If VNP Is Hard, Then so Are Equations for It

Mrinal Kumar
Indian Institute of Technology Bombay, India

C. Ramya
Chennai Mathematical Institute, India

Ramprasad Saptharishi
School of Technology and Computer Science, Tata Institute of Fundamental Research, Mumbai, India

Anamay Tengse
Department of Computer Science, University of Haifa, Israel

Abstract

Assuming that the Permanent polynomial requires algebraic circuits of exponential size, we show that the class VNP *does not* have efficiently computable equations. In other words, any nonzero polynomial that vanishes on the coefficient vectors of all polynomials in the class VNP requires algebraic circuits of super-polynomial size.

In a recent work of Chatterjee, Kumar, Ramya, Saptharishi and Tengse (FOCS 2020), it was shown that the subclasses of VP and VNP consisting of polynomials with bounded integer coefficients *do* have equations with small algebraic circuits. Their work left open the possibility that these results could perhaps be extended to all of VP or VNP. The results in this paper show that assuming the hardness of Permanent, at least for VNP, allowing polynomials with large coefficients does indeed incur a significant blow up in the circuit complexity of equations.

2012 ACM Subject Classification Theory of computation → Algebraic complexity theory

Keywords and phrases Computational Complexity, Algebraic Circuits, Algebraic Natural Proofs

Digital Object Identifier 10.4230/LIPIcs.STACS.2022.44

Related Version *Full Version*: https://arxiv.org/abs/2012.07056

Funding *C. Ramya*: Research supported by INSPIRE Faculty Fellowship of DST and by a grant from the Infosys Foundation. Part of this work was done when at the Tata Institute of Fundamental Research, Mumbai, India (DAE project 12-R&D-TFR-5.01-0500).
Ramprasad Saptharishi: Research was supported by the Department of Atomic Energy, Government of India, under project no. 12-R&D-TFR-5.01-0500, and also the SERB Ramanujan Fellowship.
Anamay Tengse: Research supported by the Israel Science Foundation (grant No. 716/20). Part of this work was done when at the Tata Institute of Fundamental Research, Mumbai, India, as a student (DAE project no. 12-R&D-TFR-5.01-0500), and later as a visitor (Prof. Prahladh Harsha's Swarnajaynti fellowship and Prof. Arkadev Chattopadhyay's Microsoft Research funds).

Acknowledgements We thank anonymous reviewers of an earlier version of this paper and Joshua Grochow, whose questions pointed us in the direction of this result. We also thank the reviewers of STACS 2022 for their helpful comments and suggestions.
We also thank Prerona Chatterjee and Ben Lee Volk for helpful discussions at various stages of this work.

1 Introduction

In the context of proving lower bounds in complexity theory, many of the existing approaches for proving Boolean circuit lower bounds were unified by Razborov and Rudich under the *Natural Proofs* framework [15] and they showed that, under standard cryptographic assumptions, any technique that fits into this framework cannot yield very strong lower

bounds. In the last few years there has been some work (e.g. [9], [10, 7]) aimed at developing an analogue of the Natural Proofs framework for algebraic circuit lower bounds. A crucial notion in this context is that of an *equation* for a class of polynomials which we now define.

For a class $\mathcal{C}$ of polynomials, an *equation* for $\mathbf{C}$ is a family of nonzero polynomials such that it vanishes on the coefficient vector of polynomials in $\mathcal{C}$.[1] Informally, an *algebraic natural proof* for a class $\mathcal{C}$ is a family of *equations* for $\mathcal{C}$ which can be computed by algebraic circuits of size and degree polynomially bounded in their number of variables. Thus, a lower bound for $\mathcal{C}$ can be proved by exhibiting an explicit polynomial on which an equation for $\mathcal{C}$ does not vanish.

Many of the known algebraic circuit lower bounds fit into this framework of algebraically natural proofs as observed by several authors [1, 9, 7, 10], thereby motivating the question of understanding whether techniques in this framework can yield strong algebraic circuit lower bounds; in particular, whether such techniques are sufficient to separate VNP from VP. In a recent breakthrough, Limaye, Srikanth and Tavenas [13] proved super-polynomial lower bounds against constant depth arithmetic circuits over fields of characteristic zero or large. It is not difficult to observe that this lower bound fits well into the natural proofs framework as it uses a complexity measure that is based on the rank of certain matrices to prove lower bounds. Thus, in the natural proofs framework, the first step towards a lower bound for VP is to understand whether VP has a family of equations which itself is in VP, that is its degree and its algebraic circuit size are polynomially bounded in the number of the variables. The next step, of course, would be to show the existence of a polynomial family in VNP which *does not* satisfy this family of equations. This work is motivated by the first step of this framework, that is the question of understanding whether natural and seemingly rich circuit classes like VP and VNP can have efficiently constructible equations. We briefly discuss prior work on this problem, before describing our results.

1.1 Complexity of Equations for classes of polynomials

In one of the first results on this problem, Forbes, Shpilka and Volk [7] and Grochow, Kumar, Saks and Saraf [10] observe that the class VP does not have efficiently constructible equations if we were to believe that there are hitting set generators for algebraic circuits with sufficiently succinct descriptions. However, unlike the results of Razborov and Rudich [15], the plausibility of the pseudorandomness assumption in [7, 10] is not very well understood. The question of understanding the complexity of equations for VP, or in general any natural class of algebraic circuits, continues to remain open.

In a recent work of Chatterjee, Kumar, Ramya, Saptharishi and Tengse [4], it was shown that if we focus on the subclass of VP (in fact, even VNP) consisting of polynomial families with bounded integer coefficients, then we indeed have efficiently computable equations. More formally, the main result in [4] was the following.

▶ **Theorem 1** ([4]). *For every constant $c > 0$, there is a polynomial family $\{P_{N,c}\} \in \mathsf{VP}_{\mathbb{Q}}$ [2] such that for all large n and $N = \binom{n+n^c}{n}$, the following are true.*

- *For every family $\{f_n\} \in \mathsf{VNP}_{\mathbb{Q}}$, where f_n is an n-variate polynomial of degree at most n^c and coefficients in $\{-1, 0, 1\}$, we have*

$$P_{N,c}(\overrightarrow{\mathrm{coeff}}(f_n)) = 0\,.$$

[1] Strictly speaking, these notions need us to work with families of polynomials, even though we sometimes drop the word *family* for ease of exposition.

[2] For a field $\mathbb{F}$, $\mathsf{VP}_{\mathbb{F}}$ denotes the class VP where the coefficients of the polynomials are from the field $\mathbb{F}$. Similarly, $\mathsf{VNP}_{\mathbb{F}}$ denotes the class VNP where the coefficients of the polynomials are from the field $\mathbb{F}$.

- *There exists a family* $\{h_n\}$ *of* n*-variate polynomials and degree at most* n^c *with coefficients in* $\{-1, 0, 1\}$ *such that*

$$P_{N,c}(\overrightarrow{\text{coeff}}(h_n)) \neq 0\,.$$

Here, $\overrightarrow{\text{coeff}}(f)$ *denotes the coefficient vector of a polynomial* f*.*

Many of the natural and well studied polynomial families like the Determinant, the Permanent, Iterated Matrix Multiplication, etc., have this property of bounded coefficients, and in fact the above result even holds when the coefficients are as large as poly(N). Thus, Theorem 1 could be interpreted as some evidence that perhaps we could still hope to prove lower bounds for one of these polynomial families via proofs which are algebraically natural. Extending Theorem 1 to obtain efficiently constructible equations for all of VP (or even for slightly weaker models like formulas or constant depth algebraic circuits) is an extremely interesting open question. In fact, even a conditional resolution of this problem in either direction, be it showing that the bounded coefficients condition in Theorem 1 can be removed, or showing that there are no such equations, would be extremely interesting and would provide much needed insight into whether or not there *is* a natural-proofs-like barrier for algebraic circuit lower bounds.

1.2 Our results

In this paper, we show that assuming the Permanent is hard, the constraint of bounded coefficients in Theorem 1 is necessary for efficient equations for VNP. More formally, we show the following theorem.

▶ **Theorem 2** (Conditional Hardness of Equations for VNP). *Let* $\varepsilon > 0$ *be a constant. Suppose, for an* m *large enough, we have that* Perm_m *requires circuits of size* 2^{m^ε}*.*

Then, for $n = m^{\varepsilon/4}$*, any* $d \leq n$ *and* $N = \binom{n+d}{n}$*, we have that every nonzero polynomial* $P(x_1, \ldots, x_N)$ *that vanishes on all coefficient vectors of polynomials in* $\mathsf{VNP}_{\mathbb{C}}(n, d)$ *has size at least* $2^{0.1n^4 - 3n}$*.*

▶ Remark. Our proof of the above theorem easily extends to any field of characteristic zero. We shall just work with the complex numbers for better readability.

Extending the result in Theorem 2 to hardness of equations for VP, even under the assumption that Permanent is sufficiently hard, is an extremely interesting open question. Such an extension would answer the main question investigated in [7, 10] and show a natural-proofs-like barrier for a fairly general family of lower bound proof techniques in algebraic complexity. Our proof of Theorem 2 however crucially relies on some of the properties of VNP and does not appear to extend to VP.

Although the proof of the above theorem is quite elementary, the main message (in our opinion) is that we do not[3] have compelling evidence to rule out, or accept, the efficacy of algebraic natural proofs towards proving strong lower bounds for rich classes of algebraic circuits.

1.3 An overview of the proof

As was observed in [7, 10], a lower bound for equations for a class of polynomials is equivalent to showing the existence of succinctly describable hitting sets for this class. For our proof we show that, assuming that the permanent is sufficiently hard, the coefficient

[3] Or rather, the results of [4] and the above theorem seem to provide *some* evidence for both sides!

vectors of polynomials in VNP form a *hitting set* for the class VP. The connection between hardness and randomness in algebraic complexity is well known via a result of Kabanets and Impagliazzo [11], and we use this connection, along with some additional ideas for our proof. We briefly describe a high level sketch of our proof in a bit more detail now.

Kabanets and Impagliazzo [11] showed that using any explicit polynomial family $\{f_n\}$ that is sufficiently hard, one can construct a hitting set generator for VP, that is, we can construct a polynomial map $\mathsf{Gen}_f : \mathbb{F}^k \to \mathbb{F}^t$ that "fools" any small algebraic circuit C on t variables in the sense that $C(y_1, y_2, \ldots, y_t)$ is nonzero if and only if the k-variate polynomial $C \circ \mathsf{Gen}_f$ is nonzero. In a typical invocation of this result, the parameter k is much smaller than t (typically $k = \operatorname{poly}\log t$). Thus, this gives a reduction from the question of polynomial identity testing for t-variate polynomials to polynomial identity testing for k-variate polynomials. Another related way of interpreting this connection is that if $\{f_n\}$ is sufficiently hard then Gen_f is a polynomial map whose image does not have an equation with small circuit size. Thus, assuming the hardness of the Permanent, this immediately gives us a polynomial map (with appropriate parameters) such that its image does not have an efficiently constructible equation.

For the proof of Theorem 2, we show that the points in the image of the map $\mathsf{Gen}_{\mathrm{Perm}}$, can be viewed as the coefficient vectors of polynomials in VNP, or, equivalently in the terminology in [7, 10], that the Kabanets-Impagliazzo hitting set generator is VNP-succinct. To this end, we work with a specific instantiation of the construction of the Kabanets-Impagliazzo generator where the underlying construction of combinatorial designs is based on Reed-Solomon codes. Although this is perhaps the most well known construction of combinatorial designs, there are other (and in some parameters, better) constructions known. However, our proof relies on the properties of this particular construction to obtain the succinct description. Our final proof is fairly short and elementary, and is based on extremely simple algebraic ideas and making generous use of the fact that we are trying to prove a lower bound for equations for VNP and not VP.

Proof Idea

Let us assume that for some constant $\varepsilon > 0$ and for all[4] $m \in \mathbb{N}$, Perm_m requires circuits of size 2^{m^ε}. Kabanets and Impagliazzo [11] showed that, for every combinatorial design $\mathcal{D}$ (a collection of subsets of a universe with small pairwise intersection) of appropriate parameters, the map

$$\mathsf{Gen}_{\mathrm{Perm}}(\mathbf{z}) = (\mathrm{Perm}(\mathbf{z}_S) \ : \ S \in \mathcal{D})$$

where $\mathbf{z}_S$ denotes the variables of in $\mathbf{z}$ restricted to the indices in S, is a hitting set generator for circuits of size $2^{o(m^\varepsilon)}$. Our main goal is to construct a polynomial $F(\mathbf{y}, \mathbf{z})$ in VNP such that

$$F(\mathbf{y}, \mathbf{z}) = \sum_{S \in \mathcal{D}} \mathrm{mon}_S(\mathbf{y}) \cdot \mathsf{Perm}(\mathbf{z}_S) \tag{1.1}$$

By choosing parameters carefully, this would immediately imply that any equation on N-variables, for $N = \binom{n+d}{d}$, that vanishes on the coefficient vector of polynomials in $\mathsf{VNP}(n, d)$ (which are n-variate polynomials in VNP of degree at most d) requires size super-polynomial in N.

[4] To be more precise, we should work with this condition for "infinitely often" $m \in \mathbb{N}$ and obtain that VNP does not have efficient equations infinitely often. We avoid this technicality for the sake of simplicity and the proof continues to hold for the more precise version with suitable additional care.

To show that the polynomial $F(\mathbf{y}, \mathbf{z})$ in Equation 1.1 is in VNP, we use a specific combinatorial design. For the combinatorial design $\mathcal{D}$ obtained via Reed-Solomon codes, every set in the design can be interpreted as a univariate polynomial g of appropriate degree over a finite field. The degree of g (say δ) and size of the finite field (say p) are related to the parameters of the design $\mathcal{D}$. Now,

$$F(\mathbf{y}, \mathbf{z}) = \sum_{\substack{g \in \mathbb{F}_p[v] \\ \deg(g) \leq \delta}} \left(\prod_{i=0}^{\delta} y_i^{g_i} \right) \cdot \mathrm{Perm}(\mathbf{z}_{S(g)}), \tag{1.2}$$

where $(g_0, \ldots, g_\delta)$ is the coefficient vector of the univariate polynomial g. Expressing $F(\mathbf{y}, \mathbf{z})$ in Equation 1.2 as a polynomial in VNP requires us to implement the product $\left(\prod_{i=0}^{\delta} y_i^{g_i} \right)$ as a polynomial when given the binary representation of coefficients $g_0, \ldots, g_\delta$ via a binary vector $\mathbf{t}$ of appropriate length (say r). This is done via the polynomial $\mathrm{Mon}(\mathbf{t}, \mathbf{y})$ in Subsection 3.1 in a straightforward manner. Furthermore, we want to algebraically implement the selection $\mathbf{z}_S$ for a set S in the combinatorial design when given the polynomial $\mathbf{g}$ corresponding to S. This is implemented via the polynomial $\text{RS-Design}(\mathbf{t}, \mathbf{z})$ in Subsection 3.2. Finally, we have

$$F(\mathbf{y}, \mathbf{z}) = \sum_{\mathbf{t} \in \{0,1\}^r} \mathrm{Mon}(\mathbf{t}, \mathbf{y}) \cdot \mathrm{Perm}(\text{RS-Design}(\mathbf{t}, \mathbf{z}))$$

which is clearly in VNP as Perm_p is in VNP and polynomials $\mathrm{Mon}(\mathbf{t}, \mathbf{y})$ and $\text{RS-Design}(\mathbf{t}, \mathbf{z})$ are efficiently computable. We refer the reader to Section 3 for complete details.

Related results

The concept of algebraically natural proofs was first studied in the works of Forbes, Shpilka and Volk [7] and Grochow, Kumar, Saks and Saraf [10] who showed that constructing efficient equations for a class directly contradicts a corresponding *succinct* derandomization of the polynomial identity testing problem. In fact, Forbes, Shpilka and Volk [7] unconditionally ruled out equations for depth-three multilinear formulas computable by certain structured classes of algebraic circuits using this connection. However, this does not imply anything about complexity of equations for general classes of algebraic circuits such as VP and VNP. In the context of proving algebraic circuit lower bounds, Efremenko, Garg, Oliveira and Wigderson [6] and Garg, Makam, Oliveira and Wigderson [8] explore limitations of proving algebraic circuit lower bounds via rank based methods. However, these results are not directly concerned with the complexity of equations for circuit classes.

Recently, Bläser, Ikenmeyer, Jindal and Lysikov [2] studied the complexity of equations in a slightly different context. They studied a problem called "matrix completion rank", a measure for tensors that is NP-hard to compute. Assuming $\mathsf{coNP} \not\subseteq \exists\mathsf{BPP}$, they construct an explicit tensor of large (border) completion rank such that any efficient equation for the class of tensors of small completion rank must necessarily also vanish on this tensor of large completion rank. That is, efficient equations cannot certify that this specific tensor has large (border) completion rank. Subsequently, this result was generalized to *min-rank* or *slice-rank* [3]. The set-up in these papers is different from the that in our paper, and that of [10, 7]. One way to interpret this difference is that [2] shows that "variety of small completion rank tensors" cannot be "cut out" by efficient equations, whereas the set-up of [10, 7] and our paper would ask if *every* equation for this variety requires large complexity.

In the context of equations for varieties in algebraic complexity, Kumar and Volk [12] proved polynomial degree bounds on the equations of the Zariski closure of the set of non-rigid matrices as well as small linear circuits over all large enough fields.

2 Preliminaries

2.1 Notation

- We use $[n]$ to denote the set $\{1, \ldots, n\}$ and $[\![n]\!]$ to denote the set $\{0, 1, \ldots, n\}$. We also use $\mathbb{N}_{\geq 0}$ to denote the set of non-negative integers.
- We use boldface letters such as $\mathbf{x}, \mathbf{y}$ to denote tuples, typically of variables. When necessary, we adorn them with a subscript such as $\mathbf{y}_{[n]}$ to denote the length of the tuple.
- We also use $\mathbf{x}^{\mathbf{e}}$ to denote the monomial $\prod x_i^{e_i}$. We write $\mathbf{x}^{\leq d}$ for the set of all monomials of degree at most d in $\mathbf{x}$, and $\mathbb{F}[\mathbf{x}]^{\leq d}$ for the set of polynomials in $\mathbf{x}$ over the field $\mathbb{F}$ of degree at most d.
- As usual, we identify the elements of $\mathbb{F}_p$ with $\{0, 1, \ldots, p-1\}$ and think of $[\![n]\!]$ as a subset of $\mathbb{F}_p$ in the natural way for any $n < p$.

2.2 Some basic definitions

Circuit classes

▶ **Definition 3** (Algebraic circuits). *An* algebraic circuit *is specified by a directed acyclic graph, with leaves (indegree zero; also called* inputs*) labelled by field constants or variables, and internal nodes labelled by $+$ or $\times$. The nodes with outdegree zero are called the* outputs *of the circuit. Computation proceeds in the natural way, where inductively each $+$ gate computes the sum of its children and each $\times$ gate computes the product of its children.*

The size *of the circuit is defined as the number of nodes in the underlying graph.*

▶ **Definition 4** (VP and VNP). *A family of polynomials $\{f_n\}$, where f_n is n-variate, is said to be in* VP *if* $\deg(f_n)$ *and the algebraic circuit complexity of f_n are bounded by a polynomial function of n. That is, there is a constant $c \geq 0$ such that for all large enough n we have* $\deg(f_n), \text{size}(f_n) \leq n^c$.

A family of polynomials $\{f_n\}$ is said to be in VNP *if there is a family $\{g_n(\mathbf{x}_{[n]}, \mathbf{y}_{[m]})\} \in$ VP such that m is bounded by a polynomial function of n and*

$$f_n(\mathbf{x}) = \sum_{\mathbf{y} \in \{0,1\}^m} g_n(\mathbf{x}, \mathbf{y}).$$

For some $n, d \in \mathbb{N}$, let $\mathcal{C}_{n,d}$ be a class of n-variate polynomials of *total* degree at most d. That is, $\mathcal{C}_{n,d} \subseteq \mathbb{F}[\mathbf{x}]^{\leq d}$. Similarly, we will use $\mathsf{VP}(n, d)$ and $\mathsf{VNP}(n, d)$ to denote the intersection of VP and VNP respectively, with $\mathbb{F}[\mathbf{x}_{[n]}]^{\leq d}$.

Equations and succinct hitting sets

▶ **Definition 5** (Equations for a class). *For $N = \binom{n+d}{n}$, a nonzero polynomial $P_N(\mathbf{Z})$ is called an* equation *for $\mathcal{C}_{n,d}$ if for all $f(\mathbf{x}) \in \mathcal{C}_{n,d}$, we have that $P_N(\overrightarrow{\text{coeff}}(f)) = 0$, where $\overrightarrow{\text{coeff}}(f)$ is the coefficient vector of f.*

Alternatively, we also say that a polynomial $P_N(\mathbf{Z})$ *vanishes* on the coefficient vectors of polynomials in class $\mathcal{C}$ if $P_N(\overrightarrow{\text{coeff}}(f)) = 0$ for all $f \in \mathcal{C}$.

▶ **Definition 6** (Hitting Set Generator (HSG)). *A polynomial map $G : \mathbb{F}^\ell \to \mathbb{F}^n$ given by $G(z_1, \ldots, z_\ell) = (g_1(\mathbf{z}), \ldots, g_n(\mathbf{z}))$ is said to be a* hitting set generator (HSG) *for a class $\mathcal{C} \subseteq \mathbb{F}[\mathbf{x}]$ of polynomials if for all nonzero $P \in \mathcal{C}$, $P \circ G = P(g_1, \ldots, g_n) \not\equiv 0$.*

We review the definition of *succinct hitting sets* introduced [10, 7].

▶ **Definition 7** (Succinct Hitting Sets for a class of polynomials [10, 7]). *For $N = \binom{n+d}{n}$, we say that a class of N-variate polynomials $\mathcal{D}_N$* has *$\mathcal{C}_{n,d}$-*succinct hitting sets *if for all nonzero $P_N(\mathbf{Z}) \in \mathcal{D}_N$, there exists some $f \in \mathcal{C}_{n,d}$ such that $P_N(\overrightarrow{\text{coeff}}(f)) \neq 0$.*

Hardness to randomness connection

For our proofs, we will need the following notion of combinatorial designs, which is a collection of subsets of a universe with small pairwise intersection.

▶ **Definition 8** (Combinatorial designs). *A family of sets $\{S_1, \ldots, S_N\} \subseteq [\ell]$ is said to be an (ℓ, m, n)-design if*

- *$|S_i| = m$ for each $i \in [N]$*
- *$|S_i \cap S_j| < n$ for any $i \neq j$.*

Kabanets and Impagliazzo [11] obtain hitting set generators from polynomials that are hard to compute for algebraic circuits. The following lemma is crucial to the proof of our main theorem.

▶ **Lemma 2.1** (HSG from Hardness [11]). *Let $\{S_1, \ldots, S_N\}$ be an (ℓ, m, n)-design and $f(\mathbf{x}_m)$ be an m-variate, individual degree d polynomial that requires circuits of size s. Then for fresh variables $\mathbf{y}_\ell$, the polynomial map* KI-gen$_{(N,\ell,m,n)}(f) : \mathbb{F}^\ell \to \mathbb{F}^N$ *given by*

$$(f(\mathbf{y}_{S_1}), \ldots, f(\mathbf{y}_{S_N})) \tag{2.2}$$

is a hitting set generator for all circuits of size at most $\left(\frac{s^{0.1}}{N(d+1)^n}\right)$.

3 Proof of the main theorem

Notation

1. For a vector $\mathbf{t} = (t_1, \ldots, t_r)$, we will use the short-hand $t_{i,j}^{(a)}$ to denote the variable $t_{(i \cdot a + j + 1)}$. This would be convenient when we consider the coordinates of $\mathbf{t}$ as blocks of length a.
2. For integers a, p, we shall use $\text{Mod}(a, p)$ to denote the unique integer $a_p \in [0, p-1]$ such that $a_p = a \bmod p$.

As mentioned in the overview, the strategy is to convert the hitting set generator given in (2.2) into a succinct hitting set generator. Therefore, we would like to associate the coordinates of (2.2) into coefficients of a suitable polynomial. That is, we would like to build a polynomial in VNP of the form

$$g(y_1, \ldots, y_\ell, z_1, \ldots, z_t) = \sum_{m \in \mathbf{y}^{\leq d}} m \cdot f(\mathbf{z}_{S_m})$$

with the monomials $m \in \mathbf{y}^{\leq d}$ suitably indexing into the sets of the combinatorial design. The above expression already resembles a VNP-definition and with a little care this can be made effective. We will first show that the different components of the above expression can be made succinct using the following constructions.

3.1 Building monomials from exponent vectors

For $n, r \in \mathbb{N}$, let $a = \lfloor r/n \rfloor$, and define $\mathrm{Mon}_{r,n}(\mathbf{t}, \mathbf{y})$ as follows.

$$\mathrm{Mon}_{r,n}(t_1, \ldots, t_r, y_1, \ldots, y_n) = \prod_{i=0}^{n-1} \prod_{j=0}^{a-1} \left(t_{i,j}^{(a)} y_{i+1}^{2^j} + (1 - t_{i,j}^{(a)}) \right)$$

The following observation is now immediate from the definition above.

▶ **Observation 9.** *For any* $(e_1, \ldots, e_n) \in [\![d]\!]^n$*, we have*

$$\mathrm{Mon}_{r,n}(\mathrm{Bin}(e_1), \ldots, \mathrm{Bin}(e_n), y_1, \ldots, y_n) = y_1^{e_1} \cdots y_n^{e_n},$$

where $\mathrm{Bin}(e)$ *is the tuple corresponding to the binary representation of* e*, and* $r = n \cdot \lceil \log_2 (d+1) \rceil$*. Furthermore, the polynomial* $\mathrm{Mon}_{r,n}$ *is computable by an algebraic circuit of size* $\mathrm{poly}(n, r)$*.*

3.2 Indexing Combinatorial Designs Algebraically

Next, we need to effectively compute the hard polynomial f on sets of variables in a combinatorial design, indexed by the respective monomials. We will need to simulate some computations modulo a fixed prime p. The following claim will be helpful for that purpose.

▷ Claim 10. For any $i, b, p \in \mathbb{N}_{\geq 0}$ with $i \leq p$, there exists a unique univariate polynomial $Q_{i,b,p}(v) \in \mathbb{Q}[v]$ of degree at most b such that

$$Q_{i,b,p}(a) = \begin{cases} 1 & \text{if } 0 \leq a < b \text{ and } a \equiv i \pmod{p}, \\ 0 & \text{if } 0 \leq a < b \text{ and } a \not\equiv i \pmod{p}. \end{cases}$$

Proof. We can define a unique univariate polynomial $Q_{i,b,p}(v)$ satisfying the conditions of the claim via interpolation to make a unique univariate polynomial take a value of 0 or 1 according to the conditions of the claim. Since, there are b conditions, there always exists such a polynomial of degree at most b. ◁

For any $n, b, p \in \mathbb{N}_{\geq 0}$ with $n \leq p$, define

$$\mathrm{Sel}_{n,b,p}(u_1, \ldots, u_n, v) \triangleq \sum_{i=1}^{n} u_i \cdot Q_{i,b,p}(v).$$

▶ **Observation 11.** *For any* $n, b, p \in \mathbb{N}_{\geq 0}$ *with* $n \leq p$*, for any* $0 \leq a < b$*, we have that*

$$\mathrm{Sel}_{n,b,p}(u_1, \ldots, u_n, a) = u_{\mathrm{Mod}(a,p)} = u_{a \bmod p}$$

The degree of $\mathrm{Sel}_{n,b,p}$ *is at most* $(b+1)$ *and can be computed by an algebraic circuit of size* $\mathrm{poly}(b)$*.*

Proof. From the definition of the univariate polynomial $Q_{i,b,p}(v)$ of degree b in Claim 10, $Q_{i,b,p}(a)$ outputs 1 if and only if $i = a \bmod p$. Hence, $\mathrm{Sel}_{n,b,p}(u_1, \ldots, u_n, a)$ is $u_{a \bmod p}$ and is of degree at most $(b+1)$. ◀

And finally, we choose a specific combinatorial design to instantiate Lemma 2.1.

3.3 Reed-Solomon-based combinatorial designs

For any prime p and any choice of $a \leq p$, the following is an explicit construction of a (p^2, p, a)-combinatorial design of size p^a, defined as follows:

> For every univariate polynomial $g(t) \in \mathbb{F}_p[t]$ of degree less than a, we add the set $S_g = \{(i, g(i)) \ : \ i \in \mathbb{F}_p\} \subseteq \mathbb{F}_p \times \mathbb{F}_p$ to the collection.

Since any two distinct univariate polynomials of degree less than a can agree on at most a points, it follows that the above is indeed a (p^2, p, a)-design.

The advantage of this specific construction is that it can be made succinct as follows. For $r = a \cdot \lfloor \log_2 p \rfloor$, let $t_1, \ldots, t_r$ be variables taking values in $\{0, 1\}$. The values assigned to $\mathbf{t}$-variables can be interpreted as a univariate over $\mathbb{F}_p$ of degree $< a$ by considering $\mathbf{t} \in \{0,1\}^r$ as a matrix with a rows and $\lfloor \log_2 p \rfloor$ columns each [5]. The binary vector in each row represents an element in $\mathbb{F}_p$. We illustrate this with an example.

$$\mathbf{t} = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{pmatrix} \longrightarrow \begin{pmatrix} 7 \\ 2 \\ 1 \\ 4 \\ 3 \end{pmatrix} \cong g(v)$$

For $p = 11$, $a = 5$, $g(v) = 7 + 2v + v^2 + 4v^3 + 2v^4 \in \mathbb{F}_{11}[v]$,

$\mathbf{t}$ is a 5×3 matrix that encodes the coefficients of $g(v)$.

Let $\mathbf{z}$ denote the p^2 variables $\{z_1, \ldots, z_{p^2}\}$, put in into a $p \times p$ matrix. Let S be a set in the Reed-Solomon-based (p^2, p, a)-combinatorial design. We want to implement the selection $\mathbf{z}_S$ algebraically. In the following, we design a vector of polynomials that outputs the vector of variables $\left(z^{(p)}_{0,g(0) \bmod p}, \ldots, z^{(p)}_{p-1,g(p-1) \bmod p}\right)$. Note that as mentioned above the polynomial g can be specified via variables $t_1, \ldots, t_r$. That is,

$$\begin{aligned} \text{RS-Design}_{p,a}(t_1, \ldots, t_r, z_1, \ldots, z_{p^2}) &\in (\mathbb{F}[\mathbf{t}, \mathbf{z}])^p \quad , \quad \text{for } r = a \cdot \lfloor \log_2 p \rfloor, \\ \text{RS-Design}_{p,a}(t_1, \ldots, t_r, z_1, \ldots, z_{p^2})_{i+1} &= \text{Sel}_{p,p^3,p}\left(z^{(p)}_{i,0}, \ldots, z^{(p)}_{i,p-1}, R_{i,a,p}(\mathbf{t})\right), \quad \forall i \in \mathbb{F}_p, \\ \text{where } R_{i,a,p}(\mathbf{t}) &= \sum_{j=0}^{a-1} \left[\left(\sum_{k=0}^{\ell_p - 1} t^{(\ell_p)}_{j,k} \cdot 2^k \right) \cdot \text{Mod}(i^j, p) \right], \\ \text{with } \ell_p &= \lfloor \log_2 p \rfloor . \end{aligned}$$

▶ **Observation 12.** *For any prime p, $a \leq p$, and $\mathbf{t} \in \{0,1\}^r$ for $r = a \cdot \lfloor \log_2 p \rfloor$, we have*

$$\text{RS-Design}_{p,a}(\mathbf{t}, \mathbf{z}) = \left(z_{i,g(i)} \ : \ i \in \mathbb{F}_p\right),$$

where $g(v) \in \mathbb{F}_p[v]$ is the univariate whose coefficient vector is represented by the bit-vector $\mathbf{t}$. Furthermore, the polynomial $\text{RS-Design}_{p,a}$ *is computable by an algebraic circuit of size* $\text{poly}(p)$.

[5] Working with $\lfloor \log_2 p \rfloor$ bits (as opposed to $\lceil \log_2 p \rceil$) makes the proofs much simpler, and does not affect the size of the design by much.

Proof. Fix some $\mathbf{t} \in \{0,1\}^r$. From the definition of $R_{i,a,p}(\mathbf{t})$, it is clear that $R_{i,a,p}(\mathbf{t})$ returns an integer α such that $g(i) = \alpha \bmod p$ where $\mathbf{t}$ encodes the coefficients of the polynomial $g(t)$ in binary. Furthermore, since $\mathrm{Mod}(i^j, p)$ is the unique integer $c \in [0, p-1]$ with $c = i^j \bmod p$, it also follows that $R_{i,a,p}(\mathbf{t})$ is an integer in the range $[0, p^3]$. Hence,

$$\mathrm{Sel}_{p,p^3,p}\left(z_{i,0}^{(p)}, \ldots, z_{i,p-1}^{(p)}, R_{i,a,p}(\mathbf{t})\right) = z_{i,g(i)}$$

as claimed. ◀

3.4 The VNP-Succinct-KI generator

We are now ready to show the VNP-succinctness of the Kabanets-Impagliazzo hitting set generator when using a hard polynomial from VNP and a Reed-Solomon-based combinatorial design.

For a prime p and for the largest number m such that $m^2 \leq p$, we will use $\mathrm{Perm}_{[p]} \in \mathbb{F}[\mathbf{y}_{[p]}]$ to denote Perm_m applied to the first m^2 variables of $\mathbf{y}$.

We now define the polynomial $F_{n,a,p}(\mathbf{y}_{[n]}, \mathbf{z}_{[p^2]})$ as follows.

$$F_{n,a,p}(y_1, \ldots, y_n, z_1, \ldots, z_{p^2}) = \sum_{\mathbf{t} \in \{0,1\}^r} \mathrm{Mon}_{r,n}(\mathbf{t}, \mathbf{y}) \cdot \mathrm{Perm}_{[p]}(\text{RS-Design}_{p,a}(\mathbf{t}, \mathbf{z})) \quad (3.1)$$
$$\text{where } r = a \cdot \lfloor \log_2 p \rfloor$$

It is evident from the above definition that the polynomial $F_{n,a,p}(\mathbf{y}, \mathbf{z})$ is in VNP for any p that is $\mathrm{poly}(n)$, when seen as a polynomial in $\mathbf{y}$-variables with coefficients from $\mathbb{C}[\mathbf{z}]$.

From the construction, we have that

$$F_{n,a,p}(y_1, \ldots, y_n, z_1, \ldots z_{p^2}) = \sum_{\mathbf{e}} \mathbf{y}^{\mathbf{e}} \cdot \mathrm{Perm}_{[p]}(\mathbf{z}_{S_{\mathbf{e}}}),$$

where $\{S_{\mathbf{e}}\}$ is an appropriate ordering of the Reed-Solomon-based (p^2, p, a)-combinatorial design of size p^a, described in Subsection 3.3. Note that we define the polynomial $F_{n,a,p}(\mathbf{y}, \mathbf{z})$ with three parameters n, a, p although for our purposes we will only use $a = n$.

3.5 Putting it all together

We are now ready to show that if the Permanent polynomial is exponentially hard, then any polynomial P that vanishes on the coefficient vectors of all polynomials in the class VNP requires super-polynomial size to compute it.

▶ **Theorem 2** (Conditional Hardness of Equations for VNP). *Let $\varepsilon > 0$ be a constant. Suppose, for an m large enough, we have that Perm_m requires circuits of size 2^{m^ε}.*

Then, for $n = m^{\varepsilon/4}$, any $d \leq n$ and $N = \binom{n+d}{n}$, we have that every nonzero polynomial $P(x_1, \ldots, x_N)$ that vanishes on all coefficient vectors of polynomials in $\mathsf{VNP}_{\mathbb{C}}(n, d)$ has size at least $2^{0.1n^4 - 3n}$.

Proof. Let p be the smallest prime larger than m^2; we know that $p \leq 2m^2$. We will again use $\mathrm{Perm}_{[p]} \in \mathbb{F}[\mathbf{y}_{[p]}]$ to denote Perm_m acting on the first m^2 variables of $\mathbf{y}$. Therefore, if Perm_m requires size 2^{m^ε} then so does $\mathrm{Perm}_{[p]}$.

Consider the polynomial $F_{n,n,p}(\mathbf{y}_{[n]}, \mathbf{z}_{[p^2]}) \in \mathsf{VNP}$ defined in (3.1), which we interpret as a polynomial in $\mathbf{y}$ with coefficients in $\mathbb{C}[\mathbf{z}]$. The individual degree in $\mathbf{y}$ is at least d, and at most p.

Let $F^{\leq d}_{n,n,p}(\mathbf{y}_{[n]}, \mathbf{z}_{[p^2]})$ denote the polynomial obtained from $F_{n,n,p}$ by discarding all terms whose total degree in $\mathbf{y}$ exceeds d. By standard homogenisation arguments, it follows that $F^{\leq d}_{n,n,p} \in \mathsf{VNP}$ as well. Therefore,

$$F^{\leq d}_{n,n,p}(\mathbf{y}, \mathbf{z}) = \sum_{\deg(\mathbf{y}^{\mathbf{e}}) \leq d} \mathbf{y}^{\mathbf{e}} \cdot \mathrm{Perm}_{[p]}(\mathbf{z}_{S_{\mathbf{e}}}),$$

where $S_{\mathbf{e}}$, for various $\mathbf{e}$, is an appropriate indexing into a (p^2, p, n)-combinatorial design of size N. Since the individual degree in $\mathbf{y}$ of $F_{n,n,p}$ was at least d, every coefficient of $F^{\leq d}_{n,n,p}$ is $\mathrm{Perm}_{[p]}(\mathbf{z}_S)$ for some S in the combinatorial design. In other words, the coefficient vector of $F^{\leq d}_{n,n,p}$ is precisely $\text{KI-gen}_{N,p^2,p,n}(\mathrm{Perm}_{[p]})$.

Suppose $P(x_1, \ldots, x_N)$ is a nonzero equation for $\mathsf{VNP}(n, d)$, then in particular it should be zero on the coefficient vector of $F^{\leq d}_{n,n,p}(\mathbf{y}, \mathbf{a}) \in \mathsf{VNP}$ for any $\mathbf{a} \in \mathbb{C}^{p^2}$. By the Polynomial Identity Lemma [14, 5, 17, 16], this implies that P must be zero on the coefficient vector of $F^{\leq d}_{n,n,p}(\mathbf{y}, \mathbf{z}) \in (\mathbb{C}[\mathbf{z}])[\mathbf{y}]$, where coefficients are formal polynomials in $\mathbb{C}[\mathbf{z}]$. Since the coefficient vector of $F^{\leq d}_{n,n,p}(\mathbf{y}, \mathbf{z})$ is just $\text{KI-gen}_{N,p^2,p,n}(\mathrm{Perm}_{[p]})$, the contrapositive of Lemma 2.1 gives that

$$\begin{aligned} & \mathrm{size}(P) > \frac{\mathrm{size}(\mathrm{Perm}_{[p]})^{0.1}}{N \cdot 2^n} > \frac{\mathrm{size}(\mathrm{Perm}_m)^{0.1}}{N \cdot 2^n} \\ \implies & \mathrm{size}(P) > \frac{2^{0.1m^{\varepsilon}}}{N \cdot 2^n} \end{aligned}$$

Since $N = \binom{n+d}{n} \leq 2^{2n}$, it follows that $\mathrm{size}(P)$ is at least at least $2^{0.1n^4 - 3n}$. ◀

Concluding that VNP has no efficient equations

Note that for a family $\{P_N\}$ to be a *family of equations* for a class $\mathcal{C}$, we want that for *all large enough* n, the corresponding polynomial P_N should vanish on the coefficient vectors of all n-variate polynomials in $\mathcal{C}$. This condition is particularly important if we want to use equations for $\mathcal{C}$ to prove lower bounds against it, since a family of polynomials $\{f_n\}$ is said to be computable in size $s(n)$ if $\mathrm{size}(f_n) \leq s(n)$ for *all large enough* n.

Theorem 2 shows that, for m large enough, if there is a constant $\varepsilon > 0$ such that $\mathrm{size}(\mathrm{Perm}_m) \geq 2^{m^{\varepsilon}}$, then for $n = m^{\varepsilon/4}$ and any $d \leq n$, the coefficient vectors of polynomials in $\mathsf{VNP}(n, d)$ form a hitting set for all N-variate polynomials (where $N = \binom{n+d}{d}$) of degree $\mathrm{poly}(N)$ that are computable by circuits of size $\mathrm{poly}(N)$. Now suppose the Permanent family is $2^{m^{\varepsilon}}$-hard for a constant $\varepsilon > 0$, which means that Perm_m is $2^{m^{\varepsilon}}$-hard for *infinitely many* $m \in \mathbb{N}$. Then using Theorem 2, we can conclude that for any family $\{P_N\} \in \mathsf{VP}$, we must have for *infinitely many* n that $P_N(\overrightarrow{\mathrm{coeff}}(f_n)) \neq 0$ for some $f_n \in \mathsf{VNP}$, which then shows that $\{P_N\}$ is not a family of equations for VNP.

4 Discussion and Open Problems

In the context of proving circuit lower bounds, and in relation to the notion of *algebraically natural proofs*, an interesting question that emerges from the recent work of Chatterjee, Kumar, Ramya, Saptharishi, Tengse [4] (stated in Theorem 1) is whether the condition of "small coefficients" is necessary for efficiently constructible equations to exist, especially for the class VP. While this question remains open for VP, our result shows that this additional restriction on the coefficients is essentially vital for the existence of efficiently constructible equations for the class VNP, and therefore provides strong evidence *against* the existence of efficient equations for VNP.

In light of Theorem 1 and Theorem 2 for VNP, one could make a case that equations for VP might also incur a super-polynomial blow up, without the restriction on coefficients. On the other hand, it could also be argued that an analogue of Theorem 2 may not be true for VP, since our proof crucially uses the fact that VNP is "closed under exponential sums". In fact, our proof essentially algebrizes the intuition that coefficient vectors of polynomials in VNP "look random" to a polynomial in VP, *provided that* VNP was exponentially more powerful than VP.

Thus, along with the previously known results on efficient equations for polynomials in VP with bounded coefficients, our result highlights that the existence of such equations for VP in general continues to remain an intriguing mystery.

Open Problems

We now conclude with some possible directions for extending our results.

- Perhaps the most interesting question here is to prove an analogue of Theorem 2 for equations for VP. This would provide concrete evidence for the possibility that we cannot hope to prove very strong lower bounds for algebraic circuits using proofs which proceed via efficiently constructible equations, from a fairly standard complexity theoretic assumption.
- At the moment, we cannot rule out the possibility of there being efficient equations for VP in general; it may be possible that the bounded coefficients condition in Theorem 1 can be removed. In particular, the question of proving upper bounds on the complexity of equations for VP is also extremely interesting, even if one proves such upper bounds under some reasonable complexity theoretic assumptions. A first step perhaps would be to prove upper bounds on the complexity of potentially simpler models, like formulas, algebraic branching programs or constant depth circuits. From the works of Forbes, Shpilka and Volk [7], we know that such equations for structured subclasses of VP (like depth-3 multilinear circuits) cannot be *too* simple (such as sparse polynomials, depth-3 powering circuits, etc.). Can we prove a non-trivial upper bound for equations for these structured classes within VP?
- Another question of interest would be to understand if the hardness assumption in Theorem 2 can be weakened further. For instance, is it true that VNP does not have efficiently constructible equations if VP $\neq$ VNP, or if Perm_n requires circuits of size $n^{\mathrm{poly}\log(n)}$? The current proof seems to need an exponential lower bound for the Permanent.

References

1 Scott Aaranson and Andrew Drucker. Arithmetic natural proofs theory is sought. Shtetl Optimized: Scott Aaranson's Blog, 2008. URL: https://www.scottaaronson.com/blog/?p=336.

2 Markus Bläser, Christian Ikenmeyer, Gorav Jindal, and Vladimir Lysikov. Generalized matrix completion and algebraic natural proofs. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*, pages 1193–1206, 2018. doi:10.1145/3188745.3188832.

3 Markus Bläser, Christian Ikenmeyer, Vladimir Lysikov, Anurag Pandey, and Frank-Olaf Schreyer. Variety membership testing, algebraic natural proofs, and geometric complexity theory. *CoRR*, abs/1911.02534, 2019. arXiv:1911.02534.

4 Prerona Chatterjee, Mrinal Kumar, C. Ramya, Ramprasad Saptharishi, and Anamay Tengse. On the existence of algebraically natural proofs. In Sandy Irani, editor, *61st IEEE Annual Symposium on Foundations of Computer Science, FOCS 2020, Durham, NC, USA, November 16-19, 2020*, pages 870–880. IEEE, 2020. arXiv:2004.14147, doi:10.1109/FOCS46700.2020.00085.

5 Richard A. DeMillo and Richard J. Lipton. A Probabilistic Remark on Algebraic Program Testing. *Information Processing Letters*, 7(4):193–195, 1978. doi:10.1016/0020-0190(78)90067-4.

6 Klim Efremenko, Ankit Garg, Rafael Oliveira, and Avi Wigderson. Barriers for rank methods in arithmetic complexity. In Anna R. Karlin, editor, *9th Innovations in Theoretical Computer Science Conference, ITCS 2018, January 11-14, 2018, Cambridge, MA, USA*, volume 94 of *LIPIcs*, pages 1:1–1:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.ITCS.2018.1.

7 Michael A. Forbes, Amir Shpilka, and Ben Lee Volk. Succinct hitting sets and barriers to proving lower bounds for algebraic circuits. *Theory of Computing*, 14(1):1–45, 2018. doi:10.4086/toc.2018.v014a018.

8 Ankit Garg, Visu Makam, Rafael Oliveira, and Avi Wigderson. More barriers for rank methods, via a "numeric to symbolic" transfer. In David Zuckerman, editor, *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 824–844. IEEE Computer Society, 2019. doi:10.1109/FOCS.2019.00054.

9 Joshua A. Grochow. Unifying known lower bounds via geometric complexity theory. *Computational Complexity*, 24(2):393–475, 2015. doi:10.1007/s00037-015-0103-x.

10 Joshua A. Grochow, Mrinal Kumar, Michael E. Saks, and Shubhangi Saraf. Towards an algebraic natural proofs barrier via polynomial identity testing. *CoRR*, abs/1701.01717, 2017. arXiv:1701.01717.

11 Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Computational Complexity*, 13(1-2):1–46, 2004. doi:10.1007/s00037-004-0182-6.

12 Mrinal Kumar and Ben Lee Volk. A polynomial degree bound on equations for non-rigid matrices and small linear circuits. In James R. Lee, editor, *12th Innovations in Theoretical Computer Science Conference, ITCS 2021, January 6-8, 2021, Virtual Conference*, volume 185 of *LIPIcs*, pages 9:1–9:9. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ITCS.2021.9.

13 Nutan Limaye, Srikanth Srinivasan, and Sébastien Tavenas. Superpolynomial lower bounds against low-depth algebraic circuits. *Electron. Colloquium Comput. Complex.*, page 81, 2021. URL: https://eccc.weizmann.ac.il/report/2021/081.

14 Øystein Ore. Über höhere kongruenzen. *Norsk Mat. Forenings Skrifter*, 1(7):15, 1922.

15 Alexander A. Razborov and Steven Rudich. Natural proofs. *Journal of Computer and System Sciences*, 55(1):24–35, 1997. doi:10.1006/jcss.1997.1494.

16 Jacob T. Schwartz. Fast Probabilistic Algorithms for Verification of Polynomial Identities. *Journal of the ACM*, 27(4):701–717, 1980. doi:10.1145/322217.322225.

17 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. doi:10.1007/3-540-09519-5_73.