Electronic Colloquium on Computational Complexity, Revision 2 of Report No. 65 (2019)

Derandomization from Algebraic Hardness*

Zeyu Guo† Mrinal Kumar‡ Ramprasad Saptharishi§ Noam Solomon¶

Abstract

A hitting-set generator (HSG) is a polynomial map $\mathsf{Gen} : \mathbb{F}^k \to \mathbb{F}^n$ such that for all n-variate polynomials C of small enough circuit size and degree, if C is nonzero, then $C \circ \mathsf{Gen}$ is nonzero. In this paper, we give a new construction of such an HSG assuming that we have an explicit polynomial of sufficient hardness. Formally, we prove the following result over any field $\mathbb{F}$ of characteristic zero:

> Suppose $P(z_1, \ldots, z_k)$ is an explicit k-variate degree d polynomial that is not computable by circuits of size s. Then, there is an explicit hitting-set generator $\mathsf{Gen}_P : \mathbb{F}^{2k} \to \mathbb{F}^n$ such that every nonzero n-variate degree D polynomial $C(\mathbf{x})$ computable by circuits of size s' circuits satisfies $C \neq 0 \Rightarrow C \circ \mathsf{Gen}_P \neq 0$, if $O(n^{10k} d^3 D s') < s$.

This is the first HSG in the algebraic setting that yields a complete derandomization of polynomial identity testing (PIT) for general circuits from a suitable algebraic hardness assumption. Unlike the prior constructions of such maps [NW94, KI04, AGS19, KST19], our construction is purely algebraic and does *not* rely on the notion of combinatorial designs.

As a direct consequence, we show that even saving *a single point* from the "trivial" explicit, exponential sized hitting sets for constant-variate polynomials of low *individual-degree* which are computable by small circuits, implies a deterministic polynomial time algorithm for PIT. More precisely, we show the following:

> Let k be a large enough constant. Suppose for every s large enough, there is an explicit hitting set of size at most $((s+1)^k - 1)$ for the class of k-variate polynomials of *individual degree* s that are computable by size s circuits. Then there is an explicit hitting set of size $s^{O(k^2)}$ for the class of s-variate polynomials, of degree s, that are computable by size s circuits.

*A preliminary version of this paper will appear in the Proceedings of FOCS 2019 [GKSS19].

†zguo@cse.iitk.ac.in. Department of Computer Science & Engineering, IIT Kanpur, India.

‡mrinalkumar08@gmail.com. Department of Computer Science & Engineering, IIT Bombay, India. Part of this work was done while at the Simons Institute for the Theory of Computing, Berkeley during the semester on Lower Bounds in Computational Complexity in Fall 2018 and during a postdoctoral stay at the University of Toronto.

§ramprasad@tifr.res.in. Tata Institute of Fundamental Research, Mumbai, India. Research supported by Ramanujan Fellowship of DST.

¶noam.solom@gmail.com. Department of Mathematics, MIT, Cambridge, MA, USA.

This research was supported in part by the International Centre for Theoretical Sciences (ICTS) during a visit for participating in the program - Workshop on Algebraic Complexity Theory (Code: ICTS/wact2019/03)

ISSN 1433-8092

1 Introduction

The interaction of hardness and randomness is one of the most well studied themes in computational complexity theory, and in this work we focus on exploring this interaction further in the realm of algebraic computation. To set the stage, we start with a brief introduction to algebraic complexity.

The field of algebraic complexity primarily focuses on studying multivariate polynomials and their complexity in terms of the number of basic operations (additions and multiplications) required to compute them. Algebraic circuits (which are just directed acyclic graphs with leaves labelled by variables or field constant, and internal gates labelled by $+$ or $\times$) form a very natural model of computation in this setting, and the size (number of gates or wires) of the smallest algebraic circuit computing a polynomial gives a robust measure of its complexity.

The main protagonists in the hardness-randomness interaction in algebraic complexity are the *hardness* component, which is the question of proving superpolynomial lower bounds for algebraic circuits for any explicit polynomial family, and the *randomness* component, which is the question of designing efficient *deterministic* algorithms for polynomial identity testing (PIT) — the algorithmic task of checking if a given circuit computes the zero polynomial. Both these questions are of fundamental importance in computational complexity and are algebraic analogues of their more well known Boolean counterparts: the P vs NP question and the P vs BPP question respectively. These seemingly different problems are closely related to each other, and in this work we focus on one direction of this relationship; namely, the use of explicit hard polynomial families for derandomization of PIT.

It is known from an influential work of Kabanets and Impagliazzo [KI04] that lower bounds on the algebraic circuit complexity of explicit polynomial families lead to non-trivial deterministic algorithms for polynomial identity testing (PIT) of algebraic circuits. Moreover, the results in [KI04] show that stronger lower bounds give faster deterministic algorithms for polynomial identity testing. For instance, from truly exponential (or $2^{\Omega(n)}$) lower bounds, we get quasipolynomial (or $n^{O(\log n)}$) time deterministic algorithms for PIT. From weaker superpolynomial (or $n^{\omega(1)}$) lower bounds, we only seem to get a subexponential (or $2^{n^{o(1)}}$) time PIT algorithm.

However, no matter how good the lower bounds for algebraic circuits are, this connection between lower bounds and derandomization does not seem to give truly polynomial time deterministic algorithms for PIT. This is different from the Boolean setting, where it is known that strong enough boolean circuit lower bounds imply that $\mathsf{BPP} = \mathsf{P}$ [IW97]. The difference stems from the fact that, in the worst case, an n-variate degree d polynomial P needs to be queried on as many as $\binom{n+d}{d} \gg 2^n$ points to be sure of its nonzeroness. A key player in this interaction of hardness and randomness, in the context of algebraic complexity, is the notion of a *hitting-set generator* (HSG), which we now define.

Definition 1.1 (Hitting-set generators)**.** *A polynomial map $G : \mathbb{F}^k \to \mathbb{F}^m$ given by $G(z_1, z_2, \ldots, z_k) = (g_1(\mathbf{z}), g_2(\mathbf{z}), \ldots, g_n(\mathbf{z}))$ 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 $Q \in \mathcal{C}$, we have that $Q \circ G = Q(g_1, g_2, \ldots, g_n)$ is also nonzero.

*We shall say that G is $t(n)$-*explicit *if, for any $\mathbf{a} \in \mathbb{F}^k$ of bit complexity at most n, we can compute $G(\mathbf{a})$ in deterministic time $t(n)$. Here k is called the* seed length *of the HSG and n is called the* stretch *of the HSG. The maximum of the degrees of $g_1, g_2, \ldots, g_n$ is called the* degree *of the HSG.* $\diamondsuit$

If a polynomial map G is an HSG for a class $\mathcal{C}$ of circuits, we say G *fools* the class $\mathcal{C}$.

Informally, an HSG G gives a polynomial map which reduces the number of variables in the polynomials in $\mathcal{C}$ from n to k while preserving their nonzeroness. It is not hard to see that such polynomial maps are helpful for deterministic PIT for $\mathcal{C}$. To test if a given n-variate polynomial $Q \in \mathcal{C}$ is nonzero, it is sufficient to check that $Q \circ G$, a k-variate polynomial, is nonzero. If the degree of each g_i is not-too-large, then a "brute-force" check (via the Ore-DeMillo-Lipton-Schwartz-Zippel Lemma [Ore22, DL78, Sch80, Zip79]) can be used to test if $Q \circ G$ is zero in at most $\text{poly}(t(n)) \cdot (\deg(G) \cdot \deg(Q))^{O(k)}$ time, if G is $t(n)$-explicit. Thus, it is desirable to have HSGs that are very explicit (small $t(n)$), low degree and large stretch ($k \ll n$).

1.1 Prior construction of generators

We shall use $\mathcal{C}(n, d, s)$ to denote the class of n-variate polynomials of degree at most d that are computable by size s circuits, and $\mathcal{C}(n, \text{i-deg: } d, s)$ to denote the class of n-variate polynomials of *individual degree*[1] at most d that are computable by size s circuits.

Generators from combinatorial designs: One of the earliest (and most well-known) applications of lower bounds to derandomization is the construction of *pseudorandom generators (PRG)* from hard explicit Boolean functions by Nisan and Wigderson [NW94]. In the algebraic setting, an analogous construction was shown to produce HSGs by Kabanets and Impagliazzo [KI04][2]. These constructions are based on the notion of a combinatorial design, which is a family of subsets that have small pairwise intersection. Given an explicit construction of such a combinatorial design (e.g. a family $\mathcal{F} = \{S_1, S_2, \ldots, S_n\}$ of subsets of $[k]$ of size t each), the PRG/HSG in [NW94, KI04] is then constructed by just taking a hard polynomial $P(x_1, \ldots, x_t)$ and defining the map as $G(y_1, y_2, \ldots, y_k) = (P(\mathbf{y}\,|_{S_1}), P(\mathbf{y}\,|_{S_2}), \ldots, P(\mathbf{y}\,|_{S_n}))$. The proof of correctness for this HSG goes via a hybrid argument and a result of Kaltofen [Kal89].

Bootstrapping hitting sets and HSGs with large stretch: In a recent line of work [AGS19, KST19] the following surprising *boostrapping* phenomenon was shown to be true for hitting sets for algebraic circuits. The following is the statement from [KST19]:

Theorem 1.2 ([KST19])**.** *Let $\delta > 0$ and $n \geq 2$ be constants. Suppose that, for all large enough s, there is an explicit hitting set of size $s^{n-\delta}$ for all degree s, size s algebraic formulas (or algebraic branching programs,*

[1] maximum degree of any variable; for example, a multilinear polynomial has individual degree 1.

[2] Even though the construction of the generator is the same in [KI04] and [NW94], there are crucial differences in the analysis. In particular, the analysis for the HSG in [KI04] relies on a deep result of Kaltofen [Kal89] about low degree algebraic circuits being closed under polynomial factorization.

or circuits respectively) over n variables. Then, there is an explicit hitting set of size $s^{\exp(\exp(O(\log^* s)))}$ for the class of degree s, size s algebraic formulas (or algebraic branching programs, or circuits respectively) over s variables.

In other words, a slightly non-trivial explicit construction of hitting sets even for constant-variate algebraic circuits implies an almost complete derandomization of PIT for algebraic circuits. A natural question in this direction which has remained open is the following.

Question 1.3 ([KST19])**.** *Can slightly non-trivial hitting sets for constant-variate algebraic circuits can be bootstrapped to get polynomial size (and not just almost polynomial size as in Theorem 1.2) hitting sets for all circuits ?*

The proof of Theorem 1.2 can also be interpreted as a different HSG for algebraic computation. This HSG, given the hypothesis of Theorem 1.2, stretches k bits to n bits (for arbitrarily large n), but the degree and explicitness of the generator grows as $n^{\exp(\exp(O(\log^* n)))}$. Thus, this construction comes very close to answering Question 1.3 without completely answering it. This HSG is essentially constructed via a repeated composition of the HSG in [KI04, NW94] where, for each step, it uses a different hard polynomial with gradually increasing hardness. Due to this inherent iterative nature of the construction, it seems difficult to reduce the degree and explicitness of such HSG constructions to $\text{poly}(n)$.

The need to go beyond design-based HSGs: In the set up of boolean computation, observe that we cannot expect to have any PRG (or even HSG) of seed length k to fool circuits of size much larger than $n2^k$ since we can construct a circuit of size $O(n2^k)$ to identify the range of the generator (consisting of 2^k strings of length n each). A similar argument gives an upper bound of $(dD)^{O(k)}$ on the size of degree D algebraic circuits which can be fooled by a HSG with seed length k and degree d. Thus, while the stretch of any boolean PRG constructed via hardness of a boolean function is upper bounded by $n2^k$, in the algebraic setting, one could hope for a construction of hitting-set generators of stretch as large as $d^{\Omega(k)}$ from sufficiently hard explicit polynomial families.[3] However, till recently, there were no known constructions of such HSGs with stretch larger than 2^k. An HSG with strong enough parameters would answer the following very natural question.

Question 1.4. *If there is an explicit polynomial family $\{P_k\}_{k\in\mathbb{N}}$, where P_k is a k-variate polynomial of degree d such that any algebraic circuit computing it has size $d^{\Omega(k)}$, then is PIT in* P *?*

Another reason for looking beyond the design-based HSGs in the algebraic setting is that by definition, a design-based HSG is combinatorial. Aesthetically, it seems desirable to have a route from algebraic lower bounds to algebraic pseudorandomness which does not rely on clever combinatorial constructions.

[3]Indeed, we know from elementary counting (or dimension counting) arguments that there exist degree d polynomials in k variables which require algebraic circuits of size nearly $\binom{d+k}{k}$, which can be approximated by $d^{\Omega(k)}$ when k is much smaller than d, which is the range of parameters we work with in this paper.

PRGs of Shaltiel & Umans [SU05] and Umans [Uma03]: An alternative to the design-based PRGs in the boolean setting is the generator of Shaltiel and Umans [SU05], and a related follow up work of Umans [Uma03]. These generators are quite different from the design-based generators of Nisan and Wigderson [NW94] and, in particular, appear to be more *algebraic* in their definition and analysis. We refer the interested reader to the original papers [SU05, Uma03] for the formal definitions of these generators and further details.

The algebraic nature of these PRGs makes them good candidates for potential HSGs in the algebraic setting and, indeed, this work was partially motivated by this goal. However, as far as we understand, it remains unclear whether there is an easy adaptation of these PRGs which works for algebraic circuits. In particular, the hardness required for the analysis of the PRGs in [SU05, Uma03] appears to be inherently functional, i.e. they assume that it is hard to evaluate the polynomial over some finite field. In the context of algebraic complexity, the more natural notion of hardness is that it is hard to compute the polynomial syntactically as a formal polynomial via a small algebraic circuit.

1.2 Our Results

Our main result is the construction of a hitting-set generator that answers Question 1.4, for characteristic zero fields.

Definition 1.5 (The generator)**.** *For any k-variate polynomial $P(\mathbf{z})$, define the map $\mathsf{Gen}_P : \mathbb{F}^k \times \mathbb{F}^k \to \mathbb{F}^{n+1}$ as follows:*

$$\mathsf{Gen}_P(\mathbf{z},\mathbf{y}) = (\Delta_0(P)(\mathbf{z},\mathbf{y}), \Delta_1(P)(\mathbf{z},\mathbf{y}), \ldots, \Delta_n(P)(\mathbf{z},\mathbf{y}))\,,$$

where $\Delta_i(P)$ is the homogeneous degree i (in $\mathbf{y}$) component in the Taylor expansion of $P(\mathbf{z}+\mathbf{y})$, i.e.

$$\Delta_i(P)(\mathbf{z},\mathbf{y}) = \sum_{\mathbf{e}\in\mathbb{N}^k, |\mathbf{e}|_1 = i} \frac{\mathbf{y}^{\mathbf{e}}}{\mathbf{e}!} \cdot \frac{\partial P}{\partial \mathbf{z}^{\mathbf{e}}}. \qquad \textit{(here, } \mathbf{e}! := e_1! \cdots e_k!\textit{)} \qquad \lozenge$$

It is clear that the above definition is $d^{O(k)}$-explicit, where $d = \deg(P)$, as we can express P as a sum of d^k monomials and compute each component of $\mathsf{Gen}_P(\mathbf{z},\mathbf{y})$ with a small additional cost. Our main theorem states that the above map is indeed a generator if the polynomial $P(\mathbf{z})$ is hard enough.

Theorem 1.6 (Main theorem)**.** *Assume that the underlying field $\mathbb{F}$ has characteristic zero and let P be a k-variate polynomial of degree d. Suppose P cannot be computed by algebraic circuits of size $\tilde{s} = \left(s \cdot D \cdot d^3 \cdot n^{10k}\right)$ for parameters n, D, s. Then, for any $(n+1)$-variate polynomial $C(x_0, \ldots, x_n) \in \mathcal{C}(n+1, D, s)$, we have*

$$C \neq 0 \iff C \circ \mathsf{Gen}_P(\mathbf{z},\mathbf{y}) \neq 0.$$

This theorem answers Question 1.4 affirmatively. As alluded to in the introduction, we do not

know of prior constructions of HSGs with these properties.

In addition to being interesting on its own, Theorem 1.6 leads to the following result which shows that bootstrapping of hitting sets can be done. We recall the following simpler version of the Ore-DeMillo-Lipton-Schwartz-Zippel Lemma[4][Ore22, DL78, Sch80, Zip79].

Lemma (Folklore)**.** *Let f be a nonzero n-variate polynomial of* individual degree *at most d. Then, for any set $S \subseteq \mathbb{F}$ with $|S| > d$, there is a point $\mathbf{a} \in S^n$ such that $f(\mathbf{a}) \neq 0$.*

This implies that the class of k-variate polynomials of individual degree d has hitting sets of size $(d+1)^k$, irrespective of the circuit size. On the other hand, a simple counting argument shows that a random set of size $O(s^2)$ is a hitting set for the class of size s circuits, with high probability. The following shows that even improving on this bound by *one point* for small circuits would yield to a complete derandomization of PIT.

Theorem 1.7 (Bootstrapping hitting sets)**.** *Assume that the underlying field $\mathbb{F}$ has characteristic zero. Let $\delta > 0$ be any constant and let $k \in \mathbb{N}$ be a large enough constant. Suppose that, for all large enough s, there is an explicit hitting set of size $(s+1)^k - 1$ for the $\mathcal{C}(k, i\text{-deg: } s, s^\delta)$. Then, there is an explicit hitting set of size $s^{O((k/\delta)^2)}$ for $\mathcal{C}(s, s, s)$.*

The above theorem answers Question 1.3 in a strong sense over characteristic zero fields. Also, both Theorem 1.6 and Theorem 1.7 continues to hold, verbatim, for the notion of *border complexity* by modifying the hypothesis and the conclusion to work with the *border of small circuits*. However, it is crucial that we work with the class of algebraic circuits and not subclasses such as algebraic branching programs or algebraic formulas.

1.3 An overview of the proof

To show that the HSG in Definition 1.5 is indeed a hitting-set generator for $\mathcal{C}(n+1, D, s)$, we focus our attention on a purported nonzero polynomial $C(\mathbf{x})$, of circuit complexity s and degree D, that is not *fooled* by the generator, i.e. $C \circ \mathsf{Gen}_P$ is identically zero. We use this identity to reconstruct a small circuit for P which contradicts its hardness. This would imply that all polynomials in $\mathcal{C}(n+1, D, s)$ are fooled by the HSG.

In order to reconstruct a circuit for P from the circuit for C, we focus on the so-called *non-degenerate* case and address it in Lemma 3.2, which is our key technical lemma. Before discussing the main ideas in the proof of Lemma 3.2, we first discuss some of the details of the reduction to the non-degenerate case.

Reducing to the non-degenerate case : In the non-degenerate case we insist that, in addition to having $C \circ \mathsf{Gen}_P = 0$, we have $(\partial_{x_n} C) \circ \mathsf{Gen}_P \neq 0$; i.e. the derivative of C with respect to the *last* variable x_n is *fooled* by the generator.

[4]This version below is an immediate consequence of the Combinatorial Nullstellensatz [Alo99] but a simpler proof is to just use the fact that a univariate degree d polynomial can have at most d roots, one variable at a time (Lemma 2.2).

Given a nonzero circuit C such that $C \circ \mathsf{Gen}_P = 0$, and we wish to construct a circuit C' with the above stronger condition. We may assume without loss of generality that the circuit C is minimal in the sense that all circuits of the same size depending on fewer variables are fooled by the generator; in particular, C depends non-trivially on the last variable x_n.

We consider the circuit $\tilde{C}$ obtained by substituting the generator for all coordinates except x_n. Interpreting this as a univariate polynomial in x_n, with coefficients in $\mathbb{F}(\mathbf{z}, \mathbf{y})$,

$$\tilde{C}(x_n) = C(g_1, \ldots, g_{n-1}, x_n)$$

where $\mathsf{Gen}_P = (g_1, \ldots, g_{n-1}, g_n)$. The minimality of C allows us to argue $\tilde{C}(x_n)$ is a nonzero polynomial (for otherwise, $C(x_1, \ldots, x_{n-1}, a)$, for a random $a \in \mathbb{F}$ is also not fooled by the HSG, and this contradicts minimality). Using the Remainder theorem, this then implies that there must be some $0 \leq i \leq \deg_{x_n}(\tilde{C})$ such that

$$\left(\partial_{x_n^i} \tilde{C}\right)(g_n) = 0,$$
$$\left(\partial_{x_n^{i+1}} \tilde{C}\right)(g_n) \neq 0.$$

Hence, the circuit $C' = \partial_{x_n^i}(C)$ satisfies the *non-degeneracy* condition. Standard interpolation arguments shows that the circuit complexity of C' is at most $O(sD)$ and we work with this circuit [5].

The proof of Lemma 3.2 : The proof of the lemma can be viewed as a variant of the standard Newton Iteration (or Hensel lifting) based argument often used in the context of root finding, although there are some crucial differences. We iteratively construct the polynomial $P(\mathbf{z})$ one homogeneous component at a time (recall that $P(\mathbf{z})$ is a k-variate polynomial of degree d). In fact, our induction hypothesis needs to be a bit stronger than this. For our proof, we maintain the invariant that at the end of the i^{th} iteration, we have a multi-output circuit which computes all the partial derivatives of order at most n of all the homogeneous components of $P(\mathbf{z})$ of degree at most $i + n$. However, for this overview, we ignore this technicality and pretend that we are directly working with the homogeneous components of $P(\mathbf{z})$.

For the base case, we assume that we have access to all the homogeneous components of P of degree at most n, which are homogeneous polynomials of degree at most n on k variables and are trivially computable by a circuit of size at most $n^{O(k)}$, which is much smaller than $d^{\Omega(k)}$, the presumed hardness of P for $d \gg n$. Thus, we have n homogeneous components of $P(\mathbf{z})$, and the goal is to use them and the non-degeneracy assumption to reconstruct all of P. Let us assume that we have already computed $P_0, \ldots, P_i$, where P_j is the homogeneous component of $P(\mathbf{z})$ of degree equal to j. We now focus on recovering the homogeneous component P_{i+1} of degree equal to $i + 1$. Observe that $\Delta_n(P_{i+1})$ is a homogeneous (in $\mathbf{z}$) polynomial of degree $(i - n + 1)$. We show that given the non-degeneracy condition in the hypothesis of the lemma, there is a small circuit such

[5] This is also where the dependency of the hardness of P on the degree D appears. We suspect that this dependency can be removed.

that, modulo the ideal $\langle \mathbf{z} \rangle^{i-n+2}$, it computes $\Delta_n(P_{i+1})(\mathbf{z}, \mathbf{y})$. Since $\Delta_n(P_{i+1})(\mathbf{z}, \mathbf{y})$ is essentially a *generic* linear combination of n-th order derivatives of $P_{i+1}(\mathbf{z})$, it is not hard to show[6] that we can obtain a small circuit that outputs each of the n-th order partial derivatives of $P_{i+1}(\mathbf{z})$, modulo higher degree monomials. Then we would be able to reconstruct $P_{i+1}(\mathbf{z})$ mod $\langle \mathbf{z} \rangle^{i+2}$ via repeated applications of the Euler's differentiation formula for homogeneous polynomials:

Fact 1.8 (Euler's formula for differentiation of homogeneous polynomials)**.** *If $A(x_1, \ldots, x_k)$ is a homogeneous polynomial of degree t, then $\sum_{i=1}^{k} x_i \cdot \partial_{x_i} A = t \cdot A(x_1, \ldots, x_k)$.*

One crucial point in this entire reconstruction is that each step of the reconstruction only incurs an *additive* blow-up in size and hence can be repeated for polynomially many steps to recover each homogeneous part of P (Figure 1 in Section 3 contains a pictorial description of the inductive step).

However, we are still left with the task of getting rid of the higher order terms as we have only constructed a circuit computing P_{i+1} modulo the ideal $\langle \mathbf{z} \rangle^{i+2}$. We need to extract the lowest degree homogeneous parts from the outputs. The standard way to proceed here is to perform a *homogenisation* but this needs to be done carefully. Typically, extracting a certain homogeneous part incurs a multiplicative blow-up in size which is unaffordable in this setting as this needs to be performed for d steps! Fortunately, the structure of the circuit built has the property that each inductive step adds more gates atop the outputs of the previous steps and hence it suffices to *only homogenise* the newly added gates. This allows us to argue that each inductive step incurs only an additive blow-up of $d^2 \cdot n^{O(k)} \cdot sD$. By performing the reconstruction step to extract all homogeneous components of P, we can construct a small enough circuit for P, contradicting the hardness of P. That would complete the proof of Theorem 1.6.

Similarities with PRGs of Shaltiel and Umans [SU05, Uma03]: We remark that at a high level, our construction of the HSG was inspired by the constructions by Shaltiel and Umans [SU05, Uma03], although the precise form of our generator seems different from those in [SU05, Uma03]. We also note that the set up of induction we have in the proof of Lemma 3.2 is very similar to the set up used by Kopparty [Kop15] in the context of list decoding Multiplicity codes. More precisely, our induction is similar to what is used in constructing a power series expansion of a non-degenerate solution of the univariate Cauchy-Kovalevski differential equations, which are used in [Kop15]. The key difference is that although we work with a multivariate setting (and hence deal with a *partial* differential equation of high order and high degree), the iterative proof of Lemma 3.2 resembles the list decoding algorithm for univariate multiplicity codes in [Kop15] (which deals with an *ordinary* differential equation of high order and high degree). It appears to be of interest to understand this analogy further.

Relating Theorem 1.7 and results of Jansen and Santhanam [JS12]: The hypothesis of Theorem 1.7 bear some similarities with a result of Jansen and Santhanam [JS12] who showed that, for

[6] In particular, this part of the argument relies on the stronger hypothesis that we have access to each of the order n partial derivatives of $P_0, P_1, \ldots, P_i$.

the class of univariate polynomial of degree d that are computable by small circuits, if there is a hitting set H of size d (any set of size $d+1$ would have been sufficient) that *can be efficiently encoded by a small* TC^0 *circuit*, then Perm does not have polynomial sized *constant-free* algebraic circuits.

Our hypothesis is similar in the sense that it requires a saving of one from the trivial hitting set for the appropriate class, but we only need the standard notion of explicitness for the hitting set. The conclusion is also in terms of derandomizing PIT, and not a lower bound. It would be interesting to see if our hypothesis is also sufficient to obtain a similar conclusion as Jansen and Santhanam.

2 Notation and preliminaries

- Throughout the paper, we think of $\mathbb{F}$ as a field of characteristic zero (or large enough).
- We use the notation $\mathcal{C}(n,d,s)$ to denote the class of n-variate polynomials of degree bounded by d that are computable by circuits of size at most s. Similarly, $\mathcal{C}(n,\text{i-deg: } d,s)$ refers to the class of n-variate polynomials of *individual degree* bounded by d that are computable by circuits of size s.
- We use boldface letters such as $\mathbf{z}$ to denote tuples: $\mathbf{z} = (z_1, z_2, \ldots, z_k)$; in almost all instances, the length of the tuple will be clear from context. For an exponent vector $\mathbf{e}$, we shall use $\mathbf{z}^{\mathbf{e}}$ to denote the monomial $z_1^{e_1} \cdots z_k^{e_k}$. Let $|\mathbf{e}| := \sum e_i$.
- We use $\partial_{\mathbf{z}^{\mathbf{e}}}(P(\mathbf{z}))$ to denote the partial derivative $\frac{\partial^{|\mathbf{e}|}(P)}{\partial \mathbf{z}^{\mathbf{e}}}$.
- We use $\langle \mathbf{z} \rangle^i$ to denote the ideal in $\mathbb{F}[\mathbf{z}]$ generated by all degree i monomials in $\mathbf{z}$.
- We use $\mathcal{P}(k,d)$ to denote the class of k-variate polynomials of degree at most d.

2.1 PIT preliminaries

The following well-known lemma gives an exponential (in the number of variables) sized hitting set for low degree polynomials.

Lemma 2.1 ([Ore22, DL78, Sch80, Zip79])**.** *Let f be a nonzero n-variate polynomial of degree at most d. Then for any set $S \subseteq \mathbb{F}$ with $|S| > d$, there is a point $\mathbf{a} \in S^n$ such that $f(\mathbf{a}) \neq 0$.* □

The following is a variant for the case of polynomials of *individual degree* bounded by d.

Lemma 2.2. *Let f be a nonzero n-variate polynomial of* individual degree *at most d. Then, for any set $S \subseteq \mathbb{F}$ with $|S| > d$, there is a point $\mathbf{a} \in S^n$ such that $f(\mathbf{a}) \neq 0$.*

Proof. Let $S \subseteq \mathbb{F}$ with $|S| \geq d+1$. We may interpret f as a univariate polynomial in x_1 of degree at most d with coefficients from $\mathbb{F}(x_2, \ldots, x_n)$. Since a nonzero univariate polynomial of degree d can have at most d roots, there must be some $a_1 \in S$ such that $f(a_1, x_2, \ldots, x_n) \neq 0$. Repeating this argument for variables $x_2, \ldots, x_n$, the lemma follows. □

It is also known that existence of non-trivial hitting sets for a class $\mathcal{C}$ can be used to construct hard polynomials. We state the version that corresponds to the individual degree as opposed to total degree.

Theorem 2.3 (Heintz and Schnorr [HS80], Agrawal [Agr05]). *Let $H(n, i\text{-deg: } d, s)$ be an explicit hitting set for the class $\mathcal{C}(n, i\text{-deg: } d, s)$. Then, for every $k \leq n$ and d' such that $d' \leq d$ and $(d'+1)^k > |H(n, i\text{-deg: } d, s)|$, there is a nonzero polynomial on k variables and individual degree d' that vanishes on the hitting set $H(n, i\text{-deg: } d, s)$, and hence cannot be computed by a circuit of size s.*

Finally, we need the following notion of *interpolating sets* for a class of polynomials.

Definition 2.4 (Interpolating sets for $\mathcal{P}(k,d)$). *Let $M_{k,d}$ denote the number of k-variate monomials of degree at most d. That is, $M_{k,d} = \binom{k+d}{d}$.*

A set of points $\mathbf{a}_1, \ldots, \mathbf{a}_r \in \mathbb{F}^k$ is said to be an interpolating set for $\mathcal{P}(k,d)$ *if the vectors*

$$\left\{ \left(\mathbf{a}_i^{\mathbf{e}} \ : \ \mathbf{e} \in \mathbb{Z}^k_{\geq 0} \ , \ |\mathbf{e}| \leq d \right) \ : \ i \in [r] \right\} \subset \mathbb{F}^{M_{k,d}}$$

form a spanning set for $\mathbb{F}^{M_{k,d}}$.
Equivalently, there exists field constants $\beta_1, \ldots, \beta_r$ such that for every $f(\mathbf{z}) \in \mathcal{P}(k,d)$ and every $\mathbf{e} \in \mathbb{Z}^k_{\geq 0}$ with $|\mathbf{e}| \leq d$, we have that

$$\mathrm{coeff}_{\mathbf{z}^{\mathbf{e}}}(f) = \sum_{i=1}^{r} \beta_i f(\mathbf{a}_i). \qquad \Diamond$$

A canonical example of an interpolating set for $\mathcal{P}(k,d)$ is $S^k = \{(s_1, \ldots, s_k) \ : \ s_i \in S \ \forall i\}$ where $S \subseteq \mathbb{F}$ is a set of at least $(d+1)$ distinct field elements. The following well-known proposition says that a random set of points, of the appropriate size, is an interpolating set for $\mathcal{P}(k,d)$ with high probability if the field $\mathbb{F}$ is large enough. [7]

Proposition 2.5 (Random sets are interpolating sets). *For any d, k, if $\mathbb{F}$ is large enough, then a random set of size $\binom{k+d}{d}$ is an interpolating set for $\mathcal{P}(k,d)$ with probability $1 - o(1)$.* □

2.2 Homogenisation

Definition 2.6 (Homogeneous circuits). *A circuit C is said to be* homogeneous *if every gate of the circuit computes a homogeneous polynomial.*

For a non-homogeneous polynomial f, we shall say that a homogeneous (multi-output) circuit C *computes f if the outputs of C are the homogeneous parts of f.* $\Diamond$

Lemma 2.7 (Strassen's homogenisation). *Let C be a circuit of size s computing a homogeneous polynomial of degree d. Then, there is a* homogeneous circuit C' *of size at most $O(sd^2)$ computing the same polynomial.* □

[7] For infinite fields, we pick the points uniformly at random from a large enough, but finite grid.

Lemma 2.8 (Partial homogenisation)**.** *Let C be a multi-output homogeneous circuit of size s, with m outputs computing homogeneous polynomials $f_1, \ldots, f_m$. Suppose C' is a multi-output, m-input circuit of size s'. Then, there is a* homogeneous circuit *D of size at most $s + O(s' \cdot d^2)$ computing $C' \circ C = C'(f_1, \ldots, f_m)$, where d is the maximum degree of the outputs of $C'(f_1, \ldots, f_m)$.*

Proof. We follow the standard homogenisation procedure, but applied only to the circuit C' — we replace each gate $g \in C'$ by copies $g_0, \ldots, g_d$ and add the following connections:

$$g_a = \begin{cases} h_a^{(L)} + h_a^{(R)} & \text{if } g = h^{(L)} + h^{(R)} \\ \sum_{b=0}^{a} h_b^{(L)} \times h_{a-b}^{(R)} & \text{if } g = h^{(L)} \times h^{(R)} \end{cases}$$

and any leaf ℓ of C' labeled with the i-th variable is renamed as $\ell_{\deg(f_i)}$. Since each of the f_i is a homogeneous polynomial, it is immediate to see that the above transformation yields a homogeneous circuit for $C' \circ C$ of size at most $s + O(s'd^2)$. □

2.3 The Generator

For a k-variate polynomial P, let $\Delta_i(P)(\mathbf{z}, \mathbf{y}) \in \mathbb{F}[\mathbf{z}, \mathbf{y}]$ be defined as

$$\Delta_i(P) = \sum_{\mathbf{e}:|\mathbf{e}|=n} \left(\frac{\mathbf{y}^{\mathbf{e}}}{\mathbf{e}!} \right) \cdot \partial_{\mathbf{z}^{\mathbf{e}}}(P)$$

where $\mathbf{e}! = e_1! \cdots e_k!$. The generator with respect to P is defined as follows:

$$\mathsf{Gen}_P(\mathbf{z}, \mathbf{y}) = \left(\Delta_0(P)(\mathbf{z}, \mathbf{y}), \ldots, \Delta_n(P)(\mathbf{z}, \mathbf{y}) \right).$$

The following are some simple observations about the operator Δ.

Observation 2.9. *For any $\mathbf{a} \in \mathbb{F}^k$, if $P'(\mathbf{z}) = P(\mathbf{z} + \mathbf{a})$, then for all i we have $\Delta_i(P')(\mathbf{z}, \mathbf{y}) = \Delta_i(P)(\mathbf{z} + \mathbf{a}, \mathbf{y})$. Hence, $\mathsf{Gen}_{P'}(\mathbf{z}, \mathbf{y}) = \mathsf{Gen}_P(\mathbf{z} + \mathbf{a}, \mathbf{y})$.*

Observation 2.10. *Let $P(\mathbf{z})$ and $Q(\mathbf{z})$ be polynomials such that $P = Q \bmod \langle \mathbf{z} \rangle^j$. Then, for any $i \leq j$, we have $\Delta_i(P) = \Delta_i(Q) \bmod \langle \mathbf{z} \rangle^{j-i}$.*

3 The Main Theorem

We start by recalling the main theorem.

Theorem 1.6 (Main theorem)**.** *Assume that the underlying field $\mathbb{F}$ has characteristic zero and let P be a k-variate polynomial of degree d. Suppose P cannot be computed by algebraic circuits of size $\tilde{s} = \left(s \cdot D \cdot d^3 \cdot n^{10k}\right)$ for parameters n, D, s. Then, for any $(n+1)$-variate polynomial $C(x_0, \ldots, x_n) \in \mathcal{C}(n+1, D, s)$, we have*

$$C \neq 0 \iff C \circ \mathsf{Gen}_P(\mathbf{z}, \mathbf{y}) \neq 0.$$

The rest of this section would be devoted to the proof of this theorem.

Let us assume the contrary. That is, there is a circuit $C(\mathbf{x})$ of size s and degree D such that $C \neq 0$ but $C \circ \mathsf{Gen}_P(\mathbf{z}, \mathbf{y}) = 0$. We shall assume, without loss of generality, that C depends non-trivially on the variable x_n and that no circuit $C'(\mathbf{x})$ of size s and degree D with C' depending on fewer variables satisfy $C' \neq 0$ and $C' \circ \mathsf{Gen}_P(\mathbf{z}, \mathbf{y}) = 0$.

The proof will proceed by inductively building a circuit that computes each homogeneous part of P but we would need the following preprocessing step.

Preprocessing the circuit: Let $C(x_0, \ldots, x_n)$ be the minimal (in terms of number of variables) size s circuit that is *not* fooled by $\mathsf{Gen}_P(\mathbf{z}, \mathbf{y})$. That is, $C \circ \mathsf{Gen}_P(\mathbf{z}, \mathbf{y}) = 0$.

Claim 3.1. *There is some $i \geq 0$ such that*

$$\begin{aligned} \partial_{x_n^i}(C) \circ \mathsf{Gen}_P(\mathbf{z}, \mathbf{y}) &= 0, \\ \partial_{x_n^{i+1}}(C) \circ \mathsf{Gen}_P(\mathbf{z}, \mathbf{y}) &\neq 0. \end{aligned}$$

Proof. Let $\mathsf{Gen}_P(\mathbf{z}, \mathbf{y}) = (g_0, \ldots, g_n)$. Consider the polynomial $C(g_0, \ldots, g_{n-1}, x_n)$ (which is the application of the generator Gen_P to all coordinates except one) as an element $\hat{C}(x_n) \in \mathbb{F}(\mathbf{y}, \mathbf{z})[x_n]$.

Case 1: $(\hat{C} = 0)$

In this case, let $a \in \mathbb{F}$ such that $C(x_0, \ldots, x_{n-1}, a) \neq 0$. Then, $C(x_0, \ldots, x_{n-1}, a)$ is also a nonzero polynomial computable by a size s circuit, depending on fewer than $n+1$ variables, that also vanishes on $\mathsf{Gen}_P(\mathbf{z}, \mathbf{y})$. This contradicts the minimality of C.

Case 2: $(\hat{C} \neq 0)$

Let $r = \deg_{x_n}(\hat{C})$. Note that if $(\partial_{x_n^i}(\hat{C}))(g_n) = 0$ for all $0 \leq i \leq t$, then $(x_n - g_n)^{t+1}$ divides $\hat{C}$. Since $\deg_{x_n}(\hat{C})$ is r, the largest e such that $(x_n - g_n)^e$ divides $\hat{C}$ is at most r. Hence, there must be some $t \leq r-1$ such that $\partial_{x_n^t}(\hat{C})(g_n) \neq 0$. Since $\hat{C}(g) = C \circ \mathsf{Gen}_P(\mathbf{z}, \mathbf{y}) = 0$ and $\partial_{x_n^t}(\hat{C})(g_n) \neq 0$, there must be an intermediate derivative where a switch from zero to nonzero occurs. Hence, there must be some $i < r$ such that

$$\begin{aligned} \partial_{x_n^i}(C) \circ \mathsf{Gen}_P(\mathbf{z}, \mathbf{y}) &= \partial_{x_n^i}(\hat{C})(g_n) = 0, \\ \partial_{x_n^{i+1}}(C) \circ \mathsf{Gen}_P(\mathbf{z}, \mathbf{y}) &= \partial_{x_n^{i+1}}(\hat{C})(g_n) \neq 0. \end{aligned}$$

□

Let $C' = \partial_{x_n^i}(C)$. In what follows, we will work with C' instead of C. By interpolation, its size $s' \leq s \cdot D$ (where, recall, $D \geq \deg(C)$) and we now have

$$\begin{aligned} C' \circ \mathsf{Gen}_P(\mathbf{z}, \mathbf{y}) &= 0, \\ \partial_{x_n}(C') \circ \mathsf{Gen}_P(\mathbf{z}, \mathbf{y}) &\neq 0. \end{aligned}$$

Without loss of generality (by translating $\mathbf{z}$ if necessary, via Observation 2.9), we may assume that

$$\left(\partial_{x_n}(C') \circ \mathsf{Gen}_P(\mathbf{0}, \mathbf{y})\right) \;=:\; \Psi(\mathbf{y}) \neq 0.$$

Let $P = P_0 + P_1 + \cdots + P_d$ be the decomposition into homogeneous parts, with P_i being the homogeneous part of degree i, and let $P_{\leq r} := \sum_{i \leq r} P_i$.

The reconstruction: We now proceed to describe the set up of the inductive reconstruction of a *small* circuit for P. The induction is on a parameter j which takes values from 0 up to $d - n$. At the end of the j^{th} step, we would have a circuit which computes all partial derivatives of order at most n of homogeneous components of P of degree at most $j + n$. We now describe the steps of the induction argument more formally.

Base case $(j = 0)$ **:** Each $\partial_{\mathbf{z}^\mathbf{e}} P_\ell$ for $|\mathbf{e}| \leq n$ and $\ell \leq n$ can be explicitly written as a sum of at most $N := \binom{n+k}{k}$ monomials. Hence, there is a circuit B_0 of size $s_0 = N^2$ that computes $\{\partial_{\mathbf{z}^\mathbf{e}}(P_\ell) \;:\; 0 \leq \ell \leq n\;,\; |\mathbf{e}| \leq n\}$.

Induction hypothesis: There is a circuit $B_{j-1}(\mathbf{z})$ of size at most s_{j-1}, with $N(n + j - 1)$ outputs that computes $\partial_{\mathbf{z}^\mathbf{e}} P_\ell$ for each $\mathbf{e}$ with $|\mathbf{e}| \leq n$ and $\ell \leq n + j - 1$.

Induction step: To construct a circuit $B_j(\mathbf{z})$ of size at most s_j (to be defined shortly) that computes $\partial_{\mathbf{z}^\mathbf{e}} P_\ell$ for each $\mathbf{e}$ with $|\mathbf{e}| \leq n$ and $\ell \leq n + j$.

Recall $N = \binom{n+k}{n}$, the number of k-variate, degree n monomials. Recall that $\Psi(\mathbf{a}) = (\partial_{x_n}(C') \circ \mathsf{Gen}_P(\mathbf{0}, \mathbf{y}))$. We shall say that $\mathbf{a} \in \mathbb{F}^k$ is "good" if $\Psi(\mathbf{a}) \neq 0$. Since $\mathbb{F}$ is large enough, by Proposition 2.5 and Lemma 2.1, a random set $\{\mathbf{a}_1, \ldots, \mathbf{a}_N\} \subset \mathbb{F}^n$ is a set of "good" points and also an interpolating set for $\mathcal{P}(k, n)$ with probability $1 - o(1)$. Let $\Gamma_{j-1,\mathbf{a}}$ be defined as

$$\Gamma_{j-1,\mathbf{a}} := (\Delta_0(P_{\leq n+j-1})(\mathbf{z}, \mathbf{a}), \ldots, \Delta_n(P_{\leq n+j-1})(\mathbf{z}, \mathbf{a})).$$

Lemma 3.2. *Let* $\mathbf{a} \in \mathbb{F}^k$ *be such that* $0 \neq \Psi(\mathbf{a}) = (\partial_{x_n} C') \circ \mathsf{Gen}_P(\mathbf{0}, \mathbf{a})$. *Then,*

$$\left(\frac{-1}{\Psi(\mathbf{a})}\right) \cdot C'(\Gamma_{j-1,\mathbf{a}}) = \Delta_n(P_{n+j})(\mathbf{z}, \mathbf{a}) \bmod \langle \mathbf{z} \rangle^{j+1}.$$

We will defer the proof of this lemma to the end of the section and finish the rest of the proof.

We can begin with the circuit $B_{j-1}(\mathbf{z})$ that computes every $\partial_{\mathbf{z}^\mathbf{e}}(P_\ell)$ for $|\mathbf{e}| \leq n$ and $\ell \leq n + j - 1$. By taking suitable linear combinations of the output gates, we can create a new circuit B, of size at most $s_{j-1} + N^5$, that computes $\left\{\Gamma_{j-1,\mathbf{a}_t} \;:\; t \in [N]\right\}$. Using Lemma 3.2 for each $\mathbf{a}_i$, we then obtain a circuit of size $s_{j-1} + N^5 + s' \cdot N$ that computes $\left\{\Delta_n(P_{n+j})(\mathbf{z}, \mathbf{a}_t) \;:\; t \in [N]\right\}$ modulo the ideal $\langle \mathbf{z} \rangle^{j+1}$.

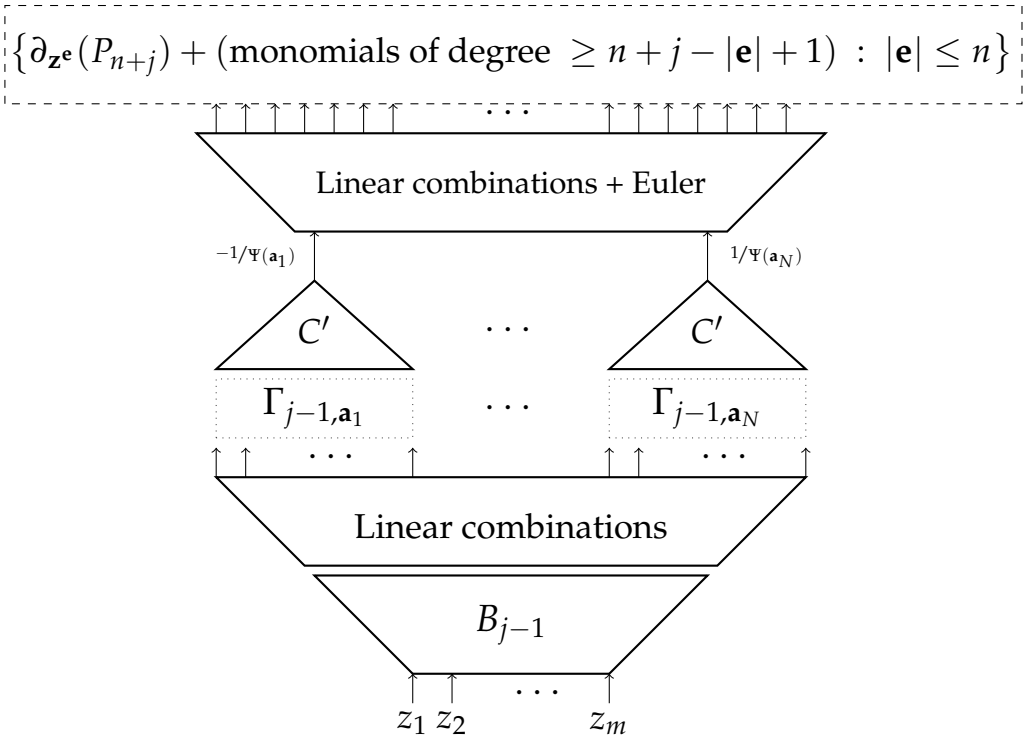

Figure 1: Pictorial representation of B'_j

By definition, $\Delta_n(P_{n+j})(\mathbf{z}, \mathbf{a})$ is a suitable linear combination of the n-th order partial derivatives of $P_{n+j}(\mathbf{z})$. As $\{\mathbf{a}_1, \ldots, \mathbf{a}_N\}$ was chosen to be an interpolating set, each $\partial_{\mathbf{z}^\mathbf{e}}(P_{n+j})$ with $|\mathbf{e}| = n$ can be written as a suitable linear combination of $\{\Delta_n(P_{n+j})(\mathbf{z}, \mathbf{a}_t) \ : \ t \in [N]\}$. Furthermore, since P_{n+j} is a homogeneous polynomial, we can also compute all its lower order derivatives via repeated applications of Euler's formula (Fact 1.8). Overall, combined with the outputs of $B_{j-1}(\mathbf{z})$, we have a circuit $B'_j(\mathbf{z})$ (shown in Figure 1) of size $s_{j-1} + N^{10} + s'N$ that computes

$$\{\partial_{\mathbf{z}^\mathbf{e}}(P_\ell) \ : \ |\mathbf{e}| \leq n \ , \ \ell \leq n+j-1\} \ \cup \ \{\partial_{\mathbf{z}^\mathbf{e}}(\tilde{P}_{n+j}) \ : \ |\mathbf{e}| \leq n\} \, ,$$

where $\partial_{\mathbf{z}^\mathbf{e}}(\tilde{P}_{n+j}) \mod \langle \mathbf{z} \rangle^{n+j-|\mathbf{e}|+1} \equiv \partial_{\mathbf{z}^\mathbf{e}}(P_{n+j})$ for every $|\mathbf{e}| \leq n$. At this point, the only task left to do is extracting the lowest degree homogeneous components of these outputs.

From Figure 1, the circuit B'_j is a composition of a circuit of size $N^{10} + s'N$ over the homogeneous circuit B_{j-1} of size s_{j-1}. By Lemma 2.8, we extract the lowest degree homogeneous parts of the outputs of B'_j by constructing an equivalent *homogeneous* circuit of size at most $s_{j-1} + ((N^{10} + s'N) \cdot d^2)$ that computes

$$\{\partial_{\mathbf{z}^\mathbf{e}}(P_\ell) \ : \ |\mathbf{e}| \leq n \ , \ \ell \leq n+j\} \, .$$

This completes the induction step.

Unraveling the induction for $d - n$ steps, we eventually obtain a circuit of size at most $s_{d-n} = O(s' \cdot d^3 \cdot N^{10}) = O(s \cdot D \cdot d^3 \cdot n^{O(k)})$ that computes all the partial derivatives of order at most n of $P_0, \ldots, P_d$, which in particular includes $P_0, \ldots, P_d$ that can be summed to produce a circuit for P of size $O(s \cdot D \cdot d^3 \cdot n^{O(k)})$. However, this contradicts the hardness assumption of P. Hence, it must be the case that $C \circ \mathsf{Gen}_P(\mathbf{z}, \mathbf{y}) \neq 0$. This completes the proof of the main theorem barring the proof of Lemma 3.2; we address this next. □(Theorem 1.6)

3.1 Proof of Lemma 3.2

We are given $\Gamma_{j-1,\mathbf{a}} = (\Delta_0(P_{\leq n+j-1})(\mathbf{z}, \mathbf{a}), \ldots, \Delta_n(P_{\leq n+j-1})(\mathbf{z}, \mathbf{a}))$. From the assumption on C', we have

$$\begin{aligned} & 0 = C'(\Delta_0(P)(\mathbf{z}, \mathbf{a}), \ldots, \Delta_n(P)(\mathbf{z}, \mathbf{a})) \\ \implies & 0 = C'(\Delta_0(P)(\mathbf{z}, \mathbf{a}), \ldots, \Delta_n(P)(\mathbf{z}, \mathbf{a})) \bmod \langle \mathbf{z} \rangle^{j+1}. \end{aligned}$$

By Observation 2.10, we have that $\Delta_i(P)(\mathbf{z}, \mathbf{a}) = \Delta_i(P_{\leq n+j-1})(\mathbf{z}, \mathbf{a}) \bmod \langle \mathbf{z} \rangle^{j+1}$ for all $i \leq n-1$, and $\Delta_n(P)(\mathbf{z}, \mathbf{a}) = \Delta_n(P_{\leq n+j-1})(\mathbf{z}, \mathbf{a}) + \Delta_n(P_{n+j})(\mathbf{z}, \mathbf{a}) \bmod \langle \mathbf{z} \rangle^{j+1}$. For the sake of brevity, let $R_i = \Delta_i(P_{\leq n+j-1})(\mathbf{z}, \mathbf{a})$ for $0 \leq i \leq n$ and $A = \Delta_n(P_{n+j})(\mathbf{z}, \mathbf{a})$. Therefore,

$$0 = C'(R_0, R_1, \ldots, R_{n-1}, R_n + A) \bmod \langle \mathbf{z} \rangle^{j+1}.$$

We now expand the above expression as a *univariate in* A (or in other words, perform a Taylor expansion of the polynomial C' around the point $(R_0, R_1, \ldots, R_n)$). Let $d' = \deg_{x_n}(C')$. Then,

$$0 = C'(R_0, \ldots, R_n) + \sum_{i=1}^{d'} A^i \cdot \left(\frac{\partial_{x_n^i}(C')(R_0, \ldots, R_n)}{i!} \right) \bmod \langle \mathbf{z} \rangle^{j+1}.$$

Moreover, since $A = \Delta_n(P_{n+j})(\mathbf{z}, \mathbf{a})$ is a homogeneous polynomial of degree $j \geq 1$, we have $A^2 = 0 \bmod \langle \mathbf{z} \rangle^{j+1}$. Therefore,

$$\begin{aligned} 0 &= C'(R_0, \ldots, R_n) + \sum_{i=1}^{d'} A^i \cdot \left(\frac{\partial_{x_n^i}(C')(R_0, \ldots, R_n)}{i!} \right) \bmod \langle \mathbf{z} \rangle^{j+1} \\ &= C'(R_0, \ldots, R_n) + A \cdot \left(\partial_{x_n}(C')(R_0, \ldots, R_n) \right) \bmod \langle \mathbf{z} \rangle^{j+1} \\ &= C'(R_0, \ldots, R_n) + A \cdot \alpha \bmod \langle \mathbf{z} \rangle^{j+1} \end{aligned}$$

where $\alpha = \partial_{x_n}(C')(R_0, \ldots, R_n)(\mathbf{0})$, the constant term of $\partial_{x_n}(C')(R_0, \ldots, R_n)(\mathbf{z})$. Note

$$\begin{aligned} \alpha = \partial_{x_n}(C')(R_0, \ldots, R_n)(\mathbf{0}) &= \partial_{x_n}(C')(\Delta_0(P_{\leq n+j-1})(\mathbf{0}, \mathbf{a}), \ldots, \Delta_n(P_{\leq n+j-1})(\mathbf{0}, \mathbf{a})) \\ &= \partial_{x_n}(C')(\Delta_0(P_{\leq n+j-1})(\mathbf{z}, \mathbf{a}), \ldots, \Delta_n(P_{\leq n+j-1})(\mathbf{z}, \mathbf{a}))(\mathbf{0}) \\ &= \partial_{x_n}(C')(\Delta_0(P)(\mathbf{z}, \mathbf{a}), \ldots, \Delta_n(P)(\mathbf{z}, \mathbf{a}))(\mathbf{0}) \\ &= \left(\partial_{x_n}(C') \circ \mathsf{Gen}(P, \mathbf{a}) \right)(\mathbf{0}) \end{aligned}$$

$$= \Psi(\mathbf{a}) \neq 0.$$

Combining this with the previous equation, we get

$$0 = C'(R_0, \ldots, R_n) + A \cdot \Psi(\mathbf{a}) \bmod \langle \mathbf{z} \rangle^{j+1}$$
$$\implies A = \Delta_n(P_{n+j})(\mathbf{z}, \mathbf{a}) = \left(\frac{-1}{\Psi(\mathbf{a})}\right) \cdot C'(R_0, \ldots, R_n) \bmod \langle \mathbf{z} \rangle^{j+1}.$$

□(Lemma 3.2)

3.2 Derandomization from hard, constant-variate families

Theorem 3.3. *Let k be a large enough positive integer, and let $\delta > 0$ be a constant. Suppose $\{P_{k,d}\}_{d \in \mathbb{N}}$ is a family[8] of explicit k-variate polynomials such that $\deg(P_{k,d}) = d$ and $P_{k,d}$ requires circuits of size at least d^δ. Then, there are explicit hitting sets of size $s^{O(k^2/\delta^2)}$ for the class of s-variate, degree s polynomials that can be computed by circuits of size s.*

Proof. We wish to construct a generator to fool the class of s-variate, degree s polynomials computable by circuits of size s. Let $t \geq \lceil \frac{8}{\delta} \rceil$ and let $d > s^{(10kt+2)\cdot t}$.

To the polynomial $P_{k,d}(z_1, \ldots, z_k)$ in the family, we associate the natural kt-variate polynomial $Q_{k,d,t}\left(\left(z_{i,j} \,:\, i \in [k]\,,\; j \in [t]\right)\right)$, of degree at most $d' = (kt) \cdot d^{1/t}$, such that $Q_{k,d,t}(z_{1,1}, \ldots, z_{k,t})$ under the substitution

$$z_{i,j} \to z_i^{d^{(j-1)/t}}$$

yields the polynomial $P_{k,d}(z_1, \ldots, z_k)$. This is achieved by replacing each monomial $z_1^{e_1} \cdots z_k^{e_k}$ in $P_{k,d}$ by $\mathbf{z_1}_{,*}{}^{(\!|e_1|\!)} \cdots \mathbf{z_k}_{,*}{}^{(\!|e_k|\!)}$ where $(\!|e_i|\!)$ is the tuple corresponding to the integer e_i expressed in base $d^{1/t}$.

Note that if $Q_{k,d,t}$ has a circuit of size $d'^4 < d^{\delta/2+o(1)}$ (recall that k and t are both constants), then (by employing repeated squaring to perform the substitution described above) $P_{k,d}$ has a circuit of size at most

$$d^{\delta/2+o(1)} + O(kt \log d) \ll d^\delta,$$

which is a contradiction to the hardness of $P_{k,d}$. Hence we have that $Q_{k,d,t}$ is a k-variate polynomial of degree d' that requires circuits of size at least d'^4. Since, by the choice of d, we have $d' > d^{1/t} > s^{10kt+2}$, we have

$$d'^4 > s^{10kt} \cdot s^2 \cdot d'^3.$$

[8]We are assuming that the polynomial family contains a degree d polynomial, $P_{k,d}$, for every positive integer d. For the purposes of this theorem, it suffices to assume that the family is *sufficiently often* in the following sense: There are absolute constants a, b such that for any $t \in \mathbb{N}$, there is some $P_{k,d}$ in the family such that $t^a \leq d \leq t^b$.

By Theorem 1.6, we hence have that $\mathsf{Gen}_{Q_{k,d,t}}$ is a hitting-set generator for the class $\mathcal{C}(s,s,s)$.

Therefore, from Lemma 2.1, this yields an explicit hitting set of size $(sd'+1)^{2kt} = s^{O(k^2/\delta^2)}$. □

Remark. The work of Kabanets and Impagliazzo [KI04] shows that if there is an explicit family of *multilinear* polynomials $\{Q_m\}$ that requires circuits of size $2^{\Omega(m)}$, then we can construct explicit hitting sets of $s^{O(\log s)}$ size for $\mathcal{C}(s,s,s)$.

From the above proof, it is easy to see that if $P_{k,d}$ requires circuits of size d^δ, then the above proof shows that the polynomial $Q_{k,d,t}$ in the above proof, for $t = \log d$, is an $m = O(k \log d)$-variate multilinear polynomial that requires circuits of size roughly $d^\delta = \exp(\Omega(m))$ to compute it. Therefore, the hypothesis of Theorem 1.7 implies the hypothesis necessary for the result of Kabanets and Impagliazzo.

However, the other direction seems unclear as it is conceivable that the circuit complexity of $P_{k,d}$ is significantly smaller than the complexity of $Q_{k,d,t}$. Hence, Theorem 1.7, when compared with the hardness-randomness trade-off of Kabanets and Impagliazzo[KI04], can be interpreted as a potentially stronger hypothesis leading to the stronger conclusion of a complete derandomization of PIT. ◊

3.3 Application to bootstrapping phenomenon for hitting sets

We now use Theorem 3.3 to prove the following theorem about bootstrapping hitting sets for algebraic circuits. The following theorem shows that any hitting set for $\mathcal{C}(k, \text{i-deg: } s, s)$, that saves *even one point* from the trivial hitting set of size $(s+1)^k$ guaranteed by Lemma 2.2, is sufficient to completely derandomize PIT.

Theorem 1.7 (Bootstrapping hitting sets)**.** *Assume that the underlying field $\mathbb{F}$ has characteristic zero. Let $\delta > 0$ be any constant and let $k \in \mathbb{N}$ be a large enough constant. Suppose that, for all large enough s, there is an explicit hitting set of size $(s+1)^k - 1$ for the $\mathcal{C}(k, \text{i-deg: } s, s^\delta)$. Then, there is an explicit hitting set of size $s^{O((k/\delta)^2)}$ for $\mathcal{C}(s,s,s)$.*

Proof. Consider an arbitrary, large enough s and let H_s be the hitting set for $\mathcal{C}(k, \text{i-deg: } s, s^\delta)$; the hypothesis guarantees that $|H_s| \leq (s+1)^k - 1 < (s+1)^k$. From Theorem 2.3, we can then obtain a polynomial $P_s(z_1, \ldots, z_k)$ of individual degree at most s that cannot be computed by circuits of size s^δ. Expressing this in terms of its total degree $d \leq ks$, we get that P_s is a degree d polynomial that requires circuits of size more than

$$\left(\frac{d}{k}\right)^\delta \gg d^{\delta/2}.$$

Hence, $\{P_s\}$ is an explicit family of k-variate, degree-d polynomials that require circuits of size $d^{\delta/2}$. Therefore Theorem 3.3 yields an explicit $s^{O((k/\delta)^2)}$-sized hitting set for the class of s-variate, degree s polynomials that can be computed by size s circuits. □

4 Open Problems

We end with some open problems.

- The construction in this paper takes a $d^{\Omega(k)}$-hard, k-variate polynomial of degree d to construct a generator for $\mathcal{C}(s,s,s)$ with $s \lessapprox d$. A k-variate degree d polynomial has $d^{\Theta(k)}$ coefficients and if P is $d^{\Omega(k)}$-hard then, intuitively, we have "enough entropy" to be able to fool $\mathcal{C}(s,s,s)$ with $s \approx d^{O(k)}$.

  On the other hand, the HSG of Kabanets and Impagliazzo [KI04], in certain ranges of parameters, is "lossless" in the above sense it uses a k-variate, multilinear, hard polynomial which has $\tilde{s} = 2^k$ coefficients to build a generator for $\mathcal{C}(s,s,s)$ with $s = \tilde{s}^{\Omega(1)}$.

  In principle, it seems conceivable that there is a construction that, starting from a $d^{\Omega(k)}$-hard k-variate degree d polynomial, gives an HSG for $\mathcal{C}(s,s,s)$ for $s = d^{\Omega(k)}$. Constructing such an explicit HSG is perhaps the most natural open question here.

- The construction of the HSG in this paper needs the characteristic of the field to be large enough or zero. Constructing a HSG with similar properties (seed length, stretch, running time, degree) over fields of small positive characteristic would be quite interesting.

- In the current statement of Theorem 1.6, the hardness required for P for the HSG to fool circuits of size s, depends also on the degree of this circuit. We suspect that this dependence on the degree can be avoided, if the hardness assumption is in terms of *border-complexity*; in that case, this HSG should fool all circuits of small size regardless of their degree.

- Lastly, it would be interesting to understand if this new HSG and the ideas in its analysis have any other applications in complexity theory.

Acknowledgements

We thank Marco Carmosino, Chi-Ning Chou, Nutan Limaye, Rahul Santhanam, Srikanth Srinivasan and Anamay Tengse for insightful conversations at various stages of this work. We are particularly grateful to Srikanth Srinivasan who pointed out an observation that strengthened our prior version of Theorem 3.3, and eliminated the use of border complexity entirely. We also thank Madhu Sudan for many insightful discussions and much encouragement; in particular for patiently sitting through a presentation of a preliminary version of the proof of Lemma 3.2.

Mrinal is also thankful to Swastik Kopparty for many helpful discussions on Nisan-Wigderson generator and list decoding algorithms for multiplicity codes while he was a PhD student at Rutgers. Swastik's interest in a purely algebraic solution for algebraic hardness randomness tradeoffs had a non-trivial role in motivating this work.

References

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

[AGS19] Manindra Agrawal, Sumanta Ghosh, and Nitin Saxena. Bootstrapping Variables in Algebraic Circuits. *Proceedings of the National Academy of Sciences of the United States of America*, 116(17):8107–8118, 2019. Preliminary version in the *50th Annual ACM Symposium on Theory of Computing (STOC 2018)*.

[Alo99] Noga Alon. Combinatorial Nullstellensatz. *Combinatorics, Probability and Computing*, 8(1-2):7–29, 1999.

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

[GKSS19] Zeyu Guo, Mrinal Kumar, Ramprasad Saptharishi, and Noam Solomon. Derandomization from Algebraic Hardness: Treading the Borders. *To appear in Proceedings of the 60th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2019)*, 2019. Online version: `https://mrinalkr.bitbucket.io/papers/newprgconf.pdf`.

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

[IW97] Russell Impagliazzo and Avi Wigderson. P = BPP if E Requires Exponential Circuits: Derandomizing the XOR Lemma. In *Proceedings of the 29th Annual ACM Symposium on Theory of Computing (STOC 1997)*, pages 220–229, 1997.

[JS12] Maurice J. Jansen and Rahul Santhanam. Marginal hitting sets imply super-polynomial lower bounds for permanent. In *Innovations in Theoretical Computer Science 2012, Cambridge, MA, USA, January 8-10, 2012*, pages 496–506. ACM, 2012. `eccc:TR11-133`.

[Kal89] Erich Kaltofen. Factorization of Polynomials Given by Straight-Line Programs. *Advances in Computing Research*, 5:375–412, 1989.

[KI04] Valentine Kabanets and Russell Impagliazzo. Derandomizing Polynomial Identity Tests Means Proving Circuit Lower Bounds. *Computational Complexity*, 13(1-2):1–46, 2004. Preliminary version in the *35th Annual ACM Symposium on Theory of Computing (STOC 2003)*.

[Kop15] Swastik Kopparty. List-Decoding Multiplicity Codes. *Theory of Computing*, 11(5):149–182, 2015.

[KST19] Mrinal Kumar, Ramprasad Saptharishi, and Anamay Tengse. Near-optimal Bootstrapping of Hitting Sets for Algebraic Circuits. In *Proceedings of the 30th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2019)*, pages 639–646, 2019.

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

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

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

[SU05] Ronen Shaltiel and Christopher Umans. Simple Extractors for All Min-entropies and a New Pseudorandom Generator. *Journal of the ACM*, 52(2):172–216, 2005.

[Uma03] Christopher Umans. Pseudo-Random Generators for All Hardnesses. *Journal of Computer and System Sciences*, 67(2):419–440, 2003.

[Zip79] Richard Zippel. Probabilistic Algorithms for Sparse Polynomials. In *Symbolic and Algebraic Computation, EUROSAM '79, An International Symposiumon Symbolic and Algebraic Computation*, volume 72 of *Lecture Notes in Computer Science*, pages 216–226. Springer, 1979.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il