moment matrices and optimization over polynomials. In *Emerging applications of algebraic geometry*, pages 157–270. Springer, 2009. 1

[Mah14] Meena Mahajan. Algebraic Complexity Classes. In *Perspectives in Computational Complexity*, pages 51–75. Springer, 2014. 8

[MH02] John C Mason and David C Handscomb. *Chebyshev polynomials*. CRC press, 2002. 2

[Mul12] Ketan D. Mulmuley. The GCT Program Toward the P vs. NP Problem. *Commun. ACM*, 55(6):98–107, June 2012. 2

[MV99] Meena Mahajan and V Vinay. Determinant: Old algorithms, new insights. *SIAM Journal on Discrete Mathematics*, 12(4):474–490, 1999. 9

[NW94] Noam Nisan and Avi Wigderson. Hardness vs randomness. *Journal of computer and System Sciences*, 49(2):149–167, 1994. 2, 4

[Ore22] Øystein Ore. Über höhere kongruenzen. *Norsk Mat. Forenings Skrifter*, 1(7):15, 1922. 2

[Pfi76] Albrecht Pfister. Hilbert's seventeenth problem and related problems on definite forms. In *Mathematical Developments Arising from Hilbert Problems, Proc. Sympos. Pure Math, XXVIII.2.AMS*, volume 28, pages 483–489, 1976. 1

[Ram17] Srinivasa Ramanujan. On the Expression of a Number in the Form axˆ 2+ byˆ 2+ czˆ 2+ duˆ 2. In *Proc. Cambridge Philos. Soc.*, volume 19, pages 11–21, 1917. 1

[Raz10] Ran Raz. Elusive Functions and Lower Bounds for Arithmetic Circuits. *Theory Comput.*, 6(1):135–177, 2010. 7

[Rez78] Bruce Reznick. Extremal PSD forms with few terms. *Duke mathematical journal*, 45(2):363–374, 1978. 1

[RY11] Ran Raz and Amir Yehudayoff. Multilinear formulas, maximal-partition discrepancy and mixed-sources extractors. *Journal of Computer and System Sciences*, 77(1):167–190, 2011. 8

[Sap19] Ramprasad Saptharishi. A survey of lower bounds in arithmetic circuit complexity. Github survey, 2019. 3, 6, 9, 20, 21

[Sau12] Nitin Saurabh. ALGEBRAIC MODELS OF COMPUTATION. MS Thesis, 2012. 9

[Sax09] Nitin Saxena. Progress on Polynomial Identity Testing. *Bulletin of the EATCS*, 99:49–79, 2009. 2

[Sax14] Nitin Saxena. Progress on Polynomial Identity Testing - II. *Perspectives in Computational Complexity*, 26:131–146, 2014. 2

[Sch80] J. T. Schwartz. Fast Probabilistic Algorithms for Verification of Polynomial Identities. *J. ACM*, 27(4):701–717, October 1980. 2

[Sma98] Steve Smale. Mathematical problems for the next century. *The mathematical intelligencer*, 20(2):7–15, 1998. 7

[Sri19] Srikanth Srinivasan. Strongly exponential separation between monotone VP and monotone VNP. *arXiv preprint arXiv:1903.01630*, 2019. 8

[SS95] Michael Shub and Steve Smale. On the intractability of Hilbert's Nullstellensatz and an algebraic version of " NP$\neq$ P?". *Duke Mathematical Journal*, 81(1):47–54, 1995. 7

[Str74] Volker Strassen. Polynomials with rational coefficients which are hard to compute. *SIAM Journal on Computing*, 3(2):128–149, 1974. 2

[SY10] Amir Shpilka and Amir Yehudayoff. Arithmetic Circuits: A survey of recent results and open questions. *Foundations and Trends® in Theoretical Computer Science*, 5(3–4):207–388, 2010. 2, 3, 8, 9

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

[Val79] Leslie G Valiant. Completeness classes in algebra. In *Proceedings of the 11th Annual ACM symposium on Theory of computing*, pages 249–261. ACM, 1979. 8, 9, 22

[VSBR83] Leslie G. Valiant, Sven Skyum, S. Berkowitz, and Charles Rackoff. Fast Parallel Computation of Polynomials Using Few Processors. *SIAM Journal of Computing*, 12(4):641–644, 1983. 3, 5, 6, 9, 11, 20

[Wig17] Avi Wigderson. Low-depth arithmetic circuits: technical perspective. *Communications of the ACM*, 60(6):91–92, 2017. 2

[Wik] Wikipedia. Binomial coefficient– bounds and asymptotic formulas. https://en.wikipedia.org/wiki/Binomial_coefficient#Bounds_and_asymptotic_formulas. 8

[Yeh19] Amir Yehudayoff. Separating monotone VP and VNP. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing*, pages 425–429, 2019. 8

[Zip79] Richard Zippel. Probabilistic Algorithms for Sparse Polynomials. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, EUROSAM '79, pages 216–226, 1979. 2

A Sum of powers of small support-union

We give a way to represent any univariate polynomial as sum of r-th powers of polynomials.

Here we use the notion of sumset. In additive combinatorics, the *sumset*, also called the *Minkowski sum* of two subsets A and B of an abelian group G is defined to be the set of all sums of an element from A with an element from B,

$$A + B \;=\; \{\, a + b \mid a \in A,\; b \in B \,\}\,.$$

The *n-fold iterated sumset* of A is $nA = A + \;\cdots\; + A$, where there are n summands.

We want a *small support-union* representation of a d-degree polynomial f as a sum of r-th powers, where r is constant. We consider a *small* B such that rB covers $\{0, 1, \ldots, d\}$. Let t be the *unique* non-negative integer such that $(t-1)^r < d+1 \le t^r$. Define the set B as

$$B \;=\; \{\, a_i\, t^k \mid 0 \le a_i \le t-1,\; 0 \le k \le r-1 \,\}\,.$$

So $|B| = rt = O(d^{1/r})$. Let $k \in \{0, 1, \ldots, d\}$. The base-$t$ representation of k is a sum of at most r elements from B. Hence, $\{0, 1, \ldots, d\} \subseteq rB$.

The largest element in B is $m := (t-1)t^{r-1}$. Note that, for any $\epsilon > 0$, we have $t < (1+\epsilon)(d+1)^{1/r}$, for all large enough d. Thus, for *any* constant $c > 1$ and large enough d, we have $m < c(d+1)$. Therefore, the largest element in rB is at most $mr < cr(d+1) = O(d)$.

Lemma 7. *Let $\mathbb{F}$ be a field of characteristic 0 or large. For any $f(x) \in \mathbb{F}[x]$ of degree d, there exist $\ell_i \in \mathbb{F}[x]$ with $\mathrm{supp}(\ell_i) \subseteq B$ and $c_i \in \mathbb{F}$, for $i = 0, 1, \ldots, mr$, such that $f(x) \;=\; \sum_{i=0}^{mr} c_i\, \ell_i^r$.*

Proof. Consider $\ell_i(\boldsymbol{z}_i, x) = \sum_{j \in B} z_{ij} x^j$, for distinct indeterminates z_{ij}, for all i, j. Surely, $\deg_x(\ell_i) = m$. There exists $mr + 1$ many degree-r polynomials Q_j over $|B| = rt$ many variables, such that

$$\ell_i(\boldsymbol{z}_i, x)^r \;=\; \sum_{j=0}^{mr} Q_j(\boldsymbol{z}_i)\, x^j \qquad \forall i \in [mr]\,.$$

Note that from any monomial in Q_j we could recover j uniquely. Denote the index set $S \subseteq [0, mr]$ such that $Q_j \neq 0$, for all $j \in S$. We could conclude that $Q_j(\boldsymbol{z}_i)$ $(j \in S)$ are $\mathbb{F}$-linearly independent. We would only focus on the Q_j's for $j \in S$, now onwards. Note: $[0 \ldots d] \subseteq S$.

Suppose $f(x) =: \sum_{i=0}^{d} f_i\, x^i$. Define $\tilde{f} \in \mathbb{F}^{|S|}$ and $A \in \mathbb{F}[\boldsymbol{z}]^{|S| \times |S|}$ as

$$\tilde{f} := \begin{pmatrix} f_0 & f_1 & \cdots & f_d & 0 & \cdots & 0 \end{pmatrix},\; A := \begin{pmatrix} Q_{j_1}(\boldsymbol{z}_1) & Q_{j_2}(\boldsymbol{z}_1) & \cdots & Q_{j_s}(\boldsymbol{z}_1) \\ Q_{j_1}(\boldsymbol{z}_2) & Q_{j_2}(\boldsymbol{z}_2) & \cdots & Q_{j_s}(\boldsymbol{z}_2) \\ \vdots & \vdots & \cdots & \vdots \\ Q_{j_1}(\boldsymbol{z}_{|S|}) & Q_{j_2}(\boldsymbol{z}_{|S|}) & \cdots & Q_{j_s}(\boldsymbol{z}_{|S|}) \end{pmatrix}.$$

We want to find $\boldsymbol{c} = \begin{pmatrix} c_1 & c_2 & \cdots & c_{|S|} \end{pmatrix} \in \mathbb{F}^{|S|}$ and $\boldsymbol{\alpha} = (\alpha_{ij})_{i,j}$ such that

$$\sum_{j \in [|S|]} c_j \cdot \ell_j(\boldsymbol{\alpha}, x)^r \;=\; \sum_{i=0}^{d} f_i\, x^i \iff \boldsymbol{c} \cdot A|_{z=\boldsymbol{\alpha}} \cdot \begin{pmatrix} \vdots \\ x^j \\ \vdots \end{pmatrix}_{j \in S} \;=\; \tilde{f} \cdot \begin{pmatrix} \vdots \\ x^j \\ \vdots \end{pmatrix}_{j \in S} \iff \boldsymbol{c} \cdot A|_{z=\boldsymbol{\alpha}} = \tilde{f}\,.$$

As the $\boldsymbol{z}_i$'s are distinct variables, the first column of A consists of different variables at each coordinate. Moreover, the first row of A contains $\mathbb{F}$-linearly independent Q_j's. Thus, for *random* $\alpha_{ij} \in \mathbb{F}$, matrix $A|_{z=\boldsymbol{\alpha}}$ has *full* rank over $\mathbb{F}$. Fix such an $\boldsymbol{\alpha}$. This fixes $\boldsymbol{c} = \tilde{f} \cdot (A|_{z=\boldsymbol{\alpha}})^{-1}$.

From the above construction, it follows that $f(x) \;=\; \sum_{j \in [|S|]} c_j \cdot \ell_j(\boldsymbol{\alpha}, x)^r$. □

The number of *distinct* monomials across $\ell_j(\boldsymbol{\alpha}, x)$'s is $|B| = O(d^{1/r})$. While the top-fanin, as seen before, is $\le mr + 1 = \Theta(d)$.

Remarks. 1. The above calculation does *not* give small support-sum representation of f, as the top-fanin is already $\Omega(d)$.

2. The above representation crucially requires a *field* $\mathbb{F}$. E.g. it does not exist for f_d over the ring $\mathbb{Z}$.

B Further optimizing the top-fanin

In this section, we show a surprising SOS (& SOC) representation for any polynomial $f(x)$, wherein both the top-fanin *and* the support-union size are nontrivially small (namely, $O(\sqrt{d})$). Wlog, assume that characteristic of $\mathbb{F}$ is $\neq 2$ (additionally $\neq 3$, in case of SOC). This is based on the discussions with Agrawal [Agr20].

B.1 Small SOS

Theorem 8 (Small SOS Representation)**.** *For any polynomial $f(x) \in \mathbb{F}[x]$ of degree d, there exist c_i and $f_i \in \mathbb{F}[x]$ such that $f = \sum_{i=1}^{s} c_i \cdot f_i^2$, where $|\bigcup_i \mathrm{supp}(f_i)| = O(\sqrt{d})$ and $s = O(\sqrt{d})$.*

We will show a more general reduction that reduces a *small* support-size, but *big* fanin, representation to a *small* fanin representation (without increasing the support-union).

Lemma 9. *Suppose $f(x) \in \mathbb{F}[x]$ such that $f = \sum_{i=1}^{s} c_i \cdot f_{i1} \cdot f_{i2}$, where $s > t := |\bigcup_{i,j} \mathrm{supp}(f_{ij})|$. Then, there exist c'_i and f'_{ij} such that $f = \sum_{i=1}^{t} c'_i \cdot f'_{i1} \cdot f'_{i2}$, where $|\bigcup_{i,j} \mathrm{supp}(f_{ij})| \leq t$.*

Let us first argue why the above lemma suffices to prove Theorem 8. Note that, Lemma 7 shows that any $f(x)$ can be written as $f(x) = \sum_{i=1}^{O(d)} c_i \cdot f_i^2$, where $|\bigcup_i \mathrm{supp}(f_i)| = O(\sqrt{d})$. Applying Lemma 9, it follows that $f(x)$ can be re-written as $f(x) = \sum_{i=1}^{O(\sqrt{d})} c'_i \cdot f_{i1} \cdot f_{i2}$, where $|\bigcup_{i,j} \mathrm{supp}(f_{ij})| = O(\sqrt{d})$. Write each $f_{i1} \cdot f_{i2} = 1/4 \cdot (f_{i1} + f_{i2})^2 - 1/4 \cdot (f_{i1} - f_{i2})^2$. Note that, the last step does not change the support-union, and at most doubles the top-fanin. Thus, assuming Lemma 9, Theorem 8 directly follows.

Proof of Lemma 9. Let $\{x^{e_1}, \ldots, x^{e_t}\} = S$ be the support-union set of the polynomials f_{ij}, i.e. $S = |\bigcup_{i,j} \mathrm{supp}(f_{i,j})|$. Assume that $e_1 > e_2 > \ldots > e_t$. We denote $\mathrm{LM}(f)$, as the monomial with the highest exponent in f. We show a simple reduction, where the *Property* $\mathbb{P}$ holds:

1. $\forall i \in [t]$, x^{e_i} appears in at most one of the products, say the j-th, and in that case $x^{e_i} = \max(\mathrm{LM}(f_{j1}), \mathrm{LM}(f_{j2}))$, and
2. the support-union set does not increase.

As $|S| = t$, the lemma would directly follow.

To exhibit our reduction, assume that there are two products where some x^{e_i} satisfies condition (1). For simplicity of exposition, let us assume the products to be $f_1 \cdot f_2$ and $f_3 \cdot f_4$ (for more than 2, one can repeatedly use this argument). Wlog both are monic and $x^{e_i} = \mathrm{LM}(f_1)$. Write $f_2 =: \delta_2 f_1 + A_2$, where $\delta_2 \in \{0, 1\}$ and $\mathrm{LM}(A_2) < x^{e_i}$. Similar to f_2, rewrite f_3 and f_4, defining $\delta_3, \delta_4 \in \{0, 1\}$; satisfying $\mathrm{LM}(A_3), \mathrm{LM}(A_4) < x^{e_i}$. It is straightforward to see that

$$\begin{aligned} c_1 \cdot f_1 f_2 + c_2 \cdot f_3 f_4 &= c_1 \cdot f_1(\delta_2 f_1 + A_2) + c_2 \cdot (\delta_3 f_1 + A_3)(\delta_4 f_1 + A_4) \\ &= f_1 \cdot ((c_1\delta_2 + c_2\delta_3\delta_4) f_1 + c_1 A_2 + c_2\delta_4 A_3 + c_2\delta_3 A_4) + c_2 A_3 A_4 \,. \end{aligned}$$

In this new sum of two products, x^{e_i} appears only in the first product (and as a leading-monomial); while in the second product, A_3, A_4 have lower leading-monomials. Also, the support-union set has not increased. Thus, Property $\mathbb{P}$ is respected till now. Repeated application of this reduction to each e_i, and each occurrence, gives us the desired conclusion. □

B.2 Small SOC

We show a similar small SOC representation using the above SOS trick.

Corollary 10 ($\sqrt{d}$-SOC representation)**.** *For any polynomial $f(x) \in \mathbb{F}[x]$ of degree d, there exist c_i and $f_i \in \mathbb{F}[x]$ such that $f = \sum_{i=1}^{s} c_i \cdot f_i^3$, where $|\bigcup_i \mathrm{supp}(f_i)| = O(\sqrt{d})$ and $s = O(\sqrt{d})$.*

Proof Sketch. Note that, for any polynomial Q, there are distinct constants λ_i and constants $c_i \in \mathbb{F}$, for $i \in [4]$, such that $Q^2 = \sum_{i\in[4]} c_i \cdot (Q + \lambda_i)^3$. This easily follows by interpolation, with $t = \lambda_i$'s in the cubic (wrt t) $(Q + t)^3$; thus, expressing Q^2 as a sum of four cubes. Theorem 8 expresses $f = \sum c_i \cdot Q_i^2$. The conclusion follows directly after using the above identity on each Q_i^2. □

Lemma 7 shows that for SOC, a $O(d^{1/3})$ support-union representation exists (with $O(d)$ top-fanin). We could show a similar reduction as Theorem 8, but with top-fanin $O(d^{2/3})$.

Theorem 11 ($d^{2/3}$-SOC representation)**.** *For any polynomial $f(x) \in \mathbb{F}[x]$ of degree d, there exist c_i and $f_i \in \mathbb{F}[x]$ such that $f = \sum_{i=1}^{s} c_i \cdot f_i^3$, where $|\bigcup_i \mathrm{supp}(f_i)| = O(d^{1/3})$ and $s = O(d^{2/3})$.*

Corollary 12. *For $s = \Omega(d^{2/3})$, we have $U_{\mathbb{F}}(f, s) = \Theta(d^{1/3})$.*

Proof of Theorem 11. The very basic idea is to show a reduction similar to Lemma 9 for sum of product-of-3. In particular, we show the following:

Claim 13. *If $f = \sum_{i\in[s]} c_i \cdot f_{i1} \cdot f_{i2} \cdot f_{i3}$ with support-union of size t, then f can be re-written as $f = \sum_{i\in[t^2]} c'_i \cdot f'_{i1} \cdot f'_{i2} \cdot f'_{i3}$ with support-union size $\leq t$.*

To show the reduction, to prove the claim, we fix the support-union set S and the monomial ordering (as seen in Lemma 9). Assume there are $m > t^2$ many products, like $f_{i1} \cdot f_{i2} \cdot f_{i3}$. Wlog assume $\mathrm{LM}(f_{11}) = x^{e_i}$. Rearrange $\sum_{i\in[m]} c_i \cdot f_{i1} \cdot f_{i2} \cdot f_{i3} =: f_{11} \cdot P + R$, so that P is an SOS; and R is an SOC without any occurrence of x^{e_i}. Apply Lemma 9, on P, to reduce its top-fanin to t. Repeat this procedure to SOC R.

Finally, the top-fanin gets upper-bounded by $t \cdot t = t^2$, and the claim follows. The theorem follows by noting that any product-of-3 can be written as a sum of four cubes, by Eqn.(9); and by Lemma 7 $t = O(d^{1/3})$. □

Lemma 14. *For any $f \in \mathbb{F}[\boldsymbol{x}]$, we have $S_{\mathbb{F}}(f) \geq \min_s (U_{\mathbb{F}}(f, 4s) - 1)$.*

Proof Sketch. Suppose $f = \sum_{i=1}^{s} c_i \cdot f_i^2$. Write each f_i^2 as $f_i^2 = \sum_{j=1}^{4} c_{ij} \cdot (f_i + \lambda_{ij})^3$, for distinct $\lambda_{ij} \in \mathbb{F}$. Thus, $U_{\mathbb{F}}(f, 4s) \leq (\sum_{i=1}^{s} |f_i|_0) + 1$. Taking minimum over s gives the desired inequality. □

C Circuit normal form and sum of product-of-2 decomposition

In an elegant and influential work, [VSBR83] showed that every efficiently computable polynomial family (by algebraic circuits) is also efficiently computable in *parallel*. Work of [AJMV98] proved a similar result with top-down approach. In both the proof techniques, the notion of *gate quotients* was used. A hybrid, detailed discussion can be sought in [Sap19].

We assume without loss of generality that the given circuit Φ has the following properties: **(i)** Φ is a *homogeneous* circuit, **(ii)** all multiplication gates in Φ have fanin at most *two*, and **(iii)** Φ is a *right heavy* circuit, i.e. the degree of the right child of any multiplication gate is at least as large as the degree of its left child[11]. For any gate u in Φ, we denote by $[u]$ the polynomial computed at gate u. We denote u_L and u_R as left and right child of u respectively.

[11] Any circuit can be made right heavy by swapping the children without incurring any blowup in size.

Definition 6 (Gate quotient). *For gates u, v, the* quotient *polynomial $[u : v]$ is defined as follows:*

1. *If u and v are same nodes, then $[u : v] = 1$*
2. *If u is a leaf, and $u \neq v$, then $[u : v] = 0$*
3. *If $u = u_1 + u_2$, then $[u : v] = [u_1 : v] + [u_2 : v]$*
4. *If $u = u_1 \times u_2$, then $[u : v] = [u_1] \cdot [u_2 : v]$*

Observation. $[u : v]$ is a homogeneous polynomial of degree $\deg(u) - \deg(v)$.

Definition 7 (Frontier). *For any parameter m, define the* frontier at degree m, *denoted $\mathcal{F}_m$, as the deepest nodes in the circuit that have degree at least m. Formally, $\mathcal{F}_m := \{v : deg(v) \geq m, \text{ and } deg(v_L), deg(v_R) < m\}$.*

Observe that all frontier nodes must be multiplication gates. They give a nice decomposition:

Lemma 15 (CNF by frontier decomposition). [Sap19, Lem.5.12] *Let Φ be a homogeneous, right-heavy circuit. Let u be a node such that $deg(u) \geq m$. Then, $[u] = \sum_{w \in \mathcal{F}_m} [u : w] \cdot [w]$.*

Lemma 16 (Sum of product-of-2). *Let $f(\boldsymbol{x})$ be an n-variate, homogeneous, degree d polynomial computed by a right-heavy homogeneous circuit Φ of size s. Then, there exist polynomials $f_{ij} \in \mathbb{F}[\boldsymbol{x}]$ s.t.*

$$f(\boldsymbol{x}) = \sum_{i=1}^{s} f_{i1} \cdot f_{i2}, \quad \text{with the following properties:} \tag{12}$$

1. *$d/3 \leq deg(f_{i1}), deg(f_{i2}) \leq 2d/3$, for all $i \in [s]$,*
2. *$deg(f_{i1}) + deg(f_{i2}) = d$, for all $i \in [s]$, and*
3. *each f_{ij} has a right-heavy homogeneous circuit of size at most $s_2 := O(s)$.*

Proof of Lemma 16. Choose $m := d/3$ in Lemma 15, to conclude that, $f = \sum_{u \in \mathcal{F}_m} [f : u] \cdot [u]$. Note that, $|\mathcal{F}_m| \leq s$. By definition of the frontier, $\deg(u_L), \deg(u_R) < d/3$ (implying $\deg([u]) < 2d/3$). Recall that, $\deg([u]) + \deg([f : u]) = d$. Combining together, we get the range $d/3 \leq \deg([u]), \deg([f : u]) \leq 2d/3$.

Size analysis. We will prove that for *any* node u in Φ, $[f : u]$ can be computed by a right-heavy homogeneous circuit of size $\leq 2s$.

Fix node u. Maintain a 'growing' disjoint *copy* of Φ as circuit Φ', which is intended to inductively compute $[v : u]$, for the 'current' node v. We induct on depth (of v in Φ). The *base case* (namely, a leaf) occupies size ≤ 1 in Φ', but no extra edge/node needs to be added.

Suppose, we have computed Φ' bottom-up, till i-th level ($i \geq 1$). For $i+1$-th level, we need to compute at most two quotients of the form $[v : u]$ (by definition).

Addition: If $v =: v_L + v_R$, then $[v : u] = [v_L : u] + [v_R : u]$. By induction hypothesis, both the quotients are already computed in Φ'. The two input edges and the $+$ gate are already in Φ'. Thus, we do not need to add any extra edge or node in Φ' (i.e. they are copies of those in Φ).

Multiplication: If $v = v_L \cdot v_R$, then $[v : u] = [v_L] \cdot [v_R : u]$. By induction hypothesis, both are pre-computed; $[v_L]$ in Φ and $[v_R : u]$ in Φ'. We *delete* the left incoming-edge of $\times$ in Φ', replacing it with an edge from v_L of Φ. So, no extra edge or node is required.

Thus, $[f : u]$ has a circuit Φ' (with Φ) of size at most $s_2 := 2s$. Φ' is *homogeneous* because all the intermediate nodes $[v]$, and $[v : u]$, are homogeneous polynomials. It has fanin 2 again, by the definition. The *right-heaviness* follows by swapping the sub-circuits appropriately without incurring any size blowup. □

Remark. For a non-homogeneous polynomial $f(\boldsymbol{x})$, we can apply the above for each homogeneous part of $f(\boldsymbol{x})$. It is well known that each homogeneous part can be computed by a homogeneous circuit of size $O(sd^2)$. Thus, for non-homogeneous polynomials, s can be replaces by $O(sd^2)$ and the same conclusion follows.

D Valiant's Criterion for VNP: Details for Section 3.2

A useful *sufficient* condition for a polynomial family $(f_n(\boldsymbol{x}))_n$ to be in VNP is known, due to Valiant [Val79].

Theorem 17 (VNP criterion, [Bür13]). *Let function $\phi_n : \{0,1\}^n \to \{0,1\}^n$ be computable in* #P/poly. *Then, the family of polynomials defined by $f_n(\boldsymbol{x}) := \sum_{\boldsymbol{e}\in\{0,1\}^n} \phi_n(\boldsymbol{e}) \cdot \boldsymbol{x}^{\boldsymbol{e}}$, is in* VNP.

One can further *relax* Theorem 17 where $\phi_n(\boldsymbol{e})$ can actually be 2^n-bit long, see Theorem 3 (restated) below. The proof idea is very similar to [KP11, Lem. 3.2]. We also use the fact that VNP is *closed* under substitution. That is, for a family of polynomials $(f(\boldsymbol{x},\boldsymbol{y})) \in$ VNP, it also holds that $(f(\boldsymbol{x},\boldsymbol{y}_0)) \in$ VNP, for any value $\boldsymbol{y}_0 \in \mathbb{F}^n$, assigned to the variables in $\boldsymbol{y}$.

Theorem 3 (restated). *Let function $\phi_n : \{0,1\}^n \to \{0,1\}^{2^n}$ such that each bit of $\phi_n(\cdot)$, is computable in* #P/poly. *Then, the family of polynomials defined by $f_n(\boldsymbol{x}) := \sum_{\boldsymbol{e}\in\{0,1\}^n} \phi_n(\boldsymbol{e}) \cdot \boldsymbol{x}^{\boldsymbol{e}}$, is in* VNP.

Proof of Theorem 3. As $\phi_n(\boldsymbol{e}) < 2^{2^n}$, it is straightforward to write $\phi_n(\boldsymbol{e}) =: \sum_{j=0}^{2^n-1} \phi_{n,j}(\boldsymbol{e}) \cdot 2^j$, where $\phi_{n,j}(\boldsymbol{e}) \in \{0,1\}$ is computable in #P/poly. We denote the base-2 representation, $\mathrm{bin}(e) := (e_1,\ldots,e_n)$, where $e =: \sum_{i=1}^n e_i \cdot 2^{i-1}$, and $\boldsymbol{x}^{\mathrm{bin}(\boldsymbol{e})} := x_1^{e_1}\ldots x_n^{e_n}$. Introduce new variables $\boldsymbol{y} = (y_1,\ldots,y_n)$ and consider the auxiliary polynomial $\tilde{\phi}_n(\boldsymbol{e},\boldsymbol{y}) := \sum_{j=0}^{2^n-1} \phi_{n,j}(\boldsymbol{e}) \cdot \boldsymbol{y}^{\mathrm{bin}(j)}$. Let $\boldsymbol{y}_0 := (2^{2^0},\ldots,2^{2^{n-1}})$. Note that, $\tilde{\phi}_n(\boldsymbol{e},\boldsymbol{y}_0) = \phi_n(\boldsymbol{e})$. Finally, consider the $2n$-variate auxiliary polynomial $h_n(\boldsymbol{x},\boldsymbol{y})$ as:

$$h_n(\boldsymbol{x},\boldsymbol{y}) := \sum_{e=0}^{2^n-1} \tilde{\phi}_n(\mathrm{bin}(e),\boldsymbol{y}) \cdot \boldsymbol{x}^{\mathrm{bin}(e)} = \sum_{e,j=0}^{2^n-1} \phi_{n,j}(\mathrm{bin}(e)) \cdot \boldsymbol{y}^{\mathrm{bin}(j)} \cdot \boldsymbol{x}^{\mathrm{bin}(e)} .$$

Then, we have $h_n(\boldsymbol{x},\boldsymbol{y}_0) = f_n(\boldsymbol{x})$. Since, $\phi_{n,j}(\mathrm{bin}(e))$ can be computed in #P/poly, we have $(h_n(\boldsymbol{x},\boldsymbol{y}))_n \in$ VNP. As VNP is *closed* under substitution, it follows that $(f_n(\boldsymbol{x}))_n \in$ VNP. □

E SOS-hardness with constant ε implies truly exponential separation between VP and VNP

We use Lemma 15 repeatedly (constant many times) to bring the degree of the intermediate polynomials 'fractional'-close to $d/2$, namely $d \cdot (1/2 + O(1))$. This would be crucially used to establish the exponential separation between VP and VNP.

Lemma 18 (Constant boosting VSBR). *Let $f(\boldsymbol{x}) \in \mathbb{F}[\boldsymbol{x}]$ be a degree-d, n-variate polynomial computed by homogeneous circuit of size s. Then, for any constant $1 < \gamma < 2$, there exist polynomials $f_{ij} \in \mathbb{F}[\boldsymbol{x}]$ such that*

$$f(\boldsymbol{x}) = \sum_{i=1}^{\mathsf{poly}(s)} f_{i1} \cdot f_{i2} \text{ , with the following properties} \tag{13}$$

1. *each f_{ij} has a homogeneous circuit of size $O(s)$,*
2. *$deg(f_{ij}) < d/\gamma$, for all i,j,*
3. *$deg(f_{i1}) + deg(f_{i2}) = d$, for all i.*

Proof. Lemma 16 shows that $f(\boldsymbol{x})$ can be decomposed as $\sum_{i=1}^s \tilde{f}_{i1} \cdot \tilde{f}_{i2}$ where $\tilde{f}_{ij}$ has circuits of size $O(s)$ and $\deg(\tilde{f}_{ij}) \le 2d/3$, with $\deg(\tilde{f}_{i1})+\deg(\tilde{f}_{i2}) = d$.

Let $\delta' := 1/\gamma - 1/2$. Choose a constant $m := \lceil \log_{3/2}(1/\delta') \rceil$ so that $(2/3)^m < \delta'$. Apply the above product-of-2 decomposition m times repeatedly on each product. As m is constant, it is direct to conclude that $f(\boldsymbol{x})$ can be decomposed as

$$f(\boldsymbol{x}) \;=\; \sum_{i=1}^{\mathsf{poly}(s)} g_{i1} \cdot g_{i2} \cdot \ldots \cdot g_{i2^m} \;,$$

where $\deg(g_{ij}) \leq (2/3)^m \cdot d < d \cdot \delta'$ and $\mathrm{size}(g_{ij}) = O(s)$. For each product $g_{i1} \cdot \ldots g_{i2^m}$, pick $j \in [2^m]$ such that $d/2 \leq \sum_{k=1}^{j} \deg(g_{ik}) < d/\gamma$. As each $\deg(g_{ik})$ is less than the gap between upper and lower bounds, namely $d \cdot \delta'$, such j exists.

Define $f_{i1} := g_{i1} \cdot \ldots \cdot g_{ij}$, and $f_{i2} := g_{i,j+1} \cdot \ldots \cdot g_{i,2^m}$. By definition, $\deg(f_{i1}) \in [d/2, d/\gamma)$. As $\deg(f_{i1}) + \deg(f_{i2}) = \sum_{k \in [2^m]} \deg(g_{ik}) = d \implies \deg(f_{i2}) \leq d/2 < d/\gamma$. As each g_{ij} has a homogeneous circuit of size $O(s)$, so does f_{ij}. This completes the proof. □

Now, we are ready to state and prove the main result of this section.

Theorem 19 (Constant ε)**.** *If there exists a univariate family $(f_d(x))_d$ that is SOS-hard with some constant ε, then* VNP *is exponentially harder than* VP *(& blackbox-PIT $\in$ QP).*

Proof Sketch. The notation, and the construction of $P_{n,k}$, mirrors that of Theorem 1 (Section 3.2). We fix k to be some constant (large enough depending on ε) and $n = O(\log d)$, thus $kn = O(\log d)$. The indexing of the family $\{P_{n,k}\}$ is over n.

We show that there exists some constant μ such that $\mathrm{size}(P_{n,k}) > d^\mu = 2^{\Omega(kn)}$. If not, then apply Lemma 18 with some constant γ (to be fixed later). It follows that

$$P_{n,k} \;=\; \sum_{i=1}^{s} c_i \cdot Q_i^2 \;\implies\; f_d \;=\; \sum_{i=1}^{s} c_i \cdot \psi_{n,k}(Q_i)^2$$

where $\deg(Q_i) \leq n/\gamma$ and $s = \mathsf{poly}(d^\mu \cdot n) = d^{\mu \cdot c}$ for some constant c. Above equation implies: $S_{\mathbb{F}}(f_d) \leq s \cdot \binom{kn+n/\gamma}{n/\gamma}$. We want to show that for some constant μ, $S_{\mathbb{F}}(f_d) \leq o(d^{1/2+\varepsilon})$, a contradiction.

The overhead $\varepsilon > 0$ is a constant. We want $s \leq d^{\delta_1}$, and $\binom{kn+n/\gamma}{n/\gamma} \leq d^{\delta_2}$ such that $d^{\delta_1+\delta_2} = o(d^{1/2+\varepsilon})$. To achieve that, fix $\delta_1 := \varepsilon/3$ and $\delta_2 := 1/2 + \varepsilon/3$. Note that, $\delta_1 + \delta_2 < 1/2 + \varepsilon$. Fix μ such that $c \cdot \mu \leq \delta_1$. To show the upper bound on the binomial, note that

$$\begin{aligned}\binom{kn + n/\gamma}{n/\gamma} \;&\leq\; (e + 2ek)^{n/\gamma} \;\leq\; (6(k-1))^{n/\gamma} \\ &\leq\; (k-1)^{n\delta_2} \;\leq\; d^{\delta_2} \;.\end{aligned}$$

The only non-trivial inequality above is the third one; for which $(k-1)^{\delta_2 - 1/\gamma} \geq 6^{1/\gamma}$ suffices. So, choose $2 > \gamma > 1/\delta_2$ (such γ exists as $\delta_2 > 1/2$), and k accordingly. The contradiction with the appropriate SOS-hardness implies that $P_{n,k}$ is exponentially hard (wrt number of variables $kn = \Theta(\log d)$).

The proof that $(P_{n,k})_n \in$ VNP remains the same. Hence, SOS-hardness with constant ε implies VNP is *exponentially*-harder than VP.

The PIT part immediately follows from the explicitness and [KI04, Thm. 7.7]. □

F Hardness to derandomization: Details for Section 3.3

Very recently, Guo et al. in [GKSS19] showed utility of the hardness of *constant* variate polynomials to derandomize PIT. To make this discussion formal, we start with the following definition.

Definition 8 (Hitting-set generator (HSG))**.** *A polynomial map $G : \mathbb{F}^k \longrightarrow \mathbb{F}^n$ given by $G(\boldsymbol{z}) = (g_1(\boldsymbol{z}), g_2(\boldsymbol{z}), \ldots, g_n(\boldsymbol{z}))$ is said to be a* hitting-set generator (HSG) *for a class $\mathcal{C} \subseteq \mathbb{F}[x_1, x_2, \ldots, x_n]$ of polynomials if for every nonzero $f \in \mathcal{C}$, we have that $f \circ G = f(g_1, g_2, \ldots, g_n)$ is nonzero.*

Remark. We say that G is *t-time* HSG if $\mathrm{coef}(g_i)$ can be computed in t-time and maximum degree of g_i is also at most t. This gives $(t \cdot d)^{O(k)}$ time blackbox-PIT algorithm, for circuits computed by degree $\leq d$, over popular fields like: rationals $\mathbb{Q}$ or their extensions, local fields $\mathbb{Q}_p$ or their extensions, or finite fields $\mathbb{F}_q$. When k is constant, we get a poly-time blackbox-PIT.

Given an HSG, it can be seen that there is a corresponding *hitting-set* H such that one needs to *only* query the given input circuit at points on H to determine non-zeroness.

Guo et al. in [GKSS19] came up with an efficient HSG construction (*without* combinatorial designs) from constant-variate circuit-hard polynomials.

Theorem 20. [GKSS19] *Let $P \in \mathbb{F}[\boldsymbol{x}]$ be a k-variate polynomial of degree d such that $\mathrm{coef}(P)$ can be computed in $\mathsf{poly}(d)$-time. If $\mathsf{size}(P) > s^{10k+2} \cdot d^3$, then there is a $\mathsf{poly}(s)$-time HSG for $\mathcal{C}(s, s, s)$.*